

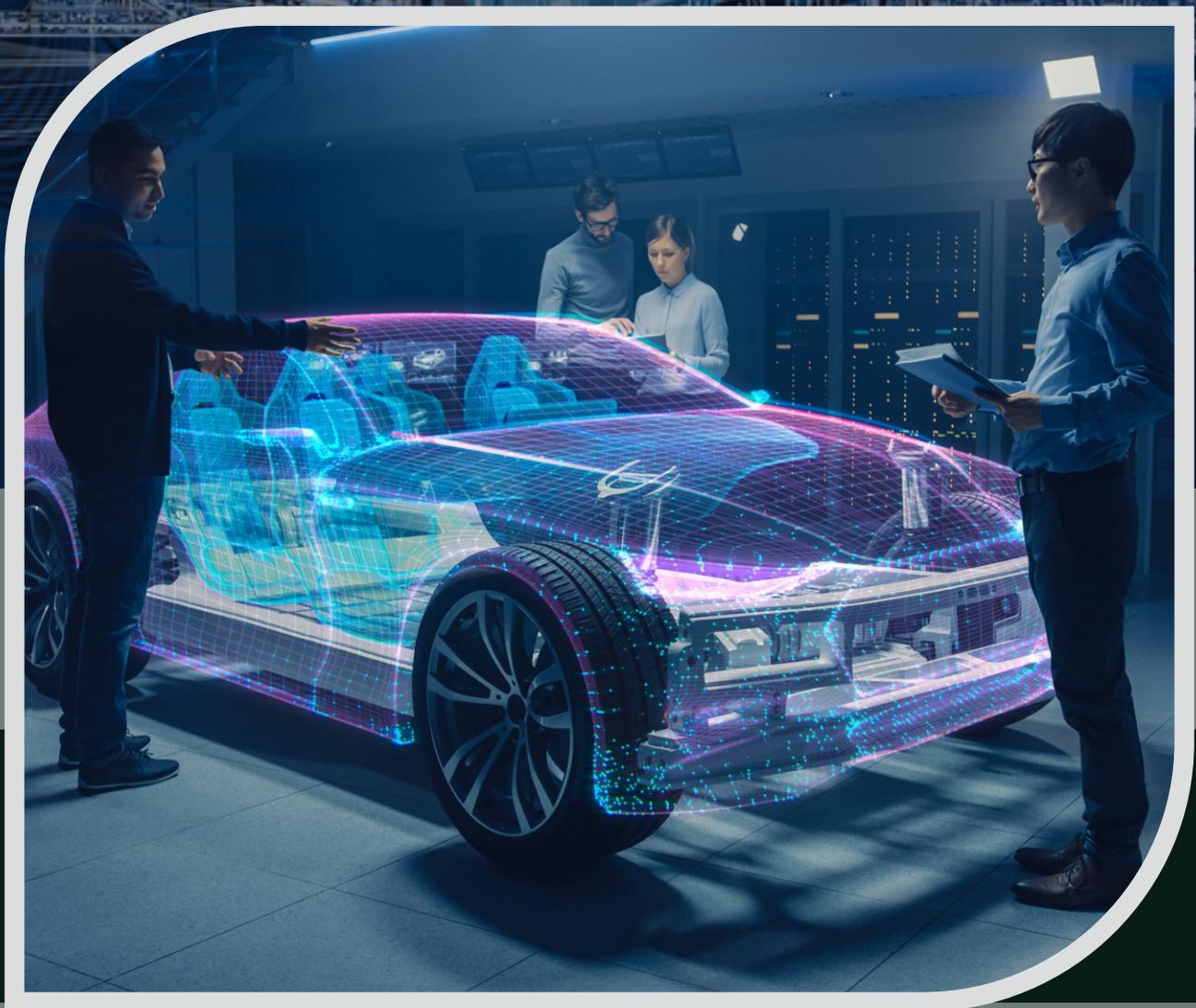
Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM

KEKUATAN AUGMENTED

dan

VIRTUAL REALITY

Dalam Bisnis JILID 1



YAYASAN PRIMA AGUS TEKNIK

KEKUATAN AUGMENTED dan VIRTUAL REALITY Dalam Bisnis JILID 1

Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM

BIO DATA PENULIS



Penulis memiliki berbagai disiplin ilmu yang diperoleh dari Universitas Diponegoro (UNDIP) Semarang. dan dari Universitas Kristen Satya Wacana (UKSW) Salatiga. Disiplin ilmu itu antara lain teknik elektro, komputer, manajemen dan ilmu sosiologi. Penulis memiliki pengalaman kerja pada industri elektronik dan sertifikasi keahlian dalam bidang Jaringan Internet, Telekomunikasi, Artificial Intelligence, Internet Of Things (IoT), Augmented Reality (AR), Technopreneurship, Internet Marketing dan bidang pengolahan dan analisa data (komputer statistik).

Penulis adalah pendiri dari Universitas Sains dan Teknologi Komputer (Universitas STEKOM) dan juga seorang dosen yang memiliki Jabatan Fungsional Akademik Lektor Kepala (Associate Professor) yang telah menghasilkan puluhan Buku Ajar ber ISBN, HAKI dari beberapa karya cipta dan Hak Paten pada produk IPTEK. Penulis juga terlibat dalam berbagai organisasi profesi dan industri yang terkait dengan dunia usaha dan industri, khususnya dalam pengembangan sumber daya manusia yang unggul untuk memenuhi kebutuhan dunia kerja secara nyata.



YAYASAN PRIMA AGUS TEKNIK

PENERBIT :

YAYASAN PRIMA AGUS TEKNIK

JL. Majapahit No. 605 Semarang

Telp. (024) 6723456. Fax. 024-6710144

Email : penerbit_ypat@stekom.ac.id

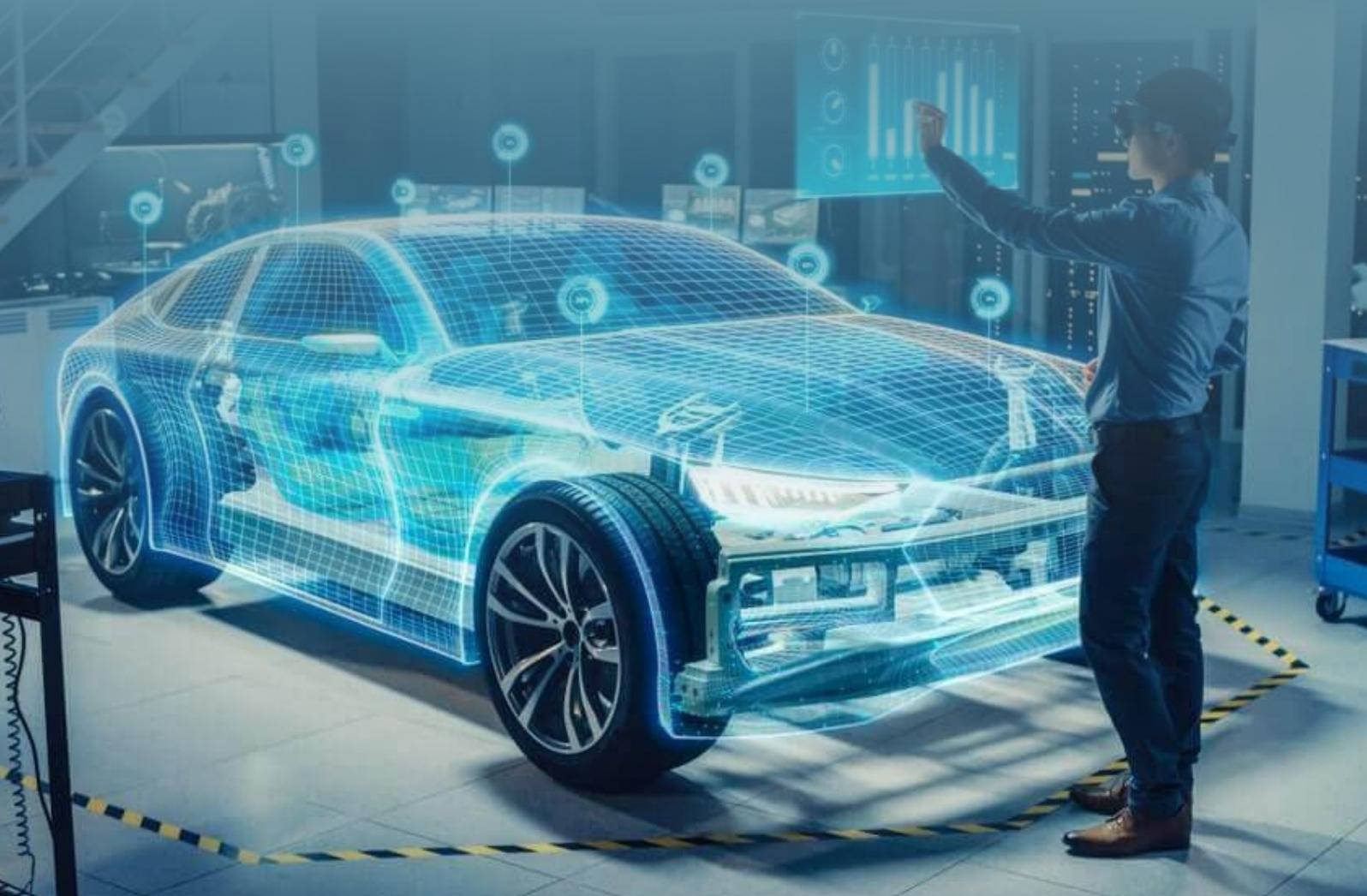
ISBN 978-623-5734-38-5 (jil.1)



9 786235 734385

KEKUATAN AUGMENTED dan VIRTUAL REALITY Dalam Bisnis JILID 1

Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM



YAYASAN PRIMA AGUS TEKNIK

PENERBIT :

YAYASAN PRIMA AGUS TEKNIK

JL. Majapahit No. 605 Semarang

Telp. (024) 6723456. Fax. 024-6710144

Email : penerbit_ypat@stekom.ac.id

KEKUATAN AUGMENTED dan VIRTUAL REALITY Dalam Bisnis Jilid 1

Penulis :

Dr. Ir. Agus Wibowo, M.Kom., M.Si., MM.

ISBN : 9 786235 734385

Editor :

Dr. Joseph Teguh Santoso, S.Kom., M.Kom.

Penyunting :

Dr. Mars Caroline Wibowo. S.T., M.Mm.Tech

Desain Sampul dan Tata Letak :

Irdha Yudianto, S.Ds., M.Kom.

Penebit :

Yayasan Prima Agus Teknik Bekerja sama dengan
Universitas Sains & Teknologi Komputer (Universitas STEKOM)

Redaksi :

Jl. Majapahit no 605 Semarang

Telp. (024) 6723456

Fax. 024-6710144

Email : penerbit_ypat@stekom.ac.id

Distributor Tunggal :

Universitas STEKOM

Jl. Majapahit no 605 Semarang

Telp. (024) 6723456

Fax. 024-6710144

Email : info@stekom.ac.id

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara
apapun tanpa ijin dari penulis

KATA PENGANTAR

Puji Syukur penulis panjatkan atas selesainya buku yang berjudul “*Kekuatan Augmented dan Virtual Reality dalam Bisnis – Jilid 1*”. Era revolusi industri 4.0 di tahun 2020 yang kebetulan pula adanya Pandemic Covid-19 ini mendorong semua kegiatan berbasis digital, dan bermunculan platform-platform digital yang mendukung perekonomian, salah satunya di bidang promosi dan pemasaran. Berkembangnya teknologi informasi di era digital ini membuat kegiatan bisnis harus memperhatikan segmentasi, sasaran dan kompetitor yang juga berubah sangat pesat. Hal ini membawa perubahan signifikan dalam pola kehidupan manusia dan bisnis.

Augmented Reality (AR) saat ini lebih banyak digunakan dalam pengolahan grafis komputer. Dengan dasar pemikiran untuk menyatukan antara imajinasi dan dunia nyata. Banyak diperoleh ide-ide untuk mempermudah pebisnis dalam menciptakan visualisasi objek yang lebih bagus, efisien, dan imajinatif untuk mendukung kegiatan bisnis mereka. Sistem ini berbeda dengan *Virtual Reality (VR)* yang sepenuhnya merupakan peralatan citra visual. *Augmented Reality* mengizinkan penggunaan untuk berinteraksi secara waktu-nyata dengan sistem.

AR dapat memberikan informasi detail kepada pelanggan dengan pengalaman yang berbeda saat mengenali produk. Selain itu, AR dan VR mampu mendukung komunikasi konsumen dengan perusahaan karena melalui teknologi tersebut calon pembeli merasa lebih tertarik dan ingin terus mencoba. Berbagai manfaat yang dihadirkan oleh teknologi AR dan VR merupakan output yang diinginkan dari suatu aktivitas komunikasi bisnis. AR dan VR memungkinkan aktivitas bisnis untuk menyampaikan pesan kepada pasar seperti melalui periklanan, merek, materi cetak, sponsorship dan lain-lain. Oleh sebab itulah penulis membuat buku ini agar pembaca atau pebisnis dapat beradaptasi terhadap era industry yang semakin berkembang pesat sejalan dengan teknologi informasi.

Buku ini terbagi dalam 2 Jilid. Pada buku jilid pertama terbagi dalam 6 Bagian dengan 13 Bab. Bagian Pertama terdiri dari 2 bab yaitu menjelaskan tentang awalmula teknologi *Augmented* dan *Virtual Reality* didesain untuk platform bisnis digital. Bagian kedua terdiri dari 2 Bab yang isinya bagaimana *Augmented* dan *Virtual Reality* merubah seni bisnis digital. Bagian ketiga hanya terdiri satu bab akan menjelaskan tentang peralatan dan spesifikasi yang mampu digunakan untuk membuat teknologi AR dan VR. Bagian keempat terdiri dari 3 bab yang menjelaskan tentang perangkat lunak yang digunakan serta arah pengembangannya. Bagian terakhir jilid pertama, pembaca akan disuguhkan teknologi AR dan VR yang diimplementasikan dalam beberapa kegiatan perusahaan hingga menganalisis pengalaman konsumen selama penggunaan teknologi ini.

Buku Jilid 2 terbagi dalam 3 Bagian pokok pembahasan. Bagian pertama pada jilid 2 pembaca akan dipandu melalui menginstall dan penjelasan semua software dan alat yang diperlukan untuk mengembangkan aplikasi *Windows Mixed Reality* dan *Unity*, *Unity* adalah platform software pilihan untuk mengembangkan *Windows Mixed Reality*. Bagian kedua buku jilid 2 ini terdiri dari 5 bab, pembaca akan mulai membangun pengalaman holografik. Disinilah pembaca dipandu melalui dasar-dasar membuat aplikasi *Mixed Reality* berfitur lengkap. Serta

mulai memanfaatkan kekuatan HoloLens dengan belajar tentang menggunakan *Spatial Mapping*, belajar tentang pentingnya *sound spatial* dan bagaimana menggunakannya dalam bisnis. Tiga bab terakhir dalam buku ini menjadi Bagian ketiga sekaligus menjadi bagian penutup Buku jilid 2 ini, memperkenalkan cara untuk mengoptimalkan dan menyempurnakan, mempublikasikan dan monitoring aplikasi, serta bergabung dengan komunitas holografik yang lebih luas untuk mendapatkan dukungan dan visibilitas. Juga memperkenalkan sumber daya yang tersedia untuk Anda, termasuk forum komunitas dan grup online yang relevan, event penting.

Akhir kata semoga buku ini berguna bagi para pembaca dan calon bisnis yang akan mengembangkan perusahaannya seiring dengan revolusi industri dan kemajuan teknologi informasi.

Semarang, Februari 2022

Penulis

Dr. Ir. Agus Wibowo, M.Kom., M.Si., M.M.

DAFTAR ISI

Halaman Judul	iii
Kata Pengantar	iv
Daftar Isi	v
BAGIAN I DESAIN DAN SENI DI REALITAS DIGITAL	
BAB 1 BAGAIMANA MANUSIA BERINTERAKSI DENGAN KOMPUTER	1
1.1 Pendahuluan	1
1.2 Modalitas Melalui Abad: Pra-Abad Kedua Puluh	1
1.3 Modalitas Melalui Zaman: Melalui Perang Dunia II	3
1.4 Modalitas Sepanjang Zaman: Miniaturisasi Komputer	4
1.5 Mengapa Kita Baru Melewati Semua Ini?	5
1.6 Modalitas visual	7
1.7 Modalitas fisik	9
1.8 Modalitas audio	10
1.9 Modalitas Baru	13
1.10 Input Suara, Tangan, dan Perangkat Keras untuk Generasi Berikutnya	17
1.11 Suara	18
1.12 Controller dan periferal fisik lainnya	21
BAB 2 MERANCANG UNTUK INDRA KITA, BUKAN UNTUK PERANGKAT KITA	25
2.1 Membayangkan Masa Depan	25
2.2 Teknologi Sensorik	26
2.3 Peran Masa Depan Desainer dan Tim	28
2.4 Peran Wanita dalam AI	30
2.5 Desain Sensorik	32
2.6 Lima Prinsip Sensorik	35
2.7 Kisah Adobe AR	37
2.8 Kesimpulan	40
BAGIAN II BAGAIMANA <i>EXTENDED REALITY</i> MENGUBAH SENI DIGITAL	
BAB 3 <i>VIRTUAL REALITY</i> (VR) UNTUK SENI	41
3.1 Cara Membuat Seni 3D yang Lebih Alami	42
3.2 VR untuk Animasi	45
BAB 4 OPTIMISASI 3D ART	46
4.1 Pengantar	47
4.2 Topologi	49
4.3 Baking	51
4.4 Menggambar Panggilan	53
4.5 Menggunakan Alat VR untuk Membuat Seni 3D	55
4.6 Memperoleh Model 3D Versus Membuatnya dari Awal	55
4.7 Ringkasan	57

BAGIAN III *HARDWARE, SLAM, TRACKING*

BAB 5 MEMBUAT <i>AUGMENTED REALITY (AR)</i> BEKERJA	58
5.1 Sejarah Singkat AR	60
5.2 Bagaimana dan Mengapa Memilih Platform AR	62
5.3 Kinerja Adalah Statistik	63
5.4 Mengintegrasikan Perangkat Keras dan Perangkat Lunak	65
5.5 Masa Depan Tracking	67
5.6 Masa Depan AR Computer Vision	68
5.7 Mapping	69
5.8 Platform	71
5.9 Pertimbangan Pengembangan Lainnya	73
5.10 Lighting	73
5.11 Multiplayer AR—Mengapa Cukup Sulit	74
5.12 Cloud AR	76
5.13 Gambaran Lebih Besar—Privasi dan Data AR Cloud	77
BAGIAN IV MEMBUAT <i>AUGMENTED REALITY (AR)</i> DAN <i>VIRTUAL REALITY (VR)</i>	
BAB 6 TEORI LINTAS PLATFORM <i>AUGMENTED REALITY</i> DAN <i>VIRTUAL REALITY</i>	78
6.1 Peran Mesin Game	78
6.2 Memahami Grafik 3D	80
6.3 Kamera Virtual	81
6.4 Pelajaran Portabilitas dari Desain Video Game	82
6.5 Ringkasan	84
BAB 7 <i>VIRTUAL REALITY TOOLKIT (VRTK): OPEN SOURCE FRAMEWORK</i> UNTUK	85
7.1 Apa Itu VRTK dan Mengapa Orang Menggunakannya?	85
7.2 Sejarah VRTK	86
7.3 Selamat datang di SteamVR Unity Toolkit	88
7.4 VRTK v4	89
7.5 Masa Depan VRTK	90
7.6 Keberhasilan VRTK	91
7.7 Memulai dengan VRTK 4	92
7.8 Menyiapkan proyek	94
7.9 Menjalankan tes	94
7.10 Contoh Scene	96
7.11 Cara "memeriksa" Repositori Contoh VRTK v4	98
7.12 Pengalih Scene	99
BAB 8 PENGEMBANGAN <i>AUGMENTED REALITY</i> DAN <i>VIRTUAL REALITY</i>	100
8.1 Berkembang untuk Virtual Reality dan Augmented Reality Itu Sulit	100
8.2 Penanganan Lokomosi	102
8.3 Gerakan linier (alias gerakan trackpad)	104
8.4 Teleport <i>Motion</i>	105
8.5 <i>Locomotion</i> di AR	106

8.6	Mode Raksasa (atau Semut)	108
8.7	Penggunaan Audio yang Efektif	109
8.8	Audio dalam VR	110
8.9	Paradigma Interaksi Umum	111
8.10	Inventaris untuk VR	113
8.11	Augmented Reality Raycasts	115
8.12	Kesimpulan	116
BAGIAN V MENINGKATKAN REPRESENTASI DATA		
VISUALISASI DATA ARTIFICIAL INTELLIGENCE DALAM KOMPUTASI SPASIAL		
BAB 9 DESAIN DAN PENGEMBANGAN VISUALISASI DATA DAN MACHINE		117
9.1	Pengantar	117
9.2	Memahami Visualisasi Data	118
9.3	Evolusi Desain Visualisasi Data dengan Munculnya XR	119
9.4	Data 2D dan 3D direpresentasikan dalam XR	121
9.5	Animasi	123
9.6	Kegagalan dalam Desain Visualisasi Data	125
9.7	Representasi Data, Infografis, dan Interaksi	125
9.8	Jenis visualisasi data	127
9.9	WebXR: Membangun Visualisasi Data untuk Web	128
9.10	Modul VR Pencitraan Medis Surg TVA	129
9.11	Holodek Medis—DICOM	130
9.12	Kesimpulan	133
BAB 10 PERILAKU DAN KARAKTER AI		134
10.1	Pengantar	134
10.2	Praktik Saat Ini: AI Reaktif	135
10.3	Adaptabilitas	135
10.4	Perencanaan pesanan sebagian	136
10.5	Perencanaan hierarkis	138
10.6	<i>Machine Learning</i>	140
10.7	<i>Reinforcement Learning</i>	141
10.8	<i>Deep Reinforcement Learning</i>	142
10.9	<i>Imitation Learning</i>	143
10.10	Menggabungkan Perencanaan Otomatis dan Pembelajaran Mesin	146
10.11	Kesimpulan	147
BAGIAN VI GUNAKAN KASUS DALAM EMBODIED REALITY		
BAB 11 TEKNOLOGI KESEHATAN AUGMENTED REALITY DAN VIRTUAL REALITY		150
11.1	Desain Aplikasi Teknologi Kesehatan VR/AR	150
11.2	UX Standar	152
11.3	Tutorial: Eksperimen Insight Parkinson	155
11.4	Analisis Dan Pelaporan Data	157
11.5	Pengalaman yang Dirancang untuk Pendidikan Kedokteran	159

11.6 Studi Kasus dari Institusi Akademik Terkemuka	162
BAB 12 PENGALAMAN PENGGUNA: SportsXR	168
12.1 Pengantar	171
12.2 Tidak Ada yang Langsung	173
12.3 Bagian 3: Membuat Masa Depan	174
12.4 Kepemilikan	177
12.5 Pikiran Terakhir	179
12.6 Kesimpulan	182
BAB 13 PENGGUNAAN VIRTUAL REALITY DALAM PELATIHAN PERUSAHAAN	182
13.1 Pendahuluan: Pentingnya Pelatihan Perusahaan	185
13.2 Kasus Penggunaan: Pelatihan Rumah Banjir	186
13.3 Grafik Komputer	188
13.4 Kasus Penggunaan: Pelatihan Keterampilan Lunak	191
13.5 Masa Depan: Fotogrametri	194
13.6 Masa Depan: Bidang Cahaya	198
13.7 Masa Depan: Pelatihan AR	203
13.8 Masa Depan: Pengenalan Suara	205
13.9 Masa Depan: Skenario Pelatihan Ideal	210
DAFTAR PUSTAKA	220

BAGIAN I

DESAIN DAN SENI DI REALITAS DIGITAL

Tapi ekspresi manusia melampaui bahasa. Desain dan seni mencerminkan apa yang mungkin tidak didefinisikan secara ringkas. Pola perilaku dunia yang tak terucapkan, tertulis besar, tercermin dalam desain yang sangat baik. Emosi dan pola sosial yang mengarahkan otak bawah sadar kita terungkap dalam seni: patung, tari, lukisan, dan musik. Namun hingga era digital, area ekspresi manusia ini, pada akhirnya, selalu terikat pada batasan fisik: fisika, materi nyata, dan waktu.

Pada 2019, kami jatuh ke situasi aneh lainnya. Seiring komputer menjadi lebih mobile, akhirnya dimungkinkan untuk membawa dunia digital ke dunia nyata. Sepintas, ini terlihat seperti prosedur yang relatif sederhana. Ada baiknya untuk berpikir bahwa kita dapat berinteraksi dengan komputer dengan cara yang realistis dan alami, hanya dengan cara yang meniru apa yang sudah kita ketahui.

Pada bagian ini, kami meninjau beberapa teka-teki yang terlibat dalam memindahkan komputer dari dua dimensi ke komputasi spasial nyata. Dalam Bab 1, Timoni West membahas sejarah interaksi manusia-komputer dan bagaimana kita sampai di tempat kita sekarang ini. Dia kemudian berbicara tentang di mana kita berada saat ini, baik untuk input manusia dan pemahaman komputer tentang dunia.

Dalam Bab 2, Silka Miesnieks, Kepala Desain Emerging Adobe, berbicara tentang konteks di mana kita melihat desain untuk berbagai realitas: bagaimana menjembatani kesenjangan antara bagaimana kita berpikir kita harus berinteraksi dengan komputer dan desain sensorik bersama yang nyata. Dia menyelidiki variabel manusia yang perlu kita perhitungkan dan bagaimana pembelajaran mesin akan berperan dalam meningkatkan komputasi spasial.

Ada banyak hal yang tidak kami bahas dalam bab ini: praktik terbaik khusus untuk standar seperti skala dunia, atau pemetaan tombol, atau sistem desain. Terus terang, itu karena kami berharap mereka sudah ketinggalan zaman pada saat buku ini diterbitkan. Kami tidak ingin mengkanonisasi apa yang mungkin terkait dengan serangkaian tombol atau input yang bahkan mungkin tidak ada dalam lima tahun. Meskipun mungkin ada manfaat sejarah untuk mencatatnya, bukan itu inti dari bab-bab ini.

BAB 1

BAGAIMANA MANUSIA BERINTERAKSI DENGAN KOMPUTER

Dalam bab ini, kita mengeksplorasi hal-hal berikut:

- Latar belakang sejarah modalitas manusia-komputer
- Deskripsi modalitas umum dan pro dan kontra mereka
- Siklus umpan balik antara manusia dan komputer
- Pemetaan modalitas ke input industri saat ini

Pandangan holistik dari siklus umpan balik dari desain imersif yang baik

Definisi Istilah Umum Saya menggunakan istilah-istilah berikut dengan cara-cara khusus yang mengasumsikan elemen yang dapat dipahami manusia:

Modality

Saluran input dan output sensorik antara komputer dan manusia

Affordances

Atribut atau karakteristik suatu objek yang menentukan kegunaan potensial objek itu

Input

Bagaimana Anda melakukan hal-hal itu; data yang dikirim ke komputer

Output

Reaksi yang dapat dipahami terhadap suatu peristiwa; data yang dikirim dari komputer

Feedback

Jenis output; konfirmasi bahwa apa yang Anda lakukan diperhatikan dan ditindaklanjuti oleh pihak lain

1.1 PENDAHULUAN

Dalam permainan Dua Puluh Pertanyaan, tujuan Anda adalah menebak objek apa yang sedang dipikirkan orang lain. Anda dapat menanyakan apa pun yang Anda inginkan, dan orang lain harus menjawab dengan jujur; tangkapannya adalah mereka menjawab pertanyaan hanya dengan menggunakan satu dari dua pilihan: ya atau tidak.

Melalui serangkaian kebetulan dan interpolasi, cara kita berkomunikasi dengan komputer konvensional sangat mirip dengan Dua Puluh Pertanyaan. Komputer berbicara dalam biner, satu dan nol, tetapi manusia tidak. Komputer tidak memiliki indra yang melekat pada dunia atau, memang, apa pun di luar biner—atau, dalam kasus komputer kuantum, probabilitas.

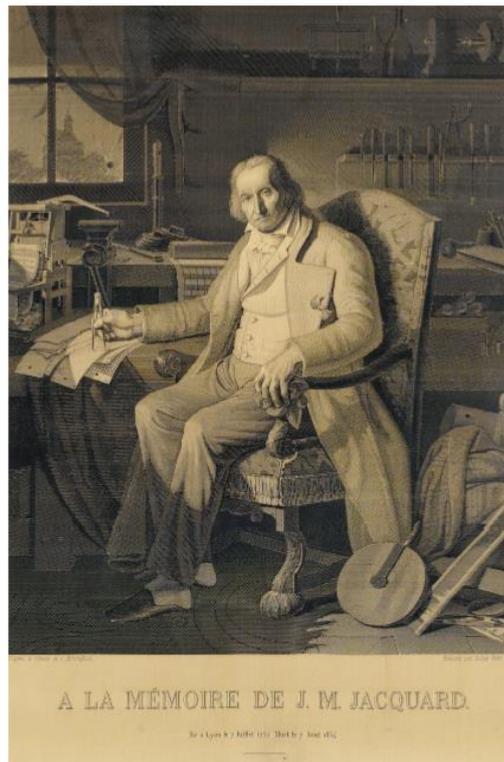
Karena itu, kami mengomunikasikan segalanya ke komputer, mulai dari konsep hingga input, melalui peningkatan level abstraksi ramah-manusia yang menutupi lapisan komunikasi dasar: satu dan nol, atau ya dan tidak.

Jadi, sebagian besar pekerjaan komputasi saat ini adalah menentukan bagaimana membuat manusia dengan mudah dan sederhana menjelaskan ide-ide yang semakin kompleks ke komputer. Pada gilirannya, manusia juga berupaya agar komputer memproses

ide-ide itu lebih cepat dengan membangun lapisan abstraksi di atas satu dan nol. Ini adalah siklus input dan output, keterjangkauan dan umpan balik, di seluruh modalitas. Lapisan abstraksi dapat mengambil banyak bentuk: metafora User interface grafis, kata-kata yang diucapkan dari pemrosesan bahasa alami (NLP), pengenalan objek dari visi komputer, dan, yang paling sederhana dan umum, input keyboard dan pointer sehari-hari, yang kebanyakan manusia digunakan untuk berinteraksi dengan komputer setiap hari.

1.2 MODALITAS MELALUI ABAD: PRA-ABAD KEDUA PULUH

Untuk memulai, mari kita bahas secara singkat bagaimana manusia secara tradisional memberikan instruksi kepada mesin. Mesin proto-komputasi paling awal, alat tenun yang dapat diprogram, kartu punch "membaca" yang terkenal. Joseph Jacquard menciptakan apa yang sebenarnya merupakan salah satu karya seni mekanik sejati pertama, potret dirinya sendiri, menggunakan kartu punch pada tahun 1839 (Gambar 1-1). Sekitar waktu yang sama di Rusia, Semyon Korsakov menyadari bahwa kartu punch dapat digunakan untuk menyimpan dan membandingkan kumpulan data.



Gambar 1-1. Potret sutra tenunan Joseph Jacquard, 1839, yang menggunakan lebih dari 24.000 kartu berlubang untuk membuat potret

Kartu punch dapat menyimpan sejumlah besar data, selama data tersebut cukup konsisten untuk dibaca oleh mesin. Dan meskipun pena dan alat genggam serupa sangat bagus untuk tugas-tugas tertentu, memungkinkan manusia untuk mengekspresikan informasi dengan cepat, rata-rata otot lengan dan jari manusia tidak memiliki kemampuan untuk secara konsisten menghasilkan bentuk yang hampir identik sepanjang waktu.

Ini telah lama menjadi masalah yang diketahui. Bahkan, sejak abad ketujuh belas—yaitu, segera setelah teknologi tersedia—orang mulai membuat keyboard. Orang-orang menemukan dan menemukan kembali keyboard untuk berbagai alasan; misalnya, untuk melawan pemalsuan, membantu saudara perempuan yang buta, dan buku-buku yang lebih baik. Memiliki bidang yang mendukung untuk mengistirahatkan tangan dan pergelangan tangan memungkinkan gerakan yang tidak konsisten untuk menghasilkan hasil yang konsisten yang tidak mungkin dicapai dengan pena.

Seperti disebutkan sebelumnya, proto-komputer memiliki motivasi yang sama menariknya: komputer membutuhkan data fisik yang sangat konsisten, dan tidak nyaman bagi manusia untuk membuat data yang konsisten. Jadi, meskipun mungkin tampak mengejutkan dalam retrospeksi, pada awal 1800-an, mesin punch-card, belum menjadi monster perhitungan, sudah memiliki keyboard yang terpasang padanya, seperti yang digambarkan pada Gambar 1-2.



Gambar 1-2. Pemotong Kartu Jacquard Masson Mills WTM 10, 1783, yang digunakan untuk membuat kartu berlubang yang dibaca oleh alat tenun Jacquard

Keyboard telah dilampirkan ke perangkat komputasi sejak awal, tetapi, tentu saja, mereka berkembang menjadi mesin tik sebelum mengulang kembali saat kedua teknologi tersebut bergabung. Kesabaran juga terkait dengan konsistensi dan kelelahan manusia. Dari Wikipedia:

Pada pertengahan abad ke-19, peningkatan kecepatan komunikasi bisnis telah menciptakan kebutuhan akan mekanisasi proses penulisan. Stenografer dan telegrafer dapat mencatat informasi dengan kecepatan hingga 130 kata per menit.

Menulis dengan pena, sebaliknya, hanya menghasilkan sekitar 30 kata per menit: penekanan tombol tidak dapat disangkal merupakan solusi alfanumerik yang lebih baik.

Abad berikutnya dihabiskan untuk mencoba menyempurnakan konsep dasar. Fitur-fitur selanjutnya, seperti penambahan tombol shift, secara substansial meningkatkan dan merampingkan desain dan ukuran mesin tik awal.

Saya ingin berhenti sejenak di sini untuk menunjukkan masalah yang lebih luas yang coba dipecahkan oleh semua orang dengan menggunakan mesin tik, dan khususnya dengan keyboard sebagai input: pada tingkat tertinggi, orang ingin menangkap ide mereka dengan lebih cepat dan lebih akurat. Ingat ini; itu adalah tema yang konsisten di semua peningkatan modalitas.

1.3 MODALITAS MELALUI ZAMAN: MELALUI PERANG DUNIA II

Begitu banyak untuk keyboard, yang, seperti yang baru saja saya tunjukkan, telah bersama kita sejak awal manusia mencoba berkomunikasi dengan mesin mereka. Sejak awal abad kedua puluh—yaitu, sekali lagi, segera setelah pengerjaan logam dan teknik manufaktur mendukungnya—kami memberi mesin cara untuk berkomunikasi kembali, untuk berdialog dengan operator mereka sebelum tahap output fisik yang mahal: monitor dan display, a bidang yang diuntungkan dari penelitian dan sumber daya yang signifikan melalui era perang melalui anggaran militer.

Tampilan komputer pertama tidak menunjukkan kata-kata: panel komputer awal memiliki bola lampu kecil yang akan menyala dan mati untuk mencerminkan keadaan tertentu, memungkinkan para insinyur untuk memantau status komputer—dan mengarah pada penggunaan kata "monitor". Selama Perang Dunia II, badan-badan militer menggunakan layar tabung sinar katoda (CRT) untuk cakupan radar, dan segera setelah perang, CRT mulai hidup mereka sebagai vektor, dan kemudian teks, tampilan komputasi untuk kelompok-kelompok seperti SAGE dan Angkatan Laut Kerajaan.



Gambar 1-3. Contoh antarmuka komputer awal untuk pemetaan ulang proprioseptif; Operator radar WAAF Denise Miley sedang merencanakan pesawat di Ruang Penerima di stasiun Bawdsey "Chain Home" pada Mei 1945 (perhatikan kenop besar di sebelah kirinya, kontrol goniometer yang memungkinkan Miley mengubah sensitivitas pencari arah radio)

Segera setelah mesin komputasi dan pemantau memiliki tampilan, kami memiliki input khusus tampilan untuk menyertainya. Joystick diciptakan untuk pesawat, tetapi penggunaannya untuk piloting pesawat jarak jauh dipatenkan di Amerika Serikat pada tahun 1926. Ini menunjukkan keanehan yang aneh dari fisiologi manusia: kita dapat secara naluriah memetakan kembali proprioception—indera kita tentang orientasi dan penempatan tubuh kita— ke volume baru dan sudut bidang (lihat Gambar 1-3). Jika kami tidak dapat melakukannya, tidak mungkin menggunakan mouse di desktop pada bidang-Z untuk memindahkan jangkar mouse pada X. Namun, kami dapat melakukannya hampir tanpa berpikir—walaupun beberapa dari kami mungkin perlu membalikkan rotasi sumbu untuk meniru pemetaan internal kita sendiri.

Modalitas Sepanjang Zaman: Pasca Perang Dunia II

Joystick dengan cepat dipindahkan dari pesawat dan di samping tampilan radar dan sonar selama Perang Dunia II. Segera setelah perang, pada tahun 1946, input khusus tampilan pertama ditemukan. Ralph Benjamin, seorang insinyur di Angkatan Laut Kerajaan, menganggap rollerball sebagai alternatif dari input joystick yang ada: “Pelacak bola yang elegan berdiri di samping tampilan arah pesawatnya. Dia memiliki satu bola, yang dia pegang di tangannya, tetapi joysticknya telah layu.” Indikasinya adalah bahwa rollerball dapat dipegang dengan tangan daripada diletakkan di atas meja. Namun, kenyataan manufaktur pada tahun 1946 berarti bahwa roller asli adalah bola bowling berukuran penuh. Tidak mengherankan, rollerball seberat 10 pon tidak menggantikan joystick.

Ini membawa kita ke lima aturan popularitas input komputer. Untuk lepas landas, input harus memiliki karakteristik sebagai berikut:

- Murah
- Dapat diandalkan
- Nyaman
- Memiliki perangkat lunak yang memanfaatkannya
- Memiliki tingkat kesalahan pengguna yang dapat diterima

Yang terakhir dapat diamortisasi dengan desain perangkat lunak yang baik yang memungkinkan tindakan tidak merusak, tetapi berhati-hatilah: setelah titik tertentu, bahkan kesalahan yang tidak berbahaya dapat mengganggu. Koreksi otomatis pada layar sentuh adalah contoh yang bagus dari kesalahan pengguna yang sering kali melampaui kemampuan perangkat lunak.

Meskipun mouse rollerball tidak akan mencapai mana-mana sampai tahun 1984 dengan munculnya komputer pribadi, banyak jenis input lain yang digunakan dengan komputer pindah dari militer melalui pertengahan 1950-an dan ke sektor swasta: joystick, tombol dan matikan, dan, tentu saja, keyboard.

Mungkin mengejutkan mengetahui bahwa stylus sudah ada sebelum mouse. Pena ringan, atau pistol, yang dibuat oleh SAGE pada tahun 1955, adalah stylus optik yang diatur waktunya untuk siklus penyegaran CRT dan dapat digunakan untuk berinteraksi langsung di monitor. Opsi mirip mouse lainnya, Grafacon Perusahaan Peralatan Data, menyerupai blok

pada poros yang dapat diayunkan untuk memindahkan kursor. Bahkan ada pekerjaan yang dilakukan pada perintah suara pada awal tahun 1952 dengan sistem Audrey Bell Labs, meskipun hanya mengenali 10 kata.

Pada tahun 1963, perangkat lunak grafis pertama ada yang memungkinkan pengguna untuk menggambar di monitor TX-2 MIT Lincoln Laboratory, Sketchpad, yang dibuat oleh Ivan Sutherland di MIT. GM dan IBM memiliki usaha patungan serupa, Design Augmented by Computer, atau DAC-1, yang menggunakan layar kapasitansi dengan pensil logam, sebagai gantinya—lebih cepat daripada light pen, yang mengharuskan CRT untuk di-refresh.

Sayangnya, baik dalam pena ringan dan kotak pensil logam, tampilannya tegak sehingga pengguna harus mengangkat tangan mereka untuk input—yang kemudian dikenal sebagai “lengan gorila” yang terkenal itu. Latihan yang bagus, tetapi ergonomi yang buruk. Perusahaan RAND telah memperhatikan masalah ini dan telah mengerjakan solusi tablet dan stylus selama bertahun-tahun, tetapi itu tidak murah: pada tahun 1964, stylus RAND—yang membingungkan, kemudian juga dipasarkan sebagai Grafacon—berharga sekitar IDR 180 juta (kira-kira IDR 15M dalam rupiah 2018). Itu bertahun-tahun sebelum kombinasi tablet-dan-stylus akan lepas landas, jauh setelah mouse dan sistem User interface grafis (GUI) dipopulerkan.

Pada tahun 1965, Eric Johnson, dari Royal Radar Establishment, menerbitkan makalah tentang perangkat layar sentuh kapasitif dan menghabiskan beberapa tahun berikutnya untuk menulis kasus penggunaan yang lebih jelas tentang topik tersebut. Itu diambil oleh para peneliti di Organisasi Eropa untuk Penelitian Nuklir (CERN), yang menciptakan versi kerja pada tahun 1973.

Pada tahun 1968, Doug Engelbart siap menunjukkan pekerjaan yang telah dilakukan labnya, Augmentation Research Center, di Stanford Research Institute sejak 1963. Di sebuah aula di bawah Civic Center San Francisco, dia mendemonstrasikan Sistem oNLine (NLS) timnya dengan sejumlah fitur sekarang standar dalam komputasi modern: kontrol versi, jaringan, konferensi video, email multimedia, banyak jendela, dan integrasi mouse yang berfungsi, di antara banyak lainnya. Meskipun NLS juga membutuhkan keyboard chord dan keyboard konvensional untuk input, mouse sekarang sering disebut-sebut sebagai salah satu inovasi kunci. Faktanya, mouse NLS memiliki peringkat yang sama dapat digunakan dengan pena ringan atau sistem input lutut milik ARC dalam penelitian tim Engelbart sendiri. Juga tidak unik: pabrikan radio dan TV Jerman, Telefunken, merilis mouse dengan RKS 100-86, Rollkugel, yang sebenarnya diproduksi komersial pada tahun Engelbart mengumumkan prototipenya.

Namun tentu saja, Engelbart mempopulerkan gagasan input komputer bentuk bebas asimetris. Perancang mouse yang sebenarnya di ARC, Bill English, juga menunjukkan salah satu kebenaran modalitas digital pada akhir penelitiannya tahun 1967, “Teknik Pemilihan Tampilan untuk Manipulasi Teks”:

[Saya] tampaknya tidak realistis untuk mengharapkan pernyataan datar bahwa satu perangkat lebih baik dari yang lain. Rincian sistem penggunaan di mana perangkat akan disematkan membuat terlalu banyak perbedaan.

Tidak peduli seberapa bagus perangkat kerasnya, aspek yang paling penting adalah bagaimana perangkat lunak menginterpretasikan input perangkat keras dan menormalkannya untuk maksud pengguna.

Untuk mengetahui lebih lanjut tentang bagaimana desain perangkat lunak dapat memengaruhi persepsi pengguna terhadap input, saya sangat merekomendasikan buku *Game Feel: A Game Designer's Guide to Virtual Sensation* oleh Steve Swink. Karena setiap game memiliki dunia dan sistemnya sendiri, "rasa" dari input dapat dipikirkan kembali. Ada sedikit ruang gerak untuk inovasi dalam sistem operasi komputer standar, yang harus terasa familiar secara default untuk menghindari kelebihan kognitif.

Aspek lain dari kemajuan teknologi yang patut dicatat dari tahun 1960-an adalah munculnya fiksi ilmiah, dan karena itu komputasi, dalam budaya populer. Acara TV seperti *Star Trek* (1966-1969) menggambarkan penggunaan perintah suara, telepresence, jam tangan pintar, dan komputer mini. Perangkat komputasi pribadi kecil yang terlihat sangat mirip dengan iPad saat ini serta perintah suara, panggilan video, dan, tentu saja, kecerdasan buatan yang sangat terkenal. Kartun animasi, *The Jetsons* (1962-1963), memiliki jam tangan pintar, serta mobil tanpa pengemudi dan bantuan robot. Meskipun teknologinya tidak umum atau bahkan tersedia, orang-orang sedang menyesuaikan diri dengan gagasan bahwa komputer akan berukuran kecil, ringan, serbaguna, dan memiliki kegunaan yang jauh melampaui input teks atau kalkulasi.

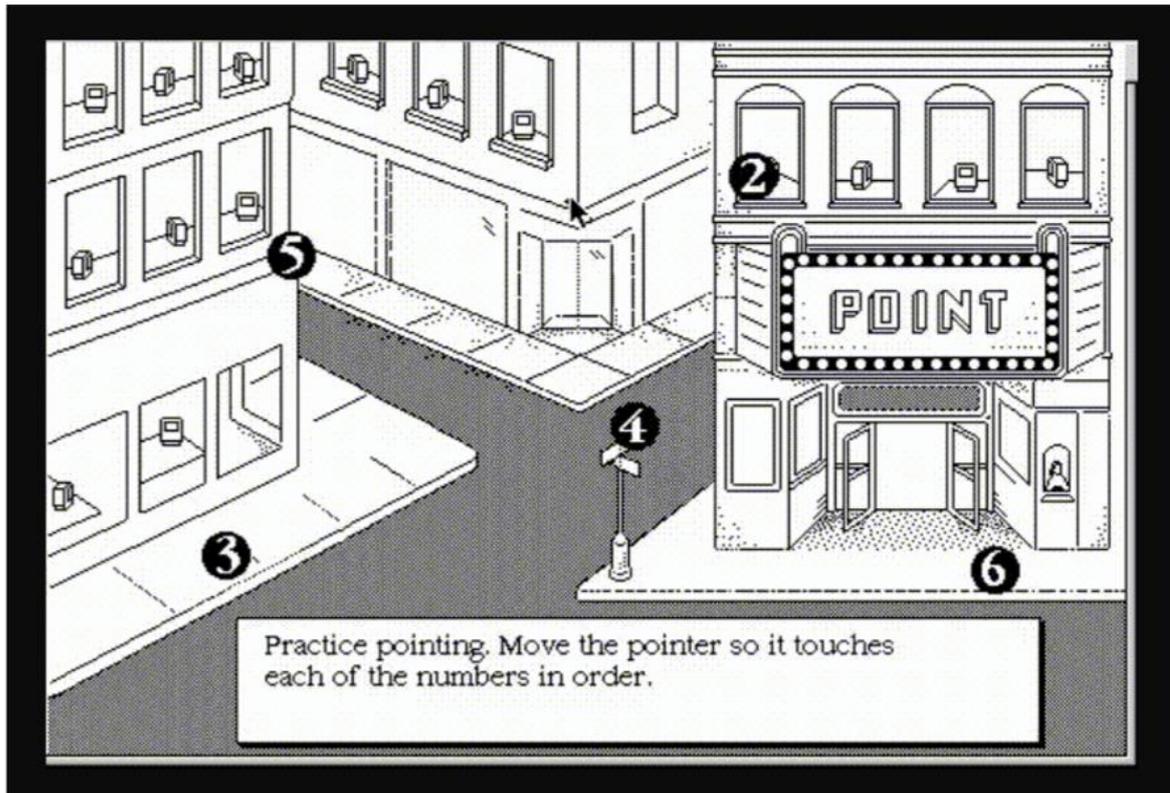
Tahun 1970-an adalah dekade sebelum komputasi pribadi. Konsol game rumahan mulai diproduksi secara komersial, dan arcade lepas landas. Komputer semakin terjangkau; tersedia di universitas terkemuka, dan lebih umum di ruang komersial. Joystick, tombol, dan sakelar dengan mudah membuat lompatan ke input video game dan memulai lintasan mereka sendiri yang terpisah sebagai controller game. Palo Alto Research Center, atau PARC, Xerox Corporation yang terkenal, mulai mengerjakan sistem kerja komputer mouse dan GUI terintegrasi yang disebut Alto. Alto dan penerusnya, Star, sangat berpengaruh untuk gelombang pertama komputer pribadi yang diproduksi oleh Apple, Microsoft, Commodore, Dell, Atari, dan lainnya pada awal hingga pertengahan 1980-an. PARC juga menciptakan prototipe KiddiComp/Dynabook 1968 Alan Kay, salah satu pendahulu tablet komputer modern.

Modalitas Sepanjang Zaman: Bangkitnya Komputasi Pribadi

Seringkali, orang menganggap mouse dan GUI sebagai tambahan yang besar dan independen untuk modalitas komputer. Tetapi bahkan di tahun 1970-an, Summagraphics membuat kombinasi tablet dan stylus kelas bawah dan atas untuk komputer, salah satunya diberi label putih untuk Apple II sebagai Tablet Grafis Apple, dirilis pada tahun 1979. Harganya relatif mahal dan didukung oleh hanya beberapa jenis perangkat lunak; melanggar dua dari lima aturan. Pada tahun 1983, HP telah merilis HP-150, komputer layar sentuh pertama. Namun, fidelitas tracking cukup rendah, melanggar aturan kesalahan pengguna.

Ketika mouse pertama kali dibundel dengan paket komputer pribadi (1984-1985), mouse didukung pada tingkat sistem operasi (OS), yang pada gilirannya dirancang untuk mengambil input mouse. Ini adalah titik balik utama untuk komputer: mouse bukan lagi input

opsional, tetapi input penting. Alih-alih perangkat antik atau opsional, komputer sekarang diminta untuk datang dengan tutorial yang mengajarkan pengguna cara menggunakan mouse, seperti yang diilustrasikan pada Gambar 1-4—mirip dengan bagaimana video game menyertakan tutorial yang mengajarkan pemain bagaimana aksi permainan dipetakan ke tombol pengontrol.



Gambar 1-4. Tangkapan layar Tur Macintosh SE, 1987

Sangat mudah untuk melihat kembali ke tahun 1980-an dan berpikir bahwa komputer pribadi adalah inovasi yang berdiri sendiri. Namun, secara umum, hanya ada sedikit inovasi dalam komputasi yang secara sendirian memajukan bidang ini dalam waktu kurang dari satu dekade. Bahkan inovasi paling terkenal, seperti FORTRAN, membutuhkan waktu bertahun-tahun untuk dipopulerkan dan dikomersialkan. Lebih sering, kekuatan pendorong di balik adopsi — dari apa yang terasa seperti inovasi baru — hanyalah hasil dari teknologi yang akhirnya memenuhi lima aturan yang disebutkan di atas: murah, andal, nyaman, memiliki perangkat lunak yang memanfaatkan teknologi, dan memiliki tingkat kesalahan pengguna yang dapat diterima.

Sangat umum untuk menemukan bahwa versi pertama dari apa yang tampak sebagai teknologi terkini sebenarnya ditemukan beberapa dekade atau bahkan berabad-abad yang lalu. Jika teknologinya cukup jelas sehingga banyak orang mencoba membangunnya tetapi tetap tidak berhasil, kemungkinan itu gagal dalam salah satu dari lima aturan. Itu hanya harus menunggu sampai teknologi meningkat atau proses manufaktur menyusul.

Kebenaran ini tentu saja dicontohkan dalam sejarah virtual reality (VR) dan augmented reality (AR). Meskipun tampilan head-mounted stereoscopic (HMD) pertama dipelopori oleh Ivan Sutherland pada 1960-an dan telah digunakan di NASA secara rutin sejak 1990-an, tidak sampai bidang elektronik seluler dan unit pemrosesan grafis (GPU) yang kuat cukup meningkat, bahwa teknologi menjadi tersedia dengan harga yang dapat diterima secara komersial, beberapa dekade kemudian. Bahkan hingga hari ini, HMD mandiri kelas atas tersedia ribuan rupiah atau tidak tersedia secara komersial. Tetapi seperti smartphone di awal 2000-an, kita dapat melihat jalur yang jelas dari perangkat keras saat ini ke masa depan komputasi spasial. Namun, sebelum kita masuk ke perangkat keras saat ini, mari kita selesaikan meletakkan jalur dari PC awal 1980-an ke jenis komputer paling umum saat ini: smartphone.

1.4 MODALITAS SEPANJANG ZAMAN: MINIATURISASI KOMPUTER

Komputer dengan perangkat keras mini muncul dari industri kalkulator dan komputer pada awal 1984 dengan Psion Organizer. Komputer tablet pertama yang sukses adalah GridPad, dirilis pada tahun 1989, yang VP penelitiannya, Jeff Hawkins, kemudian menemukan PalmPilot. Apple merilis Newton pada tahun 1993, yang memiliki sistem input karakter tulisan tangan, tetapi tidak pernah mencapai tujuan penjualan utama. Proyek tersebut berakhir pada tahun 1998 sebagai Nokia 900 Communicator—kombinasi telepon dan personal digital assistant (PDA)—dan kemudian PalmPilot mendominasi lanskap komputer mini. Diamond Multimedia juga merilis pemutar MP3 Rio PMP300 pada tahun 1998, yang ternyata menjadi hit yang mengejutkan selama musim liburan. Hal ini menyebabkan munculnya pemutar MP3 populer lainnya oleh iRiver, Creative NOMAD, Apple, dan lainnya.

Secara umum, PDA cenderung memiliki input stylus dan keyboard; lebih banyak perangkat sekali pakai seperti pemutar musik memiliki input tombol sederhana. Sejak awal pembuatannya, PalmPilots dikirimkan dengan sistem pengenalan tulisan tangan mereka, Graffiti, dan pada tahun 1999 Palm VII memiliki konektivitas jaringan. Blackberry pertama keluar pada tahun yang sama dengan input keyboard, dan pada tahun 2002 Blackberry memiliki telepon yang lebih konvensional dan perangkat kombinasi PDA.

Tetapi komputer kecil ini tidak memiliki kemewahan keyboard seukuran manusia. Ini tidak hanya mendorong kebutuhan akan pengenalan tulisan tangan yang lebih baik, tetapi juga kemajuan nyata dalam input ucapan. Dragon Dictate keluar pada tahun 1990 dan merupakan opsi konsumen pertama yang tersedia — meskipun seharga IDR 90juta, itu sangat melanggar aturan "murah". Pada tahun 1992, AT&T meluncurkan pengenalan suara untuk pusat panggilannya. Lernout & Hauspie mengakuisisi beberapa perusahaan selama tahun 1990-an dan digunakan di Windows XP. Setelah skandal akuntansi, perusahaan itu dibeli oleh SoftScan—kemudian Nuance, yang dilisensikan sebagai Siri versi pertama.

Pada tahun 2003, Microsoft meluncurkan Perintah Suara untuk PDA Windows Mobile-nya. Pada tahun 2007, Google telah mempekerjakan beberapa insinyur Nuance dan sedang dalam perjalanan dengan teknologi pengenalan suaranya sendiri. Saat ini, teknologi suara semakin tersebar di mana-mana, dengan sebagian besar platform menawarkan atau mengembangkan teknologi mereka sendiri, terutama pada perangkat seluler. Perlu dicatat

bahwa pada tahun 2018, tidak ada standar lintas platform atau bahkan lintas perusahaan untuk input suara: modalitasnya belum cukup matang.

PDA, perangkat genggam, dan telepon pintar hampir selalu dapat dipertukarkan dengan beberapa teknologi yang ada sejak awal—kalkulator, telepon, pemutar musik, pager, tampilan pesan, atau jam. Pada akhirnya, mereka semua hanyalah bagian yang berbeda dari fungsionalitas komputer. Oleh karena itu, Anda dapat menganggap peluncuran iPhone pada tahun 2007 sebagai titik balik bagi industri komputer kecil: pada tahun 2008, Apple telah menjual 10 juta lebih banyak daripada perangkat terlaris berikutnya, Nokia 2330 klasik, meskipun Nokia memegang penjualan stabil sebesar 15 juta dari 2007 hingga 2008. iPhone sendiri tidak mengambil alih penjualan iPod hingga 2010, setelah Apple mengizinkan pengguna untuk mengakses iTunes sepenuhnya. Salah satu tren yang sangat kuat dengan semua perangkat komputer kecil, apa pun mereknya, adalah perpindahan ke input sentuh. Ada beberapa alasan untuk ini.

Yang pertama adalah visual menarik dan bermanfaat, dan semakin banyak yang bisa kita lihat, semakin tinggi kualitas perangkat yang dirasakan. Dengan perangkat yang lebih kecil, ruang menjadi premium, sehingga menghapus kontrol fisik dari perangkat berarti persentase yang lebih besar dari perangkat tersedia untuk tampilan.

Alasan kedua dan ketiga adalah praktis dan fokus pada manufaktur. Selama teknologinya murah dan andal, lebih sedikit suku cadang yang bergerak berarti lebih sedikit biaya produksi dan lebih sedikit kerusakan mekanis, keduanya merupakan kemenangan besar bagi perusahaan perangkat keras.

Alasan keempat adalah bahwa menggunakan tangan Anda sebagai input dianggap alami. Meskipun tidak memungkinkan gerakan menit, GUI yang dirancang dengan baik dan disederhanakan dapat mengatasi banyak masalah yang muncul seputar kesalahan dan oklusi pengguna. Sama seperti peralihan dari keyboard ke mouse-dan-GUI, pedoman antarmuka baru untuk sentuhan memungkinkan pengalaman yang cukup konsisten dan bebas kesalahan bagi pengguna yang hampir tidak mungkin menggunakan sentuhan dengan mouse atau GUI berbasis stylus.

Alasan terakhir untuk beralih ke input sentuh hanyalah masalah selera: tren desain saat ini sedang bergeser ke arah minimalis di era ketika teknologi komputer dapat menjadi luar biasa. Dengan demikian, perangkat yang disederhanakan dapat dianggap lebih mudah digunakan, bahkan jika kurva pembelajaran jauh lebih sulit dan fitur dihilangkan.

Satu titik koneksi yang menarik antara tangan dan mouse adalah trackpad, yang dalam beberapa tahun terakhir memiliki kemampuan untuk meniru gerakan multitouch dari touchpad sambil menghindari masalah oklusi interaksi tangan-ke-tampilan. Karena tablet memungkinkan input relatif yang dapat berupa rasio ukuran layar keseluruhan, tablet memungkinkan lebih banyak gerakan menit, mirip dengan mouse atau stylus. Itu masih mempertahankan beberapa masalah yang sama yang mengganggu input tangan — kelelahan dan kurangnya dukungan fisik yang memungkinkan tangan manusia melakukan pekerjaan yang paling rumit dengan alat — tetapi dapat digunakan untuk hampir semua interaksi tingkat OS konvensional.

1.5 MENGAPA KITA BARU MELEWATI SEMUA INI?

Jadi, apa gunanya pelajaran sejarah singkat kita? Untuk mengatur tahap yang tepat ke depan, di mana kita akan berpindah dari ranah yang diketahui, komputasi hari ini, ke masa depan input spasial yang tidak diketahui. Pada titik waktu tertentu, mudah untuk berasumsi bahwa kita mengetahui segala sesuatu yang mengarah hingga saat ini atau bahwa kita selalu berada di jalur yang benar. Meninjau di mana kita berada dan bagaimana masa kini terjadi adalah cara terbaik untuk membuat keputusan yang lebih baik untuk masa depan.

Mari beralih ke eksplorasi interaksi manusia-komputer (HCI) untuk komputasi spasial. Kita bisa mulai dengan dasar-dasar yang tidak akan berubah dalam jangka pendek: bagaimana manusia dapat menerima, memproses, dan mengeluarkan informasi.

Jenis Modalitas HCI Umum

Ada tiga cara utama kita berinteraksi dengan komputer:

Visual

Pose, grafik, teks, UI, layar, animasi

Auditory

Musik, nada, efek suara, suara

Physical

Perangkat keras, tombol, haptik, objek nyata Perhatikan bahwa di latar belakang yang telah kita bahas sejauh ini, input fisik dan output audio/visual mendominasi HCI, apa pun jenis komputernya. Haruskah ini berubah untuk komputasi spasial, di dunia di mana objek digital Anda mengelilingi Anda dan berinteraksi dengan dunia nyata? Mungkin. Mari kita mulai dengan menyelami pro dan kontra dari setiap modalitas.

1.6 MODALITAS VISUAL

Kelebihan:

- 250 hingga 300 kata per menit (WPM) dipahami oleh manusia
- Sangat dapat disesuaikan
- Langsung dikenali dan dimengerti di sisi manusia
- Fidelitas yang sangat tinggi dibandingkan dengan suara atau haptics
- Waktu-independen; hanya bisa bertahan di luar angkasa selamanya
- Mudah untuk mengatur ulang atau memetakan ulang tanpa kehilangan pemahaman pengguna
- Modalitas ambien yang baik; seperti iklan atau tanda, dapat diperhatikan oleh manusia di waktu luang mereka

Kontra:

- Mudah untuk dilewatkan; tergantung lokasi
- Sebagai input, biasanya membutuhkan pendamping fisik yang kuat; Gestur dan pose sangat melelahkan
- Membutuhkan korteks prefrontal untuk memproses dan bereaksi terhadap informasi yang rumit, yang membutuhkan lebih banyak beban kognitif

- Oklusi dan tumpang tindih adalah nama permainannya
- Kemungkinan besar akan "menginterupsi" jika pengguna sedang dalam arus
- Tracking visual (mata) yang sangat presisi membutuhkan prosesor yang intensif

Penggunaan terbaik dalam interaksi khusus HMD:

- Baik untuk tampilan kamera terbatas atau situasi lain di mana pengguna terpaksa melihat ke suatu tempat
- Bagus untuk instruksi yang jelas dan jelas
- Bagus untuk menjelaskan banyak hal dengan cepat
- Bagus untuk tutorial dan orientasi

Contoh kasus penggunaan—smartphone:

- Dirancang hanya untuk visual
- Bekerja bahkan jika suaranya mati
- Bekerja dengan umpan balik fisik
- Keterjangkauan fisik minimal
- Banyak bahasa animasi baru untuk menunjukkan umpan balik

1.7 MODALITAS FISIK

Kelebihan:

- Braille: 125 WPM
- Bisa sangat cepat dan tepat
- Melewati proses berpikir tingkat tinggi, sehingga mudah untuk beralih ke "aliran" fisiologis dan mental
- Pelatihan dimasukkan ke dalam korteks motorik primer; akhirnya tidak memerlukan korteks premotor yang lebih intensif atau pemrosesan ganglia basal
- Memiliki komponen "ini nyata" otak hewan yang kuat; isyarat realitas yang kuat
- Umpan balik yang ringan diakui secara tidak sadar
- Paling sedikit penundaan antara keterjangkauan dan input
- Jenis input modalitas tunggal terbaik, seperti yang paling tepat

Kontra:

- Bisa melelahkan
- Perangkat keras fisik lebih sulit dibuat, bisa mahal, dan rusak
- Beban kognitif yang jauh lebih tinggi selama fase pengajaran
- Kurang fleksibel dibandingkan visual: tombol tidak dapat dipindahkan
- Mode membutuhkan lebih banyak hafalan untuk aliran nyata
- Variasi yang luas karena sensitivitas manusia

Penggunaan terbaik dalam interaksi khusus HMD:

- Status aliran
- Situasi di mana pengguna tidak boleh atau tidak dapat melihat UI sepanjang waktu
- Situasi di mana pengguna tidak boleh melihat tangan mereka sepanjang waktu

- Di mana penguasaan ideal atau esensial

Contoh kasus penggunaan—alat musik:

- Kecukupan fisik yang komprehensif
- Tidak ada visual yang dibutuhkan setelah tingkat penguasaan tertentu; pencipta sedang mengalir
- Hampir selalu memiliki komponen umpan balik audio
- Memungkinkan gerakan melewati bagian otak—pikiran menjadi tindakan

1.8 MODALITAS AUDIO

Kelebihan:

- 150 hingga 160 WPM dipahami oleh manusia
- Segala arah
- Mudah diegetik untuk memberikan umpan balik dan meningkatkan nuansa dunia
- Bisa sangat halus dan masih berfungsi dengan baik
- Seperti input fisik, dapat digunakan untuk memicu reaksi yang tidak memerlukan pemrosesan otak tingkat tinggi, baik pengkondisian evaluatif maupun refleks batang otak yang lebih mendasar.
- Bahkan suara yang sangat pendek dapat dikenali setelah diajarkan
- Bagus untuk harga dan umpan balik konfirmasi
- Kontra:

Mudah bagi pengguna untuk memilih keluar dengan perangkat saat ini

- Tidak ada kemampuan untuk mengontrol kesetiaan output
- Berbasis waktu: jika pengguna melewatkannya, harus mengulang
- Dapat secara fisik mengecewakan (refleks batang otak)
- Lebih lambat di seluruh papan
- Masukan yang tidak jelas dan tidak tepat karena keterbatasan bahasa Tergantung pada waktu dan implementasi
- Tidak dapat disesuaikan
- Berpotensi intensif prosesor

Penggunaan terbaik dalam interaksi khusus HMD:

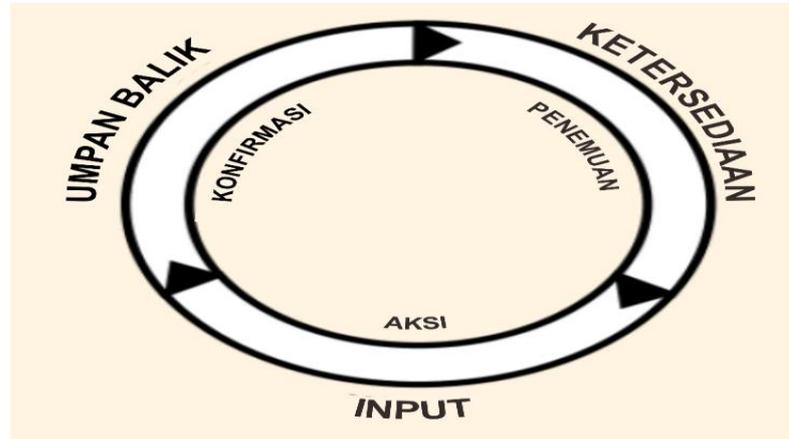
- Baik untuk reaksi visceral
- Cara yang bagus untuk membuat pengguna melihat hal tertentu
- Bagus untuk kamera yang dikendalikan pengguna
- Hebat ketika pengguna dibatasi secara visual dan fisik
- Bagus untuk beralih mode

Contoh kasus penggunaan—ruang operasi:

- Ahli bedah tawanan secara visual dan fisik; audio seringkali menjadi satu-satunya pilihan
- Pembaruan suara berkelanjutan untuk semua informasi
- Perintah suara untuk alat, permintaan, dan konfirmasi

• Suara dapat memberikan informasi yang paling padat tentang keadaan saat ini dan keadaan mental; sangat berguna dalam situasi berisiko tinggi

Sekarang setelah kami menuliskan pro dan kontra dari setiap jenis modalitas, kami dapat mempelajari proses HCI dan memetakan siklus dengan benar. Gambar 1-5 mengilustrasikan alur yang khas, diikuti dengan deskripsi tentang cara memetakan ke skenario permainan.

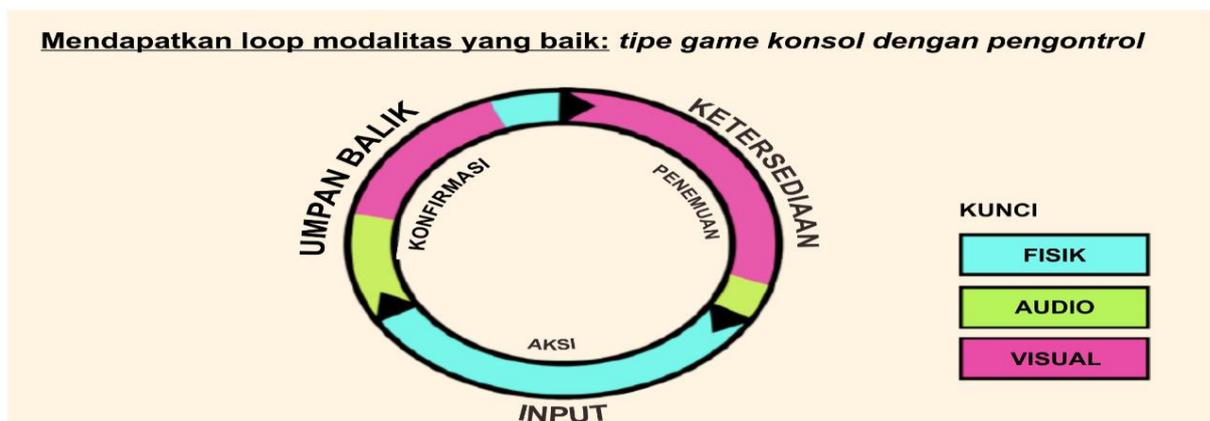


Gambar 1-5. Siklus loop modalitas HCI yang khas

Siklus ini terdiri dari tiga bagian sederhana yang berulang berulang kali di hampir semua HCI:

- Yang pertama umumnya adalah fase keterjangkauan atau penemuan, di mana pengguna mengetahui apa yang dapat mereka lakukan.
- Yang kedua adalah fase input atau tindakan, di mana pengguna melakukan sesuatu.
- Fase ketiga adalah fase umpan balik atau konfirmasi, di mana komputer mengkonfirmasi input dengan bereaksi dalam beberapa cara.

Gambar 1-6 menyajikan grafik yang sama, sekarang diisi untuk loop UX tutorial video game konsol konvensional.



Gambar 1-6. Siklus loop modalitas HCI yang khas.

Mari kita berjalan melalui ini. Dalam banyak tutorial video game, kemampuan pertama yang dapat digunakan pengguna untuk melakukan sesuatu umumnya adalah hamparan UI yang tidak dapat dilewatkan yang memberi tahu pengguna label tombol yang harus mereka tekan. Ini terkadang bermanifestasi dengan gambar atau model tombol yang sesuai. Mungkin

Kekuatan Augmented dan Virtual Reality dalam Bisnis (Dr. Agus Wibowo)

ada suara terkait seperti perubahan musik, nada, atau dialog, tetapi selama tutorial itu sebagian besar mendukung dan tidak mengajar.

Untuk video game konsol konvensional, tahap input akan sepenuhnya berupa fisik; misalnya menekan tombol. Ada video game eksplorasi yang mungkin memanfaatkan input audio seperti ucapan, atau kombinasi input fisik dan visual (mis., pose tangan), tetapi itu jarang terjadi. Dalam hampir semua kasus, pengguna cukup menekan tombol untuk melanjutkan.

Tahap umpan balik seringkali merupakan kombinasi dari ketiga modalitas: controller mungkin memiliki umpan balik haptic, visual hampir pasti akan berubah, dan akan ada suara konfirmasi.

Perlu dicatat bahwa loop khusus ini secara khusus menggambarkan fase tutorial. Saat pengguna membiasakan diri dan meningkatkan gameplay mereka, visual akan berkurang demi modalitas yang lebih mendalam. Seringkali, di kemudian hari dalam permainan, keterjangkauan suara mungkin menjadi keterjangkauan utama untuk menghindari kelebihan visual—ingat bahwa, mirip dengan modalitas fisik, audio juga dapat bekerja untuk menyebabkan reaksi yang melewati fungsi otak tingkat yang lebih tinggi. Visual adalah modalitas yang paling padat informasi, tetapi seringkali paling mengganggu dalam ruang terbatas; mereka juga membutuhkan waktu paling banyak untuk memahami dan kemudian bereaksi.

1.9 MODALITAS BARU

Dengan munculnya perangkat keras yang lebih baik dan sensor baru, kami memiliki cara baru untuk berbicara dengan komputer dan membuat mereka memantau dan bereaksi terhadap kami. Berikut daftar cepat input yang ada dalam tahap prototipe atau komersialisasi:

- Lokasi
- Tingkat pernapasan
- Nada, nada, dan frekuensi suara
- Gerakan mata
- Pelebaran pupil
- Detak jantung
- Melacak gerakan anggota tubuh yang tidak disadari

Salah satu properti aneh dari input baru ini—berlawanan dengan tiga modalitas umum yang telah kita diskusikan—adalah bahwa sebagian besar, semakin sedikit pengguna memikirkannya, semakin berguna mereka. Hampir setiap modalitas baru ini sulit atau tidak mungkin dikendalikan untuk jangka waktu yang lama, terutama sebagai mekanik input yang sadar. Demikian juga, jika tujuannya adalah untuk mengumpulkan data untuk pelatihan pembelajaran mesin, setiap upaya sadar untuk mengubah data kemungkinan akan mengotori seluruh rangkaian. Oleh karena itu, mereka paling cocok untuk digambarkan sebagai input pasif.

Salah satu properti lain dari input spesifik ini adalah bahwa input tersebut satu arah; komputer dapat bereaksi terhadap perubahan masing-masing, tetapi tidak dapat merespons dengan cara yang sama, setidaknya tidak sampai komputer berubah secara signifikan. Meski begitu, sebagian besar daftar akan mengarah ke putaran umpan balik sekitar, bukan umpan balik langsung atau instan.

Keadaan Saat Ini untuk Perangkat Komputasi Spasial

Perangkat AR dan VR memiliki metode modalitas berikut di sebagian besar penawaran perangkat keras:

Fisik

- Untuk input pengguna: pengontrol
- Untuk output komputer: haptics

Audio

- Untuk input pengguna: pengenalan suara (jarang)
- Untuk output komputer: suara dan spasial audio

Visual

- Untuk input pengguna: tracking tangan, pengenalan pose tangan, dan tracking mata
- Untuk output komputer: HMD

Satu keanehan muncul dari daftar ini: komputasi imersif, untuk pertama kalinya, menyebabkan munculnya input visual melalui visi komputer yang melacak bagian tubuh seperti tangan dan mata. Meskipun posisi dan gerakan tangan sering kali secara kebetulan penting, sejauh ini memetakan untuk menekan tombol fisik, itu belum pernah dianggap penting. Kami berbicara lebih banyak tentang ini nanti, tetapi mari kita mulai dengan jenis input paling konvensional: controller dan layar sentuh.

Controller Saat Ini untuk Sistem Komputasi Immersive

Jenis controller yang paling umum untuk headset campuran, augmented, dan virtual reality (XR), berakar pada controller permainan konvensional. Sangat mudah untuk melacak controller paket XR HMD komersial yang diberikan kembali ke desain joystick dan D-pad. Pekerjaan awal seputar sarung tangan pelacak gerak, seperti VIEWlab NASA Ames dari tahun 1989, belum dijadikan komoditas dalam skala besar. Menariknya, Ivan Sutherland telah mengemukakan bahwa controller VR harus menjadi joystick pada tahun 1964; hampir semua memilikinya, atau yang setara dengan thumbpad, pada tahun 2018.

Sebelum headset konsumen pertama, Sixsense adalah penggerak awal di ruang angkasa dengan controller terlacak magnetiknya, yang menyertakan tombol pada kedua controller yang akrab dengan konsol game apa pun: A dan B, rumah, serta tombol yang lebih umum, joystick, bumper, dan pemicu.

Sistem terikat PC yang terlacak sepenuhnya saat ini memiliki input serupa. Controller Oculus Rift, controller Vive, dan controller Windows MR semuanya memiliki kesamaan berikut:

- Tombol pilih utama (hampir selalu menjadi pemicu)
- Varian pilihan sekunder (pemicu, pegangan, atau bumper)
- Setara tombol A/B

- Masukan melingkar (thumbpad, joystick, atau keduanya)
- Beberapa tombol tingkat sistem, untuk operasi dasar yang konsisten di semua aplikasi

Umumnya, dua item terakhir ini digunakan untuk memanggil menu dan pengaturan, meninggalkan aplikasi aktif untuk kembali ke layar beranda. Headset mandiri memiliki beberapa subset dari daftar sebelumnya di pengontrolnya. Dari remote HoloLens yang tidak terlacak hingga controller tiga derajat kebebasan (3DOF) Google Daydream, Anda akan selalu menemukan tombol tingkat sistem yang dapat melakukan konfirmasi dan kemudian kembali ke layar beranda. Segala sesuatu yang lain tergantung pada kemampuan sistem tracking HMD dan bagaimana OS telah dirancang.

Meskipun secara teknis raycasting adalah input yang dilacak secara visual, kebanyakan orang akan menganggapnya sebagai input fisik, jadi perlu disebutkan di sini. Misalnya, controller Magic Leap memungkinkan pemilihan baik dengan raycast dari controller enam derajat kebebasan (6DOF) dan dari menggunakan thumbpad, seperti halnya Rift dalam aplikasi tertentu, seperti pembuat avatarnya. Namun, pada 2019, tidak ada standarisasi seputar pemilihan raycast versus analog stick atau thumbpad.

Seiring dengan peningkatan dan standarisasi sistem tracking, kami dapat mengharapkan standar ini untuk memperkuat dari waktu ke waktu. Keduanya berguna pada waktu yang berbeda, dan seperti masalah inversi sumbu Y klasik, mungkin pengguna yang berbeda memiliki preferensi yang sangat berbeda sehingga kami harus selalu mengizinkan keduanya. Terkadang, Anda ingin menunjuk sesuatu untuk memilihnya; terkadang Anda ingin menggulir untuk memilihnya. Mengapa tidak keduanya?



Gambar 1-7. Sistem input Batang Sixsense

Tracking Tubuh

Teknologi Mari kita bahas tiga jenis tracking tubuh yang paling umum dibahas hari ini: tracking tangan, pengenalan pose tangan, dan tracking mata.

Tracking tangan

Tracking tangan adalah ketika seluruh gerakan tangan dipetakan ke kerangka digital, dan kesimpulan input dibuat berdasarkan gerakan atau pose tangan. Ini memungkinkan gerakan alami seperti mengambil dan menjatuhkan objek digital dan pengenalan gerakan. Tracking tangan dapat sepenuhnya berbasis penglihatan komputer, termasuk sensor yang dipasang pada sarung tangan, atau menggunakan jenis sistem tracking lainnya.

Pengenalan pose tangan

Konsep ini sering dikacaukan dengan tracking tangan, tetapi pengenalan pose tangan adalah bidang penelitiannya sendiri. Komputer telah dilatih untuk mengenali pose tangan tertentu, seperti bahasa isyarat. Maksud dipetakan saat setiap pose tangan terkait dengan peristiwa tertentu seperti ambil, lepaskan, pilih, dan tindakan umum lainnya.

Di sisi positifnya, pengenalan pose bisa kurang intensif prosesor dan membutuhkan lebih sedikit kalibrasi individu daripada tracking tangan yang kuat. Tetapi secara eksternal, ini bisa melelahkan dan membingungkan bagi pengguna yang mungkin tidak mengerti bahwa pembuatan ulang pose lebih penting daripada gerakan tangan alami. Ini juga membutuhkan sejumlah besar tutorial pengguna untuk mengajarkan pose tangan.

Tracking mata

Mata terus bergerak, tetapi melacak posisinya membuatnya lebih mudah untuk menyimpulkan minat dan niat—kadang-kadang bahkan lebih cepat daripada yang disadari pengguna, mengingat gerakan mata diperbarui sebelum visualisasi otak menyegarkan. Meskipun cepat melelahkan sebagai input tersendiri, tracking mata adalah input yang sangat baik untuk dipadukan dengan jenis tracking lainnya. Misalnya, ini dapat digunakan untuk melakukan triangulasi posisi objek yang diminati pengguna, dalam kombinasi dengan tracking tangan atau pengontrol, bahkan sebelum pengguna sepenuhnya menyatakan minatnya.

Saya belum memasukkan tracking tubuh atau pengenalan suara dalam daftar, terutama karena tidak ada teknologi di pasaran saat ini yang bahkan mulai menerapkan kedua teknologi tersebut sebagai teknik input standar. Tetapi perusahaan seperti Leap Motion, Magic Leap, dan Microsoft membuka jalan untuk semua jenis tracking baru yang tercantum di sini.

Catatan tentang Tracking Tangan dan Pengenalan Pose Tangan

Tracking tangan dan pengenalan pose tangan keduanya harus menghasilkan perubahan yang menarik, dan agak berlawanan dengan intuisi, bagaimana manusia sering berpikir untuk berinteraksi dengan komputer. Di luar gerakan percakapan, di mana gerakan tangan sebagian besar memainkan peran pendukung, manusia umumnya tidak menganggap penting lokasi dan pose tangan mereka. Kami menggunakan tangan setiap hari sebagai alat dan dapat mengenali gerakan yang ditiru untuk tindakan yang terkait, seperti mengambil objek. Namun dalam sejarah HCI, lokasi tangan sangat berarti. Faktanya, periferal seperti mouse dan controller permainan secara khusus dirancang untuk menjadi agnostik lokasi tangan: Anda dapat menggunakan mouse di sisi kiri atau kanan, Anda dapat memegang controller satu kaki ke atas atau ke bawah di depan Anda; tidak ada bedanya dengan apa yang Anda masukkan.

Pengecualian mencolok untuk aturan ini adalah perangkat sentuh, yang lokasi tangan dan inputnya harus terhubung erat. Meski begitu, "gestur" sentuhan tidak ada hubungannya dengan gerakan tangan di luar ujung jari yang menyentuh perangkat; Anda dapat melakukan sapuan tiga jari dengan tiga jari yang Anda pilih. Satu-satunya hal yang sangat penting adalah Anda memenuhi persyaratan minimum untuk melakukan apa yang dicari komputer untuk

mendapatkan hasil yang Anda inginkan. Visi komputer yang dapat melacak tangan, mata, dan tubuh berpotensi sangat kuat, tetapi dapat disalahgunakan.

1.10 INPUT SUARA, TANGAN, DAN PERANGKAT KERAS UNTUK GENERASI BERIKUTNYA

Jika Anda bertanya kepada kebanyakan orang di jalan, asumsi umum adalah bahwa kita idealnya, dan akhirnya, berinteraksi dengan komputer kita seperti kita berinteraksi dengan manusia lain: berbicara secara normal dan menggunakan tangan untuk memberi isyarat dan berinteraksi. Banyak, banyak tim yang didanai dengan baik di berbagai perusahaan sedang mengerjakan masalah ini hari ini, dan kedua jenis input itu pasti akan disempurnakan dalam beberapa dekade mendatang. Namun, keduanya memiliki kelemahan signifikan yang tidak sering dipertimbangkan orang ketika mereka membayangkan skenario terbaik dari tracking tangan dan NLP yang instan dan lengkap.

1.11 SUARA

Dalam bahasa sehari-hari, perintah suara tidak tepat, tidak peduli seberapa sempurna dipahami. Orang sering salah paham bahkan kalimat bahasa sederhana, dan sering kali orang lain menggunakan kombinasi inferensi, metafora, dan sinonim untuk menyampaikan maksud mereka yang sebenarnya. Dengan kata lain, mereka menggunakan beberapa modalitas dan modalitas dalam modalitas untuk memastikan mereka dipahami. Jargon adalah evolusi linguistik yang menarik dari ini: kata-kata yang sangat khusus yang berarti hal tertentu dalam konteks tertentu untuk suatu kelompok adalah bentuk hotkey bahasa, jika Anda mau.

Komputer dapat bereaksi jauh lebih cepat daripada yang dapat dilakukan manusia— itulah keuntungan terbesar mereka. Mengurangi input menjadi sekadar vokalisasi manusia berarti kita secara signifikan memperlambat cara kita berkomunikasi dengan komputer mulai hari ini. Mengetik, mengetuk, dan menekan tombol yang dipetakan tindakan semuanya sangat cepat dan tepat. Misalnya, jauh lebih cepat untuk memilih sepotong teks, tekan tombol pintas untuk "potong", pindahkan kursor, dan kemudian tekan tombol pintas untuk "tempel" daripada menggambarkan tindakan tersebut ke komputer. Ini berlaku untuk hampir semua tindakan.

Namun, untuk menggambarkan sebuah skenario, menceritakan sebuah kisah, atau membuat rencana dengan manusia lain, seringkali lebih cepat menggunakan kata-kata dalam percakapan karena setiap kesalahpahaman yang potensial dapat segera dipertanyakan dan dikoreksi oleh pendengar. Ini membutuhkan tingkat pengetahuan kerja dunia yang kemungkinan besar tidak akan dimiliki komputer sampai munculnya kecerdasan buatan yang sebenarnya.

Ada keuntungan lain untuk input suara: saat Anda membutuhkan input handsfree, saat Anda sedang sibuk, saat Anda membutuhkan dikte transliterasi, atau saat Anda menginginkan sakelar modalitas cepat (misalnya, “minimalkan! keluar!”) tanpa gerakan lain. Input suara akan selalu berfungsi paling baik saat digunakan bersama dengan modalitas lain, tetapi itu bukan alasan untuk tidak menyempurnakannya. Dan, tentu saja, teknologi pengenalan suara dan transkripsi ucapan-toteks memiliki kegunaan lebih dari sekadar input.

Tangan Modalitas visual seperti tracking tangan, gerakan, dan pengenalan pose tangan secara konsisten berguna sebagai konfirmasi sekunder, sama persis dengan cara mereka berguna seperti pose tangan dan postur dalam percakapan manusia biasa. Mereka akan sangat berguna untuk komputasi spasial ketika kami memiliki cara mudah untuk melatih kumpulan data yang dipersonalisasi untuk pengguna individu dengan sangat cepat. Ini akan membutuhkan beberapa hal:

- Individu untuk mempertahankan kumpulan data biometrik yang dipersonalisasi di seluruh platform
- Sebuah cara bagi individu untuk mengajarkan komputer apa yang mereka ingin komputer tersebut perhatikan atau abaikan

Alasan untuk persyaratan ini sederhana: manusia sangat bervariasi baik dalam seberapa banyak mereka bergerak dan memberi isyarat dan apa arti gerakan itu bagi mereka. Satu orang mungkin menggerakkan tangan mereka terus-menerus, tanpa melibatkan pikiran. Orang lain mungkin hanya memberi isyarat sesekali, tetapi isyarat itu sangat penting. Kami tidak hanya perlu menyesuaikan jenis gerakan ini secara luas per pengguna, tetapi juga memungkinkan pengguna itu sendiri untuk menginstruksikan komputer apa yang harus diperhatikan secara khusus dan apa yang harus diabaikan.



Gambar 1-8. Triangulasi untuk menopang berat tangan itu penting—bahkan jika Anda memiliki ujung tajam atau pisau digital, Anda harus memiliki cara untuk menopang tangan Anda untuk gerakan yang lebih kecil

Alternatif untuk sistem yang dipersonalisasi dan terlatih sebagian besar adalah apa yang kita miliki saat ini: serangkaian pose tangan yang telah ditentukan sebelumnya yang dipetakan secara khusus untuk tindakan tertentu. Untuk Leap Motion, pose “grab” menunjukkan bahwa pengguna ingin memilih dan memindahkan objek. Untuk Hololens, gerakan “cubit” menunjukkan pemilihan dan gerakan. Magic Leap mendukung 10 pose tangan, beberapa di antaranya memetakan tindakan yang berbeda dalam pengalaman yang berbeda. Hal yang sama berlaku untuk *controller* Oculus Rift, yang mendukung dua pose tangan (titik, dan jempol ke atas), yang keduanya dapat dipetakan ulang ke tindakan pilihan pengembang.

Ini mengharuskan pengguna untuk menghafal pose dan gerakan yang diperlukan oleh perangkat keras alih-alih gerakan tangan alami, seperti halnya perangkat tablet menstandarisasi *swipe-to-move* dan *pinch-to-zoom*. Meskipun jenis bahasa isyarat manusia-komputer ini memiliki potensi untuk menstandarisasi dan menjadi norma, para pendukung harus menyadari bahwa apa yang mereka usulkan adalah alternatif dari bagaimana manusia menggunakan tangan mereka saat ini, bukan pemetaan ulang. Hal ini terutama diperburuk oleh fakta bahwa tangan manusia tidak akurat; mereka membutuhkan dukungan fisik dan alat untuk memungkinkan presisi nyata, seperti yang ditunjukkan pada Gambar 1-8.

1.12 CONTROLLER DAN PERIFERAL FISIK LAINNYA

Seperti yang kita lihat di pendahuluan, ada banyak waktu dan upaya yang dilakukan untuk menciptakan berbagai jenis input fisik untuk komputer selama hampir satu abad. Namun, karena lima aturan, periferal telah distandarisasi. Dari lima aturan, dua yang paling penting di sini: lebih murah untuk diproduksi dalam skala besar, dan input telah distandarisasi bersama perangkat keras yang mendukungnya.

Namun, kita sekarang memasuki waktu yang menarik untuk elektronik. Untuk pertama kalinya, hampir semua orang dapat membeli atau membuat periferal sendiri yang dapat bekerja dengan banyak jenis aplikasi. Orang-orang membuat segalanya dari bagian pihak ketiga: dari keyboard dan mouse, hingga pelacak Vive Frankenstein-ed di atas kelelawar bisbol atau hewan peliharaan, dan pekerjaan cat khusus untuk controller Xbox mereka.

Merupakan pertanyaan besar untuk berasumsi bahwa karena komputasi spasial akan memungkinkan lebih banyak penyesuaian pengguna, konsumen secara alami akan mulai membuat input mereka sendiri. Tetapi mudah untuk berasumsi bahwa produsen akan membuat perangkat keras yang lebih disesuaikan untuk memenuhi permintaan. Pertimbangkan mobil hari ini: Lexus 4 memiliki lebih dari 450 pilihan roda kemudi saja; ketika Anda memasukkan semua opsi, ini menghasilkan empat juta kombinasi kendaraan yang sama. Saat komputasi bersifat pribadi dan berada di rumah Anda bersama Anda, orang akan memiliki opini yang kuat tentang tampilan, rasa, dan reaksinya, seperti halnya dengan kendaraan, furnitur, dan wallpaper mereka.

Pembicaraan tentang penyesuaian yang intens ini, baik di sisi platform maupun di sisi pengguna, membawa kita ke jalur pemikiran baru: komputasi spasial memungkinkan komputer untuk dipersonalisasi dan bervariasi seperti rumah rata-rata orang dan bagaimana mereka mengatur barang-barang di rumah mereka. Oleh karena itu, input harus sama-sama bervariasi. Cara yang sama seseorang mungkin memilih satu pena versus pena lain untuk menulis akan berlaku untuk semua aspek interaksi komputer.

BAB 2

MERANCANG UNTUK INDRA KITA BUKAN UNTUK PERANGKAT DATA

Bayangkan masa depan di mana hubungan kita dengan teknologi sekaya kenyataan. Kami tidak sering mengoceh tentang berapa banyak waktu yang dihabiskan di depan layar, dan untungnya, sebagian besar perusahaan teknologi merasakan hal yang sama. Mereka telah banyak berinvestasi dalam sensor dan AI untuk membuat mesin penginderaan. Dengan memanfaatkan ucapan, spasial, dan data biometrik, yang dimasukkan ke dalam kecerdasan buatan, mereka mengembangkan teknologi ke dalam bentuk yang lebih manusiawi. Tetapi tidak banyak yang dipahami tentang bagaimana merancang teknologi yang digerakkan oleh mesin penginderaan yang menarik, dapat diakses, dan bertanggung jawab. Karena itu, industri teknologi perlu berinvestasi lebih banyak dalam memahami desain yang responsif secara manusiawi bersama dengan praktik rekayasa, kebijakan, dan alat yang dibutuhkan.

Kita semua menginginkan solusi yang lebih baik untuk masa depan yang lebih bahagia, tetapi bagaimana kita melakukannya dengan benar dalam evolusi teknologi saat ini? Dalam bab ini, kita akan mengeksplorasi topik ini dan semoga menginspirasi eksplorasi lebih lanjut.

Saya bekerja dengan beberapa tim di seluruh perusahaan untuk membawa teknologi baru ke dalam produk dan layanan untuk memecahkan tantangan nyata manusia dan masyarakat. Lebih dari 25 tahun mendorong desain ke depan melalui tiga perubahan teknologi utama, saya telah melihat internet memperkuat ekonomi pengetahuan kita, dan komputasi seluler mengubah cara kita berkomunikasi. Di masa depan, komputasi spasial, yang didukung oleh AI, akan secara dramatis memperluas sarana kami untuk berkolaborasi satu sama lain dan menggunakan informasi. Ini akan sangat mengubah cara kita bekerja, hidup, belajar, dan bermain. Saya percaya pengaruhnya terhadap masyarakat akan lebih besar daripada gabungan internet dan komputasi seluler. Sebagai seorang desainer, saya sangat bersemangat dan terkadang sedikit takut untuk mengambil bagian dalam periode sejarah manusia yang luar biasa ini.

2.1 MEMBAYANGKAN MASA DEPAN

Tea Uglow, direktur kreatif di Google, adalah pemimpin desain pertama yang perspektifnya tentang komputasi spasial sangat memengaruhi saya dan tim saya. Dia membantu kami membayangkan masa depan yang lebih baik yang dapat kami bangun. Saya ingin membawa Anda dalam perjalanan imajiner yang dibagikan Tea dalam Ted Talk:

Pejamkan matamu sebentar saja. Bayangkan tempat bahagia Anda, kita semua memilikinya. Bahkan jika itu adalah fantasi. Bagi saya, tempat ini berada di pantai di Australia dengan teman-teman saya di sekitar saya, matahari bersinar, nuansa air asin di jari kaki saya dan suara desis barbekyu. Ini adalah tempat yang membuat saya merasa bahagia karena alami, sederhana dan saya terhubung dengan teman-teman. Ketika saya duduk di depan

komputer saya atau menghabiskan terlalu banyak waktu melihat layar ponsel saya, saya tidak merasa sangat bahagia. Saya tidak merasa terhubung. Tapi setelah seharian berada di tempat bahagia saya, saya mulai kehilangan informasi dan koneksi lain yang saya dapatkan melalui telepon saya. Tapi aku tidak merindukan ponselku. Ponsel saya tidak membuat saya bahagia. Jadi, sebagai seorang desainer saya tertarik dengan bagaimana kita mengakses informasi dengan cara yang terasa alami, sederhana dan bisa membuat kita bahagia.

Pola pikir ini membantu kita sebagai desainer memahami nilai dan pentingnya pekerjaan kita kapan pun kita memulai produk atau desain baru.

2.2 TEKNOLOGI SENSORIK

Sebelum kita dapat mengeksplorasi pentingnya desain dengan komputasi spasial, kita perlu mendefinisikan teknologi yang terlibat. Pengalaman spasial didorong oleh data sensor yang dimasukkan ke mesin yang digerakkan oleh pembelajaran mesin. Berikut adalah ringkasan singkat komputasi spasial dan mesin sensoriknya.

Komputasi spasial tidak terkunci dalam empat persegi panjang tetapi dapat mengalir dengan bebas di dalam dan melalui dunia di sekitar kita, tidak seperti komputasi seluler dan komputasi desktop sebelumnya. Dengan kata lain, komputasi spasial menggunakan ruang di sekitar kita sebagai kanvas untuk pengalaman digital.

Impian komputasi spasial adalah bahwa teknologi akan memudar dan interaksi digital akan menjadi manusiawi. Misalnya, perangkat input seperti mouse, keyboard, dan bahkan layar sentuh menengahi pengalaman kami. Dengan komputasi spasial, kita dapat menggunakan suara, penglihatan, sentuhan (dalam 3D), gerakan, dan input alami lainnya untuk terhubung langsung dengan informasi. Kita tidak perlu lagi berpikir dan berperilaku seperti komputer untuk memahami kita, karena itu akan berhubungan secara manusiawi dengan kita.

Dengan asumsi bahwa komputer memahami kita, maka komputasi spasial juga dapat memahami perbedaan kita dan mendukung kemampuan dan perbedaan manusiawi kita. Misalnya, kami dapat memberikan informasi verbal tentang dunia di sekitar orang yang kehilangan penglihatan atau menerjemahkan nuansa budaya, bukan hanya bahasa, saat berkomunikasi lintas budaya. Sebaliknya, komputasi spasial dapat meningkatkan kemampuan kita yang ada, seperti memberi seseorang yang ahli matematika kemampuan untuk melihat dan berinteraksi dengan lebih banyak fakta dan data yang tidak dapat dipahami orang lain.

Data sensorik dihasilkan dari mesin sensorik kami yang didukung oleh teknologi AI. Penglihatan komputer, pendengaran mesin, dan sentuhan mesin dapat menghasilkan data seperti lokasi persis kamera Anda; dimensi ruang di sekitar Anda; mengidentifikasi objek, orang, dan ucapan; biodata, dan masih banyak lagi. Menggunakan teknologi AI, kami dapat menafsirkan data ini dengan cara yang meniru persepsi manusia. Saat kita memahami dunia, mesin juga dapat memahami dunia.

Karena indra mesin semakin ditambahkan ke dalam segala hal, dan ditempatkan di mana-mana, semakin banyak kasus penggunaan yang muncul. Berikut adalah beberapa penggunaan mesin dan data sensorik saat ini:

- Kamera berkemampuan *augmented reality* (AR) akan mencapai 2,7 miliar ponsel pada akhir 2019. Dengan kekuatan teknologi AI, kamera AR dengan cepat dapat memahami apa yang mereka "lihat". Google Lens (sistem AR Google untuk Pixel Phone) kini dapat mengidentifikasi satu miliar produk, empat kali lebih banyak dibandingkan saat diluncurkan pada 2017.
- Melalui ponsel Anda yang mendukung AR, teknologi AI dapat mendeteksi emosi dasar manusia seperti marah, jijik, jijik, takut, kebahagiaan, netral, kesedihan, dan kejutan dari ekspresi wajah Anda. Emosi-emosi ini dipahami secara lintas budaya dan umum digunakan, tetapi emosi-emosi ini tidak selalu menjadi ukuran yang benar tentang bagaimana perasaan seseorang sebenarnya di dalam. Mark Billinghurst, pelopor AR, dan Direktur Empathic Computer Laboratory di University of South Australia mengatakan, "Ekspresi wajah saja bisa menjadi ukuran emosi yang buruk. Misalnya, jika seseorang mengerutkan kening, apakah itu karena mereka tidak bahagia, atau mungkin hanya berkonsentrasi pada tugas yang rumit? Untuk perkiraan emosi yang lebih baik, penting untuk mempertimbangkan isyarat kontekstual lainnya, seperti tugas yang dilakukan orang tersebut, lingkungan tempat mereka berada, apa yang mereka katakan, dan isyarat fisiologis tubuh mereka (misalnya detak jantung), dll. Orang-orang mempertimbangkan semua isyarat ini untuk memahami bagaimana perasaan mereka, atau perasaan orang lain. Mesin harus melakukan hal yang sama."
- AR mempercepat pelatihan dengan memanfaatkan rasa propriocepsi manusiawi kita, pemahaman tentang ruang di sekitar kita, untuk pelatihan dan keselamatan.
- Mikrofon dan speaker kami telah menjadi asisten virtual kami dan semakin banyak memasuki rumah, telepon, alat dengar, dan perangkat kami lainnya.
- Pakaian dan jam tangan yang disematkan dengan sensor berpotensi mengukur intensitas emosi kita dengan keringat (respon kulit galvanis) dan memantau kesehatan kita 24/7 melalui detak jantung kita.
- Kota-kota kita menjadi "pintar" dengan sejumlah besar sensor ditempatkan di jalan-jalan, mobil, dan sistem transportasi umum kita. Mengintegrasikan data mereka memungkinkan kota mendapatkan insight yang lebih rinci tentang bagaimana memecahkan masalah yang saling berhubungan. Sensor ini memantau hal-hal seperti cuaca, kualitas udara, lalu lintas, radiasi, dan ketinggian air, dan mereka dapat digunakan untuk secara otomatis menginformasikan layanan mendasar seperti lalu lintas dan lampu jalan, sistem keamanan, dan peringatan darurat.

Komputasi spasial telah muncul dari interaksi kemajuan teknologi dalam sensor mesin, daya rendering, AI dan pembelajaran mesin, penangkapan 3D, dan tampilan. User interface suara (VUI), gerakan, dan tampilan XR memberikan konteks baru untuk komputasi. Komputasi spasial terjadi di mana pun kita berada, di pergelangan tangan kita, di mata dan telinga kita,

di meja dapur dan meja ruang konferensi, dan di ruang keluarga, kantor, dan alat transportasi favorit kita. Tanyakan saja pada GPS mobil bagaimana mencapai tujuan perjalanan Anda.

Sementara VUI telah mencapai rumah, telepon, dan mobil kami, layanan AR belum mencapai adopsi konsumen massal. Beberapa orang percaya ini akan datang ketika kacamata AR kelas konsumen ada di sini. Saya percaya titik kritis akan tiba hanya ketika perangkat, tanggal sensorik, dan sistem AI bersama-sama membuka kekuatan super kreatif manusia melalui kolaborasi spasial. Saya akan menjelaskan ini lebih lanjut nanti di bab ini.

Mesin dengan *Artificial Intelligence (Kecerdasan buatan)* berpikir secara independen dan menemukan cara baru dalam melakukan sesuatu—inilah tujuannya, tetapi belum ada mesin yang cerdas dengan sendirinya. Namun pembelajaran mesin dan adiknya yang jauh lebih pintar, pembelajaran mendalam, menyediakan cara bagi mesin untuk menginterpretasikan sejumlah besar data dengan cara baru dan menakjubkan. Mesin cerdas kita hari ini dapat belajar, tetapi mereka tidak sepenuhnya mengerti.

Untuk komputasi spasial, pembelajaran mesin bertindak sedikit seperti sistem saraf manusia untuk indera kita. Karena sistem dan sistem bangunan kota kita mengintegrasikan sensor yang jumlahnya terus bertambah, sensor tersebut juga mencerminkan sistem saraf. Data dari sensor seperti kamera (penglihatan), mikrofon (pendengaran), dan unit pengukuran inersia (IMU) dikumpulkan dan ditafsirkan oleh sistem pembelajaran mesin (saraf) yang kompleks. Jika Anda tidak dapat membaca bahasa Belanda, kamera Anda dapat menerjemahkannya untuk Anda; jika Anda tidak dapat mendengar dengan baik, speaker Anda dapat memperkuat suara itu atau menerjemahkan ucapan menjadi teks; jika mobil Anda melewati lubang, kendaraan Anda dapat segera memberi tahu departemen pekerjaan umum setempat untuk memperbaiki lubang tersebut; sebuah mainan dapat mengetahui apakah itu sedang digunakan atau ditinggalkan di dalam kotak mainan, sehingga menghasilkan mainan yang lebih baik dan mengurangi tempat pembuangan sampah.

Pembelajaran mesin dan data historis mengingat dan memahami masa lalu. Kami sudah melihat kalimat kami selesai untuk kami di Gmail berdasarkan gaya penulisan historis kami. Suatu hari, anak-anak saya mungkin mengalami hidup saya ketika mereka seusia saya; mungkin kita bisa “melihat” prediksi masa depan penemuan kita berdasarkan peristiwa sejarah.

Seiring kemajuan AI, desain sensorik akan terus menjadi lebih alami, memberikan perangkat kami indera manusia yang alami. Kami membayangkan dunia di mana alat kami lebih alami, dan saya percaya ini adalah masa depan yang diinginkan orang. Semakin alami dan intuitif alat, semakin mudah diakses, di situlah desain sensorik memainkan peran penting. Jadi, Untuk Siapa Kita Membangun Masa Depan Ini?

Kami sedang membangun masa depan untuk orang-orang seperti dua anak laki-laki pada Gambar 2-1. Mereka akan membangun produk dan layanan berdasarkan ekosistem yang kita bangun hari ini. Mari dengarkan mereka dan terinspirasi oleh kebutuhan mereka untuk masa depan yang lebih baik. Berikut adalah beberapa hal yang mereka katakan.



Gambar 2-1. Dua generasi Z

Anak laki-laki “Jadi, Untuk Siapa Kita Membangun Masa Depan Ini?” di halaman 33 adalah GenZ'ers, sebuah kelompok yang akan “mencakup 32% dari populasi global sebesar 7,7 miliar pada tahun 2019.” Generasi saat ini berusia 9 hingga 24 tahun atau lahir setelah tahun 2000. Mereka memiliki lebih banyak perangkat daripada generasi sebelumnya. Di Amerika Serikat, mereka memiliki Amazon Alexa di rumah mereka, mereka selalu membawa chip AI di ponsel mereka, dan dalam 10 tahun mereka mungkin memiliki kacamata AR di hidung mereka.

Identitas mereka tidak ditarik pada ras atau jenis kelamin tetapi pada identitas bermakna yang bergeser seperti yang mereka lakukan. Mereka dengan lancar dan terus menerus mengekspresikan kepribadian mereka. Jadi, ketika ditanya, "Apakah Anda pikir Anda akan menikah dengan perempuan atau laki-laki?" dua pria muda pada Gambar 2-1 tidak menganggap itu pertanyaan yang aneh. Salah satu dari mereka berkata "seorang gadis" dan yang lainnya berkata, "Saya sedang mengerjakannya." Jawaban mereka tidak canggung atau tidak nyaman, karena mereka bukan pemikir biner.

Saya melihat merek bergeser dari menciptakan pengalaman tipe kreasi sendiri untuk YouTube atau Instagram ke merek yang memungkinkan identitas cair dengan menggunakan masker wajah AR di Snapchat dan Facebook Messenger.

Ini adalah jenis pergeseran yang diharapkan dengan komputasi spasial. Kami berpindah dari tempat di mana informasi disimpan di layar ke dunia di mana ekspresi kreatif dapat mengalir dengan bebas ke lingkungan di sekitar kami dengan AR yang didukung oleh AI. Pemikir masa depan harus mampu menavigasi melalui kekacauan sambil membangun koneksi, itulah sebabnya kreativitas adalah keterampilan inti yang dibutuhkan untuk generasi mendatang.

Kita semua perlu membuat ekspresi kreatif lebih sederhana, lebih alami, dan tidak terlalu terikat pada perangkat dan lebih pada indra manusiawi kita. Dalam banyak hal, alat spasial akan didemokratisasi. Alat seperti animasi waktu nyata adalah keterampilan inti yang

dibutuhkan dalam komputasi spasial, tetapi, saat ini, kesulitan animasi menyebabkannya diserahkan kepada para profesional dengan akses ke alat tertentu.

Ini adalah mengapa tim saya di Adobe membuat alat yang memungkinkan Anda merekam gerakan burung terbang atau teman menari hanya dengan menangkap gerakan melalui kamera ponsel Anda dan langsung mentransfernya ke objek 3D atau desain 2D. Sungguh menakjubkan melihat keajaiban di wajah orang-orang saat mereka menggunakan keajaiban teknologi penginderaan (Gambar 2-2).



Gambar 2-2. Satu generasi Z memakai Microsoft HoloLens

Anggota GenZ ingin berkreasi secara real time. Mereka juga berharap untuk berkreasi dengan apa saja, di mana saja, hanya dengan melihatnya atau berinteraksi dengannya (yang kita sebut bermain).

Saat ini, banyak anak belajar dengan menjelajahi dunia di sekitar mereka dari ruang kelas mereka menggunakan AR seluler. Atau mereka meminta Google untuk menyelesaikan pekerjaan rumah matematika mereka—ya, anak-anak saya melakukannya. Pada saat GenZ mencapai dunia kerja, mereka akan memiliki antarmuka berkemampuan AR yang memproyeksikan informasi di dalam dan di sekitar objek sehingga mereka dapat menggunakan kedua tangan untuk belajar gitar. Seperti yang dikatakan Tea Uglow, ini akan sedikit seperti "YouTube mekanis yang luar biasa."

Kreativitas sedang ditingkatkan dan diperluas ke dunia di sekitar kita, memberikan semua orang keterampilan yang hanya dimiliki oleh para profesional terlatih saat ini. Keterampilan seperti animasi, pembuatan objek 3D, dan desain ruang 3D akan dibuat mudah dan dapat diakses dengan cara yang sama seperti internet membuat penerbitan tersedia untuk semua orang. AR, *virtual reality* (VR), dan AI akan mengubah kita dari berbagi apa yang ada di pikiran kita menjadi juga berbagi apa yang ada di hati kita.

Saat AR, AI, dan komputasi spasial berkembang ke dunia di sekitar kita, ekspresi kreatif akan menjadi sama pentingnya dengan literasi. Sebagai anggota industri teknologi yang lebih luas, Adobe ingin membuat alat kreatif kami tersedia untuk semua orang (XD gratis!), termasuk kemampuan dan budaya yang berbeda, dan menghormati hak privasi dan

transparansi orang. Ini adalah waktu yang menyenangkan untuk menciptakan alat yang membentuk hubungan kita dengan realitas spasial.

2.3 PERAN MASA DEPAN DESAINER DAN TIM

Sederhananya, desain sensorik adalah perekat yang menggabungkan disiplin desain spasial (seperti arsitektur, interior, industri, sistem, dan desainer UI) dengan sains (seperti kognitif dan neurosains), seniman, aktivis dan pembuat kebijakan, dan insinyur AI. Merancang untuk masa depan dengan komputasi spasial bertenaga AI membutuhkan keragaman keterampilan dan pemahaman mendalam tentang perilaku manusia oleh semua orang yang terlibat.

Ini adalah area pertumbuhan desain dan membutuhkan keragaman peran yang besar untuk diciptakan sehingga akan menghasilkan yang terbaik dalam kemanusiaan. Pada Agustus 2018, saya bertemu dengan seniman pertunjukan tunarungu yang menginspirasi, Rosa Lee Timm. Dia meminta Adobe Design untuk:

[h]memasukkan mereka [orang-orang dengan kemampuan berbeda seperti dirinya] dalam proses desain dan menjadi anggota tim. Dan siapa tahu, beberapa dari kita mungkin memiliki beberapa penemuan dan ide serta kreativitas baru yang tidak Anda pikirkan, lalu menjadi organik. Dan kemudian ketika selesai, itu dirancang dengan mudah dengan distribusi yang mudah dari awal.

Rosa kemudian bertanya kepada kami apakah kami dapat membuat alat yang menerjemahkan kata-kata yang diucapkan ke dalam bahasa isyarat sehingga kami dapat “membaca” dalam bahasanya sendiri. Dia menunjukkan bahwa banyak video pelatihan bahkan tidak memiliki teks teks. Ini mengilhami saya untuk memikirkan bagaimana teknologi tracking dan pengenalan wajah dan tangan dapat digunakan untuk menerjemahkan bahasa isyarat ke bahasa tradisional/lokal dan bahasa Inggris kembali ke bahasa isyarat.

Orang lain yang sangat menginspirasi tim kami untuk berpikir lebih global, lintas budaya, dan inklusif adalah Farai Madzima, Pemimpin UX Shopify. Tahun lalu, dia mengunjungi kami di Adobe Design dan membagikan pemikiran ini:

Jika Anda mendapat kesan bahwa keragaman hanyalah tentang nuansa cokelat, Anda tidak memperhatikan. Jika Anda berpikir keragaman hanya tentang gender atau kemampuan, maka Anda tidak memperhatikan. Anda perlu bekerja dengan orang-orang yang tidak berjalan, berpikir, dan berbicara seperti Anda. Anda perlu membuat orang-orang itu menjadi bagian dari cara Anda bekerja. Ini terdengar seperti hal yang sulit, selain memecahkan masalah merancang produk, tetapi ini sangat penting. Tantangan yang kita lihat di masyarakat lahir dari kenyataan bahwa kita belum melihat apa yang dibutuhkan dunia dari industri kita. Kami belum memahami apa yang dibutuhkan rekan kerja kami dari kami dan apa yang kami butuhkan untuk diri kami sendiri, yang merupakan gagasan untuk lebih berpikiran terbuka tentang apa yang berbeda di dunia ini.

2.4 PERAN WANITA DALAM AI

Visi saya untuk masa depan desain dimulai dengan inklusivitas dan keragaman. Saat kami membuat bahasa desain baru ini, kami membutuhkan tim yang beragam untuk menetapkan fondasi. Ini termasuk wanita. Saya percaya bahwa selalu ada banyak cara untuk memecahkan tantangan, dan mencari perspektif yang berbeda akan sangat penting untuk keberhasilan desain sensorik.

Saya percaya bahwa kita membutuhkan wanita dan pria yang memimpin masa depan desain digital untuk komputasi spasial dan AI. Dalam 30 tahun terakhir, kami telah melihat laki-laki mendominasi desain platform komputer kami, dan sebagai akibatnya, kami sekarang melihat kurangnya insinyur perempuan di sektor teknologi. AI mempersonalisasi keuangan, hiburan, berita online, dan sistem rumah kita. Orang-orang yang merancang sistem komputasi spasial hari ini akan berdampak langsung pada dunia di sekitar kita besok. Ini akan membutuhkan berbagai pemikiran, menyatukan perspektif yang berbeda untuk memecahkan masalah nyata dengan cara yang berkelanjutan dan empatik. Ini tidak hanya baik untuk bisnis, tetapi juga untuk masyarakat secara keseluruhan.

Untungnya, dalam dua tahun terakhir, kami telah melihat dukungan industri yang substansial dan tujuan tinggi yang ditetapkan untuk mengubah cara kami mendekati AI. Ada banyak program yang dipimpin oleh perempuan. Wanita menyukai Fei-Fei Li di Stanford; Kate Crawford, direktur Institut AI Now Microsoft; Terah Lyons, memimpin Kemitraan untuk AI; dan bahkan Michelle Obama mendukung Olga Russakovsky, salah satu pendiri AI4ALL untuk mendidik wanita di AI selama sekolah menengah, hanya untuk beberapa nama. Saya pribadi bersemangat untuk apa yang ada di depan dan apa yang akan kita capai ketika kita merangkul keragaman dalam ide.

2.5 DESAIN SENSORIK

Ini adalah keragaman ide di samping pemahaman mendalam tentang menjadi manusia yang akan mendorong desain spasial yang bertahan lama. Secara historis desain kami telah dibatasi oleh medium dan dimensi. Kita dapat melihat ke dunia di sekitar kita untuk melihat desain apa yang telah melewati ujian waktu, seperti arsitektur yang sudah dikenal atau tata letak situs web. Keterbatasan media seorang desainer, baik itu fisik atau di layar, telah menentukan desain yang dihasilkan dan dari waktu ke waktu norma yang diterima.

Di dunia yang dipenuhi komputasi spasial masa depan kita, jumlah batasan mendekati nol. Tidak lagi dibatasi oleh sumber daya fisik atau layar 2D, desain sensorik membuka dunia kemungkinan yang jauh melampaui media desain apa pun yang ada saat ini. Untuk menggunakan desain sensorik, pertama-tama kita harus memahaminya, dan itulah mengapa kami mengembangkan Bahasa Desain Sensorik.

Sebuah Pengantar

Desain sensorik adalah bahasa desain yang diadaptasi, terinspirasi kemanusiaan, di seluruh industri, untuk komputasi spasial. Sama seperti bahasa desain material yang menjadi panduan default untuk desain antarmuka seluler, kami berharap bahasa desain sensorik akan menjadi panduan desain default untuk interaksi di luar layar.

Desain sensorik membalikkan paradigma desain yang ada di kepala mereka dan karenanya membutuhkan pendekatan baru. Misalnya, desain layar berfokus pada tindakan yang diinginkan pengguna untuk dilakukan, tetapi desain sensorik berfokus pada motivasi yang dimiliki pengguna dengan melibatkan kemampuan kognitif indra mereka. Dengan pemikiran ini, kami di Adobe memutuskan untuk kembali ke dasar dan fokus pada prinsip pertama universal dari perilaku manusia. Kita juga perlu memahami perbedaan dan lapisan antara masyarakat, budaya, dan individu yang terorganisir. Beruntung bagi kami, sudah ada banyak pekerjaan yang dilakukan di bidang ini. Kami hanya perlu menyaring ratusan makalah penelitian untuk menghasilkan titik awal yang penting.

Dengan pemikiran ini, saya mengumpulkan sekelompok desainer, ilmuwan kognitif, pengusaha, dan insinyur untuk membantu menciptakan bahasa desain baru untuk komputasi spasial yang dapat kita semua pahami dan gunakan. Orang pertama yang bergabung dengan tim desain sensorik kami adalah dua ilmuwan kognitif, Stefanie Hutka dan Laura Herman dan seorang pembuat kode/desainer pembelajaran mesin Lisa Jamhoury.

Kami mulai dengan pemahaman bahwa manusia memiliki memori spasial yang sangat baik. Kami menggunakan rasa proprioception kami untuk memahami dan mengkodekan ruang di sekitar kami. Saya yakin Anda bisa menutup matanya di rumah dan masih berjalan ke dan membuka lemari es. Kami telah melihat Virtual Reality menggunakan proprioception sebagai alat yang efektif untuk pelatihan spasial, tetapi desain sensorik lebih dari spasial, ini melibatkan indera kita.

Psikolog telah membuktikan bahwa tersenyum membuat Anda merasa lebih bahagia bahkan pada tingkat kimiawi. Hubungan antara otak, tubuh, dan indera ini adalah cara kita memahami dan memahami dunia kita. Dengan merancang indera dan kemampuan kognitif manusia, kita dapat meretas persepsi kita tentang realitas. Anda bahkan bisa mengatakan Desain Sensorik adalah desain realitas yang dirasakan.

Mendekati desain sensorik

Ini adalah kesempatan yang luar biasa untuk dapat merancang untuk persepsi manusia, tetapi ini adalah salah satu yang datang dengan tanggung jawab yang besar. Pikiran untuk mengubah persepsi seseorang tentang realitas melalui desain, dan konsekuensi potensial, menakutkan. Jadi tim desain sensorik, menulis pendekatan desain sensorik yang membuat kami bertanggung jawab:

- **Berpusat pada manusia** dengan membangun bahasa di sekitar interaksi manusia yang intuitif. Kita dapat melakukan ini hanya dengan memahami perilaku dasar manusia, tubuh kita, dan kemampuan kognitif kita.
- **Jadilah kolaboratif** dengan berbagi insight kami, mendengarkan umpan balik, dan belajar dari berbagai macam orang, dari pakar industri hingga pengguna akhir kami.
- **Jadilah pemimpin desain melalui pekerjaan** kami, berbagi insight kami secara terbuka dan kolektif.
- **Tentukan prinsip, metodologi, dan pola** yang dapat kita gunakan untuk bekerja sama secara lebih efektif dan meningkatkan produk yang kita buat. • Hormati orang dengan

menghormati privasi fisik dan digital mereka; memberi mereka kendali, atau hak pilihan, atas alat yang kita buat; dan pertama-tama memikirkan kesejahteraan mereka di atas tepukan di punggung.

- **Lakukan perilaku manusia yang baik** dengan membangun sistem yang mengarah pada empati yang lebih besar terhadap keragaman keterampilan, budaya, dan kebutuhan kita.

Kami melihat daftar ini sebagai panduan atau inspirasi dan bukan daftar aturan. Kita semua bersama-sama dan hari-hari desain sensorik baru saja dimulai.

Sebuah kerangka sensorik

Selanjutnya, kami menyusun kerangka kerja, yang dapat Anda lihat pada Gambar 2-3, untuk melihat peluang dan koneksi.

Kerangka		
Penglihatan	Pendengaran	Sentuhan
Vestibular (Pergerakan)	Propriosepsi (Kesadaran spasial)	Agen (Kontrol & pilihan)

Gambar 2-3. Perincian indera manusia yang umum digunakan

Kami memecah indra manusia dan mesin kami sehingga kami dapat menyatukannya kembali dengan cara baru untuk memecahkan masalah dunia nyata. Apa saja masalah yang dapat dipecahkan oleh desain sensorik yang tidak dapat diselesaikan oleh media lain? Salah satu contohnya adalah menggunakan computer vision dan AR untuk memahami bahasa isyarat, menerjemahkannya ke teks, dan kemudian kembali lagi ke bahasa isyarat. Penglihatan komputer dapat memahami ekspresi wajah, dan jika digabungkan dengan gerakan tangan dan data biometrik, mesin dapat mengetahui perasaan Anda. Pembelajaran mesin sangat baik dalam melihat pola dalam sejumlah besar data sensorik. Organisasi sudah menggunakan data ini untuk membantu mengatur rencana kota dan memecahkan masalah iklim. Harapan saya adalah bahwa suatu hari itu akan memungkinkan kita untuk memahami satu sama lain dengan lebih baik.

Bagaimana kombinasi indra dan kecerdasan membantu kita menjadi lebih berempati di berbagai budaya dan cara berkomunikasi yang berbeda? Bisakah kita memberi orang kemampuan baru, mirip dengan bagaimana suara-ke-teks membuat saya lebih mudah mengekspresikan diri meskipun saya menderita disleksia? Kami memiliki begitu banyak pertanyaan dan begitu banyak peluang.

2.6 LIMA PRINSIP SENSORIK

Zach Lieberman dan Molmol Kuo, mantan seniman yang tinggal di Adobe, mengusulkan penggunaan tracking wajah AR sebagai input ke alat musik. Kedipan mata bisa memicu animasi dan gerakan mulut bisa menghasilkan musik.

Seniman mendobrak batasan dan menciptakan cara baru untuk melihat dunia dengan teknologi. Kita dapat meminta seniman untuk membuat pengalaman baru yang tidak pernah kita pertimbangkan sebelumnya. Karena semakin banyak seniman yang mendalami komputasi spasial dan desain sensorik, kami akan membutuhkan seperangkat prinsip untuk membantu memandu pengalaman ke arah yang akan dipahami pengguna. Generasi pertama pengguna *Sensory Design* tidak akan memiliki prasangka yang jelas tentang apa yang diharapkan. Prinsip-prinsip desain dapat memudahkan adopsi dan meningkatkan pengalaman komputasi spasial secara keseluruhan. Berikut ini adalah lima prinsip Desain Sensorik yang dibuat untuk memandu desainer menciptakan pengalaman berbasis komputasi spasial yang menarik dan dapat dipahami.

1. Pengalaman Intuitif Bersifat Multisensori

Produk kami akan intuitif ketika mereka multisensor. Dengan mengizinkan alat kami untuk mengambil dan menggabungkan indra yang berbeda, kami akan memungkinkan produk menjadi lebih kuat dan mampu lebih memahami maksud pengguna.

Kita adalah makhluk multiindrawi, jadi menambahkan lebih banyak indera meningkatkan kegembiraan dari sebuah pengalaman. Melihat band dalam konser lebih berkesan daripada mendengarkan rekaman melalui headphone. Terjun payung adalah pengalaman yang lebih mengubah hidup daripada menonton videonya. Kami senang hang out secara langsung dengan teman-teman daripada hanya Facebook atau Snap. Oksitosin, hormon ikatan sosial, dilepaskan saat kita merasakan pelukan yang sesungguhnya, bukan saat kita mengklik tombol 'suka'.

Bulan lalu saya pergi menonton konser band Massive Attack, sebuah acara yang melibatkan semua indra saya. Itu membuat saya menangis, dan pengalaman 90 menit memberi saya pemahaman yang lebih dalam tentang pesan *Massive Attack* yang belum saya dapatkan dari lebih dari 20 tahun mendengarkan album mereka. Saya percaya ini karena semua indra saya terlibat, memungkinkan saya untuk memahami dan merasakan pesan dengan cara baru dan konkret, cara yang tidak dapat diungkapkan hanya melalui suara atau layar 2D.

2. 3D Akan Menjadi Normcore

Dalam 5 sampai 10 tahun, desain digital 3D akan menjadi normal seperti desain digital 2D saat ini. Seperti fotografi, desktop *publishing*, dan internet sebelumnya, kita akan membutuhkan alat desain, perangkat keras tingkat konsumen, dan layanan cloud yang sudah tersedia, mudah digunakan, dan cepat digunakan untuk semua orang.

Saat ini, kami sedang bersenang-senang bereksperimen dengan AR seluler, menggunakannya sebagai filter efek khusus dunia nyata, yaitu wajah kami. Di masa

depan, hidup dengan AR akan lebih normal daripada selfie bagi generasi milenial saat ini.

Segera kami akan berharap dapat membuat seluruh lingkungan 3D kami menggunakan suara kami, gerakan tangan, dan lingkungan itu sendiri. Tubuh kita akan menjadi mouse dunia komputasi spasial masa depan, dan dunia di sekitar kita akan dapat diklik, dapat diedit, dirancang ulang. Input tradisional seperti keyboard, mouse, dan layar sentuh membuat perangkat lunak secara alami menjadi rumit. Mengontrol perangkat lunak secara spasial dengan semua indra alami kita dan tubuh manusia akan mengubah cara kita mengekspresikan kreativitas manusia kita.

Di dunia AR yang tidak memiliki teknologi 2D, mungkin tampak konyol melihat peta dua dimensi di perangkat seluler kita, daripada melihat melalui kacamata AR untuk melihat arah 3D yang diletakkan di atas jalan atau trotoar di depan kita. Menonton video terlebih dahulu untuk mengatur sistem audio rumah Anda akan tampak kuno ketika instruksi AR langsung dihamparkan ke peralatan memandu Anda segera.

Semua orang akan dapat berkreasi ketika inspirasi menghantam kita di ruang mana pun kita berada, tidak hanya saat kita berada di meja. Jika itu warna, cahaya, tekstur, gerakan, suara, atau bahkan objek, mereka dapat menangkap dalam 3D dengan perangkat AR mereka. Kami berharap dapat membuat menggunakan lingkungan 3D kami dan gerakan suara dan tangan kami sebagai mekanisme input, bukan mouse atau keyboard.

Input tradisional seperti keyboard, mouse, dan layar sentuh membuat perangkat lunak secara alami menjadi rumit. Mengontrol perangkat lunak dengan semua indra kita dalam 3D, akan melepaskan kekuatan super kreatif kita.

Misalnya, saya menderita disleksia, jadi mentransfer pikiran saya ke kertas sangat membuat frustrasi. Saat menulis kata-kata secara fisik, aliran kreatif saya hilang, dan saya menjadi tidak bisa berkata-kata. Saya menulis bagian ini menggunakan teknologi suara-ke-teks. Itu tidak sempurna, tetapi itu membantu saya menurunkan kata-kata dan suara saya di atas kertas.

3. Desain Menjadi Sifat Fisik

Produk kami harus bersifat fisik. Desain yang ditempatkan di dunia hanya akan diterima ketika mereka bertindak secara alami dan manusiawi. Kami masih akan meneriaki Alexa sampai teknologi mendengarkan dan merespons seperti yang dilakukan teman-teman kami. Ada standar UI baru ketika desain memasuki dunia.

Standar *User interface* baru untuk desain spasial menuntut desain digital yang ditempatkan di dunia bertindak seolah-olah mereka nyata secara fisik. Kami berharap mug virtual akan pecah seperti mug fisik jika kami melemparkannya ke tanah.

Sama seperti desain layar yang dipicu oleh klik mouse atau ketukan layar, desain di dunia dipicu oleh indra kita. Desain dan interaksinya harus terasa alami dan sesuai dengan konteks dunia di sekitar mereka. Kami terkadang dapat melanggar aturan ini, selama pengguna tidak mengganggu aplikasi tersebut rusak juga.

4. **Desain untuk yang Tak Terkendali**

Elemen desain yang ditempatkan di dunia tidak dapat dikontrol dengan cara yang sama seperti piksel pada layar. Pengalaman digital dalam ruang 3D harus beradaptasi dengan kondisi pencahayaan, dimensi, dan konteks lingkungan sekitar. Ini berarti desainer tidak dapat lagi mengontrol kamera atau tampilan. Pengguna bebas menentukan sudut pandang, lokasi, dan konteks mereka sendiri.

Ketika kami memamerkan Project Aero di Apple WWDC 2018, saya langsung mengerti apa yang dimaksud Stefano Corazza, pemimpin produk Adobe Project Aero yang berani, ketika dia berkata, “AR memaksa pembuat konten untuk memikirkan perasaan agensi (atau self- pilihan terarah), dan ini menumbuhkan lebih banyak empati terhadap pemirsa.” Memberi pemirsa kontrol atas kamera memberi mereka peran untuk dimainkan. Mereka menjadi bagian-pencipta. Saya melihat seorang pengguna berperan sebagai sinematografer saat orang tersebut menggerakkan kamera bertenaga AR melalui karya seni 2D berlapis yang ditempatkan secara virtual di atas panggung.

Cara lain kami menemukan desain untuk yang tak terkendali adalah melalui mata seniman kami yang menjelajah melalui program AR Residency Adobe yang diadakan selama tiga bulan, tiga kali per tahun. Dua dari seniman residensi ini adalah Zach Lieberman dan Molmol Kuo. Mereka berkolaborasi untuk membuat Weird Type, aplikasi AR iOS yang memungkinkan Anda menulis dan menganimasikan apa saja, di mana saja dalam ruang 3D. Setelah meluncurkan aplikasi, kita semua harus duduk dan melihat bagaimana tipografi di luar angkasa dapat ditata ulang. Orang-orang menggunakan Tipe Aneh untuk memandu seseorang melewati sebuah bangunan, menceritakan sebuah kisah tentang sebuah lokasi; membangun patung; dan berbagi perasaan oleh angin dengan menghidupkan kata, menerbangkan dan menyebarkan huruf secara acak ke angkasa, membuat teks lebih terlihat seperti salju (Gambar 2-4). Bentuk-bentuk komunikasi baru ini ditemukan dengan memberikan agensi kreatif kepada pemirsa AR, yang dengan sendirinya membuka media kreativitas baru.

5. **Buka Kunci Kekuatan Kolaborasi Spasial**

Saya percaya kekuatan kreatif dan ekonomi unik yang dimungkinkan oleh AR adalah kolaborasi spasial. Ketika Anda merasa seperti berada di ruangan yang sama, berkomunikasi secara alami dengan seluruh tubuh kita, merancang hal-hal digital-fisik secara ajaib dengan keputusan yang diperkuat oleh AI bersama anggota tim manusia nyata, maka kekuatan koneksi emosional dan fisik jarak jauh menjadi pendorong untuk adopsi dari komputasi spasial. Dengan kata lain, bisa dibilang, koneksi manusia adalah aplikasi pembunuh untuk AR.



Gambar 2-4. Gambar yang dibuat dengan aplikasi Jenis Aneh yang tersedia di App Store Apple. Salah satu seniman tinggal di Adobe, Nadine Kolodzey, mengambil ide kolaborasi AR satu langkah lebih jauh ketika dia berkata, "Saya ingin orang tidak hanya melihat gambar saya, tetapi menambahkan sesuatu." Kami menyadari saat itu dia memberi agensi pemirsa, kemampuan untuk menjadi artis juga. Saat itu Nadine menjadi pembuat alat dan penonton menjadi seniman. Dengan cara ini, AR memberikan kemampuan bercerita kepada semua orang, seperti yang dilakukan desktop publishing untuk media cetak dan internet untuk pengetahuan.

2.7 KISAH ADOBE AR

AR yang dipandu oleh AI akan sangat mengubah apa yang dibuat oleh desainer, cara perusahaan terhubung dengan konsumen mereka, dan memperluas cara kita berkolaborasi dalam kehidupan sehari-hari. Itulah sebabnya Adobe baru-baru ini mengumumkan Project Aero, alat desain AR seluler untuk desainer dan seniman.

Tujuan Project Aero adalah menghadirkan media baru AR ke dalam semua produk kami dan menetapkan disiplin desain untuk komputasi spasial yang digerakkan oleh AI. Berikut ini adalah sepotong masa depan komputasi spasial seperti yang saya lihat hari ini.

Dalam 5 hingga 10 tahun, akan tampak konyol untuk melihat peta 2D di layar kita daripada hanya melihat arah 3D yang digambar di dunia di sekitar kita. Wikipedia akan tampak kuno ketika Anda dapat mempelajari objek dan tempat di sekitar kita hanya dengan melihatnya dan memainkannya seperti mengalami mesin sinar-X tiga dimensi yang ajaib.

Desainer akan segera dapat berkreasi ketika momen inspirasi menerpa mereka, di mana pun mereka berada. Jika itu warna, cahaya, tekstur, gerakan, suara spasial, dan bahkan objek, mereka dapat menangkapnya dalam 3D dengan perangkat AR mereka. Kemudian, mereka dapat menambahkan elemen alami ke karya mereka yang sudah ada, membuat desain baru, atau berbagi inspirasi mentah. Saat ini, desainer sedang bersenang-senang dengan AR seluler yang menggunakannya sebagai filter efek khusus dunia.

Kami tahu bahwa adalah tanggung jawab kami di Adobe untuk membangun alat interaktif, animasi, dan memperkaya yang menjembatani kesenjangan antara hari ini dan masa depan bagi desainer kami dan desainer baru yang muncul.

Baru-baru ini, ketika *artist-in-residence* kami Nadine Kolodziey berkata, "Saya ingin orang tidak hanya melihat gambar, tetapi menambahkan sesuatu," kami menyadari bahwa dia memanfaatkan kebutuhan yang muncul untuk desain kolaboratif real-time yang dibuat mungkin dengan smartphone AR-enabled dan cloud AR. Adidas, "merek pencipta," menganggap konsumennya sebagai pencipta juga. Jadi, ketika kami meminta Adidas untuk membantu membangun alat AR yang "tepat" untuk kolaborasi pembuat konten, itu langsung masuk. Tapi cerita AR Adobe tidak dimulai atau diakhiri dengan Project Aero.

Dengan mengintegrasikan Aero secara mendalam ke dalam alat kami seperti After Effects; alat animasi 3D kami, Dimension CC; alat desain 3D kami, XD; alat desain UI kami, sekarang dengan suara, Adobe Capture; aplikasi kamera kami, yang memungkinkan Anda mengambil elemen dunia, bersama dengan layanan cloud kami; semua didorong oleh platform AI kami, Sensei, kami menciptakan ekosistem untuk membuka potensi AR.

Sama seperti ekosistem mesin yang menggabungkan fitur, kami menggabungkan indera, suara, sentuhan, penglihatan, dan proprioception (rasa ruang di sekitar kita) untuk memahami dunia. Mesin yang meniru indera manusia seperti indera penglihatan kita dengan AR hanya dirancang dengan baik ketika mereka bertindak seperti yang diharapkan: secara manusiawi. Kami masih akan meneriaki Alexa jika tidak mengerti apa yang saya katakan dan juga teman saya. Standar baru untuk desain sensorik yang baik ini telah mendorong Adobe Design untuk memikirkan kembali prinsip-prinsip desain, peran seorang desainer, dan mekanika inti dari alat kami.

2.8 KESIMPULAN

Saat kami menangani dunia baru komputasi spasial ini, saya mengingatkan diri sendiri apa yang dikatakan Scott Belsky, chief product officer Adobe: "Kreativitas adalah karya manusia yang paling banyak di dunia dan, terlepas dari perangkat dan media baru yang tak terhitung jumlahnya, orang-orang kreatif tetap menjadi pusatnya. Semakin dunia berubah, semakin penting kreativitas." Saya melihat kreativitas meledak di semua bagian kehidupan kita. Kreativitas sama pentingnya dengan literasi. Jadi, mari kita buat alat kreatif kita tersedia untuk semua orang, termasuk kemampuan, budaya, dan menghormati hak orang atas privasi dan transparansi yang berbeda.

Pada tahun 1970, desainer industri Dieter Rams dengan terkenal menulis 10 prinsip untuk desain yang baik. Hari ini, kita hidup di dunia di mana desain dapat mendorong kembali, merespons, atau merasakan apa pun. Desain adalah hal satu kali. Rams tidak memiliki teknologi pencitraan adaptif yang memahami niat, mengingat tindakan masa lalu, dan memberikan personalisasi antarmuka Anda. Desain telah berubah. Ini merespons secara empatik ke sistem saraf API.

Kami adalah orang-orang yang benar-benar membangun fondasi untuk periode ini. Ini kami, para desainer, insinyur, ilmuwan kognitif, pengusaha, dan banyak lainnya. Jika kita

menantang diri kita sendiri untuk melihat melampaui teknologi dan memfokuskan energi untuk membangun fondasi desain yang baik, kita sebenarnya dapat menciptakan masa depan yang sedikit lebih berempati secara alami. Mari tantang diri kita untuk mengembangkan alat yang menggunakan teknologi penginderaan yang memungkinkan produk menunjukkan empati untuk masa depan yang lebih baik.

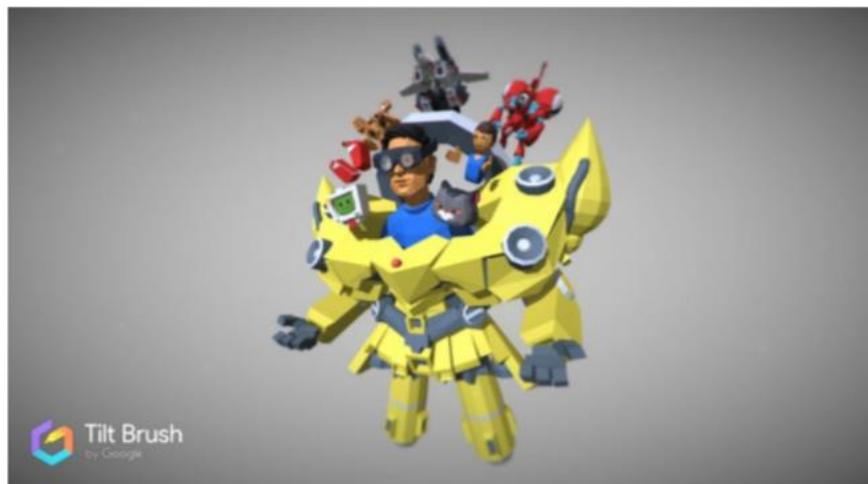
BAGIAN II

BAGAIMANA EXTENDED REALITY MENGUBAH SENI DIGITAL

Komputer telah selamanya mengubah cara kita berpikir tentang seni dan animasi, pertama dengan piksel dan kemudian dengan poligon. Dan sekarang akan ada revolusi lain berkat virtual reality (VR) dan augmented reality (AR). Karena kita sekarang memiliki tampilan yang sadar spasial, kita akhirnya bisa melihat objek digital dalam 3D yang sebenarnya. Ini berarti bahwa seni untuk VR dan AR harus dioptimalkan dengan cara yang unik untuk memanfaatkan sepenuhnya tampilan spasial ini. Dan dengan perangkat input yang sadar spasial, kita juga dapat berinteraksi dengan objek digital dalam 3D yang sebenarnya. Jadi, kita benar-benar dapat menggunakan VR untuk membuat seni 3D dengan cara baru dan intuitif.

Dalam Bab 3, Tipatat Chennavasin, seniman digital yang berubah menjadi kapitalis ventura, yang potret dirinya dapat Anda lihat di Gambar II-1, menjelaskan bagaimana VR meningkatkan cara seni dan animasi 3D dibuat sambil mendemokratisasi seni 3D. Dia membahas alat perintis yang berada di garis depan seni VR dan gerakan animasi dan mengapa mereka begitu penting. Sayangnya, karena keterbatasan waktu dan ruang yang tersedia dalam buku ini, ia tidak dapat membahas semua alat seni VR yang menakjubkan di luar sana, seperti pemodelan poli rendah Google Blocks atau pemodelan Gravity Sketch berbasis spline.

Namun, dengan mencakup konsep lukisan VR dan pemahatan VR, Anda harus memiliki kerangka kerja yang baik untuk memahami dampak VR pada seni 3D. Ada juga alat VR dan AR baru untuk pembuatan prototipe desain, seperti Microsoft Maquette dan Torch 3D, yang menyediakan lingkungan WYSIWYG untuk desainer spasial. Keduanya berada di luar cakupan bab pengantar ini, tetapi keduanya juga layak untuk ditelusuri dan sama transformatifnya untuk bidangnya masing-masing.



Gambar II-1. Potret diri VR oleh Tipatat Chennavasin, dibuat di Google Blocks dan Tilt Brush

Dalam Bab 4, seniman digital Jazmin Cano, yang potret dirinya menghiasi Gambar II-2, berbicara tentang beberapa tantangan dalam menciptakan seni untuk digunakan dalam VR

dan menjelaskan beberapa tip dan teknik praktik terbaik. Dia membahas berbagai teknik pemodelan dan tekstur yang digunakan dalam pembuatan seni 3D tradisional untuk dimasukkan ke dalam alur baru Anda untuk memastikan bahwa karya seni Anda tampak hebat sambil berjalan dengan lancar. Ini akan menjadi panduan Anda untuk menciptakan pengalaman VR terbaik yang dapat Anda tawarkan untuk membuat pengguna tetap nyaman. Semoga, kedua bab ini memberikan pemahaman yang baik tentang bagaimana VR dan AR memengaruhi dunia seni digital. Takeaway terbesar adalah bahwa alat dan teknik selalu berubah; menjadi seniman digital yang sukses berarti menjadi pelajar selamanya, mempelajari alat dan teknik baru dan seringkali merintis teknik ini sendiri dan membagikannya kepada komunitas.



Gambar II-2. Potret diri VR oleh Jazmin Cano, dibuat di aplikasi Finger Paint (oleh Mimicry), dilukis dengan High Fidelity

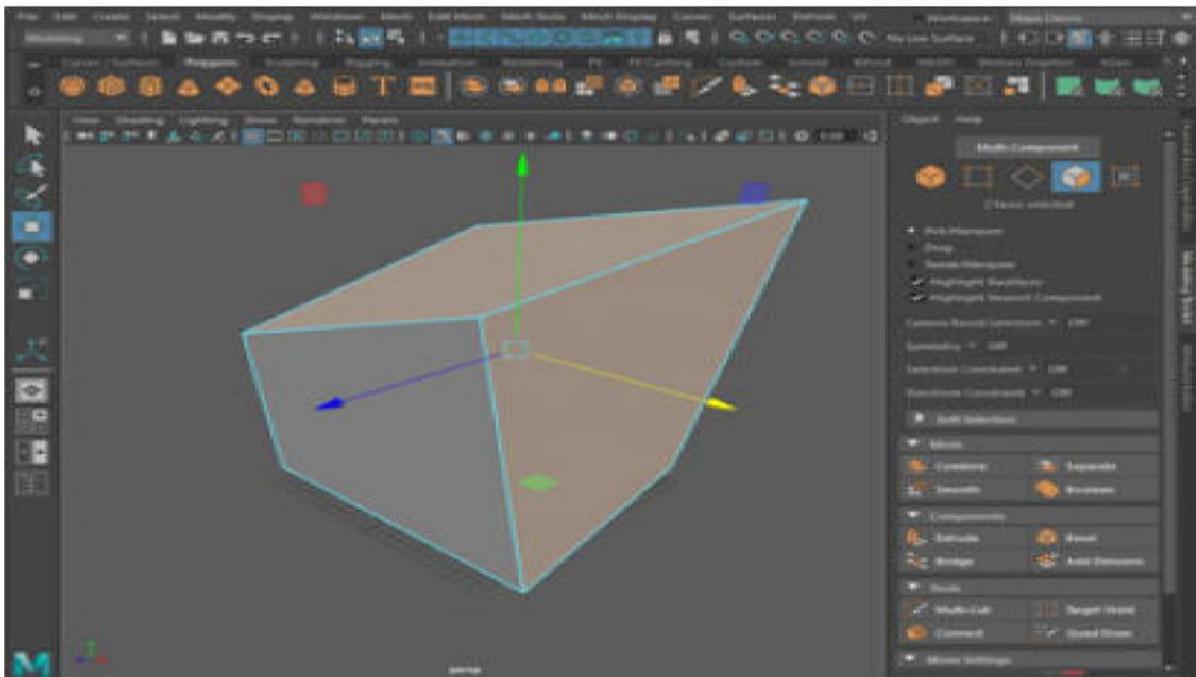
BAB 3

VIRTUAL REALITY (VR) UNTUK SENI

3.1 CARA MEMBUAT SENI 3D YANG LEBIH ALAMI

Secara tradisional, membuat seni 3D digital lebih seperti menggambar daripada melukis atau memahat. Banyak tantangannya adalah dalam memahami cara memanipulasi ruang 3D dengan antarmuka 2D. Untuk melihat objek 3D pada tampilan 2D, seniman sering kali bekerja dari beberapa tampilan, seperti mengerjakan gambar teknik. Objek 3D ini dibuat dari bentuk geometris, yang pada gilirannya terbuat dari simpul atau titik dalam ruang. Memindahkan titik-titik ini dalam ruang 3D dengan mouse 2D membutuhkan lebih banyak pemikiran abstrak daripada seni tradisional, yang lebih langsung diterapkan.

Melihat antarmuka untuk program 3D paling populer seperti Autodesk Maya (Gambar 3-1) dan 3D Studio mencerminkan kerumitan ini. Karena tantangan ini, sangat sedikit orang yang bisa membuat seni 3D. Kemudian ada gelombang baru program pemodelan 3D, seperti Z-Brush Pixologic (Gambar 3-2) yang memiliki pandangan berbeda secara fundamental. Program tersebut menggunakan tablet pena sebagai input dan antarmuka seperti pahatan yang mengubah bidang pemodelan 3D dan memungkinkan lebih banyak seniman untuk bekerja dalam 3D. Dengan menggunakan pena dan membiarkan seniman memanipulasi geometri secara langsung dengan gerakan yang lebih alami, penciptaan seni 3D semakin terdemokratisasi. Namun meskipun antarmukanya lebih langsung, masih canggung untuk mengerjakan objek 3D melalui tampilan 2D dan antarmuka 2D. Dengan diperkenalkannya gelombang konsumen Virtual Reality (VR), itu semua berubah.



Gambar 3-1. Antarmuka untuk pemodelan 3D dan perangkat lunak animasi populer Autodesk Maya (sumber: CGSpectrum)



Gambar 3-2. Seorang seniman digital yang bekerja dengan Wacom Pen Tablet dan Pixologic Z-Brush (sumber: Wacom)

Ketika kebanyakan orang memikirkan VR, mereka memikirkan *head-mounted display* (HMD), dengan sensor dan layar yang mengambil alih bidang visual dan sepenuhnya membenamkan seseorang ke dalam dunia digital. Namun yang sama, jika tidak lebih penting, adalah perangkat input atau controller yang dilengkapi dengan sensor serupa yang memungkinkan Anda berinteraksi dan memanipulasi dunia digital secara alami dan intuitif. VR HMD menjadi tampilan 3D terbaik, dan controller tangan yang dilacak menjadi antarmuka 3D terbaik. Tidak ada contoh kekuatan VR yang lebih baik daripada aplikasi yang menggabungkan tampilan dan input unik untuk memungkinkan pengguna berkreasi dan mengekspresikan diri tidak seperti sebelumnya. Untuk gelombang VR modern, semuanya dimulai dengan aplikasi bernama Tilt Brush, yang dapat Anda lihat di Gambar 3-3.



Gambar 3-3. Gambar promosi untuk Google Tilt Brush

Tilt Brush dikembangkan oleh dua orang startup Skillman & Hackett dan merupakan salah satu program seni pertama di VR modern. Karena dirancang dengan Oculus Development Kit 2, yang hanya memiliki HMD tetapi tidak memiliki perangkat input spasial, duo ini mendesainnya untuk digunakan dengan tablet pena Wacom. Pengguna menggambar pada bidang 2D yang dapat dimiringkan dan dipindahkan untuk melukis di beberapa bidang untuk dibuat dalam 3D. Ketika Valve dan HTC mengeluarkan kit pengembang Vive, Skillman & Hackett memanfaatkan controller tangan yang dilacak sepenuhnya yang disertakan dan tracking skala ruangan untuk memungkinkan pengguna melukis secara intuitif dalam ruang 3D.

Sekarang seluruh ruangan adalah kanvas dan cat digital mengalir dari ujung controller tangan dan melayang di angkasa, menciptakan nuansa magis yang melampaui kenyataan tetapi terasa benar-benar alami dan mudah dilakukan. Google kemudian akan mengakuisisi Skillman & Hackett, dan Tilt Brush akan dibundel dengan kit konsumen HTC Vive pertama. Ini akan menjadi salah satu aplikasi VR yang paling banyak digunakan hingga saat ini. Itu bukan hanya pelopor sejati dalam aplikasi seni, itu juga merupakan contoh cemerlang dari pengalaman pengguna (UX) yang sangat baik dan intuitif di VR.

Tilt Brush selalu dirancang sebagai aplikasi konsumen dan, dengan demikian, selalu menjadi alat yang sangat mudah didekati dengan desain yang sederhana dan menyenangkan. Ini fitur berbagai macam kuas yang memiliki efek visual seperti pencahayaan dan animasi, yang menciptakan tampilan bergaya yang sangat spesifik yang membedakannya dari alat lain di pasar dan pengguna pertama kali dapat dengan cepat mencapai hasil visual yang menakjubkan. Meskipun alatnya cukup fleksibel untuk mengakomodasi banyak gaya yang

berbeda secara default, mudah untuk mengenali seni yang dibuat di Tilt Brush, seperti yang Anda lihat pada Gambar 3-4.



Gambar 3-4. Lukisan VR Tilt Brush oleh seniman VR Peter Chan

Bahkan dengan seni yang dapat dikenali, Tilt Brush memiliki potensi yang tidak terbatas. Ini telah digunakan sebagai alat seni pertunjukan (lihat Gambar 3-5), untuk membuat video musik, ditampilkan dalam iklan televisi, dalam laporan berita untuk menambahkan infografis dinamis, untuk desain produksi, dan bahkan desain mode. Bahkan ada game yang dibuat di mana semua karya seni dibuat di Tilt Brush. Produk ini terus berkembang, dan dengan fitur dan fungsionalitas baru, produk ini akan terus membuka jalan bagi jenis seni digital baru di dunia spasial.



Gambar 3-5. Artis VR Danny Bittman memberikan pertunjukan Tilt Brush langsung di VMWorld 2017 (foto oleh WMWare)

Karena kesuksesan besar di dalam dan di luar industri VR, Google Tilt Brush adalah yang pertama mempopulerkan ide lukisan VR. Ini menggunakan metafora goresan cat untuk penciptaan yang sangat berbeda dari bagaimana seni 3D dibuat sebelumnya. Namun, itu bukan satu-satunya program lukisan VR yang ada. Tidak jauh dari tempat tim Google sekarang mengerjakan Tilt Brush, sebenarnya ada tim lain yang mengerjakan pendekatan yang sangat berbeda untuk lukisan VR sebagai bagian dari Oculus, yang disebut Oculus Story Studio.



Gambar 3-6. Gambar promosi Quill dari Facebook yang menampilkan karya seni dari Dear Angelica oleh seniman Wesley Allsbrook

Oculus Story Studio adalah grup pengembangan internal dalam Oculus untuk mengeksplorasi storytelling di VR. Dua film pendek VR animasi pertamanya, *Lost and Henry*, menggunakan alur kerja produksi seni yang cukup standar untuk membuat cerita pendek animasi real-time yang indah. Namun, untuk karya ketiganya, *Dear Angelica*, tampil dengan sesuatu yang sangat berbeda. *Dear Angelica* adalah mimpi nyata menggunakan gaya lukisan VR yang terasa lebih seperti animasi 2D daripada animasi 3D, seperti yang ditunjukkan pada Gambar 3-6. Untuk mencapai tampilan ini, tim Oculus Story Studio membuat program lukisan VR sendiri bernama Quill. Meskipun keduanya menggunakan pendekatan seperti goresan lukisan untuk penciptaan, mereka mencapai hasil yang terlihat sangat berbeda dan memiliki pendekatan yang sangat berbeda untuk UX.

Quill dibangun sebagai alat profesional dan UX-nya mencerminkan hal itu baik dalam tata letak visual maupun fungsi, menyerupai alat seni digital profesional lainnya seperti Adobe Photoshop. Itu juga tidak memiliki banyak efek bergaya yang dimiliki Tilt Brush, seperti pencahayaan waktu nyata atau kuas animasi. Ini memberi artis lebih banyak kontrol atas tampilan. Namun, itu juga membutuhkan lebih banyak upaya untuk mendapatkan hasil tertentu. Quill juga menghadirkan banyak ide baru tentang seni VR, dengan fitur menarik seperti skala yang hampir tak terbatas, tampilan yang bergantung pada perspektif, dan animasi berbasis garis waktu. Direktur seni dan seniman residen Oculus Goro Fujita telah memelopori beberapa kreasi yang benar-benar unik yang menggunakan fitur ini, seperti karya *Worlds in Worlds* miliknya, yang digambarkan pada Gambar 3-7, atau pohon yang mengubah

musim saat Anda memutarinya di tangan Anda. Contoh-contoh ini menunjukkan betapa orisinal dan menariknya karya seni jika tidak hanya dibuat tetapi juga dilihat dalam VR.



Gambar 3-7. Memperbesar perkembangan Dunia di Dunia seniman VR Goro Fujita, yang dilukis dengan Quill dari Facebook

Animasi khususnya benar-benar di mana Quill bersinar dan memungkinkan generasi seniman dan animator 2D untuk bekerja dalam 3D dengan cara yang sangat akrab dengan hasil yang benar-benar ajaib. Hanya dalam waktu tiga minggu, Fujita mampu membuat animasi pendek berdurasi enam menit yang indah berjudul *Beyond the Fence*. Tampilan visual dan nuansa animasi yang dibuat di Quill memiliki nuansa yang sangat organik lebih dekat dengan animasi 2D tetapi ada dalam 3D untuk menciptakan sesuatu yang orisinal dan istimewa yang akan memiliki dampak jangka panjang pada pengisahan cerita visual yang belum sepenuhnya kita pahami. Tapi lukisan virtual hanyalah salah satu pendekatan untuk menciptakan seni virtual. Pendekatan lain, lebih didasarkan pada pemahatan, juga sedang dikerjakan oleh tim lain di Oculus Story Studio, dan itu akan dikenal sebagai Medium (Gambar 3-8).



Gambar 3-8. Seni promosi Oculus Medium dari Oculus

Oculus Medium sebenarnya sedang dikerjakan jauh sebelum Quill. Sedangkan lukisan virtual adalah tentang memungkinkan seniman 2D tradisional untuk bekerja dalam 3D, patung virtual lebih tentang memungkinkan seniman 3D tradisional untuk bekerja dalam 3D. Dengan pahatan virtual, seniman selalu bekerja dengan volume 3D dengan cara yang sangat organik, membentuk dan membentuk tanah liat virtual dan kemudian menerapkan warna dan tekstur padanya. Seperti yang diilustrasikan Gambar 3-9, ini memungkinkan pembuatan model 3D yang lebih dekat dengan apa yang akrab dengan jalur produksi 3D yang ada dan dapat lebih mudah dikonversi ke geometri poligon 3D untuk digunakan dalam game dan film atau bahkan dicetak 3D ke dalam dunia nyata.

Selama masa pertumbuhannya, program ini masih memerlukan beberapa pemrosesan dan penyempurnaan, tetapi bahkan pada tahap awal, sangat jelas betapa lebih mudahnya bagi begitu banyak orang untuk membuat objek 3D dengan cara ini daripada aplikasi 3D non-VR tradisional. Sekarang digunakan dalam seni konsep; membuat prototipe; arsitektur, produk, dan desain mainan; dan bahkan aset game dan film terakhir. Menggunakan Medium mempercepat waktu iterasi desain, mengurangi praproduksi pada proyek sebanyak 80% menurut tim Oculus Medium. Namun, sekali lagi, di mana alat VR seperti Medium benar-benar membuat perbedaan adalah dengan memberdayakan seniman yang secara tradisional tidak akan membuat objek 3D dengan alat non-VR untuk sekarang membuat 3D secara intuitif.

3.2 VR UNTUK ANIMASI

VR tidak hanya bagus untuk membuat seni 3D; VR juga transformatif untuk menghidupkan seni itu. Mirip dengan kreasi seni 3D tradisional, animasi 3D tradisional mengandalkan pergerakan dan manipulasi titik 3D dalam ruang 3D tetapi dengan antarmuka

2D pada tampilan 2D. Dengan VR, kini ada peluang untuk menata kembali alur kerja ini menjadi lebih alami dan intuitif. Salah satu pendekatan adalah memperlakukan objek 3D seperti objek fisik nyata di dunia nyata dan mengambil pendekatan yang mirip dengan animasi stop motion di dunia nyata. Program animasi VR seperti Tvorl mengambil pendekatan seperti itu. Tetapi untuk animasi karakter, yang biasanya paling rumit dan memakan waktu untuk dilakukan, VR memiliki trik yang lebih baik.



Gambar 3-9. Artis VR mengerjakan pahatan 3D di Oculus Medium (gambar dari Oculus)

Motion capture adalah teknik menempatkan pelacak gerakan pada aktor dan kemudian menggunakan kamera untuk menangkap kinerja gerakan aktor tersebut dan kemudian menerapkannya pada model 3D untuk animasi karakter yang hidup. Teknik ini telah digunakan untuk animasi dalam game dan film selama beberapa dekade tetapi membutuhkan peralatan yang mahal dan banyak pembersihan. Agar VR berfungsi, komputer harus melacak kepala dan tangan pengguna dengan benar. Hal ini membuat VR secara default menjadi sistem penangkapan gerak yang sangat sederhana dengan biaya sepersepuluh atau kurang dari biaya pengaturan penangkapan gerak penuh.



Gambar 3-10. Seorang animator yang memerankan pertunjukan dalam VR yang diterjemahkan ke dalam karakter digital secara *real-time* dengan Mindshow

Pembuat program animasi VR, Mindshow, menyadari hal ini dan bahwa siapa pun yang memiliki headset VR berpotensi melakukan beberapa animasi karakter dasar dan menjadi pendongeng. Sekarang, alih-alih menggambar ribuan gambar atau menghabiskan waktu sehari-hari untuk memindahkan titik-titik di layar, seorang animator bisa saja memerankan Scene seperti yang dilakukan aktor di lokasi syuting. Demikian juga, aktor yang sama kemudian dapat masuk dan memerankan bagian lain, bekerja dengan rekaman. Ini mempersingkat animasi dasar dengan mengubah proses yang biasanya membutuhkan waktu seminggu untuk diselesaikan menjadi hanya beberapa menit, dan memungkinkan bahkan satu orang menjadi studio animasi. Meskipun kualitasnya belum setingkat Hollywood, bahkan pada tahap awal, hasilnya mengesankan dan menjanjikan. Proses ini adalah salah satu alasan utama mengapa Mindshow, yang digambarkan pada Gambar 3-10, dinominasikan untuk Penghargaan Emmy untuk Inovasi Luar Biasa dalam Media Interaktif pada tahun 2018.

Dalam tiga tahun terakhir, sungguh menakjubkan menyaksikan bagaimana VR mengubah cara seni dilihat dan dibuat. Ada bahasa penceritaan spasial yang baru mulai kami jelajahi berkat pengembang yang membuat alat luar biasa seperti Tilt Brush, Quill, Medium, Mindshow, dan lainnya. Ini terutama mencakup karya-karya penting dan seniman yang menggunakan alat-alat ini untuk menciptakan seni. Pada awalnya, banyak seni tampaknya mencerminkan apa yang telah kita lihat sebelumnya, tetapi kita sudah mendapatkan sekilas hal-hal yang unik untuk medium, dan saya pribadi tidak sabar untuk melihat ke mana kita pergi dari sini.

VR telah mendemokrasi pembuatan 3D dengan cara yang sama seperti komputer desktop mendemokratisasikan pembuatan 2D. Sekarang, orang-orang yang tidak pernah menganggap dirinya sebagai seniman tradisional dapat bergabung dalam percakapan membentuk masa depan seni visual dan spasial. Dengan VR, siapa pun dapat membuat bagian instalasi seukuran gudang yang akan menghabiskan biaya puluhan ribu untuk bahan saja. Tapi, lebih baik lagi, sekarang mereka bisa membuat instalasi seukuran planet yang tidak mungkin dilakukan di dunia nyata. VR benar-benar kanvas pamungkas, tidak hanya untuk memberdayakan seniman, tetapi dengan menyalakan seniman dalam diri setiap orang untuk menciptakan karya yang hanya dibatasi oleh imajinasi mereka.

BAB 4

OPTIMISASI 3D ART

4.1 PENGANTAR

Dalam bab ini, saya membahas mengapa pengoptimalan merupakan tantangan besar dalam hal mengembangkan aset untuk *Virtual Reality* (VR) dan *augmented reality* (AR), dengan fokus utama pada sisi VR. Saya berbagi pendekatan dan proses pemikiran yang berbeda untuk dipertimbangkan di berbagai bidang yang terlibat dalam menciptakan seni 3D. Alih-alih berfokus pada alat, teknik, dan tutorial tertentu, Anda akan melihat tampilan keseluruhan tentang apa yang dapat Anda lakukan untuk mengoptimalkan seni 3D Anda. Saya berbagi mengapa penting untuk benar-benar menjaga pengoptimalan sebagai prioritas tinggi dalam keseluruhan proses desain. Mari kita mulai dengan membahas mengapa pengoptimalan untuk VR adalah hal baru bagi sebagian besar artis yang mulai bekerja dengan konten VR.

Monitor LCD 2D tipikal memiliki kecepatan refresh 60 Hz. Saat Anda melihat monitor datar yang berjalan pada kecepatan ini, Anda akan memiliki pengalaman visual yang fantastis. Hal ini memungkinkan pengembangan aset tradisional menjadi "berat", dengan berat dalam keadaan ini yang berarti bahwa aset memiliki jumlah poli yang lebih tinggi dan gambar tekstur yang lebih besar, bersama dengan Scene itu sendiri yang memiliki jumlah model yang lebih tinggi untuk dilihat.

Tampilan yang dipasang di kepala (HMD) berjalan pada 90 Hz. Untuk mendapatkan pengalaman yang nyaman dan meyakinkan, konten VR harus berjalan pada 90 frame per detik (FPS). Jika pengalaman Anda berjalan lebih rendah dari 90 FPS, Anda berisiko menyebabkan ketidaknyamanan pengguna Anda. Ketidaknyamanan dapat mencakup sakit kepala, mual, dan sakit mata. Ini akan menghasilkan pengalaman di bawah standar yang ingin segera ditinggalkan oleh pengguna. Karena tingkat ketidaknyamanan yang tinggi ini, Anda tidak boleh memaksa pengguna ke dalam pengalaman frekuensi gambar rendah.

Segera, VR akan memasuki bidang lain. Alih-alih hanya membuat VR, Anda akan membuat alat, pengalaman, aplikasi, dan sebagainya di lapangan, dan VR adalah media Anda. Beberapa bidang ini sudah terbiasa membuat konten 3D, dan pengoptimalan akan penting untuk dipelajari. Akan ada banyak individu yang pengalaman latar belakangnya mungkin tidak mempersiapkan mereka dengan baik untuk perubahan besar dalam pengembangan aset ini, dan ini akan menjadi tantangan untuk menyesuaikan diri dengan proses baru ini. Berikut adalah beberapa contoh posisi terkait industri yang perlu dipelajari untuk dibuat dengan mempertimbangkan pengoptimalan:

- Render resolusi tinggi (membuat model realistis dari objek nyata)
- Game kelas atas untuk PC
- Penciptaan seni dalam VR

Contoh-contoh ini memiliki manfaat yang tidak lagi dapat dimanfaatkan untuk VR dan AR. Tidak termasuk headset tethered kelas atas seperti Oculus Rift atau HTC Vive, sebagian besar perangkat lain di luar sana akan lebih ringan dan lebih portabel. Anda harus ingat bahwa semakin besar file Anda dan semakin banyak konten dan panggilan, semakin dekat Anda dengan risiko pengguna memiliki kinerja yang buruk.

Individu yang membuat konten untuk film dan rendering memiliki hak istimewa untuk membuat model 3D dengan jumlah poli yang tinggi. Mereka tidak terbatas pada kompleksitas model mereka atau jumlah data rendering yang diperlukan komputer untuk memvisualisasikannya. Berikut adalah contoh dari apa yang dapat terjadi ketika seseorang yang baru mengenal pengoptimalan mencoba mengembangkan untuk VR:

Buat Model 3D Kamera

Model terkirim

Model kamera poli-tinggi dengan tekstur resolusi tinggi (tekstur 4096 x 4096)

Masalah

Model ini mengambil sebagian besar anggaran jumlah poli Scene, sehingga kualitas konten lainnya harus dikorbankan. Jika sisa konten perlu mempertahankan kualitas dan ukurannya saat ini, Anda mengalami masalah kinerja. Pengembang perlu menyeimbangkan seni mana yang memiliki prioritas lebih rendah dan memberikan lebih banyak ruang untuk model kamera poli-tinggi.

Tetapi mengapa ini menjadi masalah besar bagi pengembang? Jika latar belakang orang tersebut membuat model untuk dirender untuk foto, kemungkinan besar mereka terbiasa membuat dengan jumlah poli yang tinggi. Tidak jarang melihat angka tinggi mulai dari 50.000 segitiga (disebut "tris" dalam industri) hingga 1.000.000 segitiga. Namun ini tidak diterjemahkan dengan baik ke rendering VR waktu nyata. Seperti yang dinyatakan sebelumnya, masalah kinerja akan mencegah pengguna akhir memiliki pengalaman yang berkualitas.

Pilihan untuk dipertimbangkan Berikut adalah beberapa hal untuk mencoba memecahkan masalah:

- Menjalankan alat penipisan untuk mengurangi jumlah poli secara otomatis. Anda dapat menemukannya di perangkat lunak pemodelan 3D yang populer. Mereka biasanya melakukan pekerjaan yang baik untuk menghilangkan 50% dari jumlah segitiga tanpa mempengaruhi bentuk dan siluet model.
- Lihatlah tata letak UV model (sumbu tekstur 2D yang diproyeksikan ke 3D). Apakah tekstur UV ditata untuk memanfaatkan seluruh ruang persegi? Apakah UV menskalakan dan memprioritaskan area yang paling membutuhkan detail untuk ditampilkan? Kami mengeksplorasi tekstur dan bahan secara lebih rinci nanti di bab ini.

Pilihan bagus lainnya untuk dipertimbangkan adalah apakah model Anda akan memasuki tempat VR sosial yang mengizinkan konten buatan pengguna (UGC)? Ini kemungkinan besar

akan terus menjadi tantangan untuk waktu yang lama. Perlu diingat bahwa semakin banyak avatar yang ada di suatu ruang, semakin sedikit anggaran yang harus dimiliki setiap orang untuk menghormati frame rate setiap orang, memungkinkan pengalaman yang baik.

Solusi Ideal Solusi terbaik adalah mengurangi jumlah segitiga model ke minimum absolut yang dapat dimilikinya tanpa mempengaruhi bentuknya. Kurangi ukuran tekstur ke ukuran terkecil yang dapat dimilikinya tanpa memaksa model menjadi buram atau memiliki kualitas yang lebih rendah dari yang diinginkan. Pastikan saat objek ditempatkan di lingkungan akhirnya yang memungkinkan cukup waktu untuk frame rate sistem agar pengalaman terasa alami. Mari kita rekap. Mengapa penting untuk mengoptimalkan seni 3D Anda?

Setiap model di lingkungan 3D Anda akan memengaruhi kinerja pengalaman Anda. Semakin banyak Anda menambahkan, semakin Anda perlu mempertimbangkan seberapa dekat Anda dengan anggaran Anda. Bicarakan dengan tim Anda untuk menentukan berapa anggaran ideal Anda.

Pertimbangan lain untuk tujuan model 3D Anda termasuk platform VR sosial. Ada beberapa platform VR sosial di luar sana yang dibangun dengan UGC. Kemungkinan besar Anda akan ada di ruang ini sebagai avatar, dan jika Anda dapat menyesuaikan avatar Anda, ingatlah bahwa semua yang Anda pelajari di sini juga berlaku di sana. Seperti dengan sisa dari apa yang akan Anda pelajari di sini, cobalah untuk menjaga segala sesuatu tentang avatar Anda dan apa yang Anda kenakan poli rendah dan dengan jumlah panggilan undian terkecil yang dapat Anda buat. Anda mungkin mengalami filter yang membantu menurunkan seberapa banyak Anda membuat orang mengunduh, tetapi pikirkan terlebih dahulu apa yang Anda minta untuk dirender oleh layar orang. Perhatikan perangkat keras dan koneksi mereka dan buat diri Anda mudah untuk merender dan mengunduh. Mari lanjutkan dengan ikhtisar komprehensif tentang apa yang perlu Anda periksa saat membuat model 3D untuk VR dan AR.

Anggaran jumlah poli

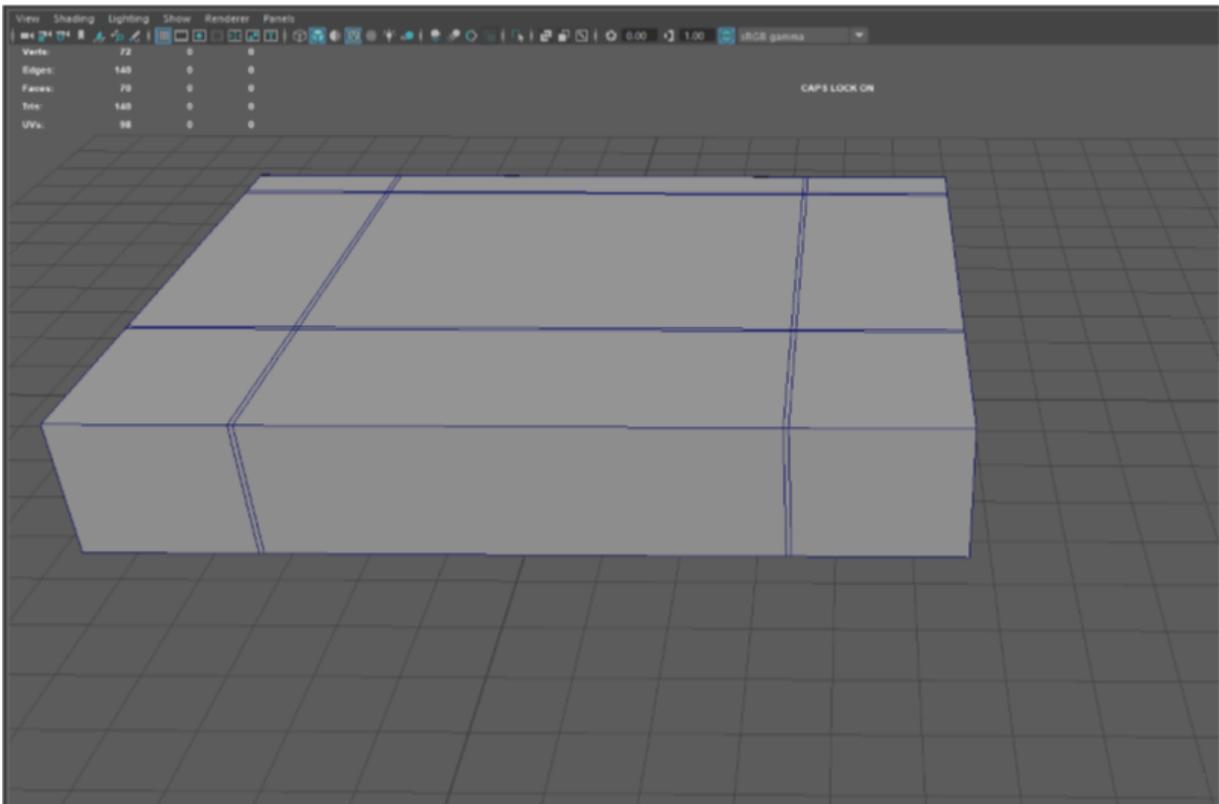
Apakah Anda memiliki jumlah poligon konkret yang tidak dapat Anda lewati dalam sebuah Scene? Apakah Anda memiliki batas jumlah poli per model? Selalu mencari jumlah segitiga. Hitungan wajah tidak selalu akurat untuk mengukur berapa banyak poli yang Anda miliki pada model Anda. Wajah yang terdiri dari empat simpul, seperti persegi, sebenarnya adalah dua segitiga dalam satu.

Hapus semua wajah yang tidak akan pernah terlihat. Jika Anda menciptakan lingkungan jalan di mana interior bangunan tidak akan pernah dimasuki, pemandangan hanya memerlukan fasad bangunan. Bagian belakang dinding dan interior tidak diperlukan. Jika Anda menggunakan konten yang telah dibuat, Anda dapat menghapus semua yang tidak akan terlihat.

Jika Anda sedang mengerjakan model 3D yang akan jauh dari pengalaman Anda, mereka tidak memerlukan semua detail yang mungkin Anda inginkan jika mereka lebih dekat. Pintu dan jendela dapat dimodelkan dan bertekstur dengan detail yang lebih sedikit. Semakin rendah jumlah poli Anda, semakin baik. Bagian berikut menyajikan beberapa hal yang perlu diingat saat membuat model.

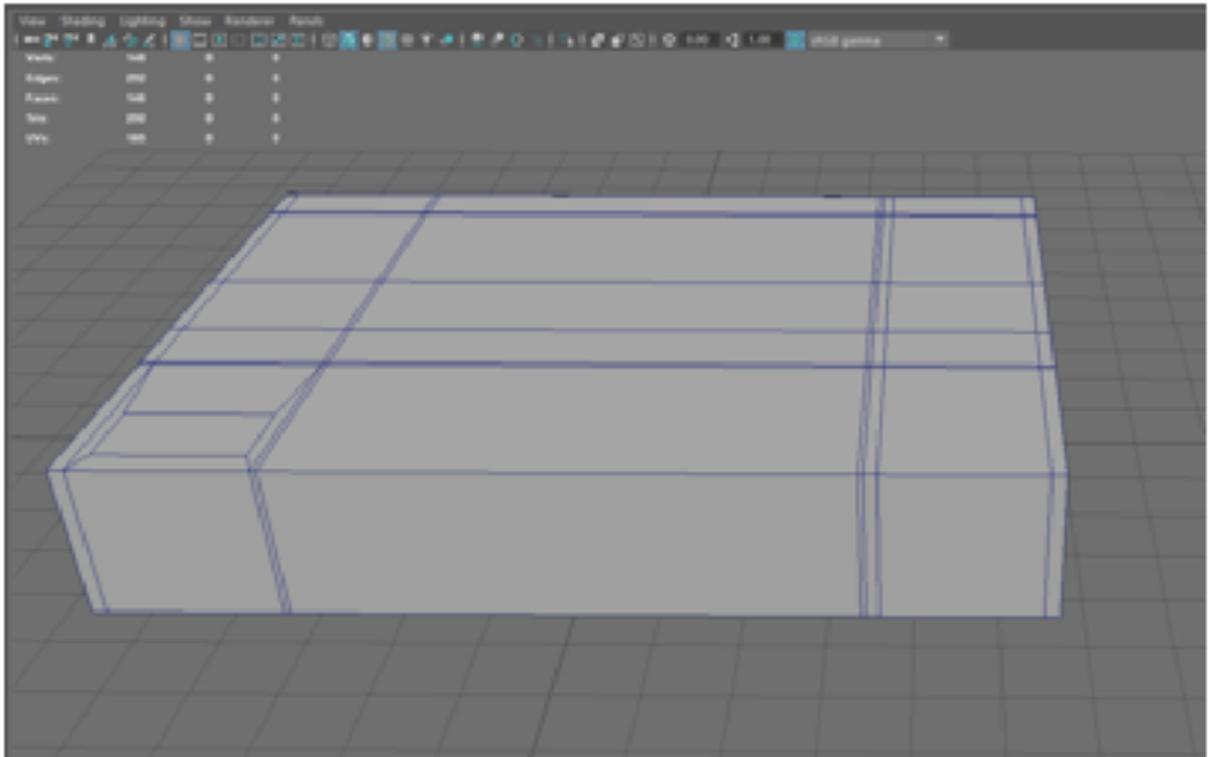
4.2 TOPOLOGI SENSORIK

Periksa loop tepi dan temukan loop tepi yang tidak berkontribusi apa-apa lagi pada bentuk. Jika tepi berjalan melintasi area datar, Anda akan tahu bahwa itu tidak diperlukan jika Anda menghapus seluruh tepi dan tidak menemukan perbedaan dalam siluet. Jika masih menahan bentuk dan memiliki kurva yang diinginkan, Anda sedang dalam perjalanan untuk mengurangi jumlah poli. Bahkan ada beberapa area di mana Anda dapat membawa tepi untuk bergabung dengan yang lain. Periksa kembali bahwa semua loop tepi yang dihapus tidak meninggalkan simpul apa pun, dan hapus simpul yang tidak menghubungkan tepi apa pun.

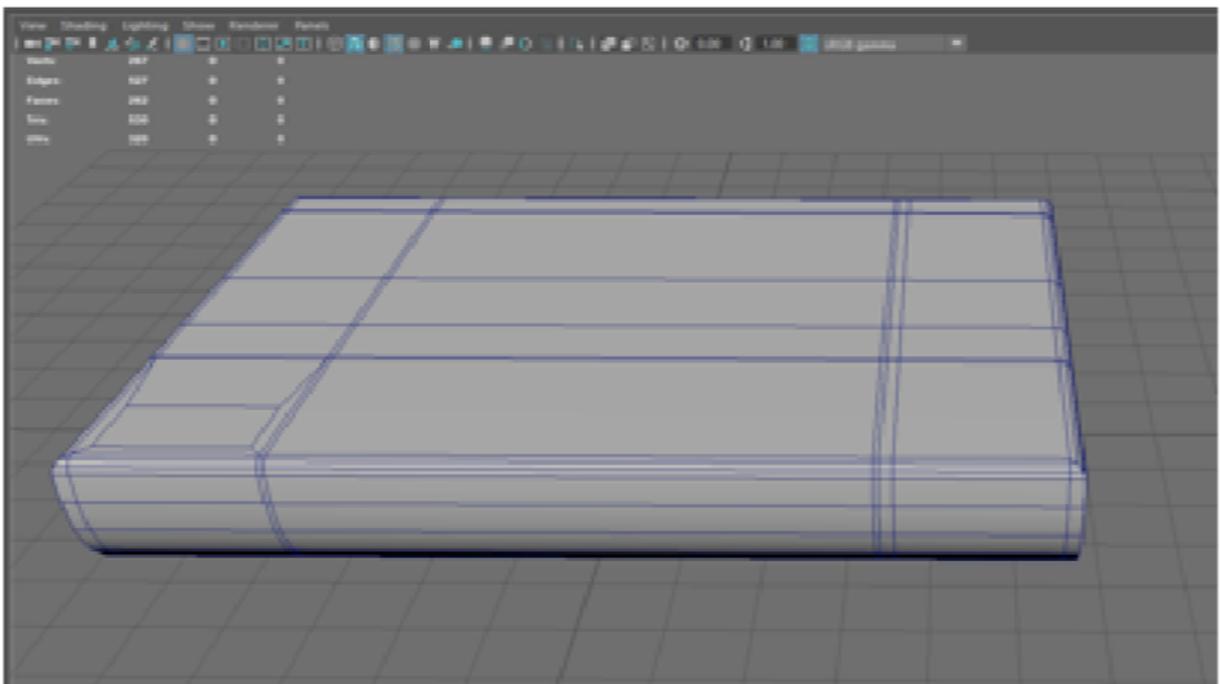


Gambar 4-1. Pass pertama di konsol game: bentuk dasar terbentuk; jumlah segitiga: 140

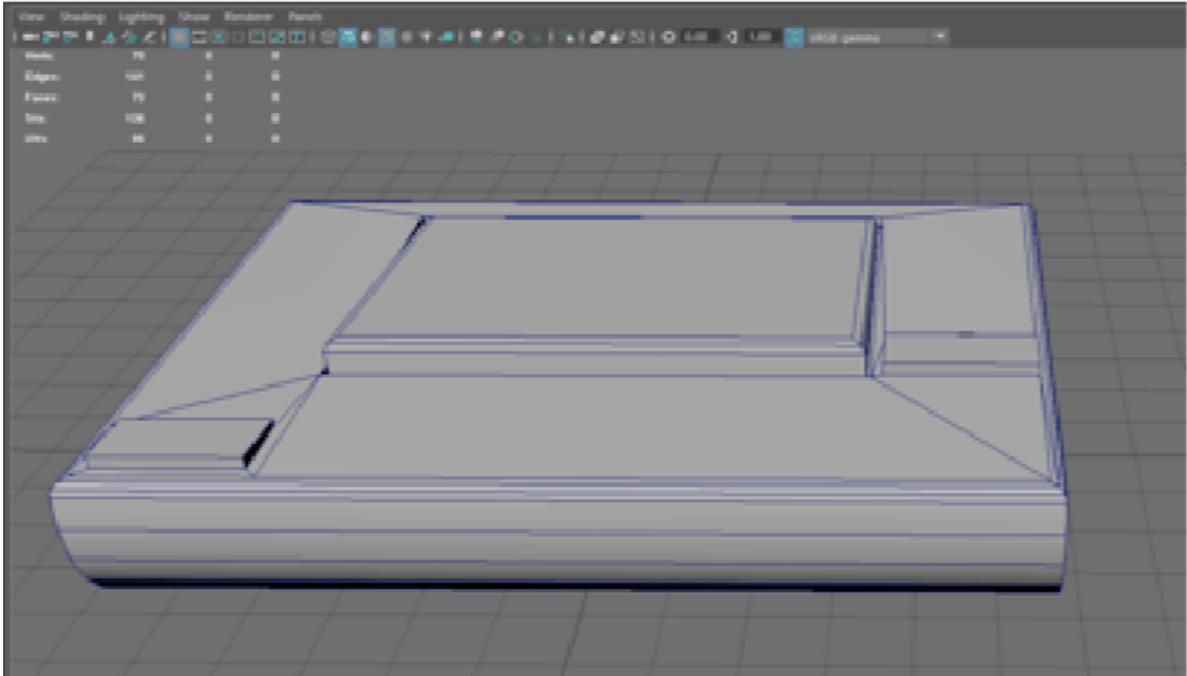
Gambar 4-1 hingga 4-4 menunjukkan proses pembuatan konsol game. Pada Gambar 4-1, Anda dapat melihat, dalam mode wireframe, awal pembuatannya menggunakan loop tepi untuk menentukan di mana lebih banyak geometri akan dibutuhkan; ada dua langkah antara poligon pereduksi dan versi final pada Gambar 4-4, yang menghasilkan lebih sedikit segitiga daripada lintasan pertamanya.



Gambar 4-2. Pass kedua di konsol game: menentukan di mana wajah dan tepi akan terangkat dan melengkung; jumlah segitiga: 292



Gambar 4-3. Pass ketiga di konsol game: melembutkan tepi dan mulai memikirkan penghapusan tepi; jumlah segitiga: 530

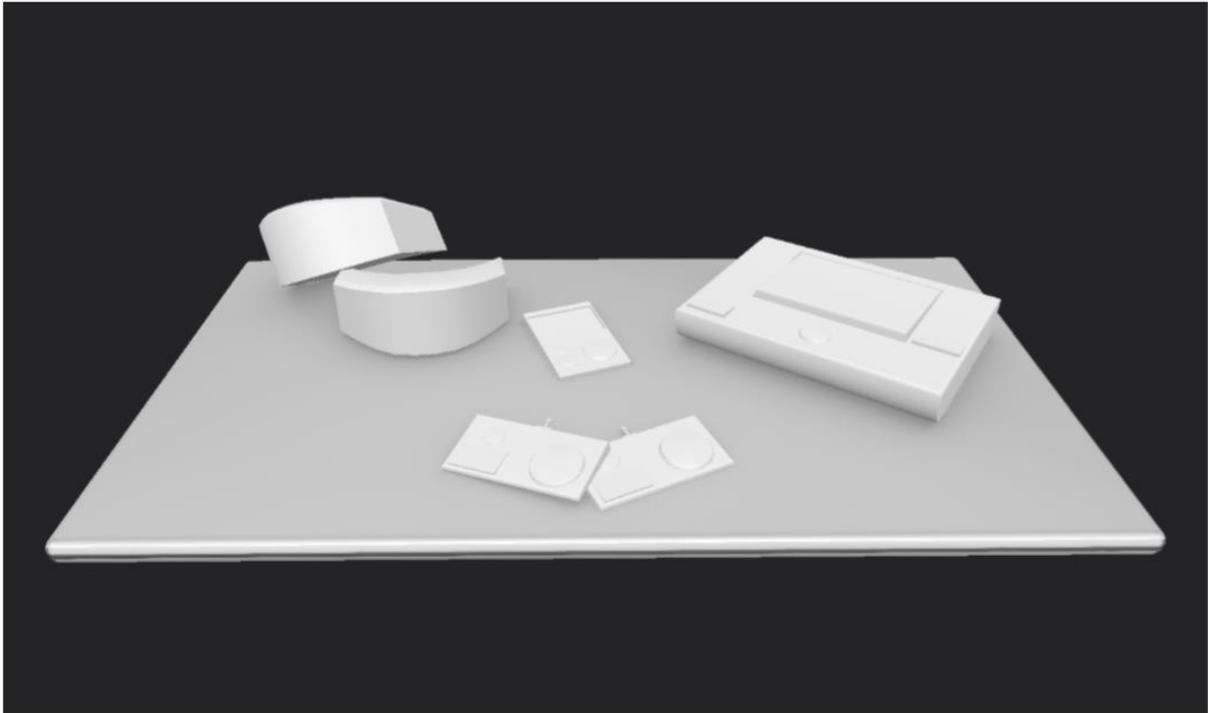


Gambar 4-4. Versi keempat dan terakhir: menghapus tepi yang tidak berkontribusi pada bentuk model; jumlah segitiga: 136

Proses yang ditunjukkan pada Gambar 4-1 hingga 4-4 serupa dengan proses yang dilakukan untuk memodelkan beberapa aset lagi untuk set konsol game ini. Gambar 4-5 menggambarkan hasil himpunan. Ini berisi beberapa model dalam satu mesh gabungan yang siap untuk diterapkan teksturnya. Bersama-sama, mereka akan berbagi satu atlas tekstur. Nanti di bab ini, Anda akan melihat bagaimana tekstur atlas terlihat.

Berikut adalah beberapa hal yang perlu diingat saat membuat model:

- Hindari n-gon. N-gon adalah wajah yang memiliki lebih dari empat sisi. Sebagian besar mesin memiliki masalah dengan n-gon. Mereka dapat menyebabkan masalah dengan tabrakan, mereka mungkin salah total, dan bahkan tidak terlihat. Perangkat lunak pemodelan 3D seperti Autodesk's Maya memberi Anda opsi untuk membersihkan pemandangan dan menghapus semua n-gon yang ditemukan.
- Jalankan pembersihan menggunakan perangkat lunak pemodelan Anda untuk menemukan dan menghapus semua permukaan koplanar. Anda mungkin sering menemukan wajah licik yang tersembunyi di dalam tiruannya sendiri, yang akan tampak tidak terlihat oleh mata telanjang dan akan meningkatkan jumlah poli Anda. Ada juga masalah z-pertempuran. Pertarungan Z adalah ketika ada dua wajah yang menempati ruang 3D yang sama.
- Nyalakan penampil untuk memastikan bahwa garis normal menghadap ke arah yang diinginkan. Normal akan ditampilkan dari satu arah di mesin pilihan Anda, jadi jangan biarkan perangkat lunak pemodelan 3D membodohi Anda dengan rendering dua sisi.



Gambar 4-5. Lihatlah aset sebelum mereka menerima materinya

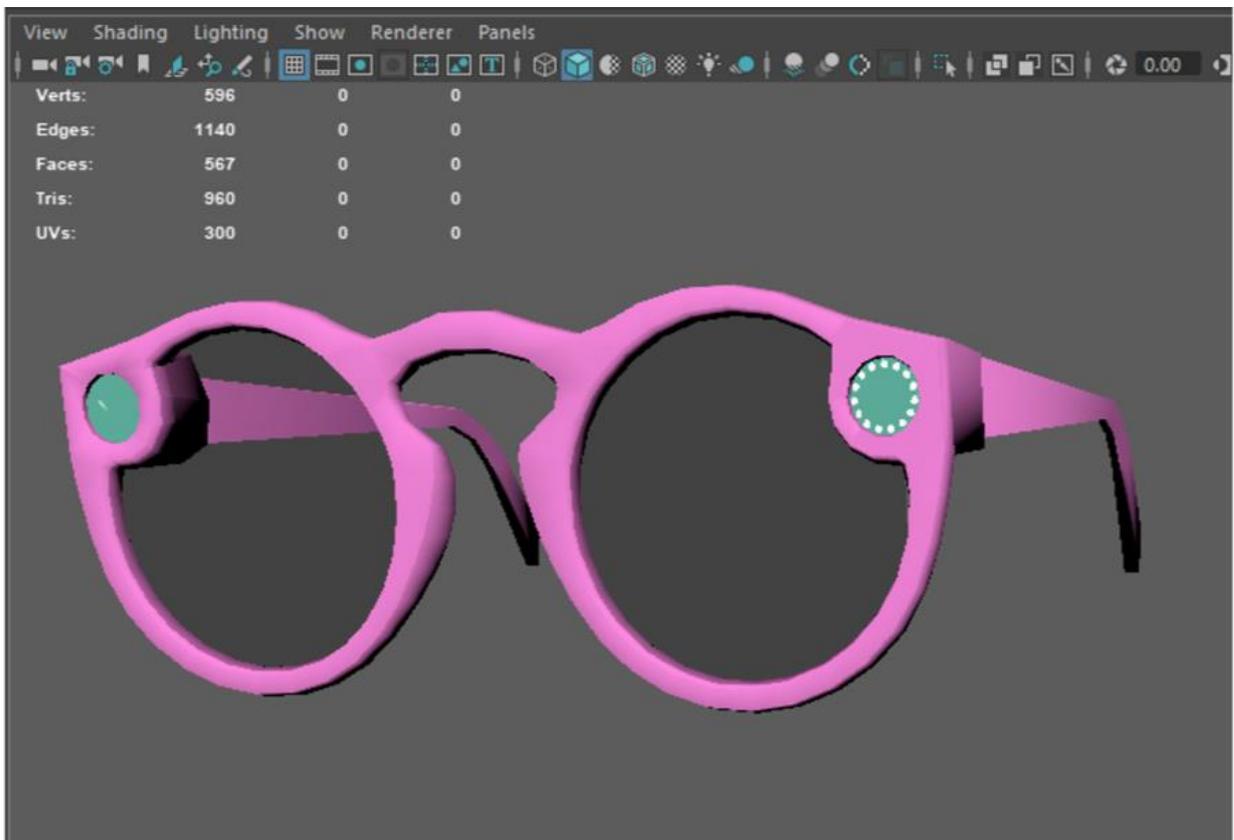
Penting untuk memikirkan semua pertimbangan ini di awal sebelum Anda mulai mengerjakan model 3D. Gambar 4-6 menyajikan contoh proyek pengoptimalan yang saya kerjakan secara pribadi. Saya diberi model kacamata 3D yang terdiri dari 69.868 segitiga. Jumlah ini lebih banyak dari avatar saya sendiri, yaitu sekitar 40.000, termasuk tubuh, pakaian, rambut, dan aksesoris. Kacamata tersebut dibeli dari toko online yang menjual file model 3D, dan jelas bahwa sang seniman membuat ini dengan maksud untuk menunjukkan bahwa mereka dapat membuat model untuk mencocokkan seperti apa objek tersebut dalam "kehidupan nyata". Sang seniman membuat model masing-masing dan setiap bagian, termasuk engsel untuk pelipis.

Karena saya akan membuat kacamata ini untuk dipakai orang di platform VR sosial, saya tahu bahwa sebagian besar detailnya tidak akan terlihat atau dibutuhkan, jadi saya menghapus banyak bagian itu. Saya berhasil mempertahankan tampilan kacamata sambil menghapus dan mengarahkan sebagian besar loop tepi. Hasil akhirnya hanya di bawah 1.000 segitiga.

Khususnya, untuk penggunaan AR, mendapatkannya di bawah 1.000 segitiga akan menjadi keharusan mutlak. Pada HoloLens, misalnya, Anda ingin membidik maksimal sekitar 60.000 segitiga di seluruh Scene. Kecuali jika aplikasi sangat berfokus pada pemeriksaan sepasang kacamata hitam yang detail secara realistis, Anda ingin menguranginya sepenuhnya seperti yang saya lakukan dalam contoh ini. Gambar 4-7 menyajikan *close-up* yang menunjukkan tepi keras yang dapat Anda lihat di sekitar bagian bulat dari bingkai, yang tidak terlalu mencolok jika dilihat dari kejauhan.



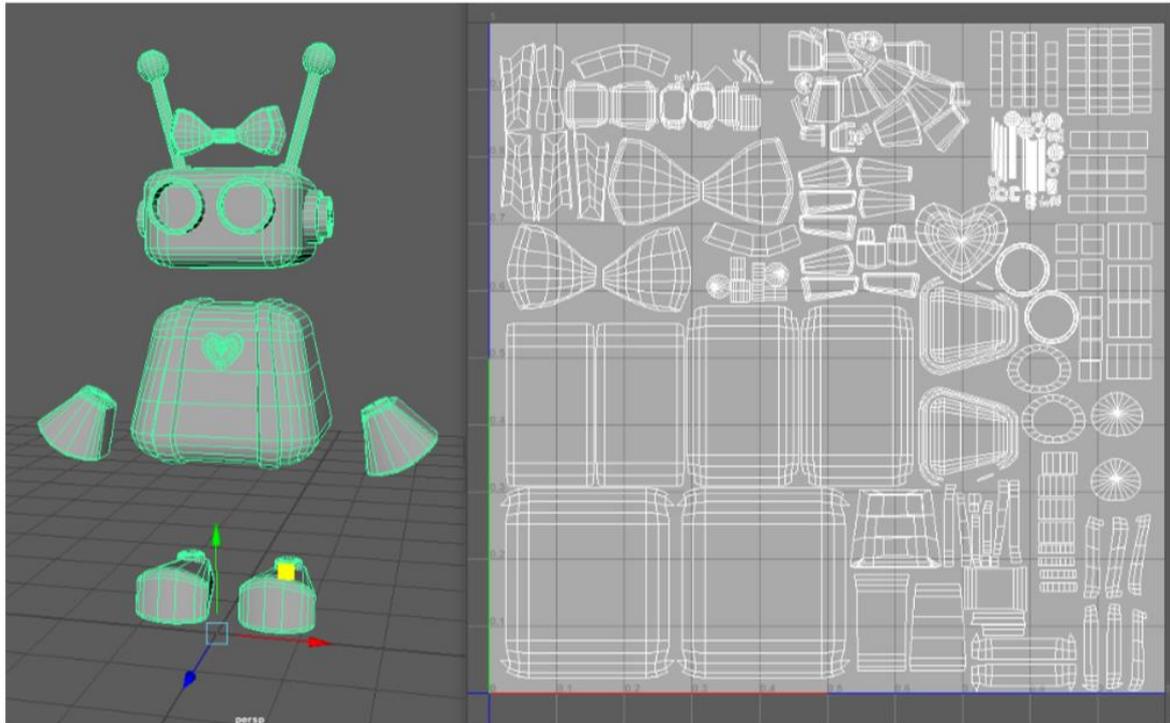
Gambar 4-6. Model kacamata untuk digunakan di ruang VR sosial



Gambar 4-7. Contoh tepi keras di sekitar bagian bingkai yang membulat

4.3 BAKING

Trik lain yang dapat Anda lakukan untuk membantu penghitungan poli Anda adalah dengan baking detail model poli-tinggi Anda menjadi model poli-rendah. Dengan melakukannya, Anda dapat membuat peta normal yang akan mengelabui pemirsa untuk melihat ketinggian dan kedalaman yang tidak ada pada geometri itu sendiri. Sekarang setelah kita membahas banyak hal tentang model, mari kita bicara tentang membuka bungkus UV dan melukis tekstur.



Gambar 4-8. Ini adalah bagian yang membentuk robot dan tekstur yang menyertainya

UV digunakan untuk menggambarkan model 3D pada bidang datar. UV tersebut merujuk pada tekstur yang digunakan model agar informasi warna dan material dipetakan dengan tepat. Untuk pengoptimalan, mari membahas pendekatan pembuatan tekstur yang dibuat, dengan tujuan menjaga jumlah panggilan undian tetap rendah.

Sebuah atlas tekstur adalah gambar tekstur yang berisi data yang menjelaskan bahan terbuat dari apa. Itu selalu lebih baik untuk membuat atlas tekstur karena secara drastis mengurangi jumlah panggilan undian.

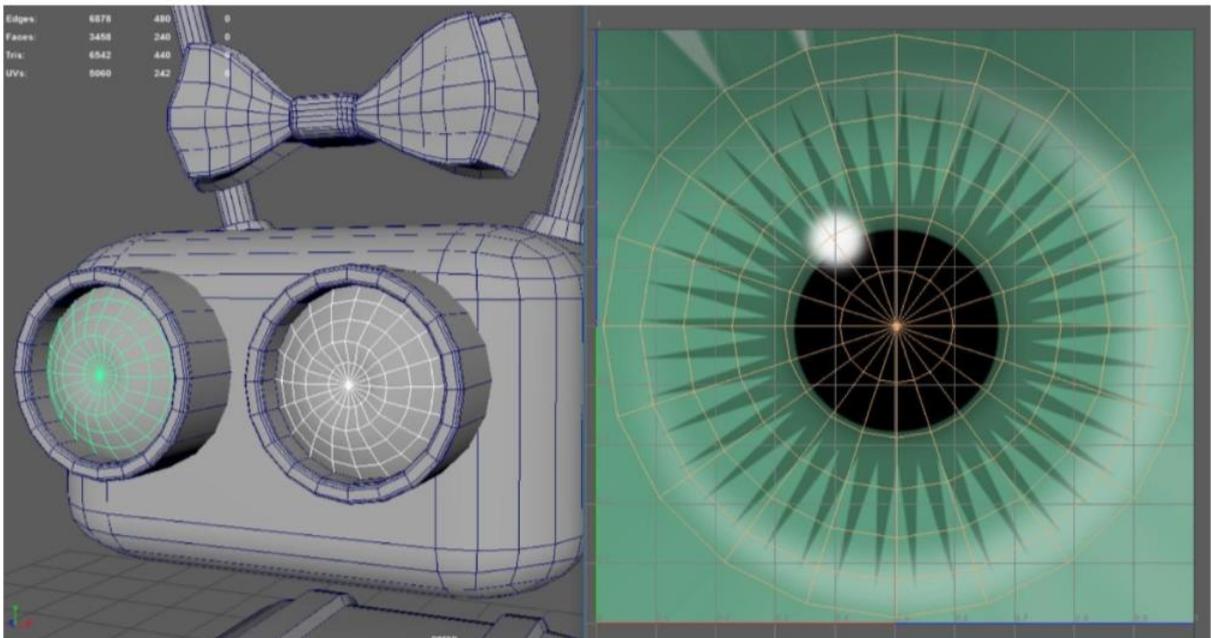
Gambar 4-8 menunjukkan avatar robot yang terdiri dari banyak bagian, telah digabungkan menjadi satu mesh, dan UV-nya dibagi dalam satu ruang, semuanya dibuka dan siap diberi tekstur.

Ada satu area pada model ini yang ingin saya pertahankan resolusi yang lebih tinggi: detail pada mata. Modelnya sendiri datar; namun, saya memberikannya peta tekstur satu mata yang dibagikan di kedua mata jaring datar dan melingkar. Detail pada gambar 2D datar menipu pemirsa untuk berpikir bahwa mungkin ada kedalaman yang lebih dari yang sebenarnya.

Jika saya memasukkannya ke dalam atlas tekstur, saya perlu memperbesar ukuran tekstur dan membuat sisa UV jauh lebih kecil karena detail pada pupil dan sorotan pada mata lebih penting, membutuhkan lebih banyak ruang UV.

Sebaliknya, UV dari eye mesh mengambil seluruh ruang UV di kuadran untuk tekstur mata. Submesh menunjukkan semua detail yang dibutuhkan mata. Submesh yang sama kemudian diduplikasi ke soket lain karena tidak perlu detail unik untuk membedakan antara mata. Gambar 4-9 menunjukkan area UV yang dibagi pada tekstur kecil untuk mata.

Untuk gaya seni yang lebih realistis, Anda masih perlu menjaga jumlah poli di sisi bawah; namun, Anda dapat menjaga kualitas model tetap tinggi dengan menggunakan shader dan rendering berbasis fisik. Model robot ini menggunakan rendering berbasis fisik (PBR) untuk memiliki tampilan yang realistis, seperti yang diilustrasikan pada Gambar 4-10. PBR menggunakan model pencahayaan realistis dan nilai permukaan yang mewakili material nyata. Mari kita bahas beberapa tekstur PBR yang saya gunakan pada model robot sebagai contoh. Semoga ini membantu Anda memahami bagaimana PBR akan bekerja pada model untuk pengalaman VR Anda.

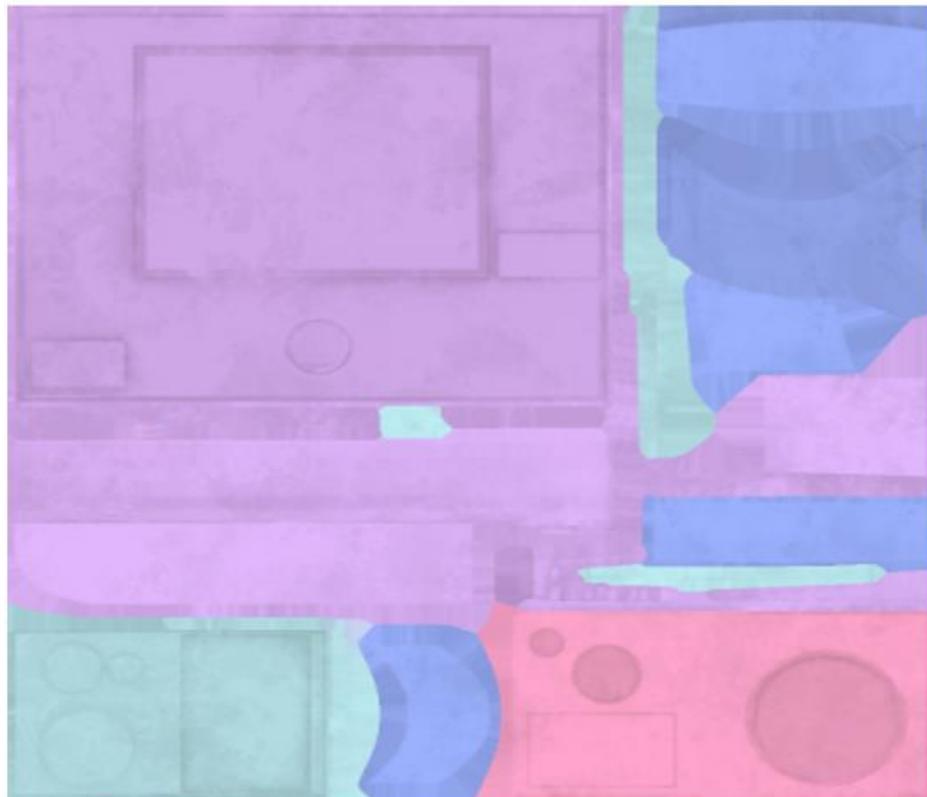


Gambar 4-9. Mata model robot memiliki UV yang sama, diduplikasi sebelum digabungkan menjadi satu mata jaring

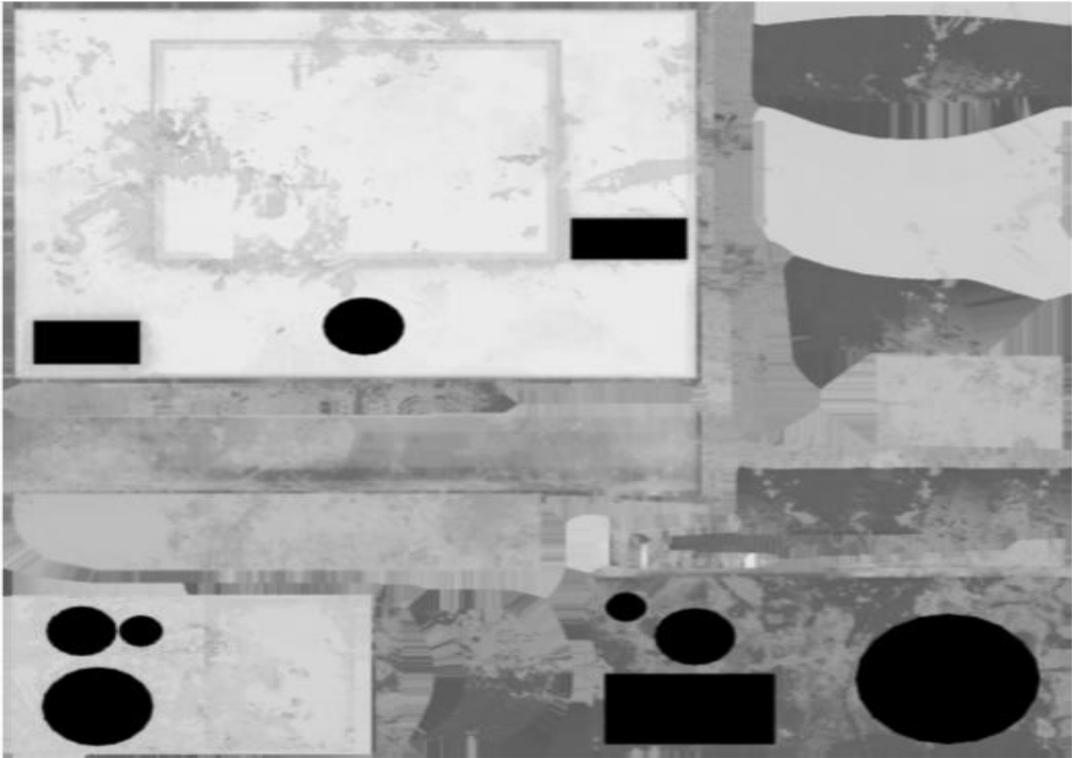


Gambar 4-10. Lihatlah robot dengan semua bahan PBR-nya

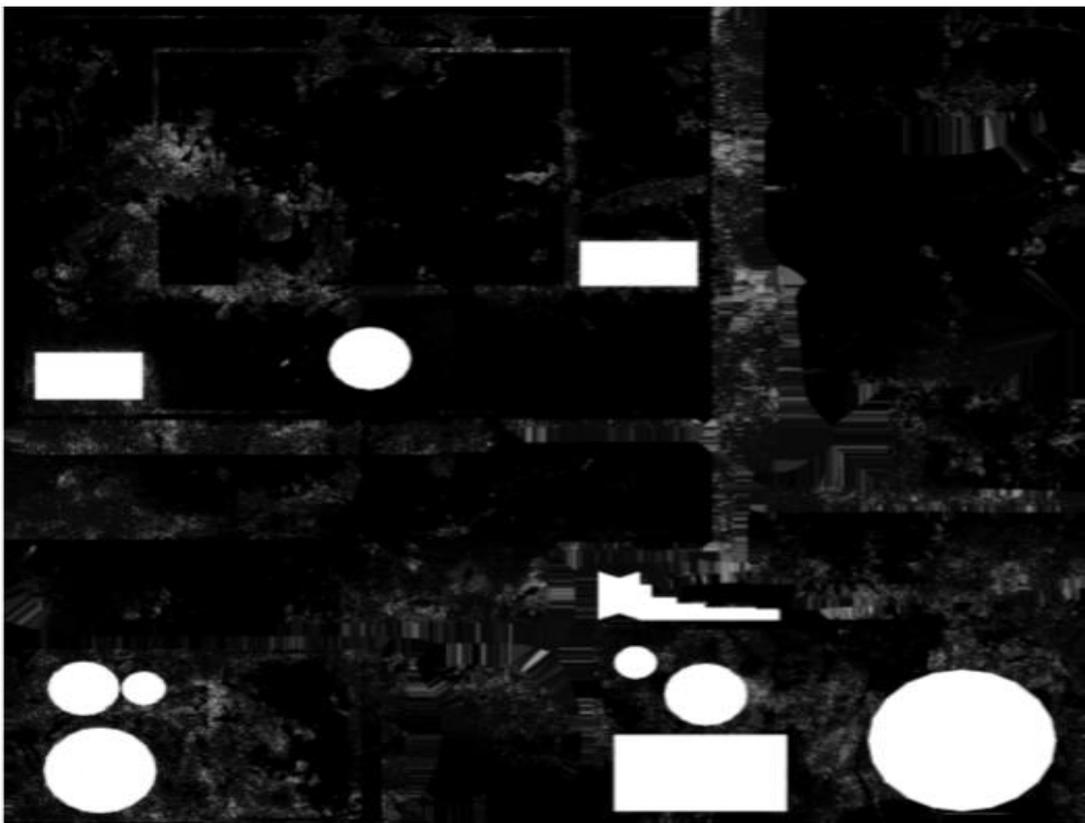
Ingat model konsol game yang kita lihat sebelumnya di bab ini? Gambar 4-11 sampai 4-13 menunjukkan atlas tekstur yang digunakan untuk set tersebut; perhatikan tekstur individu yang digunakan untuk bahan PBR-nya.



Gambar 4-11. Peta warna di mana tekstur mendefinisikan warna yang direpresentasikan pada model

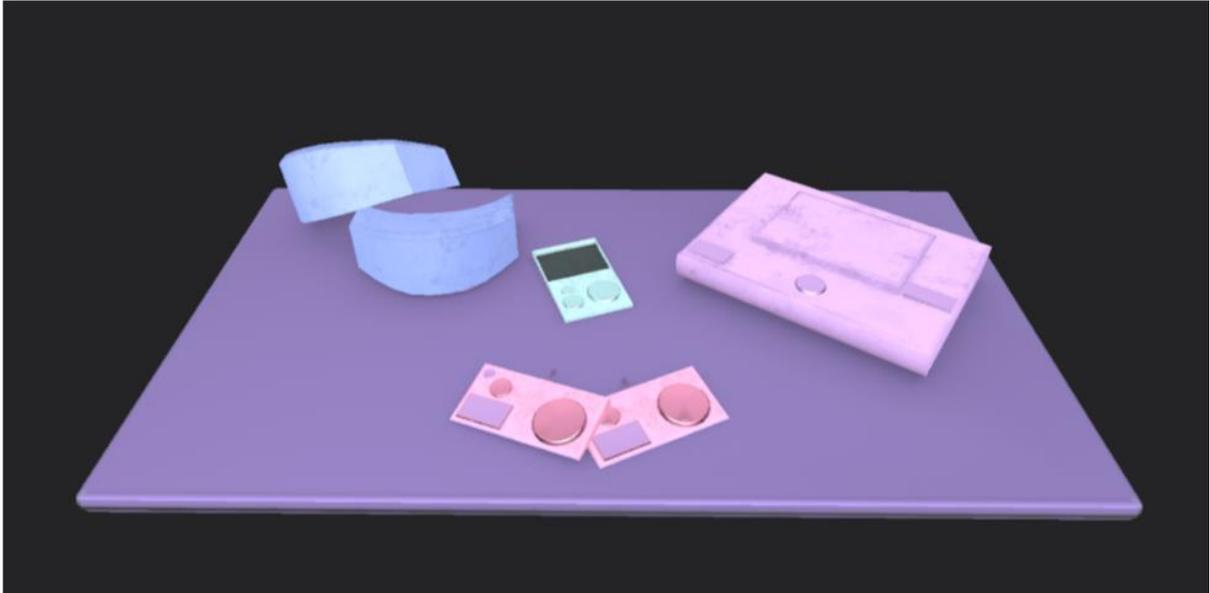


Gambar 4-12. Peta kekasaran di mana tekstur mendefinisikan permukaan model, mulai dari halus hingga kasar

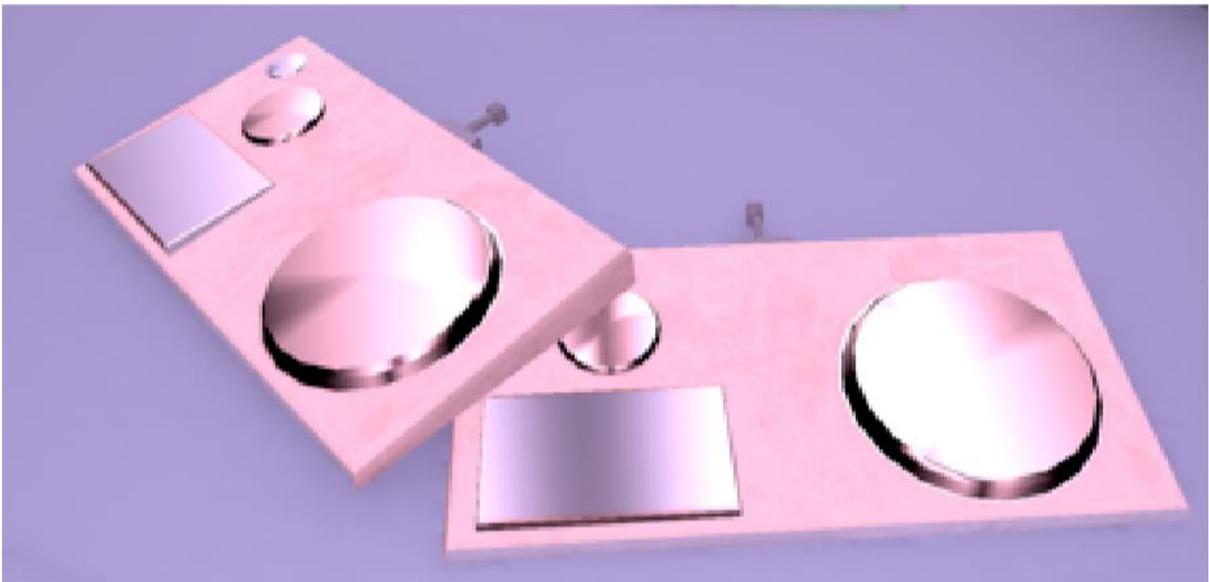


Gambar 4-13. Peta logam di mana tekstur menentukan apakah suatu permukaan adalah logam

Gambar 4-14 hingga 4-17 menunjukkan tampilan akhir model 3D dalam program yang mereka lukis, Pelukis Substansi *Allegorithmic*, dan menunjukkan tampilannya di VR dalam aplikasi VR sosial, *High Fidelity*.



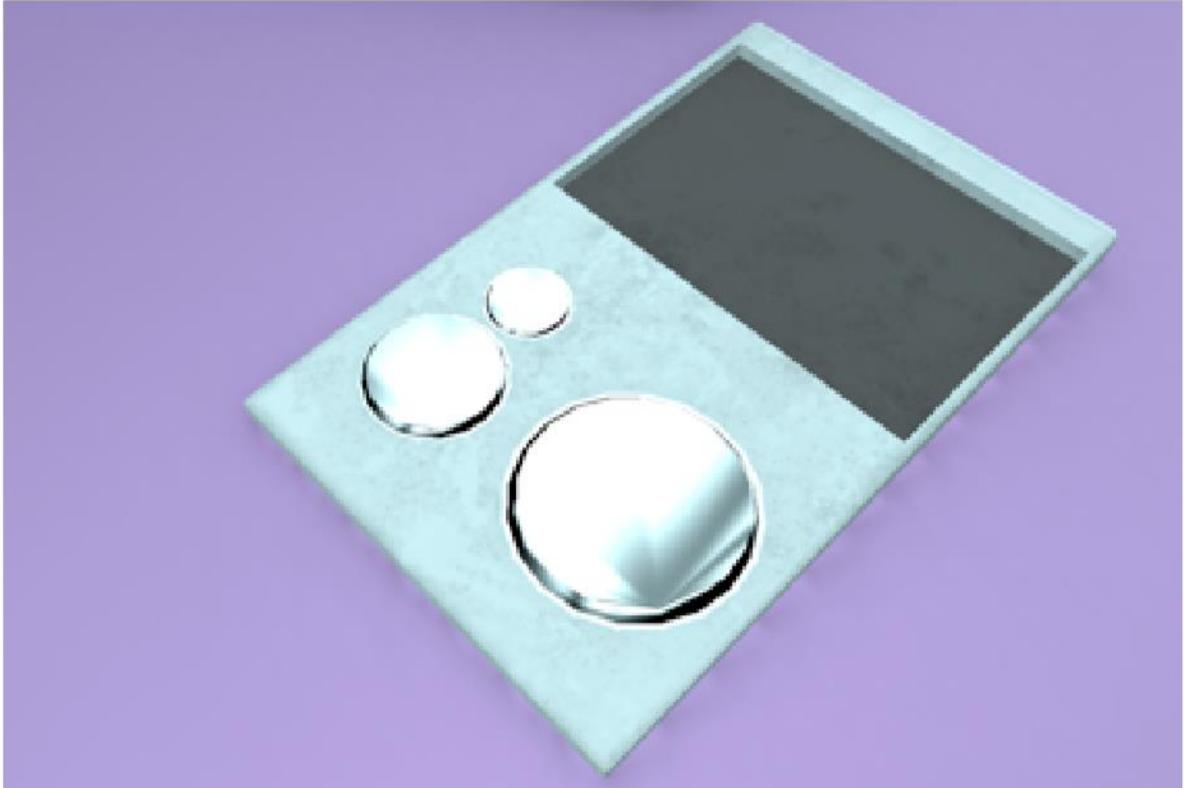
Gambar 4-14. Tampilan sistem permainan, digabungkan menjadi satu jaringan menggunakan satu bahan yang menggunakan tekstur PBR untuk menentukan warna dan permukaan



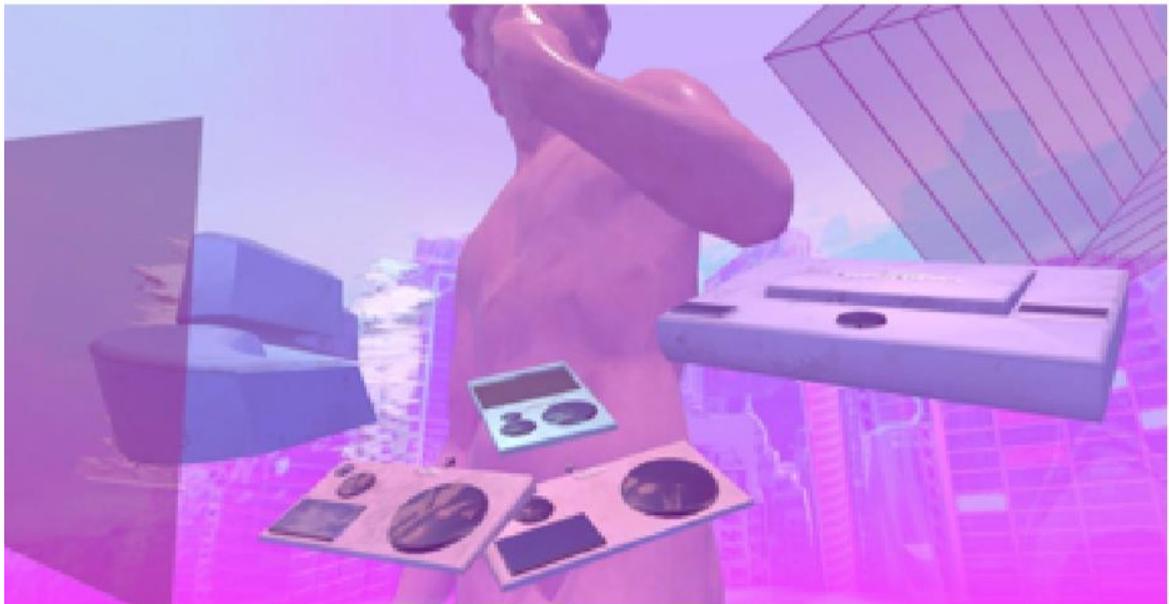
Gambar 4-15. Kontroler ini menunjukkan kontras tinggi yang digunakan tekstur untuk menentukan permukaan logam dan nonlogam

Ada jenis peta tekstur lain seperti peta normal, bump, dan ambient occlusion. Mereka masing-masing berperan dalam menentukan tampilan model apakah itu memalsukan kedalaman atau menciptakan bayangan. Luangkan waktu untuk bereksperimen dengan peta tekstur ini dan temukan apa yang dibutuhkan model Anda. Sekarang setelah Anda melihat

bagaimana Anda dapat membuat atlas tekstur, kita selanjutnya berbicara tentang mengapa penting untuk membuatnya saat kita memeriksa panggilan undian.



Gambar 4-16. Sistem permainan ini memiliki lebih banyak informasi kekasaran pada bagian nonlogam, menunjukkan kotoran dan kotoran



Gambar 4-17. Ini adalah versi terakhir mereka yang terletak di galeri seni virtual berskala besar tempat para model melayang di langit

4.4 MENGGAMBAR PANGGILAN

Panggilan menggambar adalah fungsi yang menghasilkan rendering objek di layar Anda. CPU bekerja dengan unit pemrosesan grafis (GPU) untuk menggambar setiap objek menggunakan informasi tentang mesh, teksturnya, shader, dan sebagainya. Anda harus selalu berusaha untuk mendapatkan jumlah panggilan undian sesedikit mungkin karena memiliki terlalu banyak akan menyebabkan pengurangan frekuensi gambar. Untuk menurunkan jumlah panggilan undian yang Anda miliki, ikuti panduan berikut:

- Gabungkan semua submeshes model Anda menjadi satu mesh gabungan.
- Buat atlas tekstur untuk semua UV dalam model.
- Berikan mesh Anda jumlah bahan sesedikit mungkin yang menggunakan semua tekstur yang dibutuhkan model atau model.

Pikirkan salah satu pengalaman VR favorit Anda dan gambarkan semua model 3D yang membentuk Scene tersebut. Masing-masing dari mereka berkontribusi untuk menarik jumlah panggilan dengan satu atau lain cara. Mereka selalu menambahkan. Jika konteks ini dialami dalam VR sosial, pertimbangkan juga berapa banyak orang yang akan mengalami rendering segala sesuatu di Scene Anda.

Saat kita mendekati akhir bab ini, saya ingin menyatakan kembali bahwa penting untuk menjaga pengoptimalan sebagai prioritas tinggi dalam keseluruhan proses desain—dari awal model hingga tekstur yang telah selesai. Jaga agar angka dan ukuran tetap kecil tanpa harus mengorbankan semua yang Anda inginkan untuk konten VR Anda.

4.5 MENGGUNAKAN ALAT VR UNTUK MEMBUAT SENI 3D

Pada saat ini, Anda mungkin bertanya-tanya mengapa sejauh ini bab ini difokuskan pada karya seni 3D yang dibuat pada layar 2D jika kita berbicara tentang VR di sini. Meskipun kami melihat banyak opsi untuk pembuatan karya seni muncul dengan banyak alat yang tersedia (seperti Tiltbush, Medium, Unbound, Quill, dan Google Blocks), manipulasi tradisional aset 3D akan dilakukan pada program yang ditujukan untuk pemirsa 2D.

Tidak jauh berbeda dalam hal memiliki model yang perlu dioptimalkan. Saat ini, tidak mengherankan untuk mengekspor sejumlah besar konten dari program-program ini. Perasaan magis menciptakan seni dalam ruang 3D di sekitar Anda berasal dari konten yang keluar seperti yang diharapkan. Ini berarti bahwa banyak geometri dibuat dengan loop tepi yang cukup untuk memberi Anda kurva yang diharapkan. Beberapa bahan juga dapat digunakan untuk membuat karya tersebut sangat berwarna dan cerah. Apa yang Anda buat dengan program ini kemungkinan besar perlu dioptimalkan jika ditambahkan ke ruang dengan lebih banyak konten yang perlu digambar di layar Anda.

Apa pun program yang Anda gunakan, bahkan jika Anda menemukan alat yang akan membantu mengoptimalkan aset yang digunakan untuk pengalaman imersif Anda, kemungkinan besar pembuat konten dan desainer harus membuat pilihan untuk memastikan ukuran, jumlah, dan kualitas dapat diterima untuk pengalaman tersebut. Keseimbangan yang tepat akan selalu dibutuhkan, apa pun media yang digunakan untuk membuat konten ini.

4.6 MEMPEROLEH MODEL 3D VERSUS MEMBUATNYA DARI AWAL

Berhati-hatilah saat membeli model dari toko online. Pertimbangkan berapa lama model itu dibuat. Apakah menurut Anda itu dibuat dengan VR dalam pikiran? Apakah Anda perlu membersihkan model dan mengoptimalkannya untuk penggunaan Anda? Apakah waktu yang mungkin perlu Anda habiskan untuk itu lebih murah daripada waktu Anda membuatnya dari awal? Membeli model 3D bisa cepat dan mudah, tetapi dapat memengaruhi kinerja Anda di kemudian hari dan membutuhkan banyak waktu untuk memodifikasinya agar berkinerja baik.

Berikut adalah daftar apa yang harus dicari dalam daftar item dan pertanyaan apa yang harus Anda tanyakan saat mengunduh model 3D dari tempat-tempat seperti Poly, Turbosquid, CGTrader, dan sebagainya (jika Anda tidak melihat informasi apa pun yang tercantum, harap sangat berhati-hati dan rencana untuk ketidaknyamanan):

- Jumlah poli
- Apakah ini jumlah segitiga yang sesuai?
- Jika modelnya bagus tetapi poli tinggi, berapa banyak waktu yang akan Anda habiskan untuk mengurangi jumlah poli dan membersihkan geometri untuk membuat aset siap-VR?
- Tekstur Map.
- Apakah model bertekstur dengan cara yang dioptimalkan, menggunakan atlas tekstur?
- Jika ada beberapa peta tekstur terpisah, apakah menurut Anda waktu yang diperlukan untuk mengoptimalkannya dapat diterima?
- Apakah file tekstur dalam format yang didukung oleh mesin yang akan merendernya?
- Berapa ukuran file tekstur? Waspada terhadap tekstur yang lebih besar dari 2.048, terutama jika tekstur yang besar adalah untuk model yang akan berskala kecil. Juga, cari tekstur kecil jika yang Anda inginkan adalah resolusi yang lebih tinggi pada beberapa model.
- Format file.
- Apakah Anda membeli file yang dapat digunakan untuk bekerja?
- Apakah program Anda mendukung pembukaan dan pengeditan model?

Selalu uji penampilan model Anda. Masukkan ke mesin pilihan Anda dan lihat sendiri di VR atau AR. Anda akan terkejut dengan bagaimana skala yang berbeda terasa ketika Anda tenggelam olehnya.

4.7 RINGKASAN

Dalam bab ini, Anda melihat pendekatan dan proses berpikir yang berbeda untuk dipertimbangkan dalam berbagai bidang yang terlibat dalam menciptakan seni 3D. Perlu waktu dan latihan untuk mempelajari cara mengoptimalkan seni 3D, jadi pastikan pengoptimalan selalu menjadi prioritas utama selama seluruh proses desain. Anda mungkin seorang seniman yang baru mengenal VR atau AR. Anda mungkin seorang pengembang yang mempelajari bidang pekerjaan orang lain. Anda mungkin seorang produser yang ingin tahu

tentang alur artis. Saya senang Anda sampai sejauh ini untuk belajar tentang pentingnya pengoptimalan karena ini merupakan tantangan besar dalam hal mengembangkan aset untuk VR dan AR. Setiap orang yang bekerja pada pengalaman yang mendalam harus tahu tentang pekerjaan yang menantang dalam pembuatan aset.

Dengan teknologi yang berubah dengan cepat, beberapa teknik atau program yang Anda lihat dalam bab ini mungkin tidak relevan dalam waktu dekat, jadi penting untuk mengingat alasan di balik metode ini. Seperti disebutkan sebelumnya, penting untuk membuat orang merasa nyaman dengan pengalaman Anda. Pastikan untuk berhati-hati dan jaga agar frame rate tetap tinggi dengan seni yang dioptimalkan!

BAGIAN III

HARDWARE, SLAM, TRACKING

Pengalaman pengguna augmented reality (AR) menarik dan sering kali terasa seperti sulap. Namun, dalam kondisi terbaiknya, Anda seharusnya tidak menyadarinya sama sekali. Semakin baik perancang sistem AR melakukan pekerjaan mereka, semakin sedikit Anda memperhatikan pekerjaan mereka, dan Anda dapat fokus pada konten dan interaksi yang membantu Anda mencapai apa yang ingin Anda lakukan di AR sejak awal. Memecahkan masalah teknis untuk mencapai ini adalah masalah yang sangat sulit; kemajuan besar telah dibuat, tetapi masih banyak masalah yang harus dipecahkan. Bagian ini bertujuan untuk membantu menjelaskan bagaimana segala sesuatu di bawah kap bekerja, bagaimana kita sampai di sini, dan bagaimana membuat pilihan ke mana harus menginvestasikan energi Anda ke depan. Mudah-mudahan, bab ini membantu menjelaskan mengapa, ketika sistem tampaknya merusak Anda, apa yang terjadi, dan memberi Anda beberapa petunjuk tentang cara merencangkannya. Selama beberapa tahun ke depan, membangun aplikasi AR akan sangat bergantung pada bagaimana pengembang AR membangun produk yang bekerja dalam batasan sistem sementara pembuat sistem bekerja untuk menghilangkan batasan tersebut. Kami membahas teknologi inti yang menopang semua sistem AR, lokalisasi dan pemetaan simultan (SLAM), dan mengapa itu adalah istilah luas yang tidak terlalu membantu menjelaskan apa pun! Kami membahas komponen yang masuk ke sistem SLAM dan batasannya, ditambah kami melihat bagaimana beberapa batasan ini (misalnya, peta SLAM yang lebih besar dari satu perangkat dapat menangani) diselesaikan melalui cloud AR untuk mengaktifkan pengalaman seperti konten bersama, konten persisten, dan pemahaman semantik tentang dunia sementara konten virtual dapat berinteraksi secara fisik dengan dunia fisik. Kami menyentuh beberapa perbedaan antara ARKit, ARCore, dan sistem berbasis pemetaan spasial seperti 6D.ai, Magic Leap, dan Hololens.

Pengumuman Apple tentang ARKit di WWDC 2017 memiliki dampak besar pada ekosistem AR. Pengembang menemukan bahwa untuk pertama kalinya kit pengembangan perangkat lunak (SDK) AR yang kuat dan tersedia secara luas “hanya berfungsi” untuk aplikasi mereka. Tidak perlu mengutak-atik penanda atau inisialisasi atau kamera kedalaman atau alat kreasi eksklusif. Tidak mengherankan, ini telah menyebabkan ledakan demonstrasi. Namun, sebagian besar pengembang tidak tahu cara kerja ARKit atau mengapa ARKit bekerja lebih baik daripada SDK lain. Melihat “di balik kap mesin” ARKit akan membantu kita memahami batasan ARKit saat ini, apa yang masih dibutuhkan dan mengapa, serta membantu memprediksi kapan kemampuan serupa akan tersedia di Android dan tampilan HMD yang dipasang di kepala; baik *virtual reality* (VR) atau AR. Saya telah melihat orang menyebut ARKit sebagai SLAM, atau menggunakan istilah SLAM untuk merujuk pada tracking. Untuk klarifikasi, perlakukan SLAM sebagai istilah yang cukup luas; seperti, misalnya "multimedia." Tracking itu sendiri adalah istilah yang lebih umum, sedangkan odometri lebih spesifik, tetapi dalam

praktiknya mereka cukup dekat dengan AR. Ini bisa membingungkan. Ada banyak cara untuk melakukan SLAM, dan tracking hanyalah salah satu komponen dari sistem SLAM yang komprehensif. ARKit diluncurkan sebagai "lite" atau sistem SLAM sederhana. Pada tulisan ini, sistem SLAM Tango atau Hololens memiliki lebih banyak fitur di luar odometri, seperti pemetaan yang lebih canggih, rekonstruksi 3D, dan dukungan untuk sensor kedalaman.

Istilah "*AR cloud*" benar-benar populer sejak saya dan mitra Super Ventures saya, Ori, menulis dua blog tentang topik tersebut. Kami telah melihatnya diterapkan pada sejumlah besar ide "berawan" yang memiliki beberapa sudut AR untuk mereka, tetapi bagi saya itu secara khusus mengacu pada infrastruktur untuk memungkinkan sistem AR terhubung satu sama lain dan ke dunia yang lebih besar secara umum, bukan ke konten.

BAB 5

MEMBUAT *AUGMENTED REALITY* (AR) BEKERJA

Siapakah Kita?

Di 6D.ai, kami berpikir sedikit berbeda dengan orang lain tentang AR. Kami memecahkan masalah teknis yang paling sulit dan memaparkan solusi melalui API pengembang untuk pelanggan dengan masalah AR yang paling menantang. Kami independen dan lintas platform, dan kami menjual penggunaan API kami, bukan iklan berdasarkan data pelanggan kami. Kami percaya bahwa ketekunan adalah dasar, dan Anda tidak dapat memiliki ketekunan tanpa memperlakukan privasi dengan serius. Dan untuk memperlakukan privasi dengan serius, itu berarti bahwa informasi pengenalan pribadi (PII) tidak dapat meninggalkan perangkat (kecuali diizinkan secara eksplisit oleh pengguna). Ini menciptakan masalah teknis yang jauh lebih sulit untuk dipecahkan karena ini berarti membangun dan mencari peta SLAM besar di perangkat, dan secara *real time*. Ini secara teknis mudah dilakukan dengan peta kecil dan jangkar, tetapi sangat, sangat sulit dilakukan dengan peta besar. Dan ketika saya mengatakan kecil, maksud saya setengah ruangan, dan besar berarti lebih besar dari rumah besar.

Untungnya, kami memiliki grup penelitian AR teratas dari Oxford Active Vision Lab di belakang 6D.ai, dan kami membangun sistem kami pada rekonstruksi 3D generasi berikutnya dan algoritma relocalizer dan jaringan saraf, mengambil keuntungan dari beberapa penelitian yang belum dipublikasikan. Tujuan dari semua ini adalah untuk mendapatkan multipemain dan AR yang persisten sedekat mungkin dengan pengalaman pengguna (UX) yang "hanya berfungsi", di mana tidak ada yang perlu dijelaskan, dan intuisi pengguna akhir tentang bagaimana konten AR harus berperilaku benar. Inilah yang istimewa tentang bagaimana 6D.ai melakukan AR:

- Kami melakukan semua pemrosesan di perangkat dan secara *real time*. Kami menggunakan cloud untuk penyimpanan peta persisten dan beberapa penggabungan dan pembersihan data *offline*.
- Peta dibuat di latar belakang saat aplikasi sedang berjalan. Pembaruan dari semua pengguna digabungkan menjadi satu peta, sangat meningkatkan cakupan ruang.
- Jangkar dan peta tidak memiliki PII dan disimpan secara permanen di cloud kami. Setiap kali aplikasi yang didukung 6D.ai menggunakan ruang fisik itu, jangkar tumbuh dan meningkatkan cakupan ruang itu. Ini meminimalkan dan akhirnya menghilangkan kebutuhan untuk melakukan pra-pemindaian ruang.
- Peta tersedia untuk semua aplikasi. Setiap pengguna mendapat manfaat dari setiap pengguna lain dari aplikasi 6D.ai.
- Data peta kami tidak dapat direkayasa balik menjadi gambar visual yang dapat dibaca manusia.

- Jangkar kami sangat diuntungkan dari penyimpanan dan penggabungan cloud, tetapi tidak ada ketergantungan pada cloud agar UX berfungsi. Tidak seperti sistem Google, kami dapat bekerja secara *offline*, atau dalam lingkungan peer-to-peer, atau lingkungan pribadi/aman (atau China).

Ini adalah dunia yang sangat kecil. Tidak banyak orang yang bisa membangun sistem ini dengan baik.

5.1 SEJARAH SINGKAT AR

Berikut adalah ringkasan dari pemain kunci yang membawa AR ke kualitas konsumen:

- Odometri inersia visual ditemukan di Intersense pada awal 2000-an oleh Leonid Naimark → Dekko → Samsung → FB and Magic Leap dan Tesla
- FlyBy VIO → Tango and Apple
- Oxford Active Vision Lab → George Klein (PTAM) → Microsoft
- Microsoft (David Nister) → Tesla
- Oxford → Gerhard Reitmeir → Vuforia
- Oxford → Gabe Sibley → Zook
- Oxford + Cambridge + Imperial College → Kinect → Oculus dan ML (Richard Newcomb, David Molyneux)
- Vuforia → Eitan Pilipski → Snap
- FlyBy/Vuforia → Daqri

Salah satu aspek yang menarik dan kurang dihargai tentang bagaimana sistem AR berkualitas tinggi dibangun adalah bahwa hanya ada segelintir orang di dunia yang dapat membangunnya. Karier yang saling berhubungan dari para insinyur ini telah menghasilkan sistem terbaik yang menyatu pada odometri inersia visual bermata (VIO) sebagai "solusi" untuk tracking seluler. Tidak ada pendekatan lain yang memberikan UX (hari ini).

VIO pertama kali diterapkan di pemasok militer/industri Intersense yang berbasis di Boston, Massachusetts pada awal 2000-an. Salah satu penemu, Leonid Naimark adalah kepala ilmuwan di startup saya, Dekko, pada tahun 2011. Setelah Dekko membuktikan bahwa VIO tidak dapat memberikan UX konsumen pada iPad 2 karena keterbatasan sensor, Leonid kembali ke kontrak militer, tetapi CTO Dekko, Pierre Georgel, sekarang menjadi insinyur senior di tim Google Daydream. Sekitar waktu yang sama, Ogmento didirikan oleh mitra Super Ventures saya, Ori Inbar. Ogmento menjadi FlyBy dan tim di sana berhasil membangun sistem VIO di iOS menggunakan kamera mata ikan tambahan. Basis kode ini dilisensikan ke Google, yang berkembang menjadi sistem VIO untuk Tango. Apple kemudian membeli FlyBy, dan basis kode yang sama adalah inti dari ARKit VIO. CTO FlyBy, Chris Broaddus, kemudian membangun pelacak untuk Daqri, yang sekarang berada di perusahaan robotika otonom, dengan mantan kepala ilmuwan Zook, Gabe Sibley. Gabe melakukan pekerjaan pasca-doktoralnya di Oxford (bersama dengan salah satu pendiri saya di 6D.ai, yang saat ini memimpin Active Vision Lab). Sistem SLAM seluler (PTAM) pertama dikembangkan sekitar tahun 2007 di Oxford Active Computing Lab oleh Georg Klein, yang kemudian membangun sistem VIO untuk Hololens,

bersama dengan Christopher Mei (lulusan Oxford Active Vision lainnya) dan David Nister, yang keluar untuk membangun sistem otonomi di Tesla. George memperoleh gelar PhD di Cambridge, dari mana rekannya Gerhard Reitmayr melanjutkan ke Vuforia untuk bekerja pada pengembangan sistem SLAM dan VIO Vuforia. Pengembangan Vuforia dipimpin oleh Daniel Wagner, yang kemudian mengambil alih dari Chris Broaddus (ex-FlyBy) sebagai kepala ilmuwan di Daqri. Manajer teknik Vuforia, Eitan Pilipski, sekarang memimpin rekayasa perangkat lunak AR di Snap, bekerja dengan Qi Pan, yang belajar di Cambridge bersama Gerhard dan Georg, dan kemudian pergi ke Vuforia. Qi sekarang memimpin tim AR di Snap di London bersama Ed Rosten (lulusan Cambridge lainnya, yang mengembangkan detektor fitur FAST yang digunakan di sebagian besar sistem SLAM).

Anggota kunci dari tim peneliti di Oxford, Cambridge (misalnya, David Molyneaux) dan Imperial College (lab Profesor Andy Davison, tempat Richard Newcombe, Hauke Strasdat, dan lainnya belajar) mengembangkan lebih lanjut D-SLAM dan memperluas sistem tracking Kinect, dan sekarang memimpin tim tracking di Oculus dan Magic Leap. Metaio juga merupakan inovator kunci awal di sekitar SLAM (berdasarkan keahlian dari TU Munich, tempat Pierre Georgel belajar), banyak insinyur sekarang berada di Apple, tetapi pemimpin R&D mereka, Selim Benhimane, belajar bersama Pierre dan kemudian mengembangkan SLAM untuk Intel RealSense, dan sekarang di Apple.

Menariknya, saya tidak mengetahui adanya startup AR saat ini yang bekerja di domain tracking AR yang dipimpin oleh talenta teknik dari kumpulan talenta kecil ini. Pendiri dari latar belakang Robotika atau jenis visi komputer lainnya belum dapat menunjukkan sistem yang bekerja dengan kuat di berbagai lingkungan.

5.2 BAGAIMANA DAN MENGAPA MEMILIH PLATFORM AR

Ada banyak platform yang bisa dipilih di AR, mulai dari mobile AR hingga PCAR. Berikut adalah beberapa pertimbangan teknis yang perlu diingat ketika mulai mengembangkan AR.

Saya Seorang Developer, Platform Apa yang Harus Saya Gunakan dan Mengapa?

Anda dapat mulai mengembangkan ide AR Anda di ponsel apa pun yang memiliki akses ke ARKit. Ini berfungsi dan Anda mungkin sudah memiliki telepon yang mendukungnya. Pelajari perbedaan besar dalam merancang dan mengembangkan aplikasi yang berjalan di dunia nyata di mana Anda tidak mengontrol Scene versus aplikasi smartphone dan VR, di mana Anda mengontrol setiap piksel.

Kemudian, pindah ke platform seperti Magic Leap, 6D.ai, atau Hololens yang dapat memetakan dunia secara spasial. Sekarang pelajari apa yang terjadi ketika konten Anda dapat berinteraksi dengan struktur 3D dari pemandangan yang tidak terkontrol.

Pergi dari satu ke yang lain adalah kurva belajar yang sangat curam. Faktanya, lebih curam daripada dari web ke seluler atau dari seluler ke VR. Anda harus benar-benar memikirkan kembali cara kerja aplikasi dan UX atau kasus penggunaan apa yang masuk akal. Saya melihat banyak demonstrasi ARKit yang saya lihat lima tahun lalu dibuat di Vuforia, dan empat tahun sebelumnya di Layar. Pengembang mempelajari kembali pelajaran yang sama, tetapi pada skala yang jauh lebih besar. Saya telah melihat contoh hampir semua jenis aplikasi

AR selama bertahun-tahun, dan dengan senang hati memberikan umpan balik dan dukungan. Hanya menjangkau.

Saya akan mendorong pengembang untuk tidak takut membangun aplikasi baru. Aplikasi kentut adalah hit pertama di smartphone — juga sangat menantang untuk menemukan kasus penggunaan yang memberikan utilitas melalui AR pada perangkat keras faktor bentuk tembus pandang.

5.3 KINERJA ADALAH STATISTIK

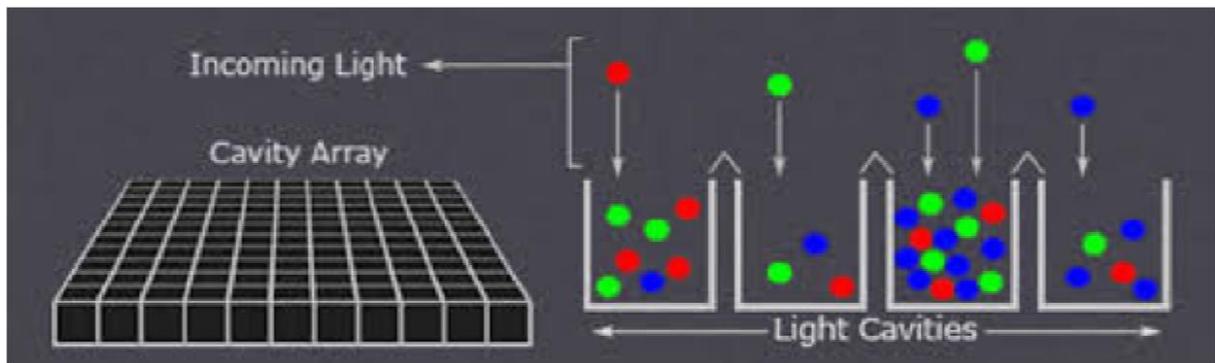
Saat pertama kali bekerja dengan AR atau sebagian besar sistem visi komputer apa pun, itu bisa membuat frustrasi karena kadang-kadang itu akan berfungsi dengan baik di satu tempat, tetapi di tempat lain itu akan bekerja dengan sangat buruk. Sistem AR tidak pernah "berfungsi" atau "tidak berfungsi." Itu selalu menjadi pertanyaan apakah segala sesuatunya bekerja dengan cukup baik dalam berbagai situasi yang cukup luas. Menjadi "lebih baik" pada akhirnya adalah masalah mendorong statistik lebih jauh sesuai keinginan Anda.

Karena alasan ini, jangan pernah percaya demonstrasi aplikasi AR, terutama jika itu terbukti luar biasa di YouTube. Ada kesenjangan besar antara sesuatu yang bekerja dengan sangat baik di lingkungan yang terkontrol atau sedikit bertahap dan yang hampir tidak berfungsi sama sekali untuk penggunaan biasa. Situasi ini tidak ada untuk demonstrasi aplikasi smartphone atau VR. Mari kita rangkum ini:

- Selalu mendemonstrasikan atau menguji sistem di dunia nyata. Ada kesenjangan besar antara Scene terkontrol dan tidak terkontrol. Jangan pernah percaya video demonstrasi.
- Apa yang dimaksud dengan bekerja dengan baik?
 - Tidak ada gerakan pengguna yang terdeteksi untuk inisialisasi
 - Konvergensi instan
 - Skala metrik <2% kesalahan
 - Tidak ada kegelisahan
 - Tidak ada penyimpangan
 - Daya rendah
 - Biaya BOM rendah
 - Jarak ratusan meter dengan drift <1% (sebelum penutupan loop)
 - Penutupan loop instan
 - Penutupan loop dari berbagai sudut
 - Scene berfitur rendah (misalnya, langit, dinding putih)
 - Scene dengan pencahayaan bervariasi/cahaya rendah
 - Scene berulang atau reflektif

Berikut adalah deskripsi teknis spesifik mengapa statistik akhirnya menentukan seberapa baik sistem bekerja. Gambar 5-1 menggambarkan kotak yang mewakili sensor gambar digital di kamera Anda. Setiap kotak adalah piksel. Agar tracking stabil, setiap piksel harus cocok dengan titik yang sesuai di dunia nyata (dengan asumsi perangkat diam sempurna). Namun, gambar kedua menunjukkan bahwa foton tidak begitu akomodatif, dan berbagai intensitas cahaya

jatuh ke mana pun mereka inginkan, dan setiap piksel hanyalah total foton yang menabraknya. Setiap perubahan cahaya dalam pemandangan (awan melewati matahari, kedipan lampu neon, dll.) mengubah susunan foton yang mengenai sensor, dan sekarang sensor memiliki piksel berbeda yang sesuai dengan dunia nyata titik. Sejauh menyangkut sistem tracking visual,



Gambar 5-1. Segala sesuatu yang berhubungan dengan kinerja visi komputer adalah masalah statistik; ini adalah dunia nyata yang bukan biner

Inilah alasan mengapa ketika Anda melihat titik-titik dalam berbagai demonstrasi ARKit, titik-titik itu berkedip-kedip; sistem harus memutuskan poin mana yang "dapat diandalkan" atau tidak dapat diandalkan. Kemudian, perlu melakukan triangulasi dari titik-titik tersebut untuk menghitung pose, merata-ratakan perhitungan untuk mendapatkan perkiraan terbaik tentang pose Anda yang sebenarnya. Jadi, pekerjaan apa pun yang dapat dilakukan untuk memastikan bahwa kesalahan statistik dihilangkan dari proses ini menghasilkan sistem yang lebih kuat. Hal ini memerlukan integrasi dan kalibrasi yang ketat antara tumpukan perangkat keras kamera (beberapa lensa dan pelapis, rana dan spesifikasi sensor gambar, dll.) dan perangkat keras unit pengukuran inersia (IMU) dan algoritma perangkat lunak.

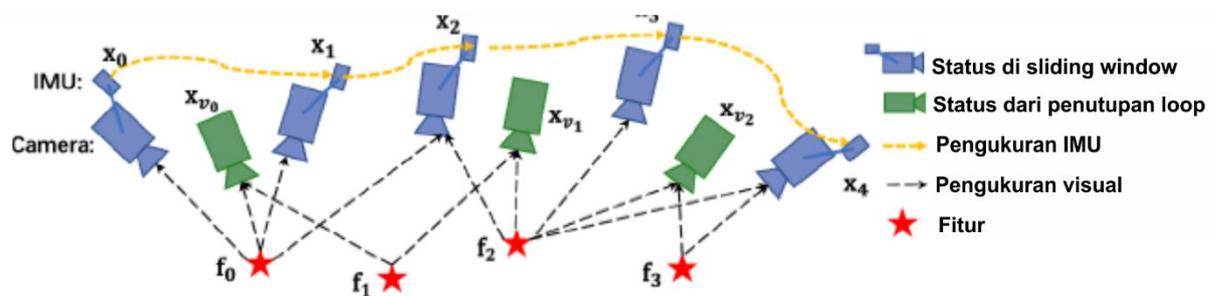
Jika Anda seorang pengembang, Anda harus selalu menguji aplikasi Anda dalam berbagai pemandangan dan kondisi pencahayaan. Jika menurut Anda menangani fragmentasi Android itu buruk, tunggu sampai Anda mencoba menguji semua yang mungkin terjadi di dunia nyata.

5.4 MENGINTEGRASIKAN PERANGKAT KERAS DAN PERANGKAT LUNAK

Menariknya, VIO tidak terlalu sulit untuk bekerja; ada sejumlah algoritma yang diterbitkan dan beberapa implementasi ada. Tapi, sangat sulit untuk membuatnya bekerja dengan baik. Maksud saya, sistem inersia dan optik bertemu hampir seketika ke peta stereoskopik, dan skala metrik dapat ditentukan dengan tingkat akurasi satu digit yang rendah. Implementasi yang kami bangun di Dekko, misalnya, mengharuskan pengguna melakukan gerakan tertentu pada awalnya dan kemudian menggerakkan ponsel maju mundur selama sekitar 30 detik sebelum menyatu. Untuk membangun sistem tracking inersia yang hebat membutuhkan insinyur yang berpengalaman. Sayangnya, hanya ada sekitar 20 insinyur di Bumi dengan keterampilan dan pengalaman yang diperlukan, dan kebanyakan dari mereka

bekerja membangun sistem tracking rudal jelajah, atau sistem navigasi penjelajah Mars, atau aplikasi seluler nonkonsumen lainnya.

Seperti yang diilustrasikan Gambar 5-2, semuanya masih bergantung pada perangkat keras dan perangkat lunak yang bekerja selaras untuk mengurangi kesalahan dengan cara terbaik. Pada intinya, ini berarti IMU yang dapat dimodelkan secara akurat dalam perangkat lunak, akses penuh ke seluruh tumpukan kamera dan spesifikasi terperinci dari setiap komponen dalam tumpukan, dan yang terpenting, IMU dan kamera harus disinkronkan dengan sangat tepat. Sistem perlu mengetahui dengan tepat pembacaan IMU mana yang sesuai dengan awal pengambilan bingkai, dan yang mana hingga akhir. Ini penting untuk menghubungkan kedua sistem, dan sampai saat ini tidak mungkin karena OEM perangkat keras tidak melihat alasan untuk berinvestasi dalam hal ini.



Gambar 5-2. AR membutuhkan integrasi yang erat antara perangkat lunak dan perangkat keras, yang memperlambat solusi pada ponsel

Ini adalah alasan mengapa sistem berbasis iPad 2 Dekko membutuhkan waktu lama untuk menyatu. Ponsel Tango Peanut pertama adalah perangkat pertama yang secara akurat menyinkronkan semua jam, dan merupakan ponsel konsumen pertama yang menawarkan tracking hebat. Saat ini, sistem pada chip dari Qualcomm dan lainnya memiliki hub sensor yang disinkronkan untuk digunakan semua komponen, yang berarti bahwa VIO dapat digunakan di sebagian besar perangkat saat ini, dengan kalibrasi sensor yang sesuai.

Karena ketergantungan yang ketat pada perangkat keras dan perangkat lunak, hampir tidak mungkin bagi pengembang perangkat lunak untuk membangun sistem yang hebat tanpa dukungan mendalam dari OEM untuk membangun perangkat keras yang sesuai. Google berinvestasi banyak untuk mendapatkan beberapa OEM untuk mendukung spesifikasi perangkat keras Tango. Microsoft, Magic Leap, dan lainnya sedang membangun perangkat keras mereka sendiri, dan pada akhirnya itulah mengapa Apple begitu sukses dengan ARKit, karena ia mampu melakukan keduanya.

Kalibrasi Optik

Agar perangkat lunak dapat secara tepat mengkorelasikan apakah piksel pada sensor kamera cocok dengan titik di dunia nyata, sistem kamera perlu dikalibrasi secara akurat. Ada dua jenis kalibrasi:

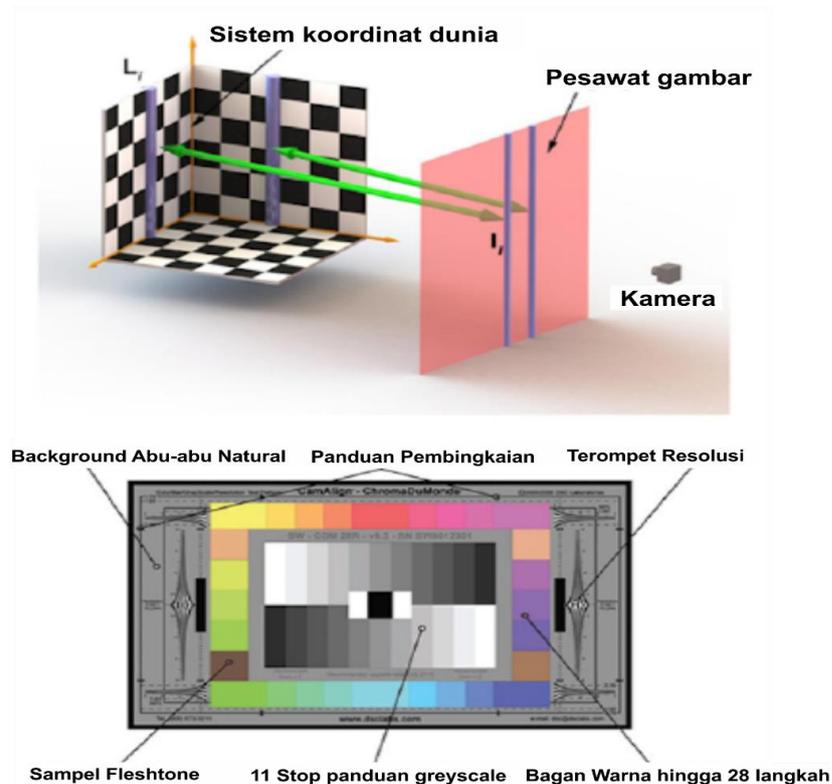
Kalibrasi geometris

Ini menggunakan model lubang jarum kamera untuk mengoreksi Bidang Pandang lensa dan hal-hal seperti efek laras lensa—pada dasarnya, semua gambar melengkung karena bentuk lensa. Sebagian besar pengembang perangkat lunak dapat melakukan langkah ini tanpa input OEM dengan menggunakan papan catur dan spesifikasi dasar kamera umum.

Kalibrasi fotometrik

Ini jauh lebih terlibat dan biasanya memerlukan keterlibatan OEM karena hal itu masuk ke spesifikasi sensor gambar itu sendiri, lapisan apa pun pada lensa internal, dan sebagainya. Kalibrasi ini berkaitan dengan pemetaan warna dan intensitas. Misalnya, kamera yang terpasang teleskop yang memotret bintang yang jauh perlu mengetahui apakah perubahan kecil dalam intensitas cahaya pada piksel pada sensor memang sebuah bintang, atau hanya penyimpangan pada sensor atau lensa. Hasil kalibrasi untuk pelacak AR ini adalah kepastian yang jauh lebih tinggi bahwa piksel pada sensor cocok dengan titik dunia nyata, dan dengan demikian tracking optik lebih kuat dengan lebih sedikit kesalahan.

Pada Gambar 5-3, gambar berbagai foton RGB yang jatuh ke dalam ember piksel pada sensor gambar menggambarkan masalah tersebut. Cahaya dari suatu titik di dunia nyata biasanya jatuh melintasi batas beberapa piksel dan masing-masing piksel tersebut akan rata-rata intensitasnya di semua foton yang mengenainya. Perubahan kecil dalam gerakan pengguna, bayangan dalam pemandangan, atau lampu neon yang berkedip-kedip akan mengubah piksel mana yang paling mewakili titik dunia nyata. Ini adalah kesalahan yang coba dihilangkan oleh semua kalibrasi optik ini sebaik mungkin.



Gambar 5-3. Kalibrasi optik sangat penting bagi sistem untuk mengetahui piksel mana yang sesuai dengan titik dunia nyata

Kalibrasi Inersia

Saat memikirkan IMU (kombinasi akselerometer dan giroskop di perangkat Anda), penting untuk diingat bahwa IMU mengukur akselerasi, bukan jarak atau kecepatan. Kesalahan dalam pembacaan IMU menumpuk dari waktu ke waktu, sangat cepat! Tujuan kalibrasi dan pemodelan adalah untuk memastikan pengukuran jarak (integrasi ganda dari akselerasi) cukup akurat untuk X sepersekian detik. Idealnya, ini adalah periode yang cukup lama untuk dicakup saat kamera kehilangan tracking untuk beberapa bingkai saat pengguna menutupi lensa atau sesuatu yang lain terjadi dalam pemandangan.

Pengukuran jarak menggunakan IMU disebut perhitungan mati. Ini pada dasarnya adalah tebakan, tetapi tebakan dibuat akurat dengan memodelkan bagaimana IMU berperilaku, menemukan semua cara ia mengumpulkan kesalahan, dan kemudian menulis filter untuk mengurangi kesalahan tersebut. Bayangkan jika Anda diminta untuk mengambil langkah dan kemudian menebak seberapa jauh Anda melangkah dalam inci. Satu langkah dan tebakan akan memiliki margin kesalahan yang tinggi. Jika Anda berulang kali mengambil ribuan langkah, mengukur masing-masing dan belajar untuk membiarkan kaki mana yang Anda pijak, penutup lantai, sepatu yang Anda kenakan, seberapa cepat Anda bergerak, seberapa lelah Anda, dan seterusnya, tebakan Anda pada akhirnya akan menjadi sangat akurat. Pada dasarnya inilah yang terjadi dengan kalibrasi dan pemodelan IMU.

Ada banyak sumber kesalahan. Lengan robot biasanya digunakan untuk menggerakkan perangkat berulang kali dengan cara yang persis sama berulang-ulang, dan output dari IMU ditangkap dan disaring hingga output dari IMU secara akurat cocok dengan gerakan *ground truth* dari lengan robot. Google dan Microsoft melangkah lebih jauh dengan mengirim perangkat mereka ke gayaberat mikro di Stasiun Luar Angkasa Internasional, atau "penerbangan gravitasi nol", untuk menghilangkan kesalahan tambahan.



Gambar 5-4. Kalibrasi inersia bahkan lebih menantang dan tidak ada kasus penggunaan yang membutuhkannya sebelumnya (untuk perangkat keras konsumen)

Ini bahkan lebih sulit daripada kedengarannya untuk mencapai akurasi nyata. Berikut adalah beberapa kesalahan accelerometer yang harus diidentifikasi dari jejak seperti garis RGB pada grafik yang ditunjukkan pada Gambar 5-5:

Bias tetap

Pengukuran akselerasi bukan nol ketika akselerasi nol terintegrasi

Kesalahan faktor skala

Penyimpangan output aktual dari model matematika output (biasanya output nonlinier)

Kopling silang

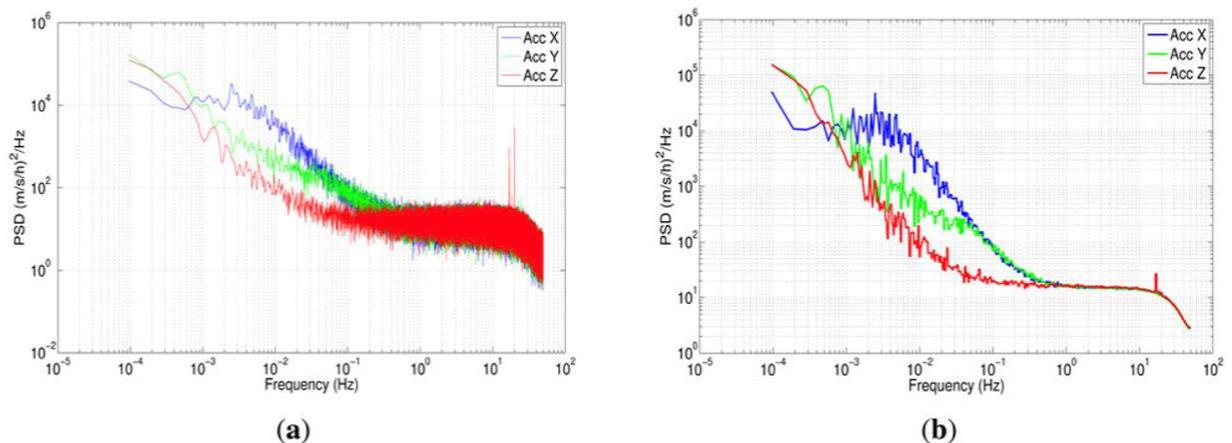
Percepatan dalam arah ortogonal ke arah pengukuran sensor diteruskan ke pengukuran sensor (ketidaksempurnaan manufaktur, sumbu sensor non-ortogonal)

Kesalahan getar yang terjumbai

Getaran sefase dengan perpindahan pendulum (bayangkan seorang anak di ayunan)

Kesalahan jam

Periode integrasi salah diukur



Gambar 5-5. Ini hanya beberapa kesalahan yang harus diidentifikasi dari jejak seperti garis RGB pada grafik

Ini juga merupakan tantangan bagi OEM untuk harus melalui proses ini untuk semua perangkat dalam portofolio mereka, dan meskipun demikian, banyak perangkat mungkin memiliki IMU yang berbeda (misalnya, Galaxy 7 mungkin memiliki IMU dari Invensense atau Bosch, dan tentu saja pemodelan untuk Bosch tidak berfungsi untuk Invensense, dll.). Ini adalah area lain di mana Apple memiliki keunggulan dibandingkan OEM Android.

5.5 MASA DEPAN TRACKING

Jadi, jika VIO berfungsi hari ini, apa yang akan terjadi selanjutnya dan apakah itu akan membuat ARKit menjadi mubazir? Anehnya, VIO akan tetap menjadi cara terbaik untuk melacak pada jarak beberapa ratus meter (lebih lama dari itu, sistem perlu direlokasi menggunakan kombinasi GPS yang digabungkan ke dalam sistem ditambah semacam pengenalan tengara). Alasan untuk ini adalah bahwa bahkan jika sistem hanya optik lainnya menjadi seakurat VIO, mereka masih memerlukan lebih banyak daya (unit pemrosesan grafis

[GPU] atau kamera), yang sangat penting dalam HMD. VIO bermata adalah solusi yang paling akurat, daya terendah, dan biaya terendah.

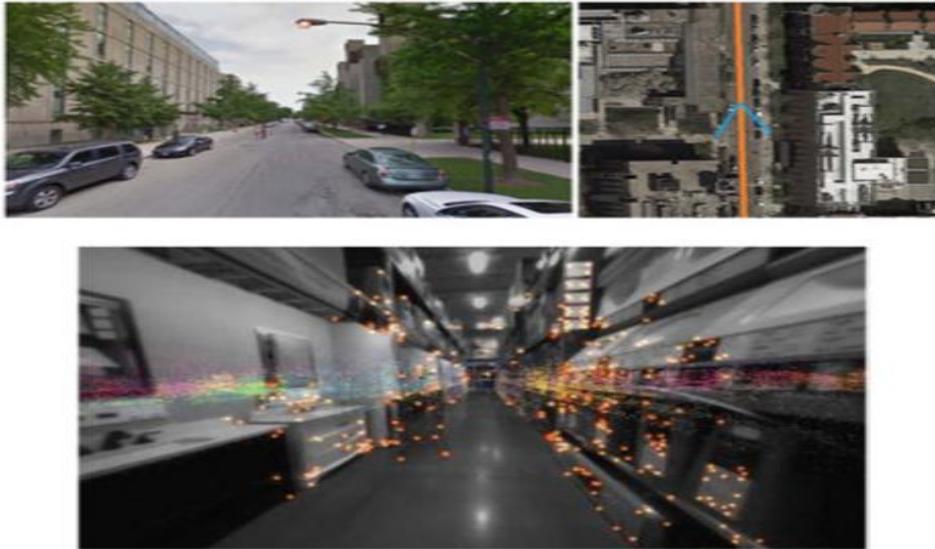
Pembelajaran mendalam benar-benar berdampak pada komunitas riset untuk tracking. Sejauh ini, sistem berbasis pembelajaran mendalam sekitar 10% keluar sehubungan dengan kesalahan, sedangkan sistem VIO teratas adalah sebagian kecil dari persen, tetapi mereka mengejar dan akan sangat membantu dengan relokasi luar ruangan.

Kamera kedalaman (Gambar 5-6) dapat membantu sistem VIO dalam beberapa cara. Pengukuran yang akurat dari kebenaran dasar dan skala metrik serta tracking tepi untuk pemandangan berfitur rendah adalah manfaat terbesarnya. Namun, mereka sangat haus daya, jadi masuk akal untuk menjalankannya hanya pada kecepatan bingkai yang sangat rendah dan menggunakan VIO di antara bingkai. Mereka juga tidak bekerja di luar ruangan karena hamburan inframerah latar belakang dari sinar matahari menyapu inframerah dari kamera kedalaman. Juga, jangkauannya tergantung pada konsumsi dayanya, yang berarti bahwa pada telepon, jangkauannya sangat pendek (beberapa meter). Mereka juga mahal dalam hal biaya BOM, jadi OEM akan menghindarinya untuk telepon bervolume tinggi.

Stereo RGB atau lensa fisheye keduanya membantu untuk dapat melihat pemandangan yang lebih besar dan dengan demikian berpotensi lebih banyak fitur optik (misalnya, lensa biasa mungkin melihat dinding putih, tapi fisheye bisa melihat langit-langit berpola dan karpet di bingkai, juga — Magic Leap dan Hololens menggunakan pendekatan ini) dan mungkin mendapatkan informasi mendalam dengan biaya komputasi yang lebih rendah daripada VIO, meskipun VIO melakukannya dengan akurat untuk BOM dan biaya daya yang lebih rendah. Karena kamera stereo pada ponsel atau bahkan HMD berdekatan, jangkauan akuratnya sangat terbatas untuk penghitungan kedalaman (kamera yang terpisah beberapa sentimeter dapat akurat untuk kedalaman hingga beberapa meter).

Hal yang paling menarik adalah dukungan untuk tracking di area yang jauh lebih besar, terutama di luar ruangan selama beberapa kilometer. Pada titik ini, hampir tidak ada perbedaan antara tracking untuk AR dan tracking untuk mobil *self-driving*, kecuali sistem AR melakukannya dengan lebih sedikit sensor dan daya yang lebih rendah. Karena pada akhirnya perangkat apa pun akan kehabisan ruang saat mencoba memetakan area yang luas, layanan yang didukung cloud diperlukan; Google baru-baru ini mengumumkan Layanan Pemosisian Visual Tango karena alasan ini. Kami akan melihat lebih banyak dari ini dalam waktu dekat. Itu juga alasan mengapa semua orang sangat peduli dengan peta 3D saat ini.





Gambar 5-6. Masa depan tracking

5.6 MASA DEPAN AR KOMPUTER VISION

Tracking posisi enam derajat kebebasan (6DOF) sudah hampir sepenuhnya menjadi komoditas, di semua perangkat; 2019 akan melihatnya sebagai fitur default di chipset dan perangkat pasar massal. Tetapi masih ada hal-hal yang perlu diselesaikan. Mari luangkan waktu sejenak untuk memeriksanya di sini saat kita melihat masa depan visi komputer AR.



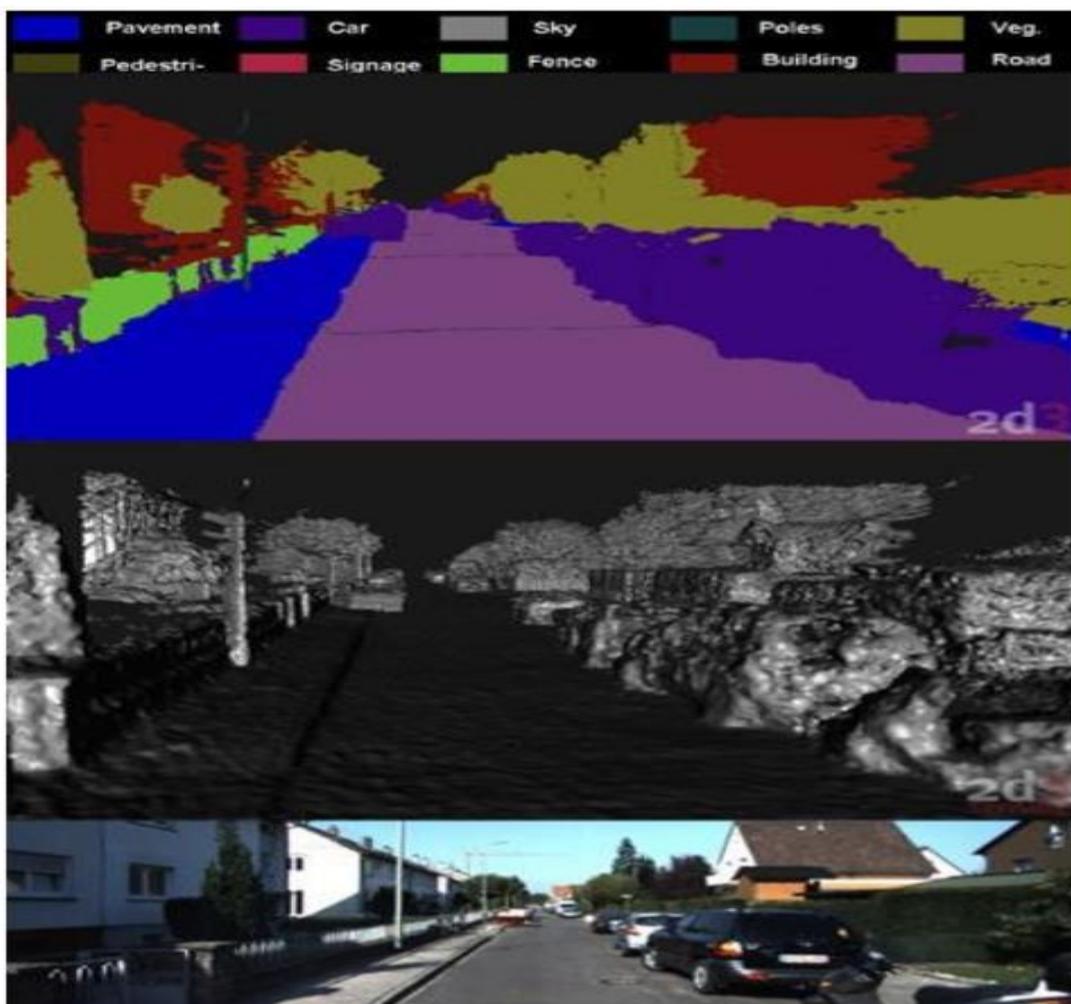
Gambar 5-7. Rekonstruksi 3D skala besar



Gambar 5-8. Aplikasi ini dibangun oleh startup penulis sebelumnya, Dekko

Rekonstruksi 3D (pemetaan spasial dalam istilah Hololens atau persepsi kedalaman dalam istilah Tango) adalah sistem yang mampu mengetahui bentuk atau struktur objek nyata dalam sebuah Scene, seperti yang ditunjukkan pada Gambar 5-7. Itulah yang memungkinkan konten virtual bertabrakan dan bersembunyi di balik (oklusi) dunia nyata. Itu juga fitur yang membingungkan orang karena mereka pikir ini berarti AR sekarang adalah realitas "campuran". Itu selalu AR, hanya saja sebagian besar demonstrasi AR yang dilihat orang tidak memiliki dukungan rekonstruksi 3D, sehingga konten tampak hanya bergerak di depan semua objek dunia nyata.

Rekonstruksi 3D bekerja dengan menangkap titik-awan padat dari tempat kejadian (hari ini menggunakan kamera kedalaman) dan kemudian mengubahnya menjadi jala dan memasukkan jala "tak terlihat" ke Unity (bersama dengan koordinat dunia nyata), dan kemudian menempatkan jala dunia nyata persis di atas dunia nyata seperti yang muncul di kamera. Ini berarti konten virtual tampak berinteraksi dengan dunia nyata. Saat rekonstruksi 3D menjadi lebih besar, kita perlu mencari cara untuk meng-host-nya di cloud dan membiarkan banyak pengguna berbagi (dan memperluas) model.



Gambar 5-9. Rekonstruksi 3D skala besar

ARKit melakukan versi 2D hari ini dengan mendeteksi pesawat 2D. Ini adalah minimal yang dibutuhkan. Tanpa ground plane, konten Unity secara harfiah tidak akan memiliki landasan untuk berdiri dan akan melayang-layang.

Gambar 5-8 menunjukkan upaya awal untuk mendemonstrasikan oklusi dengan membuat mesh menggunakan iPad 2. Ini adalah aplikasi pertama yang menunjukkan interaksi fisik antara konten virtual dan dunia nyata pada perangkat keras seluler komoditas.

Gambar 5-9 menyajikan contoh segmentasi semantik 3D dari sebuah Scene. Sumber gambar ada di bawah. Di atas itu adalah model 3D (mungkin dibangun dari kamera stereo, atau LIDAR), dan di atas adalah segmentasi melalui pembelajaran mendalam; sekarang kita bisa membedakan trotoar dari jalan. Hal ini juga berguna untuk Pokemon Go agar Pokemon tidak ditempatkan di tengah jalan yang ramai.

Kemudian, kita perlu mencari cara untuk menskalakan semua teknologi luar biasa ini untuk mendukung banyak pengguna secara bersamaan secara *real time*. Ini adalah Game *Roleplaying Online Massively Multiplayer* (MMORG, misalnya, *World of Warcraft*, tetapi untuk dunia nyata). Berikut adalah beberapa tantangan lain yang perlu kita atasi dan selesaikan:

- Semuanya di atas tumpukan
 - Rendering (koherensi, kinerja)
 - Memasukkan
 - Optik
 - GUI dan aplikasi
 - Faktor sosial

5.7 MAPPING

Pemetaan adalah "M" di SLAM. Ini mengacu pada struktur data yang disimpan perangkat dalam memori yang berisi informasi tentang Scene 3D yang dapat dilokalkan oleh pelacak (istilah umum untuk sistem VIO). Untuk melokalisasi berarti menentukan di mana saya berada di peta. Jika saya menutup mata Anda dan menjatuhkan Anda di tengah kota baru dengan peta kertas, proses yang Anda lalui dengan melihat sekeliling, lalu melihat peta, lalu melihat sekeliling lagi sampai Anda memastikan di mana Anda berada di peta adalah prosesnya dari melokalisasi diri sendiri.

Pada tingkat yang paling sederhana, peta SLAM adalah grafik titik 3D yang mewakili titik-awan yang jarang, di mana setiap titik sesuai dengan koordinat fitur optik di tempat kejadian (misalnya, sudut tabel). Mereka biasanya berisi cukup banyak metadata tambahan di sana, juga, seperti seberapa "dapat diandalkan" titik itu, diukur dengan berapa banyak bingkai yang fitur tersebut terdeteksi dalam koordinat yang sama baru-baru ini (misalnya, titik hitam pada anjing saya akan tidak ditandai dapat diandalkan karena anjing bergerak). Beberapa peta menyertakan "bingkai utama", yang hanya merupakan satu bingkai video (pada dasarnya foto) yang disimpan di peta setiap beberapa detik dan digunakan untuk membantu pelacak mencocokkan dunia dengan peta. Peta lain menggunakan cloud point padat, yang lebih andal tetapi membutuhkan lebih banyak GPU dan memori. ARCore dan ARKit keduanya menggunakan peta yang jarang (tanpa keyframe, saya pikir).

Peta yang jarang mungkin terlihat seperti gambar kanan atas pada Gambar 5-10. Kiri atas menunjukkan bagaimana titik fitur cocok dengan dunia nyata (warna digunakan untuk menunjukkan seberapa andal titik itu). Kiri bawah adalah gambar sumber, dan kanan bawah adalah peta intensitas, yang dapat digunakan untuk jenis sistem SLAM yang berbeda (semi-langsung — yang sangat bagus, tetapi belum dalam sistem SLAM produksi seperti ARCore atau ARKit).

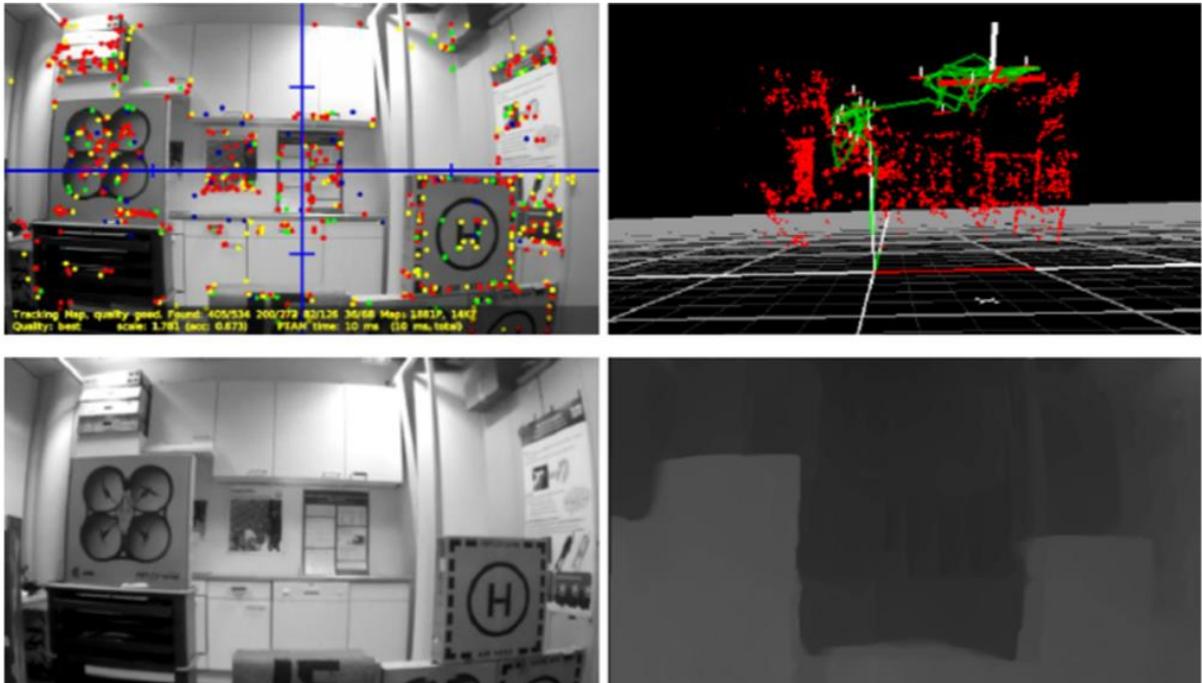
Jadi bagaimana cara kerjanya? Saat Anda meluncurkan aplikasi ARCore/ARKit, pelacak memeriksa untuk melihat apakah ada peta yang telah diunduh sebelumnya dan siap digunakan (tidak pernah ada di ARCore dan ARKit v1.0). Jika tidak ada, pelacak menginisialisasi peta baru dengan melakukan perhitungan stereo, seperti yang saya jelaskan sebelumnya. Ini berarti bahwa kami sekarang memiliki peta 3D kecil yang bagus tentang apa yang ada di bidang pandang kamera. Saat Anda mulai bergerak dan bagian baru dari latar belakang bergerak ke bidang pandang, lebih banyak titik 3D ditambahkan ke peta dan menjadi lebih besar. Dan lebih besar. Dan lebih besar.

Ini tidak pernah menjadi masalah karena pelacak sangat buruk sehingga mereka hanyut tanpa bisa digunakan sebelum peta menjadi terlalu besar untuk dikelola. Itu tidak terjadi lagi, dan mengelola peta adalah tempat banyak pekerjaan menarik di SLAM berlangsung (bersama dengan pembelajaran mendalam dan jaringan saraf convolutional). ARKit menggunakan "jendela geser" untuk petanya, yang berarti bahwa ia hanya menyimpan sejumlah variabel dari masa lalu (waktu dan jarak) baru-baru ini di peta, dan membuang apa pun yang lama. Asumsinya adalah bahwa Anda tidak perlu melakukan relokasi terhadap Scene beberapa waktu lalu. ARCore mengelola peta yang lebih besar, yang berarti bahwa sistem harus lebih andal. Jadi, hasilnya adalah dengan ARCore, bahkan jika Anda kehilangan tracking, itu akan pulih lebih baik dan Anda tidak akan terpengaruh.

ARCore dan ARKit juga menggunakan konsep cerdas yang disebut jangkar untuk membantu membuat peta terasa seperti mencakup area fisik yang lebih besar daripada yang sebenarnya. Saya melihat konsep ini pertama kali di Hololens, yang, seperti biasa, satu tahun atau lebih di depan orang lain. Biasanya, sistem mengelola peta sepenuhnya tanpa terlihat oleh pengguna atau pengembang aplikasi. Jangkar memungkinkan pengembang untuk menginstruksikan sistem untuk "mengingat bagian peta ini di sekitar sini, jangan membuangnya." Ukuran fisik jangkar sekitar satu meter persegi (itu sedikit tebakan di pihak saya; mungkin variabel tergantung pada berapa banyak fitur optik yang dapat dilihat sistem). Cukup bagi sistem untuk melakukan relokasi ketika lokasi fisik ini dikunjungi kembali oleh pengguna). Pengembang biasanya menjatuhkan jangkar setiap kali konten ditempatkan di lokasi fisik. Artinya, jika pengguna kemudian mengembara, sebelum berlabuh, peta di sekitar lokasi fisik tempat konten seharusnya ada akan dibuang dan konten akan hilang. Dengan jangkar, konten selalu berada di tempat yang seharusnya, dengan dampak UX terburuk adalah kemungkinan kesalahan kecil dalam konten saat sistem melakukan relokasi dan melompat untuk mengoreksi akumulasi penyimpangan (jika ada).

Tujuan dari peta adalah untuk membantu pelacak dalam dua cara. Pertama adalah ketika saya menggerakkan ponsel saya maju mundur, peta dibangun dari gerakan awal, dan

dalam perjalanan kembali, fitur yang terdeteksi secara real time dapat dibandingkan dengan fitur yang disimpan di peta. Ini membantu membuat tracking lebih stabil dengan hanya menggunakan fitur yang paling andal dari tampilan Scene saat ini dan sebelumnya dalam perhitungan pose.



Gambar 5-10. Contoh dari apa yang dilihat sistem AR, diamparkan pada gambar yang dapat dibaca manusia

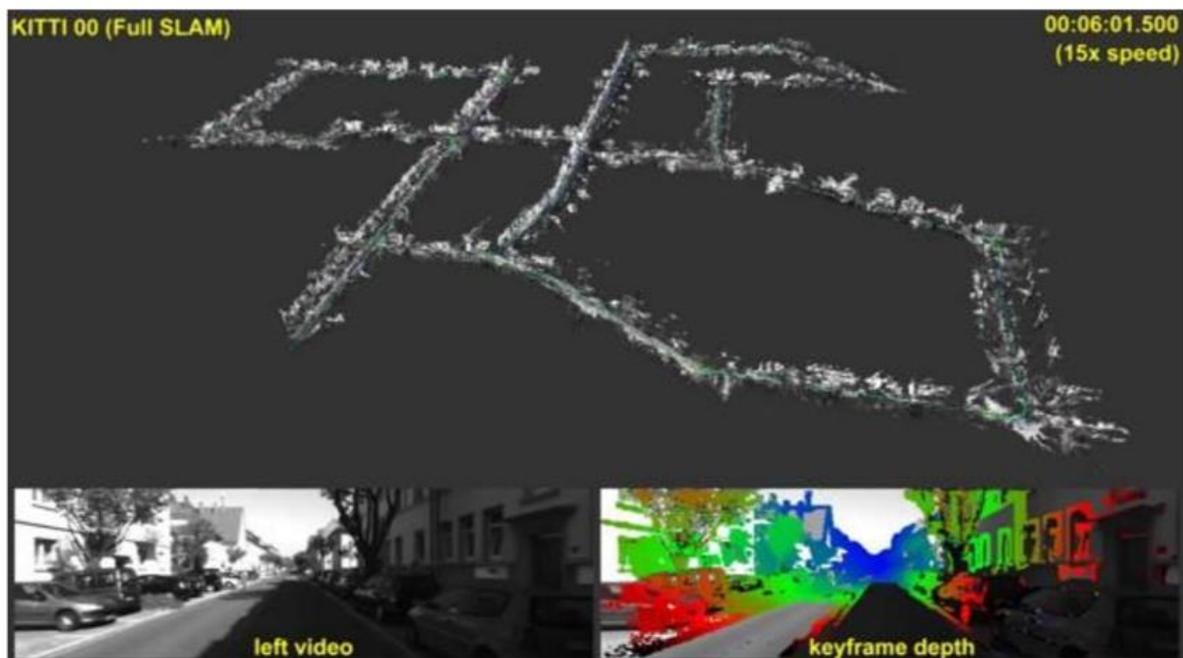
Cara kedua yang membantu Peta adalah dengan melokalkan (atau memulihkan) tracking. Akan tiba saatnya ketika Anda menutupi kamera, menjatuhkan ponsel Anda, bergerak terlalu cepat, atau sesuatu yang acak terjadi, dan ketika kamera berikutnya melihat pemandangan, itu tidak sesuai dengan apa yang menurut pembaruan terakhir dari peta yang seharusnya dilihatnya. Sudah ditutup matanya dan dijatuhkan di tempat baru. Ini adalah definisi dari "Saya kehilangan tracking", yang dikatakan oleh pengembang AR perintis sekitar seribu kali setiap hari selama beberapa tahun terakhir.

Pada titik ini sistem dapat melakukan salah satu dari dua hal:

- Atur ulang semua sistem koordinat dan mulai lagi! Inilah yang dilakukan oleh sistem odometri murni (tanpa peta sama sekali). Apa yang Anda alami adalah bahwa semua konten Anda melompat ke posisi baru dan tetap di sana. Ini bukan UX yang bagus.
- Sistem dapat mengambil serangkaian fitur 3D yang dilihatnya sekarang dan menelusuri seluruh peta untuk mencoba menemukan kecocokan, yang kemudian diperbarui sebagai posisi virtual yang benar, dan Anda dapat terus menggunakan aplikasi seolah-olah tidak terjadi apa-apa (Anda mungkin melihat kesalahan dalam konten virtual Anda saat tracking hilang, tetapi akan kembali ke tempat semula saat pulih). ada dua masalah disini. Pertama, saat peta bertambah besar, proses pencarian ini menjadi

sangat memakan waktu dan prosesor, dan semakin lama proses ini berlangsung, semakin besar kemungkinan pengguna untuk pindah lagi, yang berarti pencarian harus dimulai lagi. Kedua, posisi ponsel saat ini tidak pernah sama persis dengan posisi ponsel sebelumnya, jadi ini juga meningkatkan kesulitan pencarian peta, dan menambah komputasi dan waktu untuk upaya relokasi. Jadi, pada dasarnya, bahkan dengan pemetaan, jika Anda bergerak terlalu jauh dari peta, Anda kacau dan sistem perlu diatur ulang dan mulai lagi!

Setiap garis pada gambar yang ditunjukkan pada Gambar 5-11 adalah jalan di peta SLAM skala besar ini. Membuat perangkat seluler melakukan AR di mana saja dan di mana saja di dunia adalah masalah pemetaan SLAM yang sangat besar. Ingatlah bahwa ini adalah peta dan struktur data yang dapat dibaca mesin; mereka bukan peta gaya tampilan jalan 3D yang bagus dan nyaman yang dapat digunakan manusia (yang juga diperlukan!).



Gambar 5-11. Pemetaan SLAM skala besar merupakan tantangan bagi AR . berbasis ponsel

Juga perlu diingat bahwa dalam diskusi kami ketika saya merujuk ke peta "besar", untuk AR seluler, itu secara kasar berarti peta yang mencakup area fisik ruangan yang sangat besar atau apartemen yang sangat kecil. Perhatikan juga ini berarti untuk AR luar ruangan kita perlu memikirkan pemetaan dengan cara yang sama sekali baru.

Relokasi yang kuat terhadap peta besar adalah masalah yang sangat, sangat, sangat sulit, dan menurut saya, pada tulisan ini, belum ada yang memecahkan masalah ke tingkat konsumen-UX. Siapa pun yang mengklaim menawarkan konten AR multipemain atau persisten akan memiliki UX yang sangat dibatasi oleh kemampuan ponsel kedua (mis., Pemain 2) untuk melokalisasi dari awal yang dingin ke peta yang dibuat oleh Pemain 1 atau diunduh dari cloud . Anda akan menemukan Player 2 harus berdiri cukup dekat dengan Player 1 dan memegang telepon mereka dengan cara yang kira-kira sama. Ini adalah gangguan bagi

pengguna. Mereka hanya ingin duduk di sofa di seberang Anda dan menyalakan telepon mereka dan segera melihat apa yang Anda lihat (dari sisi yang berlawanan, tentu saja). Atau, Pemain 2 harus berdiri di mana saja dalam jarak beberapa meter dari posisi sebelumnya dan melihat konten AR "permanen" yang tertinggal di sana.

Ada solusi khusus aplikasi untuk multipemain yang juga dapat Anda coba, seperti menggunakan penanda atau hardcoding posisi awal yang jauh untuk Player 2, dan seterusnya. Secara teknis mereka dapat bekerja, tetapi Anda masih perlu menjelaskan apa yang harus dilakukan kepada pengguna, dan UX Anda dapat terkena atau gagal. Tidak ada solusi ajaib "itu hanya berfungsi" yang memungkinkan Anda melakukan relokasi (yaitu, bergabung dengan peta orang lain) dengan cara ARKit dan ARCore membuat tracking VIO "berfungsi."

Bagaimana Multiplayer AR Bekerja?

Agar multipemain berfungsi, kita perlu menyiapkan beberapa hal:

1. Kedua perangkat perlu mengetahui posisi mereka relatif satu sama lain. Secara teknis, ini berarti bahwa mereka perlu berbagi sistem koordinat yang sama dan mengetahui koordinat satu sama lain di setiap bingkai video. Sistem koordinat dapat berupa sistem dunia (misalnya, lintang dan bujur) atau mereka mungkin setuju untuk masing-masing menggunakan koordinat dari perangkat pertama untuk memulai. Ingatlah bahwa setiap perangkat ketika mulai biasanya hanya mengatakan, "Di mana pun saya sekarang adalah koordinat (0,0,0) saya," dan itu melacak pergerakan dari sana. My (0,0,0) secara fisik berada di tempat yang berbeda dengan (0,0,0) Anda. Untuk mengubah diri saya menjadi koordinat Anda, saya perlu merelokasi diri ke peta SLAM Anda dan mendapatkan pose saya di koordinat Anda dan kemudian menyesuaikan peta saya. Peta SLAM adalah semua data tersimpan yang memungkinkan saya melacak keberadaan saya.
2. Kami kemudian perlu memastikan untuk setiap bingkai bahwa masing-masing dari kita tahu di mana yang lain berada. Setiap perangkat memiliki pelacaknya sendiri yang terus memperbarui pose setiap bingkai. Jadi, untuk multipemain, kita perlu menyiarkan pose itu ke semua pemain lain di dalam game. Ini membutuhkan koneksi jaringan dari beberapa jenis, baik peer-to-peer, atau melalui layanan cloud. Seringkali juga akan ada beberapa aspek prediksi pose dan pemulusan yang terjadi untuk memperhitungkan gangguan jaringan kecil.
3. Kami berharap bahwa pemahaman 3D apa pun tentang dunia yang dimiliki setiap perangkat dapat dibagikan dengan perangkat lain (ini tidak wajib, meskipun UX akan sangat terpengaruh tanpanya). Ini berarti mengalirkan beberapa mesh 3D dan informasi semantik bersama dengan pose. Misalnya, jika perangkat saya telah menangkap model 3D yang bagus dari sebuah ruangan yang menyediakan kemampuan fisika dan oklusi, saat Anda bergabung dengan permainan saya, Anda seharusnya dapat menggunakan data yang sudah diambil itu, dan data tersebut harus diperbarui antar perangkat sebagai hasil permainan.
4. Terakhir, ada semua hal "normal" yang diperlukan untuk aplikasi multipengguna waktu nyata online. Ini termasuk mengelola izin pengguna, status waktu nyata setiap

pengguna (misalnya, jika saya mengetuk "tembak" dalam game, semua aplikasi pengguna lain perlu diperbarui agar saya memiliki "tembak"), dan mengelola semua berbagai aset bersama. Fitur teknis ini sama persis untuk aplikasi AR dan non-AR. Perbedaan utamanya adalah, hingga saat ini, mereka benar-benar dibuat hanya untuk game, sedangkan AR akan membutuhkannya untuk setiap jenis aplikasi. Untungnya, semua fitur ini telah dibuat berkali-kali untuk game MMO online dan seluler, dan mengadaptasinya untuk aplikasi non-game biasa, seperti yang ditunjukkan pada Gambar 5-12, tidak terlalu sulit.

Apa Bagian Sulitnya?

Bayangkan Anda berada di ruangan yang terkunci dan tidak berjendela dan Anda diberi foto trotoar kota. Ini menunjukkan beberapa bangunan dan nama toko di seberang jalan, mobil, orang, dan sebagainya. Anda belum pernah ke sini sebelumnya, benar-benar asing bagi Anda, bahkan tulisannya dalam bahasa asing. Tugas Anda adalah menentukan dengan tepat di mana foto itu diambil, dengan akurasi sekitar satu sentimeter (Gambar 5-13 mengilustrasikan permainan sebenarnya yang dapat Anda mainkan yang secara kasar mensimulasikan ini). Anda mendapatkan garis lintang dan garis bujur kasar dari GPS dan hanya secara kasar mengetahui arah mana yang Anda tuju, dan Anda tahu bahwa GPS bisa jadi tidak akurat 20–40 meter. Yang harus Anda lakukan hanyalah setumpuk foto yang diambil oleh orang lain di area yang kira-kira sama baru-baru ini, masing-masing ditandai dengan lokasi yang tepat.



Gambar 5-12. Bahkan aplikasi seperti ini membutuhkan cloud AR dan infrastruktur "MMO" untuk memungkinkan interaksi waktu nyata

Ini adalah masalah yang harus diselesaikan oleh sistem AR Anda setiap kali pertama kali dihidupkan atau jika "kehilangan tracking" oleh kamera yang ditutup sementara atau jika menunjuk pada sesuatu yang tidak dapat dilacak (dinding putih, langit biru, dll.). Itu juga masalah yang perlu diselesaikan jika Anda ingin bergabung dengan game AR teman Anda. Foto

Anda adalah gambar langsung dari kamera perangkat Anda, tumpukan foto adalah peta SLAM yang telah Anda muat ke dalam memori (mungkin disalin dari perangkat teman Anda atau yang dibuat sebelumnya). Anda juga harus menyelesaikan tugas sebelum pengguna menggerakkan kamera dan membuat gambar langsung terbaru menjadi tidak relevan.

Untuk mengilustrasikan masalahnya, mari kita ambil dua contoh ekstrem. Dalam kasus pertama, Anda menemukan foto di tumpukan yang terlihat hampir persis seperti foto yang Anda miliki. Anda dapat dengan mudah memperkirakan bahwa foto Anda berada di belakang dan di sebelah kiri tumpukan foto, jadi sekarang Anda memiliki perkiraan yang sangat akurat tentang posisi pengambilan foto Anda. Ini sama dengan meminta Pemain 2 untuk pergi dan berdiri tepat di samping Pemain 1 saat Pemain 2 memulai permainan mereka. Kemudian, mudah bagi sistem Player 2 untuk menentukan di mana itu relatif terhadap Player 1, dan sistem dapat menyelaraskan koordinat (lokasi) mereka dan aplikasi dapat berjalan dengan senang hati.

Dalam contoh lain, ternyata, tanpa sepengetahuan Anda, semua foto di tumpukan Anda diambil menghadap ke selatan, sedangkan foto Anda menghadap ke utara. Hampir tidak ada kesamaan antara foto Anda dan apa yang ada di tumpukan. Ini setara dengan AR mencoba memainkan permainan papan virtual dan Pemain 1 berada di satu sisi meja dan Pemain 2 duduk di sisi yang berlawanan dan mencoba untuk bergabung dengan permainan. Dengan pengecualian beberapa bagian tabel itu sendiri (yang Anda lihat terbalik dengan apa yang ada di tumpukan), sangat sulit bagi sistem untuk menyinkronkan peta mereka (merelokasi).



Gambar 5-13. Untuk merasakan betapa sulitnya relokasi, coba mainkan game Geoguessr, yang sangat mirip dengan masalah yang harus diselesaikan oleh sistem AR Anda setiap kali Anda menyalakannya

Perbedaan antara contoh-contoh ini menggambarkan mengapa hanya karena seseorang mengklaim bahwa mereka dapat mendukung AR multipemain, itu mungkin juga

berarti bahwa ada beberapa kompromi UX signifikan yang perlu dilakukan pengguna. Pengalaman saya dalam membangun sistem AR multipemain sejak 2012 memberi tahu saya bahwa tantangan UX dari contoh pertama (mengharuskan orang untuk berdiri berdampingan untuk memulai) terlalu sulit untuk diatasi oleh pengguna. Mereka membutuhkan banyak pegangan tangan dan penjelasan, dan gesekannya terlalu tinggi. Mendapatkan pengalaman multipemain tingkat konsumen berarti menyelesaikan kasus kedua (dan lebih banyak lagi).

Selain kasus kedua, foto dalam tumpukan bisa dari jarak yang sangat jauh berbeda, di bawah kondisi pencahayaan yang berbeda (bayangan pagi versus sore terbalik) atau menggunakan model kamera yang berbeda, yang memengaruhi tampilan gambar dibandingkan dengan milik Anda (yang berwarna cokelat dinding mungkin tidak berwarna cokelat pada gambar Anda seperti pada gambar saya). Anda juga mungkin bahkan tidak memiliki GPS yang tersedia (mungkin Anda berada di dalam ruangan), jadi Anda bahkan tidak dapat memulai dengan gambaran kasar tentang di mana Anda berada.

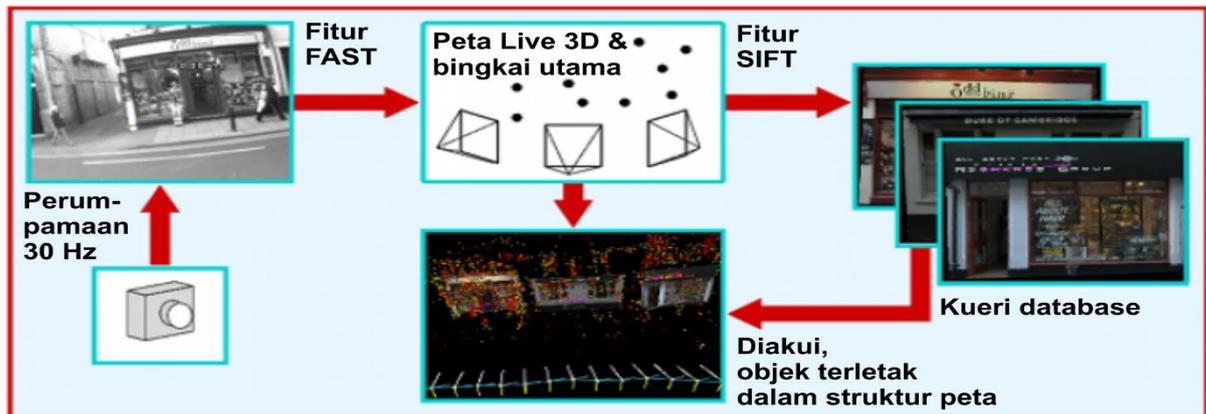
Sentuhan "menyenangkan" terakhir dari semua ini adalah bahwa pengguna menjadi bosan menunggu. Jika proses relokasi memakan waktu lebih dari satu atau dua detik, pengguna biasanya memindahkan perangkat dengan cara tertentu, dan Anda harus memulai dari awal lagi! Relokasi yang akurat dan kuat (dalam semua kasus) masih menjadi salah satu tantangan luar biasa untuk AR (dan robot, mobil otonom, dll.).

Bagaimana Relokasi Bekerja?

Jadi bagaimana cara kerjanya? Bagaimana masalah ini diselesaikan hari ini? Apa yang akan segera hadir? Pada intinya, relokasi adalah jenis masalah pencarian yang sangat spesifik. Anda mencari melalui peta SLAM, yang mencakup area fisik, untuk menemukan di mana perangkat Anda berada di koordinat peta itu. Peta SLAM biasanya memiliki dua jenis data di dalamnya: sparse point-cloud dari semua titik 3D yang dapat dilacak di ruang itu, dan sejumlah keyframe. Seperti yang saya sebutkan sebelumnya, bingkai utama adalah satu bingkai video yang diambil dan disimpan sebagai foto sesekali saat sistem berjalan. Sistem memutuskan berapa banyak keyframe yang akan diambil berdasarkan seberapa jauh perangkat telah bergerak sejak keyframe terakhir serta perancang sistem yang melakukan trade-off untuk kinerja. Lebih banyak bingkai utama yang disimpan berarti lebih banyak peluang untuk menemukan kecocokan saat relokasi, tetapi ini membutuhkan lebih banyak ruang penyimpanan dan juga membutuhkan waktu lebih lama untuk mencari melalui kumpulan bingkai utama.

Jadi, proses pencarian sebenarnya ada dua, seperti yang diilustrasikan pada Gambar 5-14. Potongan pertama seperti yang baru saja saya jelaskan pada contoh tumpukan foto. Anda membandingkan gambar kamera langsung Anda saat ini dengan rangkaian bingkai utama di peta SLAM. Bagian kedua adalah bahwa perangkat Anda juga secara instan membangun satu set kecil titik 3D sendiri segera setelah Anda menyalakannya, hanya berdasarkan apa yang dilihatnya saat ini, dan mencari kecocokan melalui SLAM sparse point-cloud. Ini seperti memiliki potongan puzzle 3D (*point-cloud* kecil dari kamera Anda) dan mencoba menemukan kecocokan dalam jigsaw 3D besar, yang setiap potongannya berwarna abu-abu rata di kedua sisinya.

Karena terbatasnya jumlah waktu yang tersedia sebelum pengguna menjadi bosan dan daya komputasi sederhana dari perangkat seluler saat ini, sebagian besar upaya relokasi dilakukan untuk mengurangi ukuran jendela pencarian sebelum harus melakukan semua jenis pencarian bruteforce melalui SLAM. peta. GPS yang lebih baik, pelacak yang lebih baik, dan sensor yang lebih baik semuanya sangat membantu dalam hal ini.



Gambar 5-14. Ikhtisar tentang bagaimana sebagian besar sistem SLAM saat ini membangun peta SLAM mereka menggunakan kombinasi fitur optik (sparse 3D point-cloud) dan database keyframe

Apa Kecanggihan Riset (dan Segera Hadir untuk Konsumen)?

Meskipun metode relokasi yang dijelaskan di bagian sebelumnya adalah pendekatan yang paling umum, ada metode lain yang melihat hasil yang bagus di laboratorium dan harus segera hadir untuk produk komersial. Salah satu metode, yang disebut PoseNet (lihat Gambar 5-15) melibatkan penggunaan regresi jaringan saraf bingkai penuh untuk memperkirakan pose perangkat. Ini tampaknya dapat menentukan pose Anda dengan akurasi sekitar satu meter di bawah berbagai kondisi. Metode lain meregresi pose kamera untuk setiap piksel dalam gambar.

Apakah Masalah Relokasi Benar-Benar Dapat Dipecahkan untuk Konsumen?

Ya! Faktanya, hingga tulisan ini dibuat, ada beberapa peningkatan yang cukup besar selama 12 bulan terakhir berdasarkan hasil penelitian mutakhir. Sistem pembelajaran mendalam memberikan hasil yang mengesankan untuk mengurangi jendela pencarian untuk relokasi di area yang luas, atau pada sudut yang sangat lebar ke pengguna awal. Mencari peta SLAM yang dibuat dari cloud titik 3D padat dari lokasi kejadian (daripada cloud titik jarang yang digunakan untuk tracking) juga memungkinkan algoritme relokasi baru yang sangat kuat. Saya telah melihat sistem rahasia yang dapat dipindahkan dari sudut mana pun pada jarak yang sangat jauh secara real time pada perangkat keras seluler serta mendukung sejumlah besar pengguna secara bersamaan. Dengan asumsi bahwa hasil yang terlihat dalam penelitian terbawa ke dalam sistem kelas komersial, saya yakin ini akan memberikan solusi "tingkat konsumen" yang kami harapkan.

Tapi ini masih hanya sebagian solusi untuk sepenuhnya menyelesaikan relokasi untuk garis lintang dan garis bujur yang tepat dan untuk lingkungan yang ditolak GPS, atau bagian

dunia di mana tidak ada sistem SLAM sebelumnya (mulai dingin). Tetapi saya telah melihat demonstrasi yang memecahkan sebagian besar masalah titik ini, dan percaya bahwa hanya dibutuhkan tim yang cerdas untuk secara bertahap mengintegrasikannya ke dalam solusi yang lengkap. Relokalisasi skala besar sekarang berada di ambang masalah teknik, bukan masalah sains.

Tidak bisakah Google atau Apple Lakukan Ini?

Tidak juga. Google telah mendemonstrasikan layanan yang disebut Visual Positioning System (VPS; lihat Gambar 5-16) untuk platform Tango yang dihentikan yang memungkinkan beberapa kemampuan relokasi antar perangkat—semacam peta SLAM bersama di cloud. Itu tidak mendukung multipemain, tetapi itu membuat langkah menuju pemecahan bagian teknis yang sulit. Itu tidak pernah tersedia untuk umum jadi saya tidak bisa mengatakan seberapa baik kerjanya di dunia nyata, tetapi demonstrasinya terlihat bagus (seperti yang mereka lakukan). Semua perusahaan platform AR utama sedang berupaya meningkatkan relocalizers mereka yang merupakan bagian dari ARKit, ARCore, HoloLens, Snap, dan sebagainya. Ini terutama untuk membuat sistem tracking mereka lebih andal, tetapi pekerjaan ini juga dapat membantu multipemain.

PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization
 Alex Kendall, Matthew Grimes and Roberto Cipolla University of Cambridge, UK

Demo, dataset and video are available here - <http://mi.eng.cam.ac.uk/projects/relocalisation/>

1. Problem
 Estimate the 3D position and orientation of the camera, given a single monocular image taken from a large previously explored scene.
 Applications: Augmented reality, Robot localization, Metric SLAM initialization.
 We use convnets for this task, as they do not require searching through a large database of landmarks that scales linearly with map size.

2. Dataset
 We built a training dataset offline with 10,000 pose labelled images. We recorded video then solved for the camera pose in each frame using structure from motion.

3. PoseNet
 Perform end-to-end supervised learning with euclidean loss to regress 6-DOF pose.
 PoseNet does not require a large database of landmarks. Instead, it learns robust high level features to regress 6-DOF pose.
 The system output (red) is accurate even when there are large differences in pose between training (green) and testing (blue) images.

4. Robust to Challenging Situations
 Blur, Occlusion, Dusk, Night

5. Regression Outperforms Classification

Scene	# Frames	Scenes	Dist. to Conv. Nearest Neighbour	PoseNet	PoseNet with 128 Crops
King's College	1225	3x3	1.02 x 40m	3.36m, 2.96°	1.85m, 2.70°
Street	3015	2023	800 x 100m	1.96m, 4.51°	3.67m, 3.26°
Old Hospital	895	162	50 x 40m	3.30m, 4.51°	2.21m, 2.69°
Shoe Factory	231	103	25 x 25m	2.19m, 5.20°	1.49m, 4.64°
St Mary's Church	1487	503	85 x 80m	4.48m, 5.66°	2.65m, 4.24°
Chess	4000	2000	3 x 2 x 1m	0.41m, 0.00°	0.32m, 4.06°
Frie	2000	2000	2.5 x 1 x 1m	0.56m, 7.77°	0.47m, 7.50°
Heads	1000	1000	2 x 0.8 x 1m	0.20m, 7.50°	0.20m, 6.60°
Office	6000	4000	2.5 x 2 x 1.5m	0.49m, 0.02°	0.48m, 3.84°
Pungkin	4000	2000	2.3 x 2 x 1.5m	0.58m, 0.08°	0.47m, 4.21°
Red Kitchen	7000	5000	4 x 3 x 1.5m	0.58m, 5.66°	0.59m, 4.32°
Stairs	2000	1000	2.5 x 2 x 1.5m	0.50m, 7.71°	0.47m, 6.92°

Median relocalisation error for each outdoor and indoor scene. PoseNet is more accurate than a location classifier

6. Summary
 End to end 6DOF pose regression with convnet from a single colour image
 5ms, 50 MB memory at runtime
 Large scale indoor and outdoor relocalisation robust to blur, lighting, occlusion and camera intrinsics
 We release a public dataset of 5 Scenes in Cambridge containing over 10,000 pose annotated images

Future work: we aim to investigate Bayesian convolutional neural networks to model relocalisation uncertainty [2].

[1] Alex Kendall, Matthew Grimes and Roberto Cipolla "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization." Proceedings of the International Conference on Computer Vision (ICCV), 2015.
 [2] Alex Kendall and Roberto Cipolla "Modeling Uncertainty in Deep Learning for Camera Relocalization." arXiv preprint arXiv:1509.03909, 2015.

Gambar 5-15. PoseNet adalah indikasi ke mana arah sistem

VPS adalah contoh yang baik dari peta SLAM bersama yang dihosting di cloud. Namun, ini sepenuhnya terkait dengan algoritma dan struktur data SLAM Google, dan tidak akan

digunakan oleh Apple, Microsoft, atau OEM SLAM lainnya (yang mungkin menginginkan sistem mereka sendiri atau bermitra dengan pihak ketiga yang netral).



Gambar 5-16. Google telah mengembangkan Layanan Pemosisian Visual skala besar selama bertahun-tahun

Masalah besar yang dimiliki setiap platform utama dengan multipemain adalah bahwa mereka hanya dapat mengaktifkan multipemain di dalam ekosistem mereka—ARCore ke ARCore, ARKit ke ARKit, dan seterusnya. Ini karena agar relokasi lintas platform berfungsi, perlu ada peta SLAM umum di kedua sistem. Ini berarti bahwa Apple perlu memberi Google akses ke data SLAM mentahnya, dan sebaliknya (ditambah HoloLens, Magic Leap juga membuka, dll.). Meskipun secara teknis memungkinkan, ini adalah jembatan komersial yang terlalu jauh, karena perbedaan utama dalam UX antara berbagai sistem AR sebagian besar merupakan kombinasi dari integrasi perangkat keras dan perangkat lunak, dan kemudian kemampuan sistem pemetaan SLAM. Jadi, dengan tidak adanya semua platform besar yang setuju untuk membuka semua data mereka satu sama lain, opsinya terbatas pada hal berikut:

- Pihak ketiga yang independen dan netral bertindak sebagai layanan relokasi lintas platform
- Platform relokasi umum terbuka muncul

Keyakinan pribadi saya adalah bahwa karena integrasi yang sangat erat antara algoritma relokasi SLAM dan struktur data, sistem khusus yang dibuat khusus akan mengungguli (dari aspek UX) sistem terbuka umum untuk beberapa waktu. Ini telah terjadi selama bertahun-tahun dalam visi komputer—platform terbuka seperti OpenCV atau berbagai sistem open slam (orb slam, lsd slam, dll.) adalah sistem yang hebat, tetapi tidak memberikan tingkat kinerja optimal yang sama. berfokus pada sistem yang dikembangkan di rumah. Sampai saat ini, tidak ada perusahaan platform AR yang saya tahu sedang menjalankan atau mempertimbangkan untuk menjalankan sistem open slam, meskipun banyak teknik algoritme serupa diterapkan dalam sistem berpemilik yang dioptimalkan.

Ini tidak berarti saya percaya platform terbuka tidak memiliki tempat di cloud AR. Sebaliknya, saya pikir akan ada banyak layanan yang akan mendapat manfaat dari pendekatan terbuka. Namun, saya tidak berpikir bahwa sebagai industri kami memahami masalah AR skala

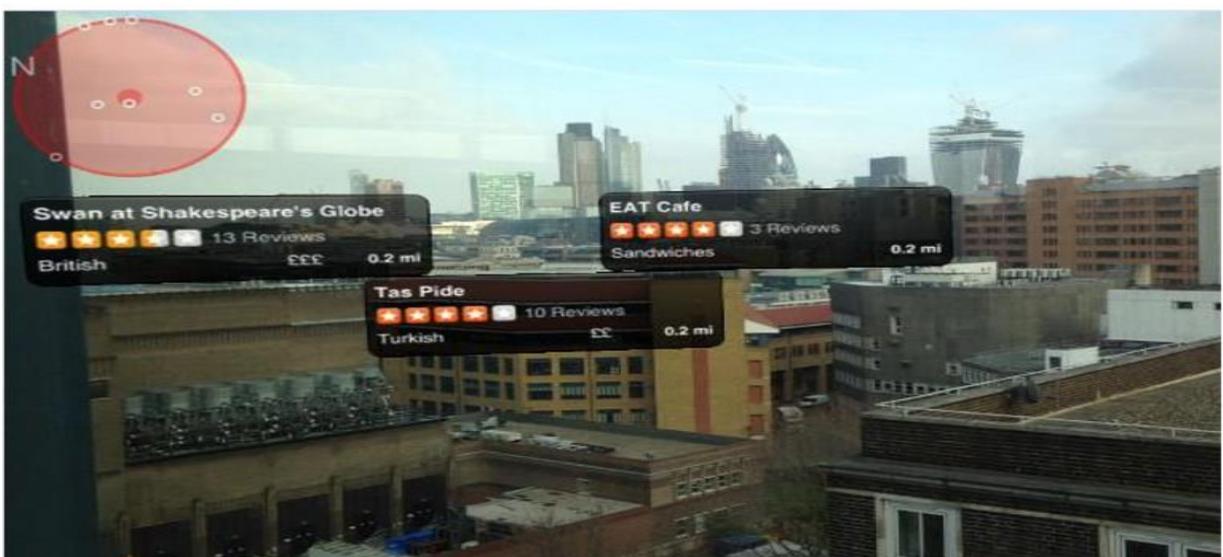
besar dengan cukup baik, namun secara khusus mengatakan bahwa sistem ini harus terbuka versus sistem itu perlu dioptimalkan mungkin.

Relokasi = Multipemain; Ini Juga Penting untuk...

Bagian ini membahas mengapa multipemain sulit diterapkan untuk AR. Pada bagian sebelumnya, kami menyinggung beberapa isu, salah satunya adalah tantangan untuk membuat relokasi yang berkelas konsumen. Seperti yang telah kita bahas, ada aspek lain yang akan sulit untuk dibangun, tetapi semuanya adalah masalah yang diselesaikan sebelumnya. Tapi relokasi yang benar-benar penting, dan lebih dari sekadar multipemain. Berikut adalah beberapa masalah yang harus kita atasi:

"Mulai dingin"

Ini mengacu pada pertama kali Anda meluncurkan aplikasi atau menyalakan HMD Anda, dan perangkat harus mencari tahu di mana itu. Umumnya, sistem saat ini bahkan tidak repot-repot untuk mencoba menyelesaikan ini, mereka hanya menelepon dari mana pun mereka memulai (0,0,0). Mobil otonom, rudal jelajah, dan sistem lain yang perlu melacak lokasi mereka jelas tidak dapat melakukan ini, tetapi mereka memiliki banyak sensor tambahan untuk diandalkan. Memiliki sistem AR yang direlokasi sebagai hal pertama yang dilakukannya berarti aplikasi AR yang persisten dapat dibangun karena sistem koordinat akan konsisten dari sesi ke sesi. Jika Anda menjatuhkan Pokemon di beberapa koordinat tertentu kemarin, saat Anda memindahkan lokasi keesokan harinya setelah menyalakan perangkat, koordinat tersebut akan tetap digunakan hari ini dan Pokemon akan tetap ada di sana. Perhatikan bahwa koordinat ini bisa unik untuk sistem Anda, dan belum tentu koordinat global mutlak (lintang dan bujur) yang dibagikan oleh orang lain (kecuali kita semua melokalisasi ke dalam sistem koordinat global yang sama, yang pada akhirnya akan berakhir)



Gambar 5-17. Inilah yang Anda dapatkan ketika Anda tidak memiliki koordinat absolut yang akurat (atau jaringan 3D kota)

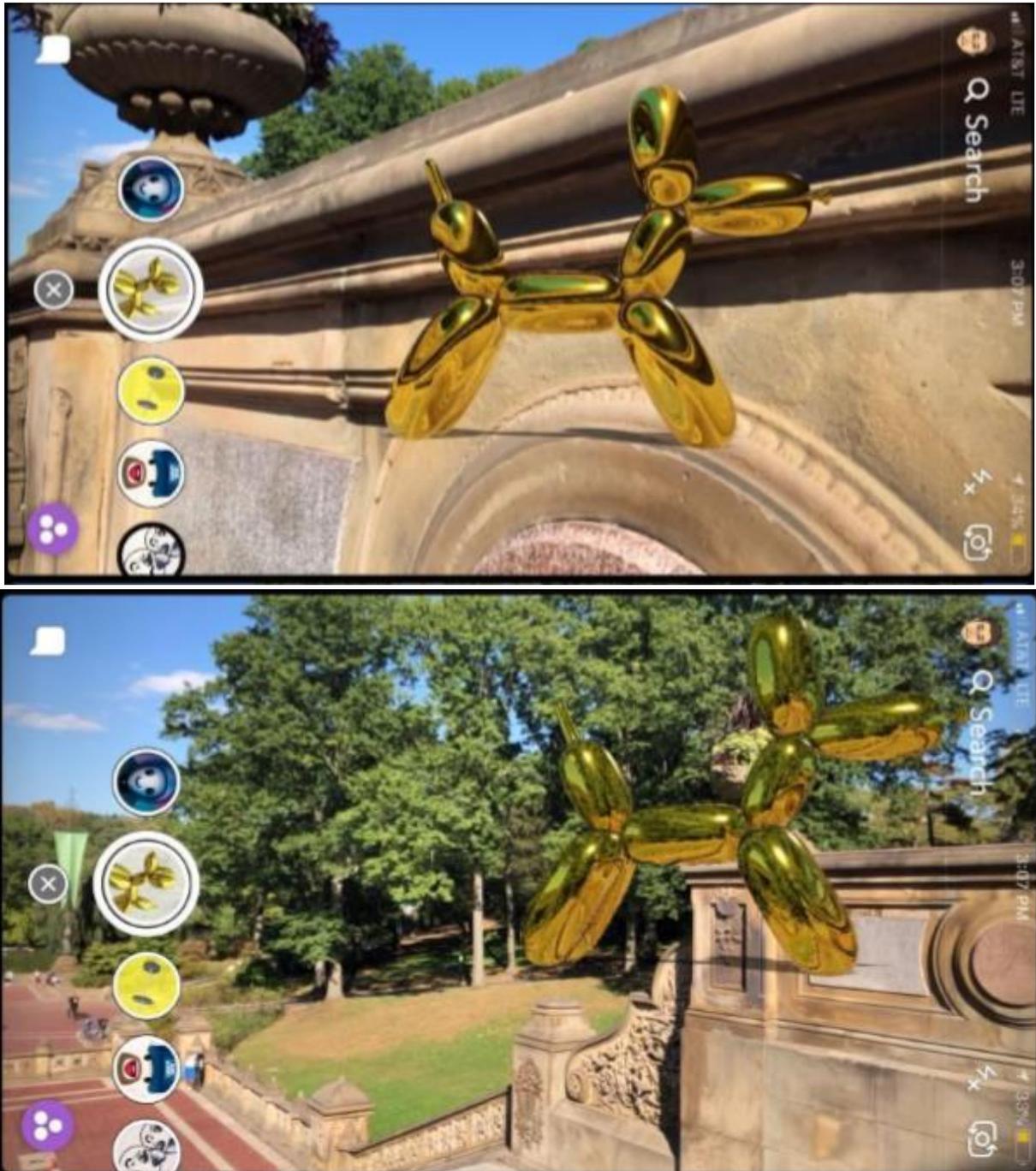
Koordinat mutlak

Ini mengacu pada menemukan koordinat Anda dalam hal garis lintang dan garis bujur ke tingkat akurasi "yang dapat digunakan AR", yang berarti akurat hingga tingkat "subpiksel". Subpiksel berarti koordinatnya cukup akurat sehingga konten virtual akan digambar menggunakan piksel yang sama di perangkat saya seperti di perangkat Anda jika berada di tempat fisik yang sama persis. Biasanya subpiksel digunakan untuk tracking untuk merujuk ke jitter/judder sehingga pose menjadi subpiksel yang akurat berarti konten tidak jitter saat perangkat diam, karena posenya bervariasi. Ini juga merupakan angka yang tidak memiliki ekuivalen metrik, karena setiap piksel dapat menyesuaikan dengan jarak fisik yang sedikit berbeda tergantung pada resolusi perangkat (ukuran piksel) dan juga seberapa jauh perangkat menunjuk (piksel mencakup lebih banyak ruang fisik jika Anda melihat jauh). Dalam praktiknya, akurasi subpiksel tidak diperlukan karena pengguna tidak dapat membedakan apakah konten tidak konsisten beberapa sentimeter antara perangkat saya dan perangkat Anda. Mendapatkan koordinat lintang dan bujur yang akurat sangat penting untuk layanan perdagangan berbasis lokasi (misalnya, tanda virtual di atas pintu harus berada di atas gedung yang tepat, seperti yang diilustrasikan pada Gambar 5-17) serta navigasi.

Tracking hilang

Gerakan cara terakhir yang penting adalah bahwa itu merupakan bagian integral dari pelacak. Mudah-mudahan pelacak tidak akan "kehilangan jejak", tetapi pelacak terbaik pun dapat menemukan sudut jatuh yang membingungkan sensor. Misalnya, menaiki kendaraan yang bergerak dapat membingungkan IMU dari sistem VIO, dan dinding kosong dapat membingungkan sistem kamera. Jika pelacakan hilang, sistem harus kembali dan memposisikan ulang input sensor saat ini terhadap peta SLAM untuk menjaga konsistensi semua konten dalam sesi aplikasi saat ini. Jika pelacakan tidak dapat dipulihkan, koordinat akan kembali ke (0,0,0) dan semua konten juga akan diatur ulang..

M



Gambar 5-18. Inilah yang terjadi jika Anda hanya mengandalkan GPS untuk relokasi—kami tidak melihat objek yang “seharusnya” berada, dan kami bahkan tidak melihatnya di tempat yang sama pada dua perangkat berbeda

Bagaimana Relokalisasi Benar-Benar Dilakukan Hari Ini di Aplikasi?

Jawaban cepat? buruk! Secara garis besar, ada lima cara relokasi dilakukan hari ini untuk sistem tracking luar-dalam (mudah untuk luar-dalam, seperti HTC Vive karena kotak mercusuar eksternal memberikan koordinat umum ke semua perangkat yang mereka lacak). Berikut adalah deskripsi masing-masing:

Kekuatan Augmented dan Virtual Reality dalam Bisnis (Dr. Agus Wibowo)

- Andalkan GPS untuk kedua perangkat dan cukup gunakan garis lintang dan garis bujur sebagai sistem koordinat umum. Ini sederhana, tetapi objek umum yang ingin kita lihat akan ditempatkan di lokasi fisik yang berbeda untuk setiap telepon, seperti yang ditunjukkan pada Gambar 5-18, hingga jumlah kesalahan di lokasi GPS (berapa meter!). Beginilah cara Pokémon Go saat ini mendukung multipemain, tetapi karena backend MMO masih cukup sederhana, ini sebenarnya lebih dekat dengan "beberapa orang memainkan game pemain tunggal yang sama di lokasi yang sama." Ini tidak sepenuhnya akurat, karena begitu Pokémon ditangkap, orang lain tidak dapat menangkapnya, jadi ada beberapa manajemen keadaan sederhana yang terjadi.
- Andalkan gambar penanda tracking fisik umum (atau kode QR). Ini berarti bahwa kami berdua mengarahkan ponsel kami ke penanda di atas meja di depan kami, seperti yang digambarkan pada Gambar 5-19, dan kedua aplikasi kami memperlakukan penanda sebagai koordinat asal (0,0,0). Ini berarti dunia nyata dan dunia maya konsisten di kedua ponsel. Ini bekerja cukup baik, hanya saja tidak ada yang akan membawa penanda, jadi ini adalah jalan buntu untuk penggunaan di dunia nyata.
- Salin peta SLAM antar perangkat dan minta pengguna untuk berdiri di samping satu sama lain dan kemudian minta Pemain 2 memegang telepon mereka sangat dekat dengan Pemain 1. Secara teknis ini bisa bekerja dengan baik; namun, UX hanyalah masalah utama yang harus diatasi pengguna. Beginilah cara kami melakukannya di Dekko untuk Tabletop Speed.
- Tebak saja. Jika saya memulai aplikasi ARKit saya di tempat tertentu, aplikasi saya akan menempatkan asal di koordinat awal. Anda dapat datang nanti dan memulai aplikasi Anda berdiri di tempat yang sama, dan hanya berharap bahwa di mana pun sistem menetapkan Asal Anda kira-kira di tempat fisik yang sama dengan asal saya. Secara teknis ini jauh lebih sederhana daripada menyalin peta SLAM, dan rintangan UX hampir sama, dan kesalahan di seluruh sistem koordinat kami tidak terlalu terlihat jika desain aplikasi tidak terlalu sensitif. Anda hanya perlu mengandalkan pengguna yang melakukan hal yang benar.
- Batasi UX multipemain untuk menerima lokasi dengan akurasi rendah dan interaksi asinkron. Game jenis pencarian harta karun dan AR termasuk dalam kategori ini. Mencapai interaksi real-time akurasi tinggi adalah tantangannya. Saya percaya akan selalu ada kasus penggunaan hebat yang mengandalkan interaksi multipengguna asinkron, dan itu adalah tugas desainer AR UX untuk mengungkapkannya.

Perlu dicatat bahwa kelima solusi ini telah ada selama bertahun-tahun, namun jumlah aplikasi multipemain waktu nyata yang digunakan orang hampir nol. Menurut pendapat saya, semua solusi jatuh ke dalam ember seorang insinyur yang dapat mengatakan, "Lihat itu berhasil, kami melakukan multipemain!" tetapi pengguna akhir merasa terlalu merepotkan karena terlalu sedikit manfaatnya.

5.8 PLATFORM

Membangun aplikasi AR berarti memilih Platform AR untuk dibangun. Platform ini adalah seperangkat API dan alat untuk memungkinkan pengembang membuat konten yang berinteraksi dengan dunia nyata. Dua yang paling banyak tersedia adalah ARKit Apple dan ARCore Google (yang berevolusi dari proyek sebelumnya dari Google yang disebut Tango yang merupakan perangkat lunak dan perangkat keras telepon khusus). Microsoft HoloLens dan Magic Leap keduanya membuat platform pengembang AR untuk perangkat keras Head Mounted Display pelanggan mereka. Bagian selanjutnya ini membahas fitur utama ARCore dan ARKit dan membandingkannya dari sudut pandang pengembang.



Gambar 5-19. Aplikasi ini menggunakan gambar cetak yang digunakan semua perangkat untuk relokasi untuk membagikan koordinatnya

Apple

ARKit Secara khusus, ARKit adalah sistem VIO, dengan beberapa deteksi pesawat 2D sederhana. VIO melacak posisi relatif perangkat Anda di luar angkasa (pose 6DOF Anda) secara real time; yaitu, pose Anda dihitung ulang di antara setiap penyegaran bingkai di layar Anda, sekitar 30 kali atau lebih per detik. Perhitungan ini dilakukan dua kali, secara paralel. Pose Anda dilacak melalui sistem visual (kamera) dengan mencocokkan titik di dunia nyata dengan piksel pada sensor kamera setiap bingkai. Pose Anda juga dilacak oleh sistem inersia (akselerometer dan giroskop Anda — IMU).

Output dari kedua sistem tersebut kemudian digabungkan melalui filter Kalman yang menentukan mana dari dua sistem yang memberikan perkiraan terbaik dari posisi "nyata" Anda (kebenaran dasar) dan menerbitkan pembaruan tersebut melalui ARKit SDK. Sama

seperti odometer Anda di mobil Anda melacak jarak yang telah ditempuh mobil, sistem VIO melacak jarak yang telah ditempuh iPhone Anda dalam ruang 6D. 6D berarti 3D gerak XYZ (terjemahan), ditambah 3D pitch/yaw/roll (rotasi).

Keuntungan besar yang dibawa VIO adalah pembacaan IMU dilakukan sekitar 1.000 kali per detik dan didasarkan pada akselerasi (gerakan pengguna). Perhitungan mati digunakan untuk mengukur pergerakan perangkat antara pembacaan IMU. Perhitungan mati cukup banyak menebak, sama seperti jika saya meminta Anda untuk mengambil langkah dan memperkirakan berapa inci langkah itu. Kesalahan dalam sistem inersia terakumulasi dari waktu ke waktu, sehingga semakin banyak waktu antara frame IMU atau semakin lama sistem inersia berjalan tanpa mendapatkan "reset" dari sistem visual, semakin banyak tracking akan menyimpang dari kebenaran dasar.

Pengukuran visual/optik dibuat pada kecepatan bingkai kamera, jadi biasanya 30 bingkai per detik, dan didasarkan pada jarak (perubahan pemandangan di antara bingkai). Sistem optik biasanya mengakumulasi kesalahan pada jarak (dan waktu pada tingkat yang lebih rendah), sehingga semakin jauh Anda melakukan perjalanan, semakin besar kesalahannya. Kabar baiknya adalah bahwa kekuatan masing-masing sistem membatalkan kelemahan yang lain.

ARKit - Apa itu?

- Odometri Inersia Visual - melacak ke mana saya bergerak saat saya bergerak
- Deteksi Pesawat Sederhana
- Pengembang APIs
- Apakah ini sama dengan SLAM?
- Bagaimana Anda melakukan 3D dari satu lensa?
- Bagaimana Anda mendapatkan skala Metrik?
- Accurate Pose adalah fondasi untuk ALL AR dibangun



Gambar 5-20. ARKit Apple Apple

Jadi, sistem tracking visual dan inersia didasarkan pada sistem pengukuran yang sama sekali berbeda tanpa saling ketergantungan. Ini berarti bahwa kamera dapat ditutupi atau mungkin melihat pemandangan dengan sedikit fitur optik (seperti dinding putih) dan sistem inersia dapat "membawa beban" untuk beberapa bingkai. Atau, perangkat bisa diam dan sistem visual dapat memberikan pose yang lebih stabil daripada sistem inersia. Filter Kalman terus-menerus memilih pose kualitas terbaik, dan hasilnya adalah tracking yang stabil.

Sejauh ini, sangat bagus, tetapi yang menarik adalah bahwa sistem VIO telah ada selama bertahun-tahun, dipahami dengan baik di industri, dan ada beberapa implementasi yang sudah ada di pasar. Jadi, fakta bahwa Apple menggunakan VIO tidak berarti banyak. Kita perlu melihat mengapa sistemnya begitu kuat.

Bagian utama kedua dari ARKit adalah deteksi pesawat sederhana. Ini diperlukan agar Anda memiliki "dasar" untuk menempatkan konten Anda; jika tidak, konten itu akan terlihat seperti mengambang mengerikan di luar angkasa. Ini dihitung dari fitur yang terdeteksi oleh sistem optik (titik-titik kecil, atau titik-awan, yang Anda lihat dalam demonstrasi) dan algoritme hanya meratakannya karena tiga titik menentukan bidang. Jika Anda melakukan ini cukup lama, Anda dapat memperkirakan di mana lokasi sebenarnya. Titik-titik ini membentuk cloud titik yang jarang, yang telah kita bahas sebelumnya di bab ini, yang digunakan untuk tracking optik. cloud titik yang jarang menggunakan lebih sedikit memori dan waktu CPU untuk dilacak, dan dengan dukungan sistem inersia, sistem optik dapat bekerja dengan baik dengan sejumlah kecil titik untuk dilacak. Ini adalah jenis cloud titik yang berbeda dengan cloud titik padat, yang dapat terlihat dekat dengan fotorealisme (beberapa pelacak yang sedang diteliti dapat menggunakan cloud titik padat untuk tracking, sehingga lebih membingungkan).

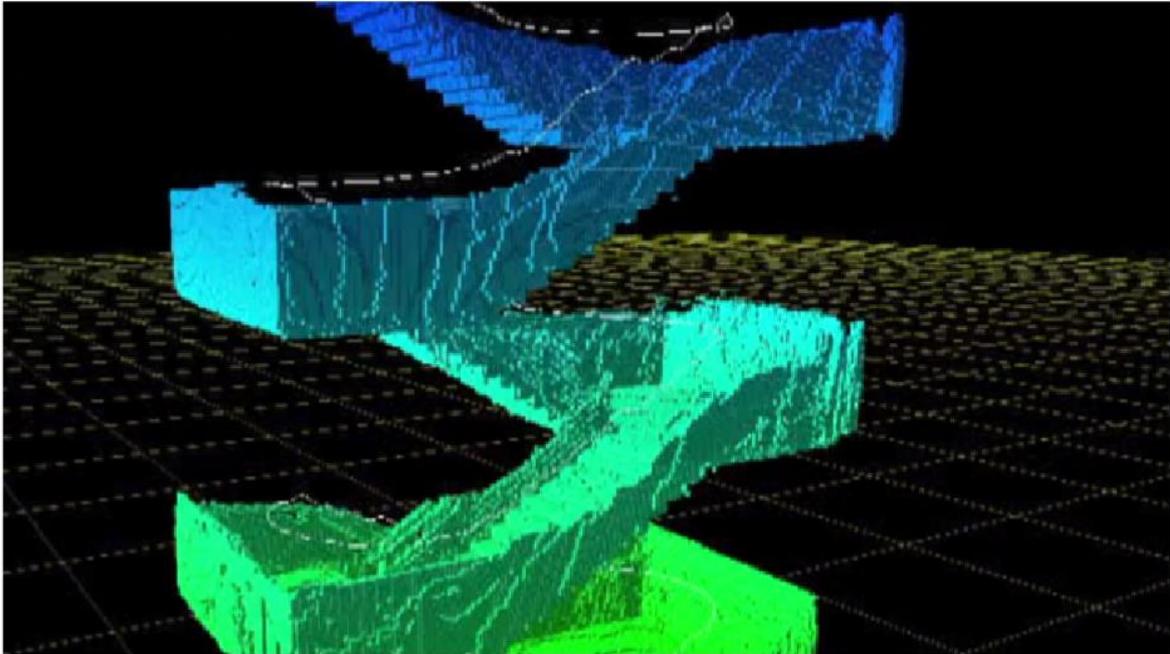
Beberapa Misteri Dijelaskan

Dua misteri ARKit adalah: "Bagaimana Anda mendapatkan 3D dari satu lensa?" dan "Bagaimana Anda mendapatkan skala metrik (seperti dalam demonstrasi pita pengukur itu)?" Rahasiannya di sini adalah memiliki penghapusan kesalahan IMU yang sangat bagus (yaitu, membuat tebakan perhitungan mati menjadi sangat akurat). Ketika Anda bisa melakukannya, inilah yang terjadi:

Untuk mendapatkan 3D dari satu lensa, Anda harus memiliki dua tampilan pemandangan dari tempat yang berbeda, yang memungkinkan Anda melakukan perhitungan stereoskopik posisi Anda. Ini mirip dengan cara mata kita melihat dalam 3D dan mengapa beberapa pelacak mengandalkan kamera stereo. Sangat mudah untuk menghitung jika Anda memiliki dua kamera karena Anda mengetahui jarak antara keduanya dan bingkai yang diambil pada saat yang sama. Untuk menghitung ini hanya dengan satu kamera, Anda perlu mengambil satu bingkai, lalu bergerak, lalu menangkap bingkai kedua. Dengan menggunakan perhitungan mati IMU, Anda dapat menghitung jarak yang dipindahkan antara dua bingkai dan kemudian melakukan perhitungan stereo seperti biasa (dalam praktiknya, Anda mungkin melakukan perhitungan dari lebih dari dua bingkai untuk mendapatkan akurasi yang lebih). Jika IMU cukup akurat, "gerakan" antara dua bingkai ini terdeteksi hanya dengan gerakan otot kecil yang Anda lakukan saat mencoba menahan tangan Anda! Jadi itu terlihat seperti sihir.

Untuk mendapatkan skala metrik, sistem juga mengandalkan perhitungan mati yang akurat dari IMU. Dari pengukuran akselerasi dan waktu yang disediakan IMU, Anda dapat mengintegrasikan mundur untuk menghitung kecepatan dan mengintegrasikan kembali untuk mendapatkan jarak yang ditempuh antara bingkai IMU. Matematikanya tidak sulit. Yang sulit adalah menghilangkan kesalahan dari IMU untuk mendapatkan pengukuran akselerasi yang hampir sempurna. Kesalahan kecil, yang terakumulasi 1.000 kali per detik selama beberapa detik yang diperlukan bagi Anda untuk memindahkan telepon, dapat berarti kesalahan skala

metrik 30% atau lebih. Fakta bahwa Apple telah mengerjakan ini hingga kesalahan satu digit persen sangat mengesankan.



Gambar 5-21. Tango mulai sebagian besar berfokus pada tracking gerakan ponsel dalam ruang 3D

Bukankah ARCore Hanya Tango-Lite?

Salah satu pengembang yang saya ajak bicara sekitar waktu peluncuran ARCore dengan bercanda mengatakan, "Saya baru saja melihat ARCore SDK, dan mereka benar-benar mengganti nama Tango SDK, mengomentari kode kamera kedalaman dan mengubah flag compiler." Saya menduga itu sedikit lebih dari itu, tetapi tidak lebih (ini bukan hal yang buruk!). Misalnya, browser web baru yang mendukung ARCore sangat bagus untuk pengembang, tetapi terpisah dari SDK inti. Dalam posting ARKit saya baru-baru ini, saya bertanya-tanya mengapa Google tidak merilis versi Tango VIO (yang tidak memerlukan kamera kedalaman) 12 bulan yang lalu, mengingat mereka memiliki semua bagian yang siap untuk digunakan. Sekarang mereka punya!

Ini adalah berita bagus, karena ini berarti bahwa ARCore adalah perangkat lunak yang sangat matang dan teruji dengan baik (setidaknya memiliki pengembangan dua tahun lebih banyak di Google daripada yang dimiliki ARKit di Apple — meskipun membeli Metaio dan Flyby membantu Apple mengejar ketinggalan), dan ada banyak peta jalan fitur yang dibariskan untuk Tango, yang tidak semuanya bergantung pada data kedalaman 3D, yang sekarang akan menemukan jalannya ke ARCore.

Mengesampingkan penamaan, jika Anda menambahkan perangkat keras sensor kamera kedalaman ke telepon yang menjalankan ARCore, Anda akan memiliki telepon Tango. Sekarang Google memiliki jalan yang lebih mudah untuk mendapatkan adopsi SDK secara luas dengan dapat mengirimkannya ke ponsel unggulan OEM. Tidak ada yang akan menyerahkan ponsel Android yang bagus untuk yang lebih buruk dengan AR (sama seperti tidak ada yang

akan menyerahkan ponsel hebat apa pun untuk ponsel Windows dengan AR, jadi Microsoft tidak repot; itu langsung ke HMD). Sekarang orang akan membeli telepon yang akan mereka beli, dan ARCore akan ditarik secara gratis. Banyak ide orisinal ditujukan untuk pemetaan dalam ruangan. Baru kemudian AR dan VR menjadi kasus penggunaan paling populer.

Jika kita mempertimbangkan namanya, saya pikir itu menarik bahwa Tango selalu digambarkan seperti “telepon yang selalu tahu lokasinya” (Gambar 5-21). Saya belum pernah bertemu satu orang pun yang terkesan dengan itu. Bagi saya, itu memosisikan ponsel sebagai sesuatu yang lebih selaras dengan Google Maps, dan AR adalah renungan (apakah itu cara Google melihatnya masih bisa diperdebatkan). Dengan nama baru, semuanya AR, sepanjang waktu, seperti yang ditunjukkan pada Gambar 5-22.



Gambar 5-22. Google ARCore adalah evolusi Tango tanpa perangkat keras kamera kedalaman

Jadi, Haruskah Saya Membangun di ARCore Sekarang?

Jika Anda menyukai Android dan memiliki S8 atau Pixel, jawabannya adalah ya. Lakukan itu. Jika Anda menyukai iPhone, jangan repot-repot menggantinya. Hal yang harus menjadi fokus pengembang adalah bahwa membangun aplikasi AR yang disukai orang sangat menantang. Akan jauh lebih sedikit upaya untuk mempelajari cara membangun ARKit atau ARCore daripada upaya mempelajari apa yang harus dibangun. Ingat juga bahwa ARKit/ARCore SDK adalah versi 1.0. Mereka benar-benar dasar (VIO, deteksi pesawat, pencahayaan dasar) dan akan menjadi fitur yang jauh lebih lengkap selama beberapa tahun ke depan (pemahaman Scene 3D, oklusi, multipemain, persistensi konten, dll.). Ini akan menjadi kurva pembelajaran yang konstan bagi pengembang dan konsumen. Namun untuk saat ini, fokuslah untuk mempelajari apa yang sulit (aplikasi apa yang akan dibuat) dan tetap berpegang pada apa yang Anda ketahui untuk teknologi yang mendasarinya (cara membuatnya: Android, IOS Xcode, dll.). Setelah Anda memahami apa yang membuat aplikasi bagus, buat keputusan tentang platform terbaik untuk diluncurkan sehubungan dengan jangkauan pasar, dukungan fitur AR, monetisasi, dan sebagainya.

Bagaimana dengan Tango, Hololens, Vuforia, dan Lainnya?

Jadi, Tango adalah merek (sudah dikalahkan oleh Google), bukan produk. Ini terdiri dari desain referensi perangkat keras (RGB, fisheye, kamera kedalaman, dan beberapa spesifikasi CPU/GPU) dan tumpukan perangkat lunak yang menyediakan VIO (tracking gerak), pemetaan jarang (pembelajaran area), dan rekonstruksi 3D padat (persepsi kedalaman).

Hololens (dan Magic Leap) memiliki tumpukan perangkat lunak yang persis sama, tetapi mencakup beberapa chip pemrosesan sinyal digital dasar (DSP), yang mereka sebut sebagai Unit Pemrosesan Holografik, untuk membongkar pemrosesan dari CPU/GPU dan menghemat daya. Desain chip yang lebih baru dari Qualcomm akan memiliki fungsi ini di dalamnya, menghilangkan kebutuhan untuk pemrograman DSP khusus dan mengurangi biaya perangkat keras di masa mendatang. Vuforia hampir sama lagi, tetapi perangkat kerasnya independen.

Masing-masing menggunakan jenis sistem VIO yang sama. Baik Hololens, Magic Leap, maupun Tango tidak menggunakan kamera kedalaman untuk tracking (meskipun saya yakin mereka mulai mengintegrasikannya untuk membantu dalam beberapa kasus sudut). Jadi mengapa ARKit begitu bagus? Jawabannya adalah bahwa ARKit tidak benar-benar lebih baik daripada Hololens, tetapi perangkat keras Hololens tidak tersedia secara luas.

Jadi, pada akhirnya, alasan ARKit lebih baik adalah karena Apple mampu melakukan pekerjaan untuk memasang algoritma VIO dengan erat ke sensor dan menghabiskan banyak waktu untuk mengkalibrasinya untuk menghilangkan kesalahan dan ketidakpastian dalam perhitungan pose.

Perlu dicatat bahwa ada banyak alternatif untuk sistem OEM besar. Ada banyak pelacak akademis (misalnya, ORB Slam bagus dan OpenCV memiliki beberapa opsi) tetapi hampir semuanya hanya optik (mono RGB, atau stereo, dan/atau berbasis kamera kedalaman; beberapa menggunakan peta jarang, beberapa padat, beberapa peta kedalaman, dan yang lainnya menggunakan data semi-langsung dari sensor—ada banyak cara untuk menguliti kucing ini. Ada sejumlah startup yang bekerja pada sistem tracking. Augmented Pixels memiliki satu yang berkinerja baik, tetapi pada akhirnya hari, setiap sistem VIO membutuhkan pemodelan dan kalibrasi perangkat keras untuk bersaing.

5.9 PERTIMBANGAN PENGEMBANGAN LAINNYA

Ingatlah pencahayaan, fitur multipemain, dan koneksi ke pengguna lain dan dunia nyata saat mengembangkan.

Lighting

Baik ARKit dan ARCore memberikan perkiraan sederhana dari pencahayaan alami di tempat kejadian, seperti yang ditunjukkan pada Gambar 5-23. Ini adalah salah satu perkiraan untuk pemandangan, terlepas dari apakah dunia nyata diterangi dengan mulus dengan cahaya sekitar atau penuh dengan lampu sorot yang tajam. ARKit mengembalikan kontrol intensitas dan suhu warna ke pengembang, sedangkan ARCore memberikan nilai intensitas piksel tunggal (Android Studio API) atau shader (Unity API).

Kedua pendekatan tersebut tampak dari demonstrasi awal memberikan hasil yang serupa. Secara subyektif, demonstrasi Google terlihat sedikit lebih baik bagi saya, tetapi itu mungkin karena pengembang Tango telah mengerjakannya lebih lama daripada ARKit dirilis. Namun, Google telah menunjukkan apa yang akan segera hadir (17:11 dalam video ini), yaitu kemampuan untuk menyesuaikan bayangan dan pantulan virtual secara dinamis dengan pergerakan cahaya dunia nyata. Ini akan memberikan dorongan besar di mana kami secara tidak sadar percaya bahwa kontennya "benar-benar ada."

Multiplayer AR

Sebelumnya dalam bab ini, kami telah memeriksa apa yang membuat aplikasi AR smartphone hebat dan mengapa ARKit dan ARCore telah memecahkan masalah teknis yang sangat sulit (tracking luar-dalam 6DOF yang kuat) dan menciptakan platform untuk AR untuk akhirnya mencapai penggunaan umum (masih beberapa tahun). jauh untuk adopsi luas, tetapi banyak ceruk besar untuk aplikasi hari ini IMO). Pengembang sekarang bekerja untuk meningkatkan kurva belajar dari aplikasi kentut ke aplikasi yang berguna (meskipun putra saya yang berusia sembilan tahun menganggap aplikasi kentut cukup berguna, terima kasih). Satu-satunya fitur yang saya tanyakan lebih banyak orang daripada yang lain adalah multipemain. Istilah "multipemain" sebenarnya keliru, karena yang kami maksud adalah kemampuan untuk berbagi pengalaman AR Anda dengan orang lain, atau banyak orang lain, secara real time. Jadi, menyebutnya "multipengguna", "Berbagi AR", "AR Sosial", dan "Komunikasi AR" adalah istilah yang sama baiknya, tetapi multipemain tampaknya tetap berlaku sekarang, mungkin karena sebagian besar alat AR 3D berasal dari latar belakang game, dan itulah istilah yang digunakan gamer. Perhatikan bahwa Anda dapat melakukan multipemain secara asinkron, tetapi itu seperti bermain catur dengan sahabat pena. Selain itu, saya tidak sabar menunggu alat yang lebih baru untuk datang ke AR yang selaras dengan alur kerja dari disiplin desain yang lebih tradisional (arsitek, desainer produk, desainer UX, dll.) karena saya pikir itu akan mendorong dorongan besar untuk utilitas aplikasi AR. Tapi itu untuk buku lain.



Gambar 5-23. ARCore dan ARKit memberikan perkiraan (sederhana) real-time dari cahaya di tempat kejadian

Sehingga pengembang dapat langsung menyesuaikan pencahayaan simulasi agar sesuai dengan dunia nyata (dan mungkin memicu animasi pada saat yang sama)

Saya pribadi percaya bahwa AR tidak akan benar-benar memengaruhi semua kehidupan kita sehari-hari sampai AR memungkinkan kita berkomunikasi dan berbagi dengan cara baru dan menarik yang belum pernah mungkin dilakukan sebelumnya. Jenis komunikasi ini membutuhkan multipemain waktu nyata. Secara pribadi, saya pikir istilah multiplayer game-centric membatasi pemikiran kita tentang betapa pentingnya kemampuan ini sebenarnya. Multiplayer AR telah dimungkinkan selama bertahun-tahun (kami membangun aplikasi AR multipemain di Dekko pada 2011), tetapi relokasi UX selalu menjadi kendala besar.

Jadi, jika multipemain adalah fitur utama yang diminta orang, mengapa kami tidak memilikinya? Jawabannya, seperti banyak fungsi AR, berarti menyelami teknologi visi komputer yang memungkinkan AR. (Kami juga memerlukan jaringan latensi rendah, pemeliharaan model dunia yang konsisten, berbagi audio dan video, dan metafora interaksi kolaboratif, juga, tetapi bagian ini berfokus pada tantangan visi komputer, yang belum benar-benar terpecahkan.) Multiplayer AR hari ini agak seperti tracking posisi 6DOF beberapa tahun yang lalu. Ini tidak terlalu sulit untuk dilakukan dengan cara yang kasar, tetapi rintangan UX yang dihasilkan terlalu tinggi bagi konsumen. Mendapatkan UX multipemain tingkat konsumen ternyata menjadi masalah teknis yang sulit. Ada banyak teknologi yang memungkinkan multipemain, tetapi yang perlu diperhatikan adalah teman lama kita: relokasi. Aspek non-intuitif lainnya dari multipemain adalah bahwa ia memerlukan beberapa infrastruktur di cloud agar berfungsi dengan baik.



Gambar 5-24. Game “The Machines” yang didemonstrasikan Apple pada keynote-nya menggunakan sistem multipemain sederhana yang dikembangkan sendiri (demonstrasi yang bagus, tetapi bukan cloud AR)



Gambar 5-25. Hal semacam ini tidak mungkin terjadi tanpa AR cloud

Bagaimana Orang Terhubung Melalui AR?

Bagaimana kami mendukung banyak pengguna untuk berbagi pengalaman? Bagaimana kita melihat barang virtual yang sama pada saat yang sama, tidak peduli perangkat apa yang kita pegang atau pakai, ketika kita berada di tempat yang sama (atau tidak)? Anda dapat memilih istilah yang sudah dikenal untuk menggambarkan kemampuan ini berdasarkan apa yang sudah Anda ketahui: aplikasi “multipemain” untuk pemain game, atau aplikasi “sosial” atau aplikasi “berkomunikasi”. Ini semua infrastruktur yang sama di bawah tenda dan dibangun di atas teknologi yang memungkinkan yang sama. Lokalisasi yang sangat kuat, streaming pose dan status sistem 6DOF, jahitan mesh 3D, dan pembaruan mesh yang bersumber dari kerumunan adalah semua masalah teknis yang harus diselesaikan di sini. Jangan lupakan tantangan tingkat aplikasi seperti hak akses, autentikasi, dan sebagainya (walaupun sekarang sebagian besar merupakan masalah teknis).

Bagaimana Aplikasi AR Terhubung ke Dunia dan Mengetahui Di Mana Mereka Sebenarnya?

GPS bukanlah solusi yang cukup baik—bahkan GPS yang akan datang yang akurat untuk satu kaki. Bagaimana kita membuat AR bekerja di luar di area yang luas? Bagaimana kita menentukan lokasi kita baik dalam koordinat absolut (lintang dan bujur) dan juga relatif terhadap struktur yang ada hingga presisi subpiksel? Bagaimana kita mencapai ini baik di dalam maupun di luar? Bagaimana kami memastikan konten tetap berada di tempatnya, bahkan sehari-hari atau bertahun-tahun kemudian? Bagaimana kita mengelola begitu banyak data? Lokalisasi terhadap koordinat absolut adalah masalah teknis yang sangat sulit untuk dipecahkan di sini.



Gambar 5-26. Agar ponsel Anda mengetahui hal ini saat Anda berjalan melewatinya sambil menangkap dan mengelola struktur data 3D yang terlibat, diperlukan cloud AR

Bagaimana Aplikasi AR Memahami dan Terhubung ke Berbagai Hal di Dunia Nyata?

Bagaimana aplikasi kami memahami struktur 3D atau geometri dunia (bentuk benda); misalnya, bagaimana Pokémon saya tahu bahwa ia dapat bersembunyi di belakang atau memantul ke dalam struktur seperti kubus besar yang ditampilkan di layar ponsel cerdas saya (Gambar 5-26). Bagaimana cara mengidentifikasi benda-benda itu—bagaimana kucing virtual saya tahu bahwa gumpalan itu sebenarnya adalah sofa, dan dia harus menjauh dari sofa? Rekonstruksi 3D padat pada perangkat waktu nyata, segmentasi Scene 3D waktu nyata, klasifikasi objek 3D, pengisian ulang pemrosesan lokal dengan model yang dilatih cloud adalah tantangan di sini.

Seperti banyak di AR, tidak sulit untuk membangun sesuatu yang menunjukkan dengan baik, tetapi sangat sulit untuk membangun sesuatu yang bekerja dengan baik dalam kondisi dunia nyata. Anda mungkin akan sering mendengar tentang cloud AR dalam beberapa bulan mendatang: jika Anda bingung, itu bukan Anda, melainkan mereka.

Tepat ketika Anda berpikir Anda memahami perbedaan antara AR, VR, dan MR, semuanya menjadi lebih dalam. Vendor akan menggunakan istilah identik yang berarti hal yang sama sekali berbeda, seperti berikut:

Multiplayer AR

Ini bisa merujuk pada cara murni tingkat permainan untuk melacak apa yang dilakukan setiap pemain dalam game itu sendiri tanpa visi komputer atau kesadaran spasial. Atau, itu bisa merujuk pada cara untuk memecahkan beberapa masalah pelokalan visi komputer yang sangat sulit. Atau, kedua hal di atas. Atau mereka bisa berarti sesuatu yang lain sama sekali.

Outdoor AR

Ini mungkin hanya berarti aplikasi ARKit yang memiliki aset konten besar yang terlihat terbaik di luar, atau bisa berarti sesuatu yang mendekati sistem pemetaan 3D kendaraan otonom global.

Rekognisi

Ini mungkin berarti secara manual mengonfigurasi satu penanda atau gambar yang dapat dikenali aplikasi Anda. Atau, itu mungkin berarti mesin klasifikasi objek 3D global yang didukung pembelajaran mesin tujuan umum secara real-time.

5.10 CLOUD AR

Jika Anda memikirkan semua bagian dari aplikasi yang ada di cloud, saya cenderung membagi "cloud" secara horizontal dan memisahkan layanan tersebut menjadi hal-hal yang "baik untuk dimiliki" di bagian atas, dan "harus dimiliki" di bagian bawah (Gambar 5-27). Hal-hal yang baik untuk dimiliki umumnya terkait dengan aplikasi dan konten dan membuatnya mudah untuk membangun dan mengelola aplikasi dan pengguna.

Apa yang Saya Bayangkan Ketika Saya Memikirkan Tentang AR Cloud

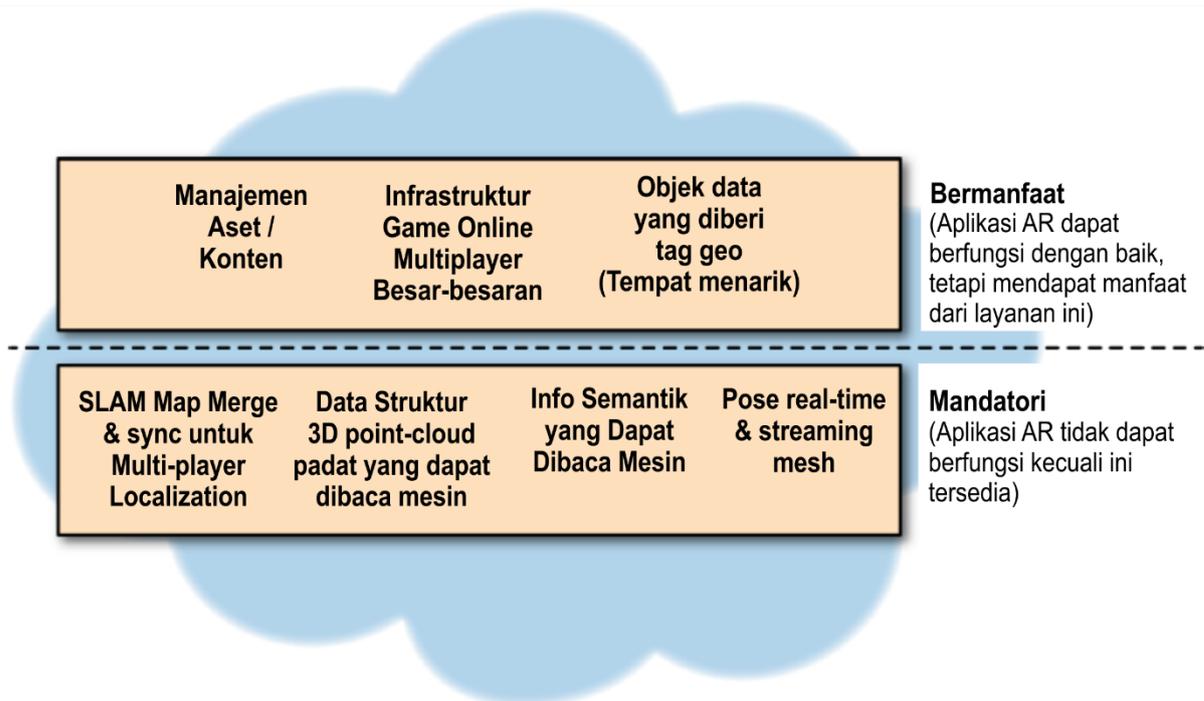
Aplikasi AR Anda hari ini tanpa koneksi cloud AR seperti memiliki ponsel yang hanya dapat memainkan Snake. Bagian bawah cloud, bagi saya, adalah bagian yang menarik. Sistem AR, pada dasarnya, terlalu besar untuk sebuah perangkat. Dunia ini terlalu besar untuk muat di dalamnya, dan itu seperti mencoba memasukkan semua peta google dan web lainnya di ponsel Anda (atau HMD). Insight utamanya adalah jika Anda ingin aplikasi AR Anda dapat berbagi pengalaman atau bekerja dengan baik (yaitu, dengan kesadaran akan dunia 3D tempat ia berada) di lokasi mana pun, aplikasi tersebut bahkan tidak dapat bekerja sama sekali tanpa akses ke layanan cloud ini. Mereka sama pentingnya dengan API sistem operasi yang memungkinkan aplikasi Anda berkomunikasi dengan driver jaringan, atau layar sentuh, atau akses disk.

Sistem AR memerlukan sistem operasi yang sebagian hidup di perangkat, dan sebagian hidup di cloud. Layanan data jaringan dan cloud sama pentingnya dengan aplikasi AR seperti halnya jaringan untuk melakukan panggilan telepon seluler. Pikirkan kembali sebelum smartphone—ponsel Nokia lama Anda tanpa jaringan masih bisa menjadi kalkulator dan Anda bisa memainkan Snake, tetapi kegunaannya sangat terbatas. Jaringan dan cloud AR akan sama pentingnya untuk aplikasi AR. Saya yakin kita akan melihat aplikasi ARKit/ARCore hari ini sebagai setara dengan hanya memiliki "Nokia Snake" offline versus telepon yang terhubung ke jaringan.

Seberapa Besar Kesepakatan AR Cloud?

Jika Anda ditanya apa satu-satunya aset paling berharga di industri teknologi saat ini, Anda mungkin akan menjawab bahwa itu adalah indeks pencarian Google atau grafik sosial Facebook atau mungkin sistem rantai pasokan Amazon. Saya percaya bahwa dalam waktu 15 tahun, akan ada aset lain yang setidaknya sama berharganya dengan yang tidak ada saat ini. Mungkin lebih berharga jika Anda melihatnya dalam konteks aset sistem operasi Microsoft Windows (yang merupakan aset teknologi paling berharga di tahun 1990-an) yang bernilai pada tahun 2017 dibandingkan tahun 1997.

Akankah satu perusahaan akhirnya memiliki (sebagian besar keuntungan) itu? Sejarah mengatakan mungkin. Apakah akan menjadi perusahaan baru? Juga mungkin. Sama seperti pada tahun 1997, tidak terbayangkan jika Microsoft kehilangan posisinya, pada tahun 2019 tampaknya tidak mungkin Google atau Facebook akan kehilangan posisinya. Tapi tidak ada yang dijamin. Saya akan mencoba memaparkan argumen yang mendukung masing-masing dari tiga pihak yang bermain di sini (pemegang saham, startup, web terbuka) di bagian terakhir bab ini.



Gambar 5-27. cloud AR dapat dikelompokkan menjadi dua lapisan: bagian cloud yang bagus untuk membantu aplikasi, dan bagian yang harus dimiliki, yang tanpanya aplikasi tidak akan berfungsi sama sekali

Sebelumnya, kami menjelajahi cara kerja ARKit dan ARCore. Kami membahas apa yang tersedia hari ini dan bagaimana kami sampai di sini. Di bagian yang akan datang, kita melihat apa yang hilang dari ARKit dan ARCore dan bagaimana bagian yang hilang itu akan bekerja.

Jadi, Apa Itu AR Cloud?

Untuk melampaui ARKit dan ARCore, kita perlu mulai berpikir melampaui diri kita sendiri. Bagaimana orang lain di perangkat AR jenis lain bergabung dengan kami dan berkomunikasi dengan kami di AR? Bagaimana cara kerja aplikasi kami di area yang lebih besar dari ruang tamu kami? Bagaimana aplikasi kami memahami dan berinteraksi dengan dunia? Bagaimana kita bisa meninggalkan konten untuk ditemukan dan digunakan orang lain? Untuk memberikan kemampuan ini, kami membutuhkan infrastruktur perangkat lunak berbasis cloud untuk AR.

Awan AR dapat dianggap sebagai model skala 1:1 yang dapat dibaca mesin dari dunia nyata. Perangkat AR kami adalah antarmuka waktu nyata ke dunia virtual paralel ini, yang dihiperkonfirmasi dengan sempurna ke dunia fisik. Menyenangkan, tapi ingat: ini adalah rilis v1.0.

Mengapa Semua "Meh" dari Pers untuk ARKit dan ARCore?

Ketika ARKit diumumkan di WWDC tahun ini, kepala eksekutif Apple Tim Cook menggembar-gemborkan augmented reality, mengatakan kepada para analis, "Ini adalah salah satu hal besar yang akan kita lihat kembali dan kagumi pada awalnya." Beberapa bulan berlalu. Pengembang bekerja dengan rajin pada hal besar berikutnya, tetapi reaksi terhadap ARKit pada keynote peluncuran iPhone adalah, "meh." Mengapa itu? Itu karena ARKit dan ARCore saat ini berada di versi 1.0. Mereka hanya memberi pengembang tiga alat AR yang sangat sederhana:

- Pose 6DOF ponsel, dengan koordinat baru setiap sesi
- Sebuah pesawat tanah parsial dan kecil
- Rata-rata sederhana dari pencahayaan pemandangan

Dalam kegembiraan kami melihat salah satu masalah teknis yang paling sulit dipecahkan (pose 6DOF yang kuat dari sistem VIO yang solid) dan Tim Cook mengucapkan kata "augmented" dan "realitas" bersama di atas panggung, kami mengabaikan bahwa Anda benar-benar tidak dapat membangun apa pun terlalu mengesankan hanya dengan tiga alat itu. Masalah terbesar mereka adalah orang mengharapkan aplikasi luar biasa sebelum set lengkap alat untuk membangunnya ada. Namun, itu bukan jika, tetapi kapan kita salah.

Apa yang Hilang untuk Membuat Aplikasi AR yang Hebat?

Singkatnya, AR-first, mobile second. Clay Bavor menyebut bagian yang hilang dari ekosistem AR sebagai jaringan ikat, yang menurut saya merupakan metafora yang bagus. Dalam posting blog saya tentang desain produk AR, saya menyoroti bahwa satu-satunya alasan keberadaan aplikasi AR (dibandingkan dengan aplikasi smartphone biasa) adalah jika aplikasi tersebut memiliki interaksi atau koneksi dengan dunia nyata—dengan orang, tempat, atau benda fisik.

Agar aplikasi AR benar-benar terhubung ke dunia, ada tiga hal yang harus dapat dilakukan. Tanpa koneksi ini, itu tidak akan pernah benar-benar menjadi AR asli. Kemampuan ini hanya dimungkinkan dengan dukungan cloud AR.

Apakah Cloud Seluler Saat Ini Sesuai dengan Pekerjaan?

Ketika saya bekerja di infrastruktur telekomunikasi, ada sedikit kebenaran seperti zen yang berbunyi, "Tidak ada cloud, itu hanya komputer orang lain." Kami selalu bekerja dengan pasangan tembaga atau untaian serat (atau spektrum radio) yang secara fisik menghubungkan satu komputer ke komputer lain, bahkan di seluruh dunia. Ini bukan sihir, hanya sulit. Apa yang membuat infrastruktur cloud AR berbeda dari cloud saat ini, yang mendukung web dan aplikasi seluler kami, adalah bahwa AR (seperti mobil dan drone dan robot *self-driving*) adalah sistem waktu nyata. Siapa pun yang pernah bekerja di bidang telekomunikasi (atau infrastruktur game MMO *fastswitch*) sangat memahami bahwa infrastruktur real-time dan infrastruktur asinkron adalah dua hal yang sama sekali berbeda.

Jadi, meskipun banyak bagian dari cloud AR akan melibatkan hosting data besar dan melayani API web dan melatih model pembelajaran mesin—seperti halnya cloud saat ini—perlu ada pemikiran ulang yang sangat besar tentang bagaimana kami mendukung aplikasi

waktu nyata dan interaksi AR dalam skala masif. Kasus penggunaan AR dasar seperti streaming model 3D langsung dari kamar kami saat kami "AR Skype"; memperbarui data dan aplikasi yang terhubung ke berbagai hal, disajikan saat saya lewat di angkutan umum; mengalirkan data (grafis kaya) kepada saya yang berubah tergantung ke mana mata saya melihat, atau siapa yang berjalan di dekat saya; dan memelihara dan memperbarui status aplikasi waktu nyata dari setiap orang dan aplikasi dalam kerumunan besar di sebuah konser. Tanpa UX jenis ini, tidak ada gunanya AR. Mari kita tetap menggunakan aplikasi smartphone. Mendukung ini untuk akhirnya miliaran orang akan menjadi peluang besar. Jaringan 5G akan memainkan peran besar dan dirancang hanya untuk kasus penggunaan ini. Jika sejarah adalah panduan, beberapa, jika bukan sebagian besar, pemain lama saat ini yang memiliki investasi besar dalam infrastruktur cloud saat ini tidak akan mencopot investasi tersebut untuk beradaptasi dengan dunia baru ini.

Apakah ARKit (atau ARCore) Tidak Berguna Tanpa AR Cloud?

Pada akhirnya, terserah kepada pengguna aplikasi AR untuk memutuskan ini. "Tidak berguna" adalah pilihan kata yang provokatif. Sejauh ini, satu bulan, berdasarkan metrik awal, pengguna condong ke arah "hampir tidak berguna." Mereka mungkin hal baru yang menyenangkan yang membuat Anda tersenyum ketika Anda membagikannya. Mungkin jika Anda membeli sofa, Anda akan mencobanya terlebih dahulu. Tapi ini bukan aplikasi penting yang digunakan sehari-hari yang menentukan platform baru. Untuk itu, kita membutuhkan aplikasi AR-native. Aplikasi yang benar-benar terhubung dengan dunia nyata. Dan untuk menghubungkan aplikasi AR kami satu sama lain dan dunia, kami membutuhkan infrastruktur untuk melakukannya. Kami membutuhkan cloud AR.

AR Cloud dini

Sejak konferensi WWDC Apple pada tahun 2017, yang meluncurkan senjata awal untuk AR konsumen dengan peluncuran ARKit, kami telah melihat setiap platform besar mengumumkan strategi AR: ARCore Google; platform kamera Facebook; Amazon Sumeria; dan Microsoft terus membangun ekosistem Mixed Realitynya. Kami juga telah melihat ribuan pengembang bereksperimen dengan aplikasi AR tetapi sangat sedikit yang diserap oleh konsumen. Pada bulan September 2017, saya memperkirakan bahwa aplikasi AR akan berjuang untuk interaksi tanpa cloud AR, dan ini memang benar. Namun, kami sekarang menyaksikan fajar layanan cloud yang akan membuka kemampuan menarik untuk pengembang AR, tetapi hanya jika penyedia cloud mendapatkan UX mereka dengan benar. Ini bukan tentang menjadi yang pertama ke pasar, tetapi yang pertama mencapai UX tingkat konsumen.

Ada yang ingat AR sebelum ARKit dan ARCore? Secara teknis berhasil, tetapi UX-nya kikuk. Anda memerlukan spidol yang dicetak atau untuk memegang dan memindahkan telepon dengan hati-hati untuk memulai, dan kemudian itu bekerja dengan cukup baik. Video demonstrasi yang bagus dibuat untuk menunjukkan pengalaman kerja terakhir, yang memukau orang-orang. Hasilnya: penyerapan nol. Memecahkan masalah teknis (walaupun masalah teknis yang cukup sulit) ternyata sangat berbeda dengan mencapai UX yang dapat digunakan konsumen. Baru setelah ARKit diluncurkan, UX "hanya berfungsi" untuk AR dasar

tersedia (dan ini 10 tahun setelah SLAM Seluler ditemukan di Oxford Active Vision Lab, yang dipimpin oleh Victor Prisacariu, salah satu pendiri 6D.ai saya).

Kami memasuki waktu yang sama dengan cloud AR. Istilah ini muncul dalam percakapan September 2017 yang saya lakukan dengan Ori Inbar sebagai cara untuk menggambarkan serangkaian masalah infrastruktur visi komputer yang perlu dipecahkan agar aplikasi AR menjadi menarik. Setelah sejumlah perusahaan rintisan awal melihat nilai dalam istilah (dan, yang lebih penting, nilai dari pemecahan masalah ini), kami sekarang melihat platform AR terbesar mulai mengadopsi bahasa ini sebagai pengakuan atas masalah yang sangat penting. Saya mendengar desas-desus yang kuat bahwa Google tidak akan menjadi perusahaan bernilai miliaran rupiah terakhir yang mengadopsi bahasa cloud AR pada tahun 2018.

Multiplayer AR (dan fitur cloud AR secara umum) memiliki tantangan yang sama dengan 6DOF AR dasar: kecuali UX dipaku, pengembang awal akan bersenang-senang membangun dan membuat video demonstrasi, tetapi pengguna tidak akan diganggu untuk menggunakannya. Saya telah membangun sistem AR multipemain beberapa kali selama 10 tahun terakhir dan bekerja dengan desainer UX di tim saya untuk menguji pengguna aspek SLAM dari UX dengan cukup ekstensif. Tidak sulit untuk mengetahui apa yang dibutuhkan UX untuk disampaikan:

- Ketahuilah bahwa orang tidak akan melompat melalui rintangan. Aplikasi tidak perlu meminta Pemain 2, 3, 4, dan seterusnya untuk "datang dan berdiri di sebelah saya" atau "ketik beberapa info". Sinkronisasi sistem SLAM hanya perlu bekerja dari mana pun pengguna berdiri ketika mereka ingin bergabung; yaitu, dari setiap sudut relatif atau jarak antara pemain.
- Hilangkan atau minimalkan "pra-pemindaian", terutama jika pengguna tidak mengerti mengapa itu diperlukan atau menerima umpan balik yang diberikan tentang apakah mereka melakukannya dengan benar.
- Setelah sistem disinkronkan (yaitu, direlokasi ke dalam set koordinat dunia bersama), konten harus memiliki keselarasan yang akurat. Ini berarti bahwa kedua sistem setuju bahwa titik x,y,z maya yang sama sama persis dengan titik yang sama di dunia nyata. Secara umum, perbedaan beberapa sentimeter di antara perangkat dapat diterima dalam hal persepsi pengguna. Namun, ketika (akhirnya) oklusi mesh dibagikan, kesalahan penyelarasan apa pun sangat terlihat karena konten "terpotong" tepat sebelum melewati di belakang objek fisik. Penting untuk dicatat bahwa pelacak ARCore dan ARKit yang mendasari akurat hanya sekitar tiga hingga lima sentimeter, jadi mendapatkan penyelarasan yang lebih baik daripada yang saat ini tidak mungkin dilakukan untuk sistem relocalizer multipemain.
- Pengguna tidak perlu menunggu. Sinkronisasi sistem koordinat harus instan dan tidak ada klik. Idealnya, instan berarti sepersekian detik, tetapi seperti yang akan dikatakan oleh perancang aplikasi seluler mana pun, pengguna akan bersabar hingga dua hingga tiga detik sebelum merasa sistemnya terlalu lambat.

- Pengalaman multipemain harus bekerja lintas platform, dan UX harus konsisten di seluruh perangkat.
- Penatagunaan data itu penting. Stewardship mengacu pada “pengelolaan yang hati-hati dan bertanggung jawab atas sesuatu yang dipercayakan kepada seseorang,” dan ini adalah kata yang kami gunakan di 6D.ai ketika kami memikirkan data cloud AR. Pengguna mempercayakannya kepada kami. Ini adalah masalah yang berkembang karena orang mulai memahami bahwa data mereka yang disimpan dapat digunakan untuk hal-hal yang tidak dijelaskan di muka atau dapat diretas dan digunakan secara kriminal. Namun, orang-orang juga pada umumnya menerima tawaran bahwa "Saya akan membagikan beberapa data jika saya mendapat manfaat sebagai balasannya." Masalah muncul ketika perusahaan menyesatkan atau tidak kompeten sehubungan dengan tawar-menawar ini daripada transparan.

Jadi, mengesampingkan semua aspek level aplikasi dari UI multipemain (seperti tombol lobi dan daftar pemilih untuk memilih untuk bergabung dengan game), bagian sinkronisasi SLAM bukan hanya kotak centang, ini adalah UX dalam dirinya sendiri. . Jika UX itu tidak memberikan "hanya berfungsi," pengguna bahkan tidak akan repot-repot untuk mencapai level aplikasi untuk kedua kalinya. Mereka akan mencoba sekali karena penasaran, yang berarti bahwa pengamat pasar tidak harus memperhatikan unduhan aplikasi AR atau pengguna terdaftar, tetapi untuk mengulangi penggunaan.



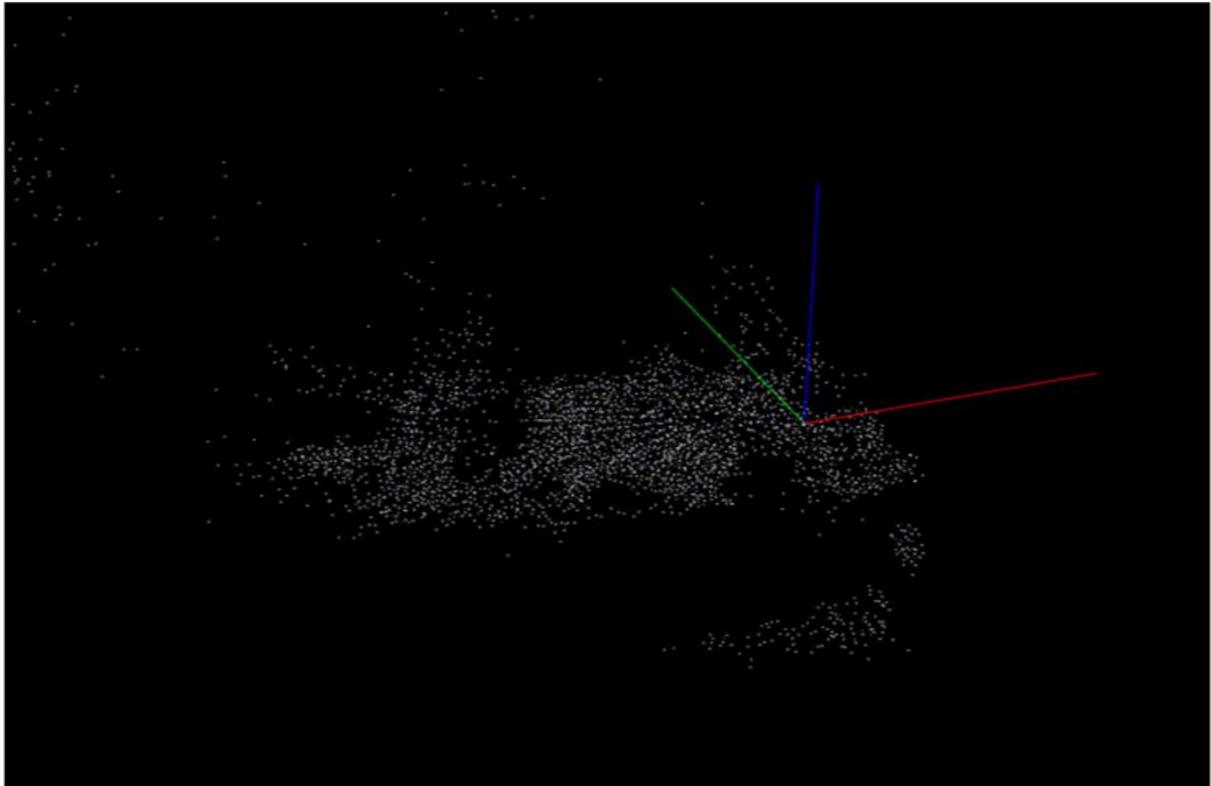
Gambar 5-28. Data gambar yang dapat dilihat oleh manusia tidak boleh keluar dari ponsel

Memungkinkan pengembang untuk membangun aplikasi AR yang menarik adalah hal yang perlu difokuskan oleh perusahaan cloud AR, dengan memecahkan masalah teknis yang menantang untuk mengaktifkan aplikasi AR-First yang asli dari AR. Ini berarti (seperti yang telah saya pelajari beberapa kali dengan menyakitkan) bahwa UX didahulukan. Meskipun kami adalah perusahaan visi komputer teknologi dalam, UX dari cara kerja sistem visi komputer tersebut adalah yang terpenting, bukan apakah sistem tersebut berfungsi sama sekali.

5.11 GAMBARAN LEBIH BESAR—PRIVASI DAN DATA AR CLOUD

Ketika datang ke Google Cloud Anchors, data gambar visual dikirim ke server Google. Ini adalah asumsi yang cukup aman bahwa ini berpotensi dapat direkayasa ulang kembali menjadi gambar yang dapat diidentifikasi secara pribadi (Google dengan hati-hati tidak jelas dalam deskripsi mereka, jadi saya berasumsi itu karena jika itu benar-benar anonim, mereka akan mengatakannya dengan sangat jelas).

Ini adalah sumber data gambar yang tidak boleh meninggalkan telepon, dan tidak pernah disimpan ke telepon atau disimpan dalam memori (Gambar 5-28). Ini adalah jenis data gambar visual yang dapat diidentifikasi secara pribadi yang tidak ingin Anda simpan atau pulihkan dari penyedia cloud AR. Google mengatakan bahwa itu tidak mengunggah bingkai video, tetapi deskriptor poin fitur dapat direkayasa ulang menjadi gambar (lihat Gambar 5-29).



Gambar 5-29. Point-cloud ini didasarkan pada data gambar kantor yang ditunjukkan pada Gambar 5-28

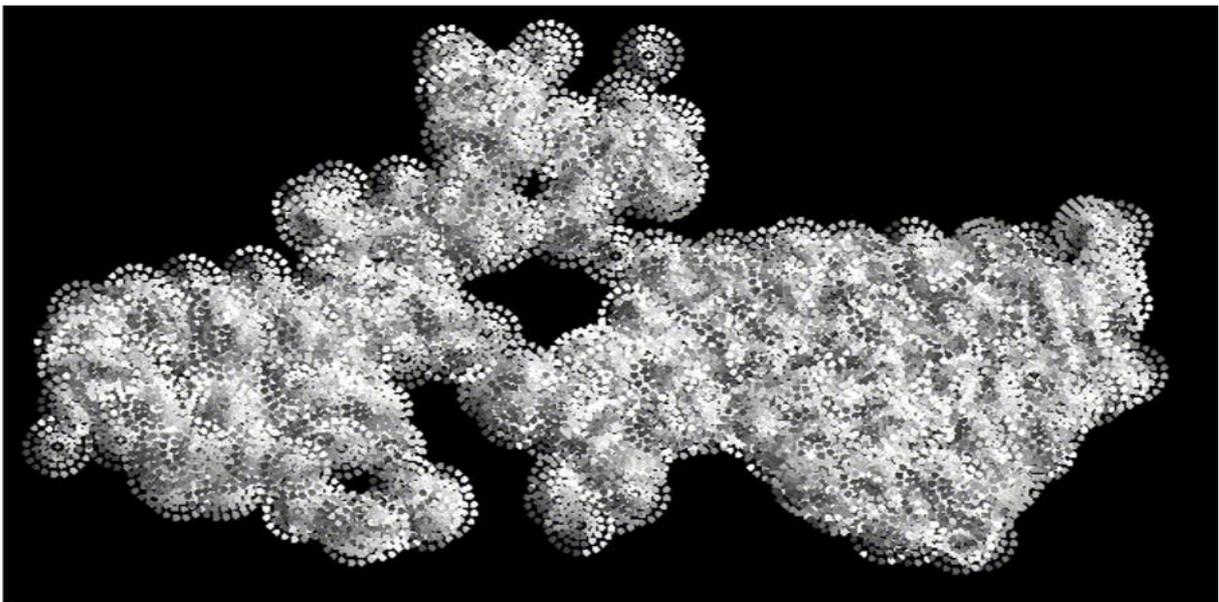
Untuk masa depan kemampuan cloud AR untuk memberikan ketekunan dan relokasi, data gambar visual tidak boleh meninggalkan telepon, dan bahkan tidak pernah disimpan di telepon. Pendapat saya adalah bahwa semua pemrosesan yang diperlukan harus dijalankan di perangkat secara real time. Dengan izin pengguna, semua yang harus diunggah adalah peta titik sparse postprocessed dan deskriptor fitur, yang tidak dapat direkayasa balik. Tantangan menarik yang kami (dan lainnya) sedang kerjakan adalah bahwa seiring perangkat mengembangkan kemampuan untuk menangkap, menggabungkan, dan menyimpan cloud titik padat, jerat, dan tekstur fotorealistik, ada lebih banyak nilai dalam produk yang lebih

"dapat dikenali" " data yang diambil adalah. Kami percaya ini akan membutuhkan pendekatan semantik baru untuk segmentasi data 3D dan identifikasi spasial untuk memberikan pengguna tingkat kontrol yang sesuai atas data mereka; ini adalah area yang sedang dieksplorasi oleh kelompok penelitian Oxford kami.

Gambar 5-29 menyajikan titik-cloud sparse untuk pemandangan di Gambar 5-28 (sistem kami memilih titik-titik sparse semi-acak, bukan sudut dan tepi geometris, yang tidak dapat digabungkan ke dalam ruang geometris yang dapat dikenali).

Bagian kedua dari teka-teki adalah "deskriptor fitur", yang disimpan oleh kami dan juga Google di cloud. Google sebelumnya telah mengatakan bahwa file Tango ADF, yang menjadi dasar ARCore, dapat membuat deskriptor fitur visualnya direkayasa balik dengan pembelajaran mendalam kembali menjadi gambar yang dapat dikenali manusia (Gambar 5-30) (dari dokumentasi ADF Tango — "ada di prinsip yang memungkinkan untuk menulis algoritma yang dapat merekonstruksi gambar yang dapat dilihat"). Perhatikan bahwa saya tidak tahu apakah ARCore mengubah spesifikasi jangkar dari ADF Tango cukup untuk mengubah fakta ini, tetapi Google telah menjelaskan bahwa ARCore didasarkan pada Tango, dan mengubah struktur data deskriptor fitur adalah perubahan yang cukup mendasar pada algoritme.

Ini sangat penting karena agar konten AR benar-benar persisten, perlu ada model data dunia nyata yang dihosting cloud yang persisten. Dan satu-satunya cara untuk mencapai ini secara komersial adalah agar pengguna akhir mengetahui bahwa deskripsi dunia nyata itu bersifat pribadi dan anonim. Selain itu, saya percaya akses ke data cloud harus dibatasi dengan mengharuskan pengguna secara fisik berdiri di tempat yang dijelaskan secara matematis data, sebelum menerapkan peta ke aplikasi.



Gambar 5-30. Ini adalah deskriptor fitur yang dihasilkan untuk setiap titik di point-cloud (sejauh data yang dihosting cloud 6D.ai dapat direkayasa balik, berdasarkan penerapan sains terbaru yang tersedia saat ini bersama dengan sumber daya komputasi yang besar)

Realitas mengenai data cloud AR ini menciptakan masalah pasar struktural untuk semua perusahaan platform AR utama saat ini, mengingat model bisnis Google dan Facebook (dan lainnya) dibangun dengan menerapkan data yang mereka kumpulkan untuk melayani iklan Anda dengan lebih baik. Platform seperti Apple dan Microsoft adalah silo dan karenanya tidak akan menawarkan solusi lintas platform. Mereka juga tidak akan memprioritaskan solusi cloud yang memungkinkan solusi P2P di perangkat yang dipatenkan.

Satu faktor yang saya anggap remeh adalah pengembang dan mitra besar memahami dengan jelas nilai data yang dihasilkan oleh aplikasi mereka, dan mereka tidak ingin memberikan data tersebut ke platform besar untuk dimonetisasi oleh organisasi tersebut. Mereka ingin membawa semuanya di rumah (seperti yang dilakukan Niantic) atau bekerja dengan mitra yang lebih kecil yang dapat memberikan paritas teknologi dengan platform besar (bukan tugas kecil) dan yang juga dapat menjamin privasi dan penyelarasan model bisnis. AR dipandang terlalu penting untuk memberikan fondasi data. Ini adalah keuntungan pasar struktural yang dimiliki oleh startup cloud AR, dan ini merupakan tanda yang menggembirakan untuk masa depan kami yang dapat diperkirakan.

Saat ARKit mengumumkan awal AR pada tahun 2017, kami yakin Cloud Anchors Google mengumumkan awal dari cloud AR. Aplikasi AR akan menjadi jauh lebih menarik, tetapi hanya jika penyedia cloud AR memberikan UX visi komputer yang “berfungsi” dan mengatasi beberapa masalah privasi yang menantang dan unik.

Glosarium

Ini bukan deskripsi teknis yang tepat dari istilah-istilah ini; jika Anda membutuhkannya, Anda dapat menemukannya di Wikipedia dan dokumen teknis online yang tak terhitung jumlahnya. Sebaliknya, ini adalah upaya untuk menyederhanakan istilah agar dapat dimengerti oleh khalayak umum.

SLAM (simultaneous location and mapping)

Ini adalah istilah luas yang mengacu pada sekelompok subsistem teknis yang membantu perangkat AR (dan robot) menentukan di mana ia berada di dunia. Ini mencakup hal-hal seperti melacak posisi Anda dari bingkai ke bingkai (VIO hanyalah salah satu jenis tracking) serta membangun peta ruang khusus yang dapat dibaca mesin untuk mengingat di mana Anda berada dalam jangka panjang (dan relokasi jika Anda tersesat). SLAM biasanya visual dan berbasis di sekitar kamera plus sensor lain, tetapi dimungkinkan untuk membangun sistem SLAM tanpa kamera, hanya menggunakan (misalnya) sinyal radio seperti WiFi.

VIO (visual inertial odometry)

Suatu bentuk tracking yang mengambil input dari kamera dan sensor inersia untuk melacak posisi perangkat secara real-time.

6DOF (6 degrees of freedom)

Mengacu pada posisi (x,y,z koordinat) dan orientasi (pitch, yaw, roll) perangkat, bersama-sama disebut sebagai pose Anda. Ini bisa dalam koordinat relatif (di mana saya relatif terhadap tempat saya memulai) atau koordinat absolut (mis., lintang, bujur, ketinggian, dll.).

Kebenaran dasar

Pose Anda yang "benar" mutlak. Biasanya diukur terhadap data yang disurvei atau diukur menggunakan sistem yang sangat akurat. Ini adalah konsep teoretis, karena setiap sistem pengukuran memiliki beberapa kesalahan kecil dari kebenaran dasar (misalnya, bahkan pengukuran berbasis laser memiliki kesalahan mikron). Tujuannya adalah agar sistem AR cukup dekat dengan kebenaran dasar yang tidak dapat diperhatikan manusia. Sebagian besar dari kita memperlakukan GPS sebagai kebenaran dasar, tetapi kita semua pernah mengalami betapa tidak akuratnya itu, dan sistem AR harus jauh lebih akurat.

IMU (ertial measurement unit)

Istilah yang mengacu pada kombinasi akselerometer dan giroskop di ponsel Anda, yang memberikan pengukuran yang dapat digabungkan dengan output kamera untuk membantu tracking.

SIFT

Deskripsi fitur yang akurat dan kuat untuk sistem SLAM untuk mengenali titik dalam ruang. Ini adalah kombinasi dari koordinat 3D ditambah deskripsi piksel di sekitar titik itu (misalnya, warna dan pencahayaan) sehingga sistem dapat mengenalnya lagi di bingkai berikutnya.

Point-cloud

Point-cloud 3D di ruang angkasa. Perhatikan bahwa ini tidak termasuk deskriptor fitur, yang diperlukan untuk relokasi dan penggunaan dalam sistem SLAM. Banyak orang salah berasumsi bahwa point-cloud adalah semua yang diperlukan untuk sistem SLAM. Sebagai gantinya, mereka membutuhkan peta, yang merupakan kombinasi dari point-cloud dan deskriptor fitur (ditambah mungkin beberapa metadata)

Kalman filter

Sebuah algoritma matematika yang memprediksi nomor berikutnya dalam seri berdasarkan input yang tidak dapat diandalkan. Ini adalah cara input dari IMU dan kamera digabungkan menjadi pose dan memprediksi beberapa frame ke depan untuk memperhitungkan waktu pemrosesan. Perhatikan bahwa tidak ada yang namanya "filter Kalman"; ada banyak jenis kompleksitas yang berbeda-beda dan setiap sistem AR akan menggunakan versi mereka sendiri yang dirancang sendiri.

BAGIAN IV

MEMBUAT *AUGMENTED REALITY* (AR) DAN *VIRTUAL REALITY* (VR)

Ketika Facebook membeli Oculus Rift pada tahun 2014 seharga IDR200 miliar, industri investasi dikatalisasi dengan *Virtual Reality* (VR) yang divalidasi sebagai pengantar awal era baru komputasi. Perlahan pasar dibanjiri dengan alternatif Oculus dalam perang untuk ruang di wajah Anda. Sampai saat itu, *augmented reality* (AR) praktis didominasi oleh Vuforia, solusi tracking berbasis gambar. Tetapi dengan akuisisi Oculus, tiba-tiba beberapa perusahaan dan produk di ruang imersif dengan cepat naik ke pengakuan: Meta dan Magic Leap untuk AR, Samsung GearVR dan Google Cardboard, dan kemudian Hololens dan Daydream di antara banyak lainnya.

Sebagai investor Qualcomm Ventures, Steve Lukas ditugaskan untuk menemukan perusahaan mana yang akan “menang” di ruang VR/AR; saat itulah masalah besar diidentifikasi: ekosistem perangkat lunak yang terbatas dan semakin retak dengan setiap rilis headset. Ini tidak sepenuhnya jelas hingga 2016, ketika semua platform mulai dirilis ke publik. Seperti kasus biasa dengan industri baru, pendukung VR yang paling vokal menyebabkan ekspektasi adopsi yang berlebihan, dan VR akhirnya terjual dalam jumlah yang lebih rendah daripada yang diantisipasi semula. Ini konsisten untuk setiap rilis headset, dengan setiap platform tampak seperti platform yang akan mengambil arus utama pasar, apakah itu kekuatan penuh HTC Vive, Samsung GearVRs gratis yang didistribusikan dengan ponsel baru mereka, biaya rendah dan jangkauan Google Daydream, atau basis besar PlayStation 4 yang terpasang, yang akan menjadikan PlayStation VR sebagai pemimpin. Kami melihat siklus sensasi yang sama dengan diperkenalkannya AR seluler pada tahun 2017, dengan ARKit Apple dan ARCore Google, yang sekali lagi menandakan penghematan industri, hanya saja berkali-kali gagal memenuhi ekspektasi analis. Periksa kuitansi.

Perusahaan mana yang akan menang di VR/AR? Kesimpulan: tidak satupun dari mereka, jika mereka tidak menyelesaikan masalah adopsi. Ada banyak faktor yang membatasi pertumbuhan dalam komputasi spasial, tetapi bahkan jika semuanya diselesaikan hari ini, masalah konten adalah salah satu yang akan menjadi lebih buruk sebelum menjadi lebih baik. Industri VR/AR di masa-masa awal masih menarik bagi ceruk pasar: mereka yang memiliki pendapatan sekali pakai (IDR10 juta – IDR 30 juta) pada pengaturan perangkat yang hanya menyediakan beberapa konten untuk dinikmati dan dinikmati. Sebuah janji masa depan, sementara masih terjebak di masa sekarang. Ternyata, jumlah itu adalah bagian yang sangat kecil dari populasi yang bersedia menghabiskan begitu banyak untuk mendapatkan sedikit hiburan dan utilitas jika dibandingkan dengan ponsel, komputer, dan sistem permainan modern.

Pemirsa ini, meskipun kecil, tidak akan tumbuh sampai pemirsa arus utama masuk. Pengadopsi awal telah melakukan pembelian, dan jumlahnya tidak cukup. Dengan tujuh platform berbeda yang keluar pada tahun 2016, kelompok kecil pengadopsi awal itu dibagi

menjadi tujuh bagian, masing-masing kamp dengan ekosistem konten mereka sendiri. Tidak ada standar untuk pengembangan VR/AR, karena bahkan mesin pengembangan Unity yang paling tersedia pun masih memerlukan integrasi SDK untuk setiap perangkat target serta tantangan desain unik yang melekat pada setiap faktor bentuk.

Berkembang hanya pada satu perangkat eXtended reality (XR) berarti membatasi audiens target Anda ke sebagian kecil dari pasar kecil yang sedang berkembang. Kemungkinannya tinggi bahwa satu perangkat yang Anda pilih akan segera digantikan oleh rilis berikutnya jika tidak seluruhnya oleh perangkat pesaing, dengan kedua situasi tersebut menyebabkan berbagai tingkat keusangan untuk produk yang mungkin telah Anda buat. Dengan demikian, keserbagunaan sangat disarankan untuk pengembang XR awal. Pengembang perlu memahami perbedaan antara mengembangkan dalam komputasi spasial umum versus mengembangkan untuk perangkat yang sebenarnya. Anda perlu tahu di mana pemisahan itu ada dan bagaimana surut dan mengalir ke iterasi baru perangkat keras serta platform perangkat baru sama sekali. Pendekatan ini dimaksudkan untuk memungkinkan pengembangan perangkat-agnostik untuk membuktikan keahlian Anda di masa depan saat kami melihat headset baru datang ke pasar setiap beberapa bulan selama belasan tahun ke depan.

Untuk memulai, di Bab 6, Steve Lukas memberikan beberapa sejarah dan filosofi teori pengembangan lintas platform, berdasarkan waktunya dengan Qualcomm Ventures dan memulai Across Realities. Dia membahas pendekatan konseptual dan menunjukkan beberapa contoh teknik abstraksi dalam pengembangan untuk XR.

Kemudian, di Bab 7, Vasanth Mohan dari FusedVR memberikan pandangan yang lebih dalam tentang taktik dan strategi pengembangan lintas platform sambil memandu Anda melalui beberapa tutorial. Di Bab 8, Harvey Ball dan Clorama Dorvilias menyelesaikan ujian kami dengan pelajaran sejarah singkat dan panduan VRTK, sebuah proyek sumber terbuka yang dimaksudkan untuk memacu pengembangan lintas platform.

Perhatikan bahwa untuk menjaga semua platform tetap agnostik dan relevan terlepas dari perubahan pengembangan, dalam bab-bab ini kami mengkode menggunakan pseudocode dan memposting tangkapan layar. Jika Anda ingin mempelajari kode kerja untuk setiap proyek, lihat semua repositori GitHub kami, tautannya disediakan di akhir setiap bab. Ini baru permulaan, dan memiliki fondasi dasar yang kuat untuk dikembangkan untuk semua platform akan menjadi penting pada tahap siklus hidup XR ini.

BAB 6

TEORI LINTAS PLATFORM AUGMENTED REALITY DAN VIRTUAL REALITY

Melompat langsung ke dalam membangun pengalaman *virtual reality* (VR) bisa sangat menakutkan. Hal yang sama berlaku untuk *augmented reality* (AR), dan *brainstorming* keputusan sederhana apakah pengalaman harus dilakukan dalam VR atau AR adalah latihan yang baik. Demi kesederhanaan, bab ini menjelaskan pengalaman VR dan AR secara bergantian sebagai "immersive" karena sebagian besar pengembangan konten imersif menggunakan prinsip yang sama. Di mana ada perbedaan benuansa antara keduanya, kami secara eksplisit merujuknya.

Langkah pertama mempelajari pengembangan imersif tergantung pada perspektif Anda dan apa tujuan Anda. Anda kemungkinan besar mengidentifikasi dengan satu atau lebih dari pernyataan berikut:

- Saya tidak memiliki pengalaman pengembangan.
- Saya memiliki pengalaman pengembangan dalam grafik 3D.
- Saya memiliki ide aplikasi dalam pikiran untuk VR.
- Saya ingin belajar cara membuat VR sebelum memikirkan ide aplikasi.

Jika Anda memiliki gagasan tentang proyek yang ingin Anda bangun, ini merupakan keuntungan karena Anda kemudian dapat menargetkan pencapaian tertentu sambil mempelajari pelajaran Anda sebagai blok bangunan menuju produk yang telah selesai. Alternatifnya, jika Anda ingin mempelajari semua alat terlebih dahulu, ini akan membantu dalam struktur ide aplikasi pertama Anda karena Anda akan memahami fitur dan batasan VR sebelum berkomitmen pada ide yang tidak layak. Terlepas dari itu, tidak ada cara kerja yang salah atau lebih baik, kecuali mengikuti sistem yang beroperasi paling baik untuk Anda.

Dalam hal ini, kami memecah blok pembangun dalam membangun pengalaman yang imersif dengan mempertimbangkan permainan akhir yang direncanakan. Bab ini lebih berfokus pada pemikiran tingkat tinggi. Kami membahas mengapa lintas platform itu penting, memberikan dasar pada mesin permainan, dan menawarkan beberapa strategi untuk membangun kerangka kerja lintas platform.

Mengapa Cross-Platform?

Menangani lintas platform dapat dilihat sebagai topik tingkat lanjut, tetapi sebenarnya ini adalah solusi desain dasar yang memengaruhi seluruh arsitektur produk apa pun yang ingin tampil imersif.

Di masa-masa awal VR dan AR ini, semuanya masih eksperimental—desain headset, desain pengontrol, skema aksesoris, dan sebagainya. Kami dapat mengidentifikasi lebih dari 16 kombinasi headset dan controller yang berbeda untuk VR dan AR, dengan lebih banyak lagi yang akan datang setiap beberapa bulan. Sampai konsistensi tercapai, konten terpecah-pecah di seluruh ekosistem. Dengan VR dan AR, kami berada di ujung spektrum kompatibilitas. Di sisi dekat, ada televisi tradisional dan perangkat media seluler, yang berpotensi dapat memutar

semua konten layar datar terlepas dari produsennya. Di tengah, kami memiliki konsol video game, di mana setiap perangkat berbagi beberapa konten (misalnya, Fortnite, Minecraft) sementara juga memiliki beberapa konten eksklusif (misalnya, Uncharted di PlayStation, Mario di Nintendo, Halo di Xbox). Kemudian, di ujung yang jauh, kami memiliki VR dan AR, di mana konten yang berpusat pada platform pada hari-hari awal lebih menjadi norma (misalnya, Robo Recall di Oculus Rift, FarPoint di PlayStation VR, Lego Brickheadz di Daydream).

Ada beberapa alasan untuk ini. Berbagai macam paradigma kontrol yang ditawarkan dengan berbagai jenis pengaturan headset VR dan AR belum mencapai serangkaian standar yang disepakati. Dengan demikian, pengalaman dibangun untuk memanfaatkan set fitur dan metode input setiap perangkat keras. Ini dapat dikategorikan menjadi berikut:

- Headset yang ditambahkan dengan satu atau lebih pengontrol
- Headset seluler dengan pengontrol
- Wadah VR drop-in tanpa pengontrol

Ini tidak benar-benar mencakup seluruh spektrum. Oculus Rift diluncurkan dengan gamepad dan remote dan kemudian merilis Touch Controllers yang dikirimkan akhir tahun yang sama, menawarkan setidaknya tiga skema kontrol alternatif di luar mouse dan keyboard tradisional. Sementara Oculus Go hadir sebagai standar dengan pengontrol, platform Samsung GearVR dikirimkan sebagai headset drop-in dengan dukungan gamepad sebelum merilis opsi controller terlacak setahun kemudian. Akibatnya, tidak dapat dijamin bahwa pemilik platform juga akan memiliki input apa pun yang awalnya tidak dikirimkan bersama platform. Untuk menjangkau khalayak terluas, kita harus menangani input inti yang tersedia untuk setiap konsumen. Saat membangun kontrol input adaptif, ini akan membantu menskalakan produk antara kontrol yang paling tersedia hingga yang paling kuat.

Sampai VR dan AR bergerak lebih dekat ke tengah dalam hal kompatibilitas, itu akan terbatas dalam pesan yang dikirimkannya ke konsumen arus utama bahwa "VR sudah siap," ketika konsumen tersebut tidak dapat membeli satu headset dan secara bersamaan mendapatkan mayoritas teratas yang tersedia. isi. Hal yang sama berlaku untuk pemasaran: laba atas investasi untuk kampanye VR terbatas karena tingginya biaya pengembangan konten ditambah dengan basis audiens yang terbatas dari satu headset.

Meskipun adopsi arus utama adalah tujuan industri, itu mungkin bukan tujuan pengembang tertentu, juga bukan tanggung jawab mereka. Jadi, ada aliran pemikiran alternatif di mana pengembang hanya ingin membuat pengalaman premium tingkat tertinggi baik sebagai hobi atau membuat aplikasi bernilai tinggi untuk audiens yang lebih kecil. Ini dapat menemukan kesuksesan ketika berkembang untuk pasar perusahaan atau di lingkungan yang sangat terkontrol seperti arcade atau instalasi VR. Dalam semua kasus ini, hanya berfokus pada satu platform mungkin baik untuk dilakukan, dan dengan demikian bab ini mungkin tidak memiliki banyak nilai.

Namun, mendesain untuk portabilitas memiliki keuntungan tambahan untuk menghindari penguncian vendor. Dengan mempertahankan desain sebagian besar platform agnostik, aplikasi akan dapat beradaptasi dengan platform perangkat keras alternatif dengan

sangat cepat dengan biaya pengembangan yang lebih rendah. Ini bisa menguntungkan ketika platform yang lebih cocok diluncurkan serta ketika peluang bisnis yang lebih baik mungkin muncul dengan perusahaan VR dan AR yang bersaing. Juga tidak pasti pada titik ini platform perangkat keras mana yang akan bertahan dari gelombang industri VR dan AR yang terus berubah, sehingga metode ini mengurangi risiko terikat pada satu platform yang mungkin akhirnya mengambil pangsa pasar terkecil di masa depan. .

Singkatnya, manfaat penargetan pengembangan lintas platform mencakup fleksibilitas, audiens potensial yang lebih besar, dan pemeriksaan masa depan dengan porting yang disederhanakan ke platform baru. Selain semua ini, pengembangan lintas platform bisa sangat bermanfaat ketika melihat karya Anda ditampilkan di setiap platform baru dalam waktu yang sangat singkat.

Perlu diingat bahwa saat ini belum ada standar industri yang ditetapkan untuk pengembangan VR dan AR lintas platform. Karena sifatnya, ini tidak akan berubah untuk beberapa waktu. Namun, ada sejumlah alat yang tersedia untuk membantu, masing-masing dengan teknik dan manfaat yang berbeda. Karena itu, Anda perlu menyadari bahwa ada beberapa pendekatan untuk menangani pengembangan lintas platform, dengan tidak ada satu solusi yang menjadi satu-satunya "cara yang benar" untuk melakukannya. Dalam bab ini, kami menyajikan beberapa solusi yang teridentifikasi. Dengan demikian, Anda sebaiknya hanya menggunakan ini sebagai panduan atau referensi untuk memahami berbagai teknik dalam lanskap yang berkembang ini, dan pada akhirnya, Anda harus mengadopsi praktik terbaik yang sesuai dengan kebutuhan Anda sendiri.

6.1 PERAN MESIN GAME

Meskipun aplikasi VR dapat dikembangkan menggunakan C++, saat ini mesin game sangat populer untuk segala hal mulai dari membuat prototipe cepat konsep game dalam hitungan jam hingga membangun produk triple-A yang dirilis sepenuhnya. Apa itu mesin permainan? Ini adalah istilah industri untuk satu set perangkat lunak yang mengambil serangkaian input (mouse, keyboard, layar sentuh, dll.), menerapkan logika padanya (misalnya, memindahkan karakter, melompat, senjata api), dan menghasilkan respons, biasanya dalam bentuk umpan balik visual dan audio (misalnya, pembaruan skor, efek suara). Nama "mesin permainan" berasal dari desain aslinya untuk menangani aplikasi game, dengan manfaat utama adalah bahwa banyak matematika yang kompleks dan logika kode tingkat rendah akan diprogram ke dalam sistem. Selain itu, mesin game juga pada akhirnya akan menjadi kompatibel multiplatform, membangun serangkaian desain kode yang sama oleh pengembang sambil dapat diterapkan ke target platform yang berbeda.

Keuntungan utama dari mesin game yang muncul adalah kemampuan untuk menargetkan semua jenis arsitektur sistem tanpa harus mempelajari banyak bahasa pemrograman dan API yang bergantung pada platform. Pengembang game dapat bekerja dengan bebas di mesin game yang mereka pilih untuk dipelajari dan kemudian diterapkan ke sistem baru saat mereka datang ke pasar.

Dengan munculnya aplikasi seluler, dan terutama dengan virtual dan augmented reality, kebutuhan mesin game 3D menjadi lebih kuat karena banyak tantangan pengembangan dunia 3D di dunia virtual telah diselesaikan oleh mesin game. Dengan demikian, Unity dan Unreal Engine dengan cepat menjadi mesin game terkemuka untuk membuat prototipe dan membuat konten VR.

Meskipun Unreal Engine memiliki manfaat tersendiri, dalam bab ini kita fokus pada Unity. Awalnya dirilis pada tahun 2005, Unity telah membantu banyak pengembang di seluruh dunia untuk memulai membangun game tiga dimensi, termasuk semuanya, mulai dari seluler, konsol, hingga desktop. Ini telah berfungsi sebagai tulang punggung untuk pengembangan 3D bagi banyak pengembang sambil membina komunitas yang luar biasa selama bertahun-tahun karena perusahaan terus mengembangkan produknya untuk kebutuhan set fitur VR dan AR baru yang terus berubah.

Selain fleksibilitas dan kemudahan penggunaan, Unity mendapat manfaat dari kemitraan integrasi yang kuat dengan semua platform komputasi utama, dan, dalam beberapa kasus, Unity bahkan diperlukan jika Anda ingin menggunakan mesin game apa pun. Salah satu contohnya adalah Microsoft HoloLens, yang hingga tulisan ini dibuat, tidak dapat ditargetkan oleh mesin game komersial lainnya. Saat melihat pengembangan lintas platform yang paling umum di VR dan AR, Unity saat ini memiliki jangkauan terluas.

Aplikasi mesin game dibuat menggunakan lingkungan pengembangan terintegrasi (IDE), yang merupakan istilah keren untuk apa yang mungkin Anda kenal sebagai aplikasi desktop yang berjalan di komputer Anda. Untuk mengikuti contoh, kami sarankan Anda mengunduh Unity IDE. Rilis publik Unity saat ini harus dilakukan, tetapi untuk tujuan kompatibilitas, kami menggunakan Unity 2018.1f1.

Unity itu kuat dan fleksibel. Alat bawaan dan plug-in eksternal telah meningkat secara signifikan dari waktu ke waktu, menanggapi umpan balik pengembang untuk mempertahankan komunitas yang kuat. Di permukaan, Unity menangani pengembangan dan penyebaran lintas platform, tetapi memanfaatkan fitur setiap platform membutuhkan kemahiran yang melampaui pemetaan ulang kontrol dasar. Unity sangat memperhatikan pengembang dalam hal angkat berat, tetapi masih ada latihan yang tersisa untuk diselesaikan oleh pengembang, yang akan kita bahas lebih lanjut di bagian ini.

Mempelajari Unity dari awal berada di luar cakupan buku ini, tetapi banyak tutorial tersedia secara online, dari situs web Unity secara langsung serta sumber daya besar yang tersedia di YouTube, Udemy, dan PluralSight, antara lain.

6.2 MEMAHAMI GRAFIK 3D

Jika Anda pernah mengembangkan game 3D di Unity, VR adalah modifikasi yang sangat kecil di atasnya. Praktis satu langkah untuk mengaktifkan VR untuk persyaratan minimum untuk mendeklarasikan aplikasi sebagai VR-enabled. Jika Anda memahami cara kerja kamera virtual dalam grafik 3D, Anda dapat melompat ke subbagian berikutnya.

6.3 KAMERA VIRTUAL

Kamera virtual berada di fondasi inti VR. Secara tradisional, di dunia nyata, kita mengenal kamera sebagai perangkat mekanis atau elektronik yang mengambil gambar dan video. Dalam obrolan video antara dua ponsel, setiap orang memegang ponsel di dunia 3D nyata yang mentransmisikan apa yang dilihat ponsel, secara real time, ke perangkat layar datar orang lain. Kamera virtual di dalam mesin permainan Unity dapat dianggap sama, tetapi alih-alih kamera menjadi perangkat nyata yang ditempatkan di dunia 3D nyata, kamera ditempatkan di lingkungan 3D virtual. Dengan demikian, umpan langsung diberikan ke televisi atau monitor layar datar. Memindahkan kamera di sekitar dunia 3D dapat dilakukan secara tradisional dengan gamepad atau kombinasi keyboard dan mouse, dengan TV atau monitor yang menampilkan sudut pandang karakter yang diperbarui secara real time.

Di VR, beberapa hal berubah di sini. Pertama, kamera menempel di kepala pengguna, jadi alih-alih menggunakan tangan untuk mengubah sudut pandang, Anda cukup menggerakkan kepala. Kedua, tampilan dirender dua kali: satu kali untuk setiap mata, masing-masing diberi layarnya sendiri, dengan posisi kamera virtual setiap mata sedikit digeser dari tengah sehingga pemirsa akan mengalami efek paralaks stereoskopik. Ini semua ditangani untuk penampil di Unity dengan sakelar sederhana, tetapi ini benar-benar berarti bahwa mengembangkan untuk VR adalah mengembangkan terlebih dahulu untuk 3D. VR bisa datang sesudahnya, yang sangat penting untuk merencanakan strategi pembelian perangkat keras Anda. Untuk memulai, yang Anda butuhkan hanyalah laptop (atau desktop jika Anda bukan tipe workaholic portabel). Anda dapat mempelajari mekanisme dasar Unity dan membuat dunia 3D dengan input keyboard untuk menggerakkan kamera, lalu memasang VR di bagian akhir untuk melompat ke dalam pengalaman. Dari sana, Anda bisa beradaptasi sesuka hati.

Sebagian besar hal yang sama berlaku untuk AR, jelas dengan beberapa perbedaan. Pertama, dengan AR seluler menggunakan ponsel, kamera ponsel dipegang di tangan Anda alih-alih dipasang di kepala Anda. Namun, umpan kamera virtual disajikan ke layar datar ponsel (tanpa split optik untuk VR) dengan cara yang hampir sama.

Memahami kamera sangat penting karena perubahan kamera dari posisi yang relatif tetap dalam video game menjadi sepenuhnya dapat dipindahkan dalam VR adalah perubahan besar yang memengaruhi cara game dirancang dan dioptimalkan. Pertama, kecepatan bingkai grafis harus berperforma tinggi untuk mengurangi efek buruk pada otak, dan, kedua, trik dan teknik kinerja tertentu (seperti area yang belum selesai di dunia yang kemungkinan besar tidak akan dilihat pengguna) tidak tersedia jika pemirsa memilikinya. Kebebasan penuh untuk menjelajahi daerah tersebut.

Tidak semua perangkat keras VR identik, bahkan di tingkat kepala, jadi topik penting berikutnya adalah bagaimana kontrol kamera virtual ditangani secara berbeda untuk setiap platform, yang membawa kita ke istilah berikut yang perlu Anda ketahui untuk VR: tiga derajat- kebebasan dan enam derajat kebebasan.

Derajat Kebebasan Derajat kebebasan, atau DOF, mengacu pada variasi gerakan yang tersedia untuk objek yang dilacak. Objek yang dilacak adalah objek yang bergerak dalam ruang fisik dan melaporkan informasi posisi dan/atau rotasinya ke mesin game. Ini dilakukan melalui

kombinasi data sensor, tetapi yang paling penting, posisi dan/atau orientasi objek yang dilacak di dunia nyata dapat direpresentasikan di dunia virtual, menyinkronkan dunia nyata dan dunia maya.

Headset Virtual Reality hadir dalam dua rasa: tiga derajat kebebasan (3DOF) dan enam derajat kebebasan (6DOF). Jika Anda telah mencoba VR, Anda mungkin tidak menyadari yang mana yang telah Anda gunakan dan mengapa headset tertentu dapat menyebabkan ketidaknyamanan lebih dari yang lain. Jika Anda pernah menggunakan headset VR seluler tanpa sambungan yang ditenagai oleh ponsel, kemungkinan besar Anda menggunakan headset 3DOF. Jika Anda dapat berbalik dan melihat ke belakang dalam VR, tetapi Anda tidak dapat merasakan berjalan menuju objek di kejauhan, Anda mungkin menggunakan headset 3DOF. Hal yang sama berlaku jika Anda berjongkok dan pandangan Anda tidak berubah. Ini karena tracking 3DOF berarti bahwa rotasi objek yang dilacak dilaporkan ke perangkat lunak, tetapi posisinya tidak. Dengan tracking rotasi, mesin game akan memiliki informasi tentang yaw, pitch, dan roll headset (rotasi sepanjang sumbu x, y, dan z, belum tentu masing-masing). Ini umumnya dikenal sebagai tracking 3DOF. Google Cardboard, Samsung GearVR, Google Daydream View, dan Oculus Go termasuk dalam kategori ini, dan tracking 3DOF dapat dilakukan menggunakan sensor akselerometer internal, giroskop, dan magnetometer yang ada di sebagian besar chipset ponsel.

Sisa 3DOF dalam headset 6DOF adalah posisi x, y, dan z di sepanjang sumbu x, y, dan z (masing-masing). Karena pengalaman 3DOF tidak dapat menggerakkan kamera menggunakan kepala Anda sendiri, kamera virtual dapat bergerak secara otomatis (seperti dalam pengalaman rollercoaster) atau beberapa bentuk gerakan diimplementasikan melalui input kontrol. Ini dikenal sebagai penggerak, dan kita akan membahasnya lebih lanjut di bagian berikutnya. Karena ketidaknyamanan otak dan tubuh, penggerak memiliki sejumlah solusi, termasuk teleportasi dan kekaburan kamera untuk mengurangi efek samping mabuk perjalanan.

Tabel 6-1. Platform VR yang tersedia

3DOF VR platforms	Input method
Mobile VR: Cardboard/phone drop-in	Gaze and hold position
Mobile VR: Cardboard/phone drop-in	Untracked
Oculus Mobile with headset touchpad	Headset touchpad, untracked. Clickable touchpad + 1 button
Oculus Mobile with Gamepad	Gamepad, untracked. Digital direction pad, analog controller, 6 face + 4 trigger buttons each
Oculus Mobile with controller	1 x 3DOF hand controller with clickable touchpad + 2 buttons
Google Daydream View	1 x 3DOF controller w/clickable touchpad + 1 button
6DOF VR platforms	Input method
Google Daydream: Mirage Solo	1 x 3DOF controller w/clickable touchpad + 1 button
HTC Vive Focus	1 x 3DOF controller w/clickable touchpad + 2 buttons
Oculus Rift with Xbox controller	Gamepad, untracked. Digital direction pad, analog controller, 6 face + 4 trigger buttons each

Oculus Rift with remote	Remote, untracked. Directional pad + 1 button
Oculus Rift with Touch controllers	2 x 6DOF controllers. Clickable joystick, 2 face + variable trigger + grip, each
HTC Vive	2 x 6DOF controllers. Clickable touchpad + 2 buttons + variable trigger, each
Microsoft mixed reality headset	2 x 6DOF controllers. Clickable touchpad + 2 buttons + variable trigger, each
6DOF AR platforms	
Microsoft Hololens	2 x positional 3DOF hands, detect hand + hand tap only
Microsoft Hololens	1 x clicker with 1 button only
Mobile AR: iPhone/Android	Touchscreen

Pengalaman 6DOF terasa lebih alami karena asosiasi 1:1 yang lengkap antara menggerakkan kepala Anda ke segala arah dan menyamakan pengalaman visual. Dalam pengalaman ini, Anda dapat berjongkok ke lantai untuk mengambil sesuatu, berjinjit untuk mendapatkan sudut pandang yang lebih baik, atau menghindar untuk menangkap bola. Namun, tracking posisi adalah masalah kompleks yang membutuhkan pemahaman tentang objek yang dilacak relatif terhadap lokasi dunia nyata di luar angkasa. Sensor visual perlu ditempatkan baik pada objek yang dilacak itu sendiri atau pada lokasi tetap yang menghadap objek. Meskipun tracking penuh memiliki kelebihan, itu juga bisa mengorbankan kebebasan bergerak. Saat ini, sebagian besar solusi 6DOF ditambahkan ke komputer, dengan kabel menjuntai yang berfungsi sebagai tali pengikat yang dapat menyebabkan bahaya tersandung atau dapat terlepas atau tercabut dari mesin jika tidak dikelola dengan benar. Kemajuan dalam teknologi seperti Vive Focus, Google Standalone Daydream, dan headset Oculus Santa Cruz membawa kita lebih dekat ke kebebasan nirkabel dengan kemampuan 6DOF penuh, tetapi mayoritas headset yang digunakan masih pengalaman 3DOF.

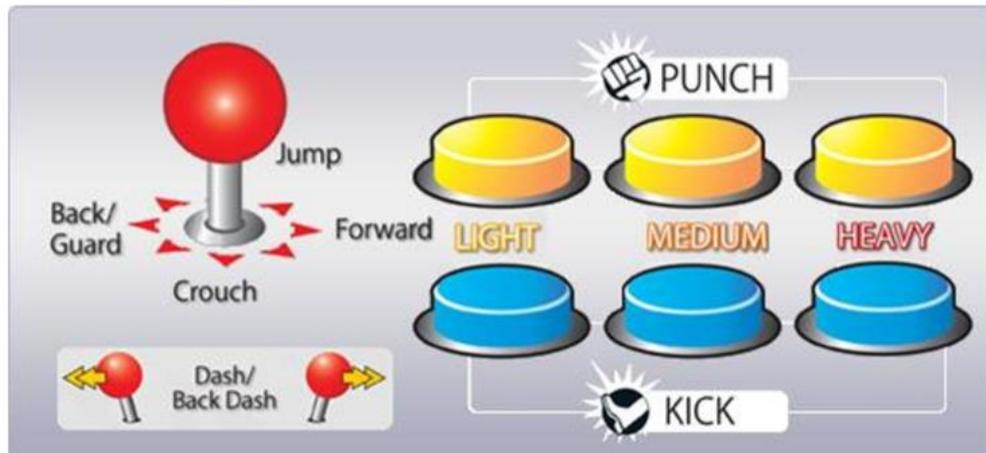
Kontroler adalah faktor lain untuk Virtual Reality. Controller dapat dilacak atau tidak, dan, seperti headset, tracking hadir dalam variasi 3DOF dan 6DOF. Tabel 6-1 mencantumkan platform VR utama yang tersedia pada pertengahan 2018.

Melihat Tabel 6-1, tidak sulit membayangkan bahwa mendukung 16 headset yang berbeda ditambah kombinasi input untuk VR dan AR dapat tampak cukup menakutkan. Namun, ini adalah tantangan yang diselesaikan setiap hari oleh siapa pun yang melakukan perencanaan ruang dunia nyata: mengakomodasi anak-anak, kereta bayi, kursi roda, orang-orang dengan ukuran besar dan kecil, pendek dan tinggi, tuna rungu, tunanetra, dan sebagainya. Dunia nyata terus-menerus direkayasa untuk mengakomodasi audiens sebanyak mungkin dengan landai kursi roda, braille, subtitle, dan banyak lagi. Dengan batasan yang jelas diterapkan jika diperlukan ("Anda harus setinggi ini untuk dikendarai"), sudah ada istilah standar industri untuk ini: aksesibilitas. Prinsip-prinsip desain aksesibilitas, ketika diterapkan pada VR, dapat disesuaikan dengan indah.

3DOF VR platforms	Input method
Oculus Rift with remote	Remote, untracked. Directional pad + 1 button
Oculus Rift with Touch controllers	2 x 6DOF controllers. Clickable joystick, 2 face + variable trigger + grip, each
HTC Vive	2 x 6DOF controllers. Clickable touchpad + 2 buttons + variable trigger, each
Microsoft mixed reality headset	2 x 6DOF controllers. Clickable touchpad + 2 buttons + variable trigger, each
6DOF AR platforms	
Microsoft Hololens	2 x positional 3DOF hands, detect hand + hand tap only
Microsoft Hololens	1 x clicker with 1 button only
Mobile AR: iPhone/Android	Touchscreen

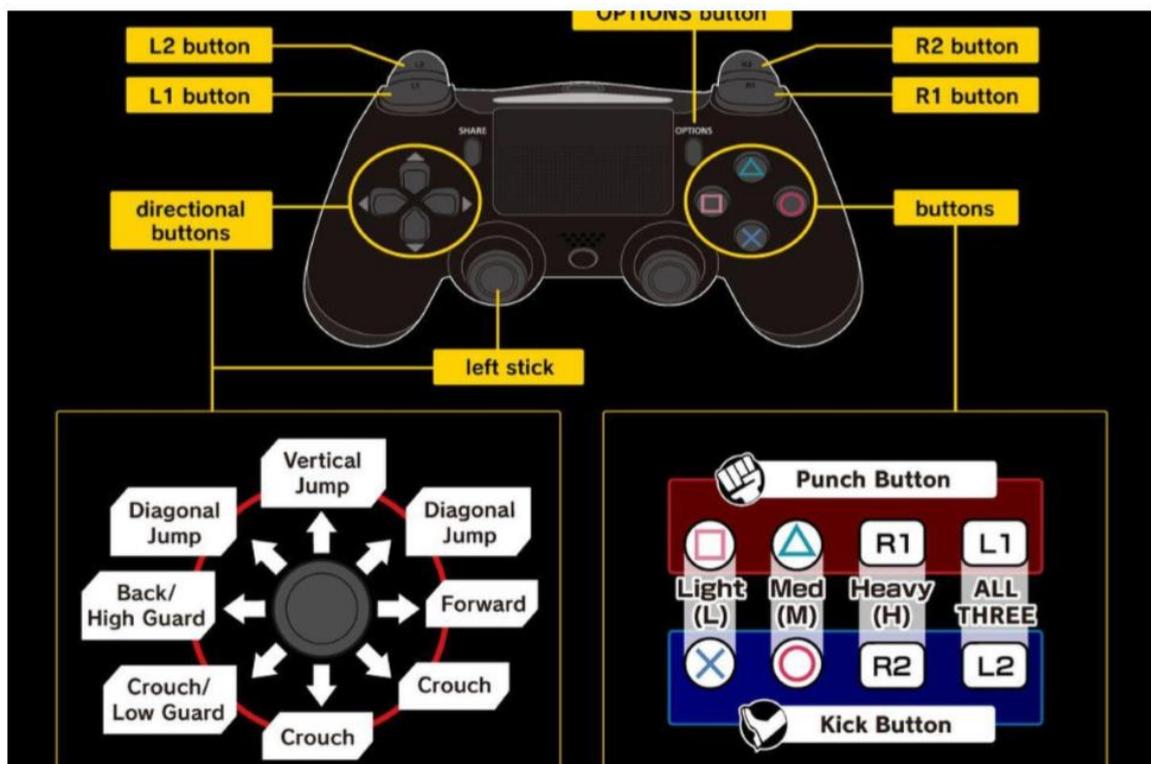
Tabel 6-1 hanya mencakup rilis yang saat ini tersedia dari lima produsen VR dan AR komersial utama: Facebook, Google, Microsoft, HTC, dan Apple. (Sony juga memiliki platform PlayStation VR, tetapi ekosistem pengembang tertutupnya lebih sulit untuk dibobol, dan semua platform di Tabel 6-1 dapat dibeli dan dikembangkan oleh rata-rata orang dalam hitungan minggu jika bukan hari.) Menekankan kembali bahwa meskipun lintas platform merupakan tantangan teknis, ini adalah tantangan desain yang bahkan lebih sulit. Ada alasan mengapa sebagian besar konten video 360 derajat yang muncul secara konsisten di setiap platform, dan itu karena sistem interaksi "duduk dan menatap, melihat-lihat" dapat dikaitkan dengan hampir semua platform yang disebutkan di atas.

Maksudnya adalah mempertimbangkan keseluruhan pengalaman yang ingin Anda sampaikan. Apa yang Anda ingin pemain lakukan di lingkungan virtual ini? Apakah itu pengalaman pasif atau aktif? Apa yang menurut mereka memuaskan? Jika pengalaman yang Anda rancang adalah murni sebuah karya untuk perangkat keras seperti tracking tangan atau antarmuka controller baru, penekanannya pasti akan berpusat pada platform dan mungkin tidak dapat dipindahkan ke sistem lain. Menjaga konsep tingkat tinggi berdasarkan pengalaman daripada interaksi memungkinkan kebebasan terbesar dalam desain pengalaman, dan fungsi controller kedua dapat diterapkan segera setelah pembangunan dunia diselesaikan. Salah satu strategi produk adalah bahwa aplikasi harus dirancang untuk perangkat keras kelas atas dengan kemampuan penuh dan kemudian dikurangi secara bertahap untuk menangani platform kelas bawah. Produk yang melakukan hal ini secara tidak benar cenderung menderita akibat awalnya tidak dirancang untuk ujung bawah selama desain produk asli. Selain masalah grafis, skema kontrol diganti dan diganti dalam upaya untuk mensimulasikan versi pengalaman input kelas atas, yang benar-benar dapat ditampilkan. Beberapa video game telah pindah dari rilis konsol rumah aslinya ke format game portabel. Salah satu contohnya adalah game Street Fighter II yang asli, di mana tata letak enam tombol disediakan di arcade (lihat Gambar 6-1); tiga tombol di bagian atas untuk pukulan variabel, dan tiga tombol di bagian bawah untuk tendangan variabel, masing-masing dengan progresi rendah/sedang/tinggi dari kiri ke kanan.



Gambar 6-1. Pengalaman input arcade enam tombol asli untuk Street Fighter II

Setelah rilis rumah untuk PlayStation asli, tata letak empat tombol pada gamepad menggunakan dua tombol bahu untuk menggantikan dua tombol wajah yang tersisa, seperti yang diilustrasikan pada Gambar 6-2.



Gambar 6-2. Tata letak empat tombol pada gamepad untuk Sony PlayStation

Perkembangan intuitif tingkat kekuatan dari kiri ke kanan tidak diterjemahkan dengan mulus, dengan tombol berat dipindahkan ke area bahu kanan dan menggunakan ibu jari kanan untuk menekan serangan ringan atau sedang, dan jari telunjuk kanan untuk menekan serangan berat.

Ini diselesaikan di iterasi waralaba di masa mendatang, seperti *Marvel vs. Capcom 2*. Game ini dirancang untuk lebih ramah port, menghilangkan serangan sedang dan mengurangi *Kekuatan Augmented dan Virtual Reality dalam Bisnis (Dr. Agus Wibowo)*

tombol serangan utama menjadi ringan dan berat, dengan dua tombol yang tersisa di set gerakan bantuan mitra baru, sehingga jatuh di luar model kekuatan linier di keenam tombol dan memungkinkan tombol bantuan untuk hidup secara independen dari dua lainnya tanpa merusak model mental. Tombol bantuan ini sekarang bisa duduk sebagai dua tombol kanan dari kontrol enam tombol, seperti yang ditunjukkan pada Gambar 6-3, atau mereka dapat dipetakan ke tombol bahu dari controller PlayStation.



Gambar 6-3. Konfigurasi empat tombol plus dua tombol yang dapat diskalakan untuk Marvel versus Capcom 2 yang dapat disesuaikan dengan tata letak controller 3 x 2 dan 4 x 4 + 2

Kesimpulan utama: pikirkan tentang pengalaman sambil mempertimbangkan beberapa opsi adaptasi yang tersedia. Apa yang perlu terjadi dalam pengalaman, lalu bagaimana kita mengakomodasi skema input dalam jumlah terbesar untuk dimasukkan ke dalam sistem kontrol itu? Jawab pertanyaan-pertanyaan tersebut untuk mengambil langkah pertama ke dalam desain input yang skalabel.

Menyederhanakan Input Controller Salah satu judul peluncuran VR paling awal, Job Simulator oleh Owlchemy Labs, dibuat untuk menjadi salah satu game VR yang paling mudah diakses dengan memiliki input yang sangat disederhanakan: satu tombol untuk sebagian besar interaksi utama. Tujuan dari gameplay ini adalah untuk mengeksplorasi keajaiban VR melalui serangkaian tugas yang biasa dilakukan dalam VR dengan pendekatan yang aneh dan menyenangkan sebagai elemen tambahan dari keceriaan dan kesenangan.

Pengembang di Owlchemy Labs telah menawarkan beberapa insight dan pembicaraan yang dapat dipetik di konferensi dan online, dan salah satu poin utama yang diberikan adalah bahwa VR memungkinkan penghapusan banyak input controller tradisional karena tindakan tersebut sekarang diganti dengan menggunakan tubuh kita yang sebenarnya di dunia nyata. Sementara video game tradisional mengontrol gerakan dan tindakan protagonis dengan menggunakan satu atau bahkan dua joystick arah, semua gerakan itu sekarang dapat

ditangani dengan memutar kepala dan benar-benar berjalan. Berjongkok atau melompat sekarang bisa dilakukan langsung oleh tubuh alih-alih menggunakan tombol pada pengontrol. Ini membebaskan skema kontrol kompleks yang secara tradisional ditemukan pada game konsol dan memungkinkan controller tangan untuk fokus hanya pada memanipulasi item di dunia.

Dengan menetapkan satu tombol pemicu untuk menangani pengambilan, pelemparan, dan manipulasi item, Job Simulator adalah salah satu game tercepat untuk diambil dan dimainkan tanpa orientasi apa pun. Lebih penting lagi, ini berlaku untuk berbagai usia yang telah mencobanya, dari 4 tahun hingga 80 tahun ke atas. Cara mereka mencapai ini adalah melalui satu pendekatan yang sangat jenius: menempatkan interaksi ke dalam dunia dan membiarkan input tersebut dimanipulasi oleh tangan pemain. Menekan tombol, memutar kenop, menarik pegangan kabinet terbuka, dan melempar benda semuanya dilakukan dengan mensimulasikan dunia nyata, memungkinkan naluri alami seseorang untuk membimbing mereka melalui pengalaman. Alih-alih menetapkan satu set kontrol kompleks ke controller tangan, mereka menjadi perpanjangan tangan pemain itu sendiri, dan User interface untuk dipelajari semuanya ada di dunia.

Pemisahan ini memungkinkan desain objek yang lebih bijaksana dan lebih fleksibel dalam memiliki satu set antarmuka kontrol dengan berbagai antarmuka, seperti microwave, blender, wastafel, lemari es, mesin kasir, dan bahkan mesin mobil. Dalam jangka panjang, pendekatan ini jauh lebih terukur. Kami menyelami sekarang dengan solusi tingkat tinggi untuk memberikan contoh bagaimana pengembangan lintas platform dapat membuat peralihan platform menjadi lebih mudah dalam jangka panjang.

Langkah Pengembangan 1: Merancang Antarmuka Dasar

Anda dapat melakukan desain objek di VR dengan memikirkan cara kerja objek di dunia nyata. Apakah ada tombol yang melekat padanya seperti remote control? Apakah itu membuka dan menutup seperti kotak? Apakah itu memiliki kehadiran fisik atau apakah itu penanda simbolis (seperti bola cahaya)? Apakah bisa diambil atau ditempelkan pada benda lain?

Pertimbangkan semua cara di mana seseorang akan berinteraksi dengan suatu objek; misalnya saklar lampu. Saat ditekan, saklar lampu berubah posisi dan tindakan dilakukan, dalam hal ini lampu dinyalakan atau dimatikan. Saklar lampu itu dapat dimanipulasi dengan berbagai cara: saklar itu dapat langsung dinyalakan oleh tangan di dekat saklar lampu, atau Anda dapat mengambil tongkat, yang akan bertindak sebagai perpanjangan tangan, sehingga mendorong saklar itu secara semi-jauh. Atau, daya bohlam juga dapat dikontrol oleh sistem rumah pintar, sehingga kekuatan eksternal seperti aplikasi pada perangkat layar sentuh seluler dapat memanipulasi status nyala/mati lampu.

Sekarang mari kita terjemahkan itu ke dalam apa yang kita ketahui tentang berbagai jenis kontrol yang tersedia untuk VR. Dengan controller tangan 6DOF, Anda dapat langsung meletakkan tangan Anda di sebelah saklar dan mengklik tombol untuk membalikinya. Misalkan pengontrolnya adalah 3DOF dan tidak dapat memindahkan posisinya di luar angkasa tetapi dapat berputar. Penunjuk laser yang terpasang pada controller dapat bertindak seperti

ekstensi tongkat fisik, dan ketika laser menunjuk ke sakelar, mengklik tombol dapat mengaktifkan sakelar. Tanpa pengontrol, seperti dalam platform tracking kepala 3DOF seperti perangkat Cardboard, laser dapat dipasang ke kepala, dan melihat langsung ke sakelar dan mengetuk tombol dapat menekan sakelar. Jika tidak ada tombol pada perangkat Cardboard, menatap sakelar dapat memulai pengatur waktu singkat yang mengukur berapa lama pandangan ditahan, dan ketika pengatur waktu mencapai panjang yang telah ditentukan, sakelar mati sendiri. Terakhir, di perangkat seluler, sakelar dapat diketuk langsung dari layar sentuh atau bahkan dapat ditarik ke menu di layar dan dikendalikan dari jarak jauh seperti aplikasi rumah pintar.

Dalam contoh ini, setiap platform VR dan AR dapat berinteraksi dengan sakelar lampu dengan caranya sendiri, dengan sakelar lampu merespons perintah sederhana seperti "hidupkan", "matikan", dan "matikan".

Anda bisa mengambil ini lebih jauh. Saklar lampu kami memiliki dua status: hidup dan mati. Bagaimana jika ada lebih banyak status, seperti intensitas cahaya dengan kisaran 0 hingga 100? Sekarang, kita membutuhkan cara untuk mengontrol input variabel. Di dunia nyata, kami menangani ini dengan sakelar peredup. Alih-alih menjadi sakelar yang dibalik, dimmer sering diimplementasikan sebagai bilah geser di panel. Jadi, sakelar lampu memerlukan kontrol tombol tambahan, tetapi yang ini akan memiliki fitur tambahan untuk dapat diangkat dan dipindahkan. Kemudian, dengan mengklik tombol, alih-alih sakelar mengaktifkan / menonaktifkan keadaan bola lampu, dimmer akan mengubah keadaan terhubung ke kontrol atau terputus.

Dalam keadaan terhubung, gerakan controller (apakah itu controller tangan, penunjuk laser, atau kepala untuk lampiran tatapan) kemudian akan menggerakkan penggeser dan mengubah nilai intensitas bola lampu secara real time hingga terputus. Dengan tombol atau layar sentuh, tindakan "turun/didorong" kemungkinan akan menghubungkan objek dengan tindakan "naik/lepas", memutuskan sambungan objek. Dalam pendekatan tatapan-dan-tatapan, setiap tindakan tatapan-dan-tatapan perlu dilakukan sepenuhnya satu kali untuk terhubung dan sekali lagi untuk memutuskan sambungan.

Tanpa pengontrol, Anda dapat menggunakan kepala untuk mengarahkannya dengan melihat sakelar, dan jika tidak ada pengontrol, tatapan lama dapat mengaktifkannya. Pada akhirnya sakelar lampu akan memiliki jenis antarmuka "tekan tombol" yang akan diaktifkan oleh sejumlah skema kontrol agar berfungsi.

Kami sekarang telah mendefinisikan dua antarmuka, satu untuk seleksi dan satu untuk meraih, dan kami telah menunjukkan cara kerjanya di semua jenis input platform. Kita dapat memperluas ini ke dalam konsep yang lebih kompleks seperti berikut:

- Cara menempelkan objek di Unity: transformasi langsung, sambungan tetap, dan gaya fisika
- Manipulasi objek dengan dua tangan untuk gerakan dan skala
- Manipulasi sekunder objek, seperti mengisi ulang senjata atau melepas sumbat botol
- Membatasi pergerakan benda yang dipasang, seperti kenop pintu yang berputar

Sayangnya, menyelidiki ini secara mendalam berada di luar cakupan bab ini. Tetapi Anda dapat melihat bagaimana ada banyak cara untuk menyelesaikan tantangan antarmuka. Dengan dua properti antarmuka dasar ini, Anda dapat mengisi lingkungan dunia dengan objek eksplorasi yang dapat berinteraksi dengan apa pun platform yang digunakan. Bagian terbaiknya adalah semua pekerjaan itu dapat dilakukan bahkan tanpa meninggalkan komputer dan memasang headset. Saat itu sudah siap, kami mulai menyiapkan bagaimana integrasi platform diperlakukan.

Langkah Pengembangan 2: Integrasi Platform

Integrasi platform melibatkan pengambilan aplikasi dan melampirkan bagian perangkat keras ke dalamnya. Ini dapat dilakukan secara langsung, tetapi desain lintas platform memperlambat kami sedikit saat kami menyiapkan lapisan abstraksi yang solid terlebih dahulu. Jika Anda sudah bermain-main dengan membangun prototipe terhadap platform VR, bagian ini adalah langkah kedua yang bagus. Jika Anda belum pernah menulis tentang platform VR, Anda masih dapat mengikuti untuk memahami lapisan abstraksi yang kami siapkan di sini. Ada dua bagian utama untuk berintegrasi dengan platform.

- Memasang kepala
- Memasang input kontrol

Kepala biasanya langsung dan, dalam beberapa kasus, ditangani secara otomatis oleh mesin permainan menggunakan kamera utama sebagai input kepala. Kit pengembangan perangkat lunak platform (SDK) mungkin menyediakan skrip kustom atau prefab objek mereka sendiri di Unity untuk melampirkan beberapa fungsionalitas dasar ke kepala, seperti merender bingkai untuk kamera di AR seluler.

Memasang input kontrol bisa jauh lebih kompleks berdasarkan berbagai sistem interaksi. Misalnya, sistem HTC Vive dapat menggunakan plug-in OpenVR atau SteamVR Valve, sedangkan controller Oculus Touch menggunakan OVRPlugin dari Oculus. Google memiliki kode sendiri untuk controller Daydream, yang meskipun memiliki konfigurasi yang mirip dengan controller Oculus Go, menggunakan skrip pengelola input yang berbeda. Contoh yang disediakan untuk setiap platform cenderung bekerja secara langsung dengan kode SDK khusus untuk melampirkan fungsionalitas pengontrol, yang bagus untuk dipelajari, tetapi untuk tujuan kita, kita akan berintegrasi pada tingkat yang berbeda.

Agar jelas, fungsi controller hadir dalam dua bagian: melacak posisi dan/atau orientasi objek, dan memantau tombol atau input sentuh. Melacak posisi/rotasi objek sederhana di Unity, dan kode kita akan dilampirkan ke objek virtual yang memetakan ke objek dunia nyata 1:1. Ini adalah input tombol/sentuh yang kami minati untuk latihan ini. Pertama-tama mari kita lihat seperti apa jalur kode “Tekan tombol menu untuk membuka menu” pada contoh untuk SDK platform VR dan AR:

```
[ Frame Entrypoint: controller default ]
```

```
For each frame:
```

```
    Did the user click down on a controller’s menu button?
```

```
    If so:
```

```
        If the menu is not visible, open the menu (make it visible)
```

Otherwise, close the menu

Ini sangat sederhana dan mudah. Mengapa kita tidak melakukan ini saja? Implementasi kode aktual dari baris Apakah pengguna mengklik tombol menu pengontrol? kemungkinan besar bergantung pada platform dan dibangun dengan API controller SteamVR atau Oculus Rift. Ini menetapkan kontrol yang ditentukan ke tombol tertentu dalam kode, mencegah penyesuaian (misalkan pengguna individu ingin menggunakan tombol pemicu sebagai gantinya untuk menarik menu) dan tidak memiliki portabilitas (controller Daydream tidak akan memahami panggilan API.) Selain itu, kode ini ditulis khusus untuk menjalankan fungsi tunggal membuka menu. Jika fungsionalitas baru ditambahkan, itu juga perlu di-hardcode dalam urutan blok kode yang sama di sini. Atau, mari kita abstrak ini:

[Frame Entrypoint: controller menu behavior]

For each frame:

Is there a control input scheme to monitor? (such as 'open menu')

If so, is there a mapped control in a state we should care about?

If so, respond

Di sini, kami telah memperkenalkan konsep skema input kontrol. Ini akan menjadi objek yang mendefinisikan satu set fungsi yang dapat dilampirkan ke controller ini. Dalam hal ini, fungsi "menu terbuka" akan dapat disetel, tidak disetel, atau dialihkan. Kemudian, kami akan memetakan satu atau lebih kontrol ke sana, dan kode ini harus menjadi kode yang bergantung pada platform, seperti **map button_menu** pada **SteamVR** controller ke fungsionalitas **open_menu**. Kemudian, garis aliran apakah ada kontrol yang dipetakan akan mencari objek **button_menu** controller SteamVR dan memeriksa untuk melihat apakah itu dalam keadaan yang penting untuk fungsi ini. Status ini bisa berupa "tombol ke bawah", "tombol ke atas", atau "klik tombol". Kemudian, jika status tombol itu berhasil, itu akan mengaktifkan kontrol untuk membuka atau menutup menu.

Ada empat skrip yang dimainkan di sini. Salah satunya adalah pemetaan controller yang memeriksa interaksi pada setiap frame. Sebut ini **ControllerModule**. Ini sederhana dan menyediakan logika untuk menangani loop bingkai dan murni platform independen. Kelas kedua adalah kelas skema kontrol; sebut saja **ControlScheme**. Ini mendefinisikan fungsionalitas khusus aplikasi yang tersedia, dan itu akan dibangun per aplikasi sesuai kebutuhan. Kelas ketiga adalah skema pemetaan kontrol—misalnya, **MappingMenu ButtonToViveController**—yang akan dibuat satu kali untuk setiap port platform untuk menjembatani **ControlScheme** dan kelas terakhir, yang merupakan implementasi pengontrol. Kelas seperti ini dapat disebut **ViveController** dan menangani pemeriksaan untuk setiap status tombol pada pengontrol, tidak memiliki pengetahuan tentang fungsionalitas aplikasi dan hanya berfungsi sebagai antarmuka ke controller itu sendiri. Bersama-sama, mereka terlihat seperti ini:

[ControllerModule] – written once ever

[ControlScheme] – written once per application

[ControllerMapping] – written once per application per ported platform

[ControllerImplementation] – written once per platform controller

Dengan pengaturan ini, **ControllerModule** kemudian dapat ditulis sekali sebagai bagian dari kode kerangka kerja. Kelas lainnya akan memiliki kelas akar fungsionalitas dasar, dengan kelas anak yang diimplementasikan sesuai kebutuhan.

ControlScheme akan menjadi implementasi konkret yang ditulis sekali per skema interaksi untuk suatu aplikasi. Beberapa contoh selain berinteraksi adalah mengambil, menggambar, dan memilih. Semua ini dapat memiliki mode respons yang berbeda untuk pengontrol. Grab akan mengambil, menggambar akan mengeluarkan seni ke dunia, dan memilih dapat memilih objek untuk dimanipulasi.

Implementasi **ControllerMapping** akan membuat koneksi jembatan antara skema kontrol dan implementasi pengontrol, menentukan tombol apa pada setiap controller yang akan dilampirkan ke bagian fungsionalitas mana. Dalam pengaturan ini, pemetaan controller yang dapat ditentukan pengguna juga dapat dibuat untuk memungkinkan pengguna saat runtime untuk merutekan tombol mana yang ingin mereka tetapkan ke fungsi apa.

Terakhir, implementasi controller akan ditulis satu kali per jenis controller di setiap platform. Ini akan menangani kode khusus platform untuk memantau pemicu tombol, input analog, dan sebagainya. Jadi bagaimana semua pekerjaan ekstra ini membantu kita? Sekarang mari kita lampirkan fungsionalitas "berinteraksi" ke pengontrol. Ini adalah diagram alur baru: [Frame Entrypoint: controller that can interact]

For each frame:

 If there is a nearest hovered object:

 Is there a control input scheme to monitor? (such as 'interact')

 If so, is it in a state the object should respond to?

 If so, respond

 Otherwise:

 Is there a control input scheme to monitor? (such as 'open menu')

 If so, is it in a state we should care about?

 If so, respond

Tidak banyak yang berubah di sini, kecuali sekarang skrip bingkai memantau fungsionalitas yang dilampirkan ke objek melayang. Skrip ini sekarang dapat mengaktifkan sakelar lampu jika tombol yang sesuai dipilih, atau beralih menu sebaliknya. Perhatikan di sini bahwa pemetaan kontrol dapat menetapkan "berinteraksi" ke pemicu sementara "menu terbuka" tetap melekat pada tombol menu.

Mari kita melangkah lebih jauh dan menambahkan kemampuan untuk mengambil objek. Perhatikan bahwa jika suatu objek diambil, kita mungkin ingin melakukan sesuatu dengan objek itu seperti melemparnya, memutarnya, dan seterusnya, sehingga logikanya muncul terlebih dahulu:

[Frame Entrypoint: controller that can pick up an object]

For each frame:

 If there is a connected object:

 Is there a control input scheme to monitor? (such as 'shoot' or 'drop')

 If so, is it in a state the object should respond to?

If so, respond

Otherwise if there is a nearest hovered object:

Is there a control input scheme to monitor? (such as 'interact')

If so, is it in a state the object should respond to?

If so, respond

Otherwise:

Is there a control input scheme to monitor? (such as 'open menu')

If so, is it in a state we should care about?

If so, respond

Bisakah Anda melihat polanya? Kami dapat beralih melalui skema kontrol dan beralih ke fungsionalitas controller dasar jika ada. Ini scalable untuk membuat fungsionalitas pengontrol. Selain itu, mari kita lihat cara mengadaptasi controller yang berbeda untuk ini. Jika kami ingin mem-port aplikasi ini ke Daydream, kami memerlukan skrip DaydreamController baru untuk menangani implementasi controller dan skrip mimpi MappingMenuButtonToDay baru untuk menggantikan pemetaan lainnya. Selain itu, kita akan selesai.

6.4 CONTOH DUNIA NYATA

Memindahkan prototipe dari Daydream ke Oculus Go menarik karena satu-satunya perbedaan di kedua jenis controller itu tampaknya adalah tombol pemicu. Dengan demikian, satu prototipe membutuhkan dua tombol yang dapat disesuaikan: satu untuk menjalankan fungsi utama dan satu lagi untuk menjalankan fungsi sekunder untuk mengaktifkan mode fungsi utama. Kemudian, kita dapat beralih dari mode cat ke mode pemilihan dengan satu tombol sambil menggunakan tombol utama untuk benar-benar melakukan pengecatan atau pemilihan. Pada controller Go, pemicu adalah tombol yang paling tepat untuk menangani fungsi utama, meninggalkan klik touchpad untuk melakukan pertukaran mode sekunder. Namun, pada controller Daydream, tidak ada pemicu, dan dengan demikian input utamanya adalah klik touchpad, yang merupakan input sekunder kami pada controller Go! Tidak masuk akal untuk menggunakan controller Go untuk input sekunder, jadi controller Daydream menggunakan tombol aplikasi untuk input sekunder untuk mengaktifkan mode input utama. Meskipun ini membuat platform berbeda, itu adalah masalah sederhana mengedit file pemetaan untuk bermain dengan opsi yang berbeda. Contoh kode disediakan di repositori GitHub buku ini.

Teori yang diberikan dimaksudkan untuk membantu memulai pemikiran tingkat tinggi dalam desain VR dan AR dan untuk membantu mendapatkan hasil maksimal dari informasi teknis berikut. Dalam bab ini, kami menjelaskan dasar-dasar mesin game dan kamera 3D, 3DOF versus 6DOF, dan mengapa pendekatan desain harus dipertimbangkan pada tingkat tinggi holistik berdasarkan tujuan pengalaman. Konsep abstraksi lintas platform akan membantu memungkinkan kode Anda beradaptasi dengan platform lain dengan sangat mudah, yang merupakan elemen dasar inti dari multipemain serta pengalaman sosial waktu nyata. Ini karena konten dunia akan konsisten di seluruh platform tanpa masalah apa pun yang diberikan pada input kontrol asimetris.

Selain informasi dalam bab-bab berikut, berikut adalah beberapa sumber daya pengembangan lintas platform tambahan yang mungkin dapat membantu:

- Torch3d: alat pengembangan VR dan AR kolaboratif
- BridgeXR: toolkit lintas platform di Unity
- Unity AR Framework: kerangka kerja AR lintas platform oleh Unity
- Wikitude: kerangka kerja AR lintas platform untuk Unity

Pada bab berikutnya, Vasanth memberikan contoh desain yang lebih konkret untuk VR dan AR, termasuk teknik penggerak serta informasi lebih lanjut tentang jenis kontrol yang tersedia. Sebagai mantan pemodal ventura di luar angkasa, saya sering ditanya apakah VR dan AR adalah mode yang akan hilang. Jawaban saya tidak pernah berubah: VR dan AR adalah masa depan komputasi yang tak terhindarkan. Apa yang tidak diketahui adalah berapa lama waktu yang dibutuhkan bagi kita untuk sampai ke sana, baik itu 2 atau 20 tahun. Namun, setiap upaya yang dapat kami lakukan untuk mempercepat pertumbuhan itu akan membantu kami mencapai masa depan di mana kami secara alami berbagi keajaiban VR dan AR dengan lebih banyak teman dan keluarga kami. Itu adalah masa depan yang membuat saya bersemangat, dan saya berharap kita bisa bersama-sama segera tiba di sana.

BAB 7

VIRTUAL REALITY TOOLKIT (VRTK)

OPEN SOURCE FRAMEWORK

Virtual Reality Toolkit (VRTK) adalah open source, toolkit lintas platform untuk membangun pengalaman virtual reality (VR) dengan cepat dengan menyediakan solusi yang mudah digunakan untuk masalah umum. VRTK berfokus pada dua bidang bantuan utama bagi pengembang: interaksi dan teknik penggerak, menawarkan banyak cara untuk memecahkan masalah umum ini.

7.1 APA ITU VRTK DAN MENGAPA ORANG MENGGUNAKANNYA?

VRTK adalah basis kode sumber terbuka yang memungkinkan pengguna untuk menarik dan melepas fungsionalitas. Dengan dapat menarik dan melepaskan Aset ke dalam Unity 3D, dengan beberapa konfigurasi, mereka dapat segera membuka contoh Scene dengan mekanisme permainan penting yang siap pakai dan berfungsi penuh seperti penggerak, navmeshes, berbagai User interface, dan interaksi fisik di mana Anda dapat mulai membangun game Anda.

Dengan keuntungan menjadi open source, siapa pun dapat segera mengurangi waktu penyiapan mereka dan segera mulai menyesuaikan aset dan kode sumber untuk mewujudkan ide-ide mereka di Unity untuk pembuatan prototipe cepat—setidaknya.

Manfaat utama dari toolkit ini adalah bahwa ini adalah satu-satunya dari jenisnya yang siap beradaptasi dengan perangkat keras apa pun yang Anda rencanakan untuk dikembangkan: Controller Oculus + Touch, HTC Vive, headset *Mixed Reality* (MR), headset VR seluler. Karena aksesibilitas perangkat keras dapat menjadi penghalang bagi banyak calon pembuat VR atau pembuat VR baru, default penggunaannya mencakup simulator VR. Simulator VR memungkinkan pembuat konten untuk dapat membangun game imersif yang berfungsi penuh di Unity bersama dengan pratinjau yang memungkinkan input keyboard untuk menggantikan penggunaan controller dalam menavigasi pengalaman.

Tujuan VRTK adalah membawa sebanyak mungkin orang kreatif dari berbagai latar belakang yang berbeda untuk mencoba memecahkan masalah umum yang dibawa oleh media baru VR. Semakin cepat solusi untuk masalah ini dapat dibangun, dicoba, dan diuji, semakin cepat kita dapat membantu mempercepat proses evolusi untuk mengetahui apa yang berhasil dan apa yang tidak, karena evolusi hanyalah sejuta kesalahan sampai sedikit dari satu hal berhasil.

Cara VRTK memberdayakan partisipasi yang begitu besar adalah dengan sepenuhnya gratis dan open source (di bawah lisensi MIT) bagi siapa saja yang ingin menggunakannya, untuk alasan apa pun mereka ingin menggunakannya, baik itu untuk mempelajari tali

pengembangan VR atau mereka menggunakannya untuk membuat game VR terbaru berikutnya, atau bahkan jika mereka adalah entitas komersial yang membuat solusi simulasi.

Dengan menjadikannya sepenuhnya gratis bagi siapa saja untuk digunakan atau dibangun, penghalang untuk masuk ke pengembangan VR telah diturunkan secara besar-besaran sehingga mereka yang ingin mengubah ide kreatif dan liar mereka menjadi kenyataan dapat memiliki kesempatan itu dengan VRTK.

7.2 SEJARAH SINGKAT VRTK

Selama akhir pekan di bulan April 2016, Harvey mulai mengambil semua pengetahuan yang diperoleh dari skrip plug-in SteamVR dan mencoba mengubahnya menjadi sesuatu yang dapat memudahkan orang lain untuk membangun sesuatu di VR. Skrip, yang hanya membutuhkan waktu sekitar dua hingga tiga jam untuk ditulis, adalah satu skrip untuk Unity3d yang diseret dan dijatuhkan ke dalam Scene dengan rig kamera SteamVR dan segera memberikan kemampuan untuk menyinari penunjuk laser di dalam Scene dan berteleportasi ke mana pun. ujungnya menunjuk. Itu juga memberi kemampuan untuk mengambil item menggunakan sistem perampasan berbasis sambungan tetap yang belum sempurna. Ini hebat: Anda dapat membuat Scene, dan dengan satu tarik dan lepas cepat dari skrip open source, Anda dapat bergerak di sekitar Scene dan mengambil sesuatu dan melemparkannya ke mana-mana. Langkah selanjutnya adalah membagikannya kepada dunia, jadi di saluran YouTube yang tidak dikenal dengan tidak lebih dari 100 pelanggan dan hampir tidak ada penayangan reguler, sebuah video diposting yang menunjukkan cara menggunakan skrip VR ini.

Setelah beberapa hari, video tersebut memiliki ribuan penayangan, dengan jelas menunjukkan bahwa ada banyak orang dalam situasi yang sama, ingin menemukan tutorial tentang cara membuat konten untuk VR. Kami segera menyadari bahwa jika ada kebutuhan dan keinginan untuk konten semacam ini, skrip dasar dan tipis ini bukanlah cara terbaik bagi orang untuk bergerak maju. Itu terlalu membatasi, terlalu digabungkan (artinya satu skrip melakukan segalanya sehingga menyesuaikannya akan sangat merepotkan), dan itu bukan sesuatu yang bisa dikerjakan orang bersama-sama dalam sebuah komunitas.

Setelah keberhasilan skrip tunggal asli ini, SteamVR Unity Toolkit lahir, yang merupakan upaya yang lebih terpadu untuk mencoba membangun kumpulan skrip yang dapat digunakan kembali dan diperluas yang membuat pembuatan untuk HTC Vive lebih mudah dan lebih cepat tidak hanya untuk pengembang berpengalaman, tetapi juga untuk pemula yang ingin mencoba tetapi tidak tahu apakah mereka bisa melakukannya. SteamVR Unity Toolkit adalah nama yang sangat tepat karena pada dasarnya adalah toolkit skrip yang dibangun di Unity3d yang membantu saat menggunakan plug-in SteamVR, menawarkan kumpulan solusi seperti teleportasi, pointer, meraih, dan penggerak touchpad. Karena semuanya benar-benar gratis dan open source, itu mulai mendapatkan daya tarik dengan orang-orang yang membangun konten yang menarik bagi mereka di VR, beberapa dari konten ini akan menjadi beberapa game paling terkenal di VR.

Itu adalah zaman perkembangan VR Wild West; tidak ada yang benar-benar tahu masalah apa yang akan dilontarkan oleh medium, dan tidak ada yang benar-benar memiliki

jawaban untuk solusinya. SteamVR Unity Toolkit menjadi repositori GitHub di mana orang dapat berbagi ide mereka untuk solusi dengan berkontribusi pada basis kode dan menyampaikan ide mereka kepada orang lain dalam upaya untuk memudahkan orang lain membangun pengalaman dan game VR mereka sendiri. Faktanya, pengembang di belakang QuiVR menyumbangkan sejumlah fitur keren untuk VRTK, terutama Scene contoh busur-dan-panah, yang mengilhami sejumlah permainan busur-dan-panah yang menyenangkan untuk dibuat oleh pengembang pemula.

Seiring bertambahnya jumlah pengembang yang menggunakan SteamVR Unity Toolkit, menjadi semakin sulit untuk membantu orang dengan masalah individu mereka. Ketika hanya ada segelintir orang yang menggunakannya, cukup mudah untuk melakukan panggilan Skype dengan seseorang dan menyelesaikan masalah mereka, tetapi ketika ada sekitar seribu orang yang menggunakan sesuatu, ini tidak akan pernah berakhir dengan baik.

Komunitas di balik toolkit telah berkembang pesat dalam waktu yang singkat dan membutuhkan suatu tempat yang dapat dengan bebas berkomunikasi dan menetaskan ide bersama. Solusinya adalah saluran Slack di mana siapa pun dapat bergabung dan berkontribusi, mencari bantuan, dan mengobrol tentang ide-ide yang pada akhirnya dapat menjadi fitur perangkat untuk digunakan orang lain.

Saluran Slack masih menjadi pusat komunitas hingga saat ini, dengan lebih dari 4.500 orang di seluruh dunia bekerja mengatasi masalah dan berbagi ide tentang cara membuat pengalaman VR lebih menarik bagi pemirsa mereka. Ini telah menjadi tempat di mana orang-orang membentuk ikatan komunitas nyata, pertemanan online, atau kemitraan ke dalam usaha baru untuk membangun beberapa game VR yang sangat keren.

Orang-orang di komunitas merasa sangat bersemangat untuk membagikan ide-ide mereka, ini memperkuat SteamVR Unity Toolkit sebagai alat upaya komunitas daripada karya satu orang, yang hanya membantunya tumbuh pada tingkat yang terus meningkat dengan semakin banyak pengalaman sedang dibuat.

Itu telah berkembang sedemikian rupa sehingga telah diperhatikan oleh beberapa perusahaan VR berpengalaman dengan perwakilan Oculus menanyakan apa yang dapat mereka lakukan untuk membuatnya berfungsi dengan headset mereka. Oculus cukup baik untuk menyediakan paket controller Oculus Rift dan Touch gratis sehingga Harvey dapat membuat toolkit bekerja pada headset selain Vive. Dalam beberapa hari, itu adalah toolkit multiheadset dengan manfaat tambahan bahwa jika sesuatu dibangun untuk bekerja di Vive, sekarang juga akan bekerja cukup mulus dengan Oculus Rift. Toolkit ini sekarang juga menjadi lapisan abstraksi software development kit (SDK) yang sangat hilang dari produk Unity3d.

Namun, ada masalah kecil dengan toolkit yang bekerja pada Oculus Rift: namanya. SteamVR Unity Toolkit tidak masuk akal lagi karena tidak hanya untuk SteamVR sehingga komunitas memutuskan untuk mengganti nama proyek menjadi Virtual Reality Toolkit, atau disingkat VRTK.

Komunitas membawanya ke depan dengan membangun lebih banyak fitur keren seperti memanjat, mengayunkan lengan, berbagai jenis mekanika meraih, mulai dari yang

menggunakan fisika untuk memindahkan objek, hingga teknik yang lebih sederhana seperti menjadikan objek sebagai anak pengontrol. .

Tetapi masalah lain muncul: toolkit ini berasal dari bagaimana SteamVR diatur dan bagaimana cara kerjanya. Integrasi Oculus SDK benar-benar hanya lapisan abstraksi di atas cara kerja bagian dalam SteamVR. Harvey menyadari bahwa ini akan menyebabkan masalah yang lebih besar ketika headset dan teknologi lain akan dirilis. Tidak semua bisa bergantung pada dasar-dasar SteamVR, karena ini hanya akan menjadi perbedaan mendasar bagaimana headset lain di masa depan dapat berperilaku.

Pada saat ini, VRTK sudah melakukan begitu banyak dan begitu banyak orang dengan senang hati menggunakannya untuk membangun segala macam hal yang indah, tetapi jelas bahwa itu perlu dipikirkan kembali, ditata ulang, dan dibangun kembali dari bawah ke atas dengan cara yang berbeda secara fundamental. tidak terikat pada bagian teknologi tertentu. Karena cara VRTK meledak dalam popularitasnya, tidak ada banyak waktu untuk keputusan pengecekan akal atau fondasi arsitektur, dan karena didasarkan pada warisan dukungan SteamVR, itu berarti basis kode terus berkembang di sekitar konsep itu. Semakin banyak kode yang ditambahkan oleh grup kontributor yang terus berkembang, semakin sulit untuk mempertahankan dan memperluasnya. Pada titik itu, jelas terlihat bahwa VRTK perlu ditulis ulang, sampai ke pertimbangan desain fundamentalnya.

7.3 VRTK V4

VRTK v4 akan menjadi pendekatan yang benar-benar baru untuk pengalaman toolkit. Daripada skrip prebuilt yang melakukan hal tertentu, itu akan menjadi pola desain mendasar yang dapat disusun dalam berbagai konfigurasi untuk menyediakan fungsionalitas yang bermanfaat bagi VR (atau kasus penggunaan lainnya dalam hal ini). Ini sangat penting karena itu berarti apa pun yang berubah dengan teknologi di masa depan, toolkit akan ada di sana untuk mendukungnya. Pengembang dapat membangun dengan mudah untuk perangkat keras pemula apa pun yang mencoba untuk berhasil di pasar yang berarti lebih banyak game dan pengalaman dapat mendukungnya, yang hanya akan membantu keberhasilan proses evolusi apa pun.

Pekerjaan pada VRTK v4 dimulai pada akhir April 2018 dengan pendekatan yang benar-benar baru tentang cara kerja toolkit untuk pengembang. Juga, kurangnya ketergantungan pada fitur inti Unity3d berarti bahwa masa depan untuk toolkit dapat, dan dicita-citakan, meluas ke platform lain seperti Unreal Engine, WebVR, dan Godot, untuk beberapa nama. Premis ini bahkan lebih menarik untuk potensi VRTK: jika pengembang dapat memahami dasar-dasar VRTK dan cara membuat solusi dengan menggunakannya di Unity3d, seharusnya tidak ada yang bisa menghentikan mereka untuk mentransfer pengetahuan itu ke platform lain. Satu-satunya pemblokir adalah mempelajari cara menggunakan antarmuka platform lain, tetapi kemampuan untuk memilih dan memilih tampilan yang dipasang di kepala (HMD) dan mesin yang akan dibangun akan sangat bermanfaat bagi semua pengembang.

Salah satu hasrat besar tim dan komunitas VRTK adalah memastikan bahwa pengembangan VRTK dan VR dapat diakses oleh sebanyak mungkin orang. VRTK sudah

digunakan di banyak hackathon, lokakarya, dan lembaga pendidikan seperti sekolah menengah dan universitas untuk mengajarkan pengembangan VR kepada pencipta gelombang baru. Bagaimana VRTK v4 juga dapat menyelaraskan kekuatan toolkit baru dengan mendidik mereka yang mungkin sudah menjadi pengembang berpengalaman dengan mereka yang tidak memiliki pengalaman sama sekali tetapi ingin belajar? Dengan demikian, konsep kurikulum VRTK dirancang. Ada pertanyaan yang perlu dijawab, meskipun: apakah mungkin untuk memiliki kumpulan panduan, tutorial, video, dan materi pembelajaran yang membantu untuk mengajarkan kekuatan VRTK tetapi juga secara konsisten dan menyediakan berbagai tingkatan yang tergantung pada keahlian pengguna dapat memiliki titik masuk yang layak dan dapat dimengerti?

7.4 MASA DEPAN VRTK

Masa depan VRTK tidak hanya menyediakan platform bagi pemula untuk memulai perjalanan pengembangan mereka, tetapi juga untuk membantu dan dengan cepat meningkatkan proses pengembangan untuk pengembang berpengalaman, dari rumah indie hingga AAA. Menyediakan alat yang andal dan teruji untuk mencegah mereka dari keharusan menemukan kembali roda berarti ide-ide baru dapat dibuat prototipe dengan cepat untuk menentukan apakah mekanik mereka berfungsi. Kemampuan untuk fokus pada konten dan bukan mekanik berarti bahwa pengembang dapat lebih berupaya untuk menghasilkan konten yang sangat halus, memajukan selera pasar yang masih baru, tetapi juga dengan kekuatan dasar VRTK v4 berarti bahwa pengembang ini dapat lebih lanjut menyesuaikan dan memperluas solusi dasar untuk memberikan pengalaman yang lebih unik.

Kemampuan untuk membuka aksesibilitas VR ke perusahaan juga merupakan misi penting bagi VRTK. Untuk memungkinkan industri dengan cepat dan murah mencoba solusi VR untuk masalah sehari-hari berarti bahwa penggunaan komersial VR akan lebih cepat, menghasilkan lebih banyak investasi untuk media baru yang luar biasa ini untuk berkembang dan makmur.

Dengan penuh harapan dan cinta dari komunitas VRTK, VRTK akan terus mendukung pengembangan VR sebagai media dan bahkan meluas ke masa depan untuk mendukung sektor komputasi spasial lainnya seperti *augmented reality* (AR). Masa depan terlihat cerah, dan konsep seputar VRTK v3 adalah untuk menyediakan satu skrip yang memberikan fungsionalitas tertentu. Meskipun ini memudahkan untuk menjalankan sesuatu hanya dengan menyeret dan menjatuhkan skrip, itu berarti bahwa menyesuaikan komponen fungsionalitas akan memerlukan perluasan skrip dan bahkan berpotensi menulis potongan kode yang besar. VRTK v4 bertujuan untuk memecah fungsionalitas yang dapat digunakan menjadi komponen umum yang memiliki tanggung jawab melakukan satu pekerjaan tertentu, dan komponen kecil ini kemudian digabungkan untuk membentuk cetakan yang melakukan fungsionalitas yang sama dengan skrip tunggal. Prefab ini yang berisi komponen yang relevan dihubungkan menggunakan peristiwa sehingga jika ada bagian dari jalur eksekusi yang perlu diubah atau diubah, itu dapat dengan mudah menyambungkan pendengar baru pada peristiwa, yang paling sering mengakibatkan tidak ada pengkodean yang benar-benar perlu dilakukan.

Manfaat dari subkomponen ini di VRTK v4 adalah bahwa banyak fungsi yang dapat digunakan kembali dapat tersebar di banyak kasus penggunaan yang berbeda, apakah itu memindahkan objek atau mendeteksi tabrakan. Ini juga berarti bahwa kode inti yang mendasari VRTK v4 tidak ada hubungannya dengan VR sama sekali, sehingga dapat digunakan untuk tujuan apa pun, baik itu VR, AR, atau bahkan hanya pengalaman desktop atau seluler. Prefab yang berada di atas kode inti menyediakan fungsionalitas khusus, memungkinkan persyaratan baru apa pun dengan mudah dipenuhi hanya dengan menyusun komponen generik bersama-sama dalam campuran yang berbeda untuk menyediakan apa pun yang diperlukan.

Masalah lain dalam VRTK v3 adalah asal-usulnya dibangun di sekitar cara kerja SteamVR, yang berarti semuanya pada dasarnya adalah lapisan di atas pengaturan SteamVR untuk semua SDK lain yang didukung. Mendukung hal-hal yang tidak memiliki kesamaan yang jelas dengan SteamVR sangat sulit. VRTK v4 tidak memiliki dasar di SDK apa pun dan oleh karena itu benar-benar generik dan harus dapat mendukung sejumlah perangkat dengan relatif mudah. Contoh yang baik adalah sesuatu seperti memindahkan pemain di sekitar Scene; di V3 ini dikenal sebagai touchpad walking dan akan mengambil data sumbu dari touchpad Vive (atau thumbstick Oculus Touch) dan mengubahnya menjadi data arah. Ini berfungsi dengan baik, tetapi selalu diharapkan bahwa informasi arah ini akan datang dari SDK sehubungan dengan touchpad atau yang setara. Ini berarti bahwa apa pun yang hanya ingin memasukkan informasi arah ke dalam skrip pergerakan pemain harus melalui seluruh pipa SDK untuk mencapainya. Di VRTK v4, karena tidak ada ketergantungan pada pengetahuan intrinsik semacam itu, sangat mudah untuk membuat "Aksi" yang memancarkan Vector2 yang berisi data arah dan kemudian operasi dapat dilakukan pada Vector2 untuk mengubahnya di sepanjang jalan, seperti mengalikan elemen itu seperti ingin membalikkan arah y, data bahkan dapat diubah menjadi tipe data lain seperti float atau Boolean.

Karena pendekatan generik baru dalam VRTK v4 yang memanfaatkan peristiwa untuk menyampaikan pesan antar subkomponen, jauh lebih mudah untuk membuat fungsionalitas khusus tanpa perlu menulis kode apa pun. Ada juga keuntungan menggunakan alat skrip visual untuk membuat fungsionalitas menggunakan drag and drop sederhana. Ini adalah langkah maju yang bagus bagi mereka yang tidak berasal dari latar belakang pengkodean tetapi masih ingin menciptakan pengalaman unik tanpa perlu mempelajari pengkodean yang mendasarinya.

7.5 KEBERHASILAH VRTK

Sejak kemunculan VRTK, ada lebih dari 30.000 unduhan toolkit dan telah digunakan di berbagai proyek mulai dari pengembang indie solo hingga studio game AAA. Gambar 7-1 menunjukkan hanya sebagian kecil dari judul yang diterbitkan yang memuji VRTK karena secara cepat mengurangi waktu pengembangan menuju produksi yang tersedia di semua platform utama, termasuk Oculus Store dan Steam.

Made With VRTK



Gambar 7-1. Berikut adalah beberapa proyek yang berhasil menggunakan VRTK. Banyak game tersedia di VRTK

Anda dapat menemukan daftar lengkap game yang diterbitkan yang menggunakan VRTK online.

7.6 MEMULAI DENGAN VRTK V4

VRTK adalah kumpulan skrip dan demonstrasi konsep yang berguna untuk membantu membangun solusi VR dengan cepat dan mudah. Ini bertujuan untuk membuat membangun solusi VR di Unity3d dengan cepat dan mudah bagi pemula dan pengembang berpengalaman. VRTK mencakup sejumlah solusi umum seperti berikut:

- Penggerak dalam ruang virtual
- Interaksi seperti menyentuh, meraih, dan menggunakan objek
- Berinteraksi dengan elemen UI Unity3d melalui pointer atau sentuhan
- fisika tubuh dalam ruang virtual
- Kontrol 2D dan 3D seperti tombol, tuas, pintu, laci, dan sebagainya

7.7 MENYIAPKAN PROYEK

Berikut adalah langkah-langkah yang perlu Anda ambil untuk menyiapkan proyek Anda:

1. Buat proyek baru di Unity3d 2018.1 atau lebih tinggi menggunakan Template 3D.
 - a. Pastikan bahwa kotak centang yang Didukung Virtual Reality dipilih.
 - b. Di menu utama Unity3d, klik "Edit," lalu "Pengaturan Proyek," lalu "Pemutar."
 - c. Di panel inspektur PlayerSettings, perluas Pengaturan XR.
 - d. Pilih kotak centang opsi Virtual Reality Supported.
2. Perbarui proyek ke Scripting Runtime Version yang didukung.
 - a. Di menu utama Unity3d, klik "Edit," lalu "Pengaturan Proyek," lalu "Pemutar."
 - b. Di panel inspektur PlayerSettings, perluas Pengaturan Lainnya.
 - c. Ubah Scripting Runtime Version menjadi .NET 4.x Equivalent.
 - d. Unity sekarang akan dimulai ulang di runtime skrip yang didukung

Cloning Repositori

Mengkloning repositori Berikut cara mengkloning repositori VRTK ke dalam proyek Anda:

1. Arahkan ke direktori Aset/ proyek.
2. Git mengkloning submodul yang diperlukan ke dalam *asset*/direktori:
 Git clone --recurse-submodules <https://github.com/thestonefox/VRTK.git>
 git submodule init && git submodule update

7.8 MENJALANKAN TES

Buka Scene VRTK/Scenes/Internal/TestRunner:

1. Di menu utama Unity3d, klik “Window”, lalu “Test Runner”.
2. Pada tab EditMode, klik Jalankan Semua.
3. Jika semua tes lulus, instalasi Anda berhasil.

Menyiapkan lingkungan Anda

1. Unduh versi terbaru VRTK dari repositori GitHub (lihat Tabel 7-1) di www.vrtk.io (atau [www.github.com/thestonefox/VRTK](https://github.com/thestonefox/VRTK)).

Tabel 7-1. SDK yang didukung

Supported SDK	Link Download
VR Simulator	Termasuk
Steam VR	https://www.assetstore.unity3d.com/en/#!/content/32647
Oculus	https://developer.oculus.com/downloads/package/oculus-utilities-for-unity-5/
Ximmerse*	https://github.com/Ximmerse/SDK/tree/master/Unity
Daydream*	https://developers.google.com/vr/unity/download

Jika Anda tidak memiliki akses ke Headset VR, atau ingin membangun secara agnostik untuk tujuan pembuatan prototipe, SDK tidak diperlukan untuk menjalankan pratinjau pengembangan game Anda di Unity 3D

- a. VRTK saat ini hanya dapat diakses melalui Command Line. Jika Anda menggunakan PC, buka Command Prompt Anda. Di Mac, buka Terminal.
- b. Salin dan tempel perintah berikut di editor:
 git clone --recurse-submodules
- c. Tekan Enter dan tunggu perintah dijalankan sebelum melanjutkan.
- d. Masukkan perintah berikut di editor:
 git submodule init && git submodule update
- e. Tekan Enter dan tunggu perintah dijalankan.
- f. Opsional: Unduh SDK untuk perangkat keras yang Anda inginkan.
- g. Impor Folder Aset VRTK 4 ke dalam proyek Unity 3D Anda.
- h. Buka *Assets/VRTK/Examples*, lalu buka salah satu Scene Contoh, tekan Mainkan untuk melihat bagaimana interaksi terlihat di Scene Game Anda.

7.9 CONTOH SCENE

Kumpulan contoh Scene telah dibuat untuk membantu Anda memahami berbagai aspek VRTK. Ini adalah tempat yang bagus untuk memulai jika Anda menggunakan VRTK untuk pertama kalinya—atau bahkan yang ke-50 juta kali—karena VRTK juga berfungsi sebagai titik awal yang bagus untuk pembuatan prototipe cepat atau permulaan proyek dasar.

Scene contoh adalah lingkungan yang siap disiapkan untuk fungsionalitas instan dengan SDK pilihan Anda. Masing-masing Scene ini diberi judul berdasarkan jenis fungsi yang akan diperlihatkan Scene tersebut. Scene contoh dapat dengan mudah diduplikasi dan disesuaikan ke dalam proyek Anda dan mereka mendukung semua SDK VR yang didukung VRTK.

Anda dapat melihat daftar lengkap contoh di Contoh/README.md, yang mencakup daftar contoh terkini yang menampilkan fitur VRTK. Untuk menggunakan perangkat VR (selain Simulator VR yang disertakan), impor VR SDK pihak ketiga yang diperlukan ke dalam proyek.

7.10 CARA “MEMERIKSA” REPOSITORI CONTOH VRTK V4

Berikut adalah langkah-langkah yang diperlukan untuk memeriksa Repositori Contoh VRTK v4:

1. Saat ini, VRTK hanya dapat diakses melalui baris perintah. Jika Anda menggunakan PC, buka Command Prompt Anda. Jika Anda menggunakan Macintosh, buka Terminal Anda.
2. Salin dan tempel baris berikut ke dalam editor:
`git clone --recurse-submodules https://github.com/thestonefox/VRTK.git`
3. Tekan Enter dan tunggu hingga selesai.
4. Pada baris baru, ketik perintah berikut:
`git submodule init && git submodule update`
5. Tekan Enter dan tunggu sampai selesai
6. Opsional: Unduh SDK untuk perangkat keras yang Anda inginkan.
7. Buka Asset/VRTK/Examples, buka salah satu Scene Contoh, lalu tekan putar untuk melihat bagaimana interaksi terlihat di Scene Game Anda.

Berikut adalah daftar contoh Scene dan fitur interaksi saat ini (sampai tulisan ini dibuat):

Input Scene

Menampilkan informasi yang diberikan oleh controller atau input keyboard Anda ke dalam game. Scene penunjuk objek

Object pointer scene

Menggunakan laser hijau yang dipancarkan oleh controller Anda. Anda dapat mengarahkan laser ke berbagai objek di tempat kejadian dan melihat reaksi berbeda yang dapat ditunjukkan oleh penunjuk Anda saat diarahkan ke objek tertentu.

Straight pointer

Penunjuk adalah emisi laser lurus dasar Anda. Paling baik digunakan untuk membuat pilihan UI, atau berinteraksi dengan objek.

Bezier pointer

Emisi garis lengkung yang mengarah ke tanah—ini bisa dibbilang pengalaman pengguna terbaik untuk teleportasi.

Teleportasi titik-dan-klik

Menggunakan pointer, Anda dapat memilih area yang ingin Anda pindahkan dan dalam "berkedip" Anda akan diposisikan ulang ke lokasi itu.

Instant teleport

Ini adalah penggunaan "bingkai hitam" yang menyerupai kedipan, di mana pengguna akan berakhir di lokasi baru ketika penglihatan telah kembali.

Dash teleport

Arahkan ke area yang ingin Anda teleportasi, dan kemudian akan mempercepat frame gerakan bagi pengguna untuk tiba di sana. Gerakan ini muncul dan terasa lebih alami daripada metode Instant teleport.

Teleport scene

Menunjukkan berbagai area dan tipe area yang dapat Anda teleportasikan dalam Scene dengan mengklik Thumbstick untuk mengaktifkan teleportasi dan menunjuk ke arah atau titik yang ingin Anda pindahkan dengan mengklik tombol pemicu.

Interactive Object (Objek yang dapat berinteraksi)

Teleport di sekitar tempat kejadian ke berbagai objek. Gunakan tombol pegangan pada controller untuk melihat berbagai jenis pegangan pada setiap objek. Berikut adalah beberapa contoh jenis interaksi yang tersedia pada saat rilis di folder aset VRTK:

Precision grab

Grabs di lokasi yang tepat dari objek tertentu. Dalam contoh pistol, di mana pun tangan berada pada objek, Anda dapat mengambilnya secara otomatis dengan cara tertentu untuk meningkatkan kemudahan penggunaan.

Gun grab

Meraih objek berbentuk pistol akan selalu diposisikan dalam posisi siap menembak di tangan saat diambil dari sudut mana pun.

Toggle grab

Memungkinkan Anda untuk melepaskan tombol saat objek dicengkeram dan mempertahankan posisi meraih hingga Anda menekan tombol lagi.

Two-handed hold

Memegang benda dalam posisi fungsional ketika kedua tangan telah meraihnya.

Pump

Memungkinkan penggunaan aksi pemompaan pada suatu objek untuk menghasilkan efek tertentu.

Hinge joint

Hanya bergerak pada sumbu tertentu—misalnya, pintu berayun terbuka dan tertutup.

Dalam mode putar, Anda dapat memilih CameraRig Switcher untuk bergantian di antara berbagai jenis SDK yang Anda gunakan untuk pengalaman Anda atau memilih simulator. Ini sangat berguna, jika bukan penggunaan VRTK yang paling menguntungkan, karena Anda dapat

membuat beberapa headset dan melihat pratinjau tampilannya secara real time selama tahap pengembangan untuk masing-masing hanya dengan mengklik tombol.

Cara menyiapkan proyek Inti VRTK dari awal

1. Dalam proyek Unity 3D 2018.1+ Anda, buka Scene kosong.
2. Pilih kotak centang yang Didukung Virtual Reality.
3. Buka Project Settings → Player → Other Settings.. Di bawah Konfigurasi, ubah Scripting Runtime Version menjadi .Net 4x Equivalent.
4. Saat diminta, tekan Mulai Ulang. 5. Unduh paket VRTK.Unity.Core dari GitHub, lalu seret dan lepas Folder Aset ke dalam proyek Unity 3D Anda.

Cara menyiapkan Unity CameraRig

1. Dalam Scene Anda, buka tab "Hierarki", lalu hapus Default Camera.
2. Buka Assets → VRTK Unity.Core → CameraRig → [UnityXRCameraRig]. Drag [UnityXRCameraRig] Prefab ke dalam tab Hierarchy Anda.¹
3. Tekan putar pada Scene untuk melihat pratinjau kamera.

Head Anchor

Objek game induk mengacu pada posisi headset.

Left Anchor

Objek permainan anak yang merujuk pada lensa mata kiri headset.

Right Anchor

Objek permainan anak yang merujuk ke lensa mata kanan headset.

Cara mengatur Alias Terlacak

1. Seret cetakan "Alias Terlacak" ke dalam Hirarki Anda.
2. Buka Aset → VRTK.Unity.Core → CameraRig → TrackedAlias.
3. Objek game alias yang dilacak adalah objek game anak yang dapat Anda sesuaikan untuk interaksi pengguna yang diwujudkan yang disesuaikan dengan sensor yang akan digunakan dalam kaitannya dengan jenis perangkat keras.

Play area alias

Mengacu pada ruang fisik yang akan dilacak oleh sensor perangkat keras untuk pengalaman

Headset alias

Mengacu pada posisi headset

Left controller alias

Mengacu pada controller tangan kiri

Right controller alias

Mengacu pada controller tangan kanan

Scene Camera

Objek game ini merujuk pada berbagai kamera yang akan diposisikan dalam pengalaman (baik untuk perspektif orang pertama atau ketiga)

Tutorial lain yang tersedia di situs web pada saat rilis ini akan mencakup:

¹ Sampai tulisan ini dibuat, VRTK v4 masih dalam pengembangan. Lihat vrk.io untuk pembaruan terbaru dalam dokumentasi dan tutorial, untuk memulai

- Cara mengatur Simulator
- Pengantar Tindakan VRTK
- Cara mengatur pointer
- Cara mengatur teleportasi dengan pointer
- Cara mengatur objek yang dapat berinteraksi (interactor/interactables)

BAB 8

TIGA PRAKTEK TERBAIK *AUGMENTED REALITY* DAN *VIRTUAL REALITY*

8.1 BERKEMBANG UNTUK *VIRTUAL REALITY* DAN *AUGMENTED REALITY* ITU SULIT

Dan mungkin itulah sebabnya Anda membaca buku ini sejak awal. Tetapi penting untuk memahami kerumitannya sebelum terjun ke pengembangan. Jadi, mari kita uraikan dulu apa yang membuat pengembangan jauh lebih rumit daripada kebanyakan bidang.

Mari kita mulai dengan alatnya. Sepanjang bab ini, kami bekerja dengan mesin game Unity. Awalnya dirilis pada tahun 2005, Unity telah membantu banyak pengembang di seluruh dunia untuk mulai membangun game tiga dimensi, mulai dari seluler, konsol, hingga desktop. Dan meskipun telah berfungsi sebagai tulang punggung untuk pengembangan 3D bagi banyak orang dan telah memupuk komunitas yang luar biasa selama bertahun-tahun, itu tidak berarti sempurna, terutama karena paradigma desain untuk *virtual reality* (VR) dan *augmented reality* (AR) terus berkembang. Sejak kit pengembangan VR modern pertama kali dirilis pada tahun 2013, alat bawaan Unity dan *plug-in* eksternal telah meningkat secara signifikan, tetapi tugas tertentu seperti pengembangan lintas platform dan multipemain masih tidak sesederhana mengaktifkan tombol. Ingatlah hal itu saat Anda melanjutkan bab ini.

Selanjutnya, perangkat keras. Lebih dari sekadar alat, jumlah perangkat keras yang berbeda dapat sangat meningkatkan kompleksitas. Dari Oculus Rift hingga PlayStation VR hingga iPhone yang menjalankan ARKit, setiap perangkat memiliki serangkaian batasannya sendiri yang perlu dioptimalkan berdasarkan kasus per kasus untuk memenuhi persyaratan unik perangkat. Meskipun ini bukan hal baru jika Anda berasal dari latar belakang grafis atau game, setiap perangkat memiliki serangkaian tombol dan persyaratan tracking unik yang perlu diintegrasikan ke dalam cara setiap aplikasi tertentu dikembangkan.

Dan yang terakhir adalah perawatan. Seperti yang disebutkan sebelumnya, VR dan AR adalah bidang yang berkembang. Dengan demikian, baik alat maupun perangkat keras terus berubah dengan kecepatan yang luar biasa. Unity merilis perubahan besar kira-kira setiap tiga bulan, dan headset atau alat baru untuk headset yang ada dapat berubah lebih cepat dari itu. Ini mengharuskan Anda untuk selalu memperbarui kode, terkadang bahkan sebelum merilis pengalaman Anda. Ini mungkin memakan waktu, tetapi penting untuk memastikan bahwa pengalaman Anda berjalan dengan sukses untuk semua orang.

Sekarang, saya tahu bahwa ada banyak masalah negatif di lapangan, tetapi ada titik terang di ujung terowongan. VR dan AR adalah pengembangan paling berharga yang dapat Anda lakukan, menurut saya. Ini mungkin tidak mudah dan pasti akan ada titik frustrasi di sepanjang jalan, tetapi ketika Anda melihat orang-orang memakai headset dengan senyum di wajah mereka, itu sangat bermanfaat. Dan alasan mengapa saya memulai bab ini dengan pendahuluan yang panjang itu adalah untuk memperjelas batasan-batasan dalam pengembangan dan benar-benar menekankan batasan-batasan yang akan Anda kerjakan. Ada solusi untuk sebagian besar masalah ini melalui perencanaan yang cermat dan dengan bekerja

dengan tim desain untuk menciptakan ruang lingkup yang memberikan pengalaman menarik dan menyembunyikan semua masalah yang disebutkan di atas.

Pengembangan VR dan AR itu sulit, jadi melangkah maju ke bab ini, mari pelajari bagaimana kita dapat menggunakan beberapa tip dan trik untuk membuatnya lebih mudah. Dengan semua itu, mari selami tiga praktik pengembangan yang dapat Anda gunakan dalam VR dan AR.

8.2 PENANGANAN LOKOMOSI

Pertama-tama, kita lihat bagaimana membangun beberapa jenis mekanisme penggerak untuk VR dan AR. Penggerak bisa sangat sederhana tetapi merupakan mekanik yang sangat penting untuk pengalaman apa pun. Ini memungkinkan pengembang untuk mengambil dunia tanpa batas dan membuatnya dapat dilalui dalam ruang terbatas; misalnya, kamar Anda. Ada banyak cara untuk menyelesaikan penggerak, dan jenis penggerak yang Anda pilih akan sering ditentukan oleh apa yang menurut audiens Anda paling mendalam. Pada bagian ini, kami membangun tiga jenis penggerak yang berbeda: gerakan linier, teleportasi, dan gerakan berskala.

Sebelum kita mulai membangun, saya ingin menyebutkan beberapa sistem penggerak penting yang tidak akan kita bangun tetapi dapat berguna untuk pembelajaran di masa mendatang:

Redirect walking

Teknik grafis yang sedikit mendistorsi gambar yang dirender ke headset untuk membuat pengguna berpikir bahwa mereka berjalan dalam garis lurus, padahal kenyataannya mereka berjalan di jalan yang melengkung. Anda membutuhkan area yang luas untuk mengelabui otak.

Dashing

Memindahkan pengguna dengan cepat ke tujuan mereka dalam waktu singkat; misalnya 0,5 detik. Keuntungannya adalah pengguna memiliki rasa yang lebih baik dalam menyelam sambil mengurangi potensi penyakit simulasi.

Climbing

Seorang pengguna menggunakan tangan mereka untuk menarik diri ke arah yang diinginkan, seringkali dengan memegang objek virtual.

Controller Assisted On the Spot (CAOTS)

Menggunakan controller terlacak posisi saat bergerak di tempat untuk bergerak secara virtual. Tracking ini digunakan dalam aplikasi Freedom Locomotion VR yang terdaftar secara gratis di Steam.

1 banding 1

Jika Anda dapat mengatur agar semua yang diperlukan untuk pengalaman Anda berada dalam jangkauan pemain, penggerak mungkin tidak diperlukan sama sekali, seperti di Job Simulator. Dan ini dapat diambil satu langkah lebih jauh di mana ruang virtual beradaptasi dengan seberapa besar ruang pemain, membuat pengalaman menjadi lebih mudah diakses.

Penggerak di VR

Sebelum kita mulai, penting untuk dicatat bahwa jenis penggerak yang diintegrasikan ke dalam aplikasi sangat bergantung pada aplikasi itu sendiri. Implementasi yang akan datang adalah yang paling umum di seluruh game VR, tetapi dengan itu, mereka mungkin tidak tepat untuk apa pun yang Anda coba buat. Namun demikian, implementasi ini berharga untuk diketahui, terutama saat Anda membuat prototipe untuk melihat apa yang berhasil dan apa yang tidak.

8.3 GERAKAN LINIER (ALIAS GERAKAN TRACKPAD)

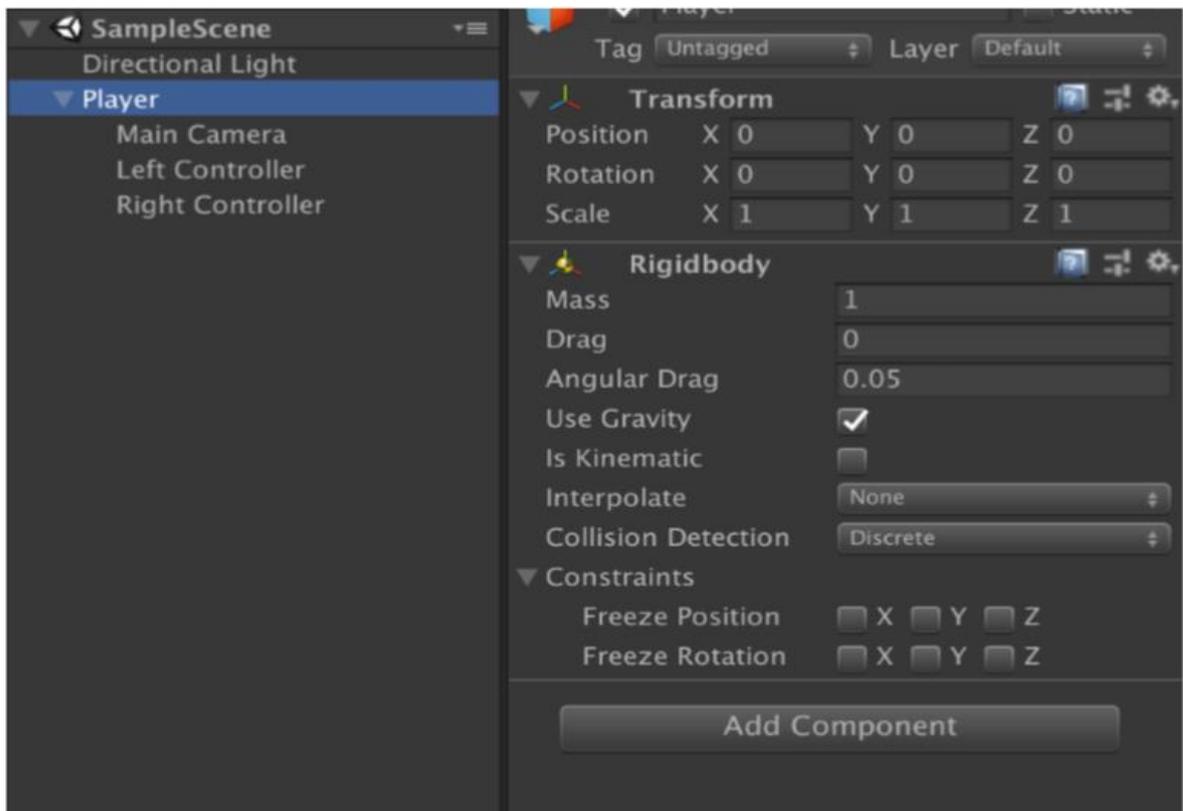
Selain Google Cardboard, semua sistem VR modern dilengkapi dengan semacam pengontrol, seperti yang ditunjukkan pada Gambar 8-1, dan semua controller ini dilengkapi dengan joystick atau trackpad.

Mengetahui hal ini, kita dapat membuat sistem gerakan 2D sederhana menggunakan joystick atau touchpad sebagai input. Jika Anda terbiasa dengan game first-person-shooter, ini akan menjadi mekanisme gerakan yang sangat mirip dengan game-game tersebut.

Untuk memulai, pertama-tama kita perlu mengatur pemutar kita. Untuk ini, kami menggunakan simulasi fisika bawaan Unity, yang berarti bahwa kami perlu memasang komponen Rigidbody (lihat Gambar 8-2) ke pemutar kami. Beberapa manfaat menggunakan Rigidbody termasuk menambahkan gaya dan kecepatan dengan mudah ke objek serta mensimulasikan tumbukan fisika antara dua objek. Ini sangat berguna bagi kami karena kami ingin menggunakan kecepatan untuk menggerakkan pemain kami secara linier serta mendeteksi ketika pemain bertabrakan dengan tanah.

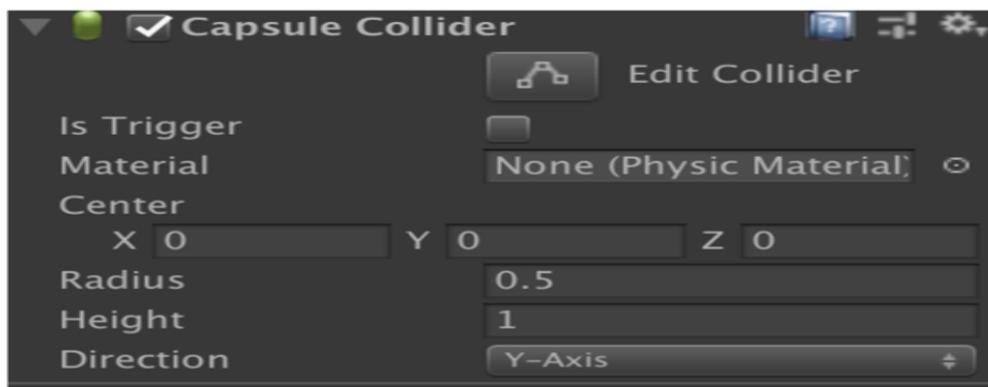


Gambar 8-1. Oculus Rift Controller (kiri) dan HTC Vive Controller (kanan)



Gambar 8-2. Rigidbody di pemutar VR kami

Selanjutnya, kita perlu menambahkan Collider untuk menentukan batasan dari player kita. Di sini, saya punya kabar baik dan kabar buruk. Kabar baiknya adalah ketika kita mendefinisikan batasan pemain, batasannya tidak harus sempurna. Penumbuk kapsul sederhana sudah cukup, yang merupakan penumbuk bawaan untuk Unity, seperti yang diilustrasikan pada Gambar 8-3, dan sangat dioptimalkan untuk kinerja.



Gambar 8-3. Penumbuk kapsul bawaan Unity (ditambahkan ke pemain)

Berita buruknya adalah untuk VR (dan AR), tidak seperti mendefinisikan batas tabrakan pemain dalam video game tradisional, tidak ada satu ukuran untuk semua tinggi badan untuk

setiap manusia yang memainkan game Anda. Ada beberapa perbaikan potensial untuk masalah ini:

- Mintalah para pemain untuk berdiri diam sebelum mereka mulai sehingga mereka dapat diukur selama sesi berlangsung
- Asumsikan bahwa tinggi pemain saat ini adalah tinggi maksimumnya

Meskipun tidak ada perbaikan yang optimal, tergantung pada kasus penggunaan Anda, satu solusi mungkin bermanfaat dibandingkan dengan yang lain. Satu hal yang perlu diperhatikan adalah bahwa di sebagian besar toolkit VR (SteamVR, VRTK, dll.), solusi kedua diimplementasikan secara default. Untuk melihat bagaimana mereka mengimplementasikannya, lihat pseudocode:

```
public Collider capsule; // set from the Unity Interface
public Transform player; // set from the Unity Interface
void AdjustCapsuleHeight() {
    var playerHeightOffset = player.localPosition; //player's height from ground
    capsule.height = playerHeightOffset; //set the height
    capsule.localPosition.y = -playerHeightOffset / 2;
    //because capsule pivot is in the center
}
```

Dengan pengaturan sistem fisika, sekarang kita dapat menyelami penciptaan gerakan linier. Sama seperti penumbuk kapsul, berikut adalah pseudocode (sekali lagi, Anda dapat menemukan solusi yang berfungsi di GitHub):

```
public Rigidbody rigidbody; // set from the Unity Interface
public float speed; // set from the Unity Interface
void LinearMovement() {
    Vector2 trackpad = null;
    if (Input.GetTouch( LeftTrackPad )) { //check if left
    trackpad is touched
    trackpad = Input.GetLeftPad(); //set left trackpad 2D position
    }
    else if (Input.GetTouch( RightTrackPad )) { //check if right trackpad is touched
    trackpad = Input.GetRightPad(); //set right trackpad 2D position
    }
    if (trackpad != null) {
        rigidbody.velocity =
    new Vector3(trackpad.x, 0, trackpad.y) * speed;
    //set XZ velocity, so we don't start flying
    }
    else {
        rigidbody.velocity = Vector3.zero; //when not pressed, set to 0;
    }
    }
```

Dan itu adalah semua yang Anda butuhkan untuk mengatur beberapa gerakan linier sederhana. Meskipun ini adalah mekanisme gerakan yang cukup sederhana untuk diatur, itu tidak berarti cocok untuk setiap penonton. Yang kami rekomendasikan adalah menyediakan sistem penggerak ini sebagai opsi bagi pengguna petualang (yang merupakan sebagian besar pengguna VR) dan kemudian memasukkan sistem penggerak berikutnya, teleportasi, sebagai sistem bagi mereka yang lebih sensitif terhadap penyakit simulator.

8.4 TELEPORT MOTION

Cukup banyak sejak Oculus Development pertama dikirimkan, teleportasi telah menjadi salah satu solusi paling sederhana, efektif, dan agak kontroversial untuk melintasi ruang virtual yang besar. Di satu sisi, ia menghindari banyak masalah yang dimiliki sistem penggerak lain dengan penyakit simulasi, menjadikannya yang paling mudah diakses. Tetapi tergantung pada jenis pengalaman yang Anda bangun, itu juga bisa kehilangan rasa pendalaman dengan sangat cepat. Yang mengatakan, ini adalah alat yang luar biasa untuk disimpan di ikat pinggang Anda karena lebih sering daripada tidak Anda ingin memasukkannya ke dalam pengalaman Anda. Jadi, mari kita membangunnya!

Ada beberapa jenis teleportasi yang berbeda, tetapi untuk tetap fokus pada apa yang sedang kita bangun, mari kita fokus pada salah satu jenis yang paling umum: teleportasi Bézier (atau melengkung), seperti yang ditunjukkan pada Gambar 8-4. Berikut adalah dua alasan mengapa jalur melengkung sering digunakan:

- Mereka membatasi seberapa jauh pemain dapat melakukan perjalanan, yang membatasi pemain melakukan perjalanan sepanjang jalan melintasi level.
- Ini mengurangi presisi yang dibutuhkan pemain untuk berakhir di lokasi yang diinginkan.

Untuk memulai, pertama-tama mari kita lakukan beberapa pengaturan. Kami ingin beberapa variabel untuk menyesuaikan teleportasi kami serta metode untuk membuat teleportasi kami. Untungnya, Unity memiliki Komponen *Line Renderer* bawaan yang sangat dapat disesuaikan untuk mendapatkan tampilan dan nuansa yang Anda inginkan.



Gambar 8-4. Teleportasi melengkung

Dengan pengaturan itu, kita bisa fokus memeriksa input untuk memulai teleportasi kita. Tergantung pada platform yang Anda kembangkan, kode ini dapat terlihat berbeda, terutama jika Anda memiliki 0, 1, atau 2 pengontrol, kodenya akan bervariasi, tetapi konsepnya adalah memilih tombol yang nyaman untuk sering ditekan oleh pemain, seperti trackpad pada controller Vive atau pemicu pada Oculus Rift. Setiap kali tombol itu ditahan, tunjukkan jalur melengkung untuk teleportasi, dan kemudian ketika dilepaskan, teleportasi ke lokasi itu

Untuk kenyamanan lebih, Anda juga dapat memilih untuk memudahkan tampilan pemain menjadi hitam dengan sangat cepat untuk meningkatkan kenyamanan teleportasi:

```
public Vector3 gravity; //set in inspector as (0, -9.8, 0)
public LineRenderer path; //the component that will render our path
private Vector3 teleportLocation; //save the location of where we want to teleport
private Player player; //the player we will be teleporting
void Update()
{
    //called every frame
    if (!CheckTeleport(LeftHand))
    { //check the left hand
        CheckTeleport(RightHand); //if not teleporting with left hand try right hand
    }
}
bool CheckTeleport(Hand hand) { //check a hand to see the status of teleporting
    List<Vector3> curvedPoints; //the points on the teleport curve
    if (hand.GetPressed(TrackPad)) {
        //check if track pad ( button for teleport) is pressed
        if (CalculateCurvedPath(hand.position, hand.forward, gravity, out curvedPoints))
        {
            //calculate teleport
            RenderPath(curvedPoints);
            //if calculate, render the path teleportLocation = curvedPoints[curvedPoints.Count - 1];
            //set teleport point return true;
        }
    } else if (hand.GetPressedUp(TrackPad)) { //time to actually teleport
        player.position = teleportLocation; //move the player instantly
    }
    return false; //we are not using this hand currently for teleporting
}
```

Sekarang mari kita lihat kode teleportasi yang sebenarnya. Metode utama adalah HitungCurvedPath, yang mengambil titik awal, arah, dan efek gravitasi dan kemudian menampilkan jalur melengkung. Dengan input tersebut, kita dapat menjalankan simulasi fisika sederhana untuk menghitung berapa lama waktu yang dibutuhkan kurva untuk mendarat di tanah. Untuk mempermudah, kami menganggap kecepatan hanya sebagai arah yang

dinormalisasi, meskipun Anda dapat mengubah nilai ini untuk mendapatkan efek yang berbeda:

```
bool CalculateCurvedPath(Vector3 position, Vector3 direction,
//calculates the teleportation path
    Vector3 gravity, out Vector3 points) {
int maxDistance = 500; //sets the max distance the path can travel
Vector3 currPos = origin, hypoPos = origin, hypoVel = direction.normalized;
//initialize variable to keep track off
List<Vector3> v = new List<Vector3>(); //list of points
RaycastHit hit; //gets raycast info at each step
float curveCastLength = 0; //current distance traveled do { //loop
    v.Add(hypoPos); //add start
    currPos = hypoPos; //set start position as previous end postion
    hypoPos = currPos + hypoVel + (gravityDirection * Time.fixedDeltaTime);
// calculate next point on curve
    hypoVel = hypoPos - currPos; //calulate the delta for the velocity
    curveCastLength += hypoVel.magnitude; // add velocity to distance
}
while (Raycast(currPos, hypoVel, out hit, hypoVel.magnitude) == false //check physics to see
if we hit the ground
    && curveCastLength < maxDistance);
    points = v; //return points
    return RaycastHit(currPos, hypoVel, out hit, hypoVel.magnitude); //check if landed
}
void RenderPath(List<Vector3> points) {
    path.pointCount = points.Count;
    for ( int i = 0; i < points.Count ; i++) {
        path.points[i] = points[i]; //set all points in Line Renderer
    }
}
```

Setelah jalur dihitung, kita bisa menetapkan semua titik ke penyaji garis, yang secara otomatis akan merender garis. Satu hal yang perlu diperhatikan di sini adalah Anda ingin kembali ke kode input dan mengaktifkan dan menonaktifkan penyaji baris setiap kali tombol ditekan dan dilepaskan. Jika tidak, poin pertama yang ditetapkan ke jalur akan selalu ada di sana, yang merupakan bug lucu tetapi mungkin tidak dimaksudkan. Dan dengan itu, kami memiliki teleportasi, sedikit lebih rumit daripada gerakan linier, tetapi itu pasti sepadan dengan banyak pengalaman.

8.5 LOCOMOTION DI AR

Kami baru saja membahas satu sistem penggerak yang sangat penting untuk VR. Sistem penggerak berikutnya bekerja dengan baik di dalam VR dan AR, tetapi ini sangat

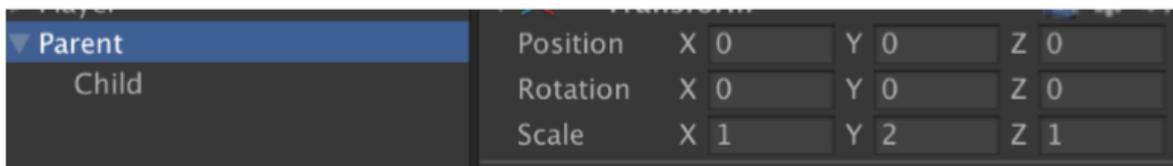
berdampak pada AR karena tidak ada banyak alat untuk bergerak karena dapat terlihat sedikit canggung jika objek virtual berubah posisi karena penggerak, sedangkan di dunia nyata mereka tetap tetap. Salah satu cara untuk menyiasatinya dengan AR adalah dengan menggunakan trik visual yang mengubah persepsi skala—dan ini sebenarnya cukup mudah untuk dicapai.

8.6 MODE RAKSASA (ATAU SEMUT)

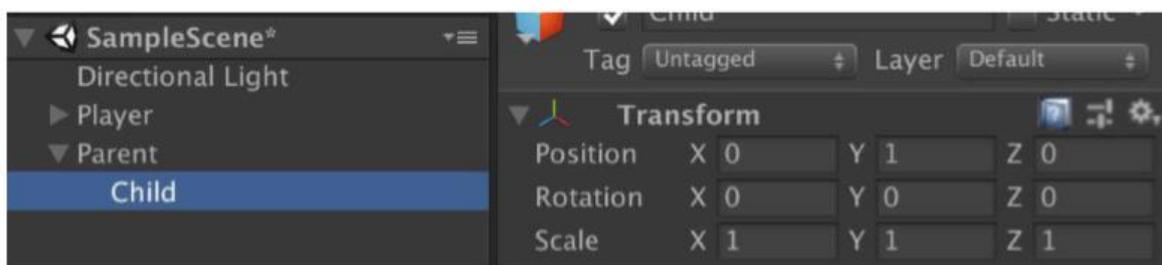
Salah satu nilai jual terbesar dari VR dan AR adalah dapat mengalami hal-hal baru, dan subpoinnya adalah mampu mengukur sesuatu dari sudut pandang baru; yaitu, menjadi sangat besar atau sangat kecil.

Salah satu cara untuk mewujudkannya adalah dengan menambah atau mengurangi skala setiap objek individu dalam sebuah Scene — masalah dengan pendekatan ini adalah pemain masih berukuran manusia dan, akibatnya, jarak yang ditempuh pemain akan tetap manusia. berskala.

Sebagai gantinya, untuk menutupi lebih banyak (atau lebih sedikit) tanah dan mengukur jarak yang ditempuh, seringkali akan lebih baik untuk mengukur skala pemain. Cara kerja skala di Unity (dan di hampir semua mesin game) adalah bahwa posisi dan nilai skala dari objek apa pun yang bersarang di bawah objek induk akan dikalikan. Untuk mengilustrasikannya, lihat Gambar 8-5 dan 8-6.



Gambar 8-5. Parameter transformasi objek induk



Gambar 8-6. Parameter transformasi objek anak

Transform dari objek Anak akan menjadi:

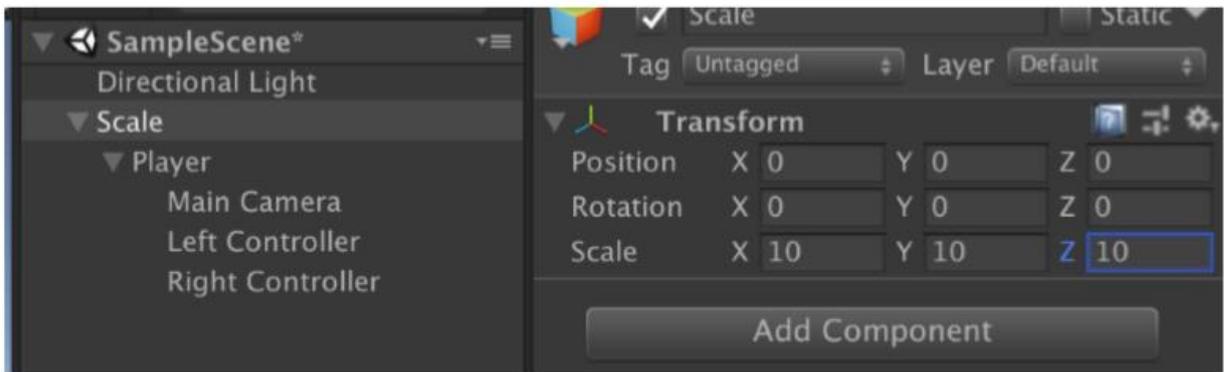
Position

Anak (0, 1, 0) dikalikan (perkalian titik) dengan Orang Tua (1, 2, 1) = (0, 2, 0)

Scale

Hasil perkalian anak (1, 1, 1) (perkalian titik) dengan orang tua (1, 2, 1) = (0, 2, 0)

Dengan menggunakan pengetahuan itu, kami dapat menerapkan ini ke pemutar AR kami. Yang perlu kita lakukan adalah membuat objek induk dengan skala lebih besar dari (1, 1, 1), seperti yang ditunjukkan pada Gambar 8-7.



Gambar 8-7. Menskalakan pemutar VR

Dan dengan itu, kami telah membuat pemain skala raksasa jika skala itu diatur ke lebih besar dari (1, 1, 1), dan semut jika skala itu kurang dari (1, 1, 1). Catatan penting yang perlu diingat adalah bahwa Anda dapat mengatur skala untuk setiap sumbu ke nilai yang berbeda, tetapi ini akan menyebabkan pengalaman yang sangat membingungkan. Saya sangat menyarankan agar Anda menjaga semua timbangan persis sama dan sebagai nilai positif untuk menghindari perilaku aneh.

Dan itu saja untuk penggerak. Seiring perubahan perangkat keras VR dan AR, berbagai jenis penggerak akan dicoba, dan mungkin saja beberapa dari sistem yang umum digunakan ini ketinggalan zaman. Tetapi kemampuan untuk membangun salah satu dan semua sistem ini akan berguna untuk memulai brainstorming cara-cara baru untuk menangani perkembangan baru di ruang angkasa.

8.7 PENGGUNAAN AUDIO YANG EFEKTIF

Ketika Anda mendengar istilah "VR" atau "AR", mana dari panca indera Anda yang muncul di benak Anda? Bagi kebanyakan orang, itu adalah visi—mampu melihat dunia dan mampu bereaksi terhadapnya. Audio hampir selalu menempati posisi kedua atau ketiga (di belakang sentuhan). Namun pada kenyataannya (dan Virtual Reality) audio sama pentingnya, jika tidak lebih penting, daripada visual dunia untuk menciptakan suasana. Bayangkan tiga skenario ini, Anda sedang berlayar di lautan maya seperti yang ditunjukkan pada Gambar 8-8, ketika tiba-tiba Anda mendengar:

- Musik disko sekitar
- Detak jantung dari belakangmu
- Tidak ada (pengembang merasa malas)



Gambar 8-8. Lautan yang luas

Dalam setiap skenario, apa yang terjadi di sebelah Anda tidak bergantung pada visual tetapi sangat bergantung pada apa yang Anda dengar atau tidak dengar. Singkatnya, audio bukanlah sesuatu untuk dilupakan atau diabaikan, jadi mari kita lihat bagaimana menerapkannya untuk mendapatkan hasil maksimal dari audio.

8.8 AUDIO DALAM VR

Di bagian ini, mari kita lihat lebih dekat bagaimana Unity memungkinkan pengembang untuk mengimplementasikan audio ke dalam aplikasinya. Metode saat ini secara tradisional digunakan oleh pengembang game 2D. Akibatnya, sebagai pengembang VR, kami perlu melakukan beberapa langkah tambahan untuk membuat audio di lingkungan VR terdengar lebih imersif.

Ambient versus 3D versus audio spasial

Dalam Unity, ada tiga jenis audio utama:

Sekelilingnya

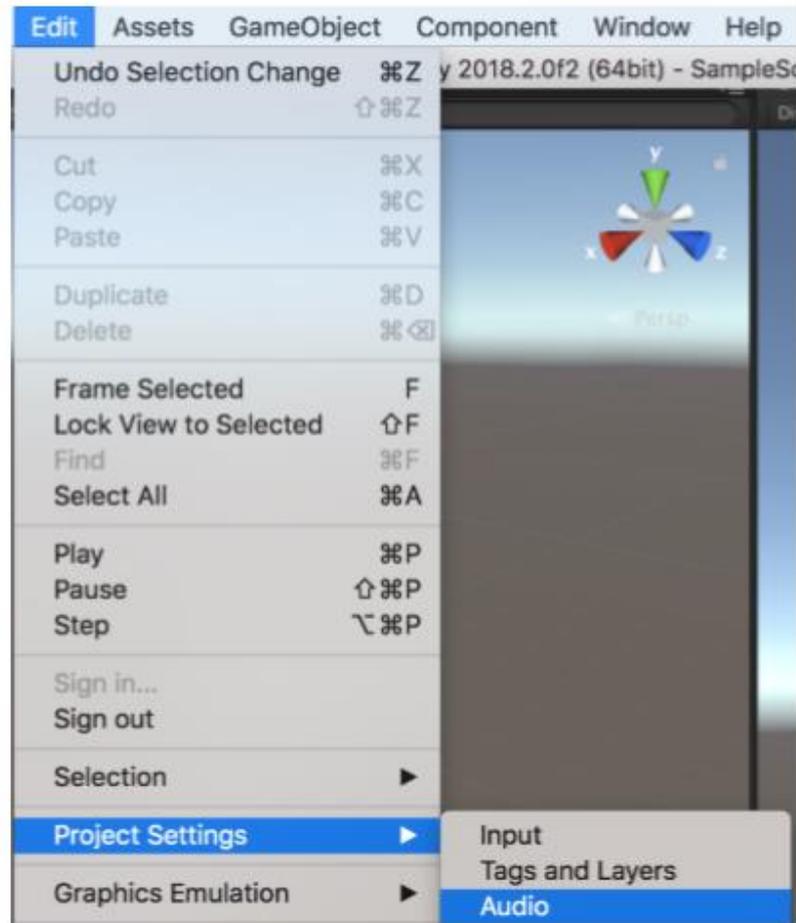
Audio 2D yang tidak bergantung pada lokasi pemutar. Pikirkan ini seperti kicauan burung yang jauh di hutan.

audio 3D

Audio yang terikat ke posisi 3D dan volumenya turun semakin jauh pemutar (pendengar audio) dari posisi itu.

Audio spasial (binaural)

Mirip dengan audio 3D, perbedaan utama adalah bahwa intensitas saluran kiri dan kanan audio bervariasi tergantung pada seberapa jauh objek dari masing-masing telinga, seperti dalam kehidupan nyata.



Gambar 8-9. Jalur ke pengaturan audio Unity

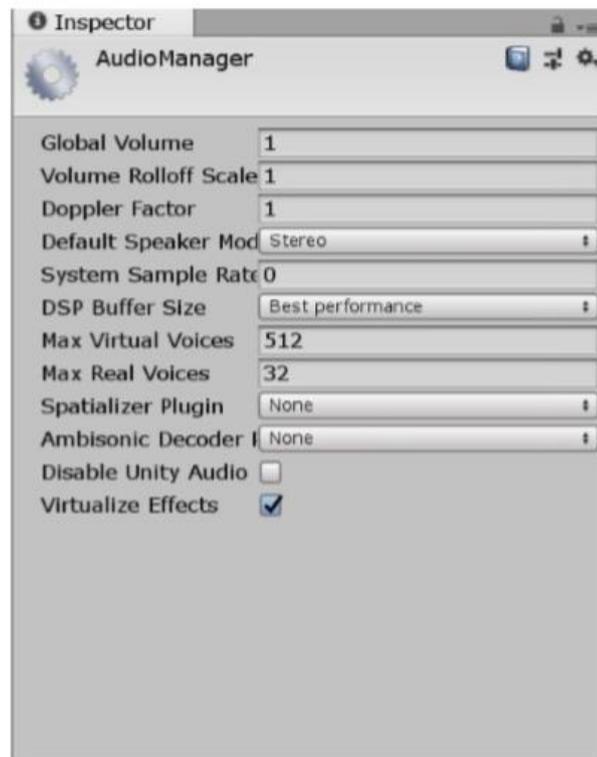
Setiap jenis audio memiliki tempatnya dalam pengembangan VR, tetapi untuk pengalaman yang paling realistis, kami ingin mensimulasikan audio menggunakan pengaturan audio spasial untuk mendapatkan efek yang sangat mirip dengan kehidupan nyata.

Ada beberapa perangkat pengembangan perangkat lunak (SDK) yang saat ini tersedia secara gratis, dan sebagian besar bekerja lintas platform menggunakan Unity. Untuk kesederhanaan, di sini kami menggunakan plug-in Audio Spasial Oculus yang saat ini ada di Unity. Pada Unity 2108.2, Anda dapat menemukan pengaturan ini pada tab Edit (Gambar 8-9), lalu klik Pengaturan Proyek → Audio → Plugin Spatializer (Gambar 8-10).

Setelah Anda mengatur Spatializer ke Oculus, Anda dapat menambahkan Sumber Audio ke Scene Anda (idealnya, di dekat kamera Anda). Saat Anda melakukan ini, opsi Spatialize menjadi aktif, seperti yang ditunjukkan pada Gambar 8-11.

Klik Spatialize pada sumber audio Anda dan kemudian impor klip audio apa pun yang Anda miliki untuk mengujinya. Untuk mencobanya, jalankan Scene Anda lalu seret sumber

audio Anda di sekitar kamera Anda. Jika Anda memiliki headphone, Anda akan mendengar audio dari telinga kiri ke telinga kanan, dan sebaliknya. Jika Anda mengaktifkan VR, Anda akan dapat mengujinya hanya dengan memutar kepala Anda! Cukup rapi, bukan?



Gambar 8-10. Pengelola audio Unity

Audio Dalam AR

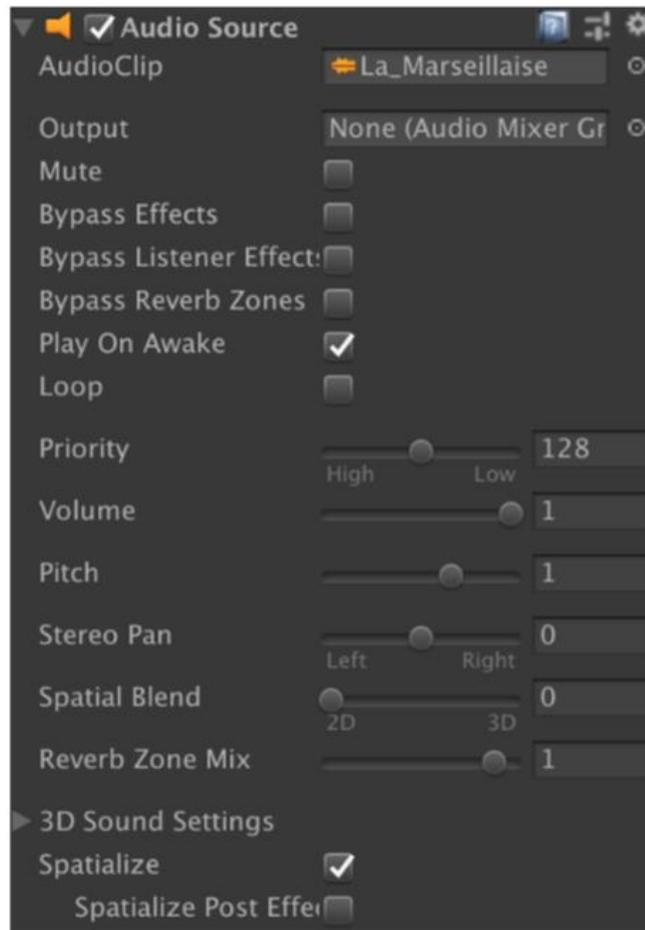
Dalam kebanyakan kasus, pengembangan untuk headset AR akan sangat mirip dengan AR seluler. Namun, dalam hal audio, Anda akan ingin mengambil pendekatan yang sangat berbeda di antara keduanya. Untuk headset AR, saya sarankan mengembangkan audio dengan gaya yang mirip dengan headset VR. Namun, untuk AR seluler, saya merekomendasikan pendekatan yang sama sekali berbeda menggunakan audio mono yang tidak memiliki spasial.

Mono versus stereo

Sebagian besar file audio yang Anda temukan online akan dalam format stereo, yang berarti audio direkam dalam dua saluran, satu untuk telinga kiri dan satu untuk kanan, yang sangat bagus untuk pengguna headphone. Namun, untuk AR seluler, yang terbaik adalah mengambil file stereo di proyek Anda dan mengonversinya menjadi mono; yaitu, hanya memiliki satu saluran. Ada tiga alasan untuk ini:

- Sebagian besar pengguna seluler tidak menggunakan headphone saat menggunakan aplikasi.
- Bahkan jika mereka menggunakan headphone, audio stereo—terutama audio spasial—dapat membingungkan pemain saat mereka memutar perangkat dan audio berubah meskipun kepala mereka tidak bergerak.

- Sebagai bonus tambahan, audio mono membutuhkan lebih sedikit ruang, yang penting pada perangkat seluler apa pun dan terutama penting saat mengembangkan menggunakan AR Studio atau Lens Studio.



Gambar 8-11. Komponen Sumber Audio default Unity

Untuk memberikan pengalaman terbaik, saya sarankan Anda menggunakan audio mono dengan sumber audio 3D untuk mengubah audio hanya saat pemutar bergerak mendekati objek. Untuk membuat klip audio mono, Anda dapat menggunakan program seperti Audacity untuk menggabungkan saluran audio kiri dan kanan menjadi satu saluran rata-rata. Ini adalah opsi yang mudah digunakan dalam Audacity. Setelah Anda mengeksport klip, Anda dapat menggunakannya dalam sumber audio Unity atau program AR apa pun yang Anda pilih.

8.9 PARADIGMA INTERAKSI UMUM

Hukum III Newton: Untuk setiap aksi, ada reaksi yang sama besar dan berlawanan arah. Saat Anda menerapkan ini ke VR dan AR, untuk setiap tindakan yang dilakukan pengguna, perlu ada respons yang sesuai dengan harapan pengguna. Entah itu melempar benda dan melihatnya terbang menjauh, menarik tuas dan membuka pintu rahasia, atau sekadar memutar kepala dan membuat dunia diperbarui. Semua tindakan ini termasuk dalam kategori

desain game yang bagus, tetapi, yang lebih penting, semua tindakan ini akan membuat pengguna merasa benar-benar tenggelam dan menghindari potensi penyakit simulator.

Selain itu, pelajaran penting lainnya dari desain game adalah jangan pernah membanjiri pengguna dengan terlalu banyak opsi input. Dan konsekuensinya adalah pengembang harus selalu memandu pengguna untuk mempelajari semua opsi input sambil tetap membuat pengguna merasa seperti mereka menemukan alat baru. Ringkasnya, ini berarti bahwa tidak peduli apakah pengguna baru saja mengunduh aplikasi Anda atau telah menghabiskan 100 jam di dalamnya, mereka harus selalu tahu apa yang harus mereka lakukan selanjutnya.

Apa yang kami buat selanjutnya adalah beberapa interaksi input umum yang terlihat di seluruh judul VR dan AR: sistem inventaris untuk VR, dan raycast layar sentuh dunia nyata untuk AR seluler.

8.10 INVENTARIS UNTUK VR

Sistem inventaris sering kali harus dimiliki saat membuat pengalaman dengan banyak objek interaktif yang perlu dipegang dan dibawa oleh pemain. Untungnya, karena kita sedang membangun dunia virtual, kita memiliki ruang "tak terbatas" untuk menempatkan objek. Jadi, mari kita ambil ruang di depan controller untuk menahan objek kita. Untuk mengatur ini, mari buat beberapa slot inventaris (diwakili oleh bola default, seperti yang ditunjukkan pada Gambar 8-12) dan letakkan di bawah controller kita (Gambar 8-13) sehingga mereka selalu mengikuti pengontrol.



Gambar 8-12. Sebuah prototipe slot inventaris



Gambar 8-13. Setup untuk inventaris yang di-parent ke controller

Dengan pengaturan itu, kita bisa fokus pada skrip kita. Secara total, kita akan membutuhkan tiga: Inventory Manager, InventorySlot, dan InventoryItem. Mari kita lihat lebih dekat satu per satu:

InventoryManager

Ini akan menjadi skrip yang menangani ketika pengguna menekan tombol Menu Aplikasi di controller kiri atau kanan untuk memungkinkan kami menyembunyikan atau menampilkan semua slot inventaris untuk controller masing-masing.

InventarisItem

Ini akan menjadi skrip yang dilampirkan ke item apa pun yang dapat ditambahkan dan dihapus dari inventaris.

InventoryItem

Ini akan menjadi skrip yang dilampirkan ke setiap slot untuk mempertahankan statusnya.

Mari kita mulai dengan InventoryManager. Untuk kelas ini, kita hanya perlu memeriksa apakah tombol Menu Aplikasi ditekan dan, jika demikian, menampilkan atau menyembunyikan slot inventaris untuk pengontrol. Untuk efek yang lebih halus, Anda juga dapat menganimasikan slot masuk dan keluar, yang ditampilkan di repositori GitHub. Setelah ini diatur, kita bisa menambahkan kelas ini sebagai komponen ke setiap controller kita:

```

public class InventoryManager : MonoBehaviour {
    public Controller controller;
    //which controller
    public GameObject inventoryParent;
    //parent of slots for this controller
    void Update() {
        //check every frame
        if (controller.PressDown(ApplicationMenu)) {
            // the first frame the button is pressed
            ShowInventorySlots(true);
        } else if (controller.PressUp(ApplicationMenu)) {
            ShowInventorySlots(false);
        } } void ShowInventorySlots(bool show) {
  
```

```

        inventoryParent.SetActive(show);
        //toggle whether shown or not based on bool parameter }
    }

```

Selanjutnya, InventoryItem. Sebelum kita menulis skrip ini, mari tambahkan semua komponen yang kita butuhkan ke objek uji; yaitu, Komponen Rigidbody dan Collider. Untuk menguji ini, kita juga perlu menyertakan SDK, seperti VRTK, untuk dapat mengambil objek, dan ini sudah akan disertakan dalam repositori GitHub.

Setelah itu diatur, saatnya menulis kode. Metode utama yang akan kita gunakan untuk memicu ketika kode mulai berjalan adalah OnTriggerEnter, metode yang dipicu dari Rigidbody yang kita lampirkan sebelumnya. Rigidbody ini memeriksa ketika ada penumbuk yang menempel pada objek kita bertabrakan dengan penumbuk lainnya. Ketika itu terjadi, objek dapat direferensikan untuk melihat apakah itu slot. Dalam contoh ini, kami memeriksa nama, tetapi cara lain yang layak untuk memeriksa adalah dengan tag atau lapisan. Jika itu adalah slot, kami kemudian dapat menetapkan item kami ke slot inventaris, yang akan kami tangani dalam skrip InventorySlot:

```

public class InventoryItem : MonoBehaviour {
    private void OnTriggerEnter(Collider other) {
        if (other.name == "Slot") {
            other.gameObject.GetComponent<InventorySlot>().SetItem(this, ItemReleased);
            //call the slot method SetSize(.01f); //set size to fit in slot
        } } void ItemReleased() {
            //callback for when item leaves slot
            SetSize(1f);
            //when item is released set size to normal }
            void SetSize(float size) {
                transform.localScale = Vector3.one * size; //set a uniform size i.e (1,1,1) }
            }
}

```

Last but not least adalah kode untuk setiap slot inventaris. Meskipun mungkin terlihat seperti banyak kode, sebagian besar sebenarnya mendapatkan dan mengatur variabel selama dua metode SetItem (dipanggil dari InventoryItem) dan OnTriggerExit, yaitu saat item ditarik keluar dari slot dari tangan. Karena kita juga menggunakan OnTriggerExit di kelas ini, kita juga perlu menyertakan Collider dan Rigidbody untuk setiap slot yang kita buat dalam Scene Unity. Dan setelah masing-masing skrip ini ditetapkan ke Objek Game yang tepat, sistem inventaris kami harus siap:

```

Public class InventorySlot : MonoBehaviour {     public MeshRenderer renderer;
//the renderer to show the slot public delegate void ItemReleasedAction();
//method signature for callback function private InventoryItem currentItem;
//stores Current Item private ItemReleasedAction currentReleasedCallback;
//callback for item void SetItem(InventoryItem item, ItemReleasedAction releasedCallback )
{ //called from inventory item
    item.transform.parent = this.transform; //set the parent

```

```

        item.transform.position = this.transform.position;
//center
        currentItem = item;
        currentReleasedCallback = releasedCallback;
        renderer.enabled = false;    }
private void OnTriggerExit(Collider other) {
        if (other.GetComponent<InventoryItem> == currentItem) {
                currentReleasedCallback();
//hand is grabbing item out of inventory
        currentItem = null;
        currentReleasedCallback = null;
        renderer.enabled = true;
        }
}
}
}

```

Dan dengan itu, inilah struktur sederhana dari sistem inventaris. Pastinya ada banyak peningkatan yang dapat kami lakukan (beberapa di antaranya akan dibuat di GitHub), tetapi ini dimaksudkan sebagai titik awal menuju inventaris yang disesuaikan dalam pengalaman Anda.

8.11 AUGMENTED REALITY RAYCASTS

Sebagai sebuah konsep, ini sangat penting untuk dikembangkan di AR. Namun, dalam praktiknya, ini hanya membutuhkan beberapa baris kode untuk diterapkan, tetapi ada juga beberapa baris yang dapat sepenuhnya mengubah perilaku augmentasi Anda jika salah. Dalam grafik komputer dan juga dalam Unity, sinar didefinisikan oleh titik awal dan arah dan kemudian digunakan sebagai bagian dari perhitungan raycast untuk menemukan penumbuk fisika pertama (jika ada) dengan mana sinar berpotongan.

Dalam konteks AR, ini bisa sedikit membingungkan karena raycast AR mirip dengan definisi ini kecuali raycast AR tidak akan mendeteksi objek virtual. Bergantung pada seberapa canggih SDK, SDK mungkin dapat mengenali dan mengembalikan objek, tetapi kemungkinan besar SDK akan mengembalikan titik di ruang virtual yang memetakan ke titik di dunia nyata. Berikut ini contohnya:

1. Seorang pengguna mengetuk layar untuk menempatkan kursi virtual di lantai:

```

// FixedUpdate is called everytime the Physics engine updates void FixedUpdate
() {
        TrackableHit hit;
//stores all raycast information if there is a hit
        // Does the ray intersect any objects excluding the player layer
        Ray ray = Camera.main.ScreenPointToRay(Input.touchPosition);
//get location of touch

```

```

        if (Frame.Raycast(ray.origin, ray.direction, //ARCore example
            out hit, Mathf.Infinity)) {
            //Raycast was Successful - now do something!
        }
    }
}

```

2. Seorang pengguna mengetuk layar untuk mendapatkan informasi lebih lanjut tentang kursi virtual:

```

// FixedUpdate is called everytime the Physics engine updates
void FixedUpdate () {
    RaycastHit hit;
    //stores all raycast information if there is a hit
    // Does the ray intersect any objects excluding the player layer
    Ray ray = Camera.main.ScreenPointToRay(Input.touchPosition);
    //get location of touch
    if (Physics.Raycast(transform.position, transform.forward,
        out hit, Mathf.Infinity)) {
        //Raycast was Successful - now do something!
    }
}

```

Seorang pengguna ingin menyeret kursi virtual melintasi ruangan.

Dan yang terakhir ini adalah di mana hal-hal mulai menjadi rumit. Dengan drag, Anda harus menggabungkan menggunakan raycast fisika built-in Unity dengan raycast AR. Secara khusus, dalam urutan eksekusi:

1. Pengguna mengetuk layar.
2. Menggunakan raycast fisika Unity, periksa untuk melihat apakah pengguna mengetuk objek untuk diseret.
3. Jika itu adalah objek yang dapat diseret, lakukan raycast AR pada setiap panggilan Pembaruan untuk menyeret objek ke posisi baru yang diinginkan pengguna.

8.12 KESIMPULAN

Ini semua adalah interaksi AR yang cukup umum yang diharapkan pengguna dalam aplikasi mereka, jadi sangat membantu untuk mengingat raycast dan mengapa mereka mungkin tidak berkinerja seperti yang diinginkan saat debugging.

Kami baru saja membahas beberapa dari beberapa tips yang penting untuk pengembangan VR. Namun sebelum kita menutup bab ini, saya ingin memberi Anda satu tip terakhir sebelum Anda memulai perjalanan pengembangan Anda untuk VR dan AR. Jika Anda pernah terjebak atau frustrasi, ingat saja PRE:

Passiom

Untuk siapa, untuk melakukan apa, dan mengapa Anda membangun proyek luar biasa Anda.

Resource

Ada banyak sumber daya online dari komunitas online di Facebook dan Slack hingga pertanyaan yang dijawab di Google. Kemungkinannya adalah seseorang juga menghadapi hal serupa.

Expeiece

Anda sedang membangun sesuatu yang unik yang akan membuat orang mengalami sesuatu yang tidak akan pernah mereka alami. Hargai itu!

Ya, saya tahu akronim ini murahan, tetapi jika itu membantu Anda menyelesaikan proyek Anda, itu membuatnya lebih berharga untuk semua orang. Dan ya, pengembangan VR dan AR itu sulit, tetapi sebagai penulis, kami semua senang melihat apa yang dapat Anda bangun dan berkontribusi untuk komunitas kami yang luar biasa!

BAGIAN V

MENINGKATKAN REPRESENTASI DATA

VISUALISASI DATA *ARTIFICIAL INTELLIGENCE* (AI) DALAM KOMPUTASI SPASIAL

Selama masa ketika perusahaan "big data" telah muncul dengan investasi besar dalam mengerjakan model kecerdasan buatan (AI) dalam produksi, sangat penting bagi insinyur perangkat lunak, perancang, dan profesional bisnis teknologi baru dan berpengalaman dalam *Virtual Reality* (VR), *augmented reality* (AR), *mixed reality* (MR), dan *eXtended Reality* (XR), atau X Reality, untuk memiliki pemahaman dasar yang kuat tentang penggunaan dan visualisasi data dunia nyata, data yang dibuat pengguna, dan data yang dibangun dalam realitas yang diwujudkan. Dalam bab-bab berikutnya, pemimpin redaksi buku antologi dan Rekan Keragaman Pembelajaran Mendalam Universitas San Francisco, Erin Pangilinan; Direktur Unity dari departemen penelitian AI, Nicolas Meuleau; dan insinyur perangkat lunak senior Unity, Arthur Juliani membahas berbagai aspek aplikasi dan pengalaman imersif melalui lensa prinsip berbasis data. Bagian ini berupaya melakukan hal berikut:

- Mendefinisikan visualisasi data, AI, pembelajaran mesin, dan paradigma dan teknik Reinforcement Learningn melalui kasus penggunaan industri praktis yang melibatkan tubuh manusia, baik yang dibuat pengguna dalam komputasi spasial, abstraksi, atau rekonstruksi tiga dimensi (3D) dari data dunia nyata yang direpresentasikan dalam komputasi spasial
- Menawarkan sumber daya dan kiat untuk mereka yang membangun dengan data dan kerangka kerja sumber terbuka serta peluang untuk eksplorasi lebih lanjut seiring kemajuan komputasi spasial dan pembelajaran mesin
- Uraikan tantangan dalam mendesain dan mengembangkan untuk komputasi spasial, termasuk menyoroti nuansa spesifik yang merupakan tantangan desain dalam AR mengingat overlaynya di dunia nyata, dan perbedaannya dari VR

Dalam Bab 9, Erin Pangilinan mendefinisikan visualisasi data dan pembelajaran mesin serta peluang desain uniknya dalam teknologi imersif. Paradigma desain baru yang tidak dirancang sebelumnya dengan desain untuk desktop dan platform seluler dimungkinkan dengan komputasi spasial. Dia menjelaskan tantangan sehubungan dengan hambatan ergonomis saat ini yang dihadapi AR dan VR, dan menawarkan sumber daya dan referensi untuk tutorial langsung untuk mulai membuat data dan visualisasi pembelajaran mesin di XR. Meskipun bab ini tidak dapat sepenuhnya membahas dan membahas startup visualisasi data dengan dasbor 2D yang menganalisis data pengguna (seperti EaseVR, CognitiveVR, Retinad, dll. melakukan analisis yang lebih terlibat daripada peta panas sederhana dalam aplikasi Unity of XR), bab ini merujuk pada aspek lain dari perwujudan kenyataan dengan contoh kasus penggunaan di berbagai vertikal industri B2B yang memvisualisasikan data dari tubuh manusia (khususnya untuk teknologi kesehatan—biotek tidak tercakup dalam diskusi Dilan Shah di Bab 11),

beberapa di antaranya dianggap "data besar" dan ditampilkan secara real time dalam skala besar .

Bab 10, oleh staf Unity Nicolas Meuleau dan Arthur Juliani, menjelaskan paradigma AI yang ada, termasuk AI reaktif, AI deliberatif, dan Reinforcement Learning; bagaimana mereka ditantang oleh XR; dan jawaban yang bisa mereka bawa. Aplikasi melibatkan perilaku seperti animasi, aktivitas karakter nonpemain (NPC), dan mendongeng —perilaku dunia. Meuleau berfokus pada perencanaan perilaku dan animasi otomatis sebagai bagian dari penawaran Unity. Anda mempelajari pendekatan berbasis pembelajaran mesin, khususnya Reinforcement Learning, dan metode Imitation Learning yang melibatkan data pemain game di XR. Menjelang akhir bab ini, Anda belajar tentang merancang rangkaian demonstrasi perilaku, baik yang disediakan oleh manusia atau dihasilkan dengan cara lain (pemain yang bermain game dan, bahkan yang lebih signifikan dalam lingkungan AR dan VR dengan gerakan tubuh manusia, gestur, kami dapat menggunakan data untuk menginformasikan perilaku agen otonom dalam simulasi).

AI dapat digunakan untuk menghasilkan konten yang menambah seniman teknis 3D dalam alur pengembangan game seperti yang terlihat pada pembicaraan Nvidia di The Game Developer Conference (GDC) 2017 “Zoom, Enhance, Synthesize! Peningkatan Ajaib dan Sintesis Materi menggunakan Pembelajaran Mendalam”, Algoritme pembelajaran mendalam seperti transfer gaya digunakan dalam pembuatan film 360 derajat oleh Facebook, seperti yang diputar selama Festival Film Tribeca 2017. Meskipun ini tidak sepenuhnya tercakup mengingat kapasitas terbatas untuk kami.

Dalam lingkup ini, para insinyur dan perancang perangkat lunak terus menunjukkan cara meningkatkan aplikasi dan pengalaman XR mereka melalui algoritme AI mutakhir dan cara-cara baru untuk merepresentasikan dan memvisualisasikan data (data pengguna atau dunia nyata) dalam media baru. AI di luar bagian ini dapat dijelaskan lebih rinci dalam bab tentang SLAM dan AR cloud yang ditulis oleh salah satu pendiri 6D.ai, Matt Miesnieks dan Profesor Victor Presacariu serta karya Dieter Schmalsteig yang menjelaskan secara rinci tumpang tindih antara XR dan visualisasi berkaitan dengan jalur pipa data di VR dan AR yang terkait dengan model spasial, deteksi objek, tracking 3D, dan rendering.

BAB 9

DESAIN DAN PENGEMBANGAN VISUALISASI DATA DAN MACHINE

9.1 PENGANTAR

Visualisasi data dan pembelajaran mesin mengubah masa depan tempat kerja. Prinsip-prinsip desain framing sangat berharga. Perusahaan seperti perusahaan yang didukung Dana VR, Virtualitics, yang didirikan oleh Caltech PhD, mengumpulkan dana dalam jumlah besar berdasarkan prinsip desain yang baik. Ada juga banyak konsultan independen visualisasi data lainnya yang bermunculan di semua platform komputasi spasial. Banyak di antaranya berada dalam lingkup layanan bisnis dengan kumpulan data yang sangat besar, di vertikal B2B fintech (finance tech), teknologi kesehatan, dan biotek.

Kami memulai bab ini dengan membahas relevansi topik bagi pengguna yang mengalami aplikasi visualisasi data dan pembelajaran mesin. Kami kemudian menawarkan kerangka kerja untuk dipertimbangkan untuk mengidentifikasi tujuan berguna yang membuat topik unik untuk komputasi spasial dibandingkan platform lainnya. Kami menguraikan tujuan kami untuk memahami, mendefinisikan, dan menetapkan prinsip desain dan pengembangan visualisasi data dan mesin dalam realitas yang diwujudkan. Kemudian, kami membahas berbagai tantangan dengan visualisasi data dan pembelajaran mesin di XR, menjelaskan berbagai contoh kasus penggunaan industri untuk visualisasi data dan pembelajaran mesin yang dibangun di atas data dan kerangka kerja sumber terbuka (meskipun beberapa pekerjaan menarik juga telah dilakukan dengan kerangka kerja sumber terbuka dan data kepemilikan). Menjelang akhir bab ini, kami menyoroti referensi ke tutorial untuk pembuat visualisasi data dan pembelajaran mesin, apakah Anda baru atau insinyur perangkat lunak atau desainer berpengalaman yang sudah terbiasa bekerja pada platform web (Anda dapat dengan mudah menggunakan A-Frame di JavaScript atau kerangka kerja lain) atau dalam pengembangan asli, C# di Unity. Ada beberapa contoh gambar yang dibuat menggunakan C++ dan Unreal Engine.

9.2 MEMAHAMI VISUALISASI DATA

Meskipun whitepaper seperti “Analisis Biaya-manfaat dari Visualisasi di Virtual Environment (VEs)” IEEE mempertanyakan relevansi dan tujuan visualisasi dalam XR, menanyakan “apakah kita benar-benar membutuhkan visualisasi 3D untuk data 3D?” Secara sederhana, dasar bab ini mengasumsikan dari awal bahwa penggunaan VE memungkinkan pemahaman yang lebih baik tentang data 3D, dengan konteks yang sesuai, desain yang matang, dan pengembangan.

Pertama, kami menjelaskan dan mendefinisikan visualisasi data dalam XR dan apa yang membuatnya unik untuk media lain sebelumnya. Kami melihat perbedaan antara visualisasi data besar interaktif versus representasi infografis murni.

Dianggap sebagai bapak baptis visualisasi data, ahli statistik Edward Tufte menulis bahwa selama berabad-abad pelukis, animator, dan arsitek telah berusaha untuk

merepresentasikan data (2D dan 3D) pada berbagai tampilan (terutama dalam ruang 2D), menggunakan perspektif dan gerakan. Perhatikan bahwa banyak infografis statis kurang gerak atau pemahaman umum tentang perspektif dan dengan demikian tidak memenuhi syarat sebagai "visualisasi data yang baik" dalam komputasi spasial; pengalaman sebagian besar pasif dan tidak memungkinkan rotasi atau prinsip lain yang dibahas nanti dalam bab ini. Visualisasi data dan machine learning memungkinkan pengguna untuk melihat, menjelajahi, dan memahami data dengan lebih baik. Seperti yang dikatakan oleh Fernanda Viegas dan Matt Wattenberg (Orang Google dan Peneliti AI yang berfokus pada visualisasi data) di NeurallIPS 2018, visualisasi data mengubah data menjadi pengkodean visual yang membantu pengguna mendidik, berkomunikasi, memberikan insight, dan menjelajahi data dengan lebih baik. Tanpa visualisasi, data hanyalah angka mati pada sebuah halaman.

Dalam buku maninya, *Tampilan Visual Informasi Kuantitatif* (Graphics Press, 2001), Tufte menulis bahwa visualisasi data membuat pemahaman manusia tentang kumpulan data besar lebih koheren. Ini melayani tujuan yang jelas untuk menggambarkan data yang direpresentasikan dalam berbagai bentuk; misalnya, sebagai abstraksi (bagan pai, diagram batang, dll.) dan sering sebagai istilah untuk menggambarkan rekonstruksi data 3D sebagai objek dalam ruang 3D (misalnya, struktur anatomi yang direkonstruksi 3D seperti data otak, dan irisan datar pencitraan resonansi magnetik [MRI] file dalam lingkungan augmented dan virtual). Data itu sendiri bersifat komparatif, relasional, multivariat, dan dapat memungkinkan pengguna untuk menanyakan pertanyaan spesifik atau menjelajahi data secara umum untuk mendapatkan pemahaman yang lebih baik tentang kualitasnya. Berikut adalah beberapa karakteristik utama dari visualisasi data interaktif di XR:

- Ini dapat memplot dan mengurutkan data relasional melalui pengintegrasian deskripsi untuk membedakan data kategoris, apakah itu data kualitatif dan dapat melibatkan beberapa sifat statistik (fokus pada data kuantitatif).
- Ini melibatkan arsitektur informasi yang menunjukkannya sebagai dinamis dan memberikan interaktivitas kepada pengguna.
- Ini menekankan estetika untuk membantu pengguna memahami data melalui desain yang baik, tidak hanya untuk tujuan dekorasi.

Data jauh lebih sulit dipahami tanpa visualisasi. Seperti yang dikatakan oleh peneliti pembelajaran mendalam, "visualisasi data dan analitik visual dapat digunakan untuk unggul dalam komunikasi pengetahuan dan penemuan insight dengan menggunakan pengkodean untuk mengubah data abstrak menjadi representasi yang bermakna."

Interaktivitas dan animasi di semua platform, desktop, seluler, dan komputasi spasial membantu pengguna membuat data lebih mudah diakses dan lunak dengan manipulasi langsung dengan berbagai set input dan kontrol.

Prinsip untuk Visualisasi Data dan Pembelajaran Mesin dalam Komputasi Spasial

Merujuk kerangka kerja oleh peneliti pembelajaran mendalam, 6 pembuat visualisasi data dan pembelajaran mesin dalam komputasi spasial harus mengeksplorasi lima W yang

akan membantu memberi mereka landasan untuk menciptakan pengalaman aplikasi yang sukses dalam komputasi spasial.

Pembuat konten harus mempertimbangkan desain pengalaman pengguna mereka dengan memulai dengan hal berikut: mengidentifikasi pengguna target mereka (Siapa) dan di mana tepat untuk menggunakan visualisasi data (Kapan), jenis visualisasi data yang dibuat (Apa), membenarkan keberadaannya sebagai optimal dalam komputasi spasial dan Mengapa sebelum mereka mengidentifikasi metode atau (jenis visualisasi apa) yang melibatkan atau tidak melibatkan pembelajaran mesin sebelum mereka mulai memilih Tempat menyimpan, memproses, memvisualisasikan data ini sebelum memilih Bagaimana (bahasa mana yang digunakan untuk platform mana) .

Kami mengeksplorasi lebih lanjut tentang pendekatan pada metode dan cara membuat visualisasi aktual di akhir bab ini, di mana kami mempertimbangkan rekayasa data-ke-visualisasi holistik dan proses desain pipa. Berikut adalah contoh dari prinsip-prinsip dalam praktek. Lebih khusus lagi, pencipta harus mempertimbangkan faktor-faktor ini untuk disengaja tentang proses pembuatan visualisasi mereka:

Identifikasi tujuannya, tanyakan pada diri Anda mengapa visualisasi data atau pembelajaran mesin ini masuk akal dalam komputasi spasial versus komputasi lainnya. Pencipta harus mempertimbangkan interaksi sehingga pengguna dapat secara langsung memanipulasi dan membuka insight lain untuk mendapatkan pengalaman visualisasi data yang efektif yang tidak mungkin dilakukan di media lain.

Mengapa

Tentukan target pengguna akhir dari data atau visualisasi pembelajaran mesin pengalaman/aplikasi komputasi spasial dan manfaat apa yang akan mereka peroleh dari pengalaman mereka dalam komputasi spasial (misalnya, ahli bedah yang memantau data otak dan informasi struktur anatomi lainnya). Kami akan membahas hal ini secara rinci saat kami menjelaskan kategori data dan bagaimana jenis interaksi yang bergantung pada platform pilihan telah berkembang dari waktu ke waktu untuk berbagai visualisasi pada Gambar 9-4.

Apa

Pilih cakupan dan ukuran tipe data, seberapa besar, dan seberapa banyak yang ingin mereka visualisasikan. Untuk data MRI spesifik pasien dengan tumor kanker). Tidak semua data otak sama; misalnya, untuk kumpulan data yang lebih besar yang melibatkan pencitraan otak, dalam ruang pemetaan otak dan konektivitas, para peneliti di Spanyol memilih untuk memvisualisasikan subset data multidimensi untuk visualisasi komputasi spasial menggunakan Unity.

Di mana

Pilih platform komputasi spasial yang paling tepat untuk menargetkan Head Mounted Display (HMD) atau tampilan seluler. Pertimbangkan berbagai alat pembuatan prototipe dalam 2D (lihat di bagian sumber daya di akhir bab ini) pada platform komputasi non-spasial (desktop dan seluler) jika memungkinkan. Pembuat harus memahami kompleksitas data sehingga mereka tahu apakah itu harus diproses sebelumnya dan di mana disimpan dan disimpan (di cloud menggunakan Amazon Web Services, dalam format JSON). Ini mungkin atau mungkin

tidak melibatkan beberapa pembuatan prototipe (terkadang ini melibatkan penggunaan 3D dengan alat 2D), sebelum memuat dan memvisualisasikan data sepenuhnya dalam XR.

Bagaimana

Pilih metode yang akan digunakan saat Anda membuat visualisasi data atau machine learning. Visualisasi dasar tidak memerlukan banyak pra-pemrosesan, tetapi bagi mereka yang melakukannya (seringkali yang menggunakan Python) seluruh jalur pipa harus dipertimbangkan. Pilih bahasa pemrograman lain yang akan Anda gunakan untuk platform yang dipilih untuk memvisualisasikan data (mis. C#, C++, JavaScript) dan program Lingkungan Pengembangan Terpadu (IDE) mana yang akan digunakan untuk (Unity, Unreal Engine, mesin game lain, mesin milik Anda sendiri untuk membuat, buku ini menampilkan contoh-contoh terutama dari IDE pertama).

Beberapa visualisasi data dibuat untuk penggunaan praktis yang membantu profesional pemasaran, analis bisnis, dan eksekutif dengan mampu menampilkan dan berinteraksi dengan data, mengarahkan mereka ke keputusan bisnis yang lebih baik. Lainnya mungkin insinyur pembelajaran mesin, ilmuwan data, atau insinyur perangkat lunak yang berusaha menemukan teknik optimasi, mengeksplorasi interpretasi model dan dapat membuat penemuan ini melalui eksplorasi visualisasi komputasi spasial. Mereka dapat melihat lebih baik lapisan yang mendasari data multidimensi yang lebih kompleks dalam komputasi spasial daripada media lain. Di sini visualisasi berfungsi sebagai solusi untuk "kutukan dimensi", yang berarti memadatkan data multidimensi ke dalam format yang lebih dapat dipahami.

Di ujung lain spektrum, beberapa visualisasi sering salah dikategorikan sebagai visualisasi data dan sebenarnya termasuk dalam spektrum presentasi dan desain infografis dan dipandang sebagai karya eksperimental artistik yang "indah"; mereka dianggap diciptakan untuk tujuan dekoratif murni, atau lebih untuk nilai estetika dan apresiasi daripada untuk penggunaan praktis. Kami masuk ke detail tentang ini saat kami menggambarkan kategori data pada Gambar 9-4 nanti dalam bab ini.

Mengapa Visualisasi Data dan Pembelajaran Mesin Bekerja di Komputasi Spasial

Kami mempelajari topik ini lebih dalam saat kami menjelaskan evolusi desain visualisasi data, bagaimana tujuannya telah meningkat dan berkembang dengan diperkenalkannya komputasi spasial sebagai media tujuannya, berbagai kategorisasi data dan desain interaksi yang efektif.'

9.3 EVOLUSI DESAIN VISUALISASI DATA DENGAN MUNCULNYA XR

Tufte melanjutkan dengan mengatakan dalam bukunya selanjutnya *Beautiful Evidence* bahwa visualisasi data menyediakan produsen dan konsumen ciptaan untuk menampilkan bukti. Dia lebih lanjut menjelaskan bahwa dasar desain visualisasi berasal dari prinsip-prinsip dasar yang mendasari desain analitis, yang agnostik untuk "bahasa atau budaya atau abad atau teknologi tampilan informasi." Tufte menjelaskan:

Powerpoint seperti terjebak dalam gaya kartun tanah datar Mesir awal daripada menggunakan alat representasi visual Renaisans yang lebih efektif. Prinsip-prinsip

pemikiran analitis—dan bukan dari kebiasaan setempat, mode intelektual, kenyamanan konsumen, pemasaran, atau apa yang disediakan oleh teknologi tampilan.

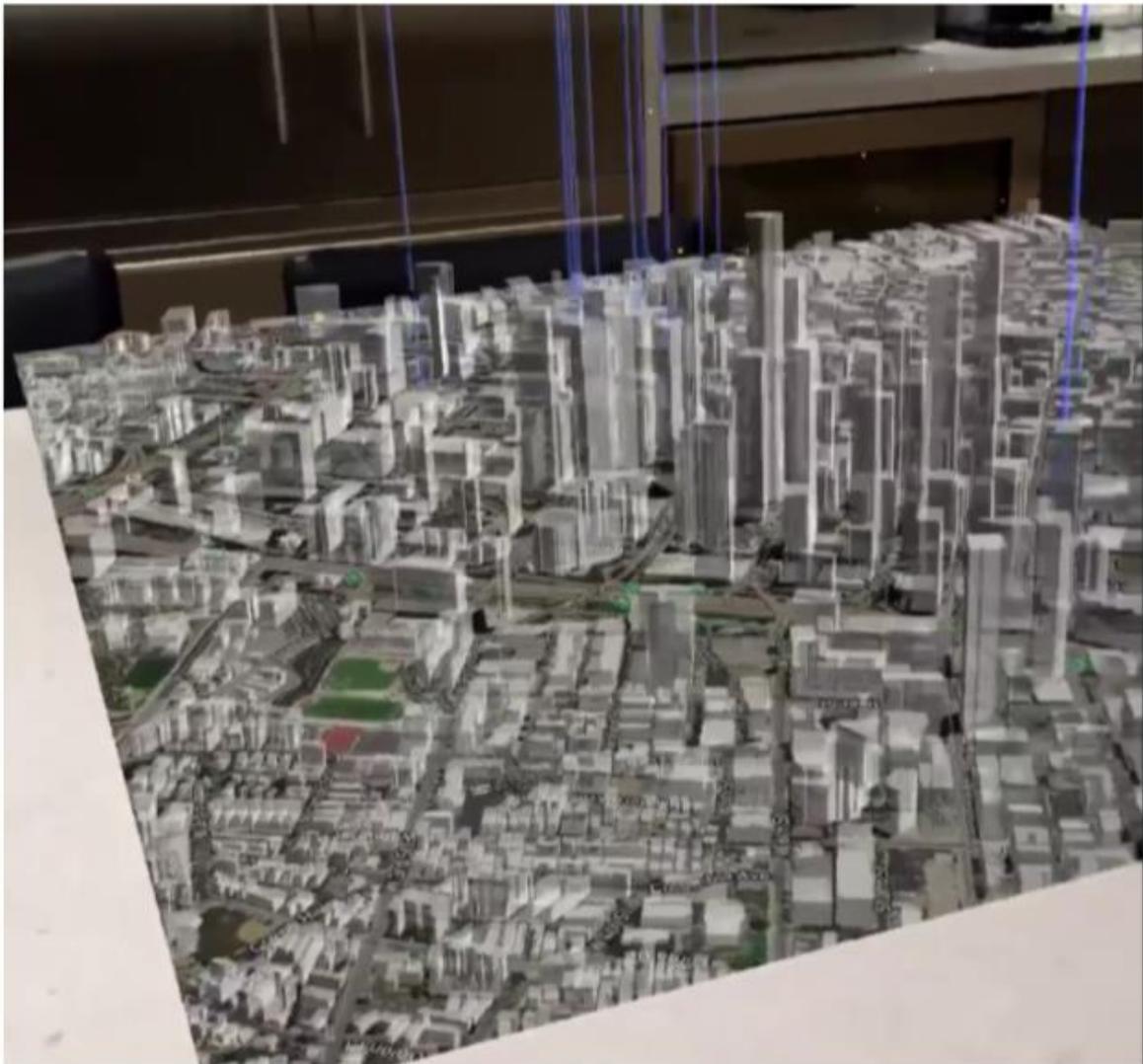
Meskipun ini benar, beberapa praktik terbaik desain visualisasi data (yang terutama dirancang untuk media kertas, desktop, dan seluler) dapat dilihat sebagai usang dan tidak semuanya langsung diterapkan dalam media spektrum XR karena memperhitungkan desain terutama hanya untuk ruang 2D dengan User interface datar (UI), bahkan dengan data 3D, atau satu jendela atau layar. Ini membatasi pengguna dan tidak memungkinkan pengguna dan produsen untuk sepenuhnya memahami penemuan data yang dapat dibuka dengan potensi teknologi yang muncul. Teknologi ini dapat meningkatkan pemikiran analitis, mengingat munculnya pencarian komputasi dan kecerdasan buatan (AI) yang menampilkan data multidimensi yang dapat lebih mudah dieksplorasi dengan teknologi baru.

Tufte, seperti banyak ilmuwan data dan akademisi lainnya telah mengkritik visualisasi data 3D yang "buruk" seperti diagram lingkaran 3D dan sebagai gantinya menawarkan pendekatan yang lebih sederhana untuk visualisasi data, yang menyatakan bahwa data geospasial 3D dengan peta sederhana pada kertas 2D sudah cukup. Mereka mengabaikan penggunaan visualisasi 3D sama sekali, tetapi ini salah arah dan mundur. Dengan diperkenalkannya AR dan VR, UI peta geospasial 3D telah berkembang pesat sejak Tufte membuat peta kertas 3D. Konsepsi data baru sekarang dikodekan ke dalam pengalaman aplikasi aktual yang meningkatkan interaksi pengguna dengan data mereka. Misalnya, di WebVR, peta cenderung lebih mirip permainan di mana pengguna dapat bergerak di bidang-z, seperti yang digambarkan pada Gambar 9-1. Visualisasi data dapat "di-gamified" ke dalam platform VR seluler (lihat Gambar 9-2) di ARKit dan MapBox.



Gambar 9-1. Plot berapa kali orang menunjukkan bahwa mereka tidak menyukai pembawa acara televisi di wilayah tertentu, dibuat dalam kerangka webXR, A-Frame, yang dapat dilihat di headset VR1

Ini menggunakan versi lama A-Frame (0.2.0). Beberapa kode mungkin tidak mencapai hasil yang diinginkan. Jika memungkinkan, tingkatkan ke versi A-Frame terbaru (yang pada saat tulisan ini dibuat adalah 0.9.0). Semua browser utama akan mulai mendukung spesifikasi WebXR pada tahun 2019. API webVR lama tidak akan digunakan lagi



Gambar 9-2. Pemetaan visualisasi data check-in FourSquare menggunakan ARKit dan MapBox oleh teknolog Aaron Ng

Lebih lanjut, input AR dan VR memungkinkan paradigma interaksi baru yang sebelumnya tidak mungkin terjadi di ruang 2D, yang mendefinisikan ulang interaksi manusia-komputer (HCI) sebagai User interface melalui pengontrol, mulai memuat secara langsung, dan langsung memanipulasi data dengan suara yang lebih canggih (via natural pemrosesan bahasa [NLP] evolusi) dan kontrol sentuh (haptics karena mereka juga terus berkembang).

Banyak yang telah ditulis tentang visualisasi data yang terbatas pada bidang 2D atau data 3D yang telah terperangkap dalam media 2D (yang sangat membatasi untuk menggali lebih dalam insight bagi mereka yang bekerja di bidang biotek dan teknologi kesehatan, dengan data mulai dari manusia anatomi dalam mikroskop pencitraan medis, visualisasi molekul DNA, dan visualisasi protein) dalam data abstrak 2D dalam ruang 3D.

Dalam bukunya, *Fundamentals of Data Visualization* (O'Reilly Media, 2018), profesor dan ahli biologi terlatih dari University of Texas, Claus O. Wilke, menekankan posisi dalam XR sebagai bagian dari diskusinya tentang visualisasi data. Dia berbicara tentang pentingnya

berbagai elemen (warna dan garis) dan sebagainya. Fokus untuk bab ini adalah pada posisi, mengingat kemampuan XR untuk menempatkan data sebagai objek pada rotasi sumbu z.

9.4 DATA 2D DAN 3D DIREPRESENTASIKAN DALAM XR

Ada berbagai jenis data yang ditampilkan dalam visualisasi data dalam platform komputasi desktop, seluler, dan spasial. Data kategorikal yang ditunjukkan pada Gambar 9-4 berkisar dari statis hingga dinamis pada berbagai platform. Jenis data yang sering direpresentasikan dalam XR adalah sebagai berikut:

- Abstraksi data 2D terlihat dalam 3D dalam XR (sering dilihat sebagai diagram batang dan diagram garis)
- Data 3D dari data 2D (struktur anatomi seperti pencitraan fMRI otak yang direkonstruksi beberapa kali agar terlihat 3D dan masuk ke ruang 2D)
- Data 3D direpresentasikan dalam ruang 3D, dalam XR (visualisasi molekul DNA terlihat dalam XR)

Setelah Anda memilih tipe data yang sedang Anda kerjakan, Anda dapat memvisualisasikan data tersebut. Beberapa data kurang masuk akal untuk divisualisasikan daripada data lainnya. Misalnya, Tufte mereferensikan diagram lingkaran 3D sebagai mundur. Wilke melanjutkan ini dan secara khusus menunjukkan visualisasi data yang efektif dalam XR yang semuanya tentang konteksnya. Wilke mengatakan, "... masuk akal untuk menggunakan visualisasi 3D saat kita ingin menampilkan objek 3D aktual dan/atau data yang dipetakan ke dalamnya."

Visualisasi Data 2D versus Visualisasi Data 3D dalam Komputasi Spasial

Meskipun Tufte telah dikutip secara luas terhadap penggunaan visualisasi data 3D yang salah (yaitu data abstrak di luar konteks seperti diagram lingkaran, yang dia yakini tidak menambah perbedaan substansial daripada visualisasi 2D) dan animasi mewah, sarjana lain seperti Wilke menunjukkan bahwa ada beberapa nilai dalam visualisasi data 3D dan komputasi spasial karena mereka memahami kemampuannya untuk melibatkan pengguna dengan data mereka dengan cara yang tidak dibatasi dalam layar 2D. Visualisasi data 3D dengan sendirinya, tanpa pemikiran yang tepat untuk jenis konten dan bagaimana itu diwakili, tidak cukup.

Interaksi demi interaksi akan membuat semacam karya seni, tetapi tidak serta merta membuat pengalaman visualisasi data dan tidak harus melibatkan data. Harus ada pertimbangan yang cermat dalam pemilihan desain di XR. Namun, rekomendasi untuk menghindarinya sama sekali tidak jelas. Jika kami menghindari visualisasi data 3D dalam ruang 2D dan 3D, kami akan merekomendasikan pembuatan visualisasi hebat seperti distill.pub, yang menunjukkan tujuan yang jelas dan meningkatkan pemahaman data oleh tim Google Brain. Kami juga akan mengabaikan semakin banyak contoh visualisasi (beberapa di antaranya ditampilkan dalam bab ini) dan lebih banyak lagi di luar bab ini yang terlalu banyak untuk disebutkan. Salah satu visualisasi pembelajaran mesin Google Brain yang lebih populer yang memungkinkan pengguna untuk lebih memahami kompleksitas mereka melalui "pengurangan dimensi" disebut t-SNE, dan Principal Component Analysis (PCA) dipopulerkan

Ini menunjukkan bagaimana insinyur dapat memperdalam pemahaman mereka tentang data oleh insinyur pembelajaran mesin terapan dan ilmuwan data. Meniadakan seluruh dimensi dan medium karena visualisasi yang dirancang dengan buruk dan menghindarinya sama sekali menunjukkan kurangnya tantangan dalam pemikiran. Jenis pemikiran ini membatasi pikiran kita untuk terbatas pada perkakas terbelakang yang menjauhkan manusia dari teknologi dan lebih jauh dari data yang dapat membantu umat manusia. Sebaliknya, kita perlu memanfaatkan teknologi untuk menjadi lebih dekat dengan data kita dan memecahkan masalah umat manusia. Oleh karena itu, visualisasi data 3D dalam komputasi spasial harus didorong tetapi dengan prinsip-prinsip desain yang tepat tetap diingat. Seperti yang dikatakan Tufte, "Kita harus menjaga data melalui integritas." Hal ini dimungkinkan dalam media baru; kita hanya perlu mendorong batas imajinasi pikiran kita dan menjadikannya relevan dengan memperbarui desain, kegunaan, dan standar untuk menjaga integritas data.

Interaktivitas dalam Visualisasi Data dalam Komputasi Spasial

Karakteristik lain yang menentukan dari visualisasi data adalah penekanannya pada estetika. Seperti yang dikatakan Tufte, "Keunggulan grafis dihargai," visualisasi data bukan tentang seni untuk seni, yang umum terjadi pada banyak desain infografis yang tidak memiliki interaksi dinamis; ini adalah sekunder. Setiap pemikiran artistik seperti teori warna harus diterapkan untuk membantu membuat data lebih mudah dipahami (yaitu, pembicaraan Viegas yang mengingatkan kita untuk mempertimbangkan buta warna dan pilihan warna untuk mengkodifikasi berbagai kunci, peta, dan kategori).¹⁵ Kemampuan dalam komputasi spasial memungkinkan pengguna dalam komputasi spasial kebebasan untuk berbuat lebih banyak dalam lingkungan 3D, tidak seperti pengalaman desktop dan seluler 2D. Bahkan dengan data 3D di desktop dan seluler, interaksi mikro terbatas pada satu tampilan dan layar. Tufte menulis bahwa visualisasi data harus "membujuk pemirsa untuk berpikir tentang substansi daripada tentang metodologi desain grafis dan teknologi produksi grafis atau sesuatu yang lain." Pada saat Tufte menulis ini, komputasi spasial tidak diaktifkan sebagai teknologi yang digunakan untuk visualisasi data seperti sekarang dan belum tentu sesuatu yang akan dia perhitungkan.

Dalam ruang 3D, desainer dan insinyur perangkat lunak harus memikirkan substansi dari apa yang mereka ciptakan dan juga cara membuatnya dengan media baru. Dengan memahami bagaimana visualisasi data terbentuk, kita dapat lebih memahami persepsi manusia, mengapa desain bekerja, dan membuat produk visualisasi data yang lebih efektif dalam media yang sama sekali baru. Pemirsa juga dapat menghargai bagaimana mereka dapat mengontrol data mereka secara lebih intuitif dan bagaimana hal itu memperdalam pemahaman mereka tentang substansi karena metodologi estetika desain (desain grafis) dan teknologi produksi grafis (cara pembuatannya dalam komputasi spasial) . Interaksi baru ini, yang hanya mungkin dalam komputasi spasial, membuka insight baru karena mampu melihat dan memanipulasi data dalam ruang 3D tidak seperti paradigma desain sebelumnya. Pembuatan konten 3D, misalnya, pada layar 2D sering kali setengah mundur, mengingat jeda waktu yang terlibat dalam memutar data 3D ke proyeksi layar yang di dunia nyata terkadang lebih baik untuk loop umpan balik yang lebih cepat melalui pembuatan objek 3D yang

sebenarnya (mis. , media pahat dan pembuatan prototipe kertas), serta visualisasi data dan mesin dalam komputasi spasial tidak terkecuali.

Komputasi spasial memungkinkan lebih banyak mekanisme untuk secara langsung memanipulasi data dan menawarkan pencipta komputasi spasial kemampuan untuk mempelajari paradigma desain baru di HCI untuk lebih memungkinkan pemahaman yang lebih dalam tentang kognisi manusia serta meningkatkan kemampuan untuk memahami dan mengungkapkan insight data baru untuk pemirsa atau pengguna.

Banyak peneliti dalam ilmu saraf, teknologi kesehatan, dan biotek mengungkapkan bahwa mereka dapat meningkatkan pemahaman mereka tentang otak manusia dan penyakit neurodegeneratif melalui kemampuan mereka untuk merekonstruksi, memanipulasi, dan berinteraksi dengan data 3D. Data ini, ketika dikunci ke layar 2D, tidak memiliki sarana apa pun yang memungkinkan manipulasi lebih langsung. Ini masih merupakan metode interaksi yang lebih disukai daripada memiliki abstraksi lain melalui keyboard, mouse, atau alat lain sebagai penghalang di antaranya. Komputasi spasial membantu mempercepat produktivitas pengguna dan mengungkap insight baru yang tidak akan mungkin terjadi jika mereka berinteraksi dengan data hanya secara 2D dengan desktop dan seluler.

9.5 ANIMASI

Animasi tidak selalu penting untuk pengalaman yang interaktif atas input dari pengguna. Namun, kurangnya responsivitas terhadap input dari pengguna membatasi batas antara visualisasi data dan infografis. Animasi saja, dengan apa yang disebut Tufte "dequantification,"¹³ dan menghapus data dan data kuantitatif akan kurang sejalan dengan visualisasi data dan infografis dan mungkin lebih merupakan karya artistik yang sering digunakan di dunia eye candy dan sering dikelompokkan dengan visualisasi data, yang belum tentu merupakan deskripsi yang akurat atau sesuai. Rekonstruksi 3D belaka tidak dapat dengan sendirinya diklasifikasikan sebagai visualisasi data; itu akan terlalu sederhana dan lebih sesuai dengan kategori di bawah infografis. Visualisasi yang tidak memiliki label data deskriptif bukanlah visualisasi data. Visualisasi data tanpa desain atau animasi UX responsif yang signifikan tetap statis dan tidak menggunakan media komputasi spasial dan mungkin juga dianggap sebagai karya seni daripada visualisasi. Bagan garis data stok dalam komputasi spasial sering dikritik seperti diagram lingkaran 3D yang dibenci Tufte. Jenis visualisasi ini dilihat sebagai perbaikan kecil dalam komputasi spasial karena dianggap menambahkan dekorasi yang tidak perlu dan asing dalam komputasi spasial.

Bagaimanapun, diagram garis dirancang, ditampilkan, dan divisualisasikan, yang harus dievaluasi sebagai ukuran keberhasilan visualisasi, bukan media untuk mengabaikan seluruhnya atau jenis visualisasi itu sendiri yang harus dipertimbangkan sendiri. Kami juga harus mengevaluasi bagaimana pengguna berinteraksi dengan pengalaman agar berguna untuk lebih memahami insight data.

Penolakan penggunaan visualisasi data dan mesin dalam komputasi spasial kembali terjadi karena praktik terbaik untuk desain dan rekayasa perangkat lunak visualisasi data masih kurang. Kita harus fokus pada manfaat komputasi spasial dan kemampuannya untuk

melihat pengguna dari semua sudut dan memutar data mereka secara lebih alami seperti yang mereka lakukan dengan objek 3D di dunia nyata. Visualisasi data yang sukses dapat dicapai ketika kami memberikan pemikiran yang baik untuk kedua fungsi tujuan yang menguntungkan pengguna akhir, yang alur kerjanya ditambah dalam ruang 3D yang bertentangan dengan layar 2D dengan interaksi manusia yang masuk akal untuk pengguna akhir.

Virtualitics dan sistem 10K (Gambar 9-3) masing-masing memiliki peta yang dirancang dengan baik yang merupakan demonstrasi visualisasi data XR yang baik dalam konteks, seperti 3D lainnya yang tidak dalam komputasi spasial (deck.GL, dll.).



Gambar 9-3. Sistem 10K, misalnya, memiliki peta hebat yang memvisualisasikan data matahari yang menunjukkan bagaimana visualisasi data VR bekerja secara efektif dalam konteks dan lebih dari sekadar konsep abstrak

9.6 KEGAGALAN DALAM DESAIN VISUALISASI DATA

Beberapa insinyur dan perancang perangkat lunak baru telah menciptakan visualisasi data yang buruk dalam komputasi spasial yang telah menolak pengguna dari media ini sama sekali karena kurangnya perhatian untuk memahami dan menerapkan prinsip-prinsip sederhana dan mendasar untuk mendesain dalam media ini. Seperti yang dicatat Tufte, "Ada cara yang benar dan cara yang salah untuk menampilkan data, ada tampilan yang mengungkapkan kebenaran dan tampilan, yang tidak." Tufte, Edward (Visual and Statistical Thinking, 45) Berikut ini adalah beberapa ciri umum visualisasi data yang buruk di XR:

- Mereka tidak memuat data dengan cara yang koheren dan mudah dimengerti.
- Mereka terutama menggunakan UI datar yang dapat ditampilkan dalam ruang 2D dan tidak dalam XR dan tidak menawarkan interaktivitas.

- Mereka mengandung terlalu banyak noise, membuat data tidak jelas untuk dipahami—yang merupakan tujuan berlawanan dari apa yang dimaksudkan untuk diaktifkan oleh visualisasi data.

Banyak visualisasi data di XR gagal karena tidak menggunakan ruang dan media 3D yang sebenarnya. Ruang ini memberi pengguna kemampuan yang tidak tersedia dengan UI 2D (yaitu, terbatas pada satu layar atau lembaran kertas dalam kehidupan nyata). Visualisasi data yang buruk dalam komputasi spasial gagal menunjukkan tujuan yang jelas mengapa data ini lebih baik direpresentasikan dalam komputasi spasial dan tidak memungkinkan pengguna untuk berinteraksi dengan data mereka secara efisien.

Di satu sisi spektrum, beberapa visualisasi data tetap statis dalam estetikanya, sedangkan yang lain memiliki terlalu banyak informasi yang mengacaukan ruang 3D sehingga tidak memiliki fokus yang jelas bagi pengguna untuk mengontrol data mereka dengan mudah dengan interaksi controller sentuh sederhana (suara kanan sekarang terlalu banyak tahap awal, yang banyak diharapkan di masa depan akan berubah saat NLP dan AI maju dengan XR).

Desain Visualisasi Data yang Baik Mengoptimalkan Ruang 3D

Banyak aplikasi visualisasi data di XR tidak menggunakan ruang 3D dan sering kali memiliki UI datar. Dalam Panduan Antarmuka Manusia (HIG) Apple, disarankan untuk menggunakan ruang sepenuhnya dan menghindari kontrol UI yang terlalu rumit:

Gunakan seluruh tampilan. Gunakan layar sebanyak mungkin untuk melihat dan menjelajahi dunia fisik dan objek virtual aplikasi Anda. Hindari mengacaukan layar dengan kontrol dan informasi yang mengurangi pengalaman imersif.

Visualisasi data harus dirancang secara intuitif, dan pengguna tidak memerlukan UI manual dan rumit untuk memahami cara menggunakannya—data mana yang x dan y, dan seterusnya. Sayangnya, banyak aplikasi visualisasi data fintech yang gagal dalam konsep desain ini dan pengguna dibiarkan kebingungan. Alih-alih menghemat waktu pengguna untuk mendapatkan pemahaman tentang data yang mereka tenggelamkan, mereka dibiarkan berjuang untuk menemukan cara-cara aneh untuk mengendalikan data dengan menu yang terlalu rumit di media baru. Banyak yang menghabiskan lebih banyak waktu mengklik 10 hingga 15 menu untuk menemukan data yang seharusnya dapat mereka muat dalam tiga interaksi sederhana.

Pada akhirnya, desain yang buruk ini membuat pengguna tidak dapat mengadopsi aplikasi data dalam teknologi yang imersif dan berkembang. Dalam buku John Maeda, *Laws of Simplicity* (MIT Press, 2006),⁸ ia menyatakan bahwa desain pada akhirnya harus menghemat waktu pengguna dengan UI sederhana.

“Penghematan Waktu Terasa Seperti Kesederhanaan”

Prinsip desain universal ini berlaku untuk kertas, desktop, seluler, dan spektrum komputasi spasial secara keseluruhan. Keanggunan kesederhanaan dalam desain masih dihargai dalam media baru dan visualisasi data yang baik menyampaikan desain yang bijaksana melalui pengalaman pengguna (UX) yang positif dalam aksesibilitas dan kemudahan penggunaan serta kemampuannya untuk meningkatkan kemampuan pengguna menjadi lebih baik. pemahaman data.

9.7 REPRESENTASI DATA, INFOGRAFIS, DAN INTERAKSI

Kami secara singkat menyebutkan jenis data dalam komputasi spasial sebelumnya dalam bab ini. Di bagian ini kami membuat perbedaan yang lebih jelas tentang subkategorisasi dan karakteristik data di berbagai platform dan bagaimana mereka dibuat.

Gambar 9-4 menampilkan tumpang tindih antara data statis dan dinamis yang divisualisasikan di berbagai platform, dengan data 2D dan 3D dalam bentuk cetak dan seluler, dan visualisasi dinamis interaktif dalam seluler, desktop, dan dalam komputasi spasial.



Gambar 9-4. Kategori jenis visualisasi data dalam 2D dan 3D yang ditampilkan dalam komputasi cetak, seluler, desktop, dan spasial

Apa yang Memenuhi Syarat sebagai Visualisasi Data?

Visualisasi data dapat dipisahkan menjadi dua kubu khusus: satu yang lebih abstrak, dan yang lain representasi literal data melalui rekonstruksi 3D, yang melibatkan langkah pra-pemrosesan yang lebih berat, terutama dalam pencitraan medis.

9.8 JENIS VISUALISASI DATA

Disebutkan sebelumnya, ada berbagai jenis data yang memenuhi syarat sebagai visualisasi data, yang dapat kita kategorikan ke dalam area ini:

- Abstraksi data—grafik garis, diagram lingkaran, grafik—mulai dari data dasar (x, y) yang diplot dalam bidang- z data dunia nyata
- Rekonstruksi 3D berdasarkan data kehidupan nyata (biotek, rekonstruksi teknologi kesehatan biologi manusia) dengan animasi dan interaktivitas

Banyak aplikasi lain di ruang AR, khususnya Microsoft HoloLens headmounted display (HMD) mengambil rekonstruksi 3D dari objek seperti MRI otak manusia dan menyatakan ini adalah visualisasi data. Meskipun ini adalah representasi data dari data dunia nyata dalam AR atau VR yang bertentangan dengan abstraksi data seperti diagram lingkaran, diagram garis, grafik, dan sebagainya, sering disalahartikan dengan objek 3D itu sendiri yang mudah diunduh melalui TurboSquid atau Sketchfab. Namun hal ini tidak serta merta memenuhi syarat sebagai visualisasi data, karena beberapa di antaranya hanyalah interpretasi longgar atau gambar

anatomi manusia tetapi tidak terhubung dengan data dunia nyata dan terkadang tanpa interaksi pengguna.

Beberapa aset ini hanyalah rekonstruksi 3D dan, dengan demikian, tidak dikategorikan (karena sering salah dikategorikan) sebagai visualisasi data; sebagian besar hanyalah interpretasi seniman dari mata telanjang dan belum tentu berdasarkan data dunia nyata. Perbedaan ini menarik garis antara keunggulan grafis dan insinyur yang memisahkan desain dan seni dari teknik. Visualisasi data sering kali merupakan hasil dari dua bidang yang menyebabkan produksi besar pengalaman luar biasa dalam komputasi spasial, tetapi kita harus membedakan juga antara "seni untuk seni" dan terlihat cantik, sementara yang lain sebenarnya dipengaruhi oleh substansi dan konten dari data aktual, yang memenuhi syarat sebagai visualisasi data nyata. Terlalu sering, eksperimen "cantik" dalam visualisasi data hampir tidak memenuhi syarat sebagai visualisasi data dan sebaliknya lebih merupakan ekspresi artistik.

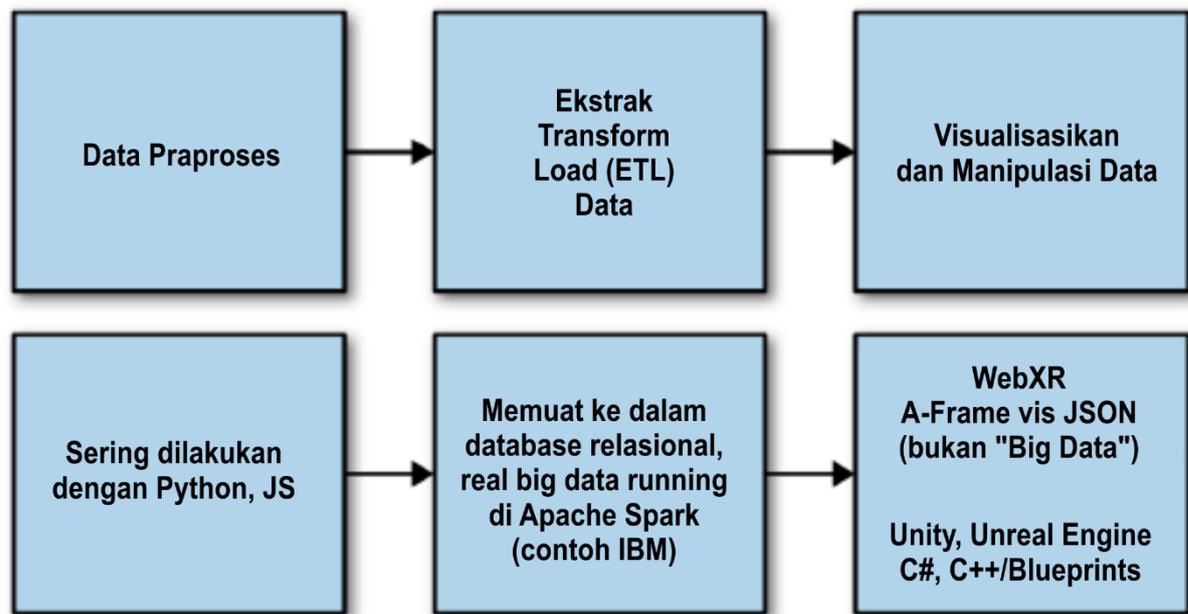
Mendefinisikan Perbedaan dalam Visualisasi Data dan Big Data atau Visualisasi Machine Learning

Visualisasi data mengungkapkan insight untuk memahami data tidak terstruktur dan data terstruktur. Banyak perusahaan data besar menangani kumpulan data besar, terkadang dalam skala terabyte hingga petabyte, seperti yang dijelaskan oleh tim Deck.GL open source Uber dengan presentasi WebXR-nya di pertemuan Silicon Valley Virtual Reality (SVVR) dan Konferensi Skala Besar Facebook. Aplikasi dan pengalaman yang melibatkan data besar secara real time dan kompleksitas arsitektur informasi serta jumlah komputasi membuatnya berbeda dari batang dan bagan infografis sederhana yang bukan visualisasi data besar, tetapi visualisasi atau representasi data sederhana yang dapat dengan mudah dibuat tanpa memperhatikan batasan dari apa yang dapat dirender oleh HMD secara real time. Visualisasi data 3D di ruang pembelajaran dalam (cabang pembelajaran mesin) serupa tetapi juga berbeda dengan yang ada di ruang XR mengingat bahwa mereka dapat melibatkan sejumlah besar data yang divisualisasikan dan bersifat 3D dan tidak semua visualisasi data dalam XR dapat melibatkan memvisualisasikan ukuran atau cakupan data yang besar.

Selain itu, "data besar" sebagai istilah itu sendiri sering digunakan secara bergantian untuk mendefinisikan sebagai besar karena mengukur data kuantitatif oleh jutaan pengguna, tetapi tidak selalu demikian. Sebagian besar data teknologi kesehatan dianggap sebagai data besar atau digunakan untuk mesin dan visualisasi data dan mungkin tidak harus besar dalam jumlah pengguna, tetapi diukur dengan ukurannya sendiri.

Cara Membuat Visualisasi Data: Jalur Pembuatan Visualisasi Data

Jalur pembuatan visualisasi data dari pengembangan asli dan pengembangan web melibatkan data pra-pemrosesan (yang sering kali merupakan pekerjaan membosankan yang dilakukan di antara para insinyur data, terutama untuk pemrosesan gambar di ruang medis). Ini dikenal sebagai Extract, Transform, and Load (ETL), di mana kami menyerap data dan mengubah serta mengonversinya ke format yang tepat untuk visualisasi dalam HMD. Gambar 9-5 menyajikan kedua contoh alur kerja.



Pipeline Visualisasi Data

Gambar 9-5. Jalur visualisasi data, menunjukkan di mana data mentah dimuat, diubah, dan divisualisasikan

9.9 WEBXR: MEMBANGUN VISUALISASI DATA UNTUK WEB

WebXR telah memperkenalkan beberapa kerangka kerja web selama beberapa tahun terakhir, mulai dari React 360 Facebook (sebelumnya dikenal sebagai ReactVR, berdasarkan kerangka kerja frontend ReactJS yang menonjol) dan A-Frame yang lebih populer, untuk membangun pengalaman Virtual Reality. A-Frame, dibuat oleh Diego Marcos dan Kevin Ngo (dengan dukungan Mozilla), menjadi kerangka kerja populer di komunitas kursus terbuka. Pengembang web telah membuat visualisasi data dengan ReactJS, d3.JS (pustaka visualisasi data masuk yang dibuat oleh Mike Bostock). Setiap insinyur perangkat lunak frontend yang memulai dengan sedikit pengalaman dalam VR dapat mulai membuat visualisasi data dengan data sumber terbuka. Pada Gambar 9-6, data dalam JavaScript Object Notation (JSON) disematkan ke dalam Scene A-Frame yang dimuat ke halaman web. Ini belum tentu dianggap sebagai "visualisasi data besar", tetapi visualisasi data, yang dapat dilihat pengguna dalam headset VR di dalam browser.

Tantangan Visualisasi Data di XR

Di seluruh spektrum komputasi spasial, konsep input adalah salah satu masalah desain yang paling menarik, unik, dan mutakhir. Suara kurang fokus karena kurangnya kedewasaan dengan NLP canggih dari Siri dan Cortana yang mampu menarik kumpulan data besar secara real time saat ini, yang banyak diantisipasi untuk berubah di tahun-tahun mendatang. Keterbatasan saat ini (tetapi juga mengantisipasi evolusi yang akan datang) dalam pengontrol, haptics, dan suara yang ingin diatasi oleh para insinyur dan perancang perangkat lunak dengan eksperimen dan solusi baru. Meskipun fokus di sini kurang pada suara, controller di VR, dan

kemajuan dalam teknik visi komputer untuk meningkatkan haptics di AR, keduanya sekarang memberikan keterjangkauan bagi pengguna yang tidak mungkin dilakukan di VR selama beberapa dekade terakhir, dan AR selama beberapa tahun terakhir. . Evolusi dalam HCI ini menimbulkan tantangan baru bagi desainer dan insinyur perangkat lunak dalam teknologi imersif dan baru, dengan berbagai kasus penggunaan untuk AR, VR, atau MR dalam spektrum XR.

Berikut adalah beberapa tantangan di seluruh spektrum komputasi spasial khusus untuk setiap jenis platform mengingat keterbatasan dan interaksinya yang kami harapkan akan berubah di masa depan seiring kemajuan media secara keseluruhan:

AR

Ada dua jenis yang perlu dipertimbangkan di sini: MobileAR (hampir tidak ada interaksi, jadi tidak masuk akal untuk melakukan banyak hal di sini) dan PCAR (masih tidak nyaman secara ergonomis dan belum siap komersial untuk sebagian besar HMD).

VR

VR membatasi pengguna dari dunia nyata dan menempatkan mereka di lingkungan tertutup saat berada di HMD. Karena itu, banyak desainer dan konsumen UX merasa bahwa HMD tidak dapat digunakan dan tidak dapat diakses. Sebagian dari ini karena keterbatasan teknis dengan optik meskipun diklaim siap untuk konsumen. Karena tantangan aksesibilitas ini dalam ergonomi, banyak aplikasi memiliki pengalaman yang lebih pendek untuk menghindari kelelahan.

Contoh Kasus Penggunaan Industri Visualisasi Data dari Visualisasi Data

Di bagian ini, saya memberikan referensi ke beberapa contoh visualisasi data yang dirancang dengan baik yang memuat kumpulan data besar secara real time dalam AR dan VR tetapi tetap bukan tanpa batasan desainnya sendiri.

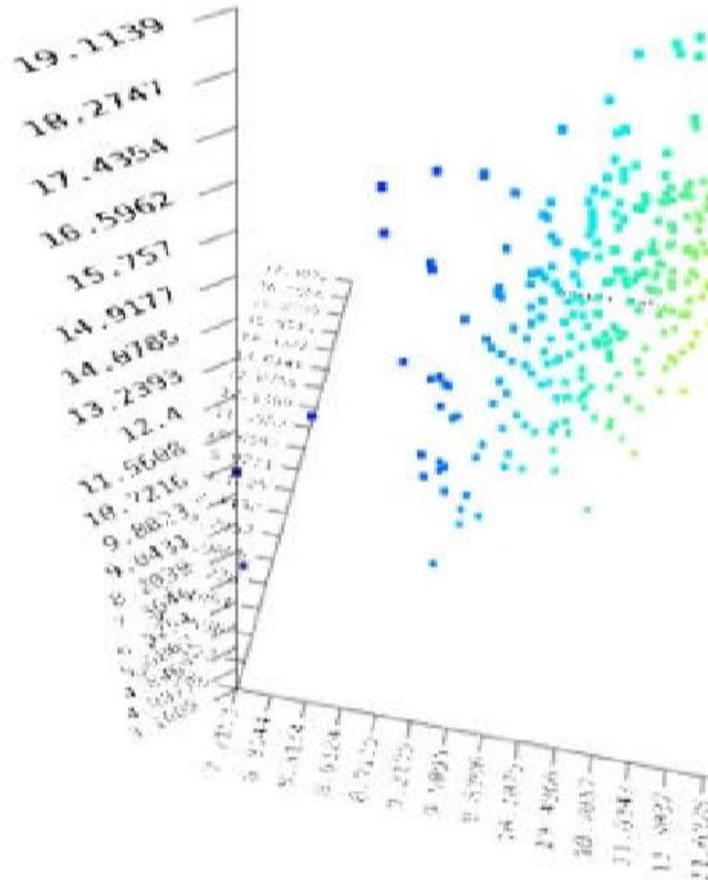
Gambar 9-7 menunjukkan visualisasi data IBM, yang memanfaatkan kerangka data sumber terbuka, Apache Spark untuk memuat data (termasuk data sumber terbuka, khususnya analisis Sentimen Twitter) secara real time.

Dalam ceramahnya di GDC 2017 “Immersive Data Visualization: AR in the Workplace,” Rosstin Murphy, mantan IBM, mempresentasikan penelitiannya tentang penggunaan AR untuk memvisualisasikan, menganalisis, dan memanipulasi data besar di tempat kerja. Tujuannya adalah untuk melakukan hal berikut:

[Gunakan] AR untuk menambah kotak peralatan ilmuwan data dan meningkatkan kecepatan dan kedalaman analisis mereka. Menjelajahi interaksi VR dalam konteks bisnis menunjukkan harapan, tetapi mengungkapkan tantangan objektif terhadap VR di lingkungan kerja. Tantangan ini termasuk biaya waktu untuk beralih antara peralatan VR dan mouse dan keyboard, menyulap lingkungan virtual 3D bersama antarmuka desktop 2D konvensional, dan memilih model controller perangkat keras yang tepat dan algoritme interaksi 3D...Teknologi AR, meskipun masih dalam tahap awal, elegan memecahkan banyak masalah.

Lebih khusus lagi, dalam perjuangan Murphy dengan ergonomi dan kemudahan penggunaan saat ia bertransisi dari notebook Jupyter di desktop PC-nya pada kenyataannya untuk

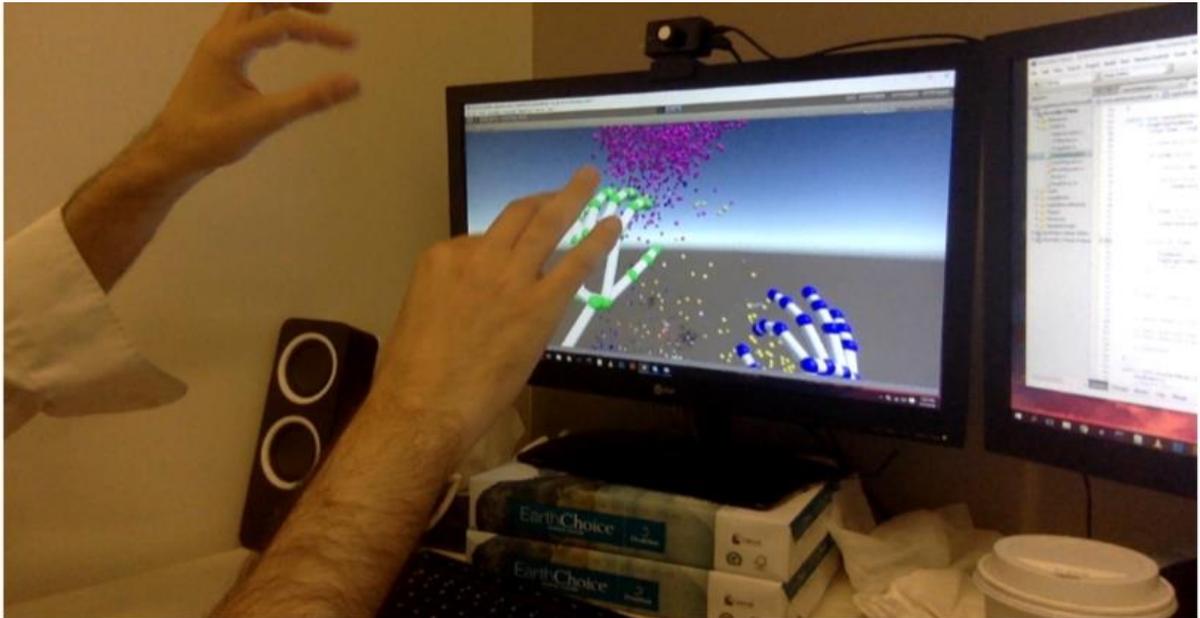
mengubah nilai data yang mengetik di keyboard sambil mengenakan headset Hololens di kepalanya. Bobot Hololens yang berat adalah penghalang masuk bagi banyak orang untuk menganggap HMD dapat digunakan. Manipulasi langsung karena kurangnya UI terintegrasi dengan Cortana (suara) dan menu visual lainnya (UI dalam AR) untuk menyesuaikan data, dan pembatasan manipulasi data masih agak tidak langsung karena Murphy membuat perubahan ini di desktop. Hal ini membuat kondisi AR saat ini (yang kami harapkan akan berubah di masa mendatang) sulit untuk diadopsi oleh pengguna.



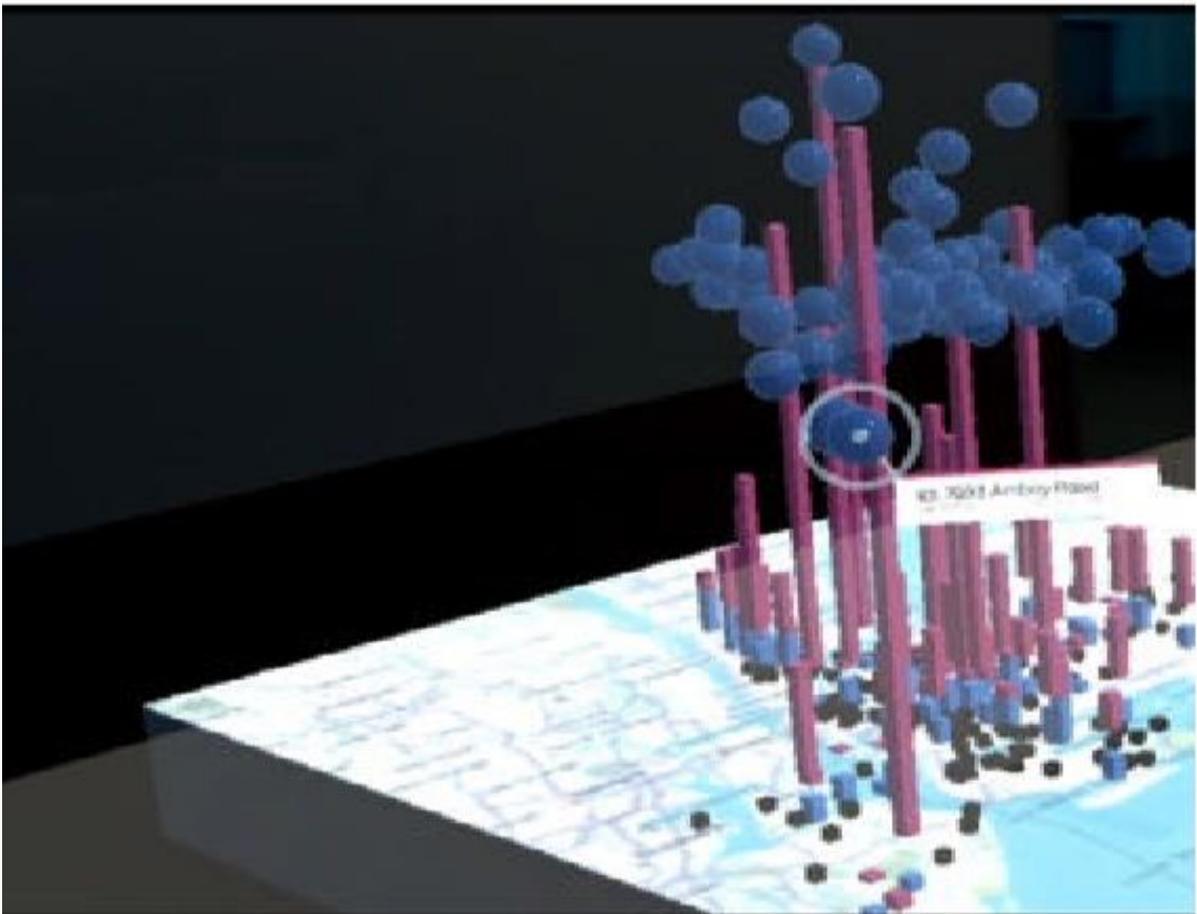
Gambar 9-6. Zac Canter memplot data suhu permukaan laut di WebXR menggunakan A-Frame

Rekonstruksi 3D dan Manipulasi Langsung Data Dunia Nyata: Struktur Anatomi di XR

Banyak visualisasi data adalah rekonstruksi 3D yang memuat data secara real time yang datang dalam bentuk struktur anatomi, yang dapat langsung dimanipulasi dan diedit dalam komputasi spasial. Interaksi komputasi spasial yang berbeda ini meningkatkan efisiensi keseluruhan alur kerja untuk berbagai vertikal B2B.



Gambar 9-7. Visualisasi data IBM di Microsoft Hololens menggunakan kerangka data sumber terbuka

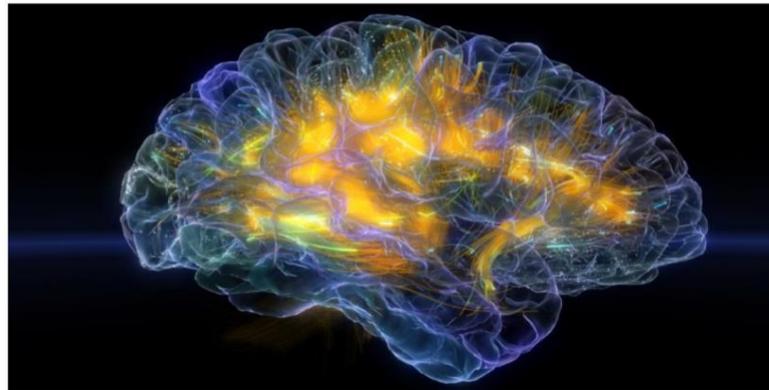


Gambar 9-8. Contoh visualisasi data yang dibuat oleh Rosstin Murphy saat dia berada di IBM, yang melapisi data pada peta

Sebagai bagian dari University of California, San Francisco, pemimpin dan ahli saraf Adam Gazzaley dan Tim Mullen, di laboratorium Gazzaley, Neuroscope, menciptakan visualisasi yang disebut "Otak Kaca" yang sering digunakan dalam demonstrasi berbagai modul untuk Meta 2 AR HMD.

Melihat Lebih Dekat Otak Kaca

Memfaatkan Unity3D, visualisasi data Glass Brain terdiri dari struktur otak, baik arsitektur jaringan dan saluran serat, yang diperoleh dari pemindaian otak resolusi tinggi (MRI dan *Diffusion Tensor Imaging* [MRI-DTI]). Aktivitas otak *real-time* dan interaksi fungsional antar jaringan ditumpangkan pada struktur otak menggunakan *high-density electroencephalography* (EEG). Ini adalah contoh yang bagus dari pemuatan visualisasi data secara real time. Visualisasi yang ditangkap, yang dapat Anda lihat pada Gambar 9-9, adalah milik pemain perkusi Mickey Hart.



Gambar 9-9. Glass Brain sering muncul dalam demonstrasi Meta 2 oleh pendiri dan CEO, ahli saraf Meron Gribetz (informasi lebih lanjut tersedia di Neuroscape University of California San Francisco)

9.10 MODUL VR PENCITRAAN MEDIS SURG TVA

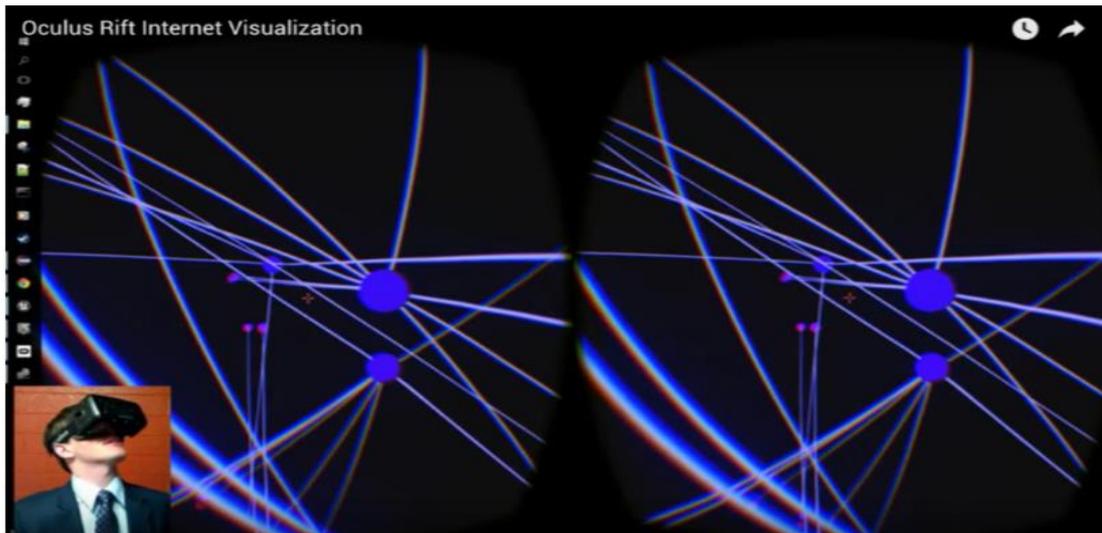
University of Toronto juga memiliki gambar *Digital Imaging and Communications in Medicine* (DICOM) nyata (tidak berwarna) yang direkonstruksi dari pemindaian MRI otak manusia dan gambar medis lainnya dari struktur anatomi manusia.

9.11 HOLODEK MEDIS—DICOM

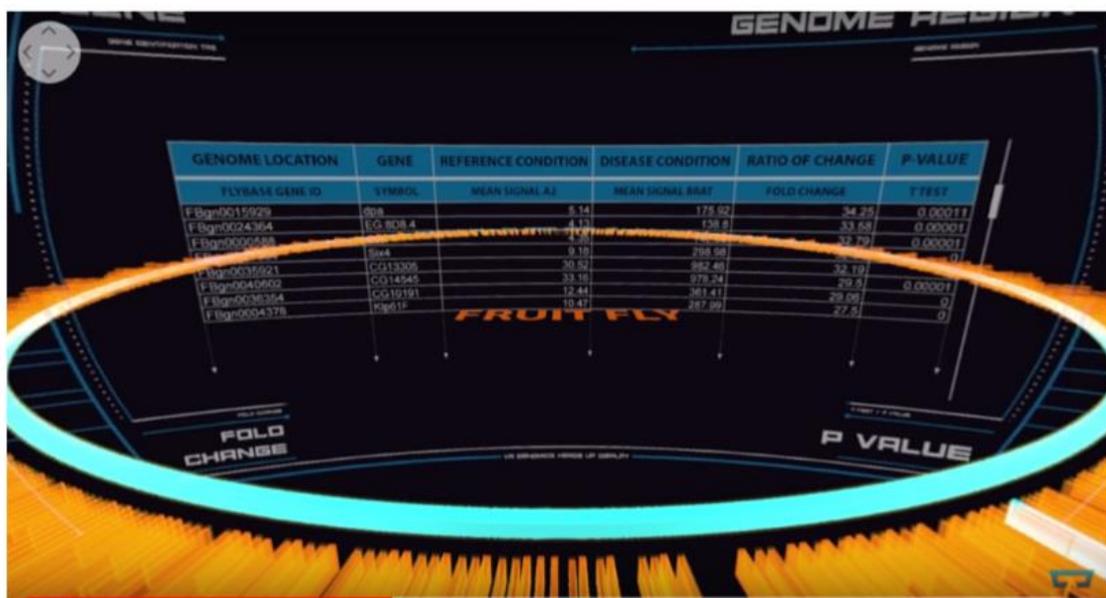
Medical Holodeck, didirikan oleh ahli radiologi Swiss yang menciptakan alat yang memungkinkan ahli radiologi dan profesional medis lainnya untuk mengambil gambar DICOM mereka ke dalam VR dan dapat mengurangi jumlah waktu yang dihabiskan dengan segmentasi gambar (sering dilakukan secara manual melalui beberapa algoritma pembelajaran mesin dasar) . Ini adalah tugas yang membosankan dalam keseluruhan jalur dalam rekayasa data yang melibatkan penguraian data dan memutar irisan berbagai gambar MRI di MATLAB, yang sebanding dengan kerning dan tweaking setiap perubahan mikro yang diperlukan dengan kebanyakan ekosistem retak dari alat desktop 3D. Holodeck Medis memungkinkan ahli radiologi, profesional medis, dan peneliti untuk berfokus pada substansi masalah mereka dalam pekerjaan mereka yang melibatkan masalah medis aktual yang ingin mereka analisis,

Kekuatan Augmented dan Virtual Reality dalam Bisnis (Dr. Agus Wibowo)

seperti menemukan tumor, membuat sayatan yang lebih tepat untuk pembedahan, menemukan korelasi ke berbagai patologi, dan meningkatkan efisiensi untuk keseluruhan jalur penelitian untuk penemuan obat dengan memungkinkan beberapa interaksi desain dalam komputasi spasial yang lebih langsung. Pendekatan serupa ditemukan oleh ahli radiologi Stanford dalam berbagai studi kasus, seperti yang dengan jelas didokumentasikan oleh Dilan Shah dalam Bab 11 buku ini.



Gambar 9-10. Visualisasi data Timothy Clancy menunjukkan subsampel pengindeksan berbagai halaman internet menggunakan Unreal Engine

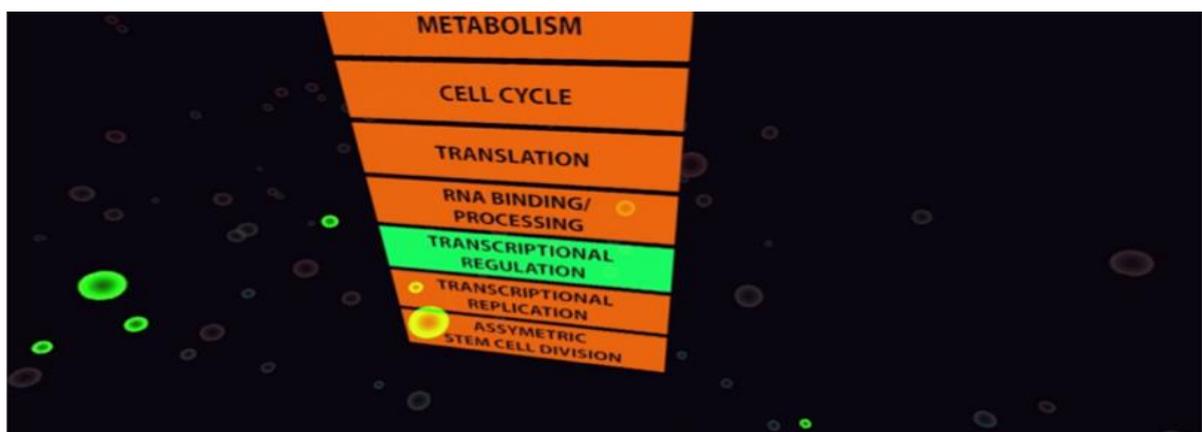


Gambar 9-11. Visualisasi data yang dibuat dari proyek yang dibuat di Unreal Big Data hackathon ini memiliki beberapa UI datar dengan tabel yang menampilkan data yang berasal dari genom lalat buah

Visualisasi Data untuk Semua Orang: Visualisasi Data Berbasis Sumber Terbuka di XR

Siapa pun dapat menggunakan data sumber terbuka untuk membuat visualisasi data pertama mereka. Jika Anda baru memulai visualisasi data, tidak perlu merasa terintimidasi. Anda dapat menemukan banyak sekali kumpulan data di Kaggle (sekarang dimiliki oleh Google) dan berbagai area lainnya (bergantung pada vertikal teknologi) yang memungkinkan pengembang baru membuat visualisasi data yang berarti. Timothy Clancy juga berhasil membuat visualisasi datanya sendiri dengan mencoba mengindeks segmen dari berbagai potongan halaman di internet dan menciptakan visualisasi data yang hebat di Unreal Engine, yang dapat Anda lihat pada Gambar 9-10.

Pada tahun 2016, Unreal Engine juga menyelenggarakan hackathon dengan data Wellcome Trust (data biotek), di mana pemenangnya, Hammer Head VR, membuat browser VR yang menganalisis genom lalat buah, seperti yang digambarkan pada Gambar 9-11 dan 9-12.



Gambar 9-12. Visualisasi data yang dibuat dalam proyek hackathon Unreal Big Data oleh HammerHeadVR ini menggunakan bola yang menyala hijau ketika "dilyangkan" seperti di web, ketika pengguna berinteraksi dengan menu melalui controller VR

Visualisasi Data Protein

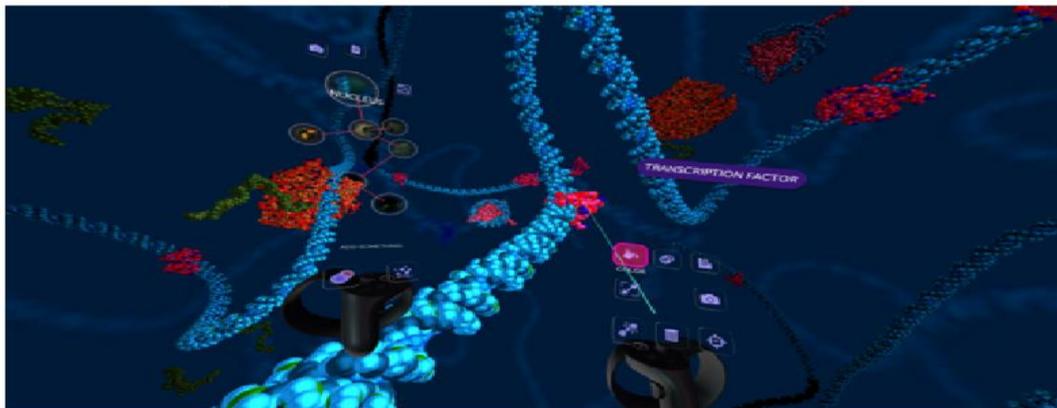
Platform 10K dari Dynamoid (lihat Gambar 9-13) juga membuat UI dinamis untuk dapat melihat Protein Data Bank (PDB) open source sebagai referensi untuk berbagai protein DNA dalam aplikasi VR-nya yang memungkinkan pengguna untuk dapat melihat up, drop in, view, protein dengan beberapa kemampuan dan mengubah ukuran berjalan secara real time. Tidak seperti banyak teknologi kesehatan dan aplikasi biotek lainnya yang sering merupakan rekonstruksi data 3D statis dengan interaksi nol atau seni 3D teknis sederhana (file OBJ atau FBX dari tampilan pemindaian MRI atau protein DNA yang dibuat-buat), sistem 10K secara dinamis menarik berdasarkan data nyata yang digunakan oleh praktisi di lapangan.

Tutorial Praktis: Cara Membuat Visualisasi Data dalam Komputasi Spasial

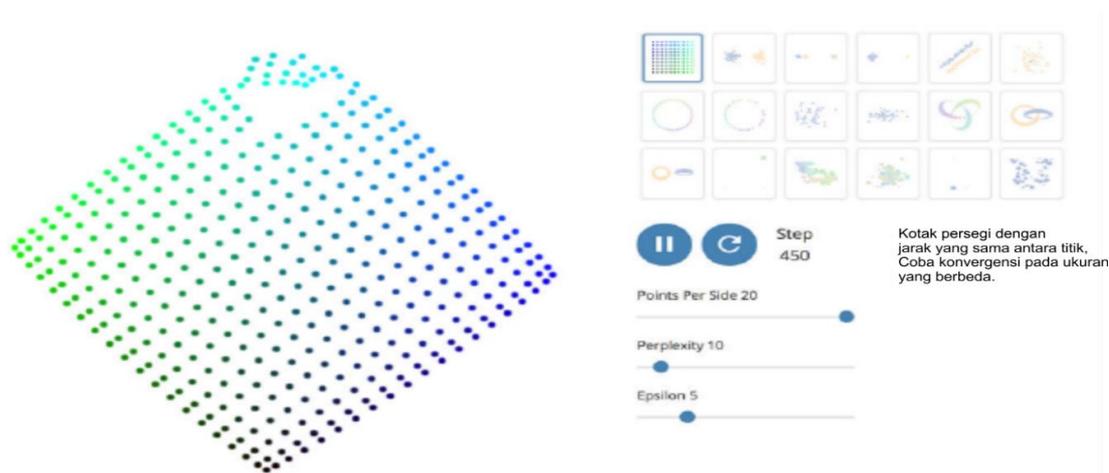
Sekarang kita memiliki kerangka kerja untuk desain visualisasi data yang baik di berbagai tipe data, platform, dan HMD, kita dapat mempelajari cara membuat visualisasi data dinamis dalam komputasi spasial.

Sisa dari bab ini menjelaskan berbagai referensi oleh para insinyur perangkat lunak baru dan berpengalaman tentang berbagai pendekatan tentang cara membuat visualisasi data dalam komputasi spasial mulai dari web dan platform asli, pada A-Frame, ReactJS, D3.JS (WebXR) menggunakan JavaScript (JS) dan Unity menggunakan C#.

Untuk mendapatkan pemahaman yang lebih baik tentang visualisasi data menggunakan data besar, beberapa contoh terbaik dapat ditemukan oleh ilmuwan data dan insinyur pembelajaran mesin terkemuka yang menampilkan data 3D, seperti yang ditunjukkan pada Gambar 9-14, karena kutukan pengurangan dimensi, yaitu berbagai visualisasi Principal Component Analysis (PCA) dan t-SNE. Meskipun kami menyadari bahwa ini tidak dalam komputasi spasial sebagai media, mereka memberikan dasar yang kuat untuk memahami memvisualisasikan kumpulan data multidimensi, kompleks, dan terkadang sangat besar, yang memberi pencipta baru beberapa prinsip dasar yang dapat mereka terapkan untuk membuat visualisasi baru dalam spasial. komputasi.

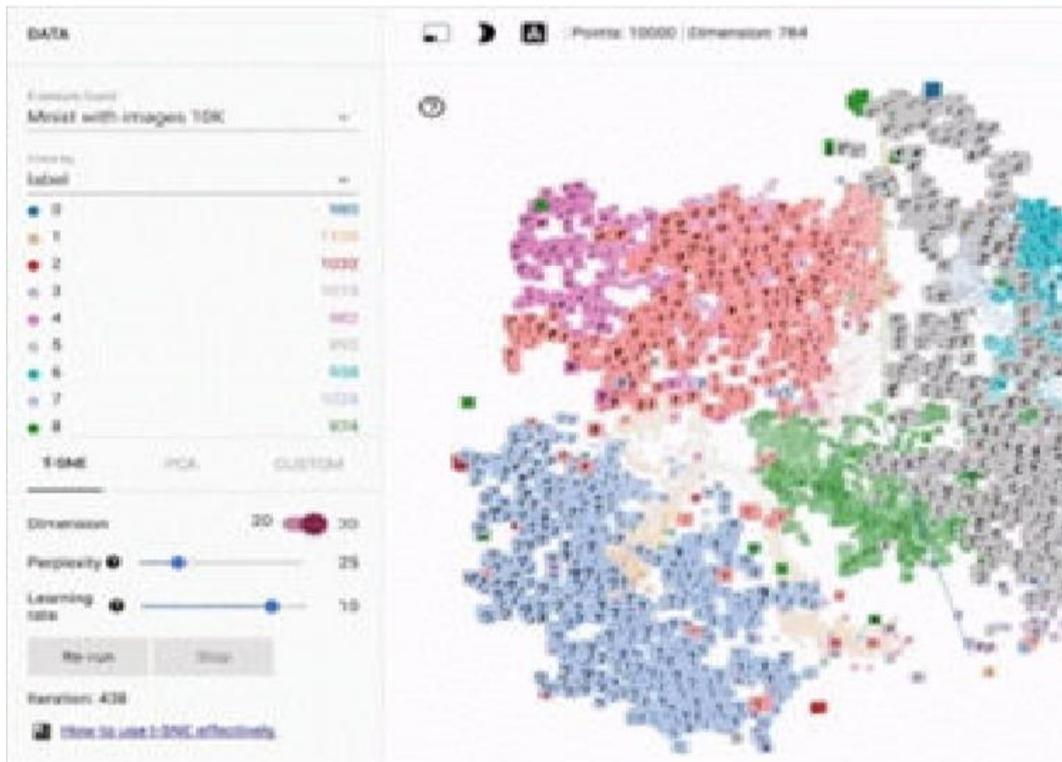


Gambar 9-13. Platform 10K yang didirikan oleh Laura Lynn Gonzalez dari Dynamoid, mendemonstrasikan cara menampilkan visualisasi data dalam VR secara efektif dalam konteks



Gambar 9-14. Jurnal pembelajaran mesin distill.pub memiliki banyak visualisasi pembelajaran mesin interaktif yang membantu peneliti lebih memahami data mereka

Untuk titik awal yang baik dalam memahami dua jenis visualisasi data yang biasa digunakan dalam komunitas pembelajaran mesin ini, lihat Projector TensorFlow dan Jurnal ML yang didukung YCombinator karya Google, distill.pub dengan karya juga oleh Ian bersama dengan pemimpin pembelajaran mesin Google, Chris Olah.



Gambar 9-15. Alat Fernanda Viegas dan Matt Wattenberg dengan kerangka kerja pembelajaran mesin Google TensorFlow, memvisualisasikan Analisis Komponen Utama (PCA) dan visualisasi lainnya

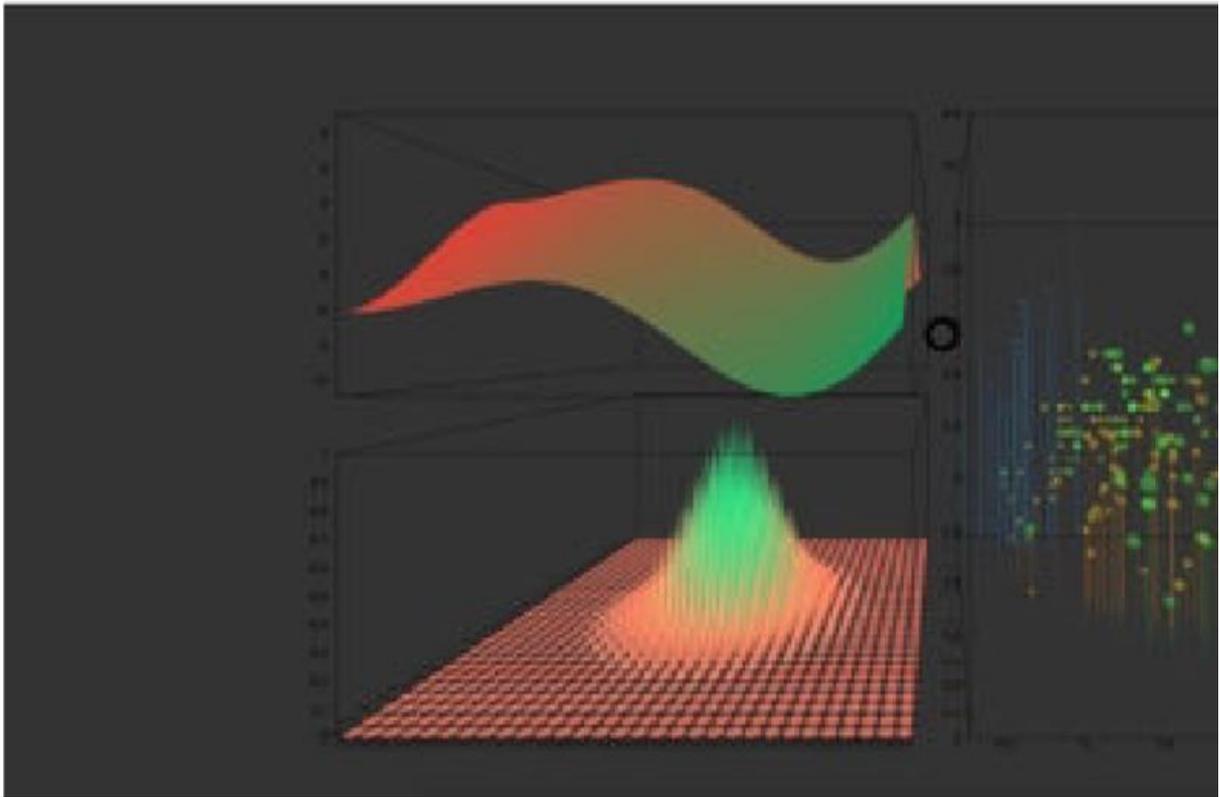
Di alam semesta Windows, C# adalah bahasa pilihan dan dengan demikian memporting visualisasi data ke Microsoft HoloLens AR HMD, yang biasa digunakan untuk berbagai kasus penggunaan B2B. Kami ingin mencatat bahwa AR tidak terbatas pada HMD ini saja. Jumlah konten pengembang independen yang dibuat di mobileAR dengan pengenalan ARKit di iOS dan ARCore di Android dirinci di Bab 5 oleh salah satu pendiri 6D.ai Matt Miesnieks dan Profesor Victor Prisacariu.

Meskipun visualisasi data dan pembelajaran mesin kurang menonjol di mobileAR, kami menyadari bahwa beberapa proyek sedang dimulai di sana yang mungkin bergabung dengan HMD dan kacamata masa depan. Kami mengantisipasi lebih banyak lagi rilis AR HMD dan kacamata di masa mendatang, termasuk versi kit pengembang baru dari Microsoft HoloLens, Magic Leap, Apple, lainnya dari China dan Israel, dan bahkan yang lain yang mungkin tidak kami ketahui yang diperkirakan akan diluncurkan pada tahun 2020 rilis atau lebih baru (di luar tanggal rilis awal antologi ini).

Cara Membuat Visualisasi Data: Sumber Daya

Meninjau kembali prinsip-prinsip di awal bab ini, pembuat konten harus mempertimbangkan sejumlah langkah yang akan membantu mereka menentukan pendekatan terbaik untuk visualisasi data yang efektif dalam komputasi spasial.

Salah satu contoh bagus untuk memulai di WebXR berasal dari Mustafee Saiffee, yang membuat kerangka kerja yang menggabungkan A-Frame dengan React (untuk manipulasi DOM) dan D3 (untuk visualisasi data) untuk menghasilkan visualisasi di VR, salah satunya dapat Anda lihat pada Gambar 9-16.



Gambar 9-16. VR-Viz menyediakan komponen reaksi tingkat tinggi untuk menghasilkan visualisasi 3D di webVR menggunakan ReactJS, A-Frame, dan D3.JS

Ini adalah salah satu dari banyak contoh yang dapat Anda temukan di blockbuilder dan sumber daya WebXR lainnya yang menunjukkan pemahaman yang baik tentang prinsip-prinsip desain yang solid untuk membangun visualisasi data komputasi spasial dengan sukses.

9.12 KESIMPULAN

Saya harap bab ini telah mengungkap data dalam XR dan memberikan contoh teknik, praktik terbaik, dan alat praktis untuk membuat visualisasi data yang indah dan fungsional yang dioptimalkan untuk media ini.

Meskipun bab ini berfokus terutama pada visualisasi data dan hanya secara tangensial menyentuh visualisasi pembelajaran mesin karena keterbatasan panjang yang melekat pada buku ini, sangat dianjurkan untuk melihat repositori GitHub tambahan dan referensi tutorial

kami untuk melanjutkan dengan materi langsung untuk memulai membuat data dan visualisasi pembelajaran mesin dalam komputasi spasial. Lihat daftar sumber daya kami untuk berbagai jenis data dan visualisasi pembelajaran mesin.

Seperti kebanyakan standar komputasi spasial, WebXR masih terus berkembang, tetapi insinyur dan perancang perangkat lunak sudah dapat memulai dengan kerangka kerja sumber terbuka yang ada ini untuk membuat visualisasi data dan pembelajaran mesin pertama mereka.

Glossarium

Human Computer Interaction (HCI)

Sebuah istilah yang digunakan untuk menggambarkan interaksi antara manusia pengguna dan komputasi

User Interface (UI)

Sebuah istilah yang digunakan untuk menggambarkan representasi visual dari sebuah aplikasi

Affordances
Hasil desain yang baik yang tidak memungkinkan pengguna untuk memiliki manual instruksi untuk mengetahui cara menggunakan

t-SNE

Secara intuitif Dari peneliti AI terkemuka, Geoff Hinton, *Stochastic Neighbor Embedding (SNE)*, sejenis visualisasi mesin

A-Frame

Kerangka web untuk membuat visualisasi dan aplikasi serta pengalaman XR lainnya, dibuat oleh Diego Marcos dan Kevin Ngo dengan dukungan Mozilla.

Block builder

Alat prototyping visualisasi yang dibuat oleh Ian Johnson

D3.js

Pustaka yang dibuat oleh Michael Bostock

Observable

Alat yang dibuat oleh Michael Bostock

VR-Viz

Komponen React berbasis A-Frame dari visualisasi data dalam VR dan AR dibuat oleh Mustafa Saifee

BAB 10

PERILAKU DAN KARAKTER AI

10.1 PENGANTAR

Virtual Reality (VR) menghidupkan janji pengalaman sensorik yang lebih mendalam daripada bentuk hiburan digital sebelumnya (film, video game, novel interaktif, dll.). Pengalaman yang lebih mendalam berarti pengalaman yang lebih emosional dan berdampak. Banyak dari kita mengingat pengalaman pertama mereka di VR dengan emosi. Banyak yang telah menyaksikan kejutan, keajaiban, dan antusiasme yang dapat ditimbulkan oleh penyelaman pertama dalam virtualitas pada orang lain. Bisa dibilang tidak ada media lain yang bisa membangkitkan emosi sebesar itu hanya dalam beberapa detik. Fakta ini telah diakui oleh komunitas yang kini memanfaatkan kekuatan emosional ini untuk menghasilkan aplikasi VR di domain yang sensitif seperti pelatihan dan terapi pribadi.

Dengan menggunakan geolokalisasi dan menggabungkan elemen realitas dengan elemen virtualitas, *augmented reality* (AR) dan *Mixed Reality* (MR) mendefinisikan taman bermain baru bagi seniman, pendongeng, dan pengembang game untuk dijelajahi. Ini sebagian besar masih wilayah yang belum dipetakan, tetapi kegilaan yang diamati untuk pengalaman AR primitif seperti Pokémon Go R memberi kita gambaran sekilas tentang potensi besar media ini untuk menghasilkan kesenangan dan permainan. Kemungkinan berburu harta karun di kota asal Anda (di mana Pokémon Go R adalah bentuk primitif) atau mempertahankan jalan Anda melawan pemain dari tim lain dengan menempatkan baterai artileri di atap rumah Anda sendiri pasti akan menarik banyak penonton gamer dan bukan gamer.

Aplikasi profesional AR juga datang dalam jumlah, dari solusi penggantian desktop, hingga alat operasi lapangan dan alat visualisasi in situ. Kegembiraan seputar AR dapat diukur dengan jumlah startup dan perusahaan mapan yang saat ini mengembangkan aplikasi AR, meskipun belum ada perangkat AR yang mampu menjangkau basis konsumen yang besar. VR, AR, dan MR—dikumpulkan di bawah akronim “XR”—adalah media yang sangat menjanjikan, dan bab ini hadir untuk membantu Anda menerapkan aplikasi pertama dari teknologi ini.

Selama lima tahun terakhir, kecerdasan buatan (AI) telah menjadi kata kunci. Ini didorong oleh keberhasilan pembelajaran mesin yang mengesankan secara umum dan subdivisinya, pembelajaran mendalam, khususnya, 11, 26, 37 dalam domain penambangan data dan persepsi robot (penglihatan komputer, pemrosesan bahasa alami [NLP], pengenalan gerakan, dll.) serta dalam pembuatan konten (misalnya, gambar, suara, animasi). Ini memperbarui minat untuk disiplin ini di luar impian terliar banyak profesional jangka panjang. Kemungkinan penerapan AI dan pembelajaran mesin di XR banyak, dan bab-bab sebelumnya telah membahas beberapa di antaranya. Tujuan bab ini bukan untuk mencakup semua aplikasi potensial AI dan pembelajaran mesin untuk pengembangan aplikasi XR. Sebaliknya, kami

mengadopsi pendekatan berlawanan yang dimulai dari masalah—menghasilkan perilaku di lingkungan virtual dan semi-virtual—dan kemudian meninjau pendekatan teknis yang tersedia.

Secara historis, penelitian dan pengembangan dalam domain XR telah dipromosikan sebagian besar oleh komunitas game. Ini diilustrasikan oleh fakta bahwa mesin video game seperti Unity dan Unreal saat ini mendukung sebagian besar aplikasi XR yang diproduksi di seluruh dunia. Ada alasan teknis dan budaya yang jelas di balik fakta ini. Diantaranya adalah kesamaan masalah teknis yang dihadapi dalam pengembangan video-game dan XR: kendala waktu nyata dengan sumber daya yang terbatas; tekanan untuk menciptakan pengalaman daripada memecahkan masalah; dan kemungkinan untuk menciptakan ilusi— yaitu, untuk membuat pengguna percaya bahwa sistem memiliki kemampuan padahal sebenarnya hanya menirunya.

Akibatnya, komunitas XR mencari solusi AI dari industri video-game untuk menghidupkan kreasi mereka. Demikian pula, kami memulai studi kami dengan melihat teknik AI game dan mendiskusikan kemampuannya untuk mengatasi tantangan XR. Ini akan membawa kita untuk menekankan beberapa batasan mereka dan kemudian mendiskusikan alternatif yang ada dan upaya penelitian saat ini untuk membawa game AI ke level berikutnya. Dalam perjalanan ini, kita harus mengungkapkan pendapat tentang pendekatan yang paling menjanjikan dan ke mana arah upaya penelitian. Pendapat ini diarahkan oleh pengalaman beberapa tahun dalam menerapkan pengambilan keputusan dan sistem pembelajaran mesin di bidang industri lainnya, terutama dalam robotika otonom. Tetapi seperti pendapat apa pun, mereka dapat mengalami perselisihan.

Bab ini membahas pendekatan teknis yang tersedia untuk menghasilkan perilaku dalam video game dan XR. Sebelum memasuki inti masalah, kami membingkai diskusi dengan memperdebatkan secara singkat pengertian perilaku (di bagian “Perilaku”). Kami menekankan bahwa perilaku dapat dipertimbangkan pada skala yang berbeda dalam hal waktu dan ruang, dari tugas monitor sensorik skala rendah hingga perencanaan aktivitas skala tinggi. Perilaku juga dapat dilampirkan ke berbagai jenis entitas: jelas ke satu karakter yang tidak dapat dimainkan (NPC), tetapi juga ke sekelompok NPC atau ke seluruh dunia game ketika kita mempertimbangkan penceritaan dan narasi interaktif. Anda akan melihat bahwa skala di mana kita menganggap perilaku memiliki pengaruh lebih besar pada sifat masalah yang kita hadapi—dan seterusnya pendekatan teknis untuk diadopsi—daripada objek yang kita terapkan perilakunya.

Bagian utama pertama dari bab ini, “Praktik Saat Ini: AI Reaktif,” mensurvei teknik-teknik canggih yang saat ini mendukung sebagian besar AI game. Praktik saat ini sederhana: pengembang game harus menulis “dengan tangan” semua perilaku yang diperlihatkan entitas yang dikendalikan saat runtime. Tingkat kecanggihan terendah adalah urutan tindakan yang sepenuhnya skrip dan tidak dapat diubah yang dijalankan apa pun yang terjadi saat runtime. Aplikasi yang lebih berkembang menggunakan arsitektur reaktif di mana tindakan yang dieksekusi bergantung pada pengamatan yang dilakukan oleh agen, dengan peristiwa permainan yang berbeda memicu perilaku yang berbeda. Namun demikian, pengembang perlu merancang dan mengimplementasikan hubungan dari pengamatan ke tindakan

sepenuhnya dengan tangan. Melihat yang pertama (urutan tindakan tetap) sebagai kasus tertentu dari tindakan selanjutnya (tindakan yang bergantung pada pengamatan), kami mengumpulkan keduanya di bawah nama mesin keadaan-terbatas (FSM) dan pohon perilaku (BT) untuk membantu pengembang mengatur pengetahuan dalam cara yang lebih mudah dibaca dan dipelihara daripada kode biasa. Mereka hanyalah bahasa pemrograman visual.

Alasan utama mengapa paradigma ini mendominasi pengembangan game adalah karena paradigma ini memungkinkan kontrol total oleh pengembang atas perilaku yang ditampilkan dalam game. Seperti yang ditekankan di banyak tempat, AI reaktif sangat baik dalam menjelaskan cara melakukan sesuatu. Namun, itu tidak memberikan bantuan apa pun dalam memutuskan apa yang harus kita lakukan. Kami percaya bahwa, untuk alasan yang berbeda, AR dan VR menempatkan model ini pada tantangan. Akibatnya, kita perlu memeriksa alternatif lain yang tersedia di AI akademik untuk pengembang game dan XR AI reaktif.

Dalam paradigma ini, tindakan diproduksi satu per satu sebagai fungsi dari keadaan lingkungan, dan aturan keputusan sepenuhnya dirancang oleh pengembang. AI tidak menunjukkan kemampuan pemecahan masalah atau pengambilan keputusan yang nyata, itu hanya mengulangi apa yang diperintahkan untuk dilakukan. Dengan ini, kita dapat menulis AI reaktif hanya dalam kode biasa (mis., C#, C++, Java). Namun, perilaku biasanya merupakan struktur kompleks yang ingin atau perlu kita uji, debug, tingkatkan, dan augmentasikan saat aplikasi sedang dikembangkan. Kode biasa bersifat samar, dan perancang AI mungkin bukan pembuat kode ahli. Untuk alasan ini, alat telah dikembangkan untuk membantu perancang menyusun perilaku.

Ini membawa kita ke bagian besar kedua dari bab ini, “Lebih Banyak Kecerdasan dalam Sistem: AI Deliberatif,” yaitu tentang AI deliberatif dan perencanaan otomatis. Dalam paradigma ini, AI diberdayakan dengan kemampuan pemecahan masalah dan pengambilan keputusan yang nyata. Perilaku dihasilkan dengan memecahkan masalah yang terdefinisi dengan baik yang dibingkai menggunakan model lingkungan. Pengembang harus membuat model ini—yang kami sebut domain perencanaan—sedemikian rupa sehingga perilaku yang diinginkan dihasilkan tetapi tidak dapat memperbaiki sendiri setiap aspek perilaku. Pendekatan ini menghindari banyak batasan AI reaktif. Secara khusus, sangat baik dalam memutuskan apa yang harus dilakukan. Meskipun umum dalam domain robotika otonom, pendekatan ini sebagian besar telah diabaikan oleh komunitas AI game, dan hanya sebagian kecil produk komersial yang menggunakan teknologi ini saat ini. Hal ini dapat dijelaskan dengan hilangnya kendali yang menyertai pendekatan ini. Lebih dari sekadar pesaing, kami percaya bahwa itu harus dilihat sebagai pelengkap AI reaktif.

Perencanaan otomatis harus digunakan untuk memecahkan masalah keputusan yang sulit, sedangkan AI reaktif harus digunakan untuk menjelaskan detail tentang cara mengeksekusi keputusan. Penjelasan lain untuk kurangnya minat ini adalah banyaknya pekerjaan yang diperlukan untuk membuat sistem AI reaktif yang berfungsi, membuat seluruh pendekatan tidak layak secara ekonomi untuk banyak studio kecil. Dalam hal itu, upaya yang dilakukan di Unity Technologies untuk menyediakan alat AI deliberatif siap pakai dapat memiliki dampak penting di pasar. AI deliberatif bekerja dengan memecahkan masalah, yang

setara dengan mencari ruang solusi yang besar untuk solusi optimal. Setiap algoritma pencarian memiliki batasan dalam hal ukuran masalah yang dapat ditanganinya sebelum mempengaruhi fluiditas game/XR dengan menghabiskan terlalu banyak waktu. Untuk menghilangkan batasan ini, kita dapat beralih ke paradigma AI ketiga: pembelajaran mesin.

Pembelajaran mesin adalah pusat dari sebagian besar buzz seputar AI saat ini, khususnya cabang yang disebut pembelajaran mendalam. Pencapaian spektakuler teknologi ini dalam domain penambangan data dan persepsi robot—memahami output serangkaian sensor—telah membuka jalan bagi spekulasi terlarang tentang dampak AI di masa depan terhadap teknologi dan masyarakat. Cabang pembelajaran mesin yang berfokus pada pengambilan keputusan dan generasi perilaku disebut Reinforcement Learning.

Dalam paradigma ini, AI belajar untuk melakukan perilaku yang benar melalui trial and error, berinteraksi dengan video game. Ini dipandu dalam proses ini oleh hadiah virtual yang diberikan setiap kali AI mencapai tujuan. Hubungan antara disiplin ini dan video game secara historis sangat erat. Komunitas riset telah mengakui relevansi video game sebagai test bed untuk algoritma Reinforcement Learning. Video game arcade sederhana sekarang umum digunakan untuk mengevaluasi kinerja algoritma Reinforcement Learning baru dalam publikasi akademik. Tampaknya ini menunjukkan masa depan yang bagus untuk teknik pembelajaran mesin yang digunakan untuk menghasilkan perilaku dalam game dan XR.

Tujuan bab ini bukan untuk memberikan survei yang lengkap dan mendalam tentang semua teknik game-AI. Satu bab tidak akan cukup. Bagian Referensi dari bab ini mencakup survei bidang game AI,^{3, 27} dan penyelaman mendalam ke dalam penelitian saat ini.^{32, 33, 34} Tujuan bab ini adalah untuk memberikan survei tingkat tinggi dari pendekatan yang tersedia untuk mengatasi tantangan XR dan untuk menekankan kekuatan dan kelemahan mereka. Selain itu, kami tidak membatasi studi kami pada “AI game resmi”, tetapi juga meminjam konsep dari AI akademis. Kami berharap visi ini akan memandu pengembang XR awal untuk membuat pilihan desain yang tepat dalam hal mendesain perilaku.

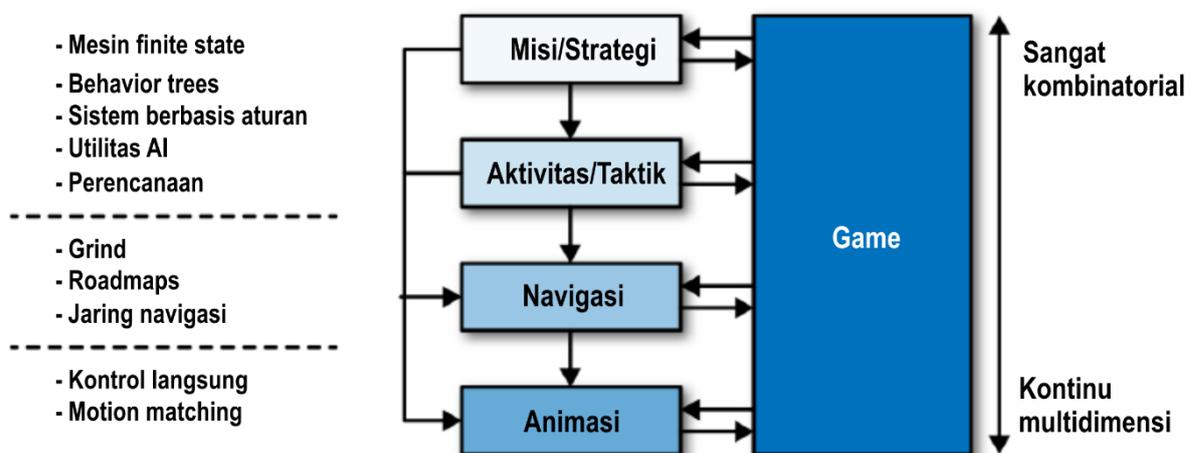
Perilaku

Ketika kita mencari kata "perilaku," kita dengan cepat menemukan definisi berikut: "Cara seseorang bertindak atau berperilaku (...)." Jadi, perilaku dibentuk oleh urutan tindakan yang dilakukan oleh subjek. Kami menekankan bahwa di balik urutan tindakan ini ada serangkaian keputusan untuk bertindak dengan cara itu. Jadi, kita melihat perilaku urutan keputusan untuk bertindak yang ditunjukkan oleh suatu entitas.

Ini adalah definisi yang sangat luas, terutama karena dapat dipahami pada banyak skala yang berbeda. Untuk mengilustrasikan ini, pertimbangkan contoh dari robotika otonom. Arsitektur kontrol robot biasanya hadir dalam tiga modul besar: persepsi, yang memiliki tugas memahami input dari rangkaian sensor yang dilengkapi robot, dan untuk membangun model dunia terkonsolidasi darinya; keputusan, yang memiliki tujuan untuk memutuskan tindakan selanjutnya yang akan dilakukan; dan kontrol, yang mencoba untuk mengeksekusi keputusan yang diambil dengan setepat mungkin.

Bab ini berfokus pada isu-isu yang terkait dengan pengembangan lapisan keputusan. Lapisan keputusan itu sendiri dibagi dalam hierarki modul yang membuat keputusan pada

waktu dan skala yang berbeda. Misalnya, dalam mobil yang mengemudi sendiri, lapisan keputusan biasanya terdiri dari tiga modul: navigasi, yang merencanakan seluruh perjalanan (seperti aplikasi navigasi di ponsel atau di mobil Anda); perilaku—untuk dipahami di sini dalam pengertian yang lebih terbatas daripada di sisa bab ini—yang menentukan tindakan taktis seperti perubahan jalur dan menunggu di tanda berhenti; dan perencanaan gerak, yang mencoba mengemudikan mobil sambil tetap berada di jalan dan menghindari rintangan seperti lubang. Arsitektur serupa dapat (dan harus) digunakan untuk NPC video-game. Misalnya, kita dapat menyusun empat lapisan keputusan: memutuskan tentang misi dan tujuan jangka panjang, perencanaan aktivitas, navigasi, dan animasi. Gambar 10-1 memberikan contoh arsitektur pengambilan keputusan hierarkis untuk NPC. Keputusan didistribusikan ke beberapa modul yang bekerja pada skala dan frekuensi yang berbeda.



Gambar 10-1. Arsitektur kontrol hierarkis untuk NPC dalam video game atau aplikasi XR

Kiri: teknik yang biasa digunakan untuk menangani berbagai tingkat pengambilan keputusan; kanan: sifat ruang masalah berkembang dari optimasi berkesinambungan ke kombinatorial saat kita menaiki hierarki

Setiap modul keputusan pada Gambar 10-1 bekerja pada skala tertentu dalam hal waktu dan ruang (skala menurun seiring kita beralih dari pencarian ke animasi). Hal terpenting dan berguna tentang arsitektur ini adalah bahwa modul yang berbeda tidak perlu bekerja pada frekuensi yang sama. Secara umum, semakin tinggi arsitektur, semakin rendah frekuensi keputusan harus dibuat dan direvisi. Dalam video game atau XR, hanya animasi yang harus ditentukan pada 60 frame per detik karena pose karakter yang berbeda harus dirender pada setiap frame. Ada sedikit minat dalam menjalankan sistem navigasi lebih cepat dari 10 frame per detik: pengguna tidak akan melihat bahwa NPC merencanakan ulang rutenya ke tujuannya 10 kali per detik, bukan 60. Demikian pula, modul keputusan tingkat yang lebih tinggi dapat bekerja pada penurunan frekuensi saat kita mendaki dalam arsitektur.

Ini adalah fenomena yang sangat alami yang kita semua patuhi: frekuensi di mana kita merevisi keputusan kita dan membuat rencana baru berkurang dengan skala di mana kita bernalar. Kami tidak merevisi rencana karir kami setiap menit, tetapi kami merencanakan

ulang tindakan kami beberapa kali dalam satu menit untuk beradaptasi dengan situasi ketika kami hanya mengemasi tas kami untuk mulai bekerja. Ini sebenarnya kabar baik dari sudut pandang programmer AI karena melepaskan tekanan dan meningkatkan sumber daya untuk semua tugas kecuali pada level terendah. Prinsip ini dikenal dan umumnya dieksploitasi dalam komunitas robotika otonom, yang akrab dengan konsep serangkaian subproses yang bekerja secara paralel dan pada frekuensi yang berbeda. Namun, hal ini paling sering diabaikan dalam komunitas pengembangan game/XR yang sering kali dengan susah payah mencoba membuat segala macam keputusan pada frekuensi yang sama saat game dirender.

Poin penting lainnya tentang hierarki pengambilan keputusan seperti yang direpresentasikan dalam Gambar 10-1 adalah bahwa sifat masalah yang kita hadapi menghasilkan perubahan perilaku yang baik saat kita bergerak melalui arsitektur. Tugas tingkat rendah yang dekat dengan keterampilan sensorik-motorik biasanya dibingkai dalam ruang kontinu multidimensi. Perencanaan gerak dalam robotika dan pembuatan animasi dalam video game dan XR melibatkan sejumlah besar variabel kontinu dan dapat dibingkai sebagai masalah optimasi kontinu dimensi tinggi. Sebaliknya, perencanaan aktivitas tingkat tinggi sebagian besar tentang mencari ruang diskrit (tidak kontinu) yang sangat kombinatorial. Mereka biasanya melibatkan kumpulan objek diskrit, lokasi dan/atau konsep, dan kesulitan utama adalah banyaknya cara di mana ini dapat digabungkan.

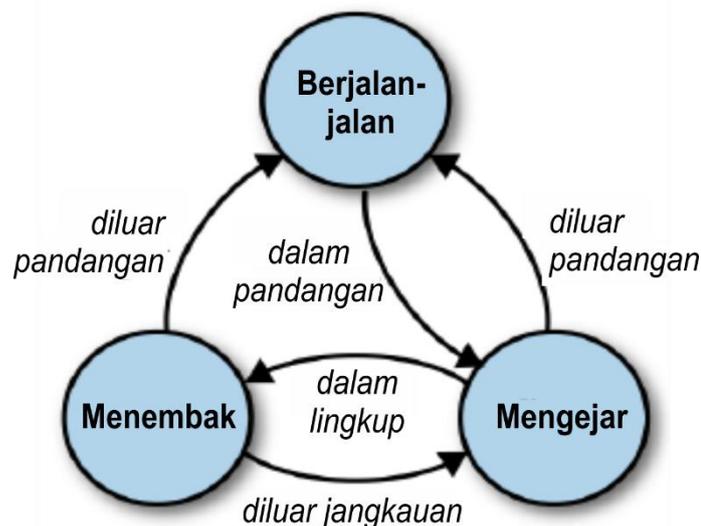
Misalnya, jika kita memutuskan untuk menjatuhkan sebuah benda dari ransel kita dan menyimpannya di lokasi untuk digunakan nanti (mungkin karena kita membutuhkan ruang di dalam tas), ada banyak kombinasi benda dan lokasi penyimpanan yang harus dipertimbangkan. Hal-hal menjadi lebih buruk jika waktu hari mempengaruhi kemampuan penyimpanan, atau jika objek yang akan ditambahkan ke dalam tas harus diperhitungkan dalam keputusan. Setiap variabel yang ditambahkan mengalikan kompleksitas dengan jumlah opsi yang tersedia untuk variabel ini, menciptakan pertumbuhan eksponensial. Fenomena ini biasa disebut ledakan kombinatorial. Pergeseran dari ruang kontinu multidimensi ke ruang diskrit yang sangat kombinatorial saat skala meningkat dapat diamati adalah banyak domain aktivitas manusia.

Sebelum menutup bagian pendahuluan ini, kami juga menekankan bahwa subjek perilaku —yaitu, entitas yang menunjukkan perilaku—dapat mengambil banyak bentuk. Kasus yang jelas adalah NPC dalam *role-playing game* (RPG). Namun, Anda juga dapat mempertimbangkan regu NPC musuh dalam game *first-person-shooter* (FPS), atau seluruh pasukan musuh dalam game *real-time strategy* (RTS), sebagai subjek perilaku. Mendorong lebih jauh, kami melihat pengisahan cerita sebagai serangkaian tindakan yang dilakukan oleh seluruh dunia game (atau, jika Anda lebih suka, modul manajemen skenario yang dapat bertindak pada skala seluruh dunia game). Memang, sebagian besar alat dan pendekatan teknis yang tersedia untuk narasi interaktif serupa dengan yang digunakan untuk NPC individu.^{20, 35, 46} Akhirnya, setiap aktor menghasilkan tindakan dan dengan demikian memiliki perilaku. Jadi, jika objek game yang mewakili objek dunia nyata yang biasanya tidak bernyawa dapat mengambil tindakan, itu berkaitan dengan isi bab ini.

10.2 PRAKTIK SAAT INI: AI REAKTIF

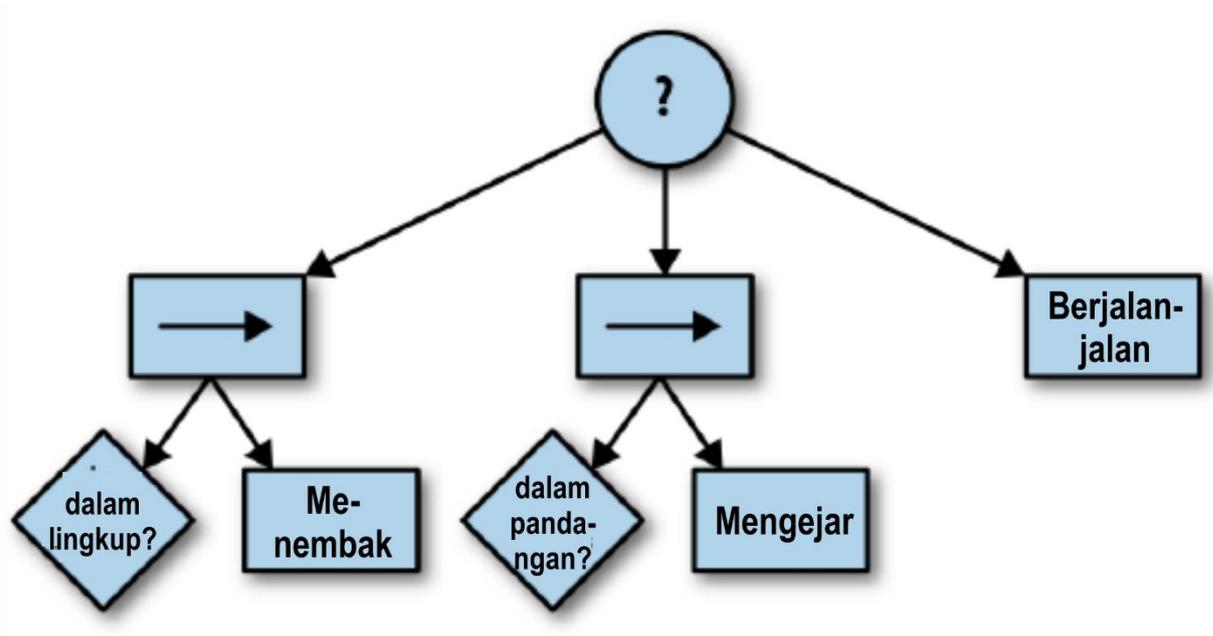
Jika Anda membuka buku tentang game AI, atau menghadiri AI Summit of the Game Developer Conference (GDC), Anda akan banyak membaca dan mendengar tentang FSM, BT, dan sistem berbasis aturan. Ketiga teknik ini, dan kombinasi yang berbeda darinya, memperkuat AI di balik sebagian besar produksi video game saat ini. Semuanya termasuk dalam paradigma AI yang sama yang disebut AI reaktif, yang telah kami singgung sebelumnya dan diskusikan lebih lanjut di bagian ini.

Anda dapat menemukan survei alat AI reaktif di akhir bab ini. Gambar 10-2 dan 10-3 menunjukkan bagaimana perilaku musuh sederhana yang disebut pengembaraan-kejar-tembak diimplementasikan menggunakan dua di antara teknik yang paling populer : FSM dan BT. Inti dari teknik ini diringkas dalam kalimat terakhir: perilaku diimplementasikan menggunakan alat ini. Artinya, terserah kepada pengembang untuk merancang setiap aspek perilaku yang akan ditunjukkan oleh AI. Alat-alat di sini untuk membantu mengatur pengetahuan dalam cara yang kurang lebih grafis dan untuk membuat aturan keputusan lebih dapat dipahami oleh manusia daripada kode biasa. Tetapi mereka tidak memiliki kemampuan pemecahan masalah yang nyata dan tidak membantu dalam keputusan desain apa pun. Dalam hal ini, mereka hanyalah bahasa pemrograman visual. Kelebihan mereka adalah menyediakan cara untuk mengimplementasikan tindakan kompleks dengan cara yang dapat dipahami.



Gambar 10-2. Mesin keadaan terbatas yang mengimplementasikan perilaku pengejaran-pengejaran-menembak

Sistem yang ditunjukkan pada Gambar 10-2 dimulai dalam status Wander, di mana ia menjelajahi lingkungan secara acak untuk mencari musuhnya; yaitu pemain. Saat pemain terlihat, AI bertransisi ke status Chase di mana ia mencoba untuk sedekat mungkin dengan pemain. Saat pemain berada dalam jangkauan, AI bergerak ke status Tembak di mana ia menyerang pemain. Jika salah satu kondisi—pemain yang terlihat dan pemain dalam jangkauan—menjadi salah, AI akan kembali ke status yang sesuai (Berjalan dan Mengejar, masing-masing).



Gambar 10-3. Pohon perilaku yang mengimplementasikan perilaku mengembara-kejar-tembak

Sistem yang ditunjukkan pada 10-3 dimulai pada simpul akar pohon (simpul putaran atas). Node ini adalah node Select. Dengan demikian, ia mengeksekusi semua anaknya secara bergantian, dari kiri ke kanan, sampai salah satu dari mereka kembali sukses. Jika tidak ada anak yang berhasil, itu mengembalikan kegagalan; jika tidak, ia mengembalikan kesuksesan (singkatnya, ini mengimplementasikan logika OR dan aturan prioritas). Node persegi panjang yang berisi panah gelap adalah node Sequence. Mereka juga mencoba mengeksekusi semua anak mereka dari kiri ke kanan, tetapi mereka berhasil hanya jika mereka dapat mencapai akhir urutan tanpa menemui kegagalan seorang anak (jadi, mereka menerapkan logika AND dan aturan urutan). Node berbentuk berlian adalah node Kondisi. Mereka memeriksa kondisi yang diberikan dan mengembalikan kesuksesan hanya jika kondisinya ternyata benar. Akhirnya, node persegi panjang abu-abu adalah tindakan primitif dari perilaku pengejaran-tembak.

Alasan mengapa industri game sangat mementingkan paradigma ini adalah karena paradigma ini memberikan kendali penuh terhadap AI. Banyak orang—termasuk kita—melihat desain game sebagai suatu bentuk seni. Seperti artis lainnya, pengembang game ingin memiliki kendali atas kreasi mereka. Karena AI reaktif memberikan kontrol total, sebagian besar diadopsi sebagai solusi default. Namun, kontrol total datang dengan beberapa kelemahan yang kami periksa di sub-bagian berikut.

10.3 ADAPTABILITAS

Poin pertama adalah, karena telah dirancang dengan tangan, perilaku reaktif sering kali terspesialisasi untuk situasi tertentu dan tidak mudah beradaptasi dengan perubahan penting. Perpanjangan video game yang menghadirkan elemen gameplay baru yang dapat

atau seharusnya memengaruhi perilaku NPC mungkin memerlukan penulisan ulang AI yang mendalam. Utilitas AI adalah teknik yang telah diperkenalkan sebagian untuk mengimbangi hal ini. Ini terdiri dari penerapan beberapa keputusan dasar berdasarkan perhitungan numerik alih-alih aturan tetap. Pertimbangkan, misalnya, node Select dari BT (Gambar 10-3). Ini mengkodekan preferensi yang ketat dan selalu dihormati di antara opsi yang tersedia (anak kiri selalu lebih disukai daripada anak kanan). Utilitas AI menggantikan urutan pilihan yang ketat ini dengan perhitungan numerik: untuk setiap alternatif yang tersedia, kami menghitung skor utilitas yang bergantung pada beberapa faktor numerik. Dalam penembak taktis, faktor-faktor ini dapat berupa, misalnya, jumlah musuh di tempat kejadian, jarak ke musuh terdekat, jumlah amunisi yang tersedia, keberadaan penutup yang dapat dijangkau, dan sebagainya.

Intinya adalah bahwa faktor-faktor ini terus menerus dan hampir tidak dapat dikendalikan dan diamati. Akibatnya, AI akan lebih mudah beradaptasi dan kurang dapat diprediksi. Perhatikan bahwa perhitungan utilitas—yaitu, aturan keputusan inti—masih diimplementasikan oleh perancang. Kami masih dalam domain AI reaktif, tetapi keputusan dapat disesuaikan dengan situasi saat ini, melalui aturan meta-keputusan yang sepenuhnya dirancang oleh aktor manusia. Dalam pendekatan ini, perhitungan utilitas adalah cara perkiraan untuk mengevaluasi peluang dari setiap tindakan yang tersedia, dan terserah kepada pengembang untuk memberikan perkiraan yang akurat. Dalam artikelnya “Simulating behavior tree: A behavior tree/planner hybrid approach,”¹³ Daniel Hilburn menjelaskan bagaimana simulasi digunakan alih-alih komputasi yang ditentukan pengguna untuk mengevaluasi alternatif yang tersedia.

10.4 KOMPLEKSITAS DAN UNIVERSALITAS

Hanya ada begitu banyak kerumitan yang dapat ditangani oleh otak manusia. Jumlah pengetahuan yang harus dimasukkan ke dalam AI untuk mencapai perilaku kompleks yang memuaskan seringkali sangat besar. Salah satu aspek tersulit dalam merancang sistem AI reaktif adalah memastikan bahwa semua situasi yang akan dihadapi sistem saat runtime dicakup oleh aturan perilaku yang tepat. Pertimbangkan, misalnya, menulis AI untuk mengendarai mobil melalui persimpangan empat arah (masalah yang dipinjam dari robotika otonom). Merancang aturan dasarnya mudah: tunggu sampai Anda memiliki hak jalan dan persimpangan jelas, lalu lanjutkan ke persimpangan. Sekarang, kita harus menutupi kasus-kasus tertentu: jika seorang pengemudi menunjukkan perilaku agresif dan mencoba untuk menipu tanda berhenti, aman dan biarkan dia lewat; jika sebuah truk pemadam kebakaran mendekat dari belakang, dengan kecepatan penuh dengan semua sirene menyala, cobalah untuk membebaskan jalan itu dengan berjalan hati-hati melalui persimpangan. Kemudian, kita dapat bertanya pada diri sendiri: apa yang harus saya lakukan jika saya memiliki cheater tanda berhenti dan truk pemadam kebakaran masuk? Menjawab pertanyaan ini mungkin perlu mempertimbangkan faktor-faktor lain; yaitu, mendefinisikan kasus-kasus yang lebih khusus yang memerlukan aturan-aturan tertentu. Ini adalah contoh ledakan kombinatorial yang menghancurkan AI reaktif dalam aplikasi dunia nyata.

10.5 KELAYAKAN

Ini adalah batas terkuat dari pendekatan AI reaktif. Merancang semua aspek perilaku dengan tangan memerlukan mengetahui solusi untuk semua masalah yang perlu dipecahkan oleh AI. Dalam beberapa kasus, mendapatkan solusi optimal untuk suatu masalah (atau solusi yang cukup baik) tidak dapat dilakukan hanya dengan memberikan aturan perilaku; itu membutuhkan sejumlah pemecahan masalah. Pertimbangkan, misalnya, navigasi; yaitu, tugas menemukan jalur terpendek dari lokasi asal ke tujuan. Hal ini biasanya diselesaikan dengan menggunakan algoritma jalur terpendek seperti A*, yang merupakan contoh dari sistem perencanaan yang dibahas di bagian berikutnya. Jalur terpendek diselesaikan dengan menjelajahi kemungkinan masa depan, memprediksi hasil dari urutan tindakan yang berbeda untuk dapat memilih yang terbaik. Alasan ini tergantung pada keadaan lingkungan saat ini dan tujuan AI, dan ini melibatkan cukup banyak pemecahan masalah. Hanya saja tidak mungkin untuk memecahkan masalah jalur terpendek secara umum dengan menggunakan seperangkat aturan yang telah ditentukan sebelumnya. Dalam praktiknya, pengembang game menggunakan alat AI reaktif untuk merancang sebagian besar perilaku mereka, tetapi mereka mendelegasikan tugas navigasi ke modul khusus yang menggunakan paradigma berbeda. Argumen ini berlaku untuk aspek lain dari penalaran di luar navigasi. Contoh lain disediakan di awal bagian berikutnya. Dalam semua kasus ini, AI reaktif bukanlah solusi yang layak.

Batasan AI reaktif yang diketahui ini menjadi kritis saat kita beralih dari video game ke XR. Untuk alasan yang berbeda, baik VR dan AR menempatkan paradigma ini pada tantangan lebih dari game. Dalam AR, kesulitan datang dari ketidakpastian lingkungan. Scene AR dibangun berdasarkan Scene dunia nyata, dengan menambahkan elemen virtual di dalamnya; oleh karena itu, tidak dapat dikendalikan dan tidak dapat diprediksi. Ada sedikit keraguan bahwa ketika aplikasi AR menjadi umum, mereka akan diuji di lingkungan paling orisinal oleh pengguna. Ini sangat kontras dengan Scene video-game yang seluruhnya dirancang dengan tangan dan dengan demikian sepenuhnya dapat dikontrol dan diprediksi (jika kita mengecualikan kasus terbatas dari level game yang dihasilkan secara prosedural).

Jelas, lebih mudah untuk merancang AI dengan tangan yang disesuaikan dengan pemandangan yang kita kendalikan sepenuhnya daripada sebagian. Dengan kata lain, AR lebih menantang batas AI reaktif dalam hal kemampuan beradaptasi, kompleksitas, dan universalitas daripada video game. Dalam kasus VR, argumennya berbeda. Seperti Scene video game, Scene VR sepenuhnya dikendalikan oleh desainer. Masalahnya di sini ada pada harapan pengguna.

Karena pengalaman sensorik dan imersi meningkat pesat saat kami beralih dari game ke VR, sebagian besar pengguna mengharapkan setiap aspek pengalaman meningkat dengan cara yang sama. “Bug AI”—perilaku AI yang tidak masuk akal bagi pengguna—dapat diterima, terkadang lucu, dan sering kali dapat dieksploitasi dalam video game. Namun, pengembang game tampaknya takut untuk mengulangi kesalahan ini di dunia VR. Hal ini tercermin dari rendahnya jumlah NPC aktual yang ditemui dalam game VR (selain NPC musuh yang hanya menjadi target bergerak untuk ditembakkan oleh pemain).

Sebelum memeriksa alternatif yang ada untuk AI reaktif, kami menyatakan bahwa, sejauh ini, bagian ini hanya membahas aspek perilaku tingkat tertinggi dan skala terbesar (lihat kembali bagian "Perilaku"). Kami telah menekankan bahwa tugas navigasi biasanya diselesaikan menggunakan algoritme jalur terpendek khusus yang tidak termasuk dalam paradigma AI reaktif. Bagaimana dengan level terendah dalam hierarki Gambar 10-1; yaitu, generasi animasi? Faktanya, situasinya sangat mirip dengan perilaku skala besar. Praktek saat ini adalah dengan menggunakan klip animasi beberapa detik dan mewakili gaya berjalan tunggal. Klip diatur ke dalam controller animasi besar, yang merupakan FSM dengan satu klip terpasang ke setiap status.²² Transisi antar klip mewakili perubahan gaya berjalan dan dipicu oleh input pemain dalam kasus karakter yang dapat dimainkan (PC) atau oleh perilaku tingkat yang lebih tinggi modul dalam kasus NPC. Transisi dari satu animasi ke animasi lainnya dikelola melalui perpaduan animasi, yang melibatkan beberapa parameter numerik. Praktik yang paling umum adalah mengatur dan menyetel parameter ini dengan tangan, yang merupakan tugas yang menakutkan.

Ini memang termasuk dalam paradigma AI reaktif: sistem animasi tidak mengantisipasi lebih dari satu frame di masa depan, dan semua aturan keputusan dirancang oleh pengembang manusia. Baru-baru ini, pendekatan berbasis tujuan telah diusulkan di mana AI berhak untuk memutuskan pose karakter berikutnya berdasarkan pose saat ini dan tujuannya. Pendekatan berbasis tujuan serupa dari generasi perilaku umum dibahas di bagian berikutnya.

Lebih Banyak Kecerdasan dalam Sistem: AI Deliberatif

Pertukaran antara kemampuan kontrol dan otonomi sangat penting dalam diskusi seputar AI game: AI yang menampilkan beberapa bentuk kekuatan keputusan (jelas) kurang dapat dikontrol daripada AI yang sepenuhnya dikodekan dengan tangan. Kami melihat di bagian sebelumnya bahwa praktik yang paling umum adalah mengorbankan otonomi untuk pengendalian, menggunakan alat AI reaktif. Kami juga melihat bahwa itu tidak dapat diterapkan pada semua masalah yang dapat dihadapi oleh agen buatan. Beberapa keputusan memerlukan prediksi dan antisipasi efek dari urutan tindakan yang berbeda. Ini adalah kasus navigasi yang tidak dapat dilakukan tanpa algoritma jalur terpendek; dengan kata lain, pemecah masalah. Contoh lain dari masalah "sulit" adalah sebagai berikut:

Pengelolaan sumber daya

Apalagi jika dicampur dengan masalah navigasi. Misalnya, agen harus menavigasi ke tujuan, tetapi gerakan menghabiskan sumber daya yang datang dalam jumlah terbatas. Sumber daya dapat diisi ulang di berbagai lokasi lingkungan. Jadi, agen perlu mengintegrasikan beberapa pemberhentian untuk mengisi kembali sumber daya dalam perjalanannya ke tujuan. Menemukan jalur terpendek ke tujuan termasuk membuat jalan memutar terpendek untuk mengisi kembali sumber daya. Alasan ini harus mengintegrasikan struktur lingkungan, termasuk lokasi stasiun isi ulang sumber daya dan tujuannya. Sangat sulit—jika mungkin, sama sekali—untuk merancang aturan keputusan umum dengan tangan untuk masalah ini.

Intelligent exploration

Mengintai karakter musuh membutuhkan penalaran tentang keadaan pengetahuan—apa yang diketahui dan tidak diketahui pada saat ini—dan merencanakan bagaimana gerakan

eksplorasi akan mengubah keadaan ini. Misalnya, AI akan memutuskan untuk membuat jalan memutar melalui bukit terdekat karena pemandangan dari atas bukit memberikan informasi tentang lokasi unit musuh saat ini. Sekali lagi, jenis penalaran ini harus mengintegrasikan informasi tentang struktur lingkungan, termasuk lokasi mana yang dapat diamati dari setiap lokasi. Ini diselesaikan dengan buruk oleh aturan keputusan tetap

Perencanaan taktis

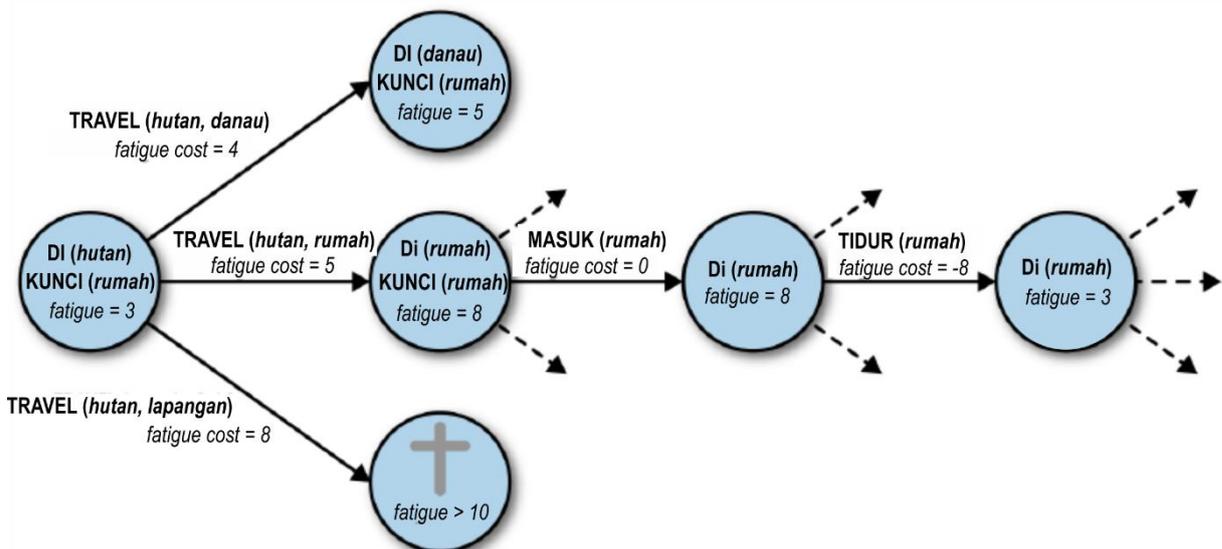
Contohnya termasuk mengelola regu NPC sehingga mereka mencoba menjebak pemain, memblokir semua jalur keluar mereka dari Scene permainan. Lagi-lagi ini terlalu bergantung pada konfigurasi masalah untuk menggunakan aturan yang telah ditentukan sebelumnya.

Tugas-tugas ini (dan lainnya) memerlukan beberapa bentuk kemampuan pencarian dan pemecahan masalah di AI. Mereka ditangani oleh alat perencanaan otomatis^{9, 10} yang menerapkan paradigma AI deliberatif yang dibahas di bagian ini. Berikut adalah dua poin kunci dari paradigma AI deliberatif:

- Kami fokus pada menghasilkan urutan tindakan daripada keputusan tunggal. Hal ini sesuai dengan jenis masalah keputusan di atas. Masalah-masalah ini dicirikan oleh fakta bahwa suatu tindakan hanya menarik sehubungan dengan tindakan yang akan mengikutinya. Untuk memahami dengan jelas perbedaan dengan AI reaktif, pertimbangkan masalah navigasi ke tujuan. Jika kita selalu bergerak lurus ke arah gawang, kita bisa terjebak dalam rintangan berbentuk sudut. Dalam beberapa situasi, kita perlu menjauh dari tujuan untuk melewati rintangan. Algoritme jalur terpendek seperti A*²⁷ memahami hal ini dan dapat bergerak menjauh dari gawang dan melewati rintangan. Langkah ini menarik hanya sebagai langkah pertama dari urutan yang dapat membawa kita ke tujuan. Diambil secara terpisah, itu tidak mencapai tujuan apa pun. Output sebenarnya dari algoritme adalah jalur lengkap menuju tujuan; yaitu, rencana, di mana keputusan yang dieksekusi hanyalah langkah pertama.
- Keputusan dihasilkan secara otomatis, dengan memecahkan masalah yang terdefinisi dengan baik daripada hardcoded oleh pengembang. Di dunia lain, AI memiliki kekuatan pengambilan keputusan yang nyata, didukung oleh kemampuan pemecahan masalah. Dalam hal ini, ini adalah pelengkap sempurna untuk AI reaktif karena membantu memutuskan apa yang harus dilakukan dalam situasi tertentu. Ini juga membantu menghindari bagian dari kerumitan yang muncul dari kebutuhan universalitas yang dibahas sebelumnya. Perencana otomatis menggunakan model masalah pengambilan keputusan yang mereka hadapi. Untuk menguraikan, menggunakan lagi contoh navigasi, algoritma jalur terpendek dapat memodelkan masalah navigasi sebagai peta jalan; yaitu, jaringan diskrit (atau grafik) di mana node disebut titik jalan, konektor antara node mewakili gerakan yang mungkin, dan ada biaya yang terkait dengan setiap konektor. (Ini adalah contoh; beberapa sistem navigasi menggunakan model masalah yang berbeda.) Selama masalah navigasi dapat dimodelkan dengan cara ini, perencana dapat menyelesaikannya. Dengan demikian, perencana bersifat universal dalam lingkup (terbatas) dari model domain mereka.

Seperti yang kami katakan, contoh paling umum dari perencana adalah sistem navigasi yang menghitung jalur terpendek antara dua lokasi di dunia game. Buku AI game pengantar apa pun memberikan survei tentang algoritme ini; lihat misalnya [3, 30]. Prinsip dasarnya disebut pencarian. Mengingat lokasi awal, diwakili misalnya sebagai titik arah dalam peta jalan, mereka memperluas beberapa kemungkinan lintasan masa depan menuju tujuan dan memilih yang terbaik. Trik algoritmik memungkinkan Anda melakukan pencarian ini secara efisien dan menghindari mempertimbangkan urutan tindakan yang terbukti tidak optimal. Tetapi prinsip dasarnya masih sangat sederhana: kami (secara implisit) menghitung rencana yang mungkin dan memilih yang terbaik.

Pencarian tidak terbatas pada masalah jalur terpendek. Gambar 10-4 menunjukkan contoh bagaimana Anda dapat menggunakannya untuk memutuskan perilaku umum. Identy adalah untuk memperluas gagasan lokasi dengan mempertimbangkan keadaan perencanaan umum. Keadaan perencanaan merangkum semua informasi yang relevan dengan masalah keputusan pada waktu tertentu. Dalam masalah navigasi, ini hanya terdiri dari lokasi agen (koordinat x dan y) karena ini adalah satu-satunya informasi yang relevan dengan masalah keputusan untuk menemukan jalur terpendek ke tujuan tertentu dan tetap. Jika masalahnya juga melibatkan beberapa bentuk kelelahan, sehingga agen menjadi lelah di sepanjang jalan dan harus berhenti di tempat yang ditentukan untuk beristirahat, kelelahan agen saat ini juga termasuk dalam keadaan perencanaan. Demikian pula, gerakan antara titik arah digeneralisasikan oleh tindakan perencanaan.



Gambar 10-4. Ruang pencarian dari algoritma perencanaan otomatis (sebagian)

Gambar 10-4 menunjukkan langkah pertama untuk memperluas lintasan masa depan dari keadaan awal yang diberikan. AI dimulai di negara bagian paling kiri, di mana agen berada di hutan, tingkat kelelahannya diatur ke 3, dan ia memiliki kunci rumah. Dari sana, beberapa tindakan Perjalanan dimungkinkan, masing-masing melibatkan biaya kelelahan yang berbeda. Tindakan yang akan menyebabkan agen mati karena kelelahan, seperti bepergian ke ladang, menyebabkan keadaan "mati" terminal. Tindakan Perjalanan lainnya mengubah lokasi dan

Kekuatan Augmented dan Virtual Reality dalam Bisnis (Dr. Agus Wibowo)

kelelahan agen. Jika agen memutuskan untuk pergi ke rumah, ia dapat menggunakan kuncinya untuk masuk ke rumah, membuatnya di rumah, bukan di rumah. Karena kita berada dalam video game, kunci dikonsumsi saat digunakan. Jadi, memasuki rumah juga memiliki efek menghilangkan kunci dari kepemilikan agen. Setelah berada di rumah, agen diizinkan untuk tidur, yang mengatur ulang kelelahannya ke 0.

Dengan cara yang sama seperti gerakan mengubah lokasi agen, tindakan mengubah statusnya. Namun, mereka secara semantik dapat mewakili aktivitas yang sangat berbeda dari penggerak. Dalam contoh kami, tindakan Tidur memiliki efek mengatur ulang kelelahan agen ke nol sambil membiarkan lokasinya tidak berubah. Tindakan lain seperti Perjalanan akan mengubah lokasi dan kelelahan agen. Secara formal, tindakan perencanaan memiliki kondisi yang harus benar dalam keadaan saat ini sebelum dapat diterapkan. Mereka juga memiliki efek, yang merupakan daftar perubahan yang mereka bawa ke negara bagian. Dalam contoh kami, tindakan Tidur hanya dapat dieksekusi ketika agen berada di rumah (Di (rumah)), dan memiliki efek mengatur kelelahan agen kembali ke 0 sambil membiarkan aspek lain dari keadaan tidak berubah. Tindakan perjalanan dari lokasi A ke lokasi B memiliki kondisi bahwa kita harus mulai di lokasi A, dan efeknya tidak lagi berada di lokasi A tetapi berada di B, sebagai gantinya. Perencanaan otomatis bekerja dengan mencari grafik yang mewakili masalah perencanaan, mirip dengan Gambar 10-4, untuk jalur terpendek/termurah antara keadaan saat ini dan keadaan tujuan yang ditentukan oleh pengembang.

Perencanaan otomatis adalah pendekatan berbasis tujuan yang sistematis untuk menghasilkan perilaku. Lihat bagian Referensi untuk survei bidang ini.9, 10 Ada banyak variasi skema dasar yang baru saja kami jelaskan, termasuk yang berikut:

Perencanaan sementara

Hal ini memungkinkan penalaran yang baik tentang durasi tindakan, dan eksekusi simultan dari beberapa tindakan yang tidak bertentangan.

Perencanaan di bawah ketidakpastian

Ini model dan alasan tentang ketidakpastian dalam efek tindakan. Alih-alih satu set efek, tindakan memiliki beberapa set efek dengan probabilitas berbeda yang menyertainya. Misalnya, menyerang musuh mungkin memiliki banyak efek (berhasil atau gagal) dengan probabilitas tertentu, mengintai lokasi dari titik pengamatan yang jauh dapat mengarah pada menemukan unit musuh atau tidak. Kemungkinan untuk memodelkan ketidakpastian yang melekat pada hasil tindakan sangat penting dalam beberapa domain. Dalam contoh penembak taktis, AI yang tidak dapat menangani ketidakpastian akan berasumsi bahwa setiap serangan selalu berhasil atau selalu gagal. Dalam kedua kasus, itu akan mengarah pada perilaku buruk (menjadi terlalu percaya diri dalam kasus pertama, dan menjadi terlalu konservatif dalam kasus kedua). Perilaku yang tepat diperoleh hanya dengan mempertimbangkan kemungkinan hasil yang berbeda dari tindakan, dan menyeimbangkan risiko dengan manfaat dari pilihan yang berbeda.

Ini menghasilkan rencana di mana tindakan tidak sepenuhnya teratur secara berurutan. Misalnya, cabang teknik menghasilkan rencana dalam bentuk rangkaian tindakan (sebagai lawan dari rangkaian tindakan). Rencana ini dieksekusi dengan cara berikut: pertama semua

tindakan di set pertama dieksekusi dalam urutan apa pun yang nyaman, maka semua tindakan di set kedua dieksekusi dalam urutan apa pun, dan seterusnya. Perencana menjamin bahwa tindakan dalam set yang sama dapat dieksekusi dalam urutan apa pun tanpa memengaruhi hasilnya.

Ini mencoba untuk memecahkan dalam satu alat beberapa lapisan pengambilan keputusan (lihat Gambar 10-1). Alat yang paling sering digunakan dalam pendekatan ini disebut jaringan tugas hierarkis (HTN). Hal ini memungkinkan dekomposisi hierarkis perilaku seperti yang diamati dalam BT, sambil mengotomatiskan semua pengambilan keputusan inti menggunakan perencanaan.

Game pertama yang menggunakan perencanaan otomatis untuk level tertinggi dari hierarki perilaku adalah F.E.A.R. Perencana itu disebut Perencanaan Tindakan Berorientasi Sasaran (GOAP). Digunakan untuk AI musuh, dan itu membuat kesan yang kuat pada komunitas gamer. Meskipun demikian, seluruh pendekatan AI deliberatif memiliki penetrasi yang sangat dangkal di industri game. Tidak mengherankan, sebagian besar telah diterapkan pada permainan taktis dan strategis yang mengandung masalah keputusan dan optimasi yang sulit yang solusinya memerlukan beberapa bentuk pertimbangan. HTN mendapat perhatian dari komunitas, terutama karena kesamaannya dengan BT yang sangat populer. Terlepas dari upaya ini, paradigma AI deliberatif masih mewakili minoritas kecil dari semua AI game komersial. Ada beberapa alasan untuk menjelaskan hal ini:

Keterkendalian

Kami sudah banyak membahas hal ini. AI Reaktif memberikan kontrol total, yang sangat bagus untuk membuat game dan XR (walaupun menyakitkan). Perencana kurang mudah dikendalikan, tetapi mereka dapat memecahkan masalah sulit di luar jangkauan aturan yang ditulis tangan. Daripada harus memilih antara satu pendekatan atau yang lain, kami sangat menyarankan menggunakan campuran dari mereka. Kami mendorong pengembang untuk merancang arsitektur modular yang bersih dan menggunakan teknik dan pendekatan yang berbeda dalam modul yang berbeda.

Selama tidak ada masalah keputusan yang sulit untuk dipecahkan, AI reaktif menyediakan cara yang bagus untuk menghasilkan perilaku yang kita inginkan. Anda harus menggunakan AI deliberatif (hanya?) untuk masalah yang sulit dipecahkan. Misalnya, perencana di mana setiap tindakan diimplementasikan oleh BT tertentu adalah arsitektur yang dapat masuk akal dalam banyak situasi. Demikian pula, BT dapat berisi simpul perencanaan yang menangani submasalah yang terbatas tetapi sulit diputuskan.

Kompleksitas run-time

Salah satu masalah utama dengan teknik pemecahan masalah seperti pencarian ruang-negara yang dijelaskan sebelumnya adalah waktu eksekusi mereka. Kompleksitas meningkat dengan ukuran masalah (ukuran peta dalam masalah navigasi, jumlah NPC untuk mengontrol dalam penembak taktis, dll). Tergantung pada posisinya dalam arsitektur Gambar 10-1, perencana memiliki kendala yang berbeda pada frekuensi di mana ia harus menghasilkan keputusan, dan selalu ada masalah yang cukup besar untuk memecahkan batas ini. Dengan kata lain, beberapa masalah terlalu besar untuk diselesaikan pada frekuensi yang diinginkan.

Kami mencatat bahwa ketika mereka terlalu lambat untuk digunakan secara real time, Anda dapat menggunakan perencana secara offline—pada saat mengembangkan game atau aplikasi XR—untuk menghasilkan terlebih dahulu sebuah rencana yang nantinya dapat diubah menjadi aturan keputusan reaktif untuk menjadi digunakan saat runtime. Kami membahas di bagian berikutnya bagaimana Anda dapat menggunakan pembelajaran mesin untuk lebih meningkatkan kemampuan pemecahan masalah perencana saat digunakan secara offline.

Kesulitan untuk mengimplementasikan

Perencanaan otomatis menjanjikan perilaku AI yang hebat, tetapi ada biayanya. Algoritme perencanaan memiliki kompleksitas tertentu dan memerlukan beberapa sumber daya untuk dikembangkan, menempatkannya di luar jangkauan tim kecil tanpa programmer AI yang ahli. Di dunia akademik, bahasa deskripsi domain perencanaan telah dikembangkan untuk memungkinkan penggunaan kembali perencana/pemecah yang sama di domain yang berbeda. Idenya adalah untuk mendefinisikan bahasa di mana kita dapat mengekspresikan (merencanakan) masalah dari sifat yang berbeda dan kemudian membuat pemecah masalah yang dapat menangani masalah apa pun yang diungkapkan dalam bahasa itu (dengan cara yang sama seperti algoritma jalur terpendek dapat menangani masalah apa pun yang diwakili sebagai peta jalan). Pendekatan ini banyak digunakan oleh komunitas robotika otonom, terutama oleh NASA, yang mengontrol beberapa perangkat (semi-)otonom dengan sifat dan ukuran yang sangat berbeda menggunakan perencana yang sama. Kami percaya ini adalah kunci untuk memproduksi alat AI deliberatif umum yang dapat digunakan kembali untuk video game dan XR, dan itu layak mendapat perhatian lebih dari komunitas ini.

Kesulitan untuk mengadopsi

Beralih dari AI reaktif ke deliberatif adalah perubahan radikal dalam alur kerja perancang AI: alih-alih memperbaiki perilaku, perancang harus merancang masalah yang solusinya menghasilkan perilaku yang baik. Karena mereka tidak dapat secara langsung mengedit perilaku tetapi harus melalui tugas pemodelan domain, banyak orang akan merasa bahwa lapisan kerumitan tambahan telah ditambahkan dalam perjalanan. Namun, ada argumen teoretis yang menunjukkan fakta bahwa—setidaknya untuk masalah yang kompleks—AI deklaratif sebenarnya lebih mudah digunakan daripada AI reaktif.

Singkatnya, argumen ini berjalan sebagai berikut. Dunia memiliki struktur, setidaknya untuk mata manusia. Ketika kita diminta untuk menggambarkan suatu masalah, kita biasanya dapat melakukannya dengan cara yang relatif kompak dan terstruktur. Khususnya, kita dapat menilai beberapa hipotesis independensi antara variabel yang berbeda. Misalnya, pertimbangkan masalah yang dihadapi oleh teknisi perbaikan mesin fotokopi yang perlu merancang jadwal untuk satu hari. Dia perlu memutuskan pelanggan mana yang akan dia kunjungi dan operasi apa yang akan dia lakukan pada mesin fotokopi pelanggannya. Ketika kita menyatakan masalah itu, kita dapat mengasumsikan banyak hubungan independensi antara variabel dan tindakan. Jelas, setiap perbaikan atau operasi diagnostik yang dilakukan teknisi servis pada mesin fotokopi tidak memengaruhi mesin lain milik pelanggan yang sama atau pelanggan lain. Sebenarnya, kita dapat dengan mudah membagi masalah menjadi beberapa submasalah, satu untuk setiap mesin yang membutuhkan servis. Ada banyak

variabel status yang terkait dengan satu mesin (misalnya, status komponen yang berbeda saat ini, hasil pengujian yang telah dilakukan, riwayat perbaikan pada mesin ini), tetapi variabel ini independen dari satu mesin ke mesin lainnya.

Masalahnya secara alami dimodelkan sebagai arsitektur dua lapis.²⁶ Pada tingkat tertinggi, kita memiliki masalah umum dalam memutuskan pelanggan mana yang akan kita kunjungi dan dalam urutan apa (masalah yang mengandung komponen kuat dari perencanaan jalur terpendek). Tingkat terendah dibentuk oleh beberapa subproses, satu untuk setiap mesin fotokopi yang perlu diperbaiki. Pada tingkat yang lebih rendah, submasalah lokal memiliki banyak variabel dan tindakan yang dapat dianggap sebagai pribadi untuk subproses ini: mengeksekusi tindakan pribadi hanya memengaruhi variabel pribadi dari proses ini.

Subproses terikat bersama pada tingkat tinggi oleh satu set kecil variabel bersama yang mewakili sumber daya bersama: total waktu yang dapat dihabiskan teknisi untuk mengerjakan mesin fotokopi pada hari ini, set suku cadang (terbatas) yang tersedia untuknya, dan lokasi agen (yang dapat dilihat sebagai sumber daya bersama tertentu). Sebagai konsekuensinya, selama kita hanya membahas deskripsi masalah, kita dapat dengan mudah menambah, menghapus, atau memodifikasi mesin fotokopi/submasalah. Modifikasi satu subproses tidak mempengaruhi proses lainnya. Mesin fotokopi baru dapat dipasang ke arsitektur umum seperti potongan Lego, variabel bersama memainkan peran kancing yang menjaga konstruksi tetap bersama.

Masalahnya memiliki struktur alami yang membuat deskripsinya kompak dan modifikasinya mudah dan bertahap. Sayangnya, struktur ini menghilang ketika kita berpindah dari deskripsi masalah ke ruang solusi. Jika ada struktur seperti itu pada tingkat solusi—yaitu, perilaku optimal—kita dapat berharap bahwa penambahan atau penghapusan mesin fotokopi tidak memengaruhi rencana kita saat kita mengerjakan mesin fotokopi lain. Tapi ini tidak benar.

Untuk melihat ini, bayangkan bahwa kami sedang memecahkan masalah pelanggan tertentu dan kami berencana untuk mengembalikan mesin pelanggan sepenuhnya sebelum pindah ke pelanggan berikutnya. Ini adalah kebijakan lokal kami saat ini untuk pelanggan ini: selesaikan semua pekerjaan di sini sebelum pergi. Sekarang, kami tiba-tiba menambahkan mesin fotokopi baru yang bersaing dengan tugas kami saat ini untuk beberapa suku cadang tertentu. Jika kami menambahkan bahwa mesin fotokopi baru ini milik pelanggan prioritas utama kami dan bahwa kami berkomitmen untuk menyelesaikan masalah pelanggan itu terlebih dahulu, kami mungkin memutuskan untuk menghentikan pekerjaan kami saat ini, menyimpan suku cadang untuk pelanggan yang paling penting, dan pindah ke prioritas teratas pelanggan segera. Penambahan proses baru telah mengubah kebijakan lokal dari proses lainnya, yang bertentangan dengan prinsip inkrementalitas. Contoh ini menunjukkan bahwa, bahkan ketika domain masalah menunjukkan struktur yang nyaman yang membantu menggambarkannya (dan sebagian besar masalah nyata melakukannya), perilaku optimal untuk menyelesaikan masalah ini tidak dapat memiliki struktur sama sekali. Oleh karena itu, pada prinsipnya, ada tingkat kerumitan tertentu di mana kita lebih baik bekerja di ruang

masalah daripada bekerja di ruang perilaku. Seperti yang kita lihat, beberapa dari batasan ini memiliki kemungkinan solusi yang layak untuk diteliti dalam domain video game dan XR.

Seperti yang kami katakan, setiap perencana memiliki batasannya dalam hal kompleksitas masalah yang dapat ditanganinya sesuai anggarannya atau bahkan dalam waktu yang wajar. Dalam penelitian akademis, kekuatan AI deliberatif telah didorong hingga batasnya saat ini dengan menggabungkannya dengan ide-ide dari pembelajaran mesin. Oleh karena itu, bagian utama terakhir dari bab ini membahas paradigma ketiga dan terakhir ini.

10.6 MACHINE LEARNING

Sejauh ini kita telah membahas perilaku dalam konteks algoritme dan metode yang menghasilkan kebijakan tetap untuk tindakan. Pendekatan-pendekatan ini bergantung pada authoring perilaku manusia, baik secara langsung dengan memberikan aturan eksplisit untuk perilaku (di bagian sebelumnya), atau secara tidak langsung dengan menyediakan model dinamika simulasi dan mekanisme perencanaan menggunakan model ini. Berbeda dengan metode tersebut, sekarang kita beralih ke pendekatan berdasarkan perilaku belajar dari data. Belajar dari data bisa menarik karena sangat mengurangi jumlah pengetahuan yang dibutuhkan untuk membangun AI. Kami mungkin dapat membangun AI untuk masalah yang tidak kami pahami dengan baik atau yang tidak kami kuasai dengan baik. Ia melakukannya dengan menukar waktu komputasi dan kumpulan data yang besar. Ini juga menarik dalam situasi di mana kita ingin perilaku yang dipelajari digeneralisasikan ke keadaan yang tidak terlihat. Pendekatan ini berada di bawah payung luas pembelajaran mesin, yang hadir dalam tiga rasa berbeda:

Pembelajaran yang diawasi

Ini berkaitan dengan mempelajari pemetaan dari himpunan X ke himpunan Y dengan contoh. Dalam kasus aplikasi pembelajaran mesin umum, kami ingin memetakan gambar ke label yang menjelaskan isinya; misalnya, (foto kucing \rightarrow "kucing"). Untuk tujuan ini, kami membuat model pembelajaran yang menerima gambar sebagai label input dan output seperti "kucing", "anjing", "burung", dan sebagainya. Selanjutnya, kami mendefinisikan fungsi kerugian yang memberikan nilai numerik untuk perbedaan antara hasil yang diinginkan dan output yang diamati dari model. Singkatnya, kami mengukur seberapa baik kami melakukannya dengan nilai numerik. Dengan menggunakan kumpulan contoh data yang besar dalam bentuk pasangan (gambar, label), kami sekarang dapat meningkatkan model untuk mengurangi kerugian di masa mendatang, sehingga meningkatkan kemungkinan pemetaan yang benar

Pembelajaran tanpa pengawasan

Pendekatan ini adalah tentang belajar struktur dalam data tanpa pemetaan yang jelas dari satu set ke yang lain sebagai tujuan. Ini berguna, misalnya, untuk memahami atau mengompres data. .

Reinforcement Learning

Ini tentang mempelajari perilaku dengan berinteraksi dengan proses dunia nyata atau simulasi. Agen dipandu ke perilaku yang diinginkan melalui penggunaan fungsi penghargaan. Tidak seperti fungsi kerugian, fungsi hadiah tidak secara langsung menggambarkan seberapa baik

kinerja model, melainkan disediakan ketika agen memasuki keadaan tertentu, dan sesuai dengan keinginan dari keadaan tersebut. Dengan kata lain, tujuannya bukan untuk mempelajari fungsi penghargaan (yang akan menjadi kasus untuk pembelajaran yang diawasi), tetapi untuk menemukan perilaku yang paling sering mengarah ke keadaan yang paling diinginkan. Ini sangat terkait dengan otomatis di mana kami prihatin dengan menghasilkan urutan tindakan alih-alih keputusan satu-shot. Memang, model yang mendukung sebagian besar algoritma Reinforcement Learning adalah model perencanaan yang parameternya awalnya tidak diketahui dan harus dipelajari dengan berinteraksi dengan simulasi.

10.7 REINFORCEMENT LEARNING

Relevansi *Reinforcement Learning* untuk masalah yang dibahas dalam bab ini sangat mudah. Tidak mengherankan, hubungan antara disiplin ini dan video game telah berkembang menjadi sangat kuat. Komunitas riset telah mengakui relevansi video game sebagai test bed untuk algoritma *Reinforcement Learning*. Video game arcade sederhana sekarang menjadi bangku tes favorit komunitas untuk mengevaluasi kinerja algoritma *Reinforcement Learning* baru.

Ada dua komponen utama untuk aplikasi pembelajaran mesin: data dan algoritma. Kami dapat mempertimbangkan keduanya dalam konteks spesifik dari perilaku yang ditulis dalam lingkungan simulasi (video game dan XR). *Reinforcement Learning* mendapatkan datanya dari interaksi dengan simulasi itu sendiri. Jenis aplikasi lain belajar dari demonstrasi manusia; yaitu, (ahli) data pengguna. Pendekatan ini disebut Imitation Learning dan dibahas kemudian dalam bab ini.

Reinforcement Learning adalah pendekatan dalam pembelajaran mesin untuk sampai pada perilaku yang diinginkan, yang kami sebut sebagai kebijakan. Pemetaan yang kami minati dalam pembelajaran mesin adalah antara status s dan tindakan a . Dalam beberapa kasus, pemetaan ini probabilistik dan berbentuk $p(a | s)$. Dalam banyak keadaan, agen mungkin tidak memiliki akses ke definisi lengkap dari keadaan simulasi. Dalam kasus seperti itu, kami mengatakan bahwa agen hanya memiliki akses ke pengamatan, yang terbatas dan diturunkan dari keadaan sebenarnya.

Contoh sederhananya adalah dengan mempertimbangkan simulasi kota besar. Dalam simulasi ini, keadaan terdiri dari seluruh posisi dan lintasan semua mobil di jalan virtual. Kita dapat membayangkan bahwa seorang agen dalam salah satu mobil virtual ini mungkin memiliki akses ke pandangan orang pertama dari kendaraan mobil lain di depannya. Kumpulan informasi yang terbatas ini kemudian sesuai dengan pengamatan. Masalah Reinforcement Learning kemudian mempelajari pemetaan ($o \rightarrow a$), atau probabilitas $p(a | o)$, yang memaksimalkan fungsi hadiah dari waktu ke waktu.

Dibandingkan dengan pendekatan berbasis rencana yang dibahas di bagian sebelumnya, Reinforcement Learning dapat terjadi tanpa adanya model dinamika simulasi yang maju. Metode ini disebut sebagai bebas model. Meskipun mereka membutuhkan lebih banyak interaksi secara signifikan dengan lingkungan, mereka sangat umum karena mereka tidak membuat asumsi atau persyaratan tentang spesifik lingkungan.

Ada dua kategori besar metode yang dirancang untuk memecahkan masalah Reinforcement Learning. Ini adalah metode berbasis nilai dan berbasis kebijakan. Dalam metode berbasis nilai, agen mencoba mempelajari perkiraan nilai setiap keadaan $V(s)$, atau nilai setiap pasangan keadaan-aksi $Q(s, a)$. Nilai ini mewakili jumlah diskon dari imbalan masa depan yang diharapkan, seperti yang ditunjukkan dalam Persamaan 10-1. Persamaan 10-1.

Equation 10-1.

$$V(s) = E\left[\sum_{t=0}^{\infty} \gamma^t r(t) \mid s(0) = s\right],$$

$$Q(s, a) = E\left[\sum_{t=0}^{\infty} \gamma^t r(t) \mid s(0) = s \wedge a(0) = a\right],$$

Dimana $s(t)$ adalah keadaan sistem pada waktu t , $a(t)$ adalah tindakan yang dilakukan pada waktu t , $r(t)$ adalah imbalan yang diterima oleh agen pada waktu t , dan $\gamma \in [0,1)$ adalah faktor diskon. Jumlah hadiah yang didiskon biasanya digunakan untuk membatasi agen pada kebijakan selama rangkaian waktu yang terbatas. Hal ini juga nyaman untuk memungkinkan trade-off antara keuntungan jangka pendek (jangka diskon yang lebih kecil) dan keuntungan jangka panjang (jangka diskon yang lebih besar). Dalam pendekatan ini, fungsi nilai dipelajari melalui interaksi dengan lingkungan itu sendiri. Setelah perkiraan nilai yang baik dipelajari, agen dapat dengan mudah menggunakan argmax atas nilai- Q dalam keadaan tertentu sebagai kebijakan, seperti yang ditunjukkan pada Persamaan 10-2.

Equation 10-2.

$$a^*(s) = \arg \max [Q(s, a)]$$

Ini adalah tindakan optimal di negara bagian s . Contoh algoritme yang termasuk dalam kelas ini adalah Q-learning, SARSA, dan TD-learning, dan metode ini biasanya diterapkan dalam pengaturan tabular di mana pasangan status atau tindakan dapat disebutkan secara eksplisit.

Selain metode berbasis nilai, ada kelas metode berbasis kebijakan. Di sini, alih-alih mempelajari serangkaian perkiraan nilai, kami langsung mempelajari kebijakan untuk bertindak. Kebijakan ini disebut sebagai $(a \mid s)$ dan menyediakan serangkaian probabilitas atas tindakan yang dikondisikan pada keadaan s . Kebijakan ini dapat ditingkatkan dengan menggunakan algoritma gradien kebijakan. Intuisi di balik pendekatan ini adalah menggunakan penghargaan diskon yang diamati yang diperoleh oleh suatu kebijakan selama evaluasi sebagai sarana untuk memperbaiki kebijakan secara langsung. Untuk kasus di mana hasilnya lebih baik dari yang diharapkan, kami meningkatkan kemungkinan tindakan yang terkait dengan hasil itu. Untuk kasus di mana hasilnya lebih buruk dari yang diharapkan, kami mengurangi kemungkinannya. Algoritma gradien kebijakan dikembangkan untuk digunakan dalam kasus aproksimasi fungsi linier.

10.8 DEEP REINFORCEMENT LEARNING

Metode yang baru saja kita diskusikan bekerja dengan baik untuk ruang keadaan kecil, di mana probabilitas untuk tindakan atau perkiraan nilai dapat dihitung untuk semua keadaan dan direpresentasikan dalam memori sebagai array dan matriks sederhana dari nilai titik-mengambang. Dalam kebanyakan simulasi yang menarik, bagaimanapun, ini bukanlah asumsi yang dapat dibuat. Jika kita kembali ke contoh kota simulasi sebelumnya, kemungkinan kombinasi kendaraan dan pejalan kaki jauh melebihi apa yang mungkin dihitung. Lebih jauh lagi, jika kita menggunakan piksel mentah yang tersedia untuk agen dari dalam mobil virtual, bahkan ruang observasi ini tidak dapat dipecahkan.

Di sini, kita membutuhkan metode yang memungkinkan aproksimasi fungsi kompleks untuk mewakili nilai fungsi $V(s)$ atau $Q(s, a)$, atau probabilitas aksi ($a | s$). Dalam banyak kasus, fungsi aproksimator pilihan adalah jaringan saraf dengan beberapa lapisan tersembunyi, yang mengarah ke teknik yang dikenal sebagai Reinforcement Learning dalam. Yang "dalam" mengacu pada beberapa lapisan inferensi yang dilakukan oleh jaringan saraf ini. Beberapa lapisan ini sering diperlukan ketika tidak ada pemetaan linier sederhana antara pengamatan dan tindakan. Dalam hal gambar mentah sebagai input, hal ini hampir selalu terjadi, kecuali untuk gambar yang paling sederhana.

Pendekatan penerapan pendekatan fungsi untuk *Reinforcement Learning* telah sukses besar dalam beberapa tahun terakhir. Mulai tahun 2013 dengan *DeepMind* yang mendemonstrasikan bahwa Deep Q-Network—jaringan saraf dalam yang digunakan untuk memperkirakan nilai Q dari gambar mentah—dapat mempelajari kebijakan untuk memainkan game Atari lebih baik daripada manusia, ada keberhasilan setiap tahun yang mendorong keadaan lapangan lebih jauh. Sekarang dimungkinkan untuk mempelajari kebijakan menggunakan *Reinforcement Deep Learning* untuk melakukan segalanya mulai dari penggerak, hingga bermain game strategi waktu nyata, hingga menyelesaikan lusinan tugas menggunakan satu jaringan. Elemen algoritma kunci untuk memungkinkan keberhasilan ini dalam *Reinforcement Deep Learning* telah berfokus untuk mendapatkan keuntungan menggunakan jaringan saraf sebagai pendekatan fungsi sambil mengurangi kerugian dari pendekatan semacam itu. Ini berarti mengambil keuntungan dari kemampuan mereka untuk memodelkan fungsi nonlinier yang kompleks tanpa jatuh ke dalam ketidakstabilan yang melekat dan kesulitan untuk menafsirkannya.

Dalam domain estimasi nilai, ketidakstabilan ini telah diatasi dengan menggunakan sesuatu yang disebut jaringan target, yang merupakan salinan lama dari model yang digunakan untuk bootstrap, bukan yang terbaru. Ini adalah pendekatan yang diambil di Deep Q-Network, dan telah diadopsi ke sebagian besar pendekatan pembelajaran mendalam berbasis nilai berikutnya. Dalam kasus metode berbasis kebijakan, ini berarti membatasi perbedaan kebijakan baru dari yang lama menggunakan berbagai metode, sering kali didasarkan pada perbedaan KL dalam ruang tindakan kebijakan. Dua pendekatan yang paling populer adalah *Trust Region Policy*, yang memberlakukan batasan KL keras, dan Optimalisasi Kebijakan Proksimal, yang memberlakukan batasan lunak.

10.9 IMITATION LEARNING

Sejauh ini, kita telah membahas perilaku belajar dari awal hanya menggunakan interaksi dengan simulasi/permainan/XR. Namun, dalam kebanyakan kasus, ini bisa menjadi sampel dan waktu yang tidak efisien karena pembelajaran berlangsung melalui coba-coba. Ini juga merupakan kasus bahwa perilaku yang diinginkan harus ditentukan melalui fungsi penghargaan. Sayangnya, fungsi penghargaan ini seringkali sulit ditentukan dengan cara yang benar-benar selaras dengan perilaku yang diinginkan. Misalnya, jika perilaku yang diinginkan adalah agar agen melakukan backflip, imbalan apa yang harus diberikan untuk mendorong perilaku itu dengan cara coba-coba? Dalam banyak kasus, menjadi jauh lebih intuitif untuk sekadar memberikan serangkaian demonstrasi perilaku yang diinginkan. Demonstrasi ini kemudian dapat digunakan untuk mempelajari model perilaku.

Ada beberapa cara di mana hal ini dapat terjadi. Yang pertama adalah bahwa demonstrasi dapat berfungsi sebagai input dan output kumpulan data yang digunakan untuk mempelajari fungsi pemetaan secara langsung dengan cara yang diawasi. Pendekatan ini disebut sebagai kloning perilaku; ini adalah yang paling mudah tetapi belum tentu yang paling efisien. Perhatikan kembali contoh agen virtual yang mengemudi jalan-jalan kota yang disimulasikan. Mungkin ada titik tertentu di mana ada persimpangan jalan. Jika data demonstrasi berisi contoh yang sama dari ke kiri di persimpangan serta ke kanan, model pembelajaran dengan kloning perilaku kemungkinan akan belajar melalui tengah! Ini juga merupakan kasus bahwa kloning perilaku menderita dari kesalahan penggabungan dari waktu ke waktu karena perilaku agen membawanya menjauh dari ruang keadaan demonstrasi yang disediakan selama proses pembelajaran.

Ada sejumlah pendekatan yang mencoba untuk mengatasi kesulitan-kesulitan ini. Mereka sebagian besar berada di bawah domain *Reinforcement Learning* terbalik. Dalam pendekatan ini, algoritma mencoba mengungkap fungsi hadiah yang diikuti oleh demonstrasi dan menggunakan fungsi hadiah itu untuk memandu model yang dipelajari. Pendekatan seperti ini memungkinkan yang terbaik dari kedua dunia di mana agen belajar melalui fungsi hadiah padat yang mencakup seluruh ruang keadaan serta fungsi hadiah yang ditentukan dengan benar, yang akan mendorong perilaku yang diinginkan. Salah satu pendekatan kontemporer khususnya adalah *Imitation Learning* permusuhan generatif, yang menggunakan diskriminator yang dipelajari untuk memberikan sinyal hadiah kepada agen pembelajaran.

10.10 MENGGABUNGKAN PERENCANAAN OTOMATIS DAN PEMBELAJARAN MESIN

Bagian sebelumnya berfokus pada metode yang menggunakan model dunia/masalah yang dihadapi AI dan memanfaatkan model ini untuk perencanaan perilaku. Sebaliknya, kita sekarang berfokus pada *Reinforcement Learning* bebas model, yang menghasilkan perilaku optimal tanpa model masalah yang eksplisit, tetapi dengan berinteraksi dengan simulasi. Kedua pendekatan ini tidak perlu ditentang dan dipisahkan. Memang, (bisa dibilang) hasil yang paling mengesankan di bidang pengambilan keputusan telah muncul dari menggabungkan keduanya secara cerdas. Contoh terbaru yang paling terkenal adalah kesuksesan *DeepMind* di Go bermain menggunakan sistem AlphaGo-nya.⁴² AlphaGo didasarkan pada model deliberatif

dari permainan Go yang dapat digunakan untuk memprediksi hasil dari berbagai urutan keputusan setelah pencarian *state-space* mekanisme yang diuraikan sebelumnya. Namun, AlphaGo menambah sistem perencanaan ini menggunakan jaringan saraf dalam yang dilatih untuk bertindak sebagai penaksir nilai dan kebijakan. Penaksir nilai mengaitkan nilai dengan keadaan. Ini digunakan untuk memangkas kedalaman pohon pencarian perencanaan, membatasi seberapa jauh ke depan proses pencarian harus berjalan. Jaringan kebijakan mengasosiasikan probabilitas ke pasangan keadaan-tindakan: jaringan ini memperkirakan probabilitas suatu tindakan menjadi optimal dalam keadaan tertentu. Ini digunakan untuk memangkas lebar pohon pencarian, memungkinkan proses pencarian untuk fokus hanya pada node yang terlihat lebih mungkin di bawah kebijakan yang optimal.

Algoritme mematuhi protokol berikut: hasil dari proses perencanaan digunakan sebagai data pelatihan untuk jaringan saraf, memberikan mereka nilai dan pemetaan kebijakan untuk dipelajari. Sebagai gantinya, sistem pembelajaran mesin digunakan untuk mempercepat proses perencanaan, memangkas kedalaman dan lebar ruang pencarian. Ini memungkinkan pencarian yang lebih dalam dan kebijakan yang lebih baik untuk ditemukan. Kebijakan yang ditingkatkan ini diumpungkan kembali ke pembelajaran mesin untuk meningkatkan akurasi nilai yang dipelajari dan distribusi tindakan. Proses tersebut berulang berkali-kali, dengan perencanaan dan sistem pembelajaran yang saling melengkapi dengan data yang semakin akurat. Dengan cara ini, kedua sistem dapat secara iteratif meningkatkan dan memperbaiki satu sama lain dalam prosesnya. Pendekatan ini memungkinkan AlphaGo untuk mengalahkan juara dunia dalam game.

Selain memainkan permainan papan dengan kapasitas manusia super, menggabungkan dua metode dapat memungkinkan pengembang untuk menukar secara real time antara akurasi (disediakan oleh perencanaan tradisional) dan kecepatan (disediakan oleh aproksimator fungsi jaringan saraf) dalam pengambilan keputusan. Peningkatan kecepatan selama evaluasi diperoleh dengan harga peningkatan waktu pelatihan sebelumnya. Namun, dalam banyak kasus, ini adalah pertukaran yang dapat diterima, dan yang serupa dengan pertukaran yang dibuat ketika mempertimbangkan waktu pengembangan prarilis pada perilaku apa pun dalam simulasi. Menggabungkan metode ini juga dapat menjadi penting untuk situasi di mana sejumlah besar keputusan harus dibuat dalam simulasi, beberapa dengan kesetiaan yang lebih besar daripada yang lain.

Aplikasi

Reinforcement Learning dan *Imitation Learning* membawa banyak janji AI hebat untuk game dan XR. *Reinforcement Learning* membuka jalan untuk menciptakan agen yang dapat memecahkan masalah yang tidak kita pahami sepenuhnya. Ini sangat kontras dengan pendekatan sebelumnya yang membutuhkan kemampuan untuk menggambarkan masalah dengan sempurna (AI deliberatif dan perencanaan otomatis) atau untuk memecahkan masalah dengan cukup baik (AI reaktif). Pembelajaran mesin juga dapat meningkatkan kekuatan perencana melalui pendekatan gabungan seperti AlphaGo, yang mewakili keadaan seni dalam pemecahan masalah dan pengambilan keputusan. Hebatnya, komunitas riset *Reinforcement Learning* telah menumbuhkan hubungan kuat dengan budaya video-game

dengan mengadopsi game sebagai tolok ukur favorit mereka. Ini kontras dengan fakta praktis bahwa hanya sebagian kecil game komersial yang menggunakan konsep pembelajaran mesin saat ini. Namun, ada sedikit keraguan bahwa kegilaan komunitas penelitian *Reinforcement Learning* untuk game akan segera memberikan pengembalian ke industri. Memang, pembelajaran mesin memang membawa solusi untuk masalah praktis yang dihadapi saat merancang game AI. Jadi, sekarang mari kita periksa apa yang saat ini dapat dilakukan, dan apa yang menurut kami kemungkinan besar akan terjadi di masa depan.

Kembali ke diskusi di bagian "Perilaku" dalam pengantar Gambar 10-1, kami menekankan bahwa dampak paling kuat dari pembelajaran mesin berada di lapisan persepsi sistem otonom: memahami data numerik kompleks yang berasal dari sensor (di antaranya penambahan data dapat dilihat sebagai contoh). Keberhasilan pertama pembelajaran mendalam berada di domain seperti visi komputer, pengenalan gerakan, dan NLP. Ketika kita sampai pada pengambilan keputusan dan generasi perilaku, pembelajaran mesin telah terbukti sangat berharga untuk menyelesaikan tugas-tugas di tingkat terendah dari arsitektur pada Gambar 10-1.

Reinforcement Deep Learning unggul dalam memecahkan permainan arcade Atari, yang lebih didasarkan pada refleks dan koordinasi yang baik daripada pemodelan dan pemecahan masalah yang sulit. Balapan, pertarungan, dan simulasi olahraga adalah domain yang bagus untuk aplikasi *Reinforcement Learning*. Ini tidak terlalu mengejutkan. Sebelumnya, kami menekankan bahwa tugas motorik sensorik tingkat rendah sebagian besar merupakan masalah pengoptimalan kontinu multidimensi (berlawanan dengan pengoptimalan kombinatorial diskrit). Pada saat yang sama, masalah persepsi juga memiliki sifat multidimensi yang berkelanjutan. Ini juga beresonansi dengan hasil pembelajaran mesin yang mengesankan dalam generasi animasi, domain (sangat) berkelanjutan lainnya. Dalam praktiknya, *Reinforcement Learning* adalah kandidat yang bagus untuk mengendalikan agen pada tingkat motorik sensorik terendah.

Apa yang kita pelajari dari studi akademis juga merupakan kesulitan teknik *Reinforcement Learning*, dalam keadaan mereka saat ini, untuk mengatasi tingkat tertinggi dari hierarki Gambar 10-1. Salah satu game arcade paling sulit untuk *Reinforcement Deep Learning* adalah Montezuma's Revenge. Ini melibatkan pemecahan teka-teki dengan mengurutkan serangkaian tindakan yang panjang seperti mengambil kunci di satu ruangan untuk membuka pintu di ruangan lain. Eksekusi rencana ini dapat berlangsung hingga beberapa menit waktu nyata, yang sangat kontras dengan beberapa detik perencanaan—paling lama—yang diperlukan untuk menyelesaikan *Space Invaders* atau *Breakout*.

10.11 KESIMPULAN

AI adalah bidang yang kaya yang mengusulkan pendekatan berbeda untuk masalah generasi perilaku. Daripada melihat mereka sebagai pesaing, kami lebih memilih untuk menekankan sifat komplementer dari pendekatan ini. Kami percaya bahwa desain sistem AI perilaku untuk video game atau aplikasi XR harus dimulai dengan dekomposisi yang jelas dari masalah umum menjadi beberapa subtugas dan memahami kendala yang terkait dengan

setiap subtugas. Kemudian, pendekatan yang paling tepat harus dipilih untuk setiap modul. Meskipun tidak ada aturan mutlak yang dapat diterapkan dalam semua kasus, beberapa prinsip umum dapat diuraikan:

- Jika kita tahu persis perilaku apa yang ingin kita hasilkan dan perilaku ini tidak melibatkan pemecahan masalah yang sulit seperti jalur terpendek, manajemen sumber daya, atau eksplorasi cerdas, AI reaktif adalah kandidat yang bagus. Namun, harus diharapkan bahwa pengembangan AI akan menjadi proses yang membosankan yang membutuhkan banyak percobaan dan kesalahan untuk memperbaiki semua kasus tertentu yang perlu ditutupi oleh perilaku tersebut.
- Jika perilaku yang ingin kita hasilkan mencakup pemecahan masalah yang sulit dan kita tahu betul bagaimana menjelaskan masalah ini, atau jika kita tidak memiliki sumber daya untuk memperbaiki kasus per kasus AI reaktif, AI deliberatif harus lebih disukai. Namun, dibutuhkan keterampilan teknis untuk mengimplementasikan mesin perencanaan.
- Jika masalah yang harus kita pecahkan terlalu sulit atau kita tidak tahu persis bagaimana menjelaskannya, kita dapat mencoba pembelajaran mesin. Hal ini terutama berlaku untuk perilaku dalam skala kecil dalam hal ruang dan waktu. Pembelajaran mesin masih merupakan bidang penelitian yang bergerak cepat, dan aplikasi dalam domain hiburan digital dan XR sangat terbatas saat ini. Oleh karena itu, beberapa upaya penelitian harus diharapkan.

Menyatukan kekuatan dari tiga paradigma utama dalam AI perilaku adalah kunci untuk mengatasi tantangan baru XR.

BAGIAN VI

GUNAKAN KASUS DALAM EMBODIED REALITY

Teknologi hanya sebaik aplikasinya. Dalam bab-bab berikut, kita melihat bagaimana teknologi imersif digunakan di dunia nyata. Pembaca mungkin akrab dengan "Siklus Hype," grafik hipotetis (lihat Gambar VI-1) yang menggambarkan kesulitan yang berkembang dari teknologi baru. Sejak head-mounted display (HMD) pertama dibuat pada tahun 1968, eXtended reality (XR) tampaknya terjebak dalam kekecewaan. Sejak XR kembali ke kesadaran publik di sini pada abad kedua puluh satu, kami telah melihat banyak awal yang salah, mulai dari runtuhnya Meta hingga serangan balik privasi terhadap Google Glass.

Meskipun beberapa orang mungkin menganggap kegagalan ini untuk menunjukkan bahwa XR adalah teknologi overhyped lainnya, aplikasi yang disajikan dalam bab ini mungkin berbeda. Perlahan tapi pasti, teknologi XR menemukan ceruknya, mendaki menuju dataran tinggi produktivitas, aplikasi demi aplikasi.

Dalam Bab 11, Dilan Shah, salah satu pendiri YUR, Inc. membahas bagaimana kita dapat menyesuaikan teknologi imersif untuk orang-orang dengan kondisi kesehatan yang berbeda. Industri perawatan kesehatan adalah ruang di mana proses dan prosedur harus dipatuhi secara ketat untuk memastikan perawatan yang optimal. Bagaimana Virtual Reality (VR) dapat disesuaikan dengan ruang ini dan manfaat apa yang dapat diberikannya? Dalam bab ini, kami memberikan contoh praktis yang mendalam tentang bagaimana tracking tangan dapat menstabilkan tremor pada pasien Parkinson di lingkungan virtual.



Gambar VI-1. Siklus Hype

Di Bab 12, Marc Rowley melihat peran XR dalam memberikan hiburan olahraga kepada penggemar. Marc baru-baru ini menutup 18 tahun di ESPN untuk menemukan startup yang menghasilkan gambar CGI langsung dari acara olahraga. Setiap bidang memiliki tuntutannya sendiri, dan tidak ada latensi dan imersif yang menjadi bagian media hiburan yang lebih populer daripada olahraga.

Akhirnya, di Bab 13, insinyur VR Rosstin Murphy dari STRIVR membawa kita melalui empat kasus penggunaan nyata dari pelatihan VR perusahaan, termasuk rumah banjir, lantai pabrik, perampokan toko, dan penyampaian beberapa berita yang sangat buruk.

Teknologi XR bukan lagi sekadar demonstrasi teknologi. Dengan investasi Walmart di 17.000 Oculus Go, XR berbelok di tikungan dan mendaki lereng pencerahan. Masih akan ada speedbumps dan awal yang salah, tetapi dalam contoh yang Anda lihat di bagian ini, kita melihat contoh nyata pertama dari aplikasi XR yang akan bertahan dalam ujian waktu. Saya berharap bab-bab ini dapat menginspirasi Anda untuk membangun aplikasi praktis Anda sendiri. Aplikasi imersif besar berikutnya mungkin datang dari Anda; keluar dan bangun!

BAB 11

TEKNOLOGI KESEHATAN *AUGMENTED REALITY* DAN *VIRTUAL REALITY*

Bab ini mencakup isu-isu yang berkaitan dengan desain pengalaman *virtual reality* (VR) dan *augmented reality* (AR) yang digunakan dalam konteks perawatan kesehatan, dan memberikan tutorial untuk menggunakan data gerak dari controller untuk mengurangi getaran yang terlihat dari pasien Parkinson di virtual lingkungan. Saat ini, prospek perawatan kesehatan global ditentukan oleh serangkaian kebijakan, tindakan kesehatan masyarakat, metode penyampaian, penelitian klinis berbasis komunitas, terapi, dan inovasi teknologi yang terus berkembang. Tidak ada teknologi tunggal yang menangani semua masalah perawatan kesehatan sendirian, dan sekarang dari pembelajaran mendalam yang diterapkan pada pelipatan protein hingga kesehatan presisi hingga kesehatan populasi, ada banyak pendekatan berbeda yang diambil untuk memecahkan tantangan kesehatan yang sulit.

Dalam perawatan kesehatan, segala sesuatu mulai dari yang canggih (yaitu, fMRI) hingga yang sederhana (yaitu, penjadwalan janji yang efisien) semuanya memiliki peran dalam pemberian perawatan. Teknologi VR dan AR relatif baru dan belum dianggap sebagai konvensi, apalagi standar perawatan, dalam domain kesehatan apa pun. Ruang masalah termasuk pengurangan rasa sakit, pengobatan gangguan stres pasca-trauma (PTSD) dengan terapi paparan, dan pengobatan ambliopia. Ruang-ruang ini telah terbukti matang untuk VR sebagai teknologi pengiriman terapi, sementara pelatihan dan perencanaan bedah telah menemukan kasus penggunaan untuk AR.

Untuk menghindari bias sistemik dan memfasilitasi lebih banyak pandangan dunia tentang subjek tentang bagaimana VR dan AR berlaku untuk kesehatan, detail penting tentang organisasi formal, badan pengawasan, dan proses persetujuan dihilangkan. Sebaliknya, bab ini membahas lebih lanjut tentang upaya tingkat tinggi yang dapat dilakukan untuk merancang teknologi kesehatan dengan lebih baik menggunakan VR dan AR. Penting untuk diketahui bahwa pasien harus menyetujui sebelum mereka dapat mencoba aplikasi atau eksperimen apa pun, dan ada papan peninjau secara tegas untuk tujuan tersebut. Akhirnya, bab ini mencakup pendekatan komersial dan akademis untuk mengatasi masalah dalam perencanaan dan bimbingan, perawatan kesehatan dan pencegahan, serta terapi yang diterapkan dalam pengaturan klinis.

11.1 DESAIN APLIKASI TEKNOLOGI KESEHATAN VR/AR

Membuat aplikasi VR dan AR mengharuskan pengembang untuk mempertimbangkan lingkungan fisik pengguna saat menggunakan teknologi tersebut. Baik, misalnya, pengguna adalah pasien di ruang pra-operasi, sendirian sebelum prosedur, atau pengguna adalah anggota keluarga yang ada di kamar pasien.

Proses desain harus mencakup menghabiskan waktu untuk memahami lingkungan ini dan apa yang terjadi selama skenario tipikal. Kami mungkin cukup mewawancarai dokter,

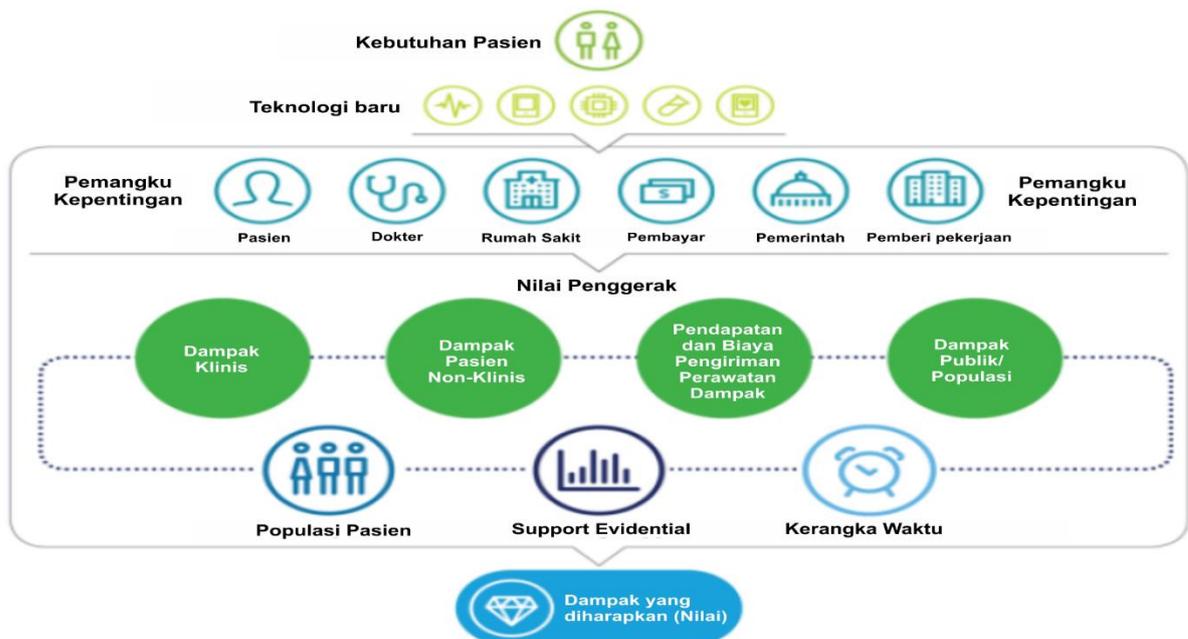
praktisi perawat (NP), staf, dan personel lain yang terlibat untuk menjawab pertanyaan seperti berikut:

- Jika pasien adalah pengguna, apakah keluarga akan tertarik untuk menghabiskan waktu berkualitas dengan pasien di lingkungan X atau pasien sendiri?
- Sejauh mana pengaturan memerlukan interupsi pada pengalaman virtual?
- Apa mobilitas pengguna?
- Berapa durasi pengalaman yang seharusnya?
- Bisakah pengguna memakai headphone?

Beberapa pertimbangan orde kedua mungkin:

- Apakah perangkat VR dan/atau AR akan bersih dan bagaimana caranya?
- Mengikuti pengalaman, bagaimana proses untuk mempertahankannya?
- Siapa yang akan memfasilitasi pengalaman tersebut dan berapa lama waktu yang dibutuhkan?
- Apakah pengguna merasa aman?

Sekali lagi, ruang lingkup bab ini tidak mencakup Badan Pengawas Obat dan Makanan AS (FDA) atau persyaratan peraturan lainnya, tetapi dengan melihat pendekatan penilaian nilai pasien, seperti yang ditunjukkan pada Gambar 11-1, Anda dapat melihat betapa terbatasnya hal ini. ruang menjadi. Menambahkan penelitian pemangku kepentingan yang tepat dan pemikiran sebelumnya tentang antarmuka fisik di jantung aplikasi teknologi kesehatan yang sukses. Ada beberapa contoh kasus penggunaan VR yang disetujui FDA, yaitu VRPhysio oleh VRHealth dan Mindmaze. VR untuk mereka yang tidak mengembangkan aplikasi untuk pasien, ada juga kerangka kerja evaluasi kasus penggunaan kesehatan preventif dan ruang fisik yang perlu dipertimbangkan untuk aplikasi tersebut.



Gambar 11-1. “Kerangka untuk Penilaian Komprehensif Teknologi Medis: Mendefinisikan Nilai dalam Ekosistem Perawatan Kesehatan Baru,” dikembangkan bersama dengan Deloitte Consulting LLP

11.2 UX STANDAR

Desain aplikasi VR dan AR telah berkembang secara signifikan dalam beberapa tahun terakhir saja, menstabilkan cara berbagai sistem dibuat dan tugas diselesaikan. Tidak terlihat lagi selain Virtual Reality Toolkit (VRTK) (lihat Bab 7). Namun, melihat "penggerak" sebagai contoh — sering ditemukan dalam aplikasi VR, penggerak adalah cara pengguna dapat bergerak melalui lingkungan virtual — ada banyak bentuk, dari kurva Bézier hingga "memilih orientasi" hingga teleportasi berbasis waypoint di kanon pengalaman pengguna VR (UX).

Namun, meskipun demikian, ketika digunakan di luar konteks game atau mereka yang berada di industri teknologi yang imersif, jelas bahwa hal-hal yang mungkin tampak biasa dan mudah dipahami sebenarnya tidak demikian. Ini meluas bahkan ke isyarat input dasar untuk perangkat keras AR karena banyak yang belum pernah menggunakan sistem ini sebelumnya. Oleh karena itu, bekerjalah dengan cerdas tanpa User interface jika memungkinkan; misalnya, dengan membuka langsung ke lingkungan virtual utama di mana kegiatan inti berlangsung.

Jenis input umum lainnya yang terlihat dalam aplikasi tertentu adalah tracking tangan, yang memungkinkan pengguna melihat tangan virtual yang mencerminkan gerakan mereka sendiri melalui teknologi seperti *Leap Motion*. Kita dapat melihat ini di salah satu pengalaman belajar yang diwujudkan dengan fokus pasien kanker baru Embodied Labs, yang merupakan bentuk menjaga pengguna tetap terlibat tetapi juga bentuk memerangi keputusan yang dirasakan pengguna ketika mereka tidak melihat tangan mereka di VR. Sebagai catatan khusus, Embodied Labs juga berfokus pada pembuatan dasbor UI desktop untuk meluncurkan pengalaman. Pertimbangkan untuk menggunakan antarmuka desktop serta antarmuka untuk VR atau AR. Ini untuk memanfaatkan input keyboard dan mouse yang tidak terlalu familiar. Lingkungan virtual yang perlu "direset" atau "kedaluwarsa berdasarkan waktu" harus berulang atau diatur dengan cara yang tidak bergantung pada klik untuk memulai ulang.

Dalam proyek contoh yang akan datang, Insight, untuk meminimalkan klik yang tidak disengaja, pengguna harus memegang controller untuk waktu yang minimal agar suatu tindakan dapat diterima. Komponen ini, yang dikenal sebagai menambahkan gesekan untuk UX yang lebih baik, digunakan untuk memperlambat pengguna agar tindakan tetap disengaja. Terkadang, ada trade-off dengan pilihan desain semacam ini, dan jika itu adalah titik frustrasi, jangan biarkan hal itu berlama-lama bagi pengguna—ubahlah dengan cepat.

Pilih Lingkungan yang Tenang

Dengan proyek Insight, tindakan diambil untuk menciptakan lingkungan yang kontras dengan lingkungan pasien atau belajar pada umumnya. Penempatan di tepi air dan audio berupa lonceng angin yang halus mengundang rasa relaksasi.

Penggunaan VR dalam perawatan paliatif meningkatkan kualitas hidup melalui lingkungan yang indah. Ini tidak diragukan lagi sebagian karena kesediaan atas nama pengembang untuk berpikir dalam istilah "ruang dunia" bukan "ruang layar" dan Scene storyboard dari pandangan luas untuk memastikan pemirsa tertarik. Menggunakan colokan audio spasial untuk isyarat relaksasi (contohnya mungkin termasuk gemerisik lembut, angin bertiup melalui beberapa lonceng, atau suara ombak) akan membantu menarik pengguna ke lingkungan dan meningkatkan kepercayaan.

Kenyamanan

Cara lain penerapan VR dan AR ke dalam organisasi kesehatan adalah dengan menciptakan efisiensi ekonomi. Untuk menghemat waktu dokter dan perawat di klinik, misalnya, Augmedix menawarkan platform otomatisasi dokumentasi yang didukung oleh pakar manusia dan perangkat lunak. Meskipun platform pilihannya adalah Google Glass (dan karenanya tidak cukup AR), pengirimannya memerlukan perangkat yang dikenakan di kepala, yang membebaskan dokter dari pekerjaan komputer dan memungkinkan mereka untuk fokus pada hal yang paling penting: perawatan pasien. Augmedix melayani 12 sistem kesehatan terkemuka di negara ini, di sebagian besar spesialisasi berbasis klinik, dengan peningkatan produktivitas dokter rata-rata 30%.

Di bagian selanjutnya, kami membahas cara mengotomatiskan penilaian sentuhan jari-hidung pada tremor visuomotor, yang juga merupakan tujuan makalah yang dikirimkan ke Arxiv pada tahun 2018 oleh tim di belakang Insight. Karena tes sentuhan jari-hidung diberikan sebagian besar melalui logika permainan yang ada di dalam aplikasi berbasis Unity, ini memungkinkan dokter untuk melanjutkan tugas-tugas lain dan memberikan kenyamanan.

11.3 TUTORIAL: EKSPERIMEN INSIGHT PARKINSON

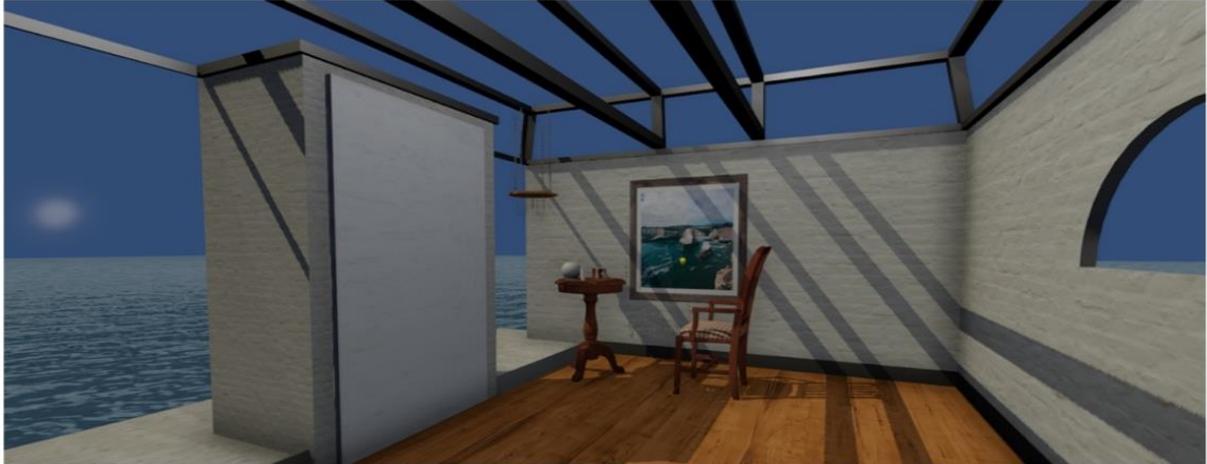
Penyakit Parkinson adalah gangguan neurodegeneratif yang berkembang lambat dengan gejala termasuk tremor, kekakuan tungkai, kelambatan gerakan, dan masalah dengan keseimbangan. Kemajuan penyakit dapat sangat mempengaruhi kualitas hidup dari cacat fisik hingga depresi. Insight, adalah platform yang berfokus pada pasien VR untuk pasien penyakit Parkinson dan keluarga mereka. Insight dibangun di atas pengamatan dasar bahwa controller VR normal dapat mengirimkan data posisi dan orientasi frekuensi tinggi.

Insight

Insight berfungsi sebagai alat penilaian VR, manajemen, dan aplikasi pendidikan kesehatan. Pasien biasanya menemui dokter untuk memantau perkembangan penyakit dan menyesuaikan pengobatan dan terapi rehabilitasi pada interval kunjungan klinik yang ditentukan berdasarkan gejala. Insight tetap bersama pasien sepanjang hidup mereka, terus menilai pengguna antara kunjungan dan membantu penyedia mereka di klinik.

Platform ini memanfaatkan perawatan saat ini melalui informasi kesehatan pihak ketiga seperti catatan medis dan data pergerakan yang dikumpulkan dalam VR untuk membuat penilaian status kesehatan pasien, memberikan latihan rehabilitasi yang dipersonalisasi, dan memandu tim dokter dalam pengambilan keputusan berdasarkan data. Sebelum pasien memulai latihan rehabilitasi, mereka menyentuh satu set lonceng angin di rumah virtual, yang ditunjukkan pada Gambar 11-2, yang kemudian mentransfer gejala ke dunia virtual. Selama sisa pengalaman, gerakan dan suara lonceng angin menandakan gejala pasien sementara gerakan pasien sekarang akan bebas dari getaran. Pasien kemudian dipandu melalui latihan rehabilitasi pribadi berbasis bukti yang sambil meningkatkan fungsi fisik juga mengumpulkan data untuk penilaian perkembangan penyakit. Di akhir penilaian, pasien menerima gambaran tentang status kesehatan mereka saat ini, termasuk obat-obatan, skor kesehatan Insight yang diperoleh dari pengukuran gejala, dan opsi untuk menghubungi dokter

melalui telemedicine. Dokter ini akan memiliki laporan yang dibuat oleh Insight yang mencakup informasi gejala yang dikumpulkan.



Gambar 11-2. Lingkungan Insight adalah ruang tepi laut yang tenang dengan pemandangan langit dan lonceng angin lembut untuk memungkinkan pasien sedikit lebih rileks saat melakukan penilaian jangkauan.

Insight menyediakan platform untuk pengumpulan data sukarela.

Bagaimana Itu Dibangun

Platform Data Pasien Insight dibangun dengan menggunakan kombinasi mesin game Unity 2017.3.0f3 dan alat analisis data MATLAB dan Python.

Filter lolos rendah untuk tremor tangan

Bagian terpenting dari proyek ini melibatkan transformasi cara seseorang dengan tremor sebenarnya bergerak versus bagaimana tampaknya mereka bergerak saat melihat tangan mereka sendiri melalui perangkat VR. Dibangun dengan menggunakan filter low-pass, atau rata-rata bergerak, Skrip Smoothed Hand C# yang dilampirkan pada model tangan pengguna menangkap data posisi dan rotasi transformasi dari objek input yang dilacak VR sebagai input dan output data yang dihaluskan untuk transformasi model tangan.

Lingkungan

Inspirasi lingkungan sangat dipengaruhi oleh salah satu mentor, Hannah Luxenberg, yang menjelaskan bahwa alih-alih arahan seni yang mirip dengan klinik, tujuannya adalah untuk menciptakan suasana yang menenangkan. Sebagian besar model dibuat di Maya oleh Serhan Ulkumen. Menggunakan sistem medan Unity, medan dihasilkan dengan menggunakan peta ketinggian dan kemudian pohon ditempatkan.

11.4 ANALISIS DAN PELAPORAN DATA

Pada dasarnya, tim mengumpulkan nilai tremor X,Y,Z dari posisi controller VR, dan setelah itu analisis data memberikan detail yang berkaitan dengan pasien dan pengasuh tentang tremor.

Berikut pseudocodenya:

Impor untuk mendukung analisis dan fungsi data

Kekuatan Augmented dan Virtual Reality dalam Bisnis (Dr. Agus Wibowo)

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec import os.path
Loading file data pasien
while True:
    try:
        file = 'patientData.csv'
        data=pd.read_csv(file)
        print(data.head())
        print()
        y=[]
        z=[]
        x=data.iloc[:,0].values
        y=data.iloc[:,1].values
        z=data.iloc[:,2].values
        print(type(x))
        nbins=30
        Xr=np.fft.fft(x,nbins)
        X=abs(Xr[1:round(len(Xr)/2)])
        Yr=np.fft.fft(y,nbins)
        Y=abs(Yr[1:round(len(Yr)/2)])
        Zr=np.fft.fft(z,nbins)
        Z=abs(Zr[1:round(len(Zr)/2)])
        x2=x-x.mean()
        y2=y-y.mean()
        z2=z-z.mean()
        fig1 = plt.figure()
        #print(type(fig))
        tt=np.linspace(0,np.pi,len(X))
        plt.plot(tt,X,tt,Y,tt,Z,alpha=0.6)
        plt.xlabel('Frequency (Normalized)')
        plt.ylabel('Amplitude')
        plt.title('Frequency Response')
        plt.legend(('X-axis', 'Y-axis', 'Z-axis'),loc='upper right')
        #plt.show()
        fig1.savefig('plotF.png')
        fig1.savefig('plotF.pdf')
        fig2 = plt.figure()
        score=int((1-(1.07*(x2.std()+y2.std()+z2.std())))*100)
        gs = gridspec.GridSpec(1, 2, width_ratios=[4,1])    print(gs)
        ax1 = plt.subplot(gs[0])    tt2=np.linspace(0,len(x2)/50,len(x2))
        plt.plot(tt2,x2,tt2,y2,tt2,z2,alpha=0.6)
        plt.xlabel('Time (s)')

```

```

plt.ylabel('Movement')
plt.title('Movement Insight')
plt.legend(('X-axis', 'Y-axis', 'Z-axis'),loc='upper right')
ax2 = plt.subplot(gs[1])
plt.bar(['Higher is
better'],score,alpha=0.6,color=['C3'])
plt.ylim((0,100))
plt.title('Insight Score: '+str(score))
#plt.show()
fig2.savefig('plotT.png')
fig2.savefig('plotT.pdf')

```

Menghitung statistik di sekitar nilai tremor

```
stats2show=[x2.std(), y2.std(), z2.std()]
```

```
fig3 = plt.figure()
```

```
plt.bar(['X', 'Y', 'Z'],
```

```
Stats2show,
```

```
alpha=0.6,
```

```
color=['C0', 'C1', 'C2'])
```

```
plt.xlabel('Axis')
```

```
plt.ylabel('Tremor')
```

```
plt.title('Tremor values')
```

```
fig3.savefig('plotS.png')
```

```
fig3.savefig('plotS.pdf')
```

```
print('Analysis Completed!')
```

Impor untuk mendukung pembuatan pembacaan PDF dari nilai getaran dan inisialisasi variabel penting

```
import time
```

```
from reportlab.lib.enums import TA_JUSTIFY
```

```
from reportlab.lib.pagesizes import letter
```

```
from reportlab.platypus import SimpleDocTemplate, Paragraph, Spacer, Image
```

```
from reportlab.lib.styles import getSampleStyleSheet, ParagraphStyle
```

```
from reportlab.lib.units import inch
```

```
doc = SimpleDocTemplate("form_letter.pdf",pagesize=letter,
```

```
rightMargin=72,leftMargin=72,
```

```
topMargin=72,bottomMargin=18)
```

```
Story=[]
```

Mari kita lihat lebih dekat apa yang terjadi:

- data akselerometer dimuat (data=pd.read_csv(file)).
- Komponen x, y, dan z masing-masing diekstraksi ke dalam variabel x, y, dan z, pada baris berikut:

```
x=data.iloc[:,0].values
```

```
y=data.iloc[:,1].values
```

```
z=data.iloc[:,2].values
```

- Fast Fourier Transform (FFT) dari setiap komponen dihitung dan paruh pertama (0 hingga pi dalam domain frekuensi yang dinormalisasi) didistribusikan ke dalam variabel X_r , Y_r , dan Z_r , masing-masing.
- Respon frekuensi setiap komponen (sebagai fungsi dari frekuensi yang dinormalisasi: 0 hingga pi) diplot (lihat Gambar 11-1).
- Skor yang mencerminkan deviasi standar (gemetar) dari sinyal yang direkam dihitung—lebih banyak guncangan akan menghasilkan skor yang lebih rendah ($\text{score}=\text{int}((1-(1.07*(x2.\text{std}()+y2.\text{std}()+z2.\text{std}())))*100)$).
- Standar deviasi setiap sumbu (x , y , z) dihitung dan diplot pada garis berikut:

```
Xr=np.fft.fft(x,nbins)
```

```
X=abs(Xr[1:round(len(Xr)/2)])
```

```
Yr=np.fft.fft(y,nbins)
```

```
Y=abs(Yr[1:round(len(Yr)/2)])
```

```
Zr=np.fft.fft(z,nbins)
```

```
Z=abs(Zr[1:round(len(Zr)/2)])
```

- Perhatikan bahwa baris 36, 38 dan 47 sedikit berubah (direvisi).

Untuk singkatnya, kode yang tersisa untuk menyusun laporan PDF pergerakan pasien dihilangkan; untuk melihat kode itu, buka repositori GitHub untuk buku ini. Ini membutuhkan string bersarang dan nama file untuk media dalam blok kode pemformatan yang disediakan oleh perpustakaan ReportLab. Postingan Devpost berisi video yang menampilkan aplikasi yang dihasilkan dalam tindakan, dan kode ditautkan ke dalam repositori GitHub.

Perangkat keras yang digunakan:

- HTC Vive

Aset eksternal yang digunakan:

- SteamVR
- Frames Pack
- Post-Processing Stack

Tool yang digunakan :

- Untuk analisis : packages used included NumPy dan Pandas
- Untuk Visualisasi : Matplotlib

Tekstur untuk model :

- GGTexture

Perusahaan

Bagian berikut mencakup perusahaan yang menggunakan VR dan AR untuk membantu orang dalam berbagai cara dalam perawatan kesehatan. Sebagai permulaan, Profesor Radiologi dan Teknik Elektro dan Bioteknologi Universitas Stanford, dan co-direktur IMMERS, Brian Hargreaves, PhD, telah mengartikulasikan perincian yang bagus tentang di mana nilai terletak di sepanjang spektrum teknologi imersif di klinik. Sebagai latar belakang, IMMERS

adalah inkubator untuk Mixed Reality medis (MR) dan realitas diperpanjang (XR) di Universitas Stanford.

MR atau AR berguna di area yang memerlukan hamparan informasi pada pasien, seperti perencanaan, bimbingan, dan penilaian. Meskipun VR digunakan untuk komponen imersifnya, hal itu mungkin memudahkan dokter dalam pelatihan untuk memahami topik medis atau menjelaskan topik tersebut kepada pasien.

Perencanaan dan Bimbingan

Perencanaan dan bimbingan telah menjadi karakteristik operasi terkait dalam kasus penggunaan teknologi kesehatan VR dan AR, tetapi beberapa, termasuk perangkat lunak pelurus ortodontik Archform, yang berbasis di Unity, melihat potensi teknologi imersif dalam alur kerja yang berbeda.

Teater Bedah

VR presisi memungkinkan ahli bedah saraf, pasien, dan keluarga mereka untuk "berjalan" melalui struktur anatomi pasien sendiri. Misalnya, ahli bedah, pasien, dan keluarga dapat berdiri dengan arteri di sebelah kanan mereka, struktur tulang tengkorak di kaki mereka, dan dengan melihat dari balik bahu mereka, mereka dapat mengamati tumor atau patologi vaskular. Pengalaman mendalam ini memungkinkan mereka untuk memahami patologi dan rencana pembedahan mereka.

Oso VR

Oso VR adalah platform pelatihan bedah VR tervalidasi terkemuka yang dirancang untuk ahli bedah, tim penjualan, dan staf rumah sakit dari semua tingkat keahlian. Produk perusahaan menawarkan interaksi berbasis tangan yang sangat realistis dalam lingkungan pelatihan imersif yang berisi prosedur dan teknologi mutakhir.

Archform

Archform, sebuah perusahaan perangkat lunak yang menyediakan alat koreksi gigi intuitif bagi para ortodontis, menunjukkan bahwa bagi penggunanya, pesona menggunakan antarmuka VR adalah dapat melihat file .stl dalam 3D—sehingga mempercepat alur kerja. Bagi penggunanya, proses memanipulasi gigi dan memeriksa kesejajaran gigi dari berbagai sudut ditingkatkan dengan kemampuan untuk mengontrol orientasi model dengan cepat dan melihatnya di VR.

Pengalaman yang Dirancang untuk Pendidikan Kedokteran

Pengalaman berikut semuanya sebanding, meskipun beberapa pengalaman seperti Stanford Virtual Heart Project atau Embodied Labs mungkin bermanfaat bagi jenis pengasuh tertentu.

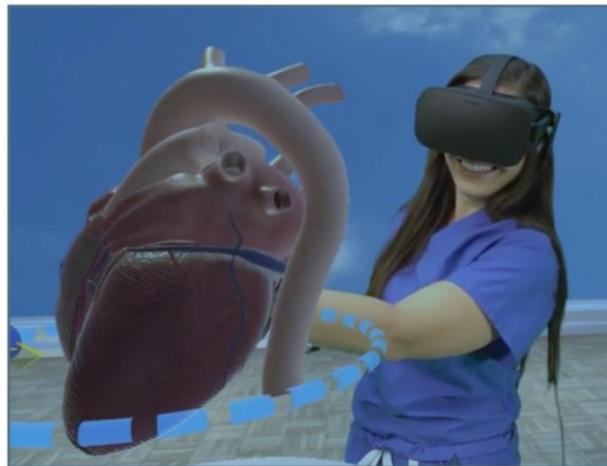
Embodied Labs

Bercerita dalam VR membutuhkan perhatian yang luar biasa terhadap detail, dan, terutama jika tujuannya adalah seputar pembelajaran yang diwujudkan, pengalaman tersebut harus menangkap banyak detail kehidupan nyata. Embodied Labs menggunakan video 360 derajat bersama dengan berbagai elemen interaktif untuk menyampaikan pengalaman pasien kepada pengasuh. Pengalaman baru-baru ini menggunakan tracking suara dan tangan,

memungkinkan pengguna untuk mengambil peran pasien di hadapan anggota keluarga selama peristiwa penting yang sangat penting selama tahap kritis berbagai penyakit.

Lighthaus

Lighthaus, sebuah perusahaan teknologi San Francisco Bay Area, dan David Axelrod, MD di Rumah Sakit Anak Stanford Lucille Packard, seorang ahli jantung pediatrik, berkolaborasi dalam sebuah proyek yang disebut Proyek Jantung Virtual Stanford (SVHP). Digunakan oleh mahasiswa dan praktisi, SVHP dibuat untuk memecahkan berbagai kelainan jantung pediatrik dan prosedur yang diperlukan untuk memperbaiki masing-masing dalam pengalaman VR interaktif. Proyek ini membuka ke perpustakaan hati, dan sebagai pengguna, Anda dapat menarik masing-masing ke area utama untuk interaktivitas. Penampil dapat memutar jantung, seperti yang digambarkan pada Gambar 11-3, dan melihat prosedur yang diperlukan untuk memperbaiki penyakit tersebut.



Gambar 11-3. Seorang pengguna yang memakai Oculus Rift memutar jantung virtual di tempatnya saat berdetak

MVI Health

MVI Health, perusahaan patungan antara Penumbra dan Sixsense, adalah perangkat keras VR yang dapat dihapus dengan controller yang dilengkapi untuk digunakan dalam skenario pelatihan medis. Dengan demikian, demonstrasi utamanya di GDC 2018 adalah trombektomi yang menggunakan teknologi MVI Health untuk melatih seseorang secara virtual cara menyedot gumpalan (trombus) dari pembuluh darah.

Keterjangkauan untuk dapat mengatur ulang semua peralatan dengan mengklik tombol, menghindari kekacauan, dan mengaktifkan tinjauan kinerja menjadikan ini contoh yang jelas mengapa pendidikan kedokteran memerlukan penawaran produk MVI Health daripada metode pelatihan lain untuk prosedur ini.

The Better Lab

Better Lab, yang berbasis di University of California, San Francisco, menerapkan pemikiran desain untuk masalah yang berpusat pada pasien. Saat ini, proyek VR perusahaan yang mencakup pasien trauma dan didanai oleh hibah HEARTS hampir selesai untuk headset

mandiri di tahun mendatang. Zuckerberg San Francisco General (ZSFG) adalah satu-satunya pusat trauma Level Satu di San Francisco. Dari 255 kasus trauma yang diterima setiap bulan, 90 adalah "900 aktivasi" tingkat tinggi yang membutuhkan kecepatan dan koordinasi yang intens di berbagai departemen. Setiap konfigurasi tim trauma baru karena penyedia dan staf dirotasi berdasarkan shift dan bulan. Untuk menjelaskan variasi dalam komposisi tim, praktisi harus memiliki bahasa dan proses standar yang diinformasikan oleh rasa empati terhadap peran, perhatian, dan prioritas satu sama lain. Pengalaman ini menangkap rekaman persetujuan pasien 360 derajat yang nyata untuk menunjukkan koordinasi perawatan seperti orkestra yang disediakan di fasilitas trauma Level Satu.

11.5 PENGALAMAN YANG DIRANCANG UNTUK DIGUNAKAN OLEH PASIEN

Perusahaan berikut menerapkan AR dan VR dengan cara yang memberikan manfaat langsung bagi pasien. VRPhysio, oleh VR Health, adalah produk yang disetujui FDA.

Vivid Vision

Vivid Vision merawat orang-orang dengan Amblyopia. Prosesnya dimulai dengan dunia VR dan membagi Scene menjadi dua gambar: satu untuk mata yang kuat dan satu untuk mata yang lemah. Selanjutnya, kurangi kekuatan sinyal objek di mata yang kuat dan tingkatkan untuk mata yang lemah agar lebih mudah bekerja sama. Seiring perkembangan pasien, tujuannya adalah tidak perlu lagi modifikasi gambar untuk menggabungkannya dan melihat secara mendalam sepanjang waktu. Setiap minggu, pasien membutuhkan sedikit bantuan, sehingga perbedaan antara mata menjadi lebih kecil dan lebih kecil. Dengan latihan, kedua mata belajar bagaimana bekerja sama dan bekerja sama.

VRHealth

Studi Kasus dari Institusi Akademik Terkemuka

VRHealth mengkhususkan diri dalam mengembangkan alat dan konten medis sambil memberikan analitik waktu nyata. Produknya, VRPhysio, terdaftar FDA sebagai alat penilaian olahraga dan jangkauan gerak. Untuk memulai, VRPhysio dibuka dengan penilaian Range of Motion (ROM), seperti yang ditunjukkan pada Gambar 11-4. Terapis fisik virtual mendemonstrasikan gerakan dan aplikasi memberikan penyesuaian sesi pasien sesuai dengan penilaian ROM dan rencana perawatan mereka.

Kemudian, mereka memilih video pendek dari berbagai konten—klip musik, TED Talks, film pendek, dan banyak lagi. Akhirnya, laporan ringkasan terperinci dibuat untuk setiap sesi pelatihan.

USC ICT Bravemind (Courtesy of USC Institute of Creative Technologies)

Bravemind adalah aplikasi untuk dokter yang mengkhususkan diri dalam mengobati PTSD. Ini memberikan alternatif untuk melakukan paparan di zona perang dan/atau membuat orang trauma kembali dengan PTSD terkait pertempuran. Dua lingkungan virtual utama termasuk Irak dan Afghanistan. Pasien dapat terlibat dalam patroli kaki, konvoi, evakuasi medis melalui helikopter dalam berbagai skenario. Setiap skenario memungkinkan dokter untuk menyesuaikan lingkungan untuk memasukkan ledakan, baku tembak, serangan pemberontak, dan bom pinggir jalan. Tingkat cedera pasukan koalisi dan sipil, kerusakan

kendaraan (jika skenario konvoi digunakan) dan arah ledakan dapat diubah. Efek suara termasuk suara khas dari zona pertempuran (yaitu, pelepasan senjata), kebisingan kota sekitar (misalnya, adzan, serangga berdengung), obrolan radio, pesawat di atas kepala, dan banyak lagi. Umpan balik vibrotaktil memberikan sensasi yang biasanya terkait dengan gemuruh mesin, ledakan, baku tembak, dan kebisingan sekitar yang terkait. Mesin pengharum dapat digunakan untuk memberikan aroma yang relevan dengan situasi (mis., Cordite, bahan bakar diesel, sampah, bubuk mesiu).



Gambar 11-4. Bagian dalam lingkungan virtual dan avatar untuk membantu administrasi tes ROM

Teknologi Tangan Pertama, SnowWorld

Lebih dari satu dekade penelitian dan studi klinis telah menunjukkan bahwa VR yang imersif dapat secara signifikan mengurangi rasa sakit, menghilangkan stres, dan membangun ketahanan. Firsthand Technology telah menjadi bagian dari tim peneliti perintis yang telah membentuk bidang kontrol nyeri VR dan membantu membangun aplikasi pereda nyeri VR pertama, SnowWorld.

Penelitian kontrol rasa sakit VR berawal dari Ramachandran dan Rogers-Ramachandran (1996) yang menemukan hubungan antara gambar visual sintetis dan rasa sakit fisik ketika mereka menggunakan "kotak Virtual Reality" berteknologi rendah yang dibuat dengan cermin untuk meredakan nyeri tungkai hantu yang diamputasi. Pada tahun 2000, sebuah tim di Human Interface Technology Lab (HITL) yang dipimpin oleh Direktur Tom Furness dan psikolog Hunter Hoffman menerbitkan hasil pertamanya, menunjukkan bahwa VR yang dihasilkan komputer dapat secara signifikan mengurangi rasa sakit pasien. Sejumlah penelitian selanjutnya menggunakan SnowWorld menemukan bahwa VR secara signifikan lebih efektif daripada hiburan lain seperti film dan permainan komputer berbasis layar. Firsthand Technology telah menyusun daftar referensi utama dan artikel jurnal tentang penelitian pereda nyeri VR dalam Bibliografi Pereda Sakit VR.

Kesehatan Proaktif

Ketika berpikir tentang perawatan kesehatan di Amerika Serikat, seringkali dengan konotasi reaktif. Seseorang jatuh sakit dan kemudian mencari antibiotik, seseorang mengalami serangan jantung dan kemudian membutuhkan obat statin dan kolesterol, dan seterusnya. Kesehatan proaktif atau preventif didefinisikan sebagai pengoptimalan untuk kesehatan pribadi ketika kesehatan Anda relatif statis. Misalnya, gagasan berolahraga termasuk dalam kesehatan proaktif karena dapat membantu mengurangi faktor risiko berbagai penyakit bila dilakukan secara konsisten. Perusahaan berikut berperan dalam kesehatan proaktif menggunakan VR atau AR.

Blackbox VR

Black Box VR mengambil VR dan menggabungkannya dengan ilmu olahraga dan penelitian perubahan perilaku selama puluhan tahun untuk menemukan kembali konsep gym fisik. Didirikan oleh CEO Bodybuilding.com sebelumnya antara lain, Black Box VR menggabungkan konsep game VR dengan mesin resistensi dengan beberapa contoh termasuk mesin kabel yang dapat secara otomatis menyesuaikan untuk memenuhi kriteria berat dan tinggi pemain.

YUR, Inc.

YUR menggunakan komputasi spasial (AR dan VR) untuk membuat individu lebih aktif, terlibat, dan terinformasi. Dari data yang dikumpulkan dari beberapa individu yang menggunakan judul aktif terkenal seperti Beat Saber, YUR telah menemukan bahwa pembakaran kalori dari menggunakan game VR bisa menjadi signifikan. Peran YUR adalah untuk menunjukkan data kesehatan pengguna yang didasarkan pada input VR saja.



Gambar 11-5. Gambar MRI payudara dan legiun (terlentang) sejajar dengan pasien dan dilihat menggunakan Microsoft HoloLens

VR sebagai alat penurunan berat badan yang efektif secara umum tidak disengaja. VR memikat indera dan tubuh cukup untuk membuat pengguna cukup bergerak sehingga penurunan berat badan bisa datang sebagai manfaat tambahan. Ini adalah perubahan

paradigma bersejarah karena kebugaran telah terkenal karena kegagalannya untuk merangsang pikiran. YUR melihat kombinasi sifat menghibur dari permainan dengan manfaat fisik dari olahraga sebagai gerakan nyata menuju gangguan format stereotip kebugaran.

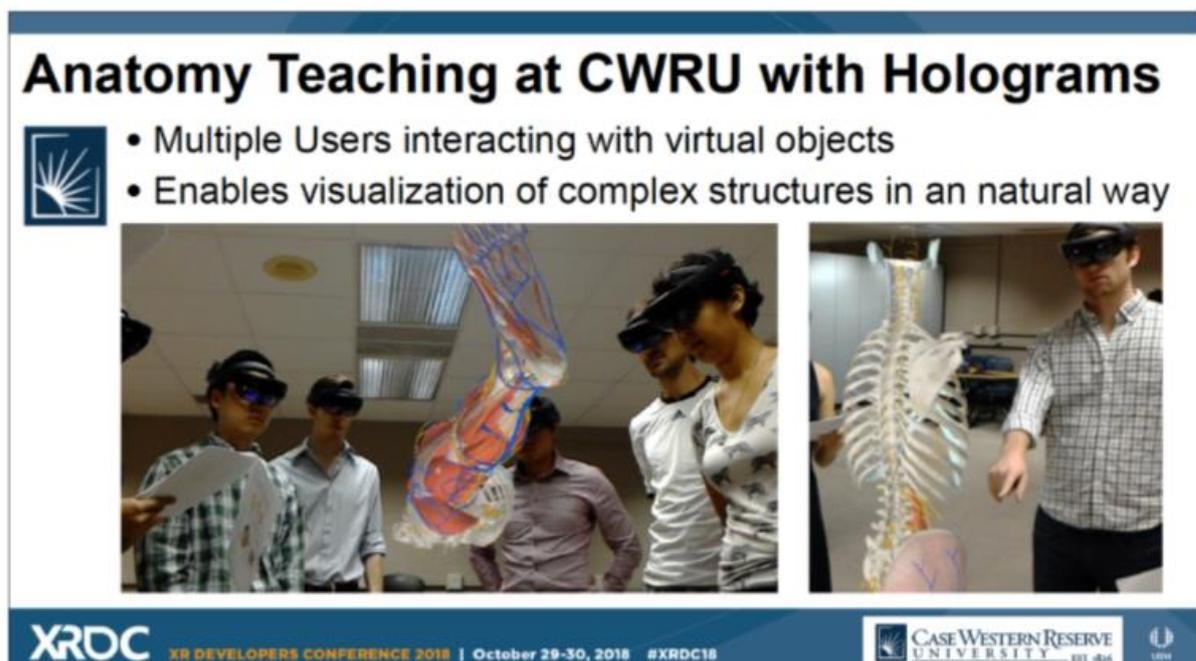
Meskipun institusi akademik yang tercakup di sini hanya mencakup University of California, San Francisco, Stanford, dan Case Western Reserve University, banyak institusi akademik lainnya bekerja untuk mengaktifkan dan memberlakukan solusi untuk tantangan nyata dalam perawatan kesehatan melalui kolaborasi grup AR dan VR.

11.6 STUDI KASUS DARI INSTITUSI AKADEMIK TERKEMUKA

Beberapa aplikasi yang dihasilkan oleh Stanford dan Case Western Reserve University adalah operasi payudara, pendidikan kedokteran menggunakan overlay AR pada pasien, panduan jarum, operasi ortopedi, prosedur otak, dan operasi lainnya.

Di Stanford, satu studi percontohan menggunakan Microsoft HoloLens untuk membuat aplikasi khusus pasien yang menyelaraskan citra MRI yang mengungkapkan lokasi lesi pada payudara pasien untuk menutupi lesi di lokasi sebenarnya, seperti yang diilustrasikan pada Gambar 11-5.

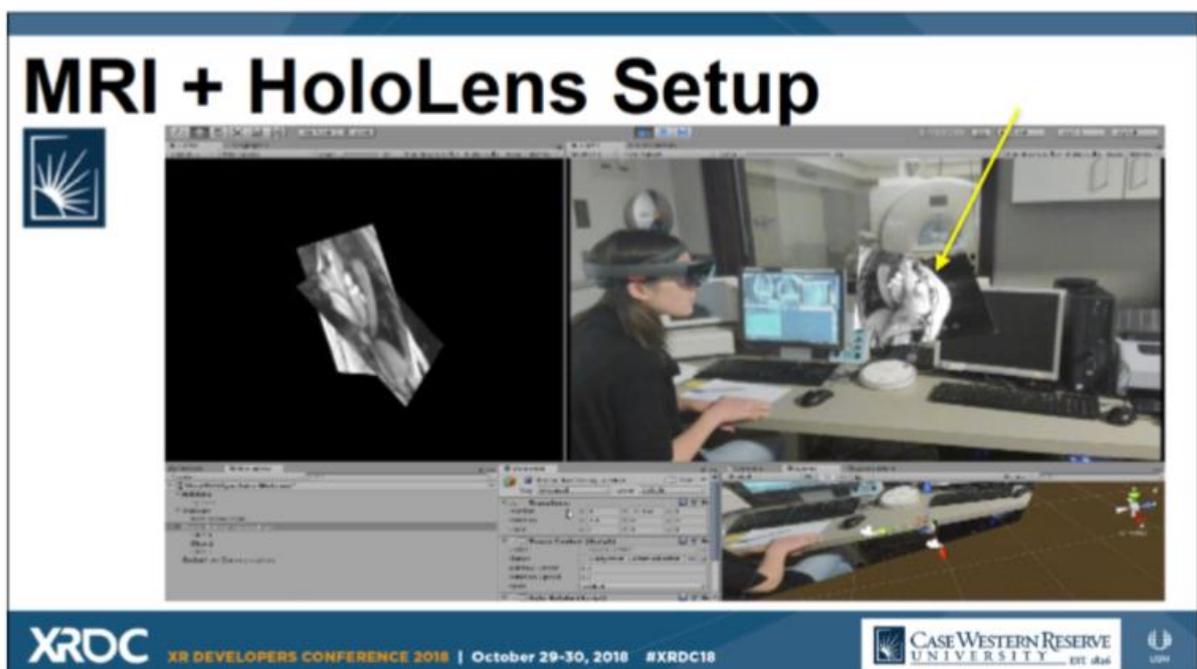
Rangkuman hasil studi tersebut mencakup perbaikan awal dalam semua tindakan; namun, menyelaraskan rendering AR dengan pasien masih merupakan tantangan. Perbaikan penyelarasan di masa depan datang dari bidang-bidang seperti visi komputer dan tracking tanpa penanda. Departemen Radiologi Kedokteran Stanford telah mengilustrasikan kemungkinan ini dengan kamera RealSense Intel. Gambar 11-6 menunjukkan perbandingan aplikasi HoloLens terhadap prosedur standar untuk memperkirakan lokasi tumor yang teraba.



Gambar 11-6. Dalam gambar ini, siswa menggunakan HoloLens dengan tangan mereka untuk berinteraksi dengan model

Pekerjaan yang dilakukan di Case Western Reserve University menggunakan AR dalam konteks pendidikan untuk mengajarkan anatomi dengan memungkinkan banyak siswa berinteraksi dengan model virtual. Untuk struktur yang kompleks, paradigma beberapa orang yang melihat model ini dapat membantu siswa dengan cepat menyelesaikan kesalahpahaman bersama.

Mengambil pemetaan gambar MRI menggunakan AR selangkah lebih maju adalah pipeline Case Western University untuk rendering MRI dan HoloLens real-time (Gambar 11-7). Hal ini memungkinkan penggunaan tampilan HoloLens yang intuitif dari data MRI volumetrik karena diperoleh dengan sedikit menunggu.



Gambar 11-7. Pipa rekonstruksi MRI yang berbasis di Unity (konduktor MRI memakai Microsoft HoloLens dan dari sudut pandang mereka, dimungkinkan untuk melihat data MRI pasien secara real time)

Stanford juga memiliki beberapa aplikasi lagi yang menggunakan HoloLens untuk memanfaatkan overlay AR pada pasien untuk perawatan yang lebih baik, di mana objek yang digunakan dalam standar perawatan rutin dilacak secara posisi. Dua contoh benda ini adalah tongkat ultrasound dan jarum, seperti yang diilustrasikan pada Gambar 11-8. Dalam kedua kasus, seorang praktisi dapat menggunakan data yang dilacak untuk lebih tepat melihat area tubuh atau menempatkan garis, masing-masing.

Mixed-Reality Ultrasound



- HoloLens tracks ultrasound probe
- Real-time reconstruction
- Imaging rendering in-place



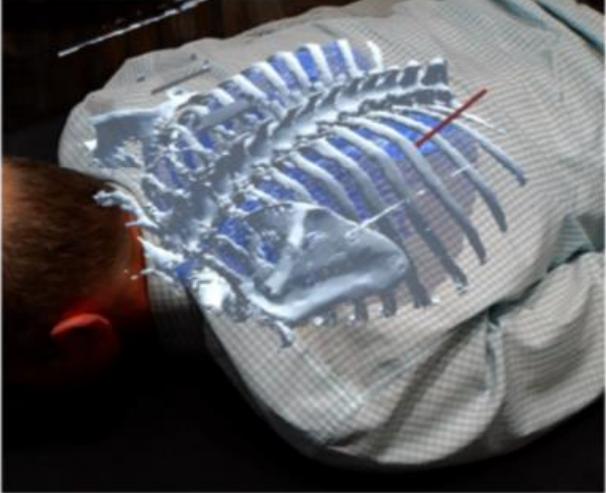

Mixed-Reality Guidance for Medical Procedures - immers.stanford.edu


Gambar 11-8. Seorang praktisi mendapatkan overlay real-time dari data gambar ultrasound di atas lengan pasien (biasanya, praktisi perlu melihat ke layar terpisah tanpa MR)

Lung Resection Surgery



- Lung Cancer: Most deaths of any cancer
- Surgical resection is a common treatment
- Must avoid vessels, resect certain “lobes”

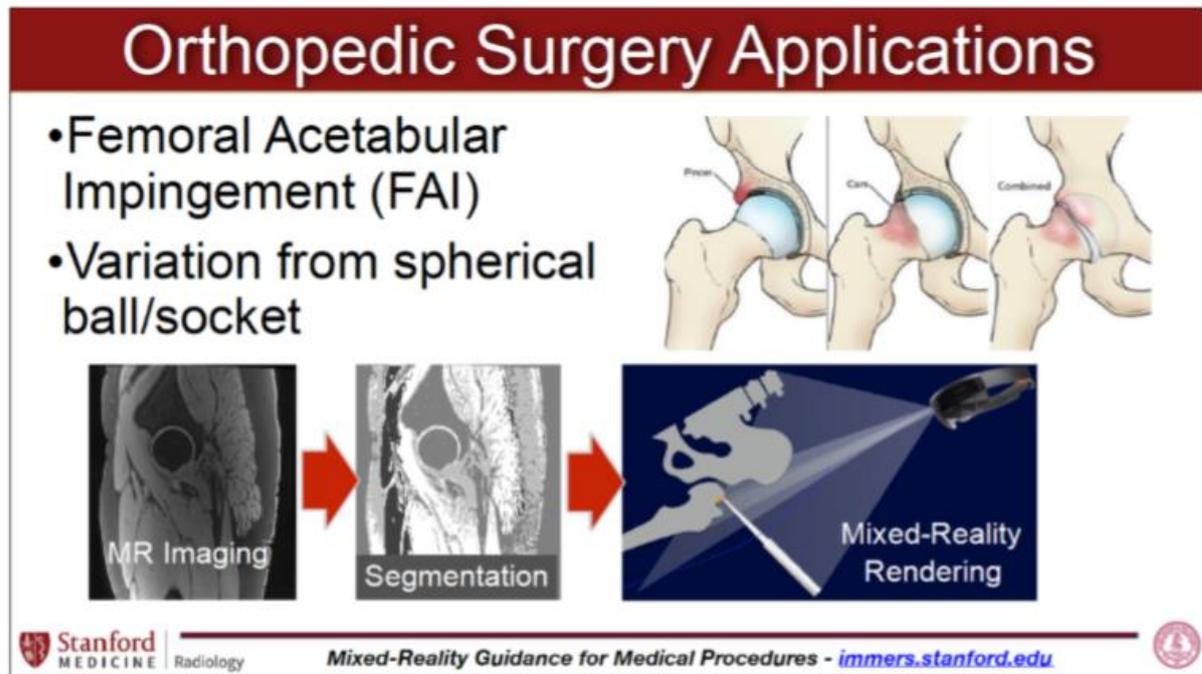



Mixed-Reality Guidance for Medical Procedures - immers.stanford.edu


Gambar 11-9. Di sini, seorang pasien ditambah dengan paru-paru virtual saat dokter mempersiapkan perawatan reseksi paru-paru

Dampak augmentasi di tempat belum benar-benar dijelaskan, tetapi menarik untuk membayangkan bagaimana hal ini dapat mempengaruhi, misalnya, kecepatan dalam

pemberian perawatan atau faktor lainnya. Akankah format tampilan ini meningkatkan kemampuan diagnostik dan mengurangi kesalahan?



Gambar 11-10. Aspek aplikasi bedah ortopedi termasuk representasi permukaan dan model 3D, alat (di dalam tubuh), reseksi, gerakan, dan simulasi pelampiasan virtual

Ada juga aplikasi MR untuk perencanaan dari (lihat Gambar 11-9) Stanford dalam domain bedah termasuk transplantasi ginjal, reseksi paru-paru karena kanker paru-paru, dan bedah ortopedi. Ini biasanya area di mana nick ditangani pembuluh darah atau "lobus" tertentu dapat menjadi masalah, dan penggunaan MR mungkin memberi dokter UI baru, lebih modern, lebih membantu untuk masing-masing tugas perencanaan di mana pasien sebenarnya ditambah. .

Upaya yang sedang berlangsung di Case Western Reserve University dan Stanford mungkin memimpin institusi pendidikan tinggi medis dalam melengkapi fakultas dan mahasiswa dengan teknologi VR dan AR. Contoh terakhir, ditunjukkan pada Gambar 11-10, yang diberikan oleh Stanford melibatkan penggunaan MR dalam berbagai konteks ortopedi.

Sebagai penutup, aplikasi teknologi kesehatan VR dan AR dan ruang masalah mencakup perencanaan dan bimbingan, pendidikan kedokteran, terapi yang digunakan dalam pengaturan klinis, dan kesehatan proaktif, antara lain. Aplikasi ini menyatukan tim peneliti, pemrogram game, seniman, dan dokter—dan memungkinkan terobosan yang berpotensi bermakna bagi penyandang disabilitas motorik, seperti yang ditunjukkan dalam tutorial kode Insight Parkinson di bab ini. Untuk proyek Insight Parkinson, VR menyediakan sarana untuk mengukur tes sentuhan jari-hidung analog. Dalam tinjauan, penggunaan VR dan AR dalam teknologi kesehatan akan berubah seiring perkembangan teknologi dan keterjangkauan dan kematangan perangkat keras meningkat.

BAB 12

PENGALAMAN PENGGEMAR: SPORTSXR

12.1 PENGANTAR

Ini benar-benar waktu yang luar biasa untuk menjadi penggemar olahraga, dan berkat teknologi, masa depan olahraga tidak terbatas. Bab ini berfokus pada *augmented reality* (AR), *virtual reality* (VR), dan olahraga. Hubungan yang kita miliki sebagai penggemar dengan olahraga telah mendorong perkembangan media dan teknologi selama beberapa tahun terakhir dengan kecepatan yang ganas. Olahraga telah menjadi salah satu kategori konten yang paling banyak dikonsumsi di pasar media digital global dan memajukan teknologi untuk lebih banyak pengalaman olahraga AR dan VR. Berikut adalah aturan dasar yang perlu diketahui pengembang:

- Olahraga adalah acara di mana aturan ditetapkan, kontestan bersaing, dan ada hasilnya.
- AR dan VR menggunakan teknologi untuk membuat dan menyempurnakan konten. Contoh terbaik dari ini adalah "Garis Pertama dan Sepuluh" dari Sportvision pada tahun 1998.
- Tindakan langsung itu penting, itu menciptakan rasa heran,antisipasi, keinginan untuk tidak ketinggalan.

Untuk mengeksplorasi raksasa ini lebih teliti, bab ini dipisahkan menjadi tiga bagian:

- Lima prinsip utama AR dan VR untuk olahraga
- Evolusi berikutnya dari pengalaman olahraga
- Membuat masa depan

Pertama, perkenalan yang tepat. Saya Marc Rowley, dan saya menganggap diri saya sebagai pelopor AR/VR, telah bekerja di AR langsung dalam olahraga selama lebih dari 20 tahun. Saya memiliki lima penghargaan Emmy, beberapa penghargaan inovasi global, dan saya telah mendirikan beberapa perusahaan AR. Seperti yang saya lihat, momen terbaik bagi seorang pendongeng adalah melihat reaksi penonton ketika Anda menunjukkan sesuatu yang belum pernah mereka lihat.

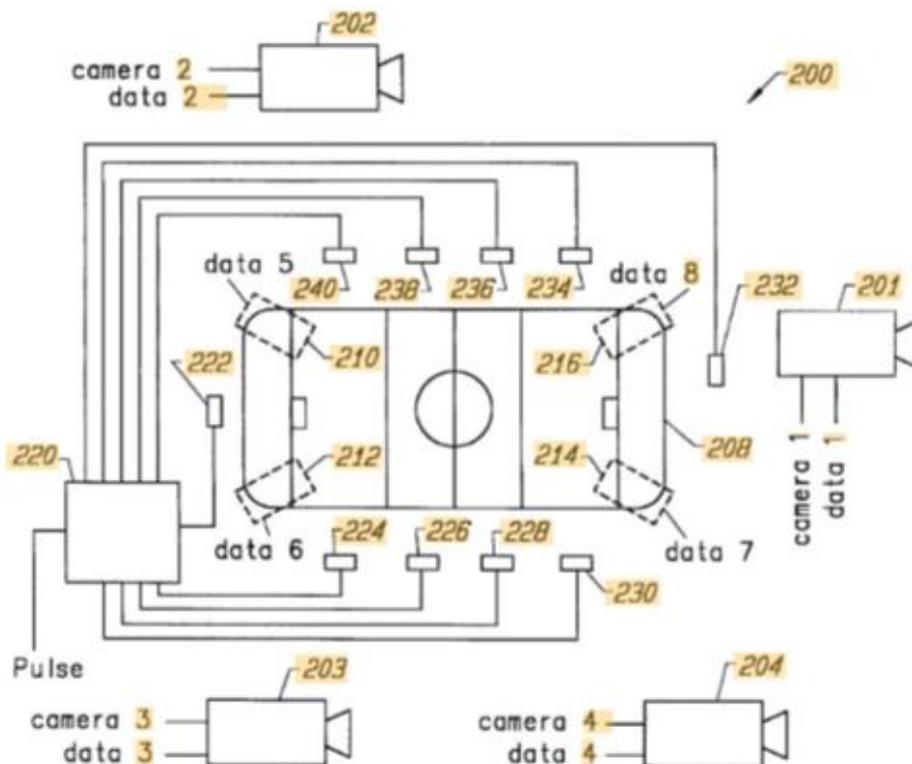
Tujuan saat Anda membuat produk untuk penggemar olahraga adalah membuat sesuatu yang ajaib. Jika Anda melakukan ini, orang akan kembali lagi. Garis Pertama dan Sepuluh untuk sepak bola Amerika dan garis offside dalam futbol/sepak bola/sepak bola adalah momen ajaib terbaik dalam olahraga. Mereka menunjukkan sesuatu yang Anda tahu ada di sana, tetapi Anda tidak dapat melihatnya tanpa teknologi.

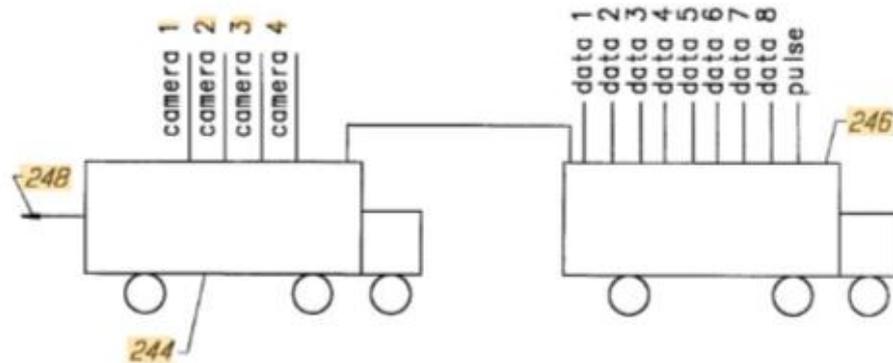
Saya menghabiskan 18 tahun di ESPN menciptakan teknologi baru seperti grafik rundown, buku permainan virtual, dan tiang kamera multiview pertama. Saya meninggalkan posting ini di ESPN pada tahun 2017 untuk mengerjakan gelombang AR dan VR berikutnya. Saat ini, saya adalah CEO Live CGI. Tim saya telah membuat pemutar AR siaran langsung digital lengkap dari acara langsung di CGI (gambar yang dihasilkan komputer) dengan kemampuan

streaming simultan ke semua perangkat. Sekarang, sebelum kita mulai, kita perlu mencapai garis dasar. Penelitian penting.

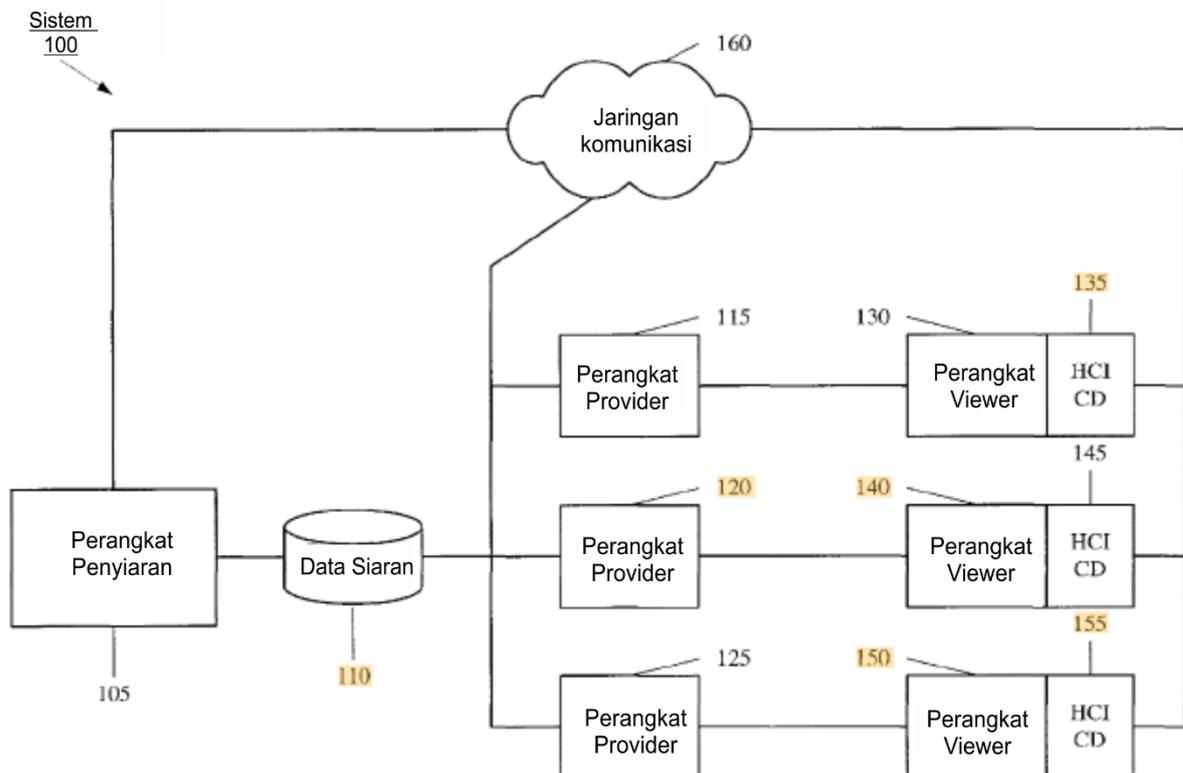
Ya, kami tahu Anda pernah mendengarnya—ketika Anda berurusan dengan produk olahraga, gandakan. Kenyataannya adalah ini, sangat mungkin bahwa seseorang telah memikirkan ide Anda. Mereka mungkin sudah mengerjakannya atau memulainya sebelum Anda. Namun, jika Anda percaya dan dapat membuktikan bahwa masalahnya ada dan solusi Anda istimewa, ada kesibukan luar biasa yang menunggu Anda. Saat Anda membuat dan menghadirkan produk yang mengubah cara orang melihat permainan mereka, Anda akan menjadi yang terbaik. Untungnya, ide-ide terbaik dalam olahraga bersifat publik. Yang perlu Anda lakukan adalah memulai dengan pencarian paten.

Gambar 12-1 dan 12-2 menyajikan dua contoh singkat dari dua paten yang membentuk cara kita menonton siaran langsung olahraga saat ini. Gambar 12-1 adalah paten untuk sistem untuk meningkatkan presentasi televisi dari suatu objek dalam acara olahraga, dan Gambar 12-2 adalah paten untuk menyajikan konten dan menambah siaran. Paten mungkin tampak tidak menarik untuk dibaca pada awalnya, tetapi jika Anda meluangkan waktu, Anda akan menemukan bahwa mereka menciptakan jalan yang jelas menuju masalah dan solusi. Anda tidak perlu menghabiskan waktu berminggu-minggu, tetapi Anda harus menghabiskan setidaknya 40 jam untuk memastikan Anda memiliki dasar yang baik tentang apa yang telah datang sebelum Anda. Ini akan membantu meningkatkan peluang Anda untuk berhasil.





Gambar 12-1. Jalur bagaimana kamera digunakan untuk menemukan posisi relatif terhadap bidang permainan



Gambar 12-2. Alur kerja tentang bagaimana data dapat diproses untuk membuat ulang output pemutaran, yang merupakan kunci untuk memahami bagaimana elemen AR dibuat

Sekarang, mari masuk ke apa yang perlu Anda ketahui saat membuat produk untuk olahraga.

Bagian 1: Lima Prinsip Utama AR dan VR untuk Olahraga

Oke, panggung diatur untuk lima prinsip AR/VR dalam olahraga:

- Momen adalah segalanya
- Tidak ada yang hidup

- Gambar datar/rasa kehadiran
- aturan 20/80
- Waktu penting

Mari kita mulai dengan mengajukan dua pertanyaan. Mengapa olahraga itu istimewa? Apa jadinya olahraga tanpa AR dan VR? Jawaban atas pertanyaan pertama itu mudah—ini adalah momen-momennya; momen adalah segalanya. Olahraga memiliki tiga fase: pra-acara, acara itu sendiri, dan pasca acara. Namun setiap kontes, setiap acara, dapat ditentukan oleh momen-momen kritis. Momen inilah yang paling didambakan orang. Mereka ingin berada di sana secara langsung, berada di saat itu, terbungkus dalam semua kemuliaan, rasa sakit, dan kemenangan. Olahraga adalah salah satu pelarian terbesar dari kenyataan yang pernah ada. Selama beberapa jam Anda dapat meninggalkan dunia Anda dan hidup di dunia yang diciptakan untuk Anda mengalami momen-momen menakjubkan. Sekarang, mari kita bahas pertanyaan kedua.

Tenis onsider, di mana ofisial membiarkan algoritme dan kamera virtual memutuskan apakah bola masuk atau keluar lapangan (Hawkeye). Anda mungkin berpikir tentang siaran sepak bola Amerika di mana ada garis kuning—garis bawah pertama—dilapiskan di layar yang sebenarnya tidak ada di kehidupan nyata. Semua pilihan bagus, tetapi tidak ke mana kita harus pergi. Kita perlu mendekonstruksi esensi olahraga.

Mari kita mulai dengan menghapus tayangan ulang. Tidak ada replay lagi. Itu mengubahnya sedikit, kan? Kemudian, ambil skor langsung dan grafik jam. Sekarang yang Anda miliki hanyalah umpan video dari sebuah kontes dan Anda tidak tahu siapa tim yang bermain dan berapa skornya. Sekarang singkirkan video siaran, audio, dan pembaruan teks langsung.

Yang tersisa adalah arena di mana orang-orang pergi untuk bersaing, dan orang-orang datang untuk menonton. Itu adalah dasar yang kita butuhkan untuk memulai dan di mana kita berada sekitar seratus tahun yang lalu. Semua lapisan yang baru saja kita kupas adalah AR. Faktanya, siaran itu sendiri adalah representasi VR. Ini layak diulang, apa yang Anda lihat tidak nyata. Itu semua adalah presentasi yang dibuat untuk Anda.

Anda mungkin berpikir, tunggu sebentar. Saya dapat melihat olahraga di TV saya tanpa menggunakan headset VR. Ya, Anda bisa, dan apa yang Anda tonton tidak langsung. Ini adalah representasi dari suatu peristiwa. Seorang sutradara menunjukkan sudut kamera, dan di otak Anda, Anda membuat peta arena dan otak Anda menghitung perubahan dan mengisi celah. Itu benar; kamera hanya menunjukkan sekitar 90% dari aksi dan kemudian sutradara membuat otak Anda membuat sisanya dengan potongan cepat dan grafik dan musik berkecepatan tinggi. Ini adalah poin kritis: jika kita tersesat di sini, sisa bab ini akan gagal.

12.2 TIDAK ADA YANG LANGSUNG

Live adalah keyakinan yang kita miliki bahwa apa yang kita proses di otak kita terjadi pada saat kita melihatnya. Faktanya, jika Anda berada di lapangan menonton pertandingan, otak Anda memproses gambar pada 13 milidetik. Ini berarti bahwa apa yang Anda lihat

sebenarnya terjadi 0,0013 detik yang lalu. Tapi bagimu rasanya seperti baru saja terjadi. Bagi Anda itu benar-benar hidup. Tetapi jika kami mencoba menjelaskan hal ini kepada konsumen rata-rata, mereka akan memiliki ekspresi yang sangat bingung di wajah mereka. Dalam olahraga, "live" umumnya diterima memiliki tiga tahap yang berbeda: Live, Live Live, dan Live to record atau live to tape (LTT).

Bagi banyak orang, konten langsung dapat berupa apa saja dalam satu menit setelah aksi terjadi. Misalnya, Anda mungkin menonton konten di rumah di Chromecast, tetapi Anda berbicara dengan seorang teman di permainan dan Anda menemukan ada penundaan 40 detik untuk sampai ke Anda. Itu masih disebut live, tetapi aturannya sedikit ditekuk.

Live adalah untuk konten yang berada di bawah jendela empat hingga lima detik

Diambil sedikit lebih jauh, jika Anda berada di sebuah apartemen di New York City dan menonton pertandingan bisbol di Bronx, Anda mungkin menontonnya dalam penundaan 5 hingga 60 detik tergantung pada layanan Anda. Ini terjadi karena sinyal video memiliki data. Data tersebut perlu ditransmisikan dari kamera ke pengalih produksi dan kemudian ke hub transmisi, yang kemudian mengirimkannya ke area pusat siaran di mana ia dialihkan ke sinyal TV atau ke sinyal internet dan dipecah menjadi paket, dan kemudian dikirim melalui beragam sakelar jaringan untuk dipasang pada perangkat yang Anda gunakan. Inilah sebabnya mengapa orang yang berbeda dapat melihat video yang sama pada waktu yang berbeda.

Live adalah sesuatu yang kami yakini ada dalam olahraga, media berita, dan di banyak bidang lainnya. Ketahuilah bahwa ini semua adalah konstruksi virtual yang kami pilih untuk disetujui secara langsung—dan bahkan itu mungkin akan berubah sedikit.

Satu cerita yang ingin kami ceritakan dari akhir 1990-an terjadi ketika data otomatis mulai ditampilkan di layar TV dan dapat diakses melalui internet. Konsumen mulai memperhatikan bahwa skor akan berubah sebelum mereka melihat video. Dalam satu contoh, seorang penjudi yang marah menelepon penyiar besar untuk mengetahui bagaimana mereka dapat memprediksi skor dengan sangat akurat. Yang lucu adalah semuanya bermuara pada matematika. File video lebih besar dan membutuhkan waktu lebih lama untuk dikirim. File data seperti jam dan skor lebih kecil dan memakan waktu lebih sedikit. Dengan demikian, Anda mungkin melihat pembaruan papan skor digital sebelum video. Sebagian besar penyiar besar sekarang memiliki kode untuk menangani ini, tetapi pada saat itu adalah kejutan besar. Ini sebagian besar telah hilang karena latensi dalam video telah berkurang, tetapi di beberapa tempat, Anda masih dapat melihatnya. Hidup adalah bagaimana Anda melihatnya. Mari kita lanjutkan dengan contoh olahraga langsung.

Di seluruh dunia, penyiar menempatkan kamera di lokasi untuk menangkap gambar datar dari suatu peristiwa yang diputar di ruang tiga dimensi. Mereka kemudian mengambil beberapa kamera dan memotongnya untuk memberikan persepsi kedalaman, waktu, dan ruang kepada orang-orang—penyiar menciptakan Virtual Reality yang diterima sebagian besar dunia sebagai peristiwa nyata. Namun, dalam bisnis kita tahu bahwa metode ini memiliki keterbatasan.

Sebuah kamera hanya sebaik lokasi di mana ia ditempatkan, titik fokus, iris, bayangan, pencahayaan, kekuatan sinyal, piksel kembali ke ujung kepala produksi, dan, yang paling

penting, tim yang merakit ini perwakilan. Tapi jangan salah, Anda sedang menonton representasi datar dari dunia nyata.

Menghasilkan acara olahraga datar membutuhkan beberapa komponen. Operator kamera, pengalih, audio, operasi, kabel, dan banyak lagi. Mereka semua bekerja untuk membuat representasi virtual dari sebuah game dengan berbagai dimensi di layar datar Anda. Selama 80 tahun terakhir, beginilah cara orang menonton olahraga. Dan seperti halnya industri mana pun saat dewasa, orang menemukan cara untuk menambahkan elemen dan meningkatkan. Seperti halnya kategori media lainnya, biasanya ada dua tingkat momen kritis yang berbeda dalam evolusi kategori tersebut. Olahraga tidak terkecuali.

Berikut adalah daftar lima fitur penceritaan AR dan VR olahraga teratas dalam 50 hingga 60 tahun pertama siaran langsung olahraga:

1. Transmisi langsung
2. Audio penyiar
3. Video
4. Tayangan ulang
5. Skor grafis

Dan berikut adalah daftar lima fitur AR dan VR olahraga berikutnya dalam 20 hingga 30 tahun terakhir:

1. Grafik mendongeng langsung di layar
2. Sorotan 8. Grafik aturan yang ditambah
3. Media streaming langsung ke perangkat internet
4. Interaksi media sosial

Masing-masing elemen ini telah membantu menyusun cara pengguna melihat dan mengalami olahraga. Semua dengan satu tujuan tunggal: memberikan konsumen rasa kehadiran untuk mewujudkan rasa heran, takjub, danantisipasi semua untuk satu momen dalam olahraga yang penting, titik kritis di mana hasilnya diragukan dan konsumen mencondongkan tubuh ke depan menginginkan lebih.

Pikirkan tentang itu, bagaimana jika tidak ada sorotan, bagaimana Anda bisa menonton pertunjukan, untuk mendapatkan rekap cepat. Bagaimana jika Anda tidak dapat melihat grafik di layar dan bagaimana jika wasit harus membuat semua keputusan dalam tenis? Bagaimana jika Anda tidak dapat melihat garis Pertama dan Sepuluh dalam pertandingan sepak bola? Bagaimana jika Anda tidak dapat melihat pesan sosial instan dari pelatih atau pemain favorit Anda? Masing-masing elemen ini menambah pengalaman olahraga kami. Tapi apa yang akan terjadi selanjutnya yang akan benar-benar revolusioner. Di akhir tahun 2000-an, saya bekerja di sebuah tim di ESPN yang ingin menggunakan teknologi First-and-Ten AR dan menggunakannya untuk meningkatkan penceritaan dengan lebih baik agar pemirsa menonton lebih lama. Itulah yang dimaksud dengan bisnis penyiaran. Anda berada dalam bisnis untuk menghasilkan uang sebanyak yang Anda bisa selama Anda bisa. Dalam olahraga, Anda menghasilkan uang dengan melakukan transaksi dengan pelanggan. Baik secara langsung dari dompetnya kepada Anda atau secara tidak langsung dengan menjual perhatiannya kepada mitra (pengiklan). Kita dapat melakukan ini melalui tayangan, pemirsa,

dan cara lainnya. Apa yang menjadi terkenal selama bertahun-tahun adalah satu metrik: waktu yang dihabiskan.

Waktu yang dihabiskan adalah faktor yang memotivasi banyak pasangan. Artinya konsumen tertarik pada produk dan cukup tertarik untuk memberikan waktu yang signifikan. Ini merangkum klik, bola mata, gerakan, dan segalanya. Salah satu eksekutif lama saya memiliki pepatah yang menjadi agak benar. Konsumsi konten turun ke aturan 20/80. Aturan 20/80 berjalan seperti ini. Jika Anda mengambil total audiens pengguna yang mengonsumsi dan menghabiskan waktu dengan produk Anda (konten/platform), 20% pengguna akan menjadi pengguna terberat Anda. Mereka akan membuat 80% dari tindakan Anda. 80% lainnya dari pengguna Anda akan menjadi 20% dari konsumsi Anda. Ini terjadi setelah audiens Anda matang. Pada bulan-bulan pertama produk Anda, itu mungkin menjadi gila atau mungkin lambat, apa pun cara Anda berolahraga, aturan 20/80 telah terbukti menjadi metrik yang baik.

Tujuan saya adalah selalu membuat 20% itu tertarik dan melihat apakah saya bisa memindahkannya ke 21% atau 22%. Alasannya adalah Anda tidak akan membuat konsumen biasa membuat perubahan besar dalam hidup. Itu tidak terjadi, tetapi jika Anda bisa membuat konsumen hardcore berubah, Anda akan melihat peningkatan pada konsumen biasa. Ini tidak berarti bahwa Anda selalu menangkap seluruh penonton. Tengok saja kebangkitan Esports. Konsumen ada di sana dan mereka terlayani, sekarang ketika konsumsi mereka tumbuh, kami melihat pola yang serupa. Hanya saja, pola Esports berkembang dengan kecepatan yang jauh lebih cepat saat konsumen beralih dari judul game ke judul game mencari hal terbaik berikutnya.

Apa hubungannya dengan AR dan VR? Semuanya. Ini adalah dasar untuk Anda saat Anda menguji, saat Anda membangun, dan saat Anda mengirimkan produk pertama yang layak. Setiap orang memiliki jumlah waktu yang sama dalam sehari. Bagaimana orang menggunakan waktu mereka adalah satu-satunya mata uang yang penting. Tidak masalah apakah Anda membuat aplikasi untuk anak-anak di rumah sakit atau memasang Super Bowl, waktu adalah tujuan Anda. Jika Anda membuat pengguna menghabiskan waktu, Anda dapat membuat dampak sosial, keuangan, atau pendidikan—atau bahkan semuanya. Waktu penting. Sekarang kita memiliki seperangkat aturan dasar, jadi mari kita rekap lima poin prinsip utama:

1. Momen adalah segalanya
2. Tidak ada yang hidup
3. Gambar datar/rasa kehadiran
4. aturan 20/80
5. Waktu penting

Bagian 2: Evolusi Selanjutnya dari Pengalaman Olahraga

Dalam beberapa tahun ke depan, kita akan melihat produk yang menyerang masing-masing dari lima poin ini; mereka akan menjadi pos panduan saat pengalaman olahraga didefinisikan ulang. Perubahan ini akan terjadi pada skala global yang sebelumnya dianggap mustahil hanya karena menjamurnya perangkat internet dan minimnya kurva pembelajaran

bagi konsumen baru. Perubahan akan berlangsung cepat dan akan fokus pada tiga bidang utama:

- Koneksi
- Menampilkan
- Interaksi

Tingkat koneksi di dunia telah berubah dengan cepat seperti halnya sistem kompresi untuk mengirimkan data. Dulu, 3G adalah ciri khasnya, sekarang digantikan dengan 4G, 5G, LTE, dan setiap peningkatan lainnya yang keluar. Apa artinya ini pada tingkat dasar adalah bahwa jumlah data yang dapat Anda kendarai menjadi lebih besar dan lebih cepat, dan ini kemungkinan akan terus tumbuh pada tingkat eksponensial, yang akan mendorong pertumbuhan pengalaman AR dan VR. Data berhubungan langsung dengan latensi. Latensi adalah penundaan yang dialami konsumen dalam tindakan langsung ke dunia nyata. Itulah yang menciptakan efek langsung. Sebagai pengembang, produsen atau distributor, Anda perlu mengikuti dan memahami hubungan antara kedua faktor tersebut. Latensi penting.

Misalnya, pengujian pertama kamera VR unit tunggal skala besar pada acara olahraga yang digantung di atas lapangan berjalan dengan kecepatan 20 hingga 30 mil per jam pada kabel yang digantung di udara. Kamera mengirimkan 9.000.000.000 bit data per detik melalui saluran serat ke konverter yang mengirim 20.000.000 bit data ke pengalih yang mengubah data itu menjadi 10.000.000 bit data. Akhirnya, itu dicampur kembali ke 20.000.000 bit dan dikirim ke konsumen. Yaitu 9 gigs, hingga 20 MB, hingga 10 MB dan mencadangkan data hingga 20 MB per detik. Alasan ditulis dengan semua nol adalah untuk memberikan cakupan penuh. 9.000.000.000 bit data setiap detik sangat gila, tidak ada perangkat seluler yang dapat menanganinya hari ini—tetapi besok mungkin.

Ini membawa kita ke inovator di perusahaan kecil dan besar yang bekerja untuk memecahkan masalah kompresi dan kecepatan sinyal untuk menciptakan pengalaman AR dan VR yang berjalan di layar baru. Headset VR lengkap, kacamata AR, dan perangkat lain semuanya membutuhkan umpan data yang masuk ke dalamnya. Dan seperti yang telah kita bahas sebelumnya, kita semua membutuhkan koneksi cepat agar konten dapat berfungsi dengan olahraga. Hidup adalah yang terpenting.

Ketika datang ke tampilan, masa depan akan dipetakan oleh prosesor grafis dan kemudian perangkat keras optik. Beberapa orang mungkin berpikir itu adalah perangkat keras visual yang pertama; namun, Apple dan Google mengubahnya dengan meluncurkan ARKit dan ARCore. Unity, Unreal, dan lainnya telah menciptakan kerangka kerja untuk membuat pengalaman yang luar biasa. Faktor-faktor ini telah membanjiri pasar dengan perangkat VR dan AR. Sekarang, sebagai pengembang, Anda tidak perlu menunggu perangkat keras berkembang biak: sudah ada satu miliar perangkat yang siap. Konsumen sudah memiliki perangkat keras.

Prosesor grafis biasanya disebut sebagai GPU. Mereka adalah mesin yang akan membuka AR dan VR langsung dalam beberapa tahun ke depan seiring dengan peningkatan

kecepatan koneksi. GPU mengambil data grafis dan membuat gambar. Kecepatan dan kemampuan GPU adalah pasar kedua Anda untuk pertumbuhan.

GPU telah tumbuh secara eksponensial karena berbagai faktor yang konsisten dalam teknologi tampilan, dengan satu pengecualian utama: ledakan Bitcoin. Peningkatan “penambangan” Bitcoin telah mendorong kecepatan prosesor ke tingkat yang sangat tinggi karena para penambang bersaing untuk mendapatkan lebih banyak Bitcoin. Ini memberikan dorongan tak terduga untuk pekerjaan VR dan AR. Ini meningkatkan kekuatan untuk menciptakan pengalaman yang benar-benar imersif dengan mendorong kecepatan prosesor. Perubahan terbaru lainnya adalah kemajuan dalam ray tracing.

Semua faktor teknologi ini hebat, tetapi semuanya bukan apa-apa tanpa cerita. Cerita adalah segalanya. Orang tidak membeli teknologi, mereka membeli cerita, mereka membeli barang, dan menggunakan barang yang menceritakan kisah mereka. Pelajaran tertua dalam pengalaman massal adalah menceritakan sebuah kisah.

Evolusi ketiga dari pengalaman olahraga akan mengubah cara konten dibuat. Saat ini, anak-anak tumbuh dengan kontrol khusus dalam bermain game, mereka mendapatkan umpan dari media sosial yang dipersonalisasi untuk mereka. Namun ketika mereka menonton siaran, mereka memiliki satu produser, satu sutradara dan beberapa kamera diumpankan kepada mereka. Ini akan berubah ketika pemegang hak mendorong pembuat dan distributor untuk memberi pengguna lebih banyak kontrol. Harinya akan tiba ketika seorang sutradara yang digerakkan oleh kecerdasan buatan secara otomatis memotong urutan untuk menceritakan kisah langsung dalam paket yang diproduksi kepada setiap konsumen individu dari game yang sama.

Perubahan besar pertama untuk ini adalah Esports, di mana pengguna dapat berkomentar, berinteraksi, dan mengontrol pengalaman mereka. Dinding antara penggemar dan atlet runtuh. Meskipun penyiar tradisional menolak keras pada awalnya, mereka perlahan-lahan muncul ketika angka pendapatan menunjukkan kenyataan bahwa kita semua tahu betul: konsumen selalu menang.

Ini adalah kumpulan data penting untuk dipertimbangkan saat Anda membuat produk. Sudah terbukti berkali-kali bahwa jika Anda mencoba menghentikan laju konsumsi konten, ruang hampa dapat dan kemungkinan besar akan tercipta. Dalam hal ini, ketika perusahaan media lama terhenti pada interaktivitas Esports, mereka mengizinkan Twitch.tv untuk menangkap audiens yang besar.

Pikirkan tentang itu. Tidak ada yang menghentikan Sky Sports, ESPN, atau FOX Sports, atau entitas lain mana pun untuk membuat Twitch; mereka baru saja melewatkannya, dan sekarang Twitch.tv adalah bagian dari Amazon dan menarik banyak pemirsa global.

Meskipun Twitch.tv memiliki banyak jenis konten, dua yang menonjol di sini adalah streamer yang bermain game dan streaming properti Esports di platform mereka. Menurut Pengamat Esports pada tahun 2018, empat entitas streaming Twitch teratas menyumbang hampir 500 juta jam. Streamers Ninja dan Shroud yang dipasangkan dengan Riot Games dan The Overwatch League menetapkan standar yang tinggi. Ketika konsumen menonton di platform ini, mereka dapat berinteraksi dengan konsumen lain melalui bilah obrolan dan

mereka juga dapat mendukung tim dan pemain mereka. Tolong jangan pernah melupakan aturan ini: konsumen selalu menang, dan mulai sekarang, ketahuilah bahwa semua konsumen menginginkan kendali.

12.3 BAGIAN 3: MEMBUAT MASA DEPAN

Masa depan dibangun di atas inovasi masa lalu. Masa depan AR dan VR olahraga akan datang dengan kecepatan yang sangat cepat. Berkat proliferasi perangkat, fokus pada latensi, dan perubahan perilaku konsumen, tidak perlu 80 tahun untuk perubahan kuantum berikutnya.

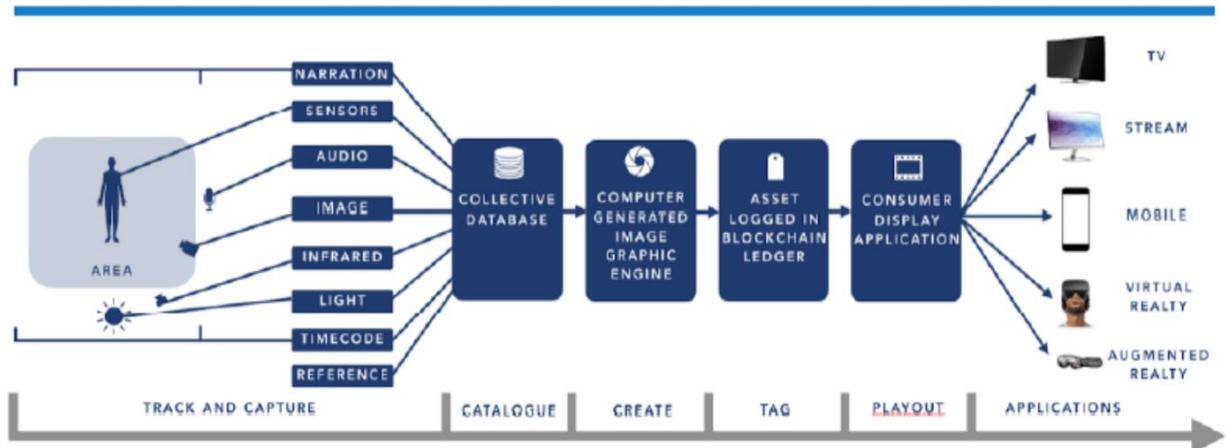
Waktu untuk mendefinisikan kembali olahraga adalah sekarang, waktu untuk mengubah AR dan VR adalah sekarang. Pengembang yang membangun produk yang cepat, bercerita, dan memberikan kontrol audiens akan menciptakan produk yang berhasil.

Ketika Anda duduk untuk membangun masa depan ini, Anda harus memikirkan alur kerja Anda. Bagaimana konten dimasukkan, ke mana perginya? Apa tampilan makro keseluruhan dan apa tampilan mikro? Untuk melakukan ini, Anda perlu membuat dua dokumen untuk tim Anda. Salah satu yang hanya menguraikan apa yang akan Anda lakukan. Dalam istilah olahraga, ini adalah rencana permainan. Setiap pelatih memiliki rencana permainan. Dalam sepak bola, Anda memutuskan apakah Anda menjalankan Information atau shotgun; dalam bisbol, ini adalah pergeseran; di Esports, berapa lama Anda hutan atau siapa yang mendukung Anda.

Kemudian, Anda perlu membuat tampilan mikro. Ini adalah naskah dramanya: ke mana setiap aset akan pergi; siapa yang akan memblokir siapa; siapa yang akan mengambil peran apa? Setelah Anda melakukan ini, Anda dapat melihat bagaimana produk Anda akan bekerja. Mari kita lihat paten terbaru yang diajukan untuk sistem transmisi acara langsung melalui gambar yang dihasilkan komputer. Gambar 12-3 menyajikan ikhtisar, pendekatan yang lebih sederhana. Gambar 12-4 menawarkan rincian lebih lanjut. Dalam kebanyakan kasus, Anda tidak ingin membagikannya, karena itu adalah bagian dari saus khusus Anda untuk bagaimana produk Anda dibuat; namun, setelah Anda mengajukan paten, dan informasinya bersifat publik, ini adalah permainan yang adil bagi siapa pun. Jadi, seperti halnya segalanya, semuanya bermuara pada eksekusi. Hal terbaik tentang olahraga adalah bahwa lapangan bermainnya terbuka untuk umum, tidak demikian dengan semua produk. Dalam olahraga, seperti kata pepatah, "Anda adalah apa yang dikatakan rekor Anda."



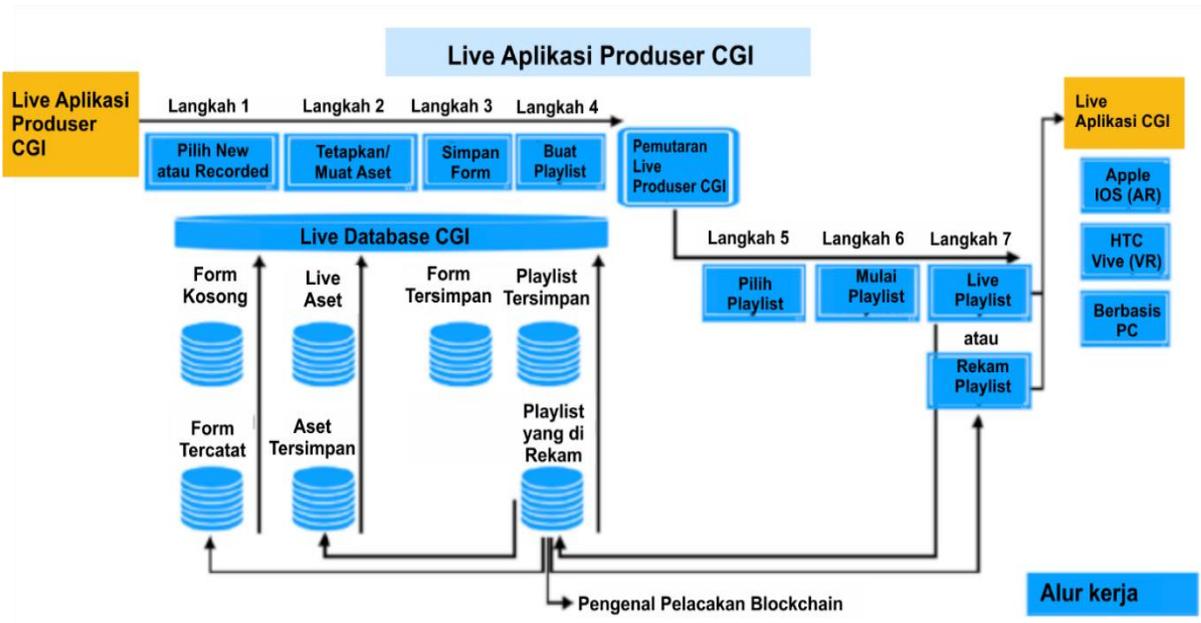
Sistem alur kerja untuk menangkap dan streaming langsung ke beberapa perangkat.



Gambar 12-3. Alur kerja untuk menangkap tujuh kumpulan data langsung, menambahkan kumpulan data yang direkam, dan merampingkan semuanya ke dalam proses presentasi visual—dengan mendorong semua titik data ke dalam mesin CGI, narasi diubah menjadi asli dari kemampuan rendering individual masing-masing platform

Sekarang ke rincian. Faktor pertama dan terpenting dalam olahraga adalah fokus pada latensi; kecepatan adalah fitur utama. Gesekan merusak latensi, perangkat keras yang rumit merusak latensi, dan sistem yang berbelit-belit merusak latensi—fokus untuk menjadi cepat. Ini juga berarti bahwa Anda harus akurat. Tidak ada gunanya bagi siapa pun untuk menjadi cepat dan salah.

Selanjutnya, ceritakan sebuah cerita. Semua pengguna ingin dihibur. Ceritakan kisah tentang peristiwa tersebut dan ceritakan kisah yang dipedulikan orang. Bahkan jika ceritanya hanya tentang pengguna, hanya tentang tim mereka. Itulah cerita yang mereka inginkan, dan dengan memberikannya kepada mereka, mereka akan kembali lagi. Sebagai pengembang, Anda tidak perlu menjadi pendongeng yang hebat, tetapi Anda perlu berbicara dengannya. Anda perlu menemukan pendongeng, penulis, seseorang dengan bakat mengobrol, dan memastikan bahwa produk Anda menceritakan sebuah kisah.



Gambar 12-4. Langkah-langkah individual yang dilakukan produser untuk membuat streaming langsung (saat Anda membuat produk, langkah ini adalah salah satu yang paling penting; Bagaimana setiap klik terjadi? Apa yang dipicu oleh setiap sentuhan? Memikirkan proses Anda adalah kuncinya)

Panduan terakhir adalah memberikan kontrol kepada konsumen. Berikan konsumen apa yang mereka inginkan dan biarkan preferensi, kesukaan, dan ketidaksukaan mereka menciptakan pengalaman. Jadi, mari kita rekap pedoman ini:

- Masalah latensi
- Ceritakan padaku sebuah cerita
- Konsumen menginginkan kontrol

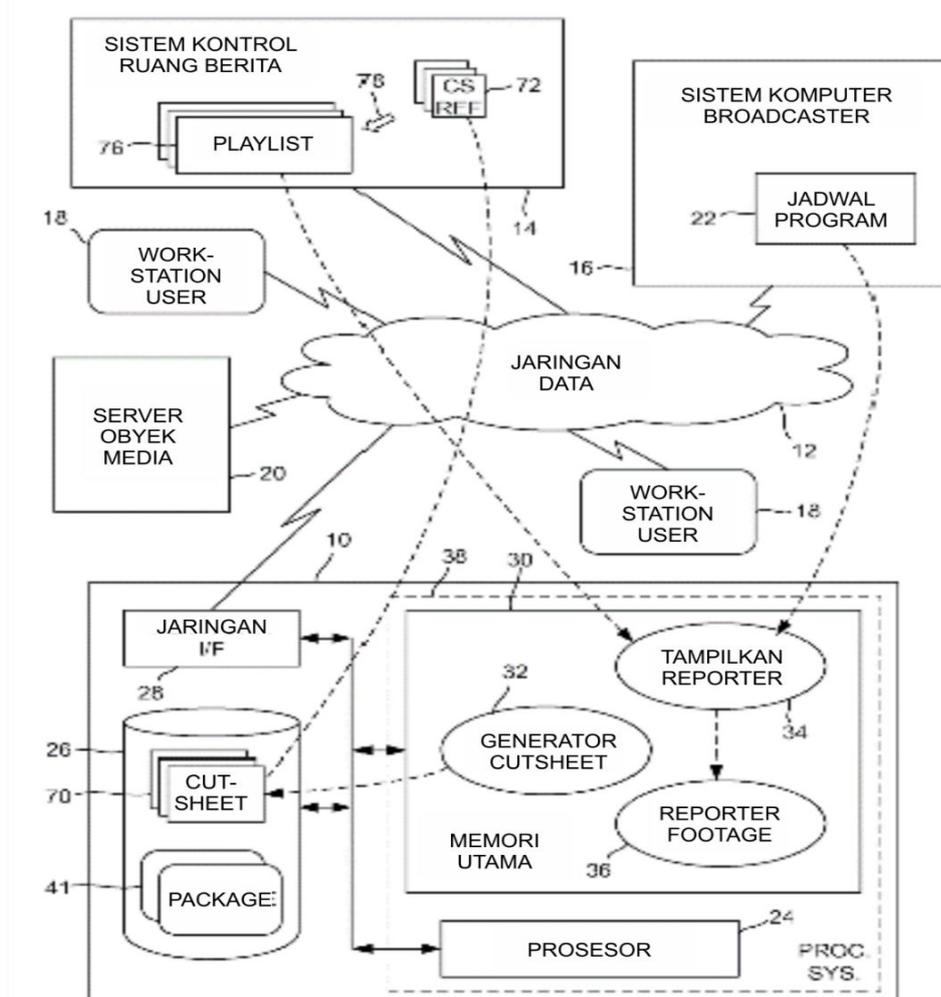
Anda mungkin bertanya, “Oke, tapi bagaimana Anda mengujinya?” Nah, Anda dapat menguji latensi dengan matematika sederhana. Berapa lama perjalanan dari A ke B? Anda dapat menguji kesinambungan cerita dengan menanyakan kepada konsumen apa yang mereka dapatkan dari produk Anda. Dan Anda dapat mengukur seberapa besar mereka menyukai kontrol yang Anda berikan dengan memeriksa gerakan mereka. Anda dapat mengukur semua hal ini, tetapi dari mana Anda memulainya? Untuk tim saya, selalu menjadi aturan 10. Kami tidak memasukkan apa pun ke dalam produk yang ingin digunakan atau dilihat pengguna kurang dari 10 kali. Sekali atau dua kali adalah hal baru; tiga sampai empat kali, Anda mungkin menunjukkan teman. Lima hingga sembilan kali, dan Anda menyukainya tetapi tidak membutuhkannya. 10 kali atau lebih dan Anda harus memilikinya.

Contoh bagus dari hal ini terjadi dalam pertandingan sepak bola pada tahun 2008. New Orleans Saints memiliki pemain listrik bernama Reggie Bush, dan produk baru yang diluncurkan untuk permainan tersebut menunjukkan kecepatannya. Mereka menangkapnya dengan kecepatan 22 mil per jam (mph). Meskipun ini luar biasa mengesankan, penonton menghitungnya dengan mobil dan 22 mph tidak terdengar mengesankan. Namun, jika Anda

memikirkannya, 22 mph berubah menjadi 32 kaki per detik, atau 10 yard per detik. Itu tidak masalah; nomor tidak beresonansi dengan penonton. Pikirkan tentang ini ketika Anda sedang membangun produk Anda, apakah itu akan beresonansi, apakah orang-orang peduli, apakah itu berdampak?

12.4 KEPEMILIKAN

Sebelum kita mengakhiri bab ini, kita perlu berbicara tentang uang dalam olahraga. Ada area licin dalam olahraga di mana orang mengira liga besar menjual permainan atau acara kepada pemegang tiket. Meskipun ada beberapa pendapatan yang diperoleh dari titik penjualan individu itu, sebagian besar pendapatan selama lebih dari 50 tahun terakhir telah melalui hak media, lisensi, dan bundel konten. Ini berarti bahwa liga kurang tentang olahraga dan lebih banyak tentang kekayaan intelektual dan konten. Namun, di muka itu, banyak orang mungkin memperdebatkan hal ini. Namun, ketika Anda melihat kontrak liga, melihat apa yang dibeli oleh perusahaan media, dan melihat nilainya, jawabannya jelas. Olahraga adalah tentang konten. Perubahan ini telah mempengaruhi bagaimana produk dibuat dan menciptakan arus besar uang tunai ke pasar.



Gambar 12-5. Alur kerja tentang bagaimana sistem siaran menggunakan data yang diberi tag dan cutsheet untuk mengaitkan data dengan video

Misalnya, jika Anda membuat penampil interaktif untuk rugby, Anda perlu melacak menit, jam, dan bagaimana konten digunakan. Ini adalah kunci bagaimana liga dan pemain memonetisasi kinerja mereka. Ini berlaku di Esports dan juga biliar. Penggerak bisnis menyeluruh untuk transaksi bernilai miliaran rupiah adalah waktu.

Ya, Anda perlu memikirkan latensi, Anda perlu menceritakan sebuah kisah, dan Anda harus melepaskan kendali. Dan, Anda perlu melacak semuanya untuk dapat memberikan akuntansi pengalaman. Alasan mengapa ini dicantumkan terakhir di bagian ini adalah untuk menegaskan betapa pentingnya hal itu. Jika Anda membangun segala sesuatu yang luar biasa tetapi kehilangan perhitungan dasar dari pengalaman, Anda membatasi kemungkinan keberhasilan serta potensi pertumbuhan Anda. Anda harus yakin untuk membangun pelacak. Gambar 12-5 dan 12-6 menunjukkan dua cuplikan alur kerja dari paten publik untuk pelaporan rekaman (US20110008018A1). Di dalamnya, Anda dapat melihat bagaimana tim menguraikan mekanisme tracking sederhana untuk menangkap rekaman saat sedang dibuat dan kemudian mengkatalogkannya sebelum dikirim melalui aliran untuk konsumen. Sistem sederhana ini mendefinisikan ulang bagaimana kesepakatan hak rekaman dinegosiasikan di akhir tahun 2000-an. Tidak perlu ribet, cukup melacak apa yang dikonsumsi dan bagaimana caranya.

44

46

48

Tampilkan Data

Nama paket

Format Audio: VO Total Waktu Berjalan: 01:17

50 52 54

Detail Lembar Potong Detail Rekaman Catatan

Waktu tang diperhitungkan: 01:15

Deskripsi Shot	Waktu	Tanggal Acara	Topik	Event	Pemegang hak	Kesopanan
menunjukkan elemen	00:00	6/24/2008	<input type="button" value="v"/>	<input type="button" value="v"/>	<input type="button" value="v"/>	
sorotan olahraga	00:04	6/24/2008	<input type="button" value="v"/>	<input type="button" value="v"/>	<input type="button" value="v"/>	
			<input type="button" value="v"/>	<input type="button" value="v"/>	<input type="button" value="v"/>	

58 60 62

Perubahan Shot Terakhir @: 01:14

Keluaran @: 01:17

Pembekuan Video @: 01:20

Jumlah Video yang Tersedia @: 01:30

64 66 68

Simpan Selesai Selesaikan Rekaman

Gambar 12-6. Entri data untuk lembar potong, di mana produsen individu menetapkan data preset ke video baru yang mereka buat

Keindahan sistem ini adalah semua poin kesepakatan, batasan, dan proses bisnis individu disembunyikan dari pengguna, yang hanya diminta untuk masukkan topik, acara, pencari lokasi, dan kesopanan. Keempat item tersebut membuat katalog rumit yang membantu bisnis memonetisasi konten

12.5 PIKIRAN TERAKHIR

Anda mungkin tidak menyukai ini, tapi begini: tidak ada yang peduli dengan perjuangan Anda sampai Anda menjadi bagian dari momen yang luar biasa. Ini mungkin terlihat kasar, tetapi semakin cepat Anda menerimanya, semakin cepat Anda bisa berhasil. Dalam olahraga, kita belajar bahwa tidak ada yang peduli dengan perjuangan atlet sampai mereka berhasil. Saya memiliki nomor di dinding kantor saya selama beberapa tahun, dan hanya sedikit orang yang pernah menebak apa artinya. Jumlahnya 1.184. Itu adalah jumlah pemain yang merupakan atlet luar biasa yang akan dilepas pada akhir pramusim NFL ketika 32 tim memangkas daftar pemain mereka dari 90 pemain menjadi 53. Beberapa dari ribuan pemain muda itu akan mendapatkan pekerjaan, tetapi untuk banyak dari mereka, itu sudah berakhir. Mereka semua melakukan latihan selama 15 jam untuk bersiap-siap menghadapi sepak bola profesional, hanya untuk latihan di kamp pelatihan. Kemudian, dalam satu saat, mereka dipotong.

BAB 13

KASUS PENGGUNAAN PELATIHAN PERUSAHAAN *VIRTUAL REALITY*

Bab ini adalah tentang pelatihan perusahaan *virtual reality* (VR), dengan fokus pada penggunaan video sferis. Dalam menulis bab ini, tujuan saya adalah meletakkan apa yang paling berguna bagi kami, ketika kami memulai. Saya harap dapat bermanfaat bagi Anda.

Pelatihan perusahaan akan menjadi kisah sukses besar pertama untuk VR karena seberapa baik kekuatan dan keterbatasan VR cocok dengan lingkungan pelatihan perusahaan. Pelatihan adalah pasar yang lebih besar daripada yang dipikirkan orang; pada tahun 2017, IDR 121,7 miliar dihabiskan untuk bermain game, tetapi IDR 362,2 miliar dihabiskan untuk pelatihan

13.1 PENDAHULUAN: PENTINGNYA PELATIHAN PERUSAHAAN

Pada tahun 2018, STRIVR mengirimkan 17.000 layar yang dipasang di kepala Oculus Go (HMD) ke Walmart. Itu adalah beberapa HMD di setiap toko Walmart di Amerika Serikat, dengan lebih dari satu juta karyawan Walmart memiliki akses ke pelatihan perusahaan di VR setiap hari. Itu adalah dampak. Gambar 13-1 menggambarkan gudang darurat STRIVR, di mana setiap orang terlibat untuk melakukan kontrol kualitas pada setiap headset.

Agar VR berhasil, ia perlu memecahkan satu masalah spesifik dalam skala besar dan melakukannya lebih baik daripada teknologi lainnya. Pelatihan perusahaan adalah industri yang siap untuk diubah. Pelatihan perusahaan adalah masalah itu.

13.2 KASUS PENGGUNAAN: PELATIHAN RUMAH BANJIR

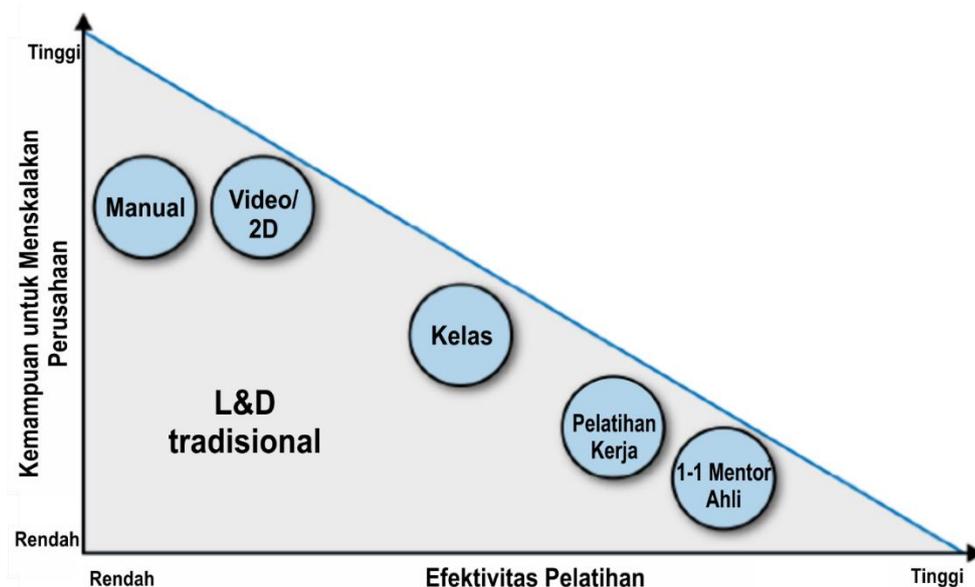
Cara terbaik untuk mempelajari sesuatu adalah dengan melakukannya. Untuk tugas-tugas seperti belajar menerbangkan pesawat atau melakukan operasi jantung terbuka, ini tidak selalu aman atau memungkinkan. Orang-orang telah menemukan berbagai metode untuk menyampaikan informasi tentang suatu tugas tanpa benar-benar mengoleskan pisau bedah ke kulit. Membaca manual tentang suatu tugas adalah salah satu cara yang paling tidak mendalam untuk belajar, sedangkan dipimpin melalui tugas oleh seorang ahli adalah salah satu yang paling mendalam.

Dilatih dalam VR saat ini tidak dapat menandingi memiliki instruktur manusia yang memandu Anda melalui tugas satu lawan satu, tetapi itu bisa mendekati sementara jauh lebih murah dan lebih terukur.



Gambar 13-1. 17.000 HMD sedang dipersiapkan untuk pengiriman (© STRIVR 2018)

Gambar 13-2 menunjukkan sebar plot dengan satu sumbu mewakili biaya dan skalabilitas, dan sumbu lainnya mewakili efektivitas. Di satu sisi, pertimbangkan manual pelatihan. Anda dapat mengirimnya ke mana saja, mencetaknya sesuai permintaan, atau membacanya di layar, tetapi itu bukan alat pengajaran yang sangat efektif, terutama ketika Anda mempertimbangkan untuk mengajarkan tugas fisik seperti mengikat simpul. Manual pelatihan sangat terukur, tetapi tidak terlalu efektif.



Gambar 13-2. Skalabilitas versus efektivitas opsi pelatihan (© STRIVR 2018)

Di ujung lain spektrum adalah bimbingan ahli satu lawan satu, bentuk pelatihan yang paling efektif. Seorang instruktur manusia tahu segalanya tentang subjek mereka dan dapat memandu Anda melaluinya langkah demi langkah, melibatkan Anda, membimbing Anda,

menantang Anda, dan menanggapi kemajuan Anda. Namun, ini membutuhkan waktu yang berharga dari seorang ahli yang dibayar tinggi. Bentuk pelatihan ini sangat efektif tetapi mahal dan sulit untuk diukur.

Janji VR adalah membangun sesuatu yang murah untuk didistribusikan seperti teks digital, tetapi seefektif bimbingan ahli satu lawan satu. Dengan mengingat hal itu, apakah pelatihan VR bisa seefektif itu? Tidak ada penelitian yang secara meyakinkan membuktikannya, tetapi semakin banyak bukti yang mengarah ke sana.

VR menciptakan respons fisiologis yang lebih dekat dengan kenyataan daripada media lainnya. Contoh klasiknya adalah "papan", di mana pengguna yang memakai HMD dengan tracking skala ruangan ditempatkan ke dalam lingkungan grafis komputer 3D di mana mereka ditanggung pada ketinggian yang tinggi di atas kota. Pada kenyataannya, pengguna berdiri di atas balok kayu yang bertumpu di lantai, tetapi dari sudut pandang pengguna, mereka tertatih-tatih di ambang kematian. Beberapa pengguna yang mencoba pengalaman ini dapat menyangkal efek fisik mendalam yang ditimbulkannya pada Anda: keseimbangan Anda goyah, kaki Anda lemas, dan setiap langkah maju membuat jantung Anda berdebar kencang.

Anda belajar paling baik saat melakukan sesuatu yang nyata, dan VR terasa nyata. Saat melatih pelajar dewasa, menciptakan pengalaman yang terasa nyata adalah kunci untuk memotivasi mereka dan membantu mereka menyerap pengetahuan baru.⁶ VR membawa pelajar lebih dekat dengan kenyataan daripada media pelatihan lainnya, dengan risiko dan biaya yang lebih kecil.

Pelatihan VR sangat cocok untuk kebutuhan pelajar dewasa. Dalam *The Adult Learner*, Malcolm S. Knowles mengemukakan bahwa orang dewasa memiliki kebutuhan belajar yang berbeda dari anak-anak. Ketika orang dewasa belajar, mereka dimotivasi oleh keprihatinan praktis. Mengapa saya belajar ini? Bagaimana itu bisa berguna? Dalam situasi kehidupan nyata apa saya dapat menerapkan pengetahuan ini? Pelatihan dalam VR memberikan banyak manfaat dibandingkan pembelajaran non-pengalaman:

Engagement

VR adalah lingkungan yang kaya interaksi di mana pelajar terus-menerus dipanggil untuk terlibat. Hanya memakai headset dan melihat-lihat berarti Anda sudah berinteraksi. Dalam skenario interaksi "Pelatihan Perampokan Toko", misalnya, perampok pertama-tama mendekati pelajar dari belakang, dan pelajar harus secara fisik menoleh untuk melihat apa yang terjadi. VR memaksa pengguna untuk menjadi peserta aktif dalam pengalaman.

Konteks

Pelatihan VR yang baik menempatkan pelajar dalam lingkungan yang realistis di mana keterampilan yang mereka pelajari akan berguna. Dalam skenario "Pelatihan Rumah Banjir", perbedaan antara air Kategori 1 dan Kategori 3 tidak bersifat akademis; inilah perbedaan antara merobek dan mengganti semua lantai di rumah versus hanya mengeringkannya.

Motivasi

Pelajar dapat melihat konsekuensi dari tindakan mereka. Menerapkan keterampilan baru secara efektif akan menunjukkan hasil yang baik, dan kegagalan menerapkan pembelajaran akan mengakibatkan kerugian. Misalnya, dalam skenario "Pelatihan Kabel Turun", kegagalan

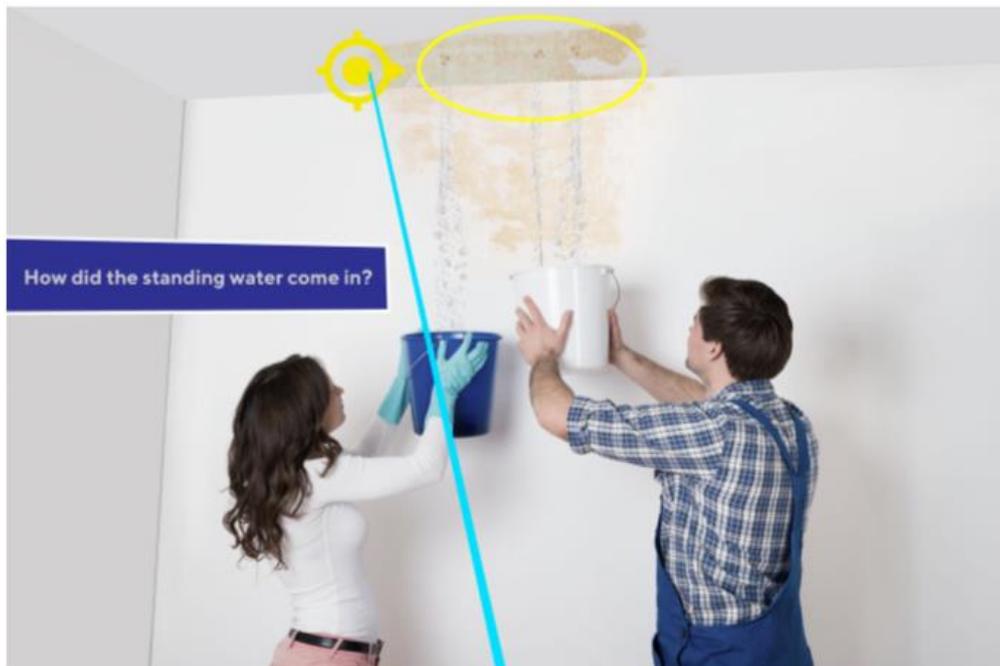
untuk mengomunikasikan bahaya kabel yang jatuh mengakibatkan anjing peliharaan tersengat listrik.

Di STRIVR, ada peluang untuk melakukan studi kecil tentang kemandirian pelatihan VR. Di bagian berikutnya, kita melihat kasus penggunaan di mana kemandirian metode pelatihan VR diuji terhadap bimbingan ahli satu lawan satu.

Penggunaan Kasus

Rumah banjir adalah rumah asli yang sengaja digenangi beberapa kali dalam setahun, agar profesional asuransi bisa melatih. Ada sekitar 30 atau lebih dari rumah-rumah ini di seluruh Amerika Serikat. Seorang instruktur ahli bekerja dengan kelas kecil, dan bersama-sama mereka mengeringkan rumah dan memperbaiki atau mengganti apa yang telah rusak. Ini adalah salah satu metode pelatihan yang paling efektif karena sangat cocok dengan kenyataan.

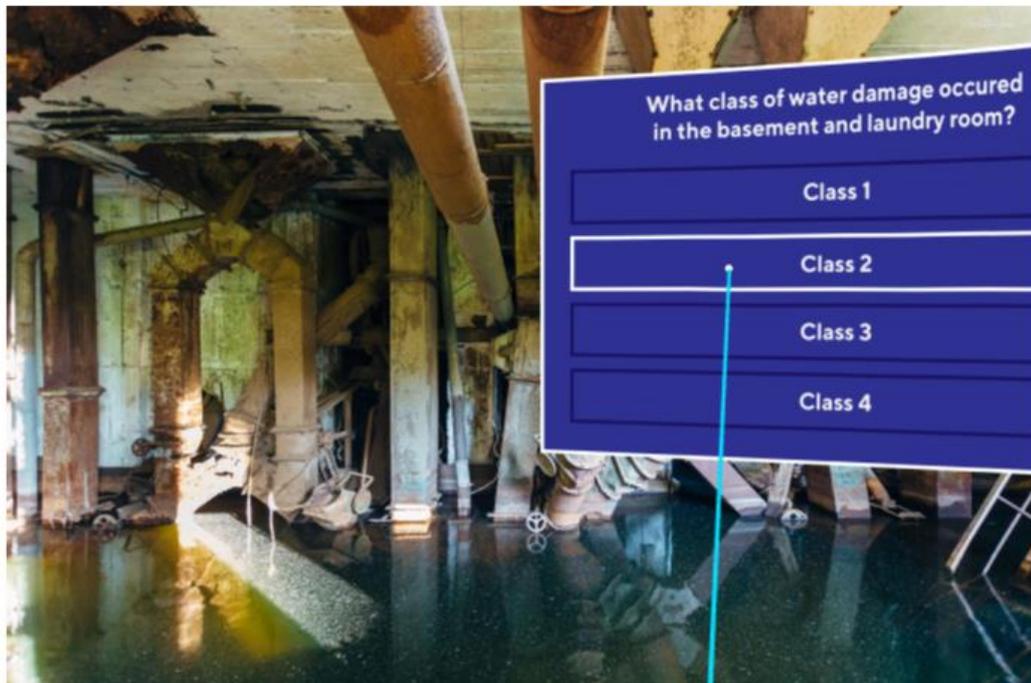
Membangun rumah, membanjirinya, dan kemudian memperbaikinya, tidak mengherankan, mahal. Kerusakan yang terjadi pada rumah itu nyata dan mahal. Ditambah lagi, karena hanya ada beberapa rumah banjir di setiap negara bagian, peserta pelatihan harus diterbangkan ke lokasi.



Gambar 13-3. Pertanyaan penanda digunakan untuk membuat pelajar tetap terlibat dan berinteraksi selama pelajaran

Namun, pengeluaran biaya ini sepadan, karena sejumlah besar uang yang dimainkan. Misalnya, Badai Florence pada 2018 menyebabkan kerugian asuransi sebesar tiga hingga lima miliar rupiah. Tidak semua klaim asuransi dibuat dengan niat jujur, dan penipuan menyumbang sekitar 10% dari klaim. Memiliki profesional asuransi terlatih yang dapat

mengurangi jumlah itu sangat penting bagi perusahaan asuransi. Tetapi bagaimana jika mereka bisa mendapatkan hasil yang sama dengan biaya yang lebih sedikit?



Gambar 13-4. Sebuah pertanyaan pilihan ganda tentang kategori kerusakan air

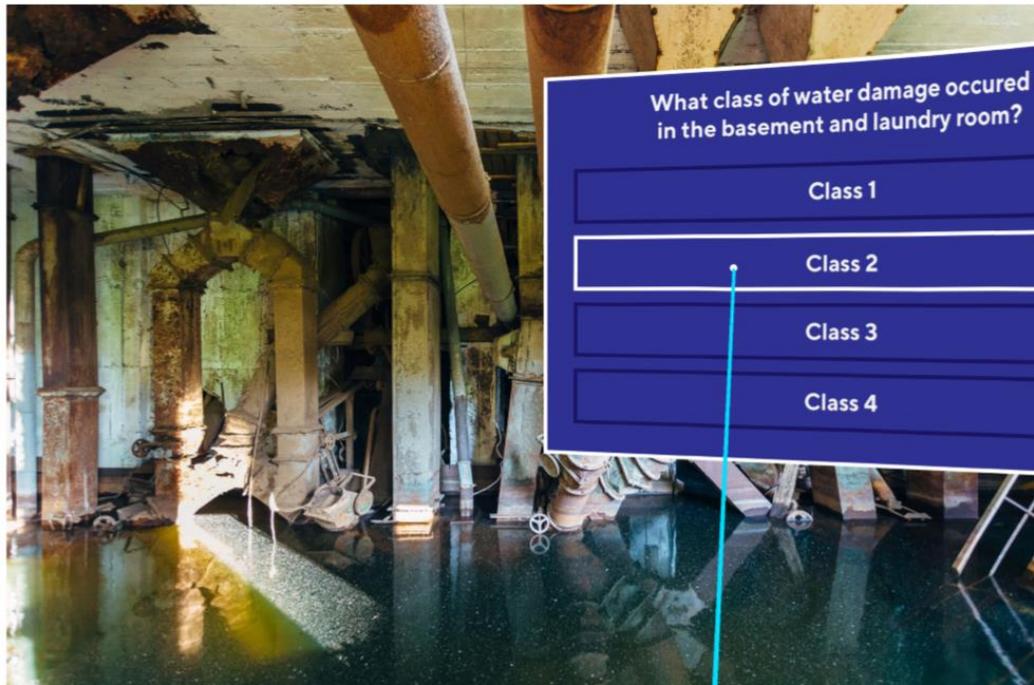
STRIVR mulai membuat versi VR dari kursus pelatihan rumah banjir, bekerja sama dengan instruktur ahli. Kru kamera merekam video bola di rumah dari sudut pandang seorang siswa yang diajari satu lawan satu, dan kemudian desainer membuat modul pelatihan VR menggunakan video tersebut. Dalam hal ini, modul pelatihan VR dibuat untuk mencakup semua hal yang akan diajarkan selama kelas secara komprehensif, terlepas dari apakah setiap aspek pelatihan "cocok" untuk VR.

Setelah modul pelatihan VR dibangun, STRIVR mengambil kelas yang terdiri dari 60 siswa dan menjalankan setengah dari mereka melalui rumah banjir yang sebenarnya, dan setengah lainnya melalui modul pelatihan VR. Tim data STRIVR kemudian menilai perbedaan efektivitas antara pelatihan rumah banjir yang sebenarnya dan pengalaman VR.

Dalam modul pelatihan VR STRIVR, seorang narator memandu peserta pelatihan melalui skenario klaim asuransi di mana mereka harus menilai kerusakan air yang terjadi di rumah Lisa. Peserta pelatihan tetap terlibat dengan diminta untuk berinteraksi dengan lokasi dalam video (Gambar 13-3) atau menjawab pertanyaan pilihan ganda tentang apa yang telah mereka pelajari (Gambar 13-4).

STRIVR menguji pengetahuan masing-masing kelompok peserta pelatihan sebelum dan sesudah setiap pelatihan, untuk melihat seberapa banyak mereka meningkat. Mereka menemukan bahwa kedua kelompok meningkat dengan jumlah yang kira-kira sama dalam pelatihan Mitigasi Air dan pelatihan Peningkatan. Ini menunjukkan komparabilitas; pengalaman dalam VR kira-kira setara dengan diterbangkan ke rumah banjir yang sebenarnya

tetapi dengan biaya yang jauh lebih sedikit. Kedua kelompok meningkat, seperti yang diilustrasikan pada Gambar 13-5, dan tidak ada perbedaan yang signifikan secara statistik antara hasil mereka.



Gambar 13-5. Pelatihan VR dan pelatihan rumah banjir di kehidupan nyata menghasilkan peningkatan yang sebanding (© STRIVR 2018)

Pengumpulan data juga merupakan manfaat dari pelatihan VR. STRIVR melacak data pergerakan pengguna dengan mencatat pergerakan kepala dan tangan mereka. Satu pertanyaan khususnya sangat sulit bagi peserta pelatihan, dengan lebih dari setengahnya mendapatkan jawaban yang salah. Tim analisis data melihat bahwa kelompok peserta pelatihan yang menjawab salah memiliki lebih banyak gerakan daripada peserta pelatihan yang menjawab pertanyaan dengan benar. Tim data berspekulasi bahwa ini bisa berarti peserta pelatihan gelisah, mengamati lingkungan, atau tidak memperhatikan dengan seksama. Karena ini adalah ukuran sampel yang kecil, tidak ada kesimpulan yang dapat ditarik, tetapi seiring dengan peningkatan alat, insight seperti ini dapat digunakan untuk membuat konten yang lebih adaptif. STRIVR mempelajari beberapa pelajaran penting dari kasus penggunaan ini: *Konten harus berukuran kecil*

Perancang konten cenderung melebih-lebihkan jumlah waktu yang ingin dihabiskan pengguna di VR. Sangat penting untuk memecah konten sehingga pengguna dapat beristirahat. 20 menit adalah patokan yang baik untuk sesi pelatihan, jadi pelajaran individu harus berada jauh di bawah jumlah waktu ini.

Tidak semua konten “cocok” untuk VR

Karena STRIVR ingin memasukkan semua konten dari pelatihan di tempat, STRIVR membuat kesalahan dengan memasukkan konten yang tidak cocok untuk VR. Misalnya, dalam satu bagian peserta harus menggunakan persamaan matematika untuk menghitung jumlah

penggerak udara yang diperlukan (lihat Gambar 13-6). Dalam keadaan normal, peserta pelatihan akan memiliki akses ke kalkulator, tetapi ini tidak disediakan dalam skenario VR. Ini adalah contoh yang baik dari jenis pembelajaran yang lebih baik dilakukan di dalam kelas: membutuhkan alat dari luar, dan tidak menggunakan kekuatan media VR.



Gambar 13-6. Meminta pelajar untuk melakukan perhitungan tanpa konteks bukanlah “cocok” untuk VR

Apa Manfaat Pelatihan VR? MENGENDARAI.

VR bukan untuk setiap kasus penggunaan. STRIVR menggunakan akronim “RIDE” untuk menentukan tempat terbaik untuk menggunakan VR:

- Langka
- Mustahil
- Berbahaya
- Mahal

Berikut adalah beberapa contoh situasi yang sesuai dengan kriteria ini:

Rare Black Friday jarang terjadi, hanya terjadi sekali setiap tahun, namun ini adalah momen keuangan yang kritis bagi pengecer. Perputaran yang tinggi berarti tidak cukupnya karyawan yang membawa pengetahuan dan pengalaman dari tahun ke tahun.

Mustahil

Perampokan toko tidak mungkin diprediksi, tetapi kegagalan untuk bereaksi dengan tepat dalam situasi ini dapat mengakibatkan hilangnya nyawa. Sulit untuk mengetahui bagaimana Anda akan bereaksi dalam situasi seperti ini sampai Anda mengalaminya sendiri.

Berbahaya

Lantai pabrik dengan prosedur keselamatan yang tidak diperhatikan dengan benar berbahaya untuk dipentaskan, tetapi penting bagi karyawan untuk dapat mengenali dan memperbaiki kesalahan dengan cepat.

Mahal

Seperti yang kita lihat dalam kasus penggunaan di awal bab ini, profesional asuransi dilatih dalam “rumah banjir”, yang menggunakan rumah dan air asli untuk menggambarkan kondisi banjir secara realistis. Meskipun pelatihan ini realistis, tetapi juga mahal.

Saat ini, pelatihan VR berorientasi spasial adalah yang terbaik untuk tugas-tugas yang melibatkan tubuh manusia yang berinteraksi di lingkungan atau dengan manusia lain. Tugas yang melibatkan antarmuka dengan layar komputer sangat tidak cocok untuk VR, karena tugas ini dapat lebih mudah dilakukan melalui antarmuka komputer umum. Karena sifat pekerjaan modern dan pekerjaan kantor, ini menghilangkan sejumlah kemungkinan kasus penggunaan.

Apa yang Membuat Pelatihan VR Bagus?

Pelatihan VR yang baik harus sebagai berikut:

- spasial
- Sederhana dan mudah diakses
- Pendek
- Berorientasi pada tujuan
- Dapat diskalakan

Mari kita lihat lebih dekat masing-masing karakteristik:

Spasial

Pelatihan VR harus spasial untuk memanfaatkan sifat 3D VR, memanggil lokasi di atas, di belakang, dan di bawah pengguna. Ini menekankan perwujudan pengguna dan membantu meningkatkan daya ingat.

Sederhana dan dapat diakses

Keuntungan besar VR adalah aksesibilitas, dan skema kontrol Anda perlu mencerminkan hal itu. Video game modern menggunakan metode kontrol yang kompleks dan tidak intuitif yang mengandalkan pengalaman dan keakraban gamer dengan genre tersebut; Anda tahu ini jika Anda pernah melihat seseorang yang tidak terbiasa dengan game first-person shooter mencoba untuk bergerak, menembak, dan melihat-lihat pada saat yang bersamaan. Sebagian besar perangkat keras VR mendukung controller tangan titik-dan-klik yang sederhana. Arahkan dan klik sangat bagus karena secara konseptual mirip dengan menggunakan mouse komputer atau penunjuk laser. Hindari membuat pengguna mempelajari berbagai tombol dan antarmuka. Anda mencoba untuk mengajarkan keterampilan kehidupan nyata, dan semakin Anda mengedepankan sifat pengontrol, semakin sedikit pengalaman Anda yang akan dipetakan dengan nyaman ke kenyataan.

Pendek

Sesi harus seukuran gigitan. Sesi pelatihan VR tidak boleh lebih dari sekitar 20 menit. Ini membantu mencegah headset menjadi tidak nyaman serta memudahkan pengguna untuk menyerap konten dengan kecepatan mereka sendiri. Jika konten Anda dibagi dengan baik, pengguna akan merasa nyaman untuk terjun dan melakukan pelatihan, mengetahui bahwa mereka akan segera dapat memutuskan apakah akan melanjutkan atau kembali ke tugas lain

di kehidupan nyata. Memiliki penghalang yang rendah untuk masuk dan pelatihan berarti pengguna akan lebih sering masuk.

Berorientasi pada tujuan

Karena waktu sesi harus singkat, dan waktu pembelajar sangat mahal, VR paling baik digunakan untuk tugas-tugas pembelajaran yang memiliki aturan dan prosedur yang jelas daripada untuk bereksperimen di lingkungan seperti kotak pasir. (Namun, seiring dengan peningkatan teknologi dan VR menjadi lebih alami dan nyaman, pelatihan sandbox mungkin menemukan lebih banyak kegunaan.)

Keuntungan

VR yang dapat diskalakan dibandingkan media pelatihan lainnya adalah kualitas tinggi dan skalabel. Ingatlah hal ini saat Anda membangun platform dan konten Anda. Baik itu harus mudah untuk membuat konten baru atau konten yang dibuat harus sangat dapat digunakan kembali oleh sejumlah besar pengguna.

Video sferis adalah rekaman di mana setiap arah pandangan ditangkap sekaligus, memungkinkan pemirsa untuk secara fisik menoleh untuk melihat berbagai aspek video. Efek keseluruhannya adalah seolah-olah penonton secara fisik hadir di lokasi pengambilan video, dengan perbedaan mencolok bahwa penonton tidak dapat bergerak atau memengaruhi lingkungan dengan cara apa pun.

Video sferis jarang dianggap sebagai "pilihan pertama" untuk konten pelatihan VR. Ketika pelanggan perusahaan menggambarkan jenis pengalaman pelatihan yang ingin mereka bangun, yang terlintas dalam pikiran adalah lingkungan grafis komputer 3D yang sepenuhnya interaktif dan realistis.

Namun, agar kerangka kerja pelatihan dapat diskalakan dan efisien, perlu dibuat konten dengan cepat dan mudah. Anda membutuhkan alat. Jika Anda mencoba membangun sistem untuk membuat konten pelatihan 3D yang dapat melakukan apa pun yang diinginkan klien (misalnya, operasi laparoskopi, simulasi kendaraan, interaksi pelanggan), pada saat Anda selesai menambahkan fitur, Anda mungkin akan mendapatkan sesuatu yang terlihat seperti mesin game berfitur lengkap.

Ingat, keuntungan utama VR untuk pelatihan adalah skalabilitas. Harus murah dan cepat untuk membuat pelatihan untuk banyak orang. Semakin padat karya dalam membuat pelatihan, semakin jarang Anda dapat memperbaruinya, dan semakin sedikit konten yang dapat Anda hasilkan. Meskipun video game telah ada selama beberapa dekade, seni pelatihan perusahaan saat ini masih berupa video 2D, karena video mudah dibuat, dipelihara, diperbarui, dan diganti.

Video sferis semudah merekam video 2D konvensional, sambil menawarkan sejumlah manfaat dalam hal interaktivitas.

Manfaat Video Spherical

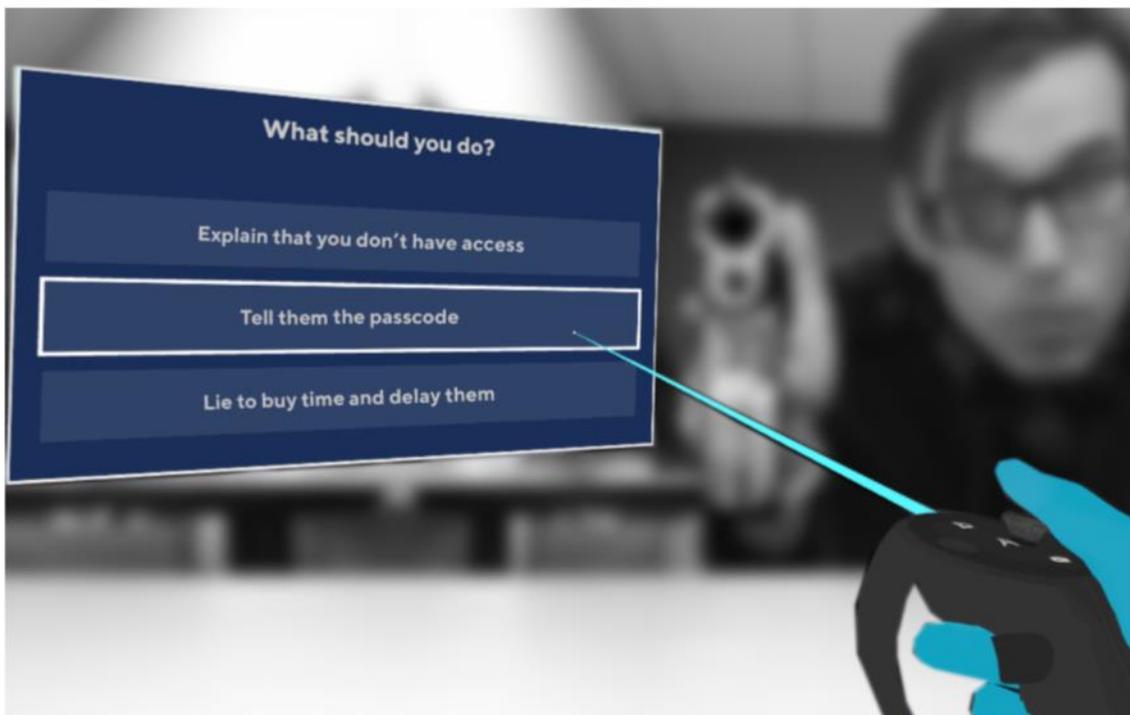
Berikut adalah beberapa manfaat menggunakan video spherical:

- Dapat diskalakan
- Mudah untuk menghasilkan konten

- Murah
- Lebih interaktif daripada video 2D

Saat membangun pelatihan VR, perlu untuk menciptakan kembali lingkungan pelatihan. Dengan video sferis, ini hanyalah masalah merekam lingkungan itu di tempat. Ini menjamin bahwa hasil film Anda benar-benar realistis dan sama persis dengan lingkungan nyata.

Ketika manusia adalah bagian penting dari pelatihan, instruktur dan karyawan yang sebenarnya dapat membantu. Orang-orang ini akan memiliki seragam yang tepat, mengetahui prosedur yang diajarkan, dan memiliki pengalaman mendemonstrasikannya. (Peringatan: seperti yang akan kita bahas nanti, selalu ada baiknya mempekerjakan aktor sungguhan saat merekam peran yang membutuhkan penggambaran emosi atau ketenangan di depan kamera.) Dengan tingkat teknologi dan grafik komputer kita saat ini, saat ini tidak ada grafik berkemampuan VR yang dapat mendekati tingkat realisme video.



Gambar 13-7. Pertanyaan pilihan ganda telah menambahkan konteks dalam lingkungan VR di mana Anda dapat melihat konsekuensi dari pilihan Anda

Tantangan Video Spherical

Namun, ada juga tantangan dalam menggunakan video sferis untuk membuat konten. Beberapa situasi sulit untuk dipentaskan, bahkan sekali untuk kamera. Untuk ini, kita dapat menggunakan pengeditan video untuk membuat efek yang diperlukan, tetapi ini dapat membuat hasil yang tidak realistis.

Tantangan lain adalah bahwa karyawan dan instruktur di lokasi mungkin bukan aktor alami yang tampil bagus di film. Jika pelatihan untuk sesuatu seperti perampokan toko, aktor nyata mungkin diperlukan, menciptakan pengeluaran tambahan. Sangat penting untuk mendapatkan konten yang benar pertama kali karena kembali ke situs untuk merekam ulang

atau membersihkan video secara digital mahal dan memakan waktu. Namun tantangan terbesar dari video sferis adalah interaktivitas dalam pengalaman.

Interaksi dengan Video Spherical

Saat mempertimbangkan apa yang dapat dicapai dengan video sferis, salah satu tempat pertama untuk mencari inspirasi adalah video 2D. Bentuk interaksi apa yang dimungkinkan dengan video 2D, dan keuntungan apa yang dihasilkan oleh video sferis dibandingkan?

Reaksi

Seperti halnya media apa pun, sebagian dari interaksi pengguna terjadi di dalam kepala pengguna itu sendiri, saat mereka menonton dan mendengarkan informasi yang disampaikan, dan kemudian memprosesnya. Bahkan tanpa meminta interaksi langsung, pengguna berinteraksi dengan media dengan harapan, pemikiran, dan pertanyaan internal mereka. Dengan cara ini, video 2D dan sferis berbagi persyaratan bahwa konten harus menarik, menarik, diproduksi dengan baik, dan relevan. Saat memproduksi video sferis untuk pelatihan, latar belakang teknik yang digunakan untuk membuat video 2D menghibur sangat penting.

Tatapan

Tatapan adalah bentuk interaksi yang jauh lebih kaya dalam video sferis daripada dalam video 2D. Dalam video 2D, pemirsa sedang menonton layar yang dibatasi. Pergerakan kamera dan transisi bidikan adalah pilihan yang disengaja oleh editor untuk memfokuskan pemirsa pada tempat dan waktu tertentu. Namun dalam video sferis, pengeditan harus dilakukan seminimal mungkin untuk menghindari disorientasi pengguna. Ini berarti bahwa permintaan yang lebih besar ditempatkan pada pemirsa; penampil adalah kamera, dan mereka harus berperan aktif dalam melihat sekeliling dan menyerap informasi. Dalam pengalaman "Store Robbery Training" STRIVR, misalnya, pelajar pertama kali didekati oleh perampok dari belakang dan harus menoleh untuk melihat mereka. Dengan cara ini, tatapan bisa menjadi alat interaktif yang kuat dalam video sferis.

Pertanyaan pilihan ganda

Dalam lingkungan elearning kontemporer, video sering diisi dengan pertanyaan pilihan ganda. Dalam video sferis, pertanyaan pilihan ganda dapat disajikan dari dalam pengalaman, bukan di luarnya, menambahkan konteks pada keputusan pengguna. Misalnya, pada Gambar 13-7, salah satu perampok menodongkan pistol ke arah Anda dan meminta Anda memberinya akses ke brankas. Melihat waktu membeku dan mendengar suara detak jantung saat Anda dengan panik mencoba memutuskan apa yang harus dilakukan adalah pengalaman yang jauh lebih menarik secara emosional dalam video berbentuk bola daripada di layar komputer. Video sferis juga memberi Anda kesempatan untuk menggunakan lokasi atau objek dalam video sebagai "jawaban" atas sebuah pertanyaan. Pada Gambar 13-8, quarterback sepak bola diminta untuk memilih lokasi "runfit"-nya setelah menonton beberapa detik pertama dari sebuah down.

Tempat Menarik

Video bulat memiliki alat interaksi canggih yang jarang digunakan dengan video 2D: menunjuk dan memilih lokasi di dalam video. Ini mungkin merupakan bentuk interaksi terbaik dengan

video sferis karena memberikan kesempatan kepada pelajar untuk berinteraksi langsung dengan media. Tempat Menarik (Pols) dapat digunakan sebagai informasi, untuk menarik perhatian ke objek utama dalam video, seperti pada Gambar 13-9.



Gambar 13-8. Seorang pemain sepak bola harus mengidentifikasi "run fit"-nya dengan memilih lokasi yang benar di video



Gambar 13-9. Tempat Menarik dapat digunakan sebagai informasi untuk menyoroti bidang minat utama dalam video



Gambar 13-10. "Perburuan Scene" adalah latihan yang melibatkan menemukan kesalahan atau item tersembunyi lainnya di lingkungan

Pols juga dapat digunakan dalam konteks "permainan objek tersembunyi", di mana pengguna hanya disajikan video dan diminta untuk mengidentifikasi kelas item atau kesalahan. Tidak ada yang terlihat di video selain prompt, dan pengguna harus mengklik di berbagai tempat, mencari item atau lokasi yang sesuai dengan target mereka. Ini adalah pelatihan yang bagus karena pelajar dipaksa untuk melihat dari dekat lingkungan mereka, mempertimbangkannya, dan berinteraksi dengan mereka dalam konteks yang mirip dengan kenyataan. Teknik ini sering digunakan dalam "Pelatihan Lantai Pabrik", di mana pengguna diminta untuk mengidentifikasi bahaya tersandung dan bahaya lain di lingkungan mereka. Digambarkan pada Gambar 13-10 adalah skenario di mana pengguna telah menemukan lima Pol dan memiliki waktu sekitar satu menit untuk mengidentifikasi 16 sisanya.

Pilih petualangan Anda sendiri

Akhirnya, ada cara untuk "menipu" dan berinteraksi lebih langsung dengan video. Game arcade laserdisc 1983 *Dragon's Lair* adalah salah satu contoh paling awal. Dalam game ini, pemain memasukkan perintah sederhana sebagai respons terhadap kartun animasi. Jika mereka memasukkan input arah yang benar pada waktu yang tepat, video akan berlanjut. Jika mereka gagal, karakter ditampilkan sekarat. Meskipun video tidak dapat berinteraksi langsung dengan pengguna seperti yang dapat dilakukan oleh pengalaman grafis komputer interaktif, dimungkinkan untuk merekam beberapa video yang menggambarkan hasil yang berbeda, yang dapat menunjukkan konsekuensi peserta pelatihan dari tindakan salah yang dapat mereka ambil. Dalam pengalaman "Pelatihan Kabel Turun" STRIVR, misalnya, jika pelajar gagal

memberi tahu penelepon tentang bahaya kabel yang jatuh untuk hewan peliharaan, pelajar akan mendengar anjing pemanggil tersengat listrik.

Untuk menghindari ledakan kombinatorial, yang terbaik adalah mengikuti pendekatan “untaian mutiara” untuk merancang konten bercabang; hanya biarkan pembelajar sedikit menyimpang dari skenario yang Anda ajarkan sebelum membimbing mereka kembali ke jalan yang benar.

Kasus yang digunakan

Lantai pabrik sangat besar, keras, kacau, dan penuh bahaya tersembunyi. Dalam sebuah penelitian, kesalahan keamanan di 10 pabrik merugikan produsen makanan utama lima digit di bulan April 2015. Lima digit. Bukan uang, jari. Benang merah untuk pelatihan pabrik adalah identifikasi kesalahan. Diberikan video bulat dari lantai pabrik, dapatkan karyawan mengidentifikasi pelanggaran keselamatan di sekitar mereka?

Lantai pabrik memberikan contoh yang jelas tentang manfaat video sferis dibandingkan video 2D. Sangat mudah untuk melihat slide PowerPoint dengan gambar kendaraan pemuatan di atasnya, dan memahami bahwa kendaraan pemuatan itu berbahaya. Tetapi kendaraan pemuatan yang menabrak Anda bukanlah yang Anda lihat datang. Kendaraan yang menabrak Anda adalah yang tidak Anda lihat, yang datang dari belakang. Video sferis, seperti kenyataan, datang kepada Anda dari semua sudut.

Untuk memenuhi kebutuhan klien, STRIVR membangun fitur yang disebut perburuan Scene, di mana peserta pelatihan harus mengidentifikasi tempat menarik yang tersembunyi dalam video. Di sekitar tempat kejadian terdapat “hitbox” yang tersembunyi, dimana peserta pelatihan memiliki waktu terbatas untuk menemukannya, seperti yang ditunjukkan pada Gambar 13-11.



Gambar 13-11. Dalam "perburuan Scene", pelajar harus menemukan kesalahan dalam batas waktu dan memilih area yang mengandung kesalahan, agar kesalahan itu terungkap.

Jika pengguna tidak menemukan semua hotspot tepat waktu, mereka akan diberitahu tentang apa yang mereka lewatkan, seperti yang ditunjukkan pada Gambar 13-12.



Gambar 13-12. Setelah waktu berakhir, kesalahan yang tidak ditemukan pelajar terungkap dan harus diinteraksikan

Salah satu hal hebat tentang pelatihan semacam ini adalah kemudahan dalam menangkap rekaman dan membangun pelatihan. Pabrik biasanya memiliki catatan kecelakaan masa lalu, karyawan veteran yang berpengalaman luas, dan manual yang mengkategorikan kesalahan potensial. STRIVR menemukan bahwa mudah untuk mengidentifikasi kesalahan nyata di lantai pabrik atau untuk membuatnya dengan aman dengan bantuan instruktur.

Peran Narasi

Anda akan mengalami kesulitan untuk melatih siswa jika pengalaman pelatihannya membosankan. Salah satu alasan utama keberadaan STRIVR adalah karena keadaan seni saat ini untuk pelatihan...yah, menyebalkan. Dan alasan utama untuk ini adalah karena membosankan. Gamifikasi telah menjadi kata kunci di industri ini selama bertahun-tahun, tetapi fokusnya adalah pada poin, bonus harian, dan aliran dopamin. Mengapa orang bermain game? Karena mereka menghibur. Mereka menyenangkan, mereka apik, mereka dipoles, dan, biasanya, mereka memberi Anda alasan untuk peduli.

Saat Anda bermain game, Anda tidak hanya belajar menerbangkan pesawat luar angkasa untuk menendang. Anda belajar menerbangkan pesawat luar angkasa sehingga Anda bisa mengalahkan kaisar jahat yang membunuh ayahmu. Khususnya untuk pelajar dewasa, memberikan tujuan dan konteks pembelajaran mereka sangat membantu untuk membuatnya lebih mendalam dan menarik.⁸ Memberikan tugas sebuah tujuan menambah dimensi ekstra padanya dan membuka sedikit ekstra dari otak pelajar. Jika Anda ingin membangun pengalaman pelatihan VR yang benar-benar akan diselesaikan oleh peserta pelatihan, Anda harus memiliki narasi. Anda membutuhkan pembelajaran berorientasi tugas.

Mungkin sulit untuk mewujudkan pengguna dalam video sferis. Kamera tidak memiliki kehadiran fisik, yang dapat membuat pengguna merasa seperti hantu. Daripada membiarkan pengguna fokus pada kekurangan tubuh mereka, Anda harus memfilmkan video Anda dengan karakter yang berbicara dengan pengguna, terlibat dengan mereka, dan membimbing mereka. Manusia adalah hewan sosial, dan otak kita dirancang untuk menganalisis wajah dan membaca isyarat sosial. Melatih pengguna dengan meminta mereka berinteraksi dengan instruktur manusia adalah peningkatan besar dibandingkan menggunakan suara tanpa tubuh.

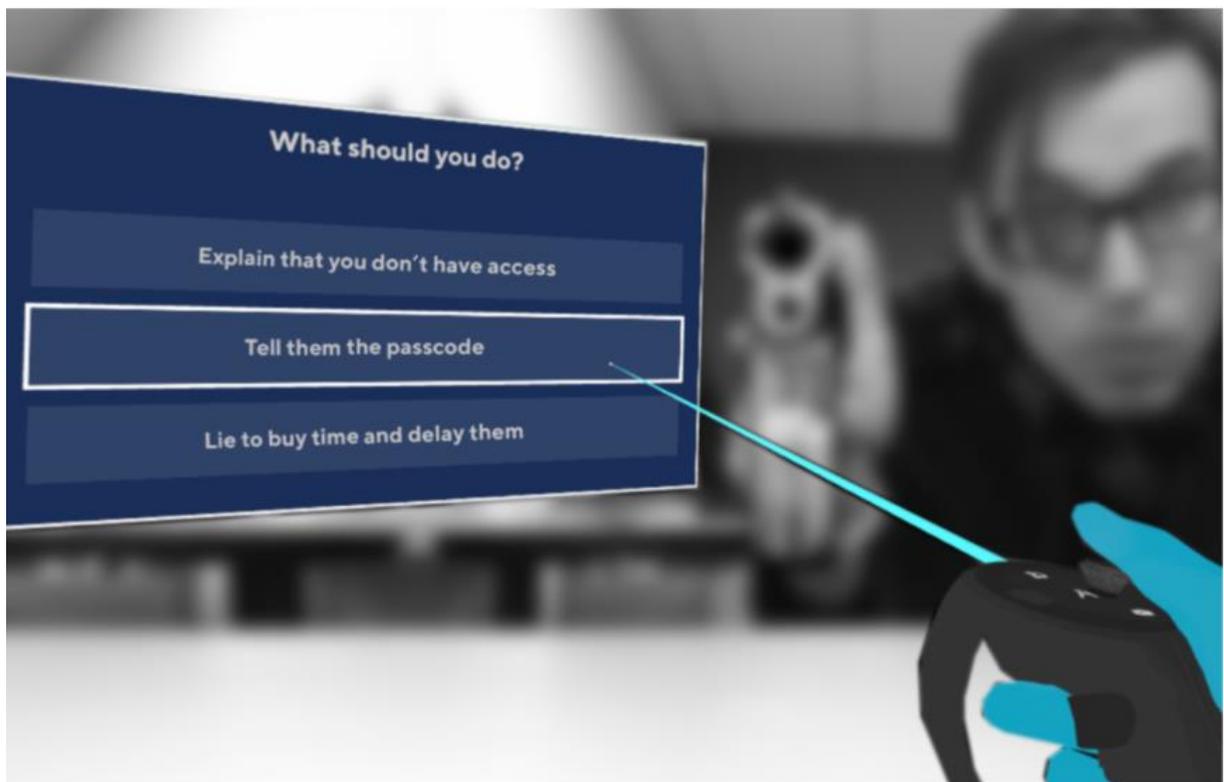
Bimbingan ahli satu-satu adalah skenario pelatihan yang ideal, dan merekam seorang instruktur adalah yang paling dekat yang bisa kita dapatkan dalam konteks video sferis.

Ini berarti bahwa merekam video sferis untuk pelatihan jauh lebih mirip dengan syuting film daripada yang mungkin Anda pikirkan. Anda harus memiliki naskah yang menarik dan menarik, dengan karakter dan alur cerita. Jika Anda ingin membuat pelatihan VR terbaik yang Anda bisa, itu juga berarti Anda perlu merekrut aktor.

Penggunaan Kasus

Perampokan toko adalah contoh bagus dari skenario yang hampir tidak mungkin untuk dilatih secara normal. Tanpa cara apa pun untuk memprediksi perampokan, skenario pelatihan terdekat adalah lokakarya atau permainan peran yang sangat terlibat.

Dalam “Pelatihan Perampokan Toko,” STRIVR membangun pengalaman yang didorong oleh narasi untuk menguji peserta pelatihan tentang apa yang telah mereka pelajari tentang tanggapan yang tepat terhadap perampokan toko. Daripada membangun pelatihan VR ini untuk mengajarkan semua keterampilan yang diperlukan, Pelatihan Perampokan Toko dibangun sebagai ujian pengantar dan ujian keluar untuk melengkapi kelas atau pelatihan online.⁸ Hasilnya, perancang instruksional STRIVR dapat fokus lebih sempit pada hal-hal yang bekerja dengan baik di VR.



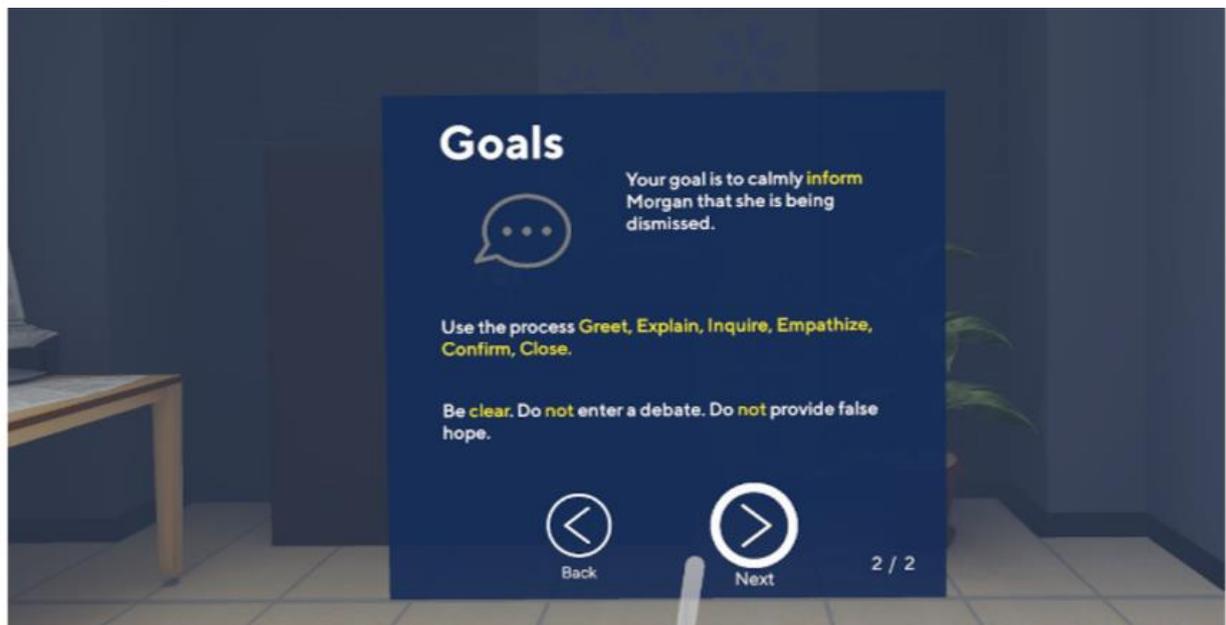
Gambar 13-13. Ketika perampok menanyai Anda, waktu melambat, lingkungan menjadi kabur, dan suara detak jantung menambah perasaan intensitas

Pelatihan Perampokan Toko menempatkan pembelajar pada posisi manajer toko yang membuka hari kerja dengan rekan kerja mereka. Sementara pelajar memperhatikan pintu, perampok datang dari belakang, mengejutkan rekan kerja pelajar dan memaksa pelajar untuk

secara fisik berbalik untuk melihat apa yang terjadi. Pelajar kemudian dipaksa untuk mengatasi situasi dan berinteraksi dengan menjawab pertanyaan pilihan ganda, yang temanya adalah bekerja sama dan tidak memusuhi para perampok. Toko memiliki asuransi dan kontinjensi, dan peran manajer toko adalah untuk mencegah bahaya apa pun yang menimpa karyawan mereka atau siapa pun di toko. Dalam pengalaman ini, jika pelajar memilih jawaban yang salah, prompt suara akan memberi tahu mereka tentang tindakan yang benar dan mengapa. Meskipun STRIVR telah menggunakan pertanyaan pilihan ganda di banyak pengalaman sebelumnya, ini adalah pertama kalinya pertanyaan tersebut digunakan dengan cara yang begitu mendalam. Ketika perampok menyanjai Anda, waktu melambat, layar menjadi buram dan berubah menjadi hitam putih, dan suara detak jantung memberi kesan intensitas. Gambar 13-13 menunjukkan efek waktu berhenti, yang mengaitkan pertanyaan pilihan ganda dalam konteks yang mendalam.

Pelatihan Perampokan Toko adalah contoh yang baik dari pengalaman pelatihan yang tidak akan mungkin terjadi tanpa aktor dan naskah. Perampokan adalah pengalaman yang menakutkan dan emosional; jika karakter manusia tidak menyampaikan emosi itu, pelajar tidak akan siap bereaksi dengan tepat, dan pengalamannya akan menjadi komedi daripada menakutkan.

Nilai dari pelatihan ini adalah kemampuannya untuk mempersiapkan mental peserta didik. Pelatihan memaksa pelajar untuk memvisualisasikan bagaimana bereaksi dan apa yang harus dilakukan. Jika situasinya benar-benar terjadi, pelajar dapat kembali pada model perilaku yang telah mereka latih secara virtual.



Gambar 13-14. Pelajar dipandu melalui garis besar tentang apa yang harus mereka sampaikan di setiap langkah (© STRIVR 2018)

Saat membangun pelatihan ini, desainer STRIVR memperhatikan betapa kuatnya tatapan pelajar tertarik pada karakter manusia, dan mereka mulai menggunakan instruktur manusia

untuk pelatihan di masa depan, daripada suara tanpa tubuh. Memiliki aktor yang berinteraksi dengan pelajar menawarkan banyak keuntungan. Para pelajar lebih terlibat, tetapi juga lebih mewujudkan diri mereka sendiri. Untuk situasi di mana peserta dalam VR tidak memiliki tubuh yang terlihat, memiliki manusia yang tingginya dapat Anda hubungkan membantu Anda merasa lebih hadir dalam Scene tersebut. Manfaat lain adalah dalam arah pandangan. Alih-alih meletakkan panah atau tanda di sekitar tempat kejadian, rasanya sangat wajar jika karakter manusia berjalan atau menunjuk untuk mengarahkan pandangan pembelajar.

Satu hal lagi yang perlu disebutkan: ketika membangun pengalaman yang berpotensi menjadi traumatis atau memicu, sangat penting untuk memberi tahu pelajar bahwa mereka dapat membatalkannya kapan saja. Untuk pelatihan ini, kami memastikan bahwa peserta menyadari bahwa mereka dapat menjeda, menekan tombol Home Oculus, atau melepas headset secara fisik kapan saja jika mereka merasa tidak nyaman.

Masa depan Training XR : Di luar spherical Video

Dalam bab ini, fokusnya adalah pada VR, dan terutama video sferis, sebagai media pelatihan. Video sferis sangat disukai karena mudah ditangkap dan memberikan hasil yang realistis. Namun, kami juga telah membahas bahwa video sferis dapat dibatasi dalam interaktivitas dan fidelitasnya. Bagaimana dengan teknologi lainnya? Ke mana arah pelatihan XR di masa depan? Di sisa bab ini, kita melihat peningkatan dan alternatif untuk video sferis.

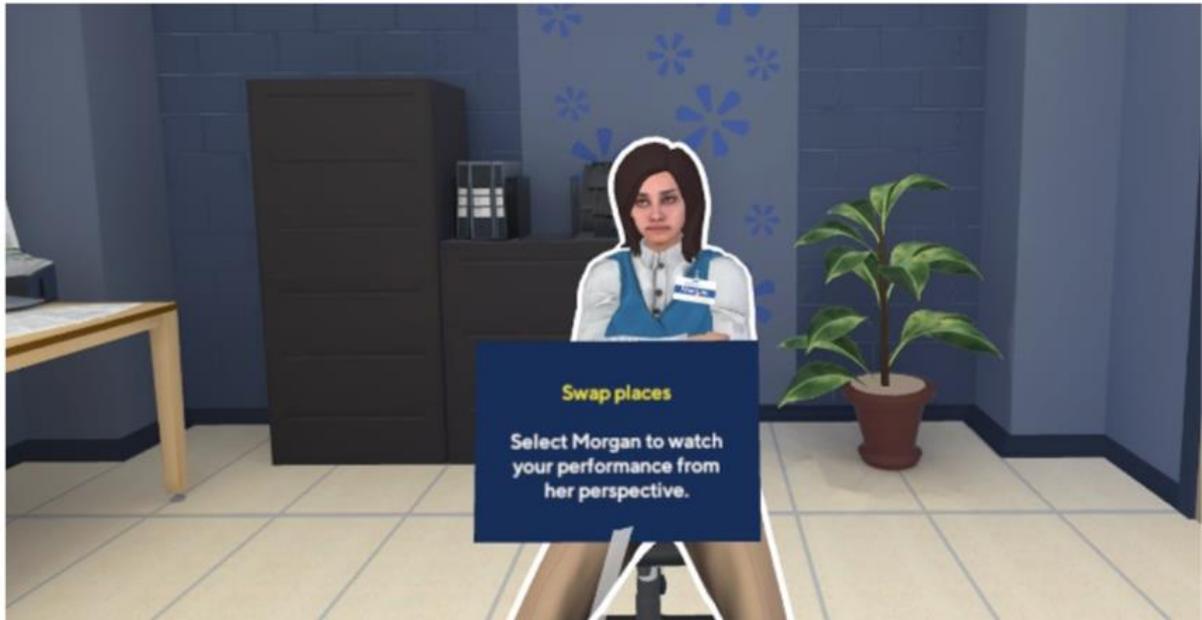
13.3 GRAFIS KOMPUTER

Pilihan utama lainnya untuk menggambarkan lingkungan pelatihan adalah grafik komputer (CG). CG memberikan manfaat besar dalam hal interaktivitas. Model 3D dapat bergerak secara dinamis dalam lingkungan tanpa perlu difilmkan dalam berbagai posisi yang tak terbatas.

Namun, grafik 3D perlu dimodelkan, dianimasikan, dan dinyalakan. Membangun aset 3D dan skenario interaktif memakan waktu. Rockstar Studios membutuhkan waktu lebih dari tiga tahun untuk mengembangkan Red Dead Redemption 2; jenis timeline ini khas untuk perusahaan game. Karena pertimbangan ini, saat membangun pengalaman pelatihan CG, pelatihan harus penting dan selalu ramah lingkungan.

13.4 KASUS PENGGUNAAN: PELATIHAN SOFT SKILL

Kasus penggunaan ini menyoroti kerangka kerja pelatihan VR yang dibuat dengan CG. Soft Skills menggunakan manusia virtual untuk mensimulasikan percakapan yang sulit, seperti memberikan penilaian kinerja yang negatif kepada karyawan. Peserta pelatihan pertama-tama memilih avatar untuk mewakili diri mereka sendiri. Mereka diperkenalkan dengan skenario dan kemudian diminta untuk mengatakan apa. Gambar 13-14 menunjukkan contoh prompt. Manusia virtual bereaksi, dan percakapan berlanjut. Terlepas dari sifat pengalaman yang ditentukan, sebagian besar pengguna percaya bahwa manusia virtual bereaksi dan beradaptasi dengan apa yang mereka katakan.



Gambar 13-15. Setelah percakapan, pembelajar menonton pemutaran kata-kata mereka sendiri dari sudut pandang lawan bicara mereka (© STRIVR 2018)

Dampak nyata dari pelatihan Soft Skills terjadi setelah percakapan. Sepanjang waktu Anda mengikuti petunjuk dan berbicara dengan karyawan virtual, suara dan gerakan Anda direkam. Di akhir pengalaman, semuanya diputar ulang dengan peran yang dibalik. Pada Gambar 13-15, pelajar akan bertukar tempat dengan Morgan, karyawan bermasalah. Duduk di seberang Anda dan melihat kata-kata Anda keluar dari mulut orang lain adalah pengalaman yang luar biasa. Ketika saya menjalani pelatihan ini untuk pertama kalinya, saya mengatakan banyak hal yang kemudian saya sadari dianggap tidak sopan. Misalnya, di awal percakapan, saya mengatakan bahwa pertemuan itu “bukan masalah besar.” Mendengarnya kembali di sisi lain meja, saya memiliki reaksi mendalam terhadap betapa tidak tulusnya itu terdengar.

Soft Skills adalah contoh yang baik dari aplikasi yang dapat digunakan kembali yang dapat mengambil manfaat dari investasi di CG. Pelatihan Soft Skills adalah kerangka kerja yang dapat digunakan kembali untuk keterampilan penting: komunikasi. Sebagian besar pekerjaan perusahaan memerlukan keterampilan komunikasi, mulai dari membantu pelanggan melalui telepon hingga mengelola tim yang terdiri dari beragam karyawan. Karena kerangka kerja CG yang sama dapat digunakan untuk berbagai jenis pelatihan komunikasi, investasi upaya dalam membangun lingkungan 3D dan avatar dapat memiliki peluang untuk membuahkan hasil seiring waktu.



Gambar 13-16. Menggunakan fotogrametri untuk membuat model objek kehidupan nyata masih membutuhkan banyak tenaga (© STRIVR 2018)

CG dalam konteks ini memiliki banyak manfaat dibandingkan video sferis. Misalnya, satu suara dapat digunakan dengan banyak avatar khusus yang berbeda untuk mengontrol penampilan manusia virtual. Animasi dapat digunakan kembali di seluruh avatar. Lingkungan dan avatar dapat dicampur dan dicocokkan. Skenario percabangan juga lebih mudah dibuat dengan CG, karena manusia virtual dapat bereaksi secara dinamis terhadap pengguna seperti halnya karakter video game.

13.5 MASA DEPAN: FOTOGRAMETRI

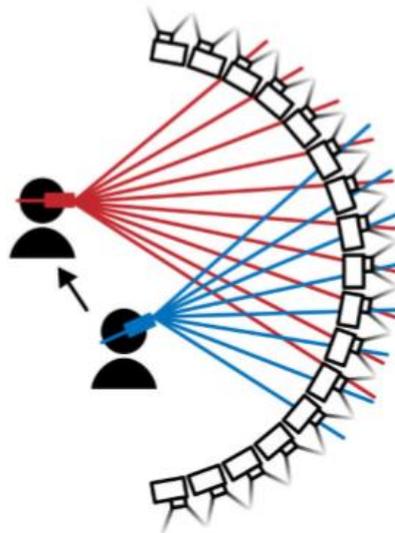
Fotogrametri adalah teknologi yang menarik untuk menangkap lingkungan kehidupan nyata, objek, atau orang dan membuatnya menjadi model 3D. Ribuan foto diambil dari subjek dari setiap sudut, dan kemudian gambar-gambar ini digabungkan menjadi model 3D.

Namun, sulit untuk membangun model 3D yang sempurna dari teknik fotogrametri saja. Banyak pekerjaan pembersihan yang harus dilakukan untuk menutup celah kecil pada model dan untuk memperbaiki cacat lainnya. Di STRIVR, kami menggunakan teknik ini untuk membangun dinding basah toko kelontong virtual, dan seniman 3D kami Tyrone harus menghabiskan lebih banyak waktu daripada kami mengantisipasi membersihkan lingkungan yang ditangkap dan elemen yang dapat berinteraksi. Gambar 13-16 menunjukkan sepotong brokoli yang sangat merepotkan.

Fotogrametri adalah teknologi yang perlu dipertimbangkan, tetapi perlu diingat tantangannya. Ini adalah teknologi yang harus diperhatikan seiring dengan peningkatan teknik.

13.6 MASA DEPAN : BIDANG CAHAYA

Bidang cahaya adalah teknologi baru yang menarik yang menggunakan cincin kamera yang berputar untuk menangkap semua cahaya yang memasuki area bola. Ini menghasilkan area yang ditangkap di mana pemirsa memiliki paralaks penuh. Sedangkan dengan video mono spherical, pemirsa dibatasi pada satu perspektif, bidang cahaya memungkinkan pengguna untuk menerjemahkan kepala mereka untuk melihat berbagai sudut pemandangan, seperti yang ditunjukkan pada Gambar 13-17.



Gambar 13-17. Bidang cahaya memungkinkan terjemahan kepala untuk memberikan perspektif yang berbeda

Paralaks penuh sangat meningkatkan rasa kehadiran pemirsa, kunci untuk menciptakan pengalaman pelatihan berkualitas lebih tinggi. Jika Anda belum sempat mencobanya, cobalah demonstrasi "Selamat Datang di Bidang Cahaya" Google. Hasilnya mencolok.

Pada 2018, masih ada terlalu banyak batasan pada teknologi ini untuk diadopsi secara luas. Area yang direkam berdiameter sekitar dua kaki, yang masih cukup kecil sehingga kepala pengguna dapat dengan mudah keluar dari ruang. Kecepatan rotasi perlengkapan kamera juga berarti bahwa hanya gambar diam yang dapat ditangkap; cocok untuk beberapa pengalaman pelatihan yang berfokus pada lingkungan statis tetapi kurang berguna untuk apa pun yang melibatkan aktor atau gerakan dinamis. Namun, bidang cahaya adalah teknologi penangkapan yang harus diperhatikan.

13.7 MASA DEPAN: TRAINING AR

VR adalah teknologi yang, di mana pun Anda berada, membawa Anda ke tempat lain dan memungkinkan Anda mengalaminya. AR adalah teknologi yang mengubah persepsi Anda

tentang dunia Anda yang ada; dibutuhkan realitas dan kemudian menambahnya. VR bersifat transportasi; AR adalah transformasional.

Tempat alami AR adalah bantuan di tempat kerja, tetapi ada beberapa teknologi utama yang belum cukup matang. Agar AR dapat memberikan bantuan di tempat kerja yang berguna, ia harus dapat melakukan sesuatu yang tidak dapat dilakukan oleh otak manusia, dengan akurasi yang cukup agar aman dan berguna. Kami semakin dekat. Kami memiliki perangkat lunak yang dapat menerjemahkan teks dan menggantinya dengan cepat, dan pengenalan gambar yang dapat bertukar wajah. Tetapi akurasi 99% tidak cukup baik untuk banyak kasus penggunaan; kita harus 100% akurat. Perangkat keras juga menjadi masalah. Headset AR harus lebih nyaman dan ringan sebelum orang tertarik untuk memakainya selama delapan jam setiap hari. Terlepas dari semua itu, ada potensi AR dalam pelatihan berbasis lokasi khusus. Ambil rumah banjir, lalu bayangkan lingkungan pelatihan di mana AR mengajari Anda dan melatih Anda di lingkungan nyata. Ini akan memberikan manfaat menjelajahi ruang pelatihan fisik yang nyata dengan kemampuan untuk memberikan umpan balik dan bantuan individual.

Teknologi serupa sudah digunakan. Museum dan tujuan wisata menggunakan perangkat tur audio yang mendeteksi lokasi Anda. Teknologi hebat lainnya untuk dilihat adalah The VOID,¹² yang menggunakan campuran VR yang dapat dipakai, tracking lokasi, dan ruang fisik untuk menciptakan pengalaman hiburan yang imersif.

13.8 MASA DEPAN : PENGENALAN SUARA

Antarmuka suara akan menjadi tambahan yang sangat besar untuk perangkat pelatihan saat kita dapat melewati ambang batas akurasi akhir itu. Aspek penting dari pelatihan yang kami lakukan adalah antarpribadi: pelatihan tentang layanan pelanggan atau resolusi konflik antarpribadi. Pengenalan suara dapat memberikan media dan skema kontrol yang sempurna segera setelah diintegrasikan ke dalam perangkat HMD, tetapi pertama-tama harus cukup andal untuk bekerja dalam semua kondisi, termasuk ruang belakang ritel yang bising.

13.9 MASA DEPAN : SKENARIO TRAINING IDEAL

Bayangkan sejenak masa depan. Dunia tanpa batas, dengan lensa kontak AR dan AI yang kuat. Seperti apa bentuk pelatihannya? Maria Poppin. Jangan tertawa! Mary Poppins keluar dari langit dengan payung ajaib dan mengubah gerombolan anak-anak yang mengerikan menjadi warga teladan. Anak-anak bahkan tidak menyadari bahwa mereka sedang dilatih, karena pelatihan itu menyenangkan. Mary Poppins mencapai keseimbangan sempurna antara membimbing, menantang, dan memelihara tanggung jawabnya.

Inilah yang diperlukan untuk mendapatkan pengalaman pelatihan yang sempurna. AI yang kuat yang dapat memahami siapa pelajar itu dan melatih mereka berdasarkan kebutuhan mereka. Skenario semacam ini jauh. Pada tahun 2018, lensa kontak AR dan AI yang kuat tidak ada. Tetapi mempertimbangkan dunia yang ideal dan memikirkan ke mana arah pelatihan di masa depan dapat menjadi alat yang hebat untuk mencari tahu bagaimana menuju ke sana,

terutama ketika mempertimbangkan teknologi apa yang akan diinvestasikan dan bagaimana mempersiapkannya.

13.10 KESIMPULAN

Pikirkan ini ketika Anda memulai produk Anda. Saat Anda membangun, saat Anda membuat untuk olahraga, fokuslah untuk membawa penggemar ke dalam perjalanan atlet yang bangkit untuk memenuhi momen itu. Ini bisa berupa permainan bisbol fantasi, aplikasi pelatihan, atau streaming langsung —apa pun yang Anda lakukan, ciptakan sesuatu yang membuat orang ingin terus kembali lagi dan lagi.

Juga, jangan bagikan bisnis Anda. Kesalahan terbesar yang dapat Anda lakukan pada saat kipas terpikat pada produk Anda—saat konsumen siap untuk saat itu—adalah menunjukkan kepada mereka model bisnis Anda atau menunggu sesuatu rusak.

Ini adalah waktu paling menakutkan dalam sejarah untuk mengerjakan produk olahraga. Kecepatan teknologi mampu mengejar dan meningkatkan kecepatan siaran langsung olahraga. Dengan begitu banyak pengalaman konten yang direkam, pengalaman yang benar-benar hiduplah yang memengaruhi sebagian besar kehidupan orang. Saat Anda membangun produk Anda, fokus pada momen, fokus pada latensi, fokus pada cerita, dan berikan kontrol kepada konsumen. Fokus pada penggemar: mereka membuat permainan menjadi menarik, mereka mengikuti cerita, dan merekalah yang paling penting.

DAFTAR PUSTAKA

- BPS. (2020). Perkembangan Pariwisata dan Transportasi Sulawesi Selatan Maret 2020. No. 26/05/73/Th.XIV
- Cearley D, Burke B. Top 10 Strategic Technology Trends for 2019 Stamford: Gartner, Inc; 2018.
- Djafar I, Simpen IW, Salmiati. Perancangan Aplikasi Virtual Reality Objek 3D pada Video Promosi Perumahan Zarindah Berbasis Android. In Prosiding Seminar Ilmiah Sistem Informasi dan Teknologi Informasi; 2019; Makasar: STMIK Dipanegara Makassar. p. 161 - 172.
- Faizin, A.A., Somantri, M., Christyono, Y. (2017). Aplikasi Media Pemasaran Properti dengan Menggunakan Teknologi Augmented reality pada Perangkat Android. Jurnal Transient: Jurnal Ilmiah Teknik Elektro Vol. 6 No. 3. pp 341-348. November 2017. <https://doi.org/10.14710/transient.6.3.341-348>
- Haryati S. Research and Development (R&D) sebagai Salah Satu Model Penelitian dalam Bidang Pendidikan. Jurnal Edutic Vol.37 No.1. 2012;; p. 11-26.
- Ismayani, A. (2020). Membuat Sendiri Aplikasi Augmented reality. Jakarta: Elex Media
- Kemp, Simon. (2020). Digital 2020: INDONESIA. <https://datareportal.com/reports/digital-2020-indonesia>. Diakses pada 24 Mei 2020
- Kotler, P., Armstrong. (2012). Marketing Management. 14th Edition. Pearson Prentice Hill
- Kotler, P., Kartajaya, H., Setiawan, I. (2020). Marketing 4.0 Bergerak Dari Tradisional Ke Digital. Jakarta: PT. Gramedia Pustaka
- Kotler, P., Kartajaya, H., Setiawan, I. (2021). Marketing 5.0 Technology for Humanity. New Jersey: Wiley
- Kotler, P., Keller, K.L. (2009). Manajemen Pemasaran Jilid 1 Edisi 13. Jakarta: Erlangga
- Prabowo AZ, Satoto KI, Teguh KM. Perancangan dan Implementasi Augmented Reality sebagai Media Promosi Penjualan Perumahan. Jurnal Teknologi dan Sistem Komputer, Vol.3, No.1, Januari 2015. 2015;; p. 161-170.
- Purnomo, A., Haryanto, H., Aplikasi Augmented reality Sebagai Alat Pengukur Baju Wisudawan Wisudawati di Universitas Dian Nuswantoro. Seminar Nasional Teknologi Informasi & Komunikasi Terapan Vol. 2 No. 1. 23 Juni 2012
- Putri RIP, Irawati DA, Aji DKP. Implementasi Augmented Reality pada Media Promosi Perumahan Patraland Place. Jurnal Informatika Polinema, Volume 1, Edisi 1, November 2014. 2014;; p. 36-40.

Sri BN, Kurniawan A. Rancang Bangun 3D Virtual Reality untuk Promosi Perumahan Berbasis Web Online. In Seminar Nasional Teknologi Informasi dan Multimedia 2014; 2014; Yogyakarta: STMIK AMIKOM Yogyakarta. p. 25-30.

Sucahyowati H. Pengantar Manajemen Malang: Wilis; 2017.

Susanto AB, Wijarnako H. Power Branding: Membangun Merek Unggul dan Organisasi Pendukungnya Bandung: Mizan Pustaka; 2004.