



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

Teori & Praktik AR dan VR (Augmented Reality dan Virtual Reality)



Teori & Praktik AR dan VR (Augmented Reality dan Virtual Reality)

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

Bio Data Penulis



Penulis lahir di Semarang pada tanggal 1 Maret 1983. Penulis menempuh pendidikan Sarjana Teknik Elektro di Universitas Kristen Satya Wacana (UKSW), lulus tahun 2004, kemudian tahun 2005 melanjutkan studi pada Magister Desain di Fakultas Seni Rupa dan Desain, Institut Teknologi Bandung (ITB), dan kemudian melanjutkan studi pada program studi Teknologi Multimedia di Swinburne University of Technology Australia. Penulis sejak tahun 2010, menjadi dosen pada program studi Desain Grafis Universitas Sains dan Teknologi Komputer (Universitas STEKOM), memiliki Jabatan Akademik Lektor Kepala 700. Penulis juga seorang wirausaha di bidang toko online yang berhasil di kota Semarang dan juga aktif sebagai freelancer dalam bidang fotografi, web design dan multimedia.



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

Teori & Praktik AR dan VR (Augmented Reality dan Virtual Reality)

Penulis :

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

ISBN :

Editor :

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

Penyunting :

Dr. Joseph Teguh Santoso, M.Kom.

Desain Sampul dan Tata Letak :

Irdha Yuniyanto, S.Ds., M.Kom.

Penebit :

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

Anggota IKAPI No: 279 / ALB / JTE / 2023

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 kepada Tuhan Yang Maha Esa, atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan buku dengan judul ***Teori dan Praktik AR (Augmented Reality) dan VR (Virtual Reality)***. Buku ini disusun untuk memberikan pemahaman yang lebih mendalam mengenai perkembangan teknologi Augmented Reality (AR) dan Virtual Reality (VR), serta penerapan praktis keduanya dalam berbagai bidang kehidupan.

AR dan VR adalah dua teknologi yang telah membawa perubahan signifikan dalam cara kita berinteraksi dengan dunia digital. Augmented Reality, yang menggabungkan elemen-elemen dunia nyata dengan objek virtual, memberikan pengalaman yang memperkaya persepsi pengguna. Sementara itu, Virtual Reality menciptakan dunia imersif yang sepenuhnya baru, memungkinkan pengguna untuk merasakan pengalaman seperti berada di dalam lingkungan yang sepenuhnya diciptakan oleh komputer. Kedua teknologi ini memiliki potensi besar dalam berbagai sektor, seperti pendidikan, hiburan, medis, serta industri kreatif dan teknologi.

Dalam buku ini, penulis akan membahas teori dasar yang mendasari perkembangan AR dan VR, serta bagaimana teknologi-teknologi ini diterapkan dalam kehidupan sehari-hari, baik dalam konteks industri maupun pengembangan produk. Penulis juga akan menggali berbagai tantangan dan peluang yang dihadapi dalam penerapan AR dan VR, serta tren-tren masa depan yang dapat mengubah cara kita bekerja, belajar, dan berinteraksi.

Penulis berharap buku ini dapat memberikan wawasan yang bermanfaat bagi pembaca dalam memahami lebih dalam tentang AR dan VR, serta menginspirasi eksplorasi lebih lanjut dalam memanfaatkan kedua teknologi ini. Semoga buku ini bermanfaat bagi perkembangan pengetahuan dan teknologi, serta memberikan kontribusi positif untuk kemajuan dunia digital di masa depan. Terima kasih

Semarang, Januari 2025

Penulis

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

DAFTAR ISI

Halaman Judul	i
Kata Pengantar	ii
Daftar Isi	iii
BAB 1 BAGAIMANA MANUSIA BEIRNTERAKSI DENGAN KOMPUTER	1
1.1 Pendahuluan	1
1.2 Modalitas Sepanjang Zaman	2
1.3 Perangkat keras, tombol, haptik, objek nyata	11
1.4 Teknologi Pelacakan Tubuh	18
1.5 Input Suara, Tangan, dan Perangkat Keras pada Generasi Berikutnya	19
BAB 2 Merancang Teknologi dengan Memahami Indra Manusia.....	23
2.1 Membayangkan Masa Depan	23
2.2 Peran Desainer dan Tim di Masa Depan	28
2.3 Desain Sensorik	30
2.4 Kisah AR Adobe	36
BAB 3 REALITAS VIRTUAL UNTUK SENI	38
3.1 Cara yang Lebih Alami untuk Membuat Seni 3D.....	38
3.2 VR untuk Animasi	45
BAB 4 OPTIMALISASI SENI 3D	47
4.1 Pendahuluan.....	47
4.2 Anggaran hitungan poligon	49
4.3 Menggunakan Alat VR untuk Membuat Seni 3D.....	59
BAB 5 PERAN COMPUTER VISION DALAM MENGAKTIFKAN AR	62
5.1 Sejarah Singkat AR	63
5.2 Mengintegrasikan Perangkat Keras dan Perangkat Lunak	66
5.3 Pemetaan.....	74
5.4 Cara Kerja AR Multipemain	77
5.5 Relokalisasi Multipemain.....	84
5.6 Privasi dan Data AR Cloud	102
BAB 6 REALITAS VIRTUAL DAN REALITAS TERTAMBAH	106
6.1 Mengapa Lintas Platform?.....	106
6.2 Peran Mesin Permainan	108
6.3 Memahami Grafik 3D	109
6.4 Portabilitas dari Desain Gim Video	113
6.5 Menyederhanakan Input Kontroler	115
BAB 7 VIRTUAL REALITY TOOLKIT	123
7.1 Sejarah VR TK.....	123
7.2 Keberhasilan VR TK.....	128
7.3 Memeriksa Repositori VR TK v4	131
BAB 8 PRAKTIK PENGEMBANGAN VR DAN AR	135
8.1 Menangani Pergerakan	136
8.2 Pergerakan dalam VR	137

8.3	Penggerakan teleportasi	140
8.4	Penggerak dalam AR.....	143
8.5	Penggunaan Audio yang Efektif	144
8.6	Paradigma Interaksi Umum	148
8.7	Raycast Realitas Tambah	151
BAB 9	DESAIN VISUALISASI DATA DALAM KOMPUTASI SPASIAL	154
9.1	Pendahuluan	154
9.2	Memahami Visualisasi Data	154
9.3	Evolusi Desain Visualisasi Data dengan Munculnya XR	157
9.4	Visualisasi Data 2D versus Visualisasi Data 3D dalam Komputasi Spasial	160
9.5	Desain Visualisasi Data yang Baik Mengoptimalkan Ruang 3D	164
9.6	Cara Membuat Visualisasi Data: Alur Pembuatan Visualisasi Data	166
9.7	Melihat Lebih Dekat Glass Brain	170
BAB 10	KARAKTER AI DAN PERILAKU	177
10.1	Pendahuluan	177
10.2	Perilaku	180
10.3	Kompleksitas dan Universalitas	185
10.4	Lebih Banyak Kecerdasan dalam Sistem	187
10.5	Pembelajaran Mesin	195
10.6	Menggabungkan Perencanaan Otomatis Dan Pembelajaran Mesin.....	199
BAB 11	EKOSISTEM TEKNOLOGI KESEHATAN REALITAS VIRTUAL DAN TERTAMBAH ..	203
11.1	Desain Aplikasi Teknologi Kesehatan VR/AR	203
11.2	Pengalaman Pengguna Standar Tidaklah Intuitif	205
11.3	Perusahaan	211
11.4	Kesehatan Proaktif	215
BAB 12	PENGALAMAN PENGGEMAR: SPORTSXR	220
12.1	Pendahuluan	220
12.2	Lima Prinsip Utama AR dan VR untuk Olahraga	222
12.3	Evolusi Berikutnya dalam Pengalaman Olahraga	226
12.4	Menciptakan Masa Depan	227
BAB 13	KASUS PENGGUNAAN PELATIHAN PERUSAHAAN REALITAS VIRTUAL	234
13.1	Pentingnya Pelatihan Perusahaan	234
13.2	Manfaat Pelatihan VR? R.I.D.E.	240
13.3	Apa yang Membuat Pelatihan VR Baik?	241
13.4	Video Sferis	242
13.5	Masa Depan Pelatihan XR.....	250
Daftar Pustaka	255

BAB 1

BAGAIMANA MANUSIA BERINTERAKSI DENGAN KOMPUTER

Definisi Istilah Umum

Saya menggunakan istilah-istilah berikut dengan cara-cara khusus yang mengasumsikan elemen yang dapat dirasakan manusia:

- Modalitas saluran input dan output sensorik antara komputer dan manusia
- Affordances atribut atau karakteristik objek yang menentukan potensi penggunaan objek tersebut
- Input cara anda melakukan hal-hal tersebut; data yang dikirim ke komputer
- Output reaksi yang dapat dirasakan terhadap suatu peristiwa; data yang dikirim dari komputer
- Umpan balik sejenis keluaran; konfirmasi bahwa apa yang anda lakukan diperhatikan dan ditindaklanjuti oleh pihak lain

1.1 PENDAHULUAN

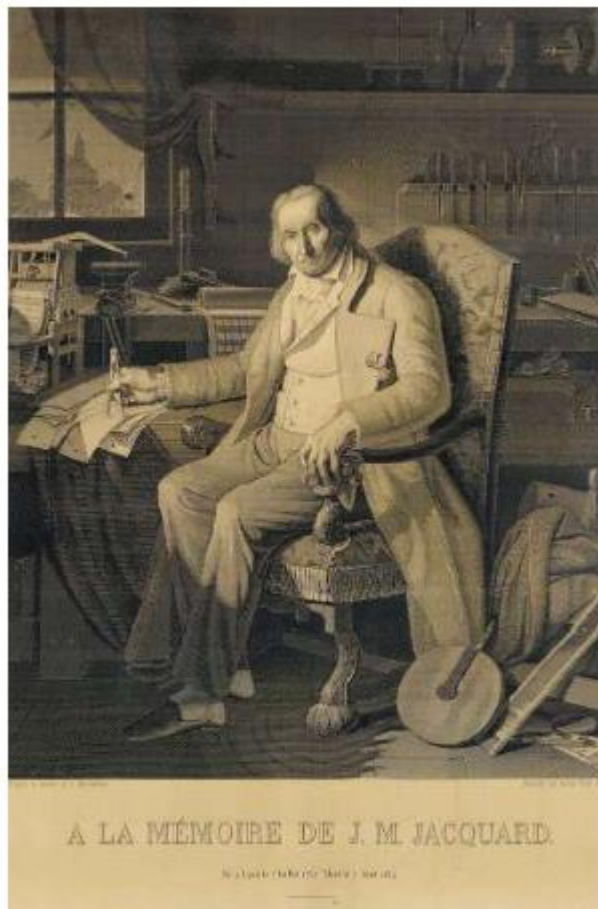
Dalam permainan Dua Puluh Pertanyaan, tujuan Anda adalah menebak objek apa yang dipikirkan orang lain. Anda dapat menanyakan apa saja yang Anda inginkan, dan orang lain harus menjawab dengan jujur; kendalanya adalah mereka menjawab pertanyaan hanya menggunakan satu dari dua pilihan: ya atau tidak. Melalui serangkaian kejadian 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 pemahaman yang melekat tentang dunia atau, memang, apa pun di luar biner atau, dalam kasus komputer kuantum, probabilitas. Karena itu, kita mengomunikasikan segalanya kepada komputer, dari konsep hingga masukan, melalui peningkatan tingkat abstraksi yang ramah manusia yang mencakup lapisan komunikasi dasar: satu dan nol, atau ya dan tidak.

Jadi, sebagian besar pekerjaan komputasi saat ini adalah menentukan cara membuat manusia menjelaskan ide-ide yang semakin kompleks kepada komputer dengan mudah dan sederhana. Pada gilirannya, manusia juga berupaya agar komputer memproses ide-ide tersebut lebih cepat dengan membangun lapisan abstraksi tersebut di atas angka satu dan nol. Ini adalah siklus masukan dan keluaran, kemampuan dan umpan balik, di seluruh modalitas. Lapisan abstraksi dapat mengambil banyak bentuk: metafora antarmuka pengguna grafis, kata-kata lisan dari pemrosesan bahasa alami (NLP), pengenalan objek dari visi komputer, dan, yang paling sederhana dan umum, masukan sehari-hari dari papan ketik dan penunjuk, yang digunakan sebagian besar manusia untuk berinteraksi dengan komputer setiap hari.

1.2 MODALITAS SEPANJANG ZAMAN

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, terkenal "membaca" kartu berlubang. Joseph Jacquard menciptakan apa yang, pada dasarnya, merupakan salah satu karya seni mekanis sejati pertama, potret dirinya, menggunakan kartu berlubang pada tahun 1839 (Gambar 1.1). Sekitar waktu yang sama di Rusia, Semyon Korsakov menyadari bahwa kartu berlubang dapat digunakan untuk menyimpan dan membandingkan kumpulan data.



Gambar 1.1 Potret sutra tenun Joseph Jacquard, 1839, yang menggunakan lebih dari 24.000 kartu berlubang untuk membuat potret tersebut

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

Ini telah lama menjadi masalah yang diketahui. Faktanya, sejak abad ketujuh belas

yaitu, segera setelah teknologi tersedia orang-orang mulai membuat papan ketik. Orang-orang menciptakan dan menciptakan kembali papan ketik untuk berbagai alasan; misalnya, untuk melawan pemalsuan, membantu saudara perempuan yang tuna netra, dan buku-buku yang lebih bagus. Memiliki bidang penyangga untuk meletakkan tangan dan pergelangan tangan memungkinkan gerakan yang tidak konsisten untuk menghasilkan hasil yang konsisten yang tidak mungkin dicapai dengan pena. Seperti yang disebutkan sebelumnya, proto-komputer memiliki motivasi yang sama menariknya: komputer membutuhkan data fisik yang sangat konsisten, dan manusia tidak nyaman untuk membuat data yang konsisten. Jadi, meskipun mungkin tampak mengejutkan jika dipikir-pikir kembali, pada awal tahun 1800-an, mesin kartu berlubang, yang belum menjadi monster kalkulasi seperti sekarang, sudah memiliki papan ketik 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

Papan ketik telah dipasangkan pada perangkat komputasi sejak awal, tetapi, tentu saja, papan ketik diperluas ke mesin ketik sebelum kembali lagi saat kedua teknologi tersebut bergabung. Kegesitan juga dikaitkan dengan konsistensi dan kelelahan manusia. Dari Wikipedia:

Pada pertengahan abad ke-19, meningkatnya kecepatan komunikasi bisnis telah menciptakan kebutuhan akan mekanisasi proses penulisan. Stenografer dan telegrafer dapat mencatat informasi dengan kecepatan hingga 130 kata per menit. Sebaliknya, menulis dengan pena hanya menghasilkan sekitar 30 kata per menit: menekan 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

menyederhanakan desain dan ukuran mesin ketik awal. Penulis ingin berhenti sejenak di sini untuk menunjukkan masalah yang lebih luas yang coba dipecahkan semua orang dengan menggunakan mesin ketik, dan khususnya dengan papan ketik sebagai input: pada level tertinggi, orang ingin menuangkan ide mereka dengan lebih cepat dan lebih akurat. Ingat ini; ini adalah tema yang konsisten di semua peningkatan modalitas.

Modalitas Sepanjang Masa: Hingga Perang Dunia II

Begitulah pembahasan tentang keyboard, yang, seperti yang baru saja Penulis tunjukkan, telah ada bersama kita sejak manusia pertama kali mencoba berkomunikasi dengan mesin. Sejak awal abad kedua puluh yaitu, sekali lagi, segera setelah teknik pengerjaan logam dan manufaktur mendukungnya kita memberi mesin cara untuk berkomunikasi kembali, untuk berdialog dengan operatornya sebelum tahap keluaran fisik yang mahal: monitor dan display, bidang yang diuntungkan oleh penelitian dan sumber daya yang signifikan selama era perang melalui anggaran militer.

Display komputer pertama tidak menampilkan kata-kata: panel komputer awal memiliki bohlam lampu kecil yang akan menyala dan mati untuk mencerminkan status tertentu, yang memungkinkan teknisi untuk memantau status komputer dan mengarah pada penggunaan kata "monitor." Selama Perang Dunia II, badan militer menggunakan layar tabung sinar katode (CRT) untuk cakupan radar, dan segera setelah perang, CRT mulai digunakan sebagai tampilan komputasi vektor, dan kemudian teks, untuk kelompok-kelompok seperti SAGE dan Royal Navy.

Begitu mesin komputasi dan pemantauan memiliki layar, kami memiliki input khusus layar untuk menyertainya. Joystick diciptakan untuk pesawat terbang, tetapi penggunaannya untuk mengemudikan pesawat terbang jarak jauh dipatenkan di Amerika Serikat pada tahun 1926. Hal ini menunjukkan keanehan fisiologi manusia: kita mampu secara naluriah memetakan kembali propriosepsi rasa kita akan orientasi dan penempatan tubuh kita ke volume dan sudut bidang yang baru (lihat Gambar 1-3). Jika kita tidak mampu melakukannya, mustahil untuk menggunakan tetikus di desktop pada bidang Z untuk menggerakkan jangkar tetikus di bidang X. Namun, kita dapat melakukannya hampir tanpa berpikir—meskipun beberapa dari kita mungkin perlu membalikkan rotasi sumbu untuk meniru pemetaan internal kita sendiri.



Gambar 1.3 Contoh antarmuka komputer awal untuk pemetaan ulang proprioseptif; operator radar WAAF Denise Miley sedang memetakan pesawat di Ruang Penerima di stasiun "Chain Home" Bawdsey pada bulan Mei 1945 (perhatikan kenop besar di sebelah kirinya, sebuah kontrol goniometer yang memungkinkan Miley mengubah sensitivitas pencari arah radio)

Modalitas Sepanjang Masa: Pasca-Perang Dunia II

Joystick dengan cepat berpindah dari pesawat terbang dan berdampingan dengan tampilan radar dan sonar selama Perang Dunia II. Segera setelah perang, pada tahun 1946, input khusus tampilan pertama diciptakan. Ralph Benjamin, seorang insinyur di Angkatan Laut Kerajaan, membayangkan rollerball sebagai alternatif untuk input joystick yang ada: "Pelacak bola yang elegan berdiri di samping tampilan arah pesawatnya. Dia memiliki satu bola, yang dipegangnya di tangannya, tetapi joystick-nya telah layu." Indikasinya tampaknya adalah bahwa rollerball dapat dipegang di tangan daripada diletakkan di atas meja. Namun, realitas produksi pada tahun 1946 berarti bahwa roller asli adalah bola bowling berukuran penuh. Tidak mengherankan, rollerball yang beratnya 10 pon tidak menggantikan joystick.

Ini membawa kita pada lima aturan popularitas input komputer. Agar berhasil, input harus memiliki karakteristik berikut:

- ✓ Murah
- ✓ Andal
- ✓ 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 yang tidak merusak, tetapi hati-hati: setelah titik tertentu, bahkan kesalahan yang tidak berbahaya pun dapat mengganggu. Koreksi otomatis pada layar sentuh

adalah contoh bagus tentang kesalahan pengguna yang sering kali mengalahkan kemampuan perangkat lunak.

Meskipun mouse rollerball tidak akan ada di mana-mana hingga tahun 1984 dengan munculnya komputer pribadi, banyak jenis input lain yang digunakan dengan komputer berpindah dari militer hingga pertengahan 1950-an dan masuk ke sektor swasta: joystick, tombol dan sakelar, dan, tentu saja, keyboard.

Mungkin mengejutkan mengetahui bahwa stylus ada sebelum mouse. Pena cahaya, atau pistol, yang diciptakan oleh SAGE pada tahun 1955, adalah stylus optik yang diatur waktunya untuk siklus penyegaran CRT dan dapat digunakan untuk berinteraksi langsung pada monitor. Pilihan lain yang mirip tetikus, Grafacon dari Data Equipment Company, menyerupai balok pada poros yang dapat diayunkan untuk menggerakkan kursor. Bahkan ada pekerjaan yang dilakukan pada perintah suara sejak tahun 1952 dengan sistem Audrey dari Bell Labs, meskipun hanya mengenali 10 kata.

Pada tahun 1963, perangkat lunak grafis pertama ada yang memungkinkan pengguna untuk menggambar pada monitor TX-2 milik MIT Lincoln Laboratory, Sketchpad, yang diciptakan 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 pena cahaya, yang mengharuskan menunggu CRT untuk menyegarkan. Sayangnya, baik pada pena cahaya maupun kotak pensil logam, layarnya tegak sehingga pengguna harus mengangkat lengannya untuk memasukkan input yang kemudian dikenal sebagai "lengan gorila" yang terkenal itu. Latihan yang hebat, tetapi ergonominya buruk. Perusahaan RAND telah menyadari masalah ini dan telah mengerjakan solusi tablet dan stylus selama bertahun-tahun, tetapi harganya tidak murah: pada tahun 1964, stylus RAND yang membingungkan, kemudian dipasarkan juga sebagai Grafacon dihargai sekitar \$18.000 (sekitar \$150.000 dalam dolar tahun 2018). Butuh waktu bertahun-tahun sebelum kombinasi tablet dan stylus mulai populer, jauh setelah mouse dan sistem antarmuka pengguna grafis (GUI) dipopulerkan.

Pada tahun 1965, Eric Johnson, dari Royal Radar Establishment, menerbitkan sebuah makalah tentang perangkat layar sentuh kapasitif dan menghabiskan beberapa tahun berikutnya untuk menulis kasus penggunaan yang lebih jelas tentang topik tersebut. Hal ini diambil oleh para peneliti di Organisasi Riset Nuklir Eropa (CERN), yang menciptakan versi yang berfungsi pada tahun 1973. Pada tahun 1968, Doug Engelbart siap untuk menunjukkan pekerjaan yang telah dilakukan oleh labnya, Pusat Riset Augmentasi, di Institut Riset Stanford sejak tahun 1963. Di sebuah aula di bawah Civic Center San Francisco, ia mendemonstrasikan Sistem oNLine (NLS) milik timnya dengan sejumlah fitur yang sekarang menjadi standar dalam komputasi modern: kontrol versi, jaringan, konferensi video, email multimedia, beberapa jendela, dan integrasi tetikus yang berfungsi, di antara banyak lainnya. Meskipun NLS juga memerlukan keyboard kord dan keyboard konvensional untuk input, tetikus sekarang sering disebut sebagai salah satu inovasi utama. Faktanya, tetikus NLS dinilai dapat digunakan dengan baik seperti pena cahaya atau sistem input lutut milik ARC dalam penelitian tim Engelbart sendiri. Hal itu juga tidak unik: produsen radio dan TV asal Jerman, Telefunken, merilis tetikus

dengan RKS 100-86, Rollkugel, yang sebenarnya diproduksi secara komersial pada tahun Engelbart mengumumkan prototipenya.

Namun, Engelbart tentu saja memopulerkan gagasan tentang masukan komputer bentuk bebas asimetris. Perancang tetikus yang sebenarnya di ARC, Bill English, juga mengemukakan salah satu kebenaran modalitas digital pada akhir makalahnya tahun 1967, "Teknik Pemilihan Tampilan untuk Manipulasi Teks":

[S]ampaknya tidak realistis untuk mengharapkan pernyataan tegas bahwa satu perangkat lebih baik daripada yang lain. Rincian sistem penggunaan tempat perangkat akan ditanamkan membuat terlalu banyak perbedaan. Seberapa pun bagusya perangkat keras, aspek terpenting adalah bagaimana perangkat lunak menginterpretasikan masukan perangkat keras dan menormalkannya sesuai dengan maksud pengguna.

Untuk informasi lebih lanjut tentang bagaimana desain perangkat lunak dapat memengaruhi persepsi pengguna terhadap masukan, Penulis sangat merekomendasikan buku *Game Feel: A Game Designer's Guide to Virtual Sensation* karya Steve Swink (Morgan Kaufmann Game Design Books, 2008). Karena setiap permainan memiliki dunia dan sistemnya sendiri, "nuansa" masukan dapat dipikirkan kembali. Ruang gerak untuk inovasi dalam sistem operasi komputer standar semakin sempit, yang secara default harus terasa familier untuk menghindari kelebihan beban kognitif.

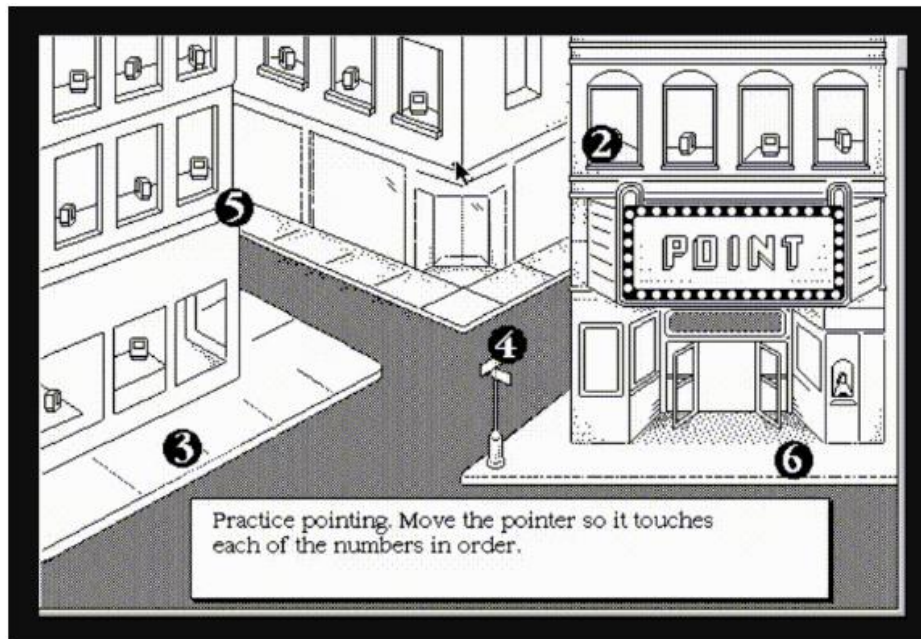
Aspek lain dari kemajuan teknologi yang perlu diperhatikan sejak tahun 1960-an adalah munculnya fiksi ilmiah, dan karenanya komputasi, dalam budaya populer. Acara TV seperti *Star Trek* (1966–1969) menggambarkan penggunaan perintah suara, telepresensi, jam tangan pintar, dan komputer mini. *2001: A Space Odyssey* (1968) menunjukkan perangkat komputasi personal kecil yang tampak sangat mirip dengan iPad masa kini 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 asisten robotik. Meskipun teknologinya tidak umum atau bahkan tersedia, orang-orang mulai terbiasa 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 personal. Konsol gim rumahan mulai diproduksi secara komersial, dan arkade mulai menjamur. Komputer semakin terjangkau; tersedia di universitas-universitas terkemuka, dan lebih umum di tempat-tempat komersial. Joystick, tombol, dan sakelar dengan mudah beralih ke input gim video dan memulai lintasannya sendiri yang terpisah sebagai pengontrol gim. Pusat Penelitian Palo Alto, atau PARC, milik Xerox Corporation yang terkenal, mulai mengerjakan sistem kerja komputer GUI dan tetikus terpadu yang disebut Alto. Alto dan penggantinya, Star, sangat berpengaruh pada gelombang pertama komputer pribadi yang diproduksi oleh Apple, Microsoft, Commodore, Dell, Atari, dan lainnya pada awal hingga pertengahan 1980-an. PARC juga membuat prototipe *KiddiComp/Dynabook* tahun 1968 karya Alan Kay, salah satu pendahulu tablet komputer modern.

Modalitas Sepanjang Masa: Kebangkitan Komputasi Pribadi

Sering kali, orang menganggap tetikus dan GUI sebagai tambahan yang besar dan

independen pada modalitas komputer. Namun, bahkan pada 1970-an, Summagraphics membuat kombinasi tablet dan stylus kelas bawah dan atas untuk komputer, salah satunya diberi label putih untuk Apple II sebagai Apple Graphics Tablet, yang dirilis pada 1979. Harganya relatif mahal dan hanya didukung oleh beberapa jenis perangkat lunak; melanggar dua dari lima aturan. Pada tahun 1983, HP telah merilis HP-150, komputer layar sentuh pertama. Akan tetapi, ketepatan pelacakannya cukup rendah, sehingga melanggar aturan kesalahan pengguna.



Gambar 1.4 Cuplikan Layar Macintosh SE Tour, 1987

Ketika tetikus pertama kali disertakan dalam paket komputer pribadi (1984–1985), tetikus didukung pada level sistem operasi (OS), yang pada gilirannya dirancang untuk menerima masukan tetikus. Ini merupakan titik balik utama bagi komputer: tetikus bukan lagi masukan opsional, tetapi masukan yang penting. Alih-alih sekadar barang aneh atau periferal opsional, komputer kini diharuskan dilengkapi dengan tutorial yang mengajarkan pengguna cara menggunakan tetikus, seperti yang diilustrasikan pada Gambar 1.4 mirip dengan cara gim video menyertakan tutorial yang mengajarkan pemain cara tindakan gim dipetakan ke tombol pengontrol.

Sangat mudah untuk melihat kembali tahun 1980-an dan berpikir bahwa komputer pribadi adalah inovasi yang berdiri sendiri. Namun, secara umum, hanya ada sedikit inovasi dalam komputasi yang secara langsung memajukan bidang ini dalam waktu kurang dari satu dekade. Bahkan inovasi yang 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 hanya merupakan hasil dari teknologi yang akhirnya memenuhi lima aturan yang disebutkan di atas: murah, andal, nyaman, memiliki perangkat lunak yang memanfaatkan teknologi tersebut, dan memiliki tingkat kesalahan pengguna yang dapat diterima.

Sangat umum untuk menemukan bahwa versi pertama dari apa yang tampaknya merupakan teknologi terkini sebenarnya telah ditemukan beberapa dekade atau bahkan berabad-abad yang lalu. Jika teknologi tersebut cukup jelas sehingga banyak orang mencoba membuatnya tetapi tetap tidak berhasil, kemungkinan besar teknologi tersebut gagal dalam salah satu dari lima aturan tersebut. Kita hanya perlu menunggu hingga teknologi meningkat atau proses produksi menyusul.

Kebenaran ini tentu saja dicontohkan dalam sejarah realitas virtual (VR) dan realitas tertambah (AR). Meskipun tampilan kepala stereoskopik (HMD) pertama dirintis oleh Ivan Sutherland pada tahun 1960-an dan telah digunakan di NASA secara rutin sejak tahun 1990-an, baru setelah bidang elektronik seluler dan unit pemrosesan grafis (GPU) yang canggih meningkat pesat, teknologi tersebut tersedia dengan harga yang dapat diterima secara komersial, beberapa dekade kemudian. Bahkan hingga saat ini, HMD mandiri kelas atas harganya ribuan dolar atau tidak tersedia secara komersial. Namun, seperti halnya telepon pintar di awal tahun 2000-an, kita dapat melihat jalur yang jelas dari perangkat keras saat ini menuju masa depan komputasi spasial. Namun, sebelum kita menyelami perangkat keras saat ini, mari kita selesaikan pembahasan tentang jalur dari PC di awal tahun 1980-an hingga jenis komputer yang paling umum saat ini: telepon pintar.

Modalitas Sepanjang Masa: Miniaturisasi Komputer

Komputer dengan perangkat keras miniatur muncul dari industri kalkulator dan komputer pada awal tahun 1984 dengan Psion Organizer. Komputer tablet pertama yang sukses adalah GriDPad, dirilis pada tahun 1989, yang VP penelitiannya, Jeff Hawkins, kemudian mendirikan PalmPilot. Apple merilis Newton pada tahun 1993, yang memiliki sistem input karakter tulisan tangan, tetapi tidak pernah mencapai target penjualan utama. Proyek tersebut berakhir pada tahun 1998 sebagai Nokia 900 Communicator kombinasi telepon dan asisten digital pribadi (PDA) dan kemudian PalmPilot mendominasi lanskap komputer miniatur. 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; perangkat yang lebih sekali pakai seperti pemutar musik memiliki input tombol sederhana. Sejak awal pembuatannya, PalmPilots dikirimkan dengan sistem pengenalan tulisan tangan, 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 perangkat kombinasi telepon dan PDA yang lebih konvensional.

Namun, komputer mungil ini tidak memiliki kemewahan keyboard seukuran manusia. Hal 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 \$9.000, perangkat ini sangat melanggar aturan "murah". Pada tahun 1992, AT&T memimpin pengenalan suara untuk pusat panggilannya. Lernout & Hauspie mengakuisisi beberapa perusahaan selama tahun 1990-an dan digunakan di Windows XP. Setelah skandal akuntansi, perusahaan tersebut dibeli

oleh SoftScan yang kemudian menjadi Nuance, yang dilisensikan sebagai Siri versi pertama.

Pada tahun 2003, Microsoft meluncurkan Voice Command untuk PDA Windows Mobile-nya. Pada tahun 2007, Google telah merekrut beberapa teknisi Nuance dan sedang dalam perjalanan dengan teknologi pengenalan suaranya sendiri. Saat ini, teknologi suara semakin ada 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 masukan 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, semuanya hanyalah bagian-bagian 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 penjualan Nokia tetap stabil sebesar 15 juta dari tahun 2007 hingga 2008. iPhone sendiri tidak mengambil alih penjualan iPod hingga tahun 2010, setelah Apple mengizinkan pengguna untuk mengakses iTunes sepenuhnya.

Satu tren yang sangat kuat pada semua perangkat komputer kecil, apa pun mereknya, adalah peralihan ke input sentuh. Ada beberapa alasan untuk ini. Yang pertama adalah visual yang menarik sekaligus bermanfaat, dan semakin banyak yang dapat kita lihat, semakin tinggi kualitas perangkat yang dirasakan. Dengan perangkat yang lebih kecil, ruang menjadi sangat penting, sehingga menghilangkan kontrol fisik dari perangkat berarti persentase perangkat yang lebih besar tersedia untuk dipajang. Alasan kedua dan ketiga bersifat praktis dan berfokus pada manufaktur. Selama teknologinya murah dan andal, lebih sedikit komponen yang bergerak berarti lebih sedikit biaya produksi dan lebih sedikit kerusakan mekanis, keduanya merupakan keuntungan besar bagi perusahaan perangkat keras.

Alasan keempat adalah menggunakan tangan Anda sebagai input dianggap wajar. Meskipun tidak memungkinkan gerakan kecil, 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-and-GUI, panduan antarmuka baru untuk sentuhan memungkinkan pengalaman yang cukup konsisten dan bebas kesalahan bagi pengguna yang hampir mustahil dilakukan menggunakan sentuhan dengan GUI berbasis mouse atau stylus. Alasan terakhir untuk beralih ke input sentuh hanyalah masalah selera: tren desain saat ini bergeser ke arah minimalis di era ketika teknologi komputer bisa sangat membebani. Dengan demikian, perangkat yang disederhanakan dapat dianggap lebih mudah digunakan, bahkan jika kurva pembelajarannya jauh lebih sulit dan fitur-fiturnya dihilangkan.

Satu titik koneksi yang menarik antara tangan dan mouse adalah trackpad, yang dalam beberapa tahun terakhir memiliki kemampuan untuk meniru gerakan multisentuh touchpad sambil menghindari masalah oklusi dari interaksi tangan-ke-layar. Karena tablet memungkinkan input relatif yang dapat berupa rasio ukuran layar keseluruhan, ia memungkinkan gerakan yang lebih kecil, mirip dengan mouse atau stylus. Ia masih

mempertahankan beberapa masalah yang sama yang mengganggu input tangan kelelahan dan kurangnya dukungan fisik yang memungkinkan tangan manusia melakukan pekerjaannya yang paling rumit dengan alat tetapi ia dapat digunakan untuk hampir semua interaksi tingkat OS konvensional.

Mengapa Kita Baru Saja Membahas Semua Ini?

Jadi, apa gunanya pelajaran sejarah singkat kita? Untuk menyiapkan panggung yang tepat untuk masa depan, di mana kita akan beralih dari ranah komputasi masa kini yang sudah diketahui, menuju masa depan input spasial yang belum diketahui. Pada titik waktu tertentu, mudah untuk berasumsi bahwa kita mengetahui segala hal yang mengarah ke masa kini atau bahwa kita selalu berada di jalur yang benar. Meninjau kembali di mana kita telah berada dan bagaimana masa kini terjadi adalah cara yang sangat baik untuk membuat keputusan yang lebih baik untuk masa depan. Mari kita lanjutkan dengan menjelajahi interaksi manusia-komputer (HCI) untuk komputasi spasial. Kita dapat memulai dengan dasar-dasar yang tidak akan berubah dalam jangka pendek: bagaimana manusia dapat menerima, memproses, dan mengeluarkan informasi.

Jenis-Jenis Modalitas Hci Umum

Ada tiga cara utama kita berinteraksi dengan komputer:

- ✓ Visual
- ✓ Pose, grafik, teks, UI, layar, animasi
- ✓ Auditori
- ✓ Musik, nada, efek suara, suara
- ✓ Fisik

1.3 PERANGKAT KERAS, TOMBOL, HAPTİK, OBJEK NYATA

Perhatikan bahwa di latar belakang yang telah kita bahas sejauh ini, masukan fisik dan keluaran audio/visual mendominasi HCI, terlepas dari jenis komputer. Haruskah ini berubah untuk komputasi spasial, di dunia tempat objek digital Anda mengelilingi Anda dan berinteraksi dengan dunia nyata? Mungkin. Mari kita mulai dengan menyelami pro dan kontra dari setiap modalitas.

Modalitas visual

Kelebihan:

- 250 hingga 300 kata per menit (WPM) dipahami oleh manusia
- Sangat dapat disesuaikan
- Langsung dikenali dan dipahami oleh manusia
- Fidelitas yang sangat tinggi dibandingkan dengan suara atau haptik
- Tidak bergantung waktu; dapat menggantung di ruang selamanya
- Mudah diatur ulang atau dipetakan ulang tanpa menghilangkan pemahaman pengguna
- Modalitas ambient yang baik; seperti iklan atau rambu, dapat diperhatikan oleh manusia di waktu luang mereka

Kekurangan:

- Mudah terlewatkan; bergantung pada lokasi
- Sebagai input, biasanya memerlukan padanan fisik yang kuat; gestur dan pose sangat melelahkan
- Memerlukan 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 "mengganggu" jika pengguna sedang mengikuti alur
- Pelacakan visual (mata) yang sangat tepat membutuhkan prosesor yang intensif

Penggunaan terbaik dalam interaksi khusus HMD:

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

Contoh kasus penggunaan ponsel pintar:

- Dirancang untuk hanya visual
- Berfungsi bahkan jika suara dimatikan
- Berfungsi dengan umpan balik fisik
- Kemampuan fisik minimal
- Banyak bahasa animasi baru untuk menunjukkan umpan balik

Modalitas fisik**Kelebihan:**

- Braille: 125 WPM
- Dapat dilakukan dengan sangat cepat dan tepat
- Melewati proses berpikir tingkat tinggi, sehingga mudah untuk beralih ke "aliran" fisiologis dan mental
- Pelatihan memberi makan korteks motorik primer; pada akhirnya tidak memerlukan korteks premotorik atau pemrosesan ganglia basal yang lebih intensif
- Memiliki komponen otak hewan yang kuat "ini nyata"; isyarat realitas yang kuat
- Umpan balik ringan diakui secara tidak sadar
- Penundaan paling sedikit antara affordance dan input
- Jenis input modalitas tunggal terbaik, karena paling tepat

Kekurangan:

- Bisa melelahkan
- Perangkat keras fisik lebih sulit dibuat, bisa mahal, dan rusak
- Beban kognitif jauh lebih tinggi selama fase pengajaran
- Kurang fleksibel daripada visual: tombol tidak dapat benar-benar dipindahkan
- Mode memerlukan 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 penting Contoh kasus penggunaan—instrumen musik:
- Affordance fisik yang komprehensif
- Tidak diperlukan visual setelah tingkat penguasaan tertentu; kreator sedang mengalir
- Hampir selalu memiliki komponen umpan balik audio
- Memungkinkan gerakan melewati bagian otak—pikiran menjadi Tindakan

Modalitas audio

Kelebihan:

- 150 hingga 160 WPM yang dipahami manusia
- Omnidirectional
- Mudah diegetik untuk memberikan umpan balik dan meningkatkan nuansa dunia
- Dapat sangat halus dan tetap berfungsi dengan baik
- Seperti masukan 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 affordance dan umpan balik konfirmasi

Kekurangan:

- Mudah bagi pengguna untuk memilih keluar dengan perangkat saat ini
- Tidak ada kemampuan untuk mengontrol ketepatan keluaran
- Berbasis waktu: jika pengguna melewatkannya, harus mengulang
- Dapat secara fisik tidak menyenangkan (refleks batang otak)
- Lebih lambat secara keseluruhan
- Masukan yang tidak jelas dan tidak tepat karena keterbatasan bahasa
- Bergantung pada waktu dan implementasi
- Tidak dapat disesuaikan
- Berpotensi intensif prosesor

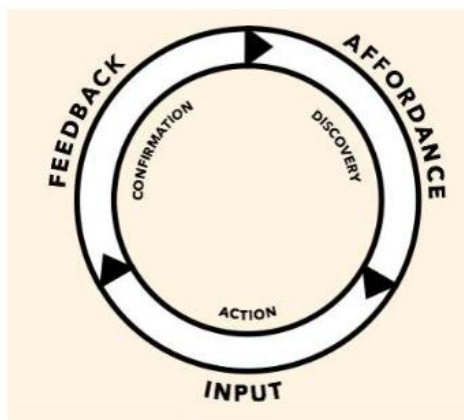
Penggunaan terbaik dalam HMD khusus Interaksi:

- Baik untuk reaksi mendalam
- Cara hebat untuk membuat pengguna melihat hal tertentu
- Baik untuk kamera yang dikendalikan pengguna
- Baik saat pengguna dibatasi secara visual dan fisik
- Baik untuk peralihan mode

Contoh kasus penggunaan ruang operasi:

Dokter bedah terkungkung secara visual dan fisik; audio sering kali menjadi satu-satunya pilihan. Pembaruan suara terus-menerus untuk semua informasi. Perintah suara untuk

peralatan, permintaan, dan konfirmasi. Suara dapat memberikan informasi paling padat tentang keadaan terkini dan kondisi mental; sangat berguna dalam situasi berisiko tinggi



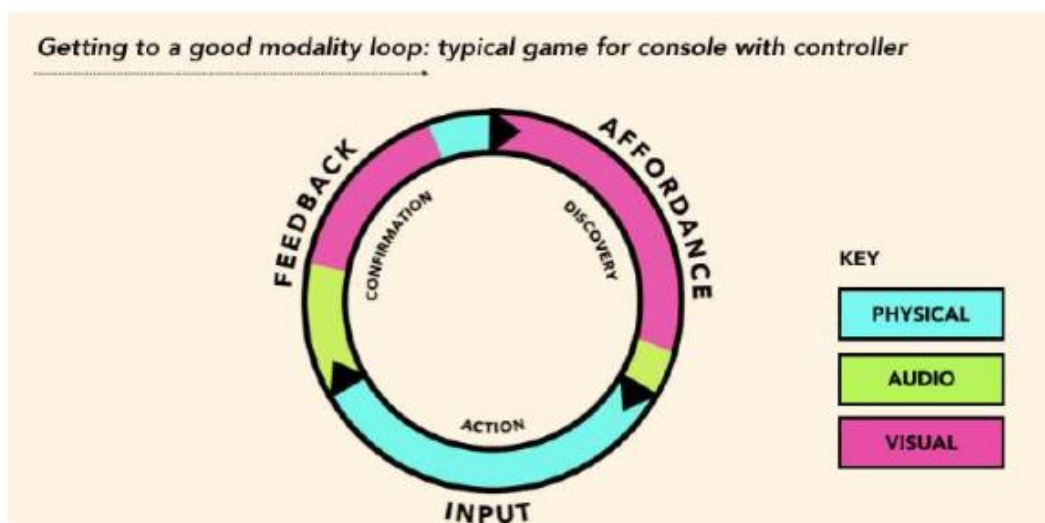
Gambar 1.5. Siklus dari loop modalitas HCI yang umum

Sekarang setelah kita menuliskan pro dan kontra dari setiap jenis modalitas, kita dapat mempelajari proses HCI dan memetakan siklusnya dengan tepat. Gambar 1-5 mengilustrasikan aliran umum, diikuti dengan deskripsi tentang cara memetakannya ke skenario permainan.

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

- Yang pertama umumnya adalah fase affordance 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 mengonfirmasi input dengan bereaksi dengan cara tertentu.

Gambar 1.6 menyajikan grafik yang sama, sekarang diisi untuk loop UX tutorial gim video konsol konvensional.



Gambar 1.6 Siklus loop modalitas HCI yang umum, dengan contoh

Mari kita bahas ini. Dalam banyak tutorial gim video, sarana pertama yang dapat digunakan pengguna untuk melakukan sesuatu umumnya adalah hamparan UI yang tidak dapat dilewatkan yang memberi tahu pengguna label tombol yang perlu ditekan. Hal ini terkadang terwujud dengan gambar atau model tombol yang sesuai. Mungkin ada suara terkait seperti perubahan musik, nada, atau dialog, tetapi selama tutorial sebagian besar mendukung dan tidak mengajar.

Untuk gim video konsol konvensional, tahap input akan sepenuhnya bersifat fisik; misalnya, penekanan tombol. Ada gim video eksplorasi yang mungkin memanfaatkan input audio seperti ucapan, atau kombinasi input fisik dan visual (misalnya, pose tangan), tetapi itu jarang terjadi. Dalam hampir semua kasus, pengguna cukup menekan tombol untuk melanjutkan. Tahap umpan balik sering kali merupakan kombinasi dari ketiga modalitas: pengendali mungkin memiliki umpan balik haptik, visual hampir pasti akan berubah, dan akan ada suara konfirmasi.

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

Modalitas Baru

Dengan munculnya perangkat keras yang lebih baik dan sensor baru, kita memiliki cara baru untuk berbicara dengan komputer dan membuat mereka memantau dan bereaksi terhadap kita. Berikut adalah daftar singkat masukan yang masih dalam tahap prototipe atau komersialisasi:

- ✓ Lokasi
- ✓ Laju napas
- ✓ Nada, tinggi nada, dan frekuensi suara
- ✓ Gerakan mata
- ✓ Pelebaran pupil
- ✓ Denyut jantung
- ✓ Melacak gerakan anggota tubuh yang tidak disadari

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

sebagai masukan pasif.

Satu sifat lain dari masukan khusus ini adalah bahwa masukan tersebut satu arah; komputer dapat bereaksi terhadap perubahan pada masing-masing, tetapi tidak dapat merespons dengan cara yang sama, setidaknya tidak sampai komputer berubah secara signifikan. Bahkan saat itu, sebagian besar daftar akan mengarah pada putaran umpan balik ambien, bukan umpan balik langsung atau instan.

Keadaan Modalitas Saat Ini untuk Perangkat Komputasi Spasial

Pada saat tulisan ini dibuat, perangkat AR dan VR memiliki metode modalitas berikut di sebagian besar perangkat keras yang ditawarkan:

Fisik

- ✓ Untuk input pengguna: pengontrol
- ✓ Untuk output komputer: haptik

Audio

- ✓ Untuk input pengguna: pengenalan ucapan (jarang)
- ✓ Untuk keluaran komputer: suara dan audio Visual spasial
- ✓ Untuk masukan pengguna: pelacakan tangan, pengenalan pose tangan, dan pelacakan mata
- ✓ Untuk keluaran komputer: HMD

Satu keanehan muncul dari daftar ini: komputasi imersif, untuk pertama kalinya, telah menyebabkan munculnya masukan visual melalui pelacakan visi komputer pada bagian tubuh seperti tangan dan mata. Meskipun posisi dan gerakan tangan sering kali penting, sejauh hal itu terkait dengan penekanan tombol fisik, hal itu belum pernah menjadi penting sebelumnya. Kita akan membahas lebih lanjut tentang hal ini nanti, tetapi mari kita mulai dengan jenis masukan yang paling konvensional: pengontrol dan layar sentuh.

Pengontrol Saat Ini untuk Sistem Komputasi Imersif

Jenis pengontrol yang paling umum untuk headset realitas campuran, augmented, dan virtual (XR), berakar pada pengontrol permainan konvensional. Sangat mudah untuk melacak setiap pengontrol paket XR HMD komersial kembali ke desain joystick dan D-pad. Pekerjaan awal seputar sarung tangan yang dilacak gerakannya, seperti VIEWlab milik NASA Ames dari tahun 1989, belum dikomoditiasi dalam skala besar. Menariknya, Ivan Sutherland telah mengajukan bahwa pengendali VR seharusnya berupa joystick pada tahun 1964; hampir semua memilikinya, atau padanan thumbpad, pada tahun 2018.



Gambar 1.7 Sistem masukan Sixsense Stem

Sebelum headset konsumen pertama, Sixsense merupakan pelopor dalam bidang ini dengan pengendali magnetik yang dilacak, yang mencakup tombol pada kedua pengendali yang familiar bagi konsol gim apa pun: A dan B, home, serta tombol, joystick, bumper, dan pemicu yang lebih generik.

Sistem yang sepenuhnya dilacak dan terikat PC saat ini memiliki masukan yang serupa. Kontroler Oculus Rift, kontroler Vive, dan kontroler Windows MR semuanya memiliki kesamaan berikut:

- ❖ Tombol pilih utama (hampir selalu berupa pemicu)
 - ❖ Varian pilih sekunder (pemicu, pegangan, atau bumper)
 - ❖ Tombol setara A/B
 - ❖ Input 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, dan membiarkan aplikasi yang aktif kembali ke layar beranda.

Headset mandiri memiliki beberapa subset dari daftar sebelumnya di pengontrolnya. Dari remote HoloLens yang tidak terlacak hingga pengontrol 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 bergantung pada kemampuan sistem pelacakan HMD dan bagaimana OS telah dirancang.

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

Seiring dengan peningkatan dan standarisasi sistem pelacakan, kita dapat memperkirakan standar ini akan menguat seiring berjalannya waktu. Keduanya berguna pada waktu yang berbeda, dan seperti masalah inversi sumbu Y klasik, mungkin saja pengguna yang

berbeda memiliki preferensi yang sangat berbeda sehingga kita harus selalu memperhitungkan keduanya. Terkadang, Anda ingin mengarahkan sesuatu untuk memilihnya; terkadang Anda ingin menggulir untuk memilihnya. Mengapa tidak keduanya?

1.4 TEKNOLOGI PELACAKAN TUBUH

Mari kita bahas tiga jenis pelacakan tubuh yang paling umum dibahas saat ini: pelacakan tangan, pengenalan pose tangan, dan pelacakan mata.

Pelacakan tangan

Pelacakan tangan terjadi saat seluruh gerakan tangan dipetakan ke kerangka digital, dan inferensi masukan dibuat berdasarkan gerakan atau pose tangan. Hal ini memungkinkan gerakan alami seperti mengambil dan menjatuhkan objek digital dan pengenalan gerakan. Pelacakan tangan dapat sepenuhnya berbasis visi komputer, menyertakan sensor yang terpasang pada sarung tangan, atau menggunakan jenis sistem pelacakan lainnya.

Pengenalan pose tangan

Konsep ini sering kali disalahartikan dengan pelacakan tangan, tetapi pengenalan pose tangan merupakan bidang penelitiannya sendiri. Komputer telah dilatih untuk mengenali pose tangan tertentu, seperti bahasa isyarat. Maksud dipetakan saat setiap pose tangan dikaitkan dengan peristiwa tertentu seperti meraih, melepaskan, memilih, dan tindakan umum lainnya.

Di sisi positifnya, pengenalan pose dapat lebih sedikit menggunakan prosesor dan memerlukan kalibrasi individual yang lebih sedikit daripada pelacakan tangan yang kuat. Namun secara eksternal, hal ini dapat melelahkan dan membingungkan bagi pengguna yang mungkin tidak memahami bahwa pembuatan ulang pose lebih penting daripada gerakan tangan alami. Diperlukan juga sejumlah besar tutorial pengguna untuk mengajarkan pose tangan.

Pelacakan mata

Mata terus bergerak, tetapi melacak posisi mata memudahkan untuk menyimpulkan minat dan maksud terkadang bahkan lebih cepat daripada yang disadari pengguna, mengingat gerakan mata diperbarui sebelum visualisasi otak diperbarui. Meskipun cepat melelahkan sebagai input tersendiri, pelacakan mata merupakan input yang sangat baik untuk dipadukan dengan jenis pelacakan lainnya. Misalnya, pelacakan mata dapat digunakan untuk melakukan triangulasi posisi objek yang diminati pengguna, dikombinasikan dengan pelacakan tangan atau pengontrol, bahkan sebelum pengguna benar-benar menunjukkan minat.

Penulis belum memasukkan pelacakan tubuh atau pengenalan ucapan ke dalam daftar, terutama karena tidak ada teknologi di pasaran saat ini yang bahkan mulai menerapkan salah satu teknologi tersebut sebagai teknik input standar. Namun, perusahaan seperti Leap Motion, Magic Leap, dan Microsoft membuka jalan bagi semua jenis pelacakan baru yang tercantum di sini.

Catatan tentang Pelacakan Tangan dan Pengenalan Pose Tangan

Pelacakan tangan dan pengenalan pose tangan keduanya pasti menghasilkan perubahan yang menarik, dan agak berlawanan dengan intuisi, pada cara manusia sering berpikir tentang berinteraksi dengan komputer. Di luar gerakan percakapan, di mana gerakan

tangan sebagian besar memainkan peran pendukung, manusia umumnya tidak menganggap lokasi dan pose tangan mereka penting. Kita menggunakan tangan setiap hari sebagai alat dan dapat mengenali gerakan yang ditiru untuk tindakan yang terkait dengannya, seperti mengambil benda. Namun dalam sejarah HCI, lokasi tangan sangat tidak berarti. Faktanya, periferal seperti tetikus dan pengontrol permainan secara khusus dirancang untuk tidak bergantung pada lokasi tangan: Anda dapat menggunakan tetikus di sisi kiri atau kanan, Anda dapat memegang pengontrol satu kaki di atas atau bawah di depan Anda; tidak ada bedanya dengan apa yang Anda masukkan.

Pengecualian mencolok dari aturan ini adalah perangkat sentuh, yang lokasi tangan dan masukannya tentu saja terkait erat. Bahkan saat itu, "gerakan" sentuh tidak banyak berhubungan dengan gerakan tangan di luar ujung jari yang menyentuh perangkat; Anda dapat melakukan sapuan tiga jari dengan tiga jari mana pun yang Anda pilih. Satu-satunya hal yang benar-benar penting adalah Anda memenuhi persyaratan minimum untuk melakukan apa yang diinginkan komputer guna memperoleh hasil yang Anda inginkan. Penglihatan komputer yang dapat melacak tangan, mata, dan tubuh berpotensi sangat kuat, tetapi dapat disalahgunakan.

1.5 INPUT SUARA, TANGAN, DAN PERANGKAT KERAS PADA GENERASI BERIKUTNYA

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

Suara

Dalam bahasa sehari-hari, perintah suara tidaklah tepat, tidak peduli seberapa sempurna dipahami. Orang sering salah memahami bahkan kalimat dalam bahasa yang 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 bahwa perintah tersebut dipahami. Jargon merupakan evolusi linguistik yang menarik dari hal ini: kata-kata yang sangat terspesialisasi yang memiliki arti tertentu dalam konteks tertentu bagi suatu kelompok merupakan bentuk tombol pintas bahasa, jika Anda mau menyebutnya demikian.

Komputer dapat bereaksi jauh lebih cepat daripada manusia—itulah keuntungan terbesarnya. Untuk mengurangi input menjadi sekadar vokalisasi manusia berarti kita memperlambat secara signifikan cara kita berkomunikasi dengan komputer mulai saat ini. Mengetik, mengetuk, dan menekan tombol yang dipetakan tindakan semuanya sangat cepat dan tepat. Misalnya, jauh lebih cepat untuk memilih sepotong teks, menekan tombol pintas

untuk "memotong", menggerakkan kursor, lalu menekan tombol pintas untuk "menempel" daripada menjelaskan tindakan tersebut kepada komputer. Hal ini berlaku untuk hampir semua tindakan.

Namun, untuk menjelaskan skenario, menceritakan kisah, atau membuat rencana dengan manusia lain, sering kali lebih cepat untuk sekadar menggunakan kata-kata dalam percakapan karena setiap kesalahpahaman yang mungkin terjadi dapat segera dipertanyakan dan diperbaiki oleh pendengar. Hal ini memerlukan tingkat pengetahuan praktis tentang dunia yang kemungkinan tidak akan dimiliki komputer hingga munculnya kecerdasan buatan yang sesungguhnya. Ada keuntungan lain dari masukan suara: saat Anda memerlukan masukan tanpa menggunakan tangan, saat Anda sedang sibuk, saat Anda memerlukan dikte transliterasi, atau saat Anda menginginkan perubahan modalitas yang cepat (misalnya, "minimalkan! keluar!") tanpa gerakan lain. Masukan suara akan selalu berfungsi paling baik saat digunakan bersamaan dengan modalitas lain, tetapi itu bukan alasan untuk tidak menyempurnakannya. Dan, tentu saja, teknologi pengenalan suara dan transkripsi ucapan ke teks memiliki kegunaan lebih dari sekadar masukan.

Tangan

Modalitas visual seperti pelacakan tangan, gerakan, dan pengenalan pose tangan secara konsisten berguna sebagai konfirmasi sekunder, sama seperti kegunaannya sebagai pose tangan dan postur dalam percakapan manusia biasa. Modalitas tersebut akan sangat berguna untuk komputasi spasial saat kita memiliki cara mudah untuk melatih kumpulan data yang dipersonalisasi untuk pengguna individu dengan sangat cepat. Ini akan memerlukan beberapa hal:

- Individu untuk memelihara kumpulan data biometrik yang dipersonalisasi di seluruh platform
- Cara bagi individu untuk mengajarkan komputer apa yang mereka inginkan agar diperhatikan atau diabaikan oleh computer

Alasan untuk persyaratan ini sederhana: manusia sangat bervariasi baik dalam seberapa banyak mereka bergerak dan memberi isyarat serta apa arti isyarat tersebut bagi mereka. Seseorang mungkin menggerakkan tangannya terus-menerus, tanpa berpikir. Yang lain mungkin hanya memberi isyarat sesekali, tetapi isyarat itu sangat penting. Kita 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. 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.



Gambar 1.8 Triangulasi untuk menopang berat tangan penting bahkan jika Anda memiliki pisau atau benda tajam digital, Anda perlu memiliki cara untuk menopang tangan Anda untuk gerakan yang lebih kecil

Untuk Leap Motion, pose "ambil" 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 dipetakan ke tindakan yang berbeda dalam pengalaman yang berbeda. Hal yang sama berlaku untuk pengendali Oculus Rift, yang mendukung dua pose tangan (menunjuk, dan mengacungkan jempol), yang keduanya dapat dipetakan ulang ke tindakan pilihan pengembang.

Hal ini mengharuskan pengguna untuk menghafal pose dan gerakan yang diperlukan oleh perangkat keras, bukan gerakan tangan alami, seperti bagaimana perangkat tablet menstandarisasi gerakan menggeser untuk memindahkan dan mencubit untuk memperbesar. Meskipun jenis bahasa isyarat manusia-komputer ini memiliki potensi untuk distandardisasi dan menjadi norma, para pendukung harus menyadari bahwa apa yang mereka usulkan adalah alternatif untuk cara manusia menggunakan tangan mereka saat ini, bukan pemetaan ulang. Hal ini diperburuk oleh fakta bahwa tangan manusia sendiri tidak memiliki ketepatan; mereka memerlukan dukungan fisik dan alat untuk memungkinkan ketepatan yang nyata, seperti yang ditunjukkan pada Gambar 1.8.

Pengendali dan periferal fisik lainnya

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

Namun, kita sekarang memasuki masa yang menarik untuk elektronik. Untuk pertama kalinya, hampir semua orang dapat membeli atau membuat sendiri periferal yang dapat bekerja dengan berbagai jenis aplikasi. Orang-orang membuat semuanya dari suku cadang pihak ketiga: mulai dari keyboard dan mouse, hingga pelacak Vive yang dibuat seperti Frankenstein di atas tongkat bisbol atau hewan peliharaan, dan pekerjaan pengecatan khusus

untuk pengendali Xbox mereka. Asumsi yang besar adalah bahwa karena komputasi spasial akan memungkinkan kustomisasi pengguna yang lebih banyak, konsumen secara alami akan mulai membuat masukan mereka sendiri.

Namun, mudah untuk berasumsi bahwa produsen akan membuat perangkat keras yang lebih disesuaikan untuk memenuhi permintaan. Pertimbangkan mobil saat ini: Lexus 4 memiliki lebih dari 450 opsi roda kemudi saja; jika Anda memasukkan semua opsi, ini menghasilkan empat juta kombinasi kendaraan yang sama. Ketika komputasi bersifat pribadi dan berada di rumah Anda bersama Anda, orang akan memiliki pendapat yang kuat tentang bagaimana tampilannya, rasanya, dan reaksinya, seperti halnya yang mereka lakukan dengan kendaraan, furnitur, dan wallpaper mereka.

Pembicaraan tentang kustomisasi yang intens ini, baik di sisi platform maupun di sisi pengguna, membawa kita ke alur 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, masukan harus bervariasi secara merata. Cara yang sama seseorang mungkin memilih satu pena dibandingkan pena lain untuk menulis akan berlaku untuk semua aspek interaksi komputer.

BAB 2

MERANCANG TEKNOLOGI DENGAN MEMAHAMI INDRA MANUSIA

Bayangkan masa depan di mana hubungan kita dengan teknologi sekaya kenyataan. Kita tidak sering membicarakan berapa banyak waktu yang dihabiskan di depan layar, dan untungnya, sebagian besar perusahaan teknologi merasakan hal yang sama. Mereka telah berinvestasi besar dalam sensor dan AI untuk menciptakan mesin penginderaan. Dengan memanfaatkan data ucapan, spasial, dan biometrik, yang dimasukkan ke dalam kecerdasan buatan, mereka mengembangkan teknologi ke dalam bentuk yang lebih berhubungan dengan manusia. Namun, tidak banyak yang dipahami tentang cara merancang teknologi yang digerakkan oleh mesin penginderaan yang menarik, mudah diakses, dan bertanggung jawab. Karena itu, industri teknologi perlu berinvestasi lebih banyak dalam memahami desain yang responsif terhadap manusia beserta praktik, kebijakan, dan alat rekayasa 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 membahas topik ini dan semoga menginspirasi eksplorasi lebih lanjut. Sebagai Kepala Desain Baru di Adobe, Penulis bekerja dengan beberapa tim di seluruh perusahaan untuk menghadirkan teknologi baru ke dalam produk dan layanan guna memecahkan tantangan nyata manusia dan masyarakat. Selama lebih dari 25 tahun mendorong desain maju melalui tiga pergeseran teknologi utama, Penulis telah melihat internet memberdayakan ekonomi pengetahuan kita, dan komputasi seluler mengubah cara kita berkomunikasi. Di masa depan, komputasi spasial, yang didukung oleh AI, akan secara dramatis memperluas cara kita untuk berkolaborasi satu sama lain dan menggunakan informasi. Ini akan sangat mengubah cara kita bekerja, hidup, belajar, dan bermain. Penulis yakin dampaknya terhadap masyarakat akan lebih besar daripada gabungan internet dan komputasi seluler. Sebagai seorang desainer, Penulis sangat bersemangat dan terkadang sedikit takut untuk mengambil bagian dalam periode sejarah manusia yang luar biasa ini.

2.1 MEMBAYANGKAN MASA DEPAN

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

Tutup mata Anda sebentar saja. Bayangkan tempat bahagia Anda, kita semua punya satu. Meskipun itu hanya khayalan. Bagi saya, tempat ini berada di pantai di Australia dengan teman-teman di sekitar saya, matahari bersinar, air asin terasa di jari kaki saya, dan suara panggangan yang mendesis. Ini adalah tempat yang membuat saya merasa bahagia karena tempat ini alami, sederhana, dan saya terhubung dengan teman-teman. Ketika saya duduk di depan komputer atau menghabiskan terlalu banyak waktu menatap layar ponsel, saya tidak merasa bahagia. Saya tidak merasa terhubung. Namun setelah seharian berada di tempat

bahagia, saya mulai merindukan informasi dan koneksi lain yang saya dapatkan melalui ponsel. Namun, saya tidak merindukan ponsel saya. Ponsel saya tidak membuat saya bahagia. Jadi, sebagai desainer, saya tertarik pada cara kita mengakses informasi dengan cara yang terasa alami, sederhana, dan dapat membuat kita bahagia.

Pola pikir ini membantu kita sebagai desainer memahami nilai dan pentingnya pekerjaan kita setiap kali kita memulai produk atau desain baru. Penjelasan 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 dalam mesin yang digerakkan oleh pembelajaran mesin. Berikut adalah ringkasan singkat tentang komputasi spasial dan mesin sensoriknya. Komputasi spasial tidak terkunci dalam persegi panjang tetapi dapat mengalir 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 menjadi perantara pengalaman kita. Dengan komputasi spasial, kita dapat menggunakan suara, penglihatan, sentuhan (dalam 3D), gerakan, dan masukan alami lainnya untuk terhubung langsung dengan informasi. Kita tidak perlu lagi berpikir dan berperilaku seperti komputer agar dapat memahami kita, karena komputer akan berhubungan dengan kita secara manusiawi.

Dengan asumsi bahwa komputer memahami kita, maka komputasi spasial juga dapat memahami perbedaan kita dan mendukung kemampuan dan perbedaan manusiawi kita. Misalnya, kita dapat memberikan informasi verbal tentang dunia di sekitar seseorang dengan gangguan penglihatan atau menerjemahkan nuansa budaya, bukan hanya bahasa, saat berkomunikasi lintas budaya. Sebaliknya, komputasi spasial dapat meningkatkan kemampuan kita yang sudah 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 kita yang didukung oleh teknologi AI. Penglihatan komputer, pendengaran mesin, dan sentuhan mesin dapat menghasilkan data seperti lokasi pasti kamera Anda; dimensi ruang di sekitar Anda; mengidentifikasi objek, orang, dan ucapan; biodata, dan banyak lagi. Dengan menggunakan teknologi AI, kita dapat menginterpretasikan data ini dengan cara yang meniru persepsi manusia. Saat kita memahami dunia, mesin juga dapat memahami dunia.

Karena indra mesin semakin banyak 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 yang mendukung augmented reality (AR) akan mencapai 2,7 miliar ponsel pada akhir tahun 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 daripada saat

diluncurkan pada tahun 2017.

- Melalui ponsel Anda yang mendukung AR, teknologi AI dapat mendeteksi emosi dasar manusia seperti marah, jijik, takut, senang, netral, sedih, dan terkejut dari ekspresi wajah Anda. Emosi-emosi ini dipahami sebagai sesuatu yang lintas budaya dan umum digunakan, tetapi tidak selalu menjadi ukuran sebenarnya tentang bagaimana perasaan seseorang sebenarnya. Mark Billinghurst, pelopor AR, dan Direktur Laboratorium Komputer Empatik di Universitas Australia Selatan berkata, “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 estimasi emosi yang lebih baik, penting untuk memperhitungkan 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 memperhitungkan semua isyarat ini untuk memahami perasaan mereka, atau perasaan orang lain. Mesin harus melakukan hal yang sama.”
- AR mempercepat pelatihan dengan memanfaatkan rasa proprioepsi manusiawi kita, pemahaman tentang ruang di sekitar kita, untuk pelatihan dan keselamatan.
- Mikrofon dan pengeras suara kita telah menjadi asisten virtual kita dan semakin memasuki rumah, telepon, perangkat yang dapat didengar, dan perangkat lainnya.
- Pakaian dan jam tangan yang tertanam dengan sensor berpotensi untuk mengukur intensitas emosional kita dengan keringat (respons kulit galvanik) dan memantau kesehatan kita 24/7 melalui detak jantung kita.
- Kota-kota kita menjadi “pintar” dengan sejumlah besar sensor yang ditempatkan di jalan-jalan, mobil, dan sistem transportasi umum kita. Dengan mengintegrasikan data mereka, pemerintah kota memperoleh wawasan yang lebih rinci tentang cara memecahkan masalah yang saling terkait. Sensor-sensor ini memantau berbagai hal seperti cuaca, kualitas udara, lalu lintas, radiasi, dan ketinggian air, dan dapat digunakan untuk secara otomatis menginformasikan layanan dasar seperti lampu lalu lintas dan lampu jalan, sistem keamanan, dan peringatan darurat.

Komputasi spasial muncul dari interaksi kemajuan teknologi dalam sensor mesin, daya rendering, AI dan pembelajaran mesin, tangkapan 3D, dan tampilan. Antarmuka pengguna suara (VUI), gestur, dan tampilan XR menyediakan 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 sarana transportasi favorit kita. Tanyakan saja pada GPS mobil bagaimana cara mencapai tujuan perjalanan darat Anda.

Sementara VUI telah mencapai rumah, ponsel, dan mobil kita, layanan AR belum mencapai adopsi konsumen secara massal. Beberapa orang percaya ini akan terjadi ketika kacamata AR kelas konsumen hadir. Saya percaya titik kritis akan tiba hanya ketika perangkat, data sensorik, dan sistem AI bersama-sama membuka kekuatan super kreatif alami manusia kita melalui kolaborasi spasial. Saya akan menjelaskannya lebih lanjut nanti di bab ini.

Mesin dengan 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 saudaranya yang jauh lebih cerdas, pembelajaran mendalam, menyediakan cara bagi mesin untuk menginterpretasikan sejumlah besar data dengan cara yang baru dan menakjubkan. Mesin cerdas kita saat ini dapat belajar, tetapi tidak sepenuhnya mengerti.

Untuk komputasi spasial, pembelajaran mesin bertindak seperti sistem saraf manusia untuk indera kita. Karena sistem kota dan sistem bangunan kita mengintegrasikan sensor yang jumlahnya terus bertambah, sistem tersebut juga mencerminkan sistem saraf. Data dari sensor seperti kamera (penglihatan), mikrofon (pendengaran), dan unit pengukuran inersia (IMU) dikumpulkan dan diinterpretasikan 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 jalan, kendaraan Anda dapat segera memberi tahu departemen pekerjaan umum setempat tentang perbaikan lubang tersebut; mainan dapat mengetahui apakah mainan itu sedang digunakan atau ditinggalkan di dalam kotak mainan, yang menghasilkan mainan yang lebih baik dan mengurangi tempat pembuangan sampah.

Pembelajaran mesin dan data historis mengingat dan memahami masa lalu. Kita sudah melihat kalimat-kalimat kita diselesaikan di Gmail berdasarkan gaya penulisan historis kita. Suatu hari nanti, anak-anak saya mungkin akan mengalami kehidupan saya saat mereka seusia saya; mungkin kita bisa "melihat" masa depan penemuan kita yang diprediksi berdasarkan peristiwa-peristiwa historis. Seiring dengan kemajuan AI, desain sensorik akan terus menjadi lebih alami, memberikan perangkat kita indra manusia yang alami. Kita membayangkan dunia di mana alat-alat kita lebih alami, dan saya percaya inilah masa depan yang diinginkan orang-orang. Semakin alami dan intuitif alat-alat tersebut, semakin mudah diakses, di situlah desain sensorik memainkan peran penting.

Jadi, Untuk Siapa Kita Membangun Masa Depan Ini?

Kita membangun masa depan untuk orang-orang seperti dua anak laki-laki dalam Gambar 2-1. Mereka akan membangun produk dan layanan berdasarkan ekosistem yang kita bangun saat ini. Mari kita dengarkan mereka dan dapatkan inspirasi dari kebutuhan mereka untuk masa depan yang lebih baik. Berikut ini beberapa hal yang mereka katakan.

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



Gambar 2.1 Dua generasi Z

Identitas mereka tidak ditentukan oleh ras atau gender, tetapi oleh identitas yang bermakna yang berubah seiring dengan perubahan yang mereka alami. Mereka dengan lancar dan terus-menerus mengekspresikan kepribadian mereka. Jadi, ketika ditanya, “Menurutmu, apakah kamu akan menikahi seorang perempuan atau laki-laki?” kedua pemuda di Gambar 2-1 tidak menganggapnya sebagai pertanyaan yang aneh. Salah satu dari mereka menjawab “perempuan” dan yang lainnya menjawab, “Saya sedang mencari tahu.” Jawaban mereka tidak canggung atau tidak nyaman, karena mereka bukanlah pemikir biner.

Saya melihat merek beralih dari menciptakan pengalaman bertipe kreasi sendiri untuk YouTube atau Instagram ke merek yang memungkinkan identitas yang cair dengan menggunakan masker wajah AR di Snapchat dan Facebook Messenger. Inilah jenis pergeseran yang diharapkan dengan komputasi spasial. Kita beralih dari tempat informasi disimpan di layar ke dunia tempat ekspresi kreatif dapat mengalir bebas ke lingkungan sekitar kita dengan AR yang didukung oleh AI. Pemikir masa depan harus mampu menavigasi kekacauan sambil membangun koneksi, itulah sebabnya kreativitas merupakan keterampilan inti yang dibutuhkan untuk generasi mendatang.

Kita semua perlu membuat ekspresi kreatif lebih sederhana, lebih alami, dan tidak terlalu terikat pada perangkat tetapi lebih pada indera manusia. Dalam banyak hal, perangkat spasial akan didemokratisasi. Perangkat seperti animasi waktu nyata merupakan keterampilan inti yang dibutuhkan dalam komputasi spasial, tetapi, saat ini, kesulitan animasi menyebabkannya diserahkan kepada para profesional yang memiliki akses ke perangkat tertentu. Itulah sebabnya tim saya di Adobe membuat perangkat yang memungkinkan Anda merekam gerakan burung yang terbang atau teman yang menari hanya dengan menangkap gerakan melalui kamera ponsel Anda dan langsung mentransfernya ke objek 3D atau desain 2D. Sungguh luar biasa melihat keajaiban di wajah orang-orang saat mereka menggunakan keajaiban teknologi penginderaan (Gambar 2.2).



Gambar 2.2. Satu generasi Z mengenakan Microsoft HoloLens

Anggota GenZ ingin berkreasi secara kolaboratif dalam waktu nyata. Mereka juga berharap dapat berkreasi dengan apa pun, di mana pun, hanya dengan melihatnya atau berinteraksi dengannya (yang kita sebut bermain). Saat ini, banyak anak belajar dengan menjelajahi dunia di sekitar mereka dari ruang kelas menggunakan AR seluler. Atau mereka meminta Google untuk menyelesaikan pekerjaan rumah matematika mereka ya, anak-anak saya melakukannya. Pada saat GenZ memasuki dunia kerja, mereka akan memiliki antarmuka yang mendukung AR yang memproyeksikan informasi pada dan di sekitar objek sehingga mereka dapat menggunakan kedua tangan untuk belajar bermain gitar. Seperti yang dikatakan Tea Uglow, ini akan menjadi seperti "YouTube mekanis yang luar biasa."

Kreativitas 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 yang membuat penerbitan tersedia untuk semua orang. AR, realitas virtual (VR), dan AI akan mengubah cara kita untuk tidak hanya sekadar berbagi apa yang ada di pikiran kita, tetapi juga apa yang ada di hati kita.

Seiring dengan meluasnya AR, AI, dan komputasi spasial ke dunia di sekitar kita, ekspresi kreatif akan menjadi sama pentingnya dengan literasi. Sebagai bagian dari industri teknologi yang lebih luas, Adobe ingin menyediakan perangkat kreatif kami untuk semua orang (XD gratis!), yang mencakup berbagai kemampuan dan budaya, serta menghormati hak privasi dan transparansi orang-orang. Ini adalah waktu yang tepat untuk menciptakan perangkat yang membentuk hubungan kita dengan realitas spasial.

2.2 PERAN DESAINER DAN TIM DI MASA DEPAN

Desain sensorik secara sederhana adalah perekat yang menyatukan disiplin desain spasial (seperti desainer arsitektur, interior, industri, sistem, dan UI) dengan sains (seperti kognitif dan ilmu saraf), seniman, aktivis dan pembuat kebijakan, serta insinyur AI. Mendesain untuk masa depan dengan komputasi spasial bertenaga AI memerlukan berbagai keterampilan

dan pemahaman mendalam tentang perilaku manusia oleh semua orang yang terlibat. Ini adalah area desain yang sedang berkembang dan membutuhkan keragaman peran yang besar untuk diciptakan sehingga akan menghasilkan yang terbaik dalam diri manusia.

Pada bulan Agustus 2018, saya bertemu dengan seorang seniman pertunjukan tuna rungu yang inspiratif, Rosa Lee Timm. Ia meminta Adobe Design untuk:

[m]encakup mereka [orang-orang dengan kemampuan yang 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 akan Anda pikirkan, sehingga menjadi organik. Dan kemudian ketika selesai, itu dirancang dengan mudah dengan distribusi yang mudah sejak awal.

Rosa kemudian bertanya kepada kami apakah kami dapat membangun alat yang menerjemahkan kata-kata yang diucapkan ke dalam bahasa isyarat sehingga kami dapat "membaca" dalam bahasanya sendiri. Ia menunjukkan bahwa banyak video pelatihan bahkan tidak memiliki teks. Hal ini mengilhami saya untuk berpikir tentang bagaimana teknologi pelacakan dan pengenalan wajah dan tangan dapat digunakan untuk menerjemahkan bahasa isyarat ke dalam bahasa Inggris dan bahasa Inggris kembali ke dalam bahasa isyarat. Orang lain yang telah sangat menginspirasi tim kami untuk berpikir lebih global, lintas budaya, dan inklusif adalah Farai Madzima, UX Lead Shopify. Tahun lalu, ia mengunjungi kami di Adobe Design dan berbagi pemikiran berikut:

Jika Anda beranggapan bahwa keberagaman hanya tentang warna kulit, berarti Anda tidak memperhatikan. Jika Anda beranggapan keberagaman hanya tentang gender atau kemampuan, berarti Anda tidak memperhatikan. Anda perlu bekerja dengan orang-orang yang tidak berjalan, berpikir, dan berbicara seperti Anda. Anda perlu menjadikan orang-orang tersebut sebagai bagian dari cara Anda bekerja. Ini terdengar seperti hal yang sulit, selain memecahkan masalah dalam mendesain produk, tetapi ini benar-benar penting. Tantangan yang kita lihat di masyarakat lahir dari fakta bahwa kita belum melihat apa yang dibutuhkan dunia dari industri kita. Kita belum memahami apa yang dibutuhkan rekan kerja kita dari kita dan apa yang kita butuhkan untuk diri kita sendiri, yaitu gagasan untuk bersikap lebih berpikiran terbuka tentang apa yang berbeda di dunia.

Peran Perempuan dalam AI

Visi saya untuk masa depan desain dimulai dengan inklusivitas dan keberagaman. Saat kita menciptakan bahasa desain baru ini, kita membutuhkan tim yang beragam untuk membangun fondasi. Ini termasuk perempuan. Saya percaya bahwa selalu ada banyak cara untuk memecahkan tantangan, dan mencari perspektif yang berbeda akan sangat penting bagi keberhasilan desain sensorik.

Saya percaya bahwa kita membutuhkan perempuan dan laki-laki untuk memimpin masa depan desain digital untuk komputasi spasial dan AI. Dalam 30 tahun terakhir, kita telah melihat laki-laki lebih banyak memimpin desain platform komputer kita, dan, sebagai hasilnya, kita sekarang melihat kurangnya insinyur perempuan di sektor teknologi. AI mempersonalisasi keuangan, hiburan, berita daring, dan sistem rumah kita. Orang-orang yang merancang sistem komputasi spasial saat ini akan memiliki dampak langsung pada dunia di sekitar kita di masa

mendatang. Ini akan membutuhkan berbagai macam pikiran, menyatukan berbagai perspektif 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, kita telah melihat dukungan industri yang substansial dan tujuan-tujuan luhur yang ditetapkan untuk mengubah cara kita mendekati AI. Ada banyak program yang dipimpin oleh perempuan. Perempuan seperti Fei-Fei Li di Stanford; Kate Crawford, direktur AI Now Institute milik Microsoft; Terah Lyons, yang mengepalai Partnership for AI; dan bahkan Michelle Obama yang mendukung Olga Russakovsky, salah seorang pendiri AI4ALL untuk mendidik perempuan dalam bidang AI selama sekolah menengah, hanya untuk menyebutkan beberapa. Saya pribadi bersemangat untuk apa yang akan terjadi dan apa yang akan kita capai ketika kita merangkul keberagaman dalam ide.

2.3 DESAIN SENSORIK

Keberagaman ide di samping pemahaman mendalam tentang keberadaan manusialah yang akan mendorong desain spasial yang paling tahan lama. Secara historis, desain kita telah dibatasi oleh media dan dimensi. Kita dapat melihat 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 desainer, baik fisik maupun di layar, telah menentukan desain yang dihasilkan dan dari waktu ke waktu norma-norma yang diterima.

Di dunia masa depan kita yang dipenuhi komputasi spasial, jumlah keterbatasan 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 perlu memahaminya, dan itulah sebabnya kami mengembangkan Bahasa Desain Sensorik.

Pengantar

Desain sensorik adalah bahasa desain yang diadaptasi, terinspirasi dari manusia, dan digunakan 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 membalik paradigma desain yang ada dan karenanya memerlukan pendekatan baru. Misalnya, desain layar berfokus pada tindakan yang diinginkan pengguna untuk dilakukan, tetapi desain sensorik justru berfokus pada motivasi yang dimiliki pengguna dengan melibatkan kemampuan kognitif indra mereka. Dengan mengingat hal ini, kami di Adobe memutuskan untuk kembali ke dasar dan berfokus pada prinsip dasar universal perilaku manusia. Kami juga perlu memahami perbedaan dan lapisan antara masyarakat, budaya, dan individu yang terorganisasi. Beruntung bagi kami, sudah ada banyak sekali pekerjaan yang dilakukan di bidang ini. Kami hanya perlu menyaring ratusan makalah penelitian untuk menghasilkan titik awal yang penting. Dengan ide ini, saya mengumpulkan sekelompok desainer, ilmuwan kognitif, wirausahawan, 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, serta seorang pembuat kode/desainer pembelajaran mesin Lisa Jamhoury.

Kami mulai dengan pemahaman bahwa manusia memiliki memori spasial yang sangat baik. Kita menggunakan indra proprioepsi untuk memahami dan mengodekan ruang di sekitar kita. Saya yakin Anda bisa saja ditutup matanya di rumah dan tetap berjalan ke dan membuka lemari es. Kita telah melihat realitas virtual menggunakan proprioepsi sebagai alat yang efektif untuk pelatihan spasial, tetapi desain sensorik lebih dari sekadar spasial, ia melibatkan indra kita.

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

Mendekati desain sensorik

Merupakan peluang yang fantastis untuk dapat merancang persepsi manusia, tetapi hal itu disertai dengan tanggung jawab yang besar. Pemikiran untuk mengubah persepsi seseorang tentang realitas melalui desain, dan potensi konsekuensinya, merupakan hal yang menakutkan. Jadi, tim desain sensorik menulis pendekatan terhadap desain sensorik yang membuat kita bertanggung jawab:

- ❖ Berpusat pada manusia dengan membangun bahasa di sekitar interaksi manusia yang intuitif. Kita dapat melakukan ini hanya dengan memahami perilaku manusia yang mendasar, tubuh kita, dan kemampuan kognitif kita.
- ❖ Bersikap kolaboratif dengan berbagi wawasan kita, mendengarkan masukan, dan belajar dari berbagai orang, dari pakar industri hingga pengguna akhir kita.
- ❖ Menjadi pemimpin desain melalui pekerjaan kita, berbagi wawasan kita secara terbuka dan kolektif.
- ❖ Menetapkan prinsip, metodologi, dan pola yang dapat kita gunakan untuk bekerja sama secara lebih efektif dan meningkatkan produk yang kita buat.
- ❖ Menghormati orang dengan menghormati privasi fisik dan digital mereka; memberi mereka kendali, atau wewenang, atas perangkat yang kita buat; dan mengutamakan kesejahteraan mereka daripada sekadar tepukan di punggung.
- ❖ Melakukan perilaku manusia yang baik dengan membangun sistem untuk menghasilkan 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 dalam hal ini dan hari-hari desain sensorik baru saja dimulai. Kerangka kerja sensorik. Selanjutnya, kami menyusun kerangka kerja, yang dapat Anda lihat pada Gambar 2-3, untuk melihat peluang dan koneksi.

FRAMEWORK		
Penglihatan	Pendengaran	Sentuhan
Vestibular (Gerakan)	Propriosepsi (Kesadaran Spasial)	Agency (Kontrol & Pilihan)

Gambar 2.3. Rincian indra manusia yang umum digunakan

Kami memisahkan indra manusia dan mesin sehingga kami dapat menyatukannya kembali dengan cara baru untuk memecahkan masalah di dunia nyata. Apa saja masalah yang dapat dipecahkan oleh desain sensorik yang tidak dapat diselesaikan oleh media lain? Salah satu contohnya adalah menggunakan visi komputer dan AR untuk memahami bahasa isyarat, menerjemahkannya ke teks, lalu kembali lagi ke bahasa isyarat. Visi komputer dapat memahami ekspresi wajah, dan ketika dikombinasikan dengan gerakan tangan dan data biometrik, mesin dapat memperoleh gambaran tentang perasaan Anda. Pembelajaran mesin sangat baik dalam melihat pola dalam sejumlah besar data sensorik. Organisasi telah menggunakan data ini untuk membantu mengatur rencana kota dan memecahkan masalah iklim. Harapan saya adalah suatu hari nanti hal ini akan memungkinkan kita untuk lebih memahami satu sama lain.

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

Lima Prinsip Sensorik

Zach Lieberman dan Molmol Kuo, mantan seniman residen di Adobe, mengusulkan penggunaan pelacakan wajah AR sebagai input ke instrumen musik. Mata yang berkedip dapat memicu animasi dan gerakan mulut dapat menghasilkan musik.

Seniman mendobrak batasan dan menciptakan cara baru melihat dunia dengan teknologi. Kita dapat meminta seniman untuk menciptakan pengalaman baru yang belum pernah kita pertimbangkan sebelumnya. Karena semakin banyak seniman yang mendalami komputasi spasial dan desain sensorik, kita akan membutuhkan seperangkat prinsip untuk membantu memandu pengalaman ke arah yang dapat dipahami pengguna. Generasi pertama pengguna Desain Sensorik tidak akan memiliki prakonsepsi yang jelas tentang apa yang diharapkan. Prinsip desain dapat mempermudah adopsi dan meningkatkan pengalaman

komputasi spasial secara keseluruhan.

Berikut ini adalah lima prinsip Desain Sensorik yang dibuat untuk memandu desainer dalam menciptakan pengalaman komputasi spasial yang menarik dan mudah dipahami.

1. Pengalaman Intuitif Bersifat Multisensorik

Produk kami akan intuitif jika bersifat multisensorik. Dengan memungkinkan alat kami untuk menerima dan menggabungkan berbagai indera, kami akan memungkinkan produk menjadi lebih tangguh dan mampu memahami maksud pengguna dengan lebih baik. Kita adalah makhluk multisensori, jadi menambahkan lebih banyak indra akan meningkatkan kegembiraan dalam sebuah pengalaman. Melihat konser sebuah band lebih berkesan daripada mendengarkan rekamannya melalui headphone. Terjun payung adalah pengalaman yang lebih mengubah hidup daripada menonton videonya. Kita senang menghabiskan waktu bersama teman-teman secara langsung daripada sekadar Facebook atau Snap. Oksitosin, hormon ikatan sosial, dilepaskan saat kita merasakan pelukan sungguhan, bukan saat kita mengklik tombol 'suka'.

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

2. 3D Akan Menjadi Norma

Dalam 5 hingga 10 tahun, desain digital 3D akan menjadi hal yang biasa seperti desain digital 2D saat ini. Seperti fotografi, penerbitan desktop, dan internet sebelumnya, kita akan membutuhkan alat desain, perangkat keras kelas konsumen, dan layanan cloud yang tersedia, mudah digunakan, dan cepat dipahami oleh semua orang. Saat ini, kami bersenang-senang bereksperimen dengan AR seluler, menggunakannya sebagai filter efek khusus dari dunia nyata, yaitu wajah kita. Di masa depan, kehidupan dengan AR akan lebih normal daripada swafoto bagi generasi milenial saat ini.

Sebentar lagi kita akan dapat berkreasi di seluruh lingkungan 3D kita menggunakan suara, gerakan tangan, dan lingkungan itu sendiri. Tubuh kita akan menjadi tetikus dunia komputasi spasial masa depan, dan dunia di sekitar kita akan dapat diklik, diedit, dan didesain ulang. Input tradisional seperti papan ketik, tetikus, dan layar sentuh membuat perangkat lunak menjadi rumit secara alami. Mengendalikan perangkat lunak secara spasial dengan semua indra alami kita dan tubuh manusia akan mengubah cara kita mengekspresikan kreativitas manusia.

Di dunia AR tanpa teknologi 2D, mungkin tampak konyol untuk melihat peta dua dimensi di perangkat seluler kita, alih-alih melihat melalui kaca mata AR kita untuk melihat petunjuk arah 3D yang diletakkan di atas jalan atau trotoar di depan kita. Menonton video terlebih dahulu untuk menyiapkan sistem audio rumah Anda akan tampak kuno ketika instruksi AR yang langsung ditumpangkan pada peralatan memandu Anda dengan segera.

Setiap orang akan dapat berkreasi ketika inspirasi datang kepada kita di tempat mana pun kita berada, tidak hanya saat kita berada di meja kerja. Jika itu adalah warna, cahaya, tekstur, gerakan, suara, atau bahkan sebuah objek, mereka dapat menangkapnya dalam bentuk 3D dengan perangkat AR mereka. Kita berharap dapat berkreasi menggunakan lingkungan 3D dan suara serta gerakan tangan kita sebagai mekanisme input, bukan mouse atau keyboard. Input tradisional seperti keyboard, mouse, dan layar sentuh membuat perangkat lunak menjadi rumit secara alami. Mengendalikan perangkat lunak dengan semua indra kita dalam bentuk 3D akan melepaskan kekuatan super kreatif kita. Misalnya, saya disleksia, jadi menuangkan pikiran saya ke atas kertas sangatlah membuat frustrasi. Ketika secara fisik menuliskan kata-kata, aliran kreatif saya hilang, dan saya menjadi tidak bisa berkata-kata. Saya menulis karya ini menggunakan teknologi suara-ke-teks. Teknologi ini tidak sempurna, tetapi membantu saya menuangkan kata-kata dan suara saya di atas kertas.

3. Desain Menjadi Sifat Fisik

Produk kita harus bersifat fisik secara alami. Desain yang ditempatkan di dunia hanya akan diterima jika berfungsi secara alami dan manusiawi. Kita akan tetap berteriak pada Alexa hingga teknologi mendengarkan dan merespons sebaik yang dilakukan teman-teman kita. Ada standar UI baru saat desain memasuki dunia.

Standar antarmuka pengguna baru untuk desain spasial menuntut desain digital yang ditempatkan di dunia bertindak seolah-olah nyata secara fisik. Kita berharap cangkir virtual akan pecah seperti cangkir fisik jika kita melemparkannya ke tanah. Sama seperti desain layar yang dipicu oleh klik tetikus atau ketukan layar, desain di dunia dipicu oleh indera kita. Desain dan interaksinya harus terasa alami dan dalam konteks dengan dunia di sekitarnya. Terkadang kita dapat melanggar aturan ini, selama pengguna tidak menganggap aplikasinya juga rusak.

4. Desain untuk yang Tidak Dapat Dikendalikan

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 mengendalikan 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 Project Aero Adobe yang tak kenal takut, ketika dia berkata, "AR memaksa kreator untuk memikirkan rasa agensi pemirsa (atau pilihan yang diarahkan sendiri), dan ini menumbuhkan lebih banyak empati terhadap pemirsa." Memberikan pemirsa kendali atas kamera memberi mereka peran untuk dimainkan. Mereka menjadi bagian dari kreator. 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.



Gambar 2.4. Gambar yang dibuat dengan aplikasi Weird Type yang tersedia di App Store Apple

Cara lain kami menemukan desain untuk hal yang tidak dapat dikontrol adalah melalui mata para seniman kami yang menjelajahi program Residensi AR Adobe yang diadakan selama tiga bulan, tiga kali per tahun. Dua seniman residensi ini adalah Zach Lieberman dan Mol-mol Kuo. Mereka berkolaborasi untuk membuat Weird Type, aplikasi iOS AR yang memungkinkan Anda menulis dan menganimasikan apa saja, di mana saja dalam ruang 3D. Setelah meluncurkan aplikasi, kita semua dapat duduk santai dan menyaksikan bagaimana tipografi dalam ruang dapat dibayangkan kembali. Orang-orang menggunakan Weird Type untuk memandu seseorang melalui sebuah gedung, menceritakan kisah tentang suatu lokasi; membangun patung; dan berbagi perasaan melalui angin dengan menganimasikan kata-kata, terbang dan menyebarkan huruf secara acak ke ruang angkasa, membuat teks tampak lebih seperti salju (Gambar 2.4). Bentuk-bentuk komunikasi baru ini ditemukan dengan menyediakan agensi kreatif bagi pemirsa AR, yang dengan sendirinya membuka media kreativitas baru.

5. Membuka Kekuatan Kolaborasi Spasial

Saya yakin 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, secara ajaib merancang hal-hal digital-fisik dengan keputusan yang diperkuat oleh AI bersama anggota tim manusia yang nyata, maka kekuatan koneksi emosional dan fisik jarak jauh menjadi pendorong adopsi komputasi spasial. Dengan kata lain, dapat dikatakan, koneksi manusia adalah aplikasi pembunuh untuk AR.

Salah satu seniman residensi Adobe, Nadine Kolodzey, membawa gagasan kolaborasi AR satu langkah lebih jauh ketika dia berkata, "Saya ingin orang tidak hanya melihat gambar saya, tetapi juga menambahkan sesuatu." Kami kemudian menyadari bahwa dia memberi pemirsa agensi, kemampuan untuk menjadi seniman juga. Pada saat itu Nadine menjadi pembuat alat dan pemirsa menjadi seniman. Dengan cara ini, AR memberikan kemampuan bercerita kepada semua orang, seperti halnya penerbitan desktop untuk media cetak dan internet untuk pengetahuan.

2.4 KISAH AR ADOBE

AR yang dipandu oleh AI akan mengubah secara mendasar apa yang diciptakan desainer, bagaimana 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. Sasaran Project Aero adalah menghadirkan media AR baru ke semua produk kami dan membangun disiplin desain untuk komputasi spasial yang digerakkan oleh AI. Berikut ini adalah sebagian dari masa depan komputasi spasial seperti yang saya lihat saat ini.

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

Desainer akan segera dapat berkreasi saat momen inspirasi menghampiri mereka, di mana pun mereka berada. Jika itu adalah 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 bersenang-senang dengan AR seluler yang menggunakannya sebagai filter efek khusus dunia. Kami tahu bahwa merupakan tanggung jawab kami di Adobe untuk membangun alat interaktif, animasi, dan memperkaya yang menjembatani kesenjangan antara masa kini dan masa depan bagi para desainer kami dan desainer baru yang sedang naik daun.

Baru-baru ini, ketika artis residen kami Nadine Kolodzey berkata, "Saya ingin orang-orang tidak hanya melihat gambar [saya], tetapi juga menambahkan sesuatu," kami menyadari bahwa ia memanfaatkan kebutuhan yang muncul akan desain kolaboratif waktu nyata yang dimungkinkan dengan ponsel pintar yang mendukung AR dan cloud AR. Adidas, "merek kreator", menganggap konsumennya sebagai kreator juga. Jadi, ketika kami meminta Adidas untuk membantu membangun alat AR yang "tepat" untuk kolaborasi kreator, mereka langsung melakukannya. Namun, kisah 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; semuanya digerakkan 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 proprioepsi (indra ruang di sekitar kita) untuk memahami dunia. Mesin yang meniru indra manusia seperti indra penglihatan kita dengan AR hanya dirancang dengan baik jika berfungsi seperti yang diharapkan: secara manusiawi. Kami akan tetap membentak Alexa jika tidak memahami apa yang saya katakan sebaik teman saya. Standar baru desain sensorik yang baik ini telah membuat Adobe Design memikirkan kembali prinsip desain, peran desainer, dan mekanisme inti alat kami.

Kesimpulan

Saat kita menghadapi dunia komputasi spasial yang baru ini, saya teringat pada apa yang dikatakan Scott Belsky, kepala bagian produk Adobe: “Kreativitas adalah keahlian paling manusiawi di dunia dan, meskipun ada banyak perangkat dan media baru, orang-orang kreatif tetap menjadi pusatnya. Semakin dunia berubah, semakin penting pula kreativitas.” Saya melihat kreativitas meledak di semua bagian kehidupan kita. Kreativitas sama pentingnya dengan literasi. Jadi, mari kita sediakan alat-alat kreatif kita untuk semua orang, yang mencakup berbagai kemampuan, budaya, dan menghormati hak privasi dan transparansi orang.

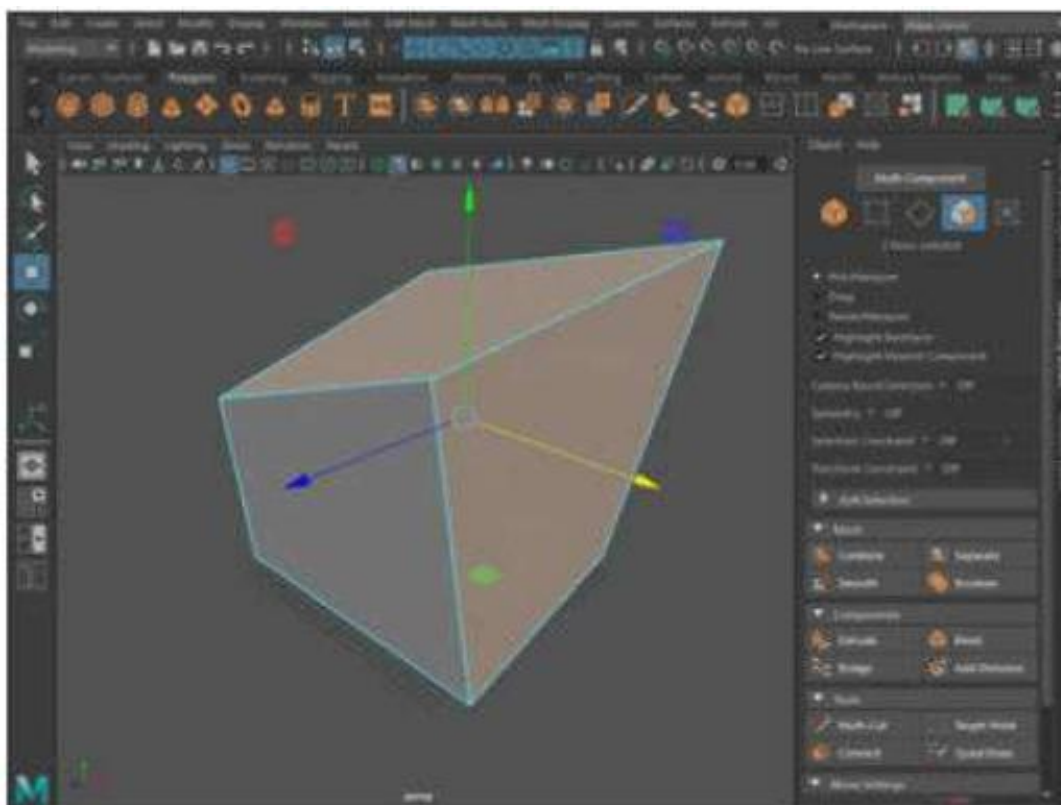
Pada tahun 1970, desainer industri Dieter Rams menulis 10 prinsip untuk desain yang baik. Saat ini, kita hidup di dunia di mana desain dapat menolak, menanggapi, atau merasakan apa pun. Desain adalah sesuatu yang hanya terjadi sekali. Rams tidak memiliki teknologi pencitraan adaptif yang memahami maksud, mengingat tindakan masa lalu, dan menyediakan personalisasi antarmuka Anda. Desain telah berubah. Desain merespons sistem saraf API secara empatik. Kitalah orang-orang yang sebenarnya membangun fondasi untuk periode ini. Kitalah para desainer, insinyur, ilmuwan kognitif, wirausahawan, dan banyak lainnya. Jika kita menantang diri kita sendiri untuk melihat lebih jauh dari sekadar teknologi dan memfokuskan sebagian energi untuk membangun fondasi desain yang baik, kita sebenarnya dapat menciptakan masa depan yang sedikit lebih berempati secara alami. Mari kita tantang diri kita sendiri untuk mengembangkan alat yang menggunakan teknologi penginderaan yang memungkinkan produk menunjukkan empati untuk masa depan yang lebih baik.

BAB 3

REALITAS VIRTUAL UNTUK SENI

3.1 CARA YANG LEBIH ALAMI UNTUK MEMBUAT SENI 3D

Secara tradisional, membuat seni 3D digital lebih seperti membuat rancangan daripada melukis atau memahat. Banyak tantangan yang dihadapi 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 titik sudut atau titik dalam ruang. Memindahkan titik-titik ini dalam ruang 3D dengan tetikus 2D membutuhkan pemikiran yang jauh lebih abstrak daripada seni tradisional, yang lebih langsung diterapkan.



Gambar 3.1. Antarmuka untuk perangkat lunak pemodelan dan animasi 3D populer Autodesk Maya (sumber: CGSpectrum)

Melihat antarmuka untuk program 3D paling populer seperti Autodesk Maya (Gambar 3-1) dan 3D Studio mencerminkan kompleksitas ini. Karena tantangan ini, sangat sedikit orang yang dapat membuat seni 3D. Kemudian muncul gelombang baru program pemodelan 3D, seperti Z-Brush dari Pixologic (Gambar 3-2) yang memiliki cara pandang yang berbeda secara mendasar. Program semacam itu 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 secara langsung memanipulasi geometri dengan gerakan yang lebih alami, kreasi seni 3D semakin demokratis. Namun, meskipun antarmukanya lebih langsung, tetap saja sulit untuk bekerja pada objek 3D melalui tampilan 2D dan antarmuka 2D. Dengan diperkenalkannya gelombang realitas virtual (VR) konsumen, semuanya berubah.



Gambar 3.2. Seorang seniman digital bekerja dengan Wacom Pen Tablet dan Pixologic Z-Brush (sumber: Wacom)

Ketika kebanyakan orang berpikir tentang VR, mereka berpikir tentang head-mounted display (HMD), dengan sensor dan layar yang mengambil alih bidang visual dan sepenuhnya membenamkan seseorang ke dalam dunia digital. Namun yang sama pentingnya, jika tidak lebih penting, adalah perangkat input atau pengontrol yang dilengkapi dengan sensor serupa yang memungkinkan Anda berinteraksi dan memanipulasi dunia digital dengan cara yang alami dan intuitif. VR HMD menjadi tampilan 3D terbaik, dan pengontrol tangan yang dilacak menjadi antarmuka 3D terbaik. Tidak ada contoh yang lebih baik tentang kekuatan VR selain dalam aplikasi yang menggabungkan tampilan dan input yang unik untuk memungkinkan pengguna berkreasi dan mengekspresikan diri mereka sendiri seperti yang belum pernah ada sebelumnya. Untuk gelombang VR modern, semuanya dimulai dengan aplikasi bernama Tilt Brush, yang dapat Anda lihat pada Gambar 3.3.

Tilt Brush dikembangkan oleh perusahaan rintisan Skillman & Hackett yang beranggotakan dua orang dan merupakan salah satu program seni pertama dalam VR modern. Karena dirancang dengan Oculus Development Kit 2, yang hanya memiliki HMD tetapi tidak memiliki perangkat input spasial, duo tersebut merancang untuk digunakan dengan tablet pena Wacom. Pengguna menggambar pada bidang 2D yang dapat dimiringkan dan digerakkan untuk melukis dalam beberapa bidang untuk dibuat dalam 3D. Ketika Valve dan HTC mengeluarkan kit pengembang Vive, Skillman & Hackett memanfaatkan pengontrol tangan

yang dilacak sepenuhnya dan pelacakan skala ruangan yang disertakan untuk memungkinkan pengguna melukis secara intuitif dalam ruang 3D. Kini seluruh ruangan menjadi kanvas dan cat digital mengalir dari ujung pengontrol tangan dan melayang di angkasa, menciptakan nuansa magis yang melampaui kenyataan tetapi terasa sepenuhnya alami dan mudah dilakukan. Google kemudian 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. Aplikasi ini bukan hanya pelopor sejati dalam aplikasi seni, tetapi juga contoh cemerlang dari pengalaman pengguna (UX) yang sangat baik dan intuitif dalam VR.



Gambar 3.3. Gambar promosi untuk Google Tilt Brush

Tilt Brush selalu dirancang sebagai aplikasi konsumen dan, karenanya, selalu menjadi alat yang sangat mudah digunakan dengan desain yang sederhana dan menyenangkan. Aplikasi ini memiliki berbagai macam kuas yang memiliki efek visual seperti pencahayaan dan animasi, yang menciptakan tampilan bergaya yang sangat spesifik yang membedakannya dari alat lain di pasaran dan pengguna pertama kali dapat dengan cepat memperoleh hasil visual yang menakjubkan. Meskipun alat-alat tersebut cukup fleksibel untuk mengakomodasi berbagai gaya yang berbeda secara default, mudah untuk mengenali seni yang dibuat di Tilt Brush, seperti yang dapat Anda lihat pada Gambar 3.4.

Meskipun seninya mudah dikenali, Tilt Brush memiliki potensi yang tak terbatas. Alat 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 permainan yang dibuat dengan semua karya seni yang dibuat di Tilt Brush. Produk ini terus berkembang, dan dengan fitur dan fungsi baru, alat ini akan terus membuka jalan bagi jenis seni digital baru di dunia spasial.



Gambar 3.4. Lukisan VR Tilt Brush oleh seniman VR Peter Chan

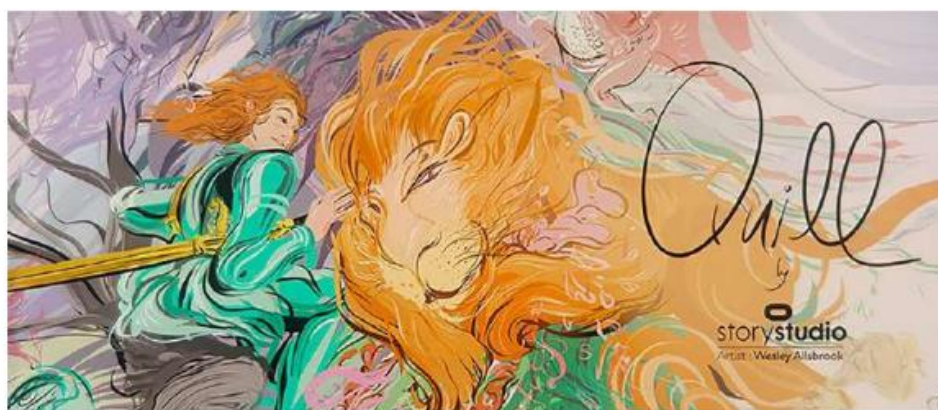


Gambar 3.5. Seniman VR Danny Bittman memberikan pertunjukan Tilt Brush secara langsung di VMWorld 2017 (foto oleh WMWare)

Karena kesuksesannya yang luar biasa di dalam dan luar industri VR, Google Tilt Brush adalah yang pertama mempopulerkan ide melukis VR. Aplikasi ini menggunakan metafora sapuan cat untuk kreasi yang sangat berbeda dari cara pembuatan seni 3D sebelumnya. Namun, aplikasi ini bukan satu-satunya program melukis VR yang ada. Tidak jauh dari tempat tim Google saat ini mengerjakan Tilt Brush, sebenarnya ada tim lain yang mengerjakan

pendekatan yang sangat berbeda terhadap lukisan VR sebagai bagian dari Oculus, yang disebut Oculus Story Studio.

Oculus Story Studio adalah grup pengembangan internal dalam Oculus untuk mengeksplorasi penceritaan dalam VR. Dua film pendek animasi VR pertamanya, *Lost* dan *Henry*, menggunakan alur kerja produksi seni yang cukup standar untuk menciptakan cerita pendek animasi real-time yang indah. Namun, untuk karya ketiganya, *Dear Angelica*, aplikasi ini menggunakan sesuatu yang sangat berbeda. *Dear Angelica* adalah mimpi surealis yang menggunakan gaya melukis 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 menciptakan program melukis VR-nya sendiri yang bernama Quill. Meskipun keduanya menggunakan pendekatan seperti sapuan lukisan untuk kreasi, keduanya menghasilkan hasil yang tampak sangat berbeda dan memiliki pendekatan UX yang sangat berbeda.



Gambar 3.6. Gambar promosi Quill dari Facebook yang menampilkan karya seni dari Dear Angelica oleh seniman Wesley Allsbrook

Quill dibuat 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. Quill juga tidak memiliki banyak efek bergaya yang dimiliki Tilt Brush, seperti pencahayaan waktu nyata atau kuas animasi. Hal ini memberi seniman lebih banyak kendali atas tampilan. Namun, dibutuhkan lebih banyak upaya untuk mendapatkan hasil tertentu. Quill juga menghadirkan banyak ide baru tentang seni VR, dengan fitur-fitur menarik seperti skala yang hampir tak terbatas, tampilan yang bergantung pada perspektif, dan animasi berbasis garis waktu. Direktur seni Oculus dan seniman residen Goro Fujita telah memelopori beberapa kreasi yang benar-benar unik yang menggunakan fitur-fitur ini, seperti karyanya *Worlds in Worlds*, 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 seni jika tidak hanya dibuat tetapi juga dilihat dalam VR.



Gambar 3.7. Memperbesar gambar Worlds in Worlds karya seniman VR Goro Fujita, yang dilukis di Quill dari Facebook

Animasi khususnya adalah bagian yang paling menonjol dari Quill dan memungkinkan generasi seniman dan animator 2D untuk bekerja dalam 3D dengan cara yang sangat familiar 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 dan nuansa visual animasi yang dibuat di Quill memiliki nuansa yang sangat organik, jauh lebih dekat dengan animasi 2D, tetapi hadir dalam 3D untuk menciptakan sesuatu yang orisinal dan istimewa yang akan berdampak jangka panjang pada penceritaan visual yang belum sepenuhnya kita pahami. Namun, lukisan virtual hanyalah salah satu pendekatan untuk menciptakan seni virtual. Pendekatan lain, yang lebih berbasis pada seni pahat, juga sedang dikerjakan oleh tim lain di Oculus Story Studio, dan pendekatan ini kemudian dikenal sebagai Medium (Gambar 3.8).



Gambar 3.8. Seni promosi Oculus Medium dari Oculus

Oculus Medium sebenarnya sudah dikerjakan jauh sebelum Quill. Sementara lukisan virtual memungkinkan seniman 2D tradisional untuk bekerja dalam 3D, pahatan virtual lebih 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, lalu menerapkan warna dan tekstur padanya. Seperti yang diilustrasikan pada Gambar 3-9, hal ini memungkinkan terciptanya model 3D yang lebih dekat dengan apa yang sudah dikenal dalam alur produksi 3D yang ada dan dapat lebih mudah diubah menjadi geometri poligon 3D untuk digunakan dalam permainan dan film atau bahkan dicetak 3D ke dunia nyata. Selama masa pertumbuhannya, program ini masih memerlukan beberapa pemrosesan dan penyempurnaan, tetapi bahkan pada tahap awal, sangat jelas betapa lebih mudahnya bagi banyak orang untuk membuat objek 3D dengan cara ini daripada aplikasi 3D non-VR tradisional. Sekarang, hal ini digunakan dalam seni konsep; pembuatan prototipe; desain arsitektur, produk, dan mainan; dan bahkan aset akhir game dan film. Penggunaan Medium mempercepat waktu iterasi desain, mengurangi praproduksi pada proyek hingga 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.



Gambar 3.9. Seniman VR sedang mengerjakan pahatan 3D di Oculus Medium (gambar dari Oculus)

VR untuk Animasi

VR tidak hanya hebat untuk membuat karya seni 3D; VR juga bersifat transformatif untuk menghidupkan karya seni tersebut. Mirip dengan kreasi seni 3D tradisional, animasi 3D tradisional mengandalkan gerakan dan manipulasi titik 3D dalam ruang 3D tetapi dengan antarmuka 2D pada tampilan 2D. Dengan VR, kini ada peluang untuk menata ulang alur kerja

ini agar lebih alami dan intuitif. Salah satu pendekatannya 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 Tвори mengambil pendekatan seperti itu. Namun untuk animasi karakter, yang biasanya paling rumit dan memakan waktu, VR memiliki trik yang lebih baik. Motion capture adalah teknik memasang pelacak gerak pada aktor, lalu menggunakan kamera untuk menangkap gerakan aktor tersebut, lalu menerapkannya pada model 3D untuk animasi karakter yang tampak nyata. Teknik ini telah digunakan untuk animasi dalam game dan film selama beberapa dekade, tetapi memerlukan peralatan mahal dan banyak pembersihan. Agar VR berfungsi, komputer harus melacak kepala dan tangan pengguna dengan benar. Hal ini menjadikan VR sebagai sistem motion capture yang sangat mendasar secara default, yang biayanya hanya sepersepuluh atau kurang dari biaya pengaturan motion-capture penuh.

Pembuat program animasi VR, Mindshow, menyadari hal ini dan bahwa siapa pun yang memiliki headset VR berpotensi melakukan animasi karakter dasar dan menjadi pendongeng. Sekarang, alih-alih menggambar ribuan gambar atau menghabiskan waktu berhari-hari menggerakkan titik-titik di layar, seorang animator dapat memerankan adegan seperti yang dilakukan aktor di lokasi syuting. Demikian pula, aktor yang sama kemudian dapat masuk dan memerankan bagian lain, bekerja dengan rekaman. Hal ini mempersingkat animasi dasar dengan mengubah proses yang biasanya memakan waktu seminggu menjadi beberapa menit saja, dan memungkinkan satu orang saja menjadi studio animasi. Meskipun kualitasnya belum setara dengan Hollywood, bahkan pada tahap awal, hasilnya mengesankan dan menjanjikan. Proses ini merupakan 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.



Gambar 3.10. Seorang animator memerankan sebuah pertunjukan dalam VR yang diterjemahkan ke karakter digital secara real-time dengan Mindshow

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

VR telah mendemokratisasi kreasi 3D dengan cara yang sama seperti komputer desktop mendemokratisasi kreasi 2D. Kini, orang-orang yang tidak pernah menganggap diri mereka sebagai seniman tradisional dapat bergabung dalam perbincangan tentang pembentukan masa depan seni visual dan spasial. Dengan VR, siapa pun dapat membuat karya instalasi seukuran gudang yang menghabiskan biaya puluhan ribu untuk bahan saja. Namun, yang lebih hebat lagi, kini mereka dapat membuat karya instalasi seukuran planet yang tidak mungkin dibuat di dunia nyata. VR benar-benar kanvas terbaik, tidak hanya untuk memberdayakan seniman, tetapi juga dengan menyalakan jiwa seniman dalam diri setiap orang untuk menciptakan karya yang hanya dibatasi oleh imajinasi mereka.

BAB 4

OPTIMALISASI SENI 3D

4.1 PENDAHULUAN

Dalam bab ini, saya membahas mengapa pengoptimalan merupakan tantangan besar dalam hal pengembangan aset untuk realitas virtual (VR) dan realitas tertambah (AR), dengan fokus utama pada sisi VR. Saya berbagi berbagai pendekatan dan proses berpikir untuk dipertimbangkan dalam berbagai bidang yang terlibat dalam pembuatan seni 3D. Alih-alih berfokus pada alat, teknik, dan tutorial tertentu, Anda akan melihat gambaran menyeluruh tentang apa yang dapat Anda lakukan untuk mengoptimalkan seni 3D Anda. Saya berbagi mengapa penting untuk benar-benar menjadikan pengoptimalan sebagai prioritas tinggi dalam keseluruhan proses desain. Mari kita mulai dengan membahas mengapa pengoptimalan untuk VR merupakan hal baru bagi sebagian besar seniman yang mulai bekerja dengan konten VR.

Monitor LCD 2D pada umumnya memiliki kecepatan refresh 60 Hz. Saat Anda melihat monitor datar yang berjalan pada kecepatan ini, Anda akan mendapatkan pengalaman visual yang fantastis. Hal ini memungkinkan pengembangan aset tradisional menjadi "berat", dengan berat dalam keadaan ini berarti aset memiliki jumlah poli yang lebih tinggi dan gambar tekstur yang lebih besar, bersama dengan pemandangan 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 perlu berjalan pada 90 bingkai per detik (FPS). Jika pengalaman Anda berjalan lebih rendah dari 90 FPS, Anda berisiko menyebabkan ketidaknyamanan bagi pengguna Anda. Ketidaknyamanan dapat mencakup sakit kepala, mual, dan nyeri mata. Ini akan menghasilkan pengalaman di bawah standar yang ingin segera ditinggalkan pengguna. Karena tingkat ketidaknyamanan yang tinggi ini, Anda tidak boleh memaksa pengguna ke dalam pengalaman dengan kecepatan bingkai rendah.

Segera, VR akan memasuki bidang lain. Alih-alih hanya membuat VR, Anda akan membuat alat, pengalaman, aplikasi, dan sebagainya di suatu bidang, dan VR adalah media Anda. Beberapa bidang ini sudah terbiasa dengan pembuatan konten 3D, dan pengoptimalan akan menjadi penting untuk dipelajari. Akan ada banyak individu yang latar belakangnya mungkin tidak mempersiapkan mereka dengan baik untuk perubahan besar dalam pengembangan aset ini, dan 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:

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

Contoh-contoh ini memiliki manfaat yang tidak lagi dapat dimanfaatkan untuk VR dan AR. Tidak termasuk headset tethering 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 gambar, 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 yang dimilikinya atau jumlah data rendering yang dibutuhkan komputer untuk memvisualisasikannya. Berikut ini adalah contoh dari apa yang dapat terjadi ketika seseorang yang baru mengenal pengoptimalan mencoba mengembangkan untuk VR:

Buat Model 3D Kamera

Model yang dikirim

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

Masalah

Model tersebut menghabiskan sebagian besar anggaran hitungan poli adegan, jadi kualitas konten lainnya harus dikorbankan. Jika konten lainnya perlu mempertahankan kualitas dan ukurannya saat ini, Anda akan mengalami masalah kinerja. Pengembang perlu menyeimbangkan seni mana yang memiliki prioritas lebih rendah dan memberi lebih banyak ruang untuk model kamera poli tinggi.

Namun mengapa ini menjadi masalah besar bagi pengembang? Jika latar belakang orang tersebut adalah membuat model yang akan 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 dapat diterjemahkan dengan baik ke dalam rendering VR waktu nyata. Seperti yang dinyatakan sebelumnya, masalah kinerja akan mencegah pengguna akhir mendapatkan pengalaman yang berkualitas.

Opsi yang Perlu Dipertimbangkan

Berikut ini beberapa hal yang dapat dicoba untuk memecahkan masalah:

- ❖ Menjalankan alat desimasi untuk mengurangi jumlah poli secara otomatis. Anda dapat menemukannya di perangkat lunak pemodelan 3D yang populer. Alat ini biasanya berfungsi dengan baik dalam menghilangkan 50% jumlah segitiga tanpa memengaruhi bentuk dan siluet model.
- ❖ Lihat tata letak UV model (sumbu tekstur 2D yang diproyeksikan ke 3D). Apakah tekstur UV ditata untuk memanfaatkan seluruh ruang persegi? Apakah UV berskala dan memprioritaskan area yang paling membutuhkan detail untuk ditampilkan? Kami akan membahas tekstur dan material secara lebih rinci nanti di bab ini.

Opsi bagus lainnya yang perlu dipertimbangkan adalah apakah model Anda akan memasuki tempat VR sosial yang mengizinkan konten yang dibuat pengguna (UGC)? Ini kemungkinan besar akan terus menjadi tantangan untuk waktu yang lama. Perlu diingat bahwa semakin banyak avatar di suatu ruang, semakin sedikit anggaran yang harus dimiliki setiap orang untuk

menghormati frame rate setiap orang, sehingga memungkinkan pengalaman yang baik.

Solusi Ideal

Solusi terbaik adalah mengurangi jumlah segitiga model ke jumlah minimum absolut yang dapat dimilikinya tanpa memengaruhi 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, objek tersebut memberikan kelonggaran yang cukup 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 performa pengalaman Anda. Semakin banyak yang Anda tambahkan, semakin Anda perlu mempertimbangkan seberapa dekat Anda dengan anggaran Anda. Bicarakan dengan tim Anda untuk menentukan berapa anggaran ideal Anda. Pertimbangan lain untuk ke mana model 3D Anda akan pergi termasuk platform VR sosial. Ada beberapa platform VR sosial yang dibuat dengan UGC. Anda kemungkinan besar akan hadir di tempat-tempat ini sebagai avatar, dan jika Anda dapat menyesuaikan avatar Anda, ingatlah bahwa semua yang Anda pelajari di sini juga berlaku di sana. Seperti halnya dengan hal-hal lain yang akan Anda pelajari di sini, cobalah untuk menjaga semua hal tentang avatar Anda dan apa yang Anda kenakan tetap poli rendah dan dengan jumlah panggilan gambar terkecil yang dapat Anda buat. Anda mungkin menemukan filter yang membantu mengurangi jumlah unduhan yang Anda buat, tetapi pikirkan terlebih dahulu apa yang Anda minta agar layar orang-orang tampilkan. Perhatikan perangkat keras dan koneksi mereka dan buat diri Anda mudah dirender dan diunduh.

4.2 ANGGARAN HITUNGAN POLIGON

Apakah Anda memiliki jumlah poligon konkret yang tidak dapat Anda lewati dalam sebuah adegan? Apakah Anda memiliki batasan jumlah poligon per model?

Selalu perhatikan jumlah segitiga. Jumlah sisi tidak akan selalu akurat untuk mengukur berapa banyak poligon yang Anda miliki pada model Anda. Sisi yang terdiri dari empat titik sudut, seperti persegi, sebenarnya adalah dua segitiga dalam satu.

Hapus semua sisi yang tidak akan pernah terlihat. Jika Anda membuat lingkungan jalan di mana bagian dalam bangunan tidak akan pernah dimasuki, adegan tersebut hanya memerlukan fasad bangunan. Bagian belakang dinding dan bagian dalam tidak diperlukan. Jika Anda menggunakan konten yang telah dibangun, Anda dapat menghapus semua yang tidak akan terlihat.

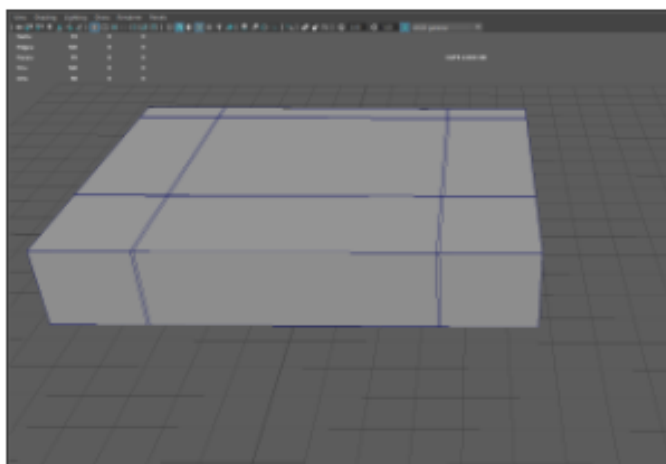
Jika Anda mengerjakan model 3D yang akan tetap jauh dari Anda dalam pengalaman tersebut, model tersebut tidak memerlukan semua detail yang mungkin Anda inginkan jika lebih dekat. Pintu dan jendela dapat dimodelkan dan diberi tekstur dengan lebih sedikit detail. Semakin rendah jumlah poligon Anda, semakin baik.

Bagian berikut menyajikan beberapa hal yang perlu diingat saat membuat model.

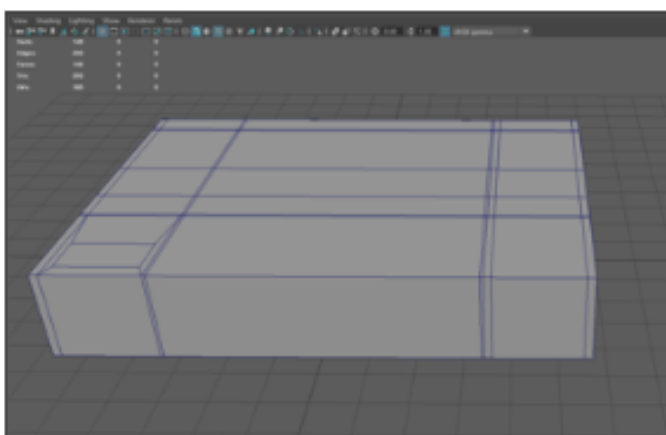
Topologi

Periksa loop tepi dan temukan loop tepi yang tidak memberikan kontribusi apa pun pada bentuknya. 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 mempertahankan bentuk dan memiliki kurva yang diinginkan, Anda sedang dalam perjalanan untuk mengurangi jumlah poligon. Bahkan ada beberapa area tempat Anda dapat menambahkan tepi untuk digabungkan dengan yang lain. Periksa kembali bahwa semua loop tepi yang dihapus tidak meninggalkan simpul apa pun, dan hapus simpul yang tidak menghubungkan tepi mana pun.

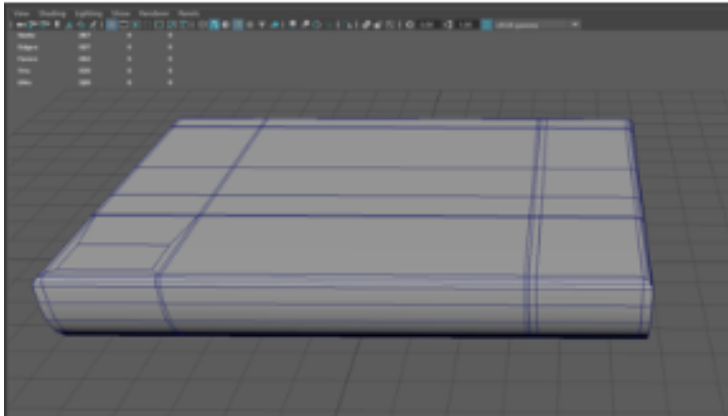
Gambar 4.1 hingga 4.4 menunjukkan proses pembuatan konsol gim. Pada Gambar 4.1, Anda dapat melihat, dalam mode rangka kawat, awal pembuatannya menggunakan loop tepi untuk menentukan di mana lebih banyak geometri akan dibutuhkan; ada dua langkah antara pengurangan poligon dan versi akhir pada Gambar 4.4, yang menghasilkan lebih sedikit segitiga daripada tahap pertama.



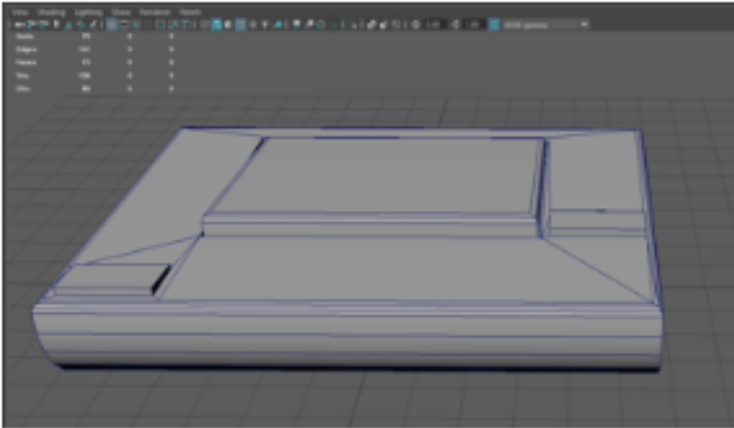
Gambar 4.1. Tahap pertama pada konsol gim: bentuk dasar terbentuk; jumlah segitiga: 140



Gambar 4.2. Tahap kedua pada konsol gim: menentukan di mana sisi dan tepi akan terangkat dan melengkung; jumlah segitiga: 292

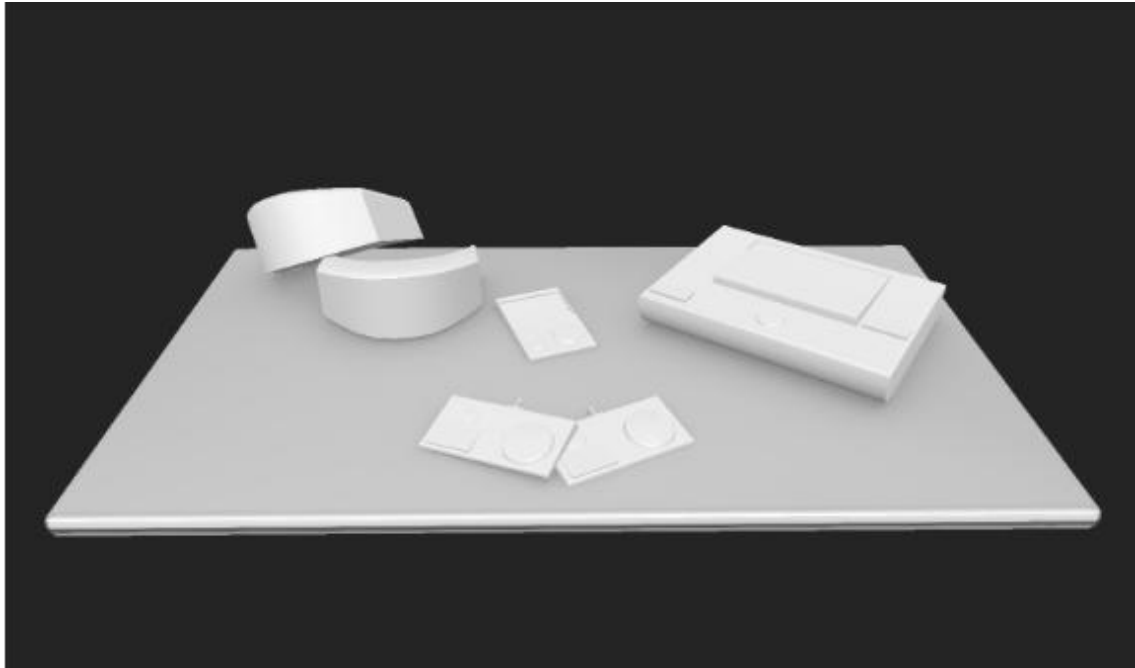


Gambar 4.3. Tahap ketiga pada konsol gim: melembutkan tepi dan mulai berpikir tentang penghilangan tepi; jumlah segitiga: 530



Gambar 4.4. Versi keempat dan terakhir: tepi yang tidak berkontribusi pada bentuk model dihilangkan; jumlah segitiga: 136

Proses yang ditunjukkan pada Gambar 4.1 hingga 4.4 mirip dengan proses yang dilakukan untuk memodelkan beberapa aset lagi untuk set konsol gim ini. Gambar 4-5 menggambarkan hasil set tersebut. Set ini berisi beberapa model dalam satu jaring gabungan yang siap untuk diberi tekstur. Bersama-sama, mereka akan berbagi satu atlas tekstur. Nanti di bab ini, Anda akan melihat tampilan atlas tekstur.



Gambar 4.5. Melihat aset sebelum menerima materialnya

Berikut beberapa hal yang perlu diingat saat membuat model:

- ✓ Hindari n-gon. N-gon adalah bidang yang memiliki lebih dari empat sisi. Sebagian besar mesin memiliki masalah dengan n-gon. N-gon dapat menyebabkan masalah tabrakan, dapat ditampilkan sepenuhnya salah, dan bahkan dapat tidak terlihat. Perangkat lunak pemodelan 3D seperti Maya dari Autodesk memberi Anda opsi untuk membersihkan pandangan dan menghapus n-gon yang ditemukan.
- ✓ Jalankan pembersihan menggunakan perangkat lunak pemodelan Anda untuk menemukan dan menghapus semua bidang koplanar. Anda mungkin sering menemukan bidang tersembunyi yang tersembunyi di dalam klonnya sendiri, yang akan tampak tidak terlihat oleh mata telanjang dan akan meningkatkan jumlah poligon Anda. Ada juga masalah z-fighting. Z-fighting terjadi saat ada dua bidang yang menempati ruang 3D yang sama.
- ✓ Nyalakan penampil untuk memastikan bahwa normal menghadap ke arah yang diinginkan. Normal akan dirender dari satu arah di mesin pilihan Anda, jadi jangan biarkan perangkat lunak pemodelan 3D membodohi Anda dengan rendering dua sisi.

Penting untuk memikirkan semua pertimbangan ini di awal sebelum Anda mulai mengerjakan model 3D.

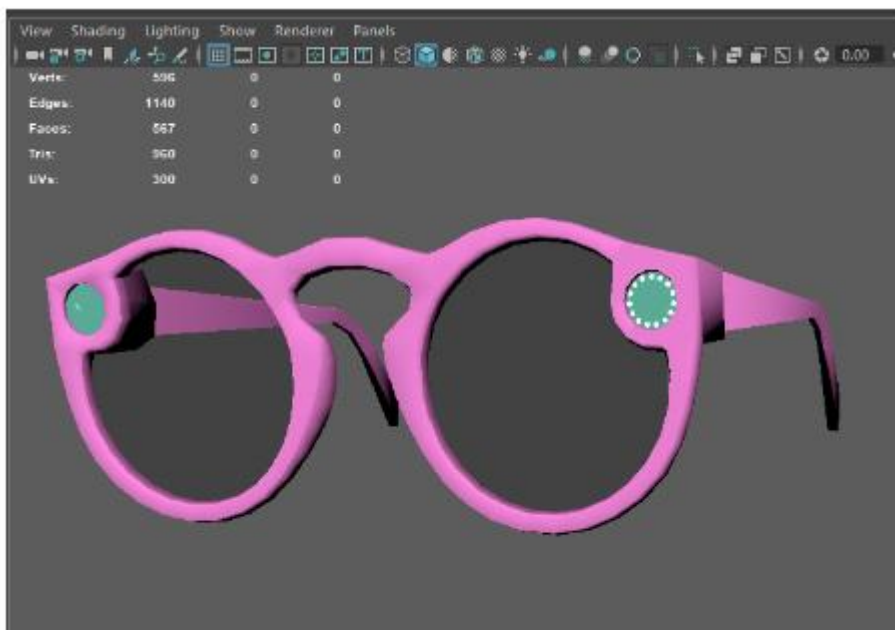
Gambar 4.6 menyajikan contoh proyek pengoptimalan yang saya kerjakan sendiri. Saya diberi model kacamata 3D yang terdiri dari 69.868 segitiga. Jumlah ini lebih banyak dari avatar saya sendiri, yaitu sekitar 40.000, termasuk tubuh, pakaian, rambut, dan aksesori. Kacamata tersebut dibeli dari toko daring yang menjual file model 3D, dan jelas bahwa seniman menciptakan ini dengan maksud untuk menunjukkan bahwa mereka dapat membuat model yang sesuai dengan objek dalam "kehidupan nyata." Seniman tersebut membuat model setiap bagian dengan tangan, termasuk engsel untuk pelipis. Karena saya akan membuat kacamata

ini untuk dikenakan orang-orang di platform VR sosial, saya tahu bahwa sebagian besar detailnya tidak akan terlihat atau dibutuhkan, jadi saya menghapus banyak bagian tersebut. Saya berhasil mempertahankan tampilan kacamata sambil menghapus dan mengalihkan sebagian besar simpul tepi. Hasil akhirnya hanya sekitar 1.000 segitiga.



Gambar 4.6. Model kacamata untuk digunakan di ruang VR sosial

Khususnya, untuk penggunaan AR, mendapatkan kurang dari 1.000 segitiga akan menjadi keharusan mutlak.



Gambar 4.7. Contoh tepi keras di sekitar bagian bingkai yang membulat

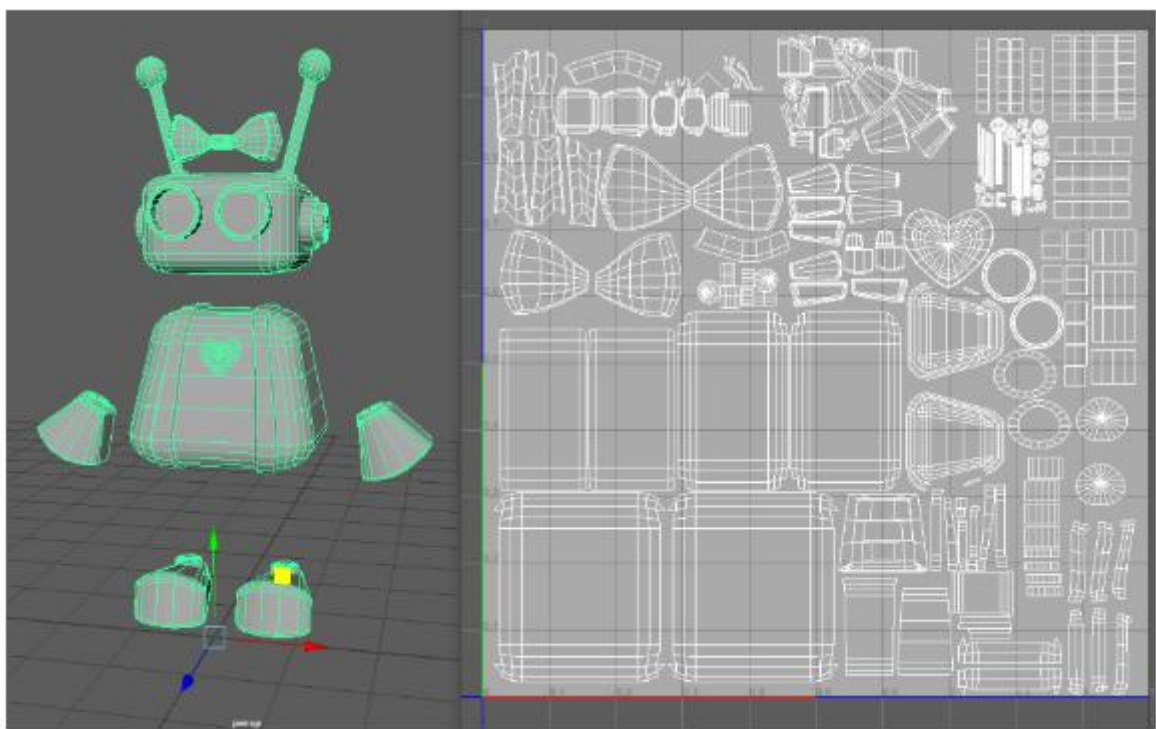
Pada Hololens, misalnya, Anda akan ingin membidik maksimal sekitar 60.000 segitiga di seluruh pemandangan. Kecuali jika aplikasi tersebut sangat berfokus pada pemeriksaan

sepasang kacamata hitam yang sangat detail, Anda akan ingin menguranginya sepenuhnya seperti yang saya lakukan dalam contoh ini. Gambar 4.7 menyajikan tampilan dekat yang memperlihatkan tepi keras yang dapat Anda lihat di sekitar bagian bingkai yang membulat, yang tidak terlihat jika dilihat dari kejauhan.

Memanggang

Trik lain yang dapat Anda lakukan untuk membantu penghitungan poli adalah dengan memanggang detail model poli tinggi Anda ke dalam model poli rendah. Dengan melakukannya, Anda dapat membuat peta normal yang akan mengelabui pemirsa agar melihat tinggi dan kedalaman yang tidak ada pada geometri itu sendiri.

UV digunakan untuk menggambarkan model 3D pada bidang datar. UV tersebut merujuk pada tekstur yang digunakan model agar informasi warna dan material dipetakan sesuai dengannya. Untuk pengoptimalan, mari kita bahas pendekatan pembuatan tekstur yang dibuat, dengan tujuan untuk menjaga agar jumlah panggilan gambar tetap rendah. (Lebih lanjut tentang panggilan gambar nanti.)



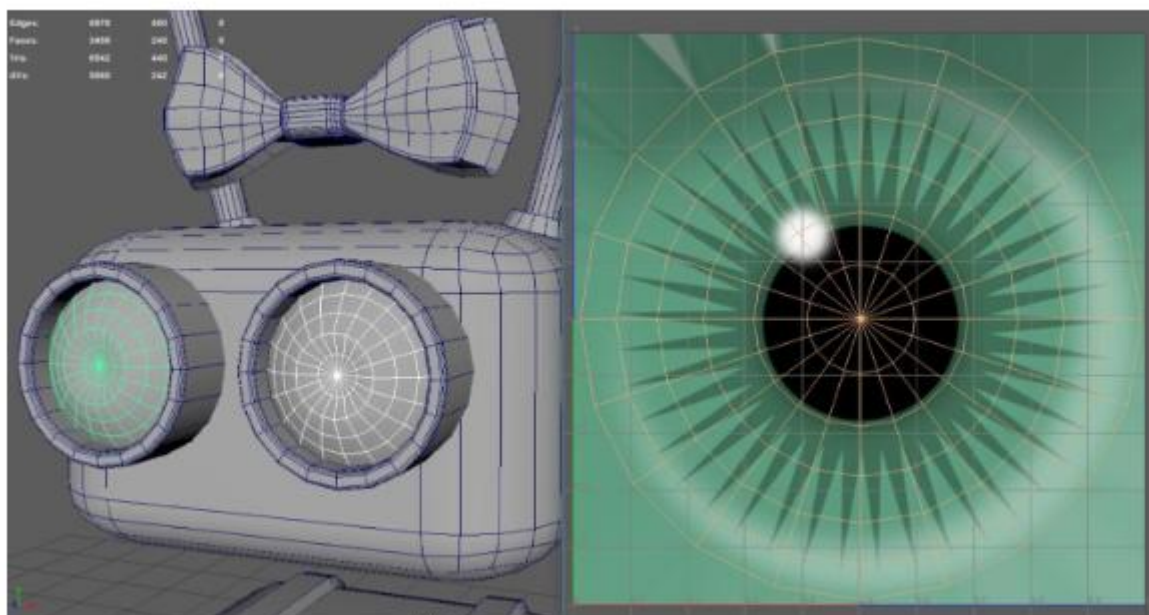
Gambar 4.8. Ini adalah bagian-bagian yang menyusun robot dan tekstur yang menyertainya

Atlas tekstur adalah gambar tekstur yang berisi data yang menjelaskan dari bahan apa bahan tersebut dibuat. Selalu lebih baik untuk membuat atlas tekstur karena secara drastis mengurangi jumlah panggilan gambar. Gambar 4.8 memperagakan avatar robot yang tersusun dari banyak bagian, telah digabungkan menjadi satu jaring, dan UV-nya dibagikan dalam satu ruang, semuanya terbuka dan siap diberi tekstur.

Ada satu area pada model ini yang ingin saya pertahankan resolusinya lebih tinggi: detail pada mata. Model itu sendiri datar; namun, saya memberinya peta tekstur dari satu

mata yang dibagi di kedua jaring datar dan melingkar. Detail pada gambar 2D datar mengecoh pemirsa hingga berpikir bahwa mungkin ada lebih banyak kedalaman daripada yang sebenarnya. Jika saya memasukkannya dalam atlas tekstur, saya perlu menambah ukuran tekstur dan membuat UV lainnya jauh lebih kecil karena detail pada pupil dan sorotan pada mata lebih penting, yang membutuhkan lebih banyak ruang UV.

Sebaliknya, UV dari jaring mata mengambil seluruh ruang UV di kuadran untuk tekstur mata. Subjaring menunjukkan semua detail yang dibutuhkan mata. Subjaring yang sama kemudian diduplikasi ke socket lainnya karena tidak diperlukan detail unik untuk membedakan antara mata. Gambar 4.9 menunjukkan area UV yang dibagi pada tekstur kecil untuk mata.



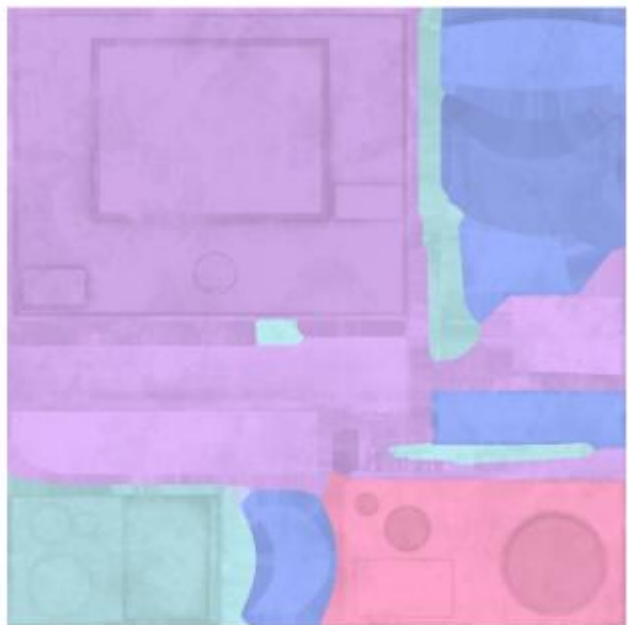
Gambar 4.9 Mata model robot memiliki UV yang sama, diduplikasi sebelum digabungkan menjadi satu jaring

Untuk gaya seni yang lebih realistis, Anda tetap perlu menjaga jumlah poli pada sisi yang lebih rendah; namun, Anda dapat menjaga kualitas model tetap tinggi dengan menggunakan shader dan rendering berbasis fisik. Model robot ini menggunakan rendering berbasis fisik (PBR) agar terlihat realistis, seperti yang diilustrasikan pada Gambar 4.10. PBR menggunakan model pencahayaan realistis dan nilai permukaan yang mewakili material nyata.

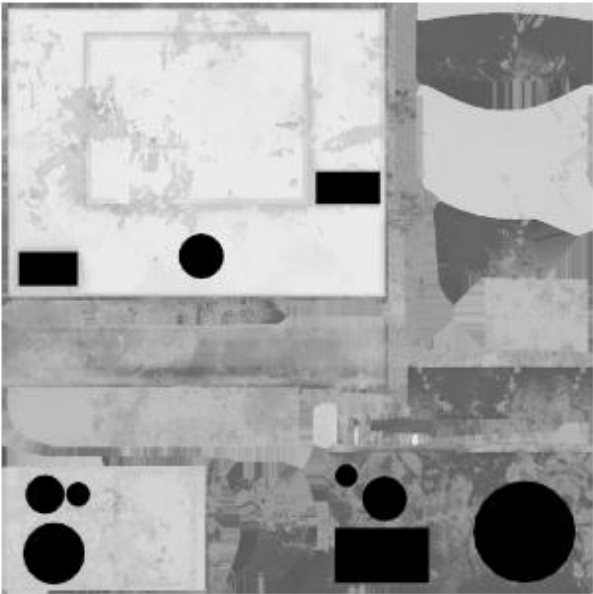


Gambar 4.10. Melihat robot dengan semua material PBR-nya

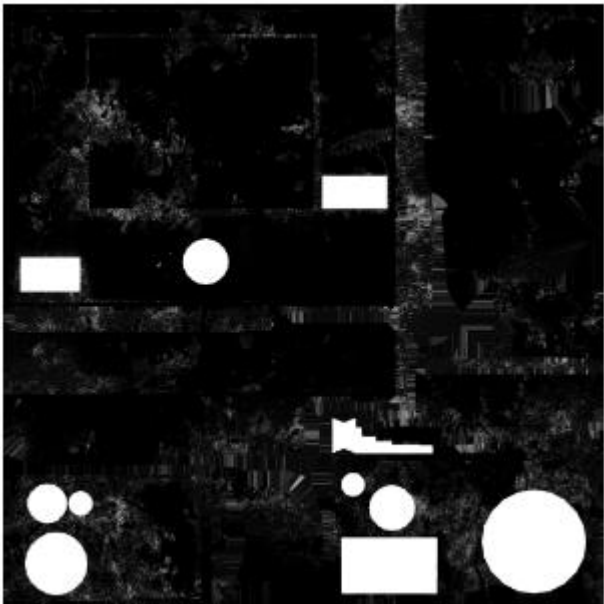
Mari kita bahas beberapa tekstur PBR yang saya gunakan pada model robot sebagai contoh. Semoga ini membantu Anda memahami cara kerja PBR pada model untuk pengalaman VR Anda. Ingat model konsol gim yang kita lihat sebelumnya dalam bab ini? Gambar 4-11 hingga 4-13 menunjukkan peta tekstur yang digunakan untuk set tersebut; perhatikan tekstur individual yang digunakan untuk material PBR-nya.



Gambar 4-11. Peta warna tempat tekstur menentukan warna yang ditampilkan pada model

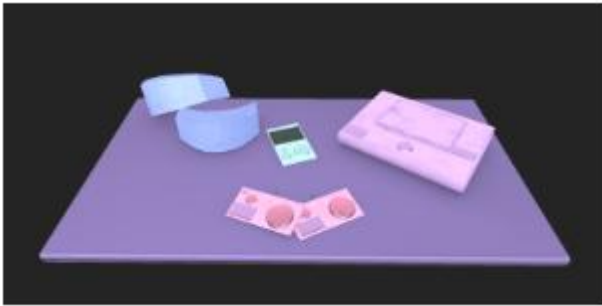


Gambar 4.12. Peta kekasaran di mana tekstur menentukan permukaan model, mulai dari halus hingga kasar

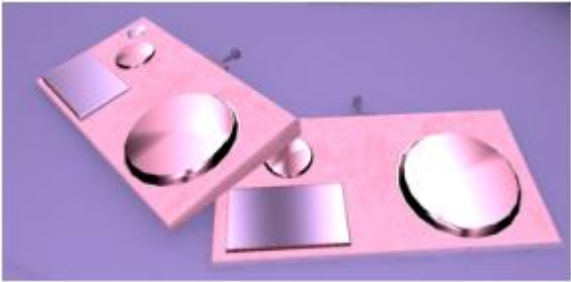


Gambar 4.13. Peta metalik di mana tekstur menentukan apakah suatu permukaan bersifat metalik

Gambar 4.14 hingga 4.17 menunjukkan tampilan akhir model 3D dalam program tempat model tersebut dilukis, Allegorithmic Substance Painter, dan menunjukkan tampilannya dalam VR dalam aplikasi VR sosial, High Fidelity.



Gambar 4.14. Tampilan sistem permainan, yang digabungkan menjadi satu jarring menggunakan satu material 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



Gambar 4.16. Sistem permainan ini memiliki lebih banyak informasi kekasaran pada bagian nonlogam, yang menunjukkan kotoran dan debu



Gambar 4.17 Berikut adalah versi finalnya yang terletak di galeri seni virtual berskala besar tempat model melayang di langit

Ada jenis peta tekstur lain seperti peta oklusi normal, bump, dan ambient. Masing-masing berperan dalam menentukan tampilan model, baik 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 cara membuat atlas tekstur, selanjutnya kita akan membahas mengapa penting untuk membuatnya saat kita memeriksa panggilan gambar.

Panggilan Gambar

Panggilan gambar 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 memiliki jumlah panggilan gambar sesedikit mungkin karena jika terlalu banyak akan menyebabkan penurunan frame rate.

Untuk mengurangi jumlah panggilan gambar yang Anda miliki, ikuti panduan berikut:

- ❖ Gabungkan semua submesh model Anda menjadi satu mesh gabungan.
- ❖ Buat atlas tekstur untuk semua UV dalam model.
- ❖ Berikan mesh Anda jumlah material sesedikit mungkin yang menggunakan semua tekstur yang dibutuhkan model.

Pikirkan salah satu pengalaman VR favorit Anda dan bayangkan semua model 3D yang membentuk adegan tersebut. Masing-masing dan setiap model tersebut berkontribusi pada jumlah panggilan gambar dengan satu atau lain cara. Mereka selalu bertambah. Jika konteks ini dialami dalam VR sosial, pertimbangkan juga berapa banyak orang yang akan mengalami rendering semua hal dalam adegan Anda.

Saat kita mendekati akhir bab ini, saya ingin menegaskan kembali bahwa penting untuk tetap menjadikan pengoptimalan sebagai prioritas tinggi dalam seluruh proses desain—dari awal model hingga tekstur yang telah selesai. Pertahankan jumlah dan ukuran tetap kecil tanpa harus mengorbankan semua yang Anda inginkan untuk konten VR Anda.

4.3 MENGGUNAKAN ALAT VR UNTUK MEMBUAT SENI 3D

Pada titik 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 kita melihat banyak pilihan 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 mengeksplor sejumlah besar konten dari program ini. Perasaan ajaib dalam menciptakan seni dalam ruang 3D di sekitar Anda berasal dari konten yang keluar seperti yang diharapkan. Ini berarti bahwa banyak geometri dibuat dengan cukup banyak loop tepi 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-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 kreator dan desainer harus membuat pilihan untuk memastikan ukuran,

jumlah, dan kualitas dapat diterima untuk pengalaman tersebut. Keseimbangan yang tepat akan selalu diperlukan, apa pun media yang digunakan untuk membuat konten ini.

Memperoleh Model 3D Versus Membuatnya dari Awal

Berhati-hatilah saat membeli model dari toko daring. Pertimbangkan berapa lama model tersebut dibuat. Apakah menurut Anda model tersebut dibuat dengan mempertimbangkan VR? Apakah Anda perlu membersihkan model dan mengoptimalkannya untuk penggunaan Anda? Apakah waktu yang mungkin Anda perlukan untuk itu lebih murah daripada waktu Anda membuatnya dari awal? Membeli model 3D dapat dilakukan dengan cepat dan mudah, tetapi dapat memengaruhi kinerja Anda di kemudian hari dan menghabiskan banyak waktu untuk memodifikasinya agar berkinerja baik. Berikut ini adalah daftar hal-hal yang perlu diperhatikan dalam daftar item dan pertanyaan apa yang harus Anda ajukan saat mengunduh model 3D dari tempat-tempat seperti Poly, Turbosquid, CGTrader, dan sebagainya (jika Anda tidak melihat informasi yang tercantum, berhati-hatilah dan persiapkan diri untuk ketidaknyamanan):

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

Selalu uji tampilan model Anda. Masukkan ke mesin pilihan Anda dan lihat sendiri dalam VR atau AR. Anda akan terkejut dengan betapa berbedanya skala yang terasa saat Anda tenggelam di dalamnya.

Ringkasan

Dalam bab ini, Anda melihat berbagai pendekatan dan proses berpikir yang perlu dipertimbangkan di berbagai area yang terlibat dalam pembuatan seni 3D. Diperlukan waktu dan latihan untuk mempelajari cara mengoptimalkan seni 3D, jadi pastikan pengoptimalan selalu menjadi prioritas tinggi selama seluruh proses desain. Anda mungkin seorang seniman yang baru dalam membuat karya untuk VR atau AR. Anda mungkin seorang pengembang yang mempelajari area tempat orang lain bekerja. Anda mungkin seorang produser yang ingin tahu tentang alur kerja para seniman. Saya senang Anda berhasil sejauh ini untuk mempelajari tentang pentingnya pengoptimalan karena ini merupakan tantangan besar dalam hal

mengembangkan aset untuk VR dan AR. Setiap orang yang mengerjakan pengalaman imersif harus tahu tentang pekerjaan menantang yang dilakukan 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 yang disebutkan sebelumnya, penting untuk membuat orang merasa nyaman dalam pengalaman Anda. Pastikan untuk memperhatikan dan menjaga frame rate tetap tinggi dengan seni yang dioptimalkan!

BAB 5

PERAN COMPUTER VISION YANG MEMUNGKINKAN AUGMENTED REALITY

Nama saya Matt Miesnieks, dan saya adalah CEO dari perusahaan rintisan bernama 6D.ai, yang merupakan spinoff dari Oxford University Active Vision Lab, tempat salah satu pendiri saya, Profesor Victor Prisacariu, mengawasi salah satu kelompok penelitian visi komputer AR terbaik di dunia. Saya telah menghabiskan 10 tahun bekerja di bidang AR, sebagai pendiri (Dekko), investor (Super Ventures), dan eksekutif (Samsung, Layar). Saya memiliki latar belakang yang luas dalam infrastruktur perangkat lunak di telepon pintar (Openwave) dan Wireline (Ascend Communications), sebagai teknisi dan wakil presiden penjualan.

Di 6D.ai, kami berpikir sedikit berbeda dari 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 persistensi adalah hal mendasar, dan Anda tidak dapat memiliki persistensi tanpa memperlakukan privasi dengan serius. Dan untuk memperlakukan privasi dengan serius, itu berarti bahwa informasi yang dapat diidentifikasi secara pribadi (PII) tidak dapat meninggalkan perangkat (kecuali secara eksplisit diizinkan oleh pengguna). Ini menciptakan masalah teknis yang jauh lebih sulit untuk dipecahkan karena itu berarti membangun dan mencari peta SLAM yang besar di perangkat, dan secara real time. Secara teknis ini cukup mudah dilakukan dengan peta dan jangkar kecil, tetapi sangat, sangat sulit dilakukan dengan peta besar. Dan ketika saya mengatakan kecil, yang saya maksud adalah setengah ruangan, dan besar berarti lebih besar dari rumah besar.

Untungnya, kami memiliki kelompok penelitian AR teratas dari Oxford Active Vision Lab di belakang 6D.ai, dan kami membangun sistem kami pada algoritma rekonstruksi dan relokalisasi 3D generasi berikutnya dan jaringan saraf, memanfaatkan beberapa penelitian yang belum dipublikasikan. Tujuan dari semua ini adalah untuk mendapatkan AR multipemain dan persisten sedekat mungkin dengan pengalaman pengguna (UX) yang "berfungsi dengan baik", di mana tidak ada yang perlu dijelaskan, dan intuisi pengguna akhir tentang bagaimana konten AR seharusnya berperilaku sudah benar. Berikut ini adalah hal-hal istimewa tentang cara 6D.ai melakukan AR:

- ✓ Kami melakukan semua pemrosesan pada 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, yang sangat meningkatkan cakupan ruang.
- ✓ Anchor dan peta tidak memiliki PII dan disimpan secara permanen di cloud kami. Setiap kali aplikasi bertenaga 6D.ai menggunakan ruang fisik tersebut, anchor akan tumbuh dan meningkatkan cakupan ruang tersebut. Hal ini meminimalkan dan akhirnya menghilangkan kebutuhan untuk melakukan pra-pemindaian suatu ruang.

- ✓ Peta tersedia untuk semua aplikasi. Setiap pengguna mendapatkan keuntungan dari setiap pengguna aplikasi 6D.ai lainnya.
- ✓ Data peta kami tidak dapat direkayasa ulang menjadi gambar visual yang dapat dibaca manusia.
- ✓ Jangkar kami sangat diuntungkan dari penyimpanan dan penggabungan cloud, tetapi tidak ada ketergantungan pada cloud agar UX dapat berfungsi. Tidak seperti sistem Google, kami dapat bekerja secara offline, atau dalam lingkungan peer-to-peer, atau lingkungan privat/aman (atau China).

Dunia ini sangat kecil. Tidak banyak orang yang dapat membangun sistem ini dengan baik.

5.1 SEJARAH SINGKAT AR

Berikut ini adalah ringkasan dari para pemain kunci yang menghadirkan AR ke kualitas konsumen:

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

Satu aspek yang menarik dan kurang dihargai dari cara membangun sistem AR berkualitas tinggi adalah bahwa hanya ada segelintir orang di dunia yang dapat membangunnya. Karier yang saling terkait dari para insinyur ini telah menghasilkan sistem terbaik yang menyatu pada odometri inersia visual monokuler (VIO) sebagai "solusi" untuk pelacakan seluler. Tidak ada pendekatan lain yang memberikan UX (saat ini). VIO pertama kali diimplementasikan di pemasok militer/industri Intersense yang berbasis di Boston, Massachusetts pada awal tahun 2000-an. Salah satu penemu bersama, Leonid Naimark adalah kepala ilmuwan di perusahaan rintisan 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 dikembangkan menjadi sistem VIO untuk Tango. Apple kemudian membeli FlyBy, dan basis kode yang sama menjadi inti dari ARKit VIO. CTO FlyBy, Chris Broaddus, melanjutkan untuk membangun pelacak untuk Daqri, yang sekarang berada di perusahaan robotika otonom, dengan mantan kepala ilmuwan Zoox, Gabe Sibley.

Gabe melakukan pekerjaan pascadoktoralnya di Oxford (bersama dengan salah seorang pendiri saya di 6D.ai, yang saat ini memimpin Active Vision Lab). Sistem SLAM seluler pertama (PTAM) 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 pergi untuk membangun sistem otonomi di Tesla. Georg memperoleh gelar doktornya di Cambridge, tempat rekannya Gerhard Reitmayr melanjutkan ke Vuforia untuk mengerjakan pengembangan sistem SLAM dan VIO Vuforia. Pengembangan Vuforia dipimpin oleh Daniel Wagner, yang kemudian mengambil alih dari Chris Broaddus (mantan FlyBy) sebagai kepala ilmuwan di Daqri.

Manajer teknik Vuforia, Eitan Pilipski, sekarang memimpin rekayasa perangkat lunak AR di Snap, bekerja sama dengan Qi Pan, yang belajar di Cambridge bersama Gerhard dan Georg, dan kemudian pindah 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 utama tim peneliti di Oxford, Cambridge (misalnya, David Molyneaux) dan Imperial College (laboratorium Profesor Andy Davison, tempat Richard Newcombe, Hauke Strasdat, dan lainnya belajar) mengembangkan lebih lanjut D-SLAM dan memperluas sistem pelacakan Kinect, dan sekarang memimpin tim pelacakan di Oculus dan Magic Leap. Metaio juga merupakan inovator utama awal seputar SLAM (dengan memanfaatkan keahlian dari TU Munich, tempat Pierre Georgel belajar), banyak teknisi sekarang bekerja di Apple, tetapi pimpinan R&D mereka, Selim Benhimane, belajar bersama Pierre dan kemudian mengembangkan SLAM untuk Intel RealSense, dan sekarang bekerja di Apple.

Menariknya, saya tidak mengetahui adanya perusahaan rintisan AR saat ini yang bergerak di bidang pelacakan AR yang dipimpin oleh teknisi berbakat dari kelompok kecil ini. Para pendiri yang berlatar belakang Robotika atau jenis visi komputer lainnya belum mampu menunjukkan sistem yang bekerja dengan baik di berbagai lingkungan.

Bagaimana dan Mengapa Memilih Platform AR

Ada banyak platform yang dapat dipilih dalam AR, mulai dari AR seluler hingga PCAR. Berikut adalah beberapa pertimbangan teknis yang perlu diingat saat mulai mengembangkan AR.

Saya Pengembang, Platform Apa yang Harus Saya Gunakan dan Mengapa?

Anda dapat mulai mengembangkan ide AR Anda di ponsel mana pun yang memiliki akses ke ARKit. Aplikasi ini berfungsi dan Anda mungkin sudah memiliki ponsel yang mendukungnya. Pelajari perbedaan besar dalam mendesain dan mengembangkan aplikasi yang berjalan di dunia nyata, tempat Anda tidak mengendalikan pemandangan, dibandingkan dengan aplikasi ponsel pintar dan VR, tempat Anda mengendalikan setiap piksel.

Kemudian, beralihlah ke platform seperti Magic Leap, 6D.ai, atau HoloLens yang dapat memetakan dunia secara spasial. Sekarang pelajari apa yang terjadi saat konten Anda dapat berinteraksi dengan struktur 3D dari adegan yang tidak terkontrol. Beralih dari satu ke yang lain adalah kurva pembelajaran yang sangat curam. Bahkan, lebih curam daripada dari web ke seluler atau dari seluler ke VR. Anda perlu memikirkan ulang sepenuhnya 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 dalam skala yang jauh lebih besar. Saya telah melihat contoh dari hampir setiap jenis aplikasi AR selama bertahun-tahun, dan dengan senang hati memberikan umpan balik dan dukungan. Hubungi saja.

Saya akan mendorong pengembang untuk tidak takut membuat aplikasi baru. Aplikasi kentut adalah yang pertama kali populer di ponsel pintar—juga sangat sulit untuk menemukan kasus penggunaan yang memberikan utilitas melalui AR pada perangkat keras faktor bentuk tembus pandang genggam.

Performa adalah Statistik

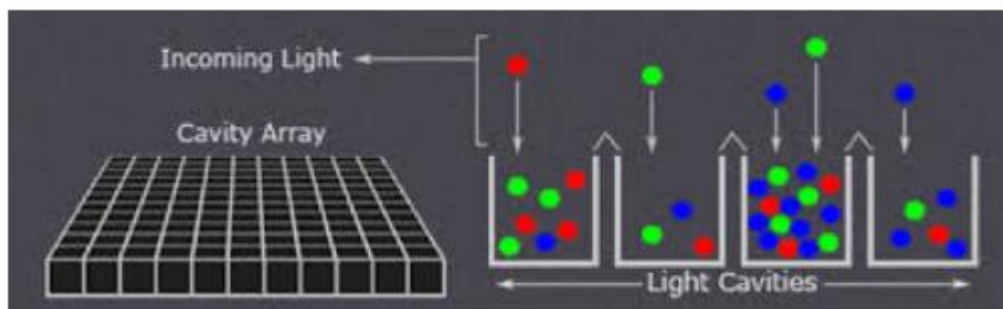
Saat pertama kali bekerja dengan AR atau sebagian besar sistem visi komputer, hal itu bisa membuat frustrasi karena terkadang sistem itu akan berfungsi dengan baik di satu tempat, tetapi di tempat lain akan berfungsi dengan sangat buruk. Sistem AR tidak pernah "berfungsi" atau "tidak berfungsi." Selalu menjadi pertanyaan apakah segala sesuatunya berfungsi dengan cukup baik dalam berbagai situasi. Pada akhirnya, menjadi "lebih baik" adalah masalah mendorong statistik lebih jauh agar menguntungkan Anda.

Karena alasan ini, jangan pernah percaya demonstrasi aplikasi AR, terutama jika sudah terbukti luar biasa di YouTube. Ada kesenjangan besar antara sesuatu yang bekerja dengan sangat baik di lingkungan yang terkontrol atau sedikit dipentaskan dan sesuatu yang hampir tidak berfungsi sama sekali untuk penggunaan biasa. Situasi ini tidak terjadi pada demonstrasi aplikasi ponsel pintar atau VR.

Mari kita rangkum ini:

- Selalu demonstrasikan atau uji sistem di dunia nyata. Ada kesenjangan besar antara adegan yang 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 jitter
 - Tidak ada drift
 - Daya rendah
 - Biaya BOM rendah
 - Ratusan meter jangkauan dengan drift <1% (sebelum penutupan loop)
 - Penutupan loop instan
 - Penutupan loop dari jangkauan atau sudut lebar
 - Adegan dengan fitur rendah (misalnya, langit, dinding putih)
 - Adegan dengan pencahayaan bervariasi/cahaya redup
 - Adegan berulang atau reflektif

Berikut adalah deskripsi teknis khusus tentang mengapa statistik akhirnya menentukan seberapa baik sistem bekerja. Gambar 5-1 menggambarkan kisi yang mewakili sensor gambar digital di kamera Anda. Setiap kotak adalah piksel. Agar pelacakan stabil, setiap piksel harus cocok dengan titik yang sesuai di dunia nyata (dengan asumsi bahwa perangkat benar-benar diam). Namun, gambar kedua menunjukkan bahwa foton tidak terlalu akomodatif, dan berbagai intensitas cahaya jatuh ke mana pun yang mereka inginkan, dan setiap piksel hanyalah total foton yang mengenainya. 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 titik di dunia nyata. Sejauh menyangkut sistem pelacakan visual, Anda telah bergerak!



Gambar 5.1 Segala hal yang berkaitan dengan kinerja visi komputer adalah masalah statistik; dunia nyata bukanlah biner

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

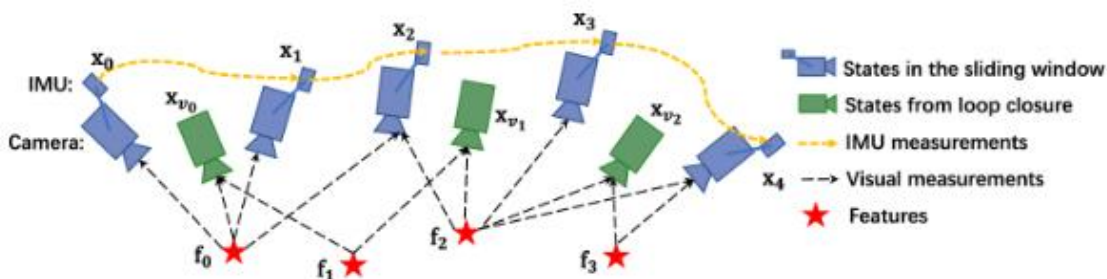
Jika Anda seorang pengembang, Anda harus selalu menguji aplikasi Anda dalam berbagai pemandangan dan kondisi pencahayaan. Jika Anda berpikir bahwa menangani fragmentasi Android itu buruk, tunggu hingga Anda mencoba menguji segala hal yang mungkin terjadi di dunia nyata.

5.2 MENINGTEGRASIKAN PERANGKAT KERAS DAN PERANGKAT LUNAK

Menariknya, VIO tidak terlalu sulit untuk dijalankan; ada sejumlah algoritme yang diterbitkan dan cukup banyak implementasi yang ada. Namun, sangat sulit untuk membuatnya berfungsi dengan baik. Yang saya maksud adalah sistem inersia dan optik bertemu hampir seketika pada peta stereoskopik, dan skala metrik dapat ditentukan dengan tingkat akurasi satu digit yang rendah. Implementasi yang kami buat di Dekko, misalnya, mengharuskan pengguna membuat gerakan tertentu pada awalnya dan kemudian menggerakkan ponsel maju mundur selama sekitar 30 detik sebelum bertemu. Untuk membangun sistem pelacakan inersia yang hebat diperlukan teknisi yang berpengalaman. Sayangnya, hanya ada sekitar 20 teknisi di Bumi dengan keterampilan dan pengalaman yang diperlukan, dan sebagian besar dari mereka bekerja membangun sistem pelacakan rudal jelajah, atau sistem navigasi penjelajah Mars, atau aplikasi seluler nonkonsumen lainnya.

Seperti yang diilustrasikan pada Gambar 5.2, semuanya masih bergantung pada perangkat keras dan perangkat lunak yang bekerja secara bersamaan untuk mengurangi kesalahan. 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 dengan akhir. Ini penting untuk mengkorelasikan kedua sistem, dan hingga saat ini tidak mungkin dilakukan karena OEM perangkat keras tidak melihat alasan untuk berinvestasi dalam hal ini.



Gambar 5.2. AR memerlukan integrasi yang erat antara perangkat lunak dan perangkat keras, yang memperlambat solusi pada ponsel

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

Karena ketergantungan yang erat pada perangkat keras dan perangkat lunak ini, hampir mustahil 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 guna mendukung spesifikasi perangkat keras Tango. Microsoft, Magic Leap, dan yang lainnya membangun perangkat keras mereka sendiri, dan itulah sebabnya Apple sangat sukses dengan ARKit, karena 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

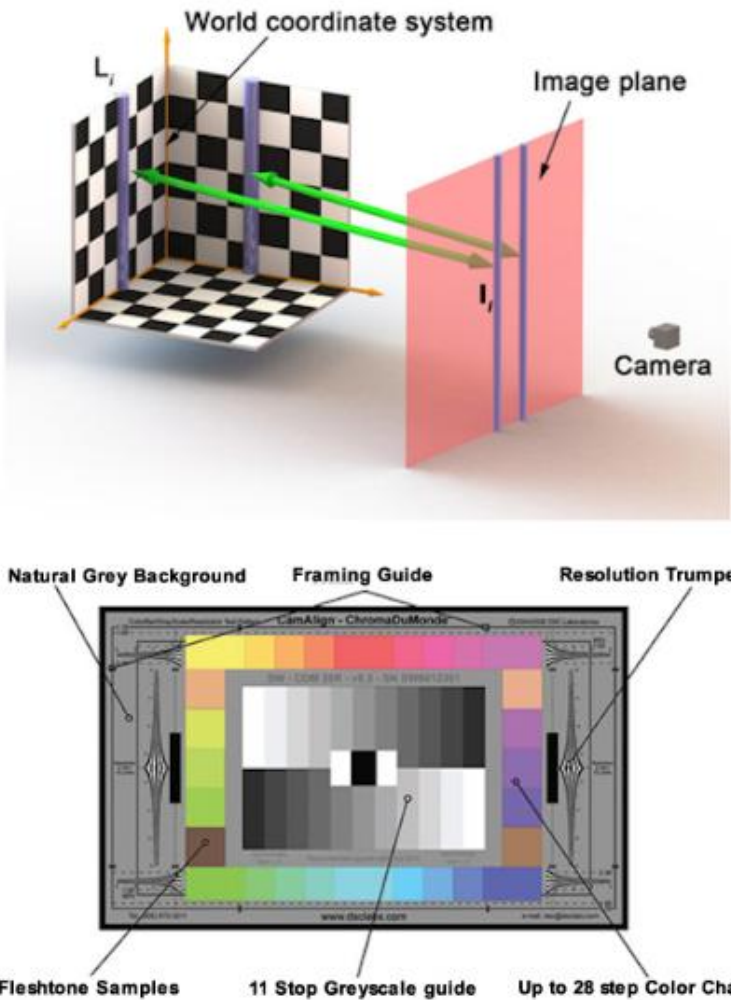
Ini menggunakan model lubang jarum kamera untuk mengoreksi Bidang Pandang lensa dan hal-hal seperti efek laras lensa—pada dasarnya, semua lengkungan gambar karena bentuk lensa. Sebagian besar pengembang perangkat lunak dapat melakukan langkah ini tanpa masukan OEM dengan menggunakan papan catur dan spesifikasi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

Kalibrasi fotometrik

Ini jauh lebih rumit dan biasanya memerlukan keterlibatan OEM karena masuk ke spesifikasi sensor gambar itu sendiri, pelapis apa pun pada lensa internal, dan sebagainya. Kalibrasi ini berkaitan dengan pemetaan warna dan intensitas. Misalnya, kamera yang terpasang pada teleskop yang memotret bintang-bintang jauh perlu mengetahui apakah perubahan kecil dalam Intensitas cahaya pada piksel pada sensor memang bintang, atau hanya

aberasi pada sensor atau lensa. Hasil kalibrasi untuk pelacak AR ini adalah kepastian yang jauh lebih tinggi bahwa piksel pada sensor cocok dengan titik di dunia nyata, dan dengan demikian pelacakan optik lebih kuat dengan lebih sedikit kesalahan.

Pada Gambar 5-3, gambar berbagai foton RGB yang jatuh ke dalam keranjang 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 merata-ratakan intensitas di semua foton yang mengenainya. Perubahan kecil pada gerakan pengguna, bayangan dalam pemandangan, atau lampu fluoresensi yang berkedip-kedip akan mengubah piksel mana yang paling mewakili titik di 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 di dunia nyata

Kalibrasi Inersia

Saat memikirkan IMU (kombinasi akselerometer dan giroskop di perangkat Anda), penting untuk diingat bahwa IMU mengukur percepatan, bukan jarak atau kecepatan. Kesalahan dalam pembacaan IMU terakumulasi dari waktu ke waktu, dengan sangat cepat! Tujuan kalibrasi dan pemodelan adalah untuk memastikan pengukuran jarak (diintegrasikan

ganda dari percepatan) cukup akurat untuk X sepersekian detik. Idealnya, ini adalah periode yang cukup lama untuk mencakup saat kamera kehilangan pelacakan selama beberapa bingkai saat pengguna menutupi lensa atau sesuatu yang lain terjadi dalam pemandangan.

Mengukur jarak menggunakan IMU disebut perhitungan mati. Pada dasarnya ini adalah tebakan, tetapi tebakan dibuat akurat dengan memodelkan bagaimana IMU berperilaku, menemukan semua cara IMU mengakumulasi kesalahan, lalu menulis filter untuk mengurangi kesalahan tersebut. Bayangkan jika Anda diminta untuk melangkah 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 setiap langkah dan belajar memperhitungkan kaki mana yang Anda gunakan untuk melangkah, penutup lantai, sepatu yang Anda kenakan, seberapa cepat Anda bergerak, seberapa lelah Anda, dan seterusnya, tebakan Anda 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 kali, dan keluaran dari IMU ditangkap dan difilter hingga keluaran dari IMU secara akurat cocok dengan gerakan dasar dari lengan robot. Google dan Microsoft bahkan mengirim perangkat mereka ke gravitasi mikro di Stasiun Luar Angkasa Internasional, atau "penerbangan gravitasi nol," untuk menghilangkan kesalahan tambahan.

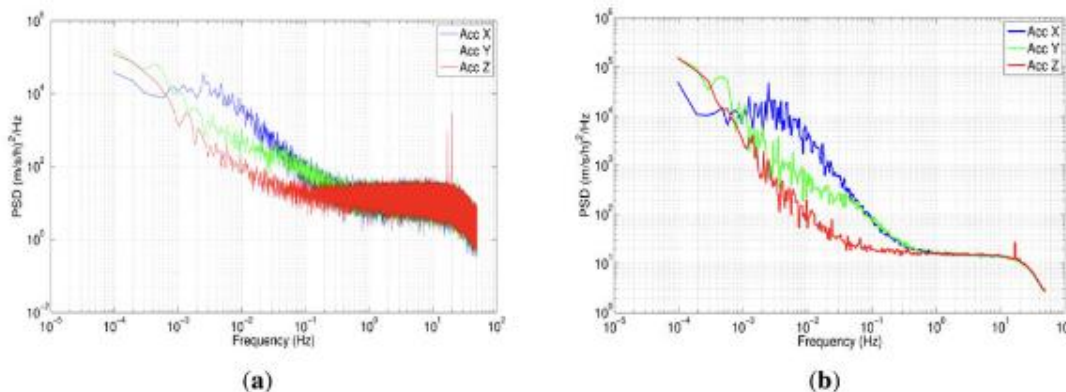


Gambar 5.4. Kalibrasi inersia bahkan lebih menantang dan tidak ada kasus penggunaan yang membutuhkannya sebelumnya (untuk perangkat keras konsumen)

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

- ✓ *Bias tetap*: Pengukuran percepatan bukan nol saat percepatan nol diintegrasikan
- ✓ *Kesalahan faktor skala*: Penyimpangan keluaran aktual dari model keluaran matematis (biasanya keluaran nonlinier)
- ✓ *Kopling silang*: Percepatan dalam arah ortogonal terhadap arah pengukuran sensor yang diteruskan ke pengukuran sensor (ketidaktepatan produksi, sumbu sensor non-ortogonal)

- ✓ *Kesalahan vibro-pendulous*: Getaran dalam fase dengan perpindahan pendulum (pikirkan seorang anak di ayunan)
- ✓ *Kesalahan jam*: Periode integrasi diukur secara tidak tepat



Gambar 5.5. Ini hanyalah beberapa kesalahan yang harus diidentifikasi dari jejak seperti garis RGB dalam grafik

Proses ini juga sulit bagi OEM untuk harus melalui proses ini untuk semua perangkat dalam portofolionya, dan bahkan saat itu, 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.

Masa Depan Pelacakan

Jadi, jika VIO adalah yang berfungsi saat ini, apa yang akan terjadi selanjutnya dan apakah itu akan membuat ARKit menjadi tidak berguna? Anehnya, VIO akan tetap menjadi cara terbaik untuk melacak dalam jarak beberapa ratus meter (untuk waktu yang lebih lama dari itu, sistem perlu melakukan relokasi menggunakan kombinasi GPS yang menyatu ke dalam sistem ditambah semacam pengenalan landmark). Alasannya adalah bahwa meskipun sistem optik lainnya menjadi seakurat VIO, sistem tersebut tetap memerlukan daya yang lebih besar (unit pemrosesan grafis [GPU] atau kamera), yang sangat penting dalam HMD. VIO monokuler adalah solusi yang paling akurat, berdaya paling rendah, dan berbiaya paling rendah.

Pembelajaran mendalam benar-benar berdampak pada komunitas penelitian untuk pelacakan. Sejauh ini, sistem berbasis pembelajaran mendalam memiliki sekitar 10% kesalahan, sedangkan sistem VIO teratas hanya sebagian kecil dari satu persen, tetapi mereka mengejar dan akan sangat membantu relokasi di luar ruangan. Kamera kedalaman (Gambar 5.6) dapat membantu sistem VIO dalam beberapa cara. Pengukuran akurat dari ground truth dan skala metrik serta pelacakan tepi untuk adegan dengan fitur rendah merupakan manfaat terbesar. Akan tetapi, kamera ini sangat boros daya, jadi masuk akal untuk menjalankannya hanya pada frame rate yang sangat rendah dan menggunakan VIO di antara frame.



Gambar 5.6. Masa depan pelacakan

Kamera ini juga tidak berfungsi di luar ruangan karena hamburan inframerah latar belakang dari sinar matahari menghilangkan inframerah dari kamera kedalaman. Jangkauannya juga bergantung pada konsumsi daya, yang berarti bahwa pada ponsel, jangkauannya sangat pendek (beberapa meter). Kamera ini juga mahal dalam hal biaya BOM, jadi OEM akan menghindarinya untuk ponsel bervolume tinggi. Lensa stereo RGB atau fisheye membantu melihat pemandangan yang lebih besar dan dengan demikian berpotensi memiliki lebih banyak fitur optik (misalnya, lensa biasa mungkin melihat dinding putih, tetapi fisheye juga dapat melihat langit-langit berpola dan karpet dalam bingkai — Magic Leap dan Hololens menggunakan pendekatan ini) dan mungkin mendapatkan informasi kedalaman dengan biaya komputasi yang lebih rendah daripada VIO, meskipun VIO melakukannya dengan sama akuratnya untuk biaya BOM dan daya yang lebih rendah. Karena kamera stereo pada ponsel

atau bahkan HMD berdekatan, jangkauan akuratnya sangat terbatas untuk kalkulasi kedalaman (kamera yang berjarak beberapa sentimeter dapat akurat untuk kedalaman hingga beberapa meter).

Hal paling menarik yang akan datang adalah dukungan untuk pelacakan di area yang jauh lebih luas, terutama di luar ruangan sejauh beberapa kilometer. Pada titik ini, hampir tidak ada perbedaan antara pelacakan untuk AR dan pelacakan 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, diperlukan layanan yang didukung cloud; Google baru-baru ini mengumumkan Layanan Pemosisian Visual Tango karena alasan ini. Kita akan melihat lebih banyak layanan semacam ini dalam waktu dekat. Itulah juga alasan mengapa semua orang sangat peduli dengan peta 3D saat ini.

Masa Depan Visi Komputer AR

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

Rekonstruksi 3D (pemetaan spasial dalam istilah Hololens atau persepsi kedalaman dalam istilah Tango) adalah sistem yang mampu mengetahui bentuk atau struktur objek nyata dalam suatu pemandangan, 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 mengira ini berarti AR sekarang menjadi realitas "campuran". 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 pemandangan (saat ini menggunakan kamera kedalaman) lalu mengubahnya menjadi jaring dan memasukkan jaring "tak terlihat" ke Unity (bersama dengan koordinat dunia nyata), lalu menempatkan jaring dunia nyata tepat 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 menghostingnya di awan dan memungkinkan banyak pengguna berbagi (dan memperluas) model. ARKit melakukan versi 2D saat ini dengan mendeteksi bidang 2D. Ini adalah minimum yang dibutuhkan. Tanpa bidang dasar, konten Unity secara harfiah tidak akan memiliki dasar untuk berdiri dan akan melayang.



Gambar 5.7. Rekonstruksi 3D skala besar

Gambar 5.8 menunjukkan upaya awal untuk menunjukkan oklusi dengan membangun jaring menggunakan iPad 2. Ini adalah aplikasi pertama yang menunjukkan interaksi fisik antara konten virtual dan dunia nyata pada perangkat keras seluler komoditas.

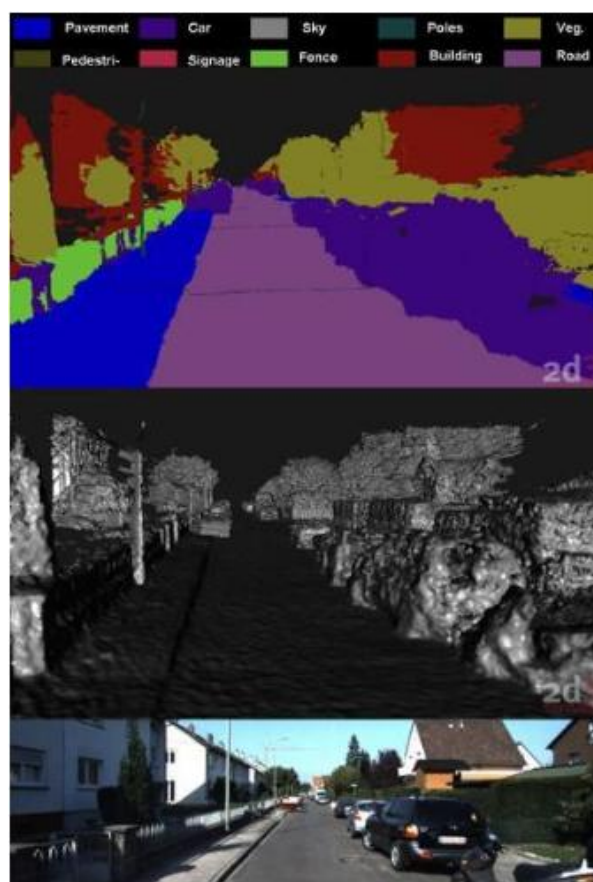


Gambar 5.8. Aplikasi ini dibuat oleh perusahaan rintisan penulis sebelumnya, Dekko

Gambar 5.9 menyajikan contoh segmentasi semantik 3D dari sebuah pemandangan. Gambar sumber ada di bagian bawah. Di atasnya adalah model 3D (mungkin dibuat dari kamera stereo, atau LIDAR), dan di bagian atas adalah segmentasi melalui pembelajaran mendalam; sekarang kita dapat membedakan trotoar dari jalan. Ini juga berguna untuk Pokémon Go sehingga Pokémon tidak ditempatkan di tengah jalan yang ramai.

Kemudian, kita perlu mencari tahu cara untuk menskalakan semua teknologi menajukubkan ini untuk mendukung banyak pengguna secara bersamaan dalam waktu nyata. Ini adalah Massively Multiplayer Online Roleplaying Game (MMORG, misalnya, World of Warcraft, tetapi untuk dunia nyata) terbaik. Berikut adalah beberapa tantangan lain yang perlu kita tangani dan selesaikan:

- Semua hal di tumpukan
 - Rendering (koherensi, kinerja)
 - Input
 - Optik
 - GUI dan aplikasi
 - Faktor sosial



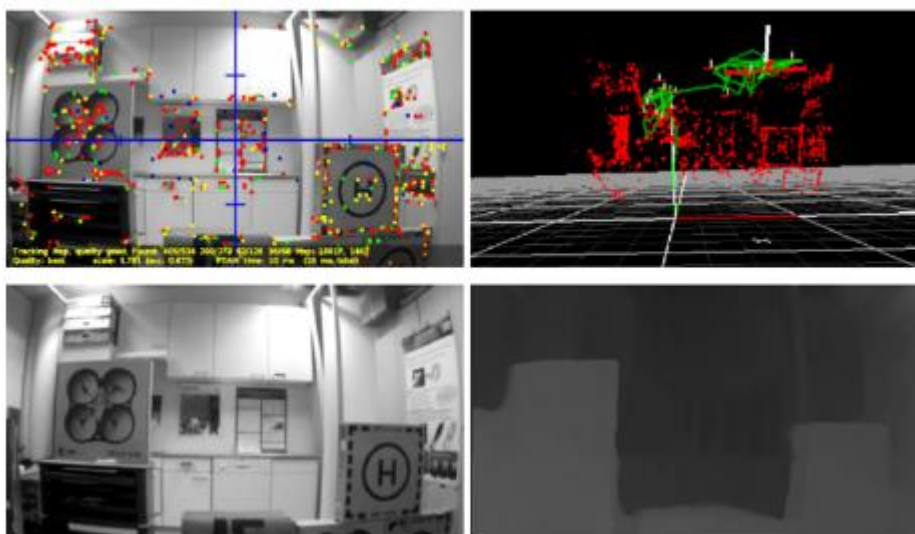
Gambar 5.9 Rekonstruksi 3D skala besar

5.3 PEMETAAN

Pemetaan adalah "M" dalam SLAM. Pemetaan mengacu pada struktur data yang disimpan perangkat dalam memori yang berisi informasi tentang pemandangan 3D yang dapat dilokalkan oleh pelacak (istilah umum untuk sistem VIO). Melokalkan 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 hingga Anda memastikan di mana Anda berada di peta adalah proses melokalkan diri Anda 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 dalam pemandangan (misalnya, sudut meja). Biasanya peta juga berisi sejumlah besar metadata tambahan di sana, seperti seberapa "andal" titik itu, diukur dari berapa banyak bingkai fitur itu yang terdeteksi dalam koordinat yang sama baru-baru ini (misalnya, titik hitam pada anjing saya tidak akan ditandai andal karena anjing itu bergerak-gerak). Beberapa peta menyertakan "bingkai utama", yang hanya berupa bingkai video tunggal (pada dasarnya foto) yang disimpan di peta setiap beberapa detik dan digunakan untuk membantu pelacak mencocokkan dunia dengan peta. Peta lain menggunakan titik-awan padat, yang lebih andal tetapi membutuhkan lebih banyak GPU dan memori. ARCore dan ARKit keduanya menggunakan peta renggang (tanpa bingkai utama, menurut saya).

Peta renggang 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). Bagian kiri bawah adalah gambar sumber, dan bagian kanan bawah adalah peta intensitas, yang dapat digunakan untuk jenis sistem SLAM yang berbeda (semi-langsung — yang sangat bagus, tetapi belum ada dalam sistem SLAM produksi seperti ARCore atau ARKit).



Gambar 5.10. Contoh apa yang dilihat sistem AR, ditumpangkan pada gambar yang dapat dibaca manusia

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

Ini tidak pernah menjadi masalah karena pelacak sangat buruk sehingga mereka akan menjauh tanpa dapat digunakan sebelum peta menjadi terlalu besar untuk dikelola. Hal itu tidak berlaku lagi, dan mengelola peta adalah tempat sebagian besar pekerjaan menarik di SLAM berlangsung (bersama dengan pembelajaran mendalam dan jaringan saraf konvolusional). ARKit menggunakan "jendela geser" untuk peta, yang berarti bahwa peta tersebut hanya menyimpan sejumlah variabel dari masa lalu terkini (waktu dan jarak) di peta, dan membuang apa pun yang lama. Asumsinya adalah bahwa Anda tidak akan pernah perlu melakukan relokasi terhadap pemandangan dari beberapa waktu lalu. ARCore mengelola peta yang lebih besar, yang berarti bahwa sistem tersebut seharusnya lebih andal.

Jadi, hasilnya adalah bahwa dengan ARCore, bahkan jika Anda kehilangan pelacakan, sistem tersebut akan pulih dengan 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 luas daripada yang sebenarnya. Saya melihat konsep ini pertama kali di Holo- lens, yang, seperti biasa, lebih maju satu tahun atau lebih dari yang lain. Biasanya, sistem mengelola peta sepenuhnya tanpa terlihat oleh pengguna atau pengembang aplikasi. Anchor memungkinkan pengembang untuk memerintahkan sistem untuk "mengingat bagian peta ini di sekitar sini, jangan membuangnya." Ukuran fisik anchor sekitar satu meter persegi (itu sedikit tebakan dari saya; mungkin bervariasi tergantung pada berapa banyak fitur optik yang dapat dilihat sistem). Itu cukup bagi sistem untuk melakukan relokasi saat lokasi fisik ini dikunjungi kembali oleh pengguna). Pengembang biasanya menjatuhkan anchor setiap kali konten ditempatkan di lokasi fisik. Ini berarti bahwa jika pengguna kemudian pergi, sebelum anchor, peta di sekitar lokasi fisik tempat konten seharusnya berada akan dibuang dan konten akan hilang. Dengan anchor, konten selalu berada di tempat yang seharusnya, dengan dampak UX terburuk adalah kemungkinan gangguan kecil pada konten saat sistem melakukan relokasi dan melompat untuk mengoreksi penyimpangan yang terakumulasi (jika ada).

Tujuan peta adalah untuk membantu pelacak dalam dua cara. Pertama, saat saya menggerakkan ponsel maju mundur, peta dibuat dari gerakan awal, dan saat kembali, fitur yang terdeteksi secara real time dapat dibandingkan dengan fitur yang tersimpan di peta. Ini membantu membuat pelacakan lebih stabil dengan hanya menggunakan fitur yang paling andal dari tampilan pemandangan saat ini dan sebelumnya dalam perhitungan pose.

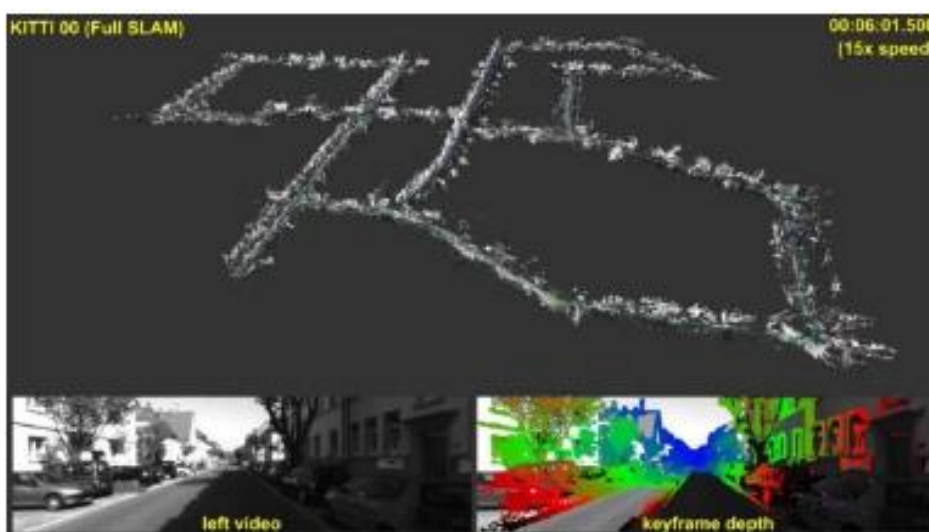
Cara kedua Peta membantu adalah dengan melokalisasi (atau memulihkan) pelacakan. Akan tiba saatnya ketika Anda menutupi kamera, menjatuhkan ponsel, bergerak terlalu cepat,

atau sesuatu yang acak terjadi, dan saat kamera melihat pemandangan berikutnya, pemandangan tersebut tidak sesuai dengan apa yang menurut pembaruan peta terakhir seharusnya dilihatnya. Kamera ditutup matanya dan dijatuhkan di tempat baru. Ini adalah definisi dari "Saya kehilangan pelacakan," yang telah 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:

- Mengatur ulang semua sistem koordinat dan memulai lagi! Inilah yang dilakukan sistem odometri murni (tanpa peta sama sekali). Yang Anda alami adalah bahwa semua konten Anda melompat ke posisi baru dan tetap di sana. Ini bukan UX yang baik.
- Sistem dapat mengambil serangkaian fitur 3D yang dilihatnya saat ini dan mencari melalui 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 gangguan pada konten virtual Anda saat pelacakan hilang, tetapi kembali ke tempatnya saat dipulihkan). Ada dua masalah di sini. Pertama, saat peta bertambah besar, proses pencarian ini menjadi sangat memakan waktu dan prosesor, dan semakin lama proses ini berlangsung, semakin besar kemungkinan pengguna akan berpindah 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 akan kesulitan dan sistem perlu mengatur ulang dan memulai lagi!

Setiap garis pada gambar yang ditunjukkan pada Gambar 511 adalah jalan dalam peta SLAM skala besar ini. Membuat perangkat seluler melakukan AR di mana saja dan di mana saja di dunia adalah masalah pemetaan SLAM yang besar. Ingatlah bahwa ini adalah peta dan struktur data yang dapat dibaca mesin; peta tersebut bukan peta bergaya tampilan jalan 3D yang bagus dan nyaman digunakan manusia (yang juga diperlukan!).



Gambar 5.11 Pemetaan SLAM skala besar merupakan tantangan bagi AR berbasis ponsel

Ingat juga bahwa dalam diskusi kita ketika saya merujuk pada peta "besar", untuk AR seluler, itu kira-kira 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 benar-benar baru.

Relokasi yang kuat terhadap peta yang besar adalah masalah yang sangat, sangat, sangat sulit, dan menurut pendapat saya, hingga tulisan ini dibuat, belum ada yang memecahkan masalah tersebut ke tingkat UX konsumen. Siapa pun yang mengklaim menawarkan konten multipemain atau AR persisten akan memiliki UX yang sangat terbatas oleh kemampuan ponsel kedua (misalnya, Pemain 2) untuk melakukan relokasi dari cold-start ke peta yang dibuat oleh Pemain 1 atau diunduh dari cloud. Anda akan menemukan bahwa Pemain 2 harus berdiri cukup dekat dengan Pemain 1 dan memegang ponsel mereka dengan cara yang hampir sama. Ini mengganggu bagi pengguna.

Mereka hanya ingin duduk di sofa di seberang Anda dan menyalakan ponsel 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 membuat kode posisi awal yang jauh untuk Pemain 2, dan seterusnya. Secara teknis, solusi tersebut dapat berfungsi, tetapi Anda tetap perlu menjelaskan apa yang harus dilakukan kepada pengguna, dan UX Anda bisa saja tidak berhasil. Tidak ada solusi ajaib yang "berfungsi begitu saja" yang memungkinkan Anda melakukan relokalisasi (misalnya, bergabung dengan peta orang lain) seperti ARKit dan ARCore yang membuat pelacakan VIO "berfungsi begitu saja."

5.4 CARA KERJA AR MULTIPEMAIN

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 umum dan mengetahui koordinat satu sama lain di setiap bingkai video. Sistem koordinat dapat berupa sistem dunia (misalnya, lintang dan bujur) atau mereka mungkin hanya setuju untuk menggunakan koordinat dari perangkat pertama untuk memulai. Ingatlah bahwa setiap perangkat ketika mulai biasanya hanya mengatakan, "Di mana pun saya berada saat ini adalah koordinat (0,0,0) saya," dan melacak pergerakan dari sana. (0,0,0) saya secara fisik berada di tempat yang berbeda dengan (0,0,0) Anda. Untuk mengubah diri saya menjadi koordinat Anda, saya perlu melokalisasi ulang diri saya ke peta SLAM Anda dan mendapatkan pose saya di koordinat Anda, lalu menyesuaikan peta saya sebagaimana mestinya. Peta SLAM adalah semua data tersimpan yang memungkinkan saya melacak di mana saya berada.
2. Kita kemudian perlu memastikan untuk setiap frame bahwa masing-masing dari kita tahu di mana yang lain berada. Setiap perangkat memiliki pelacaknya sendiri yang terus-menerus memperbarui pose setiap frame. Jadi, untuk multipemain, kita perlu

menyiarkan pose itu ke semua pemain lain dalam permainan. Ini memerlukan koneksi jaringan dari beberapa jenis, baik peer-to-peer, atau melalui layanan cloud. Sering kali juga akan ada beberapa aspek prediksi pose dan penghalusan yang terjadi untuk memperhitungkan gangguan jaringan kecil apa pun.

3. Kami berharap bahwa pemahaman 3D tentang dunia yang dimiliki setiap perangkat dapat dibagikan dengan perangkat lain (ini tidak wajib, meskipun UX akan sangat terpengaruh tanpanya). Ini berarti mengalirkan beberapa informasi 3D mesh dan 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 memanfaatkan data yang telah ditangkap itu, dan data tersebut seharusnya diperbarui antarperangkat saat permainan berlangsung.
4. Terakhir, ada semua hal "normal" yang diperlukan untuk aplikasi multipengguna online waktu nyata. Ini termasuk mengelola izin pengguna, status waktu nyata setiap pengguna (misalnya, jika saya mengetuk "tembak" dalam permainan, semua aplikasi pengguna lain perlu diperbarui bahwa saya telah "tembak"), dan mengelola semua berbagai aset bersama. Fitur teknis ini persis sama untuk aplikasi AR dan non-AR. Perbedaan utamanya adalah, hingga saat ini, fitur-fitur tersebut hanya dibuat untuk game, sedangkan AR akan membutuhkannya untuk setiap jenis aplikasi. Untungnya, semua fitur ini telah dibuat berkali-kali untuk game MMO daring dan seluler, dan mengadaptasinya untuk aplikasi non-game biasa, seperti yang ditunjukkan pada Gambar 5.12, tidaklah terlalu sulit.



Gambar 5.12. Bahkan aplikasi seperti ini memerlukan infrastruktur AR cloud dan "MMO" untuk memungkinkan interaksi waktu nyata

Apa Bagian yang Sulit?

Bayangkan Anda berada di ruangan terkunci tanpa jendela dan Anda diberi foto trotoar kota. Foto tersebut memperlihatkan beberapa bangunan dan nama toko di seberang jalan, mobil, orang, dan sebagainya. Anda belum pernah ke sana sebelumnya, tempat ini benar-benar asing bagi Anda, bahkan tulisannya menggunakan 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 hal ini). Anda memperoleh garis lintang dan garis bujur kasar dari GPS dan hanya mengetahui secara kasar arah yang Anda hadapi, dan Anda tahu GPS bisa saja tidak akurat hingga 20–40 meter.

Yang harus Anda lakukan hanyalah setumpuk foto yang diambil oleh orang lain di area yang hampir sama baru-baru ini, masing-masing ditandai dengan lokasi yang tepat. Ini adalah masalah yang harus dipecahkan oleh sistem AR Anda setiap kali pertama kali dinyalakan atau jika "kehilangan pelacakan" karena kamera tertutup sementara atau jika menunjuk pada sesuatu yang tidak dapat dilacak (dinding putih, langit biru, dll.). Ini juga masalah yang perlu dipecahkan jika Anda ingin bergabung dengan permainan 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 tidak relevan.



Gambar 5.13 Untuk mengetahui seberapa sulitnya melakukan relokasi, coba mainkan permainan Geoguessr, yang sangat mirip dengan masalah yang harus dipecahkan sistem AR Anda setiap kali Anda menyalakannya

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

Dalam contoh lainnya, 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 adalah padanan AR seperti mencoba memainkan permainan papan virtual dan Pemain 1 berada di satu sisi meja dan Pemain 2 duduk di sisi yang berlawanan dan mencoba bergabung dalam permainan. Dengan pengecualian beberapa bagian meja itu sendiri (yang Anda lihat terbalik dengan apa yang ada di tumpukan), sangat sulit bagi sistem untuk menyinkronkan peta mereka (melokalisasi ulang). 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.

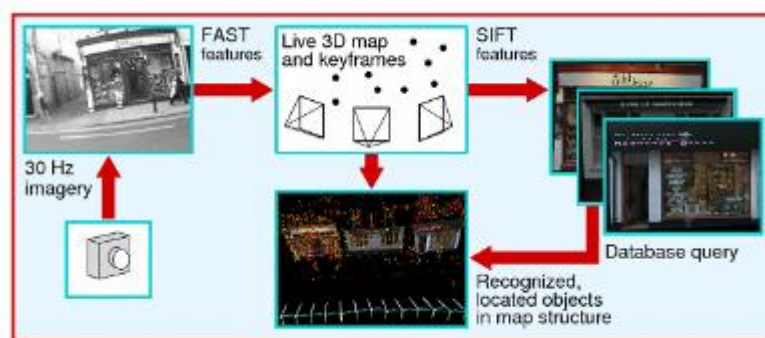
Pengalaman saya dalam membangun sistem AR multipemain sejak 2012 memberi tahu saya bahwa tantangan UX dari contoh pertama (mengharuskan orang untuk berdiri berdampingan untuk memulai) terlalu sulit untuk diatasi oleh pengguna. Mereka membutuhkan banyak bimbingan dan penjelasan, dan gesekannya terlalu tinggi. Mendapatkan pengalaman multipemain tingkat konsumen berarti menyelesaikan kasus kedua (dan lebih banyak lagi). Selain kasus kedua, foto-foto dalam tumpukan tersebut dapat diambil dari jarak yang sangat berbeda, dalam kondisi pencahayaan yang berbeda (bayangan pagi versus sore terbalik) atau menggunakan model kamera yang berbeda, yang memengaruhi tampilan gambar dibandingkan dengan milik Anda (dinding cokelat itu mungkin tidak sama cokelatnya pada gambar Anda seperti pada gambar saya). Anda mungkin juga tidak memiliki GPS (mungkin Anda berada di dalam ruangan), jadi Anda bahkan tidak dapat memulai dengan gambaran kasar tentang di mana Anda mungkin berada.

Hal "menyenangkan" terakhir dari semua ini adalah pengguna menjadi bosan menunggu. Jika proses relokasi memakan waktu lebih dari satu atau dua detik, pengguna umumnya menggerakkan perangkat dengan cara tertentu, dan Anda perlu 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 sebenarnya? Bagaimana masalah-masalah ini dipecahkan saat ini? Apa yang akan segera terjadi? Pada intinya, relokasi adalah jenis masalah pencarian yang sangat spesifik. Anda mencari melalui peta SLAM, yang mencakup area fisik, untuk menemukan lokasi perangkat Anda dalam koordinat peta tersebut. Peta SLAM biasanya memiliki dua jenis data di dalamnya: titik-awan yang jarang dari semua titik 3D yang dapat dilacak di ruang tersebut, dan sejumlah besar bingkai utama. Seperti yang saya sebutkan sebelumnya, bingkai utama adalah satu bingkai video yang diambil dan disimpan sebagai foto setiap saat sistem berjalan.

Sistem memutuskan berapa banyak bingkai utama yang akan diambil berdasarkan seberapa jauh perangkat telah bergerak sejak bingkai utama terakhir serta perancang sistem membuat pengorbanan untuk kinerja. Lebih banyak bingkai utama yang disimpan berarti lebih banyak peluang untuk menemukan kecocokan saat melakukan relokasi, tetapi ini membutuhkan lebih banyak ruang penyimpanan dan juga membutuhkan waktu lebih lama untuk mencari melalui rangkaian bingkai utama.



Gambar 5.14. Tinjauan umum tentang bagaimana sebagian besar sistem SLAM saat ini membangun peta SLAM mereka menggunakan kombinasi fitur optik (titik awan 3D yang jarang) dan basis data bingkai utama

Jadi, proses pencarian sebenarnya memiliki dua bagian, seperti yang diilustrasikan pada Gambar 5.14. Bagian pertama seperti yang baru saja saya jelaskan dalam 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 membuat serangkaian kecil titik 3D miliknya sendiri segera setelah Anda menyalakannya, hanya berdasarkan apa yang saat ini dilihatnya, dan mencari melalui titik-awan SLAM yang jarang untuk menemukan kecocokan. Ini seperti memiliki potongan puzzle 3D (titik-awan kecil dari kamera Anda) dan mencoba menemukan kecocokan dalam puzzle 3D besar, yang setiap bagiannya berwarna abu-abu datar di kedua sisi.

Karena terbatasnya waktu yang tersedia sebelum pengguna merasa bosan dan daya komputasi perangkat seluler saat ini yang paling rendah, sebagian besar upaya dalam relokasi dilakukan untuk mengurangi ukuran jendela pencarian sebelum harus melakukan pencarian brute-force apa pun melalui peta SLAM. GPS yang lebih baik, pelacak yang lebih baik, dan sensor yang lebih baik semuanya sangat membantu dalam hal ini.

Meskipun metode relokasi yang dijelaskan di bagian sebelumnya adalah pendekatan yang paling umum, ada pendekatan lain yang menunjukkan hasil yang bagus di laboratorium dan akan segera hadir dalam produk komersial. Salah satu metode, yang disebut PoseNet (lihat Gambar 5.15) melibatkan penggunaan regresi jaringan saraf bingkai penuh untuk memperkirakan pose perangkat. Tampaknya ini dapat menentukan pose Anda hingga akurasi sekitar satu meter atau lebih dalam berbagai kondisi. Metode lain meregresikan pose kamera untuk setiap piksel dalam gambar.

PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization
 UNIVERSITY OF CAMBRIDGE
 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/>

Single RGB Input Image
 6-DOF Camera Pose

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

2. Dataset
 We built a training dataset of 10M 1024x1024 pixel images. We recorded video then sub-sampled the camera pose in each frame using absolute frame error.

3. PoseNet
 performs end-to-end supervised learning with softmax 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 outputs 6-DOF location even when there are large differences in scale between training (green) and testing (blue) images.

4. Robust to Challenging Situations

Day	Occasion	Dark	Night

5. Regression Outperforms Classification

Scene	# Points	Scale	Dist to Ctr.	Mean Angle	Rotation	RMSE (Cm)
Ward College	1200	1.21 x 10 ²	1.5e-2 0.0	1.5e-2 0.0	1.5e-2 0.0	1.5e-2 0.0
West	90	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1
UCL facade	100	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1
Old Chapel	120	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1
St Marys Church	100	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1
Green	100	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1
Tree	100	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1
Parade	100	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1
Parade	100	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1
Red building	100	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1
Green	100	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1	1.0e+1 1.0e+1

6. Summary
 End-to-end 6-DOF pose regression with context from a single 224x224 image.
 6-DOF, 50 MB, memory of 10 MB.
 Large scale indoor and outdoor relocalization 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.
 Failure: unable to estimate 6-DOF pose from a single image.
 Failure: unable to estimate 6-DOF pose from a single image.
 Failure: unable to estimate 6-DOF pose from a single image.
 Failure: unable to estimate 6-DOF pose from a single image.

Gambar 5.15 PoseNet menunjukkan arah perkembangan sistem

Dapatkan Masalah Relokasi Benar-benar Diselesaikan bagi Konsumen?

Ya! Bahkan, hingga tulisan ini dibuat, telah terjadi beberapa peningkatan yang cukup besar selama 12 bulan terakhir berdasarkan hasil penelitian terkini. Sistem pembelajaran mendalam memberikan hasil yang mengesankan untuk mengurangi jendela pencarian untuk relokasi di area yang luas, atau pada sudut yang sangat lebar terhadap pengguna awal. Pencarian peta SLAM yang dibangun dari titik-titik awan 3D yang padat dari pemandangan (bukan titik-titik awan yang jarang digunakan untuk pelacakan) juga memungkinkan algoritme relokasi baru yang sangat tangguh. Saya telah melihat sistem rahasia yang dapat melakukan relokasi 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 tersebut dapat diterapkan ke sistem kelas komersial, saya yakin ini akan memberikan solusi "kelas konsumen" yang kami harapkan. Namun, ini masih merupakan solusi parsial untuk menyelesaikan relokasi secara menyeluruh untuk lintang dan bujur yang tepat dan untuk lingkungan yang tidak memiliki GPS, atau bagian dunia yang belum pernah memiliki sistem SLAM sebelumnya (cold-start). Namun, saya telah melihat demonstrasi yang memecahkan sebagian besar masalah titik ini, dan percaya bahwa hanya perlu tim yang cerdas untuk mengintegrasikannya secara bertahap menjadi solusi yang lengkap. Relokasi skala besar kini hampir menjadi masalah teknik, bukan masalah sains.

Tidak bisakah Google atau Apple Melakukan Ini?

Tidak juga.

Google telah mendemonstrasikan sebuah layanan yang disebut Visual Positioning System (VPS; lihat Gambar 5-16) untuk platform Tango yang dihentikan produksinya yang memungkinkan beberapa kemampuan relokasi antarperangkat—semacam peta SLAM bersama di awan.



Gambar 5.16. Google telah mengembangkan Layanan Pemosisian Visual berskala besar selama bertahun-tahun

Layanan ini tidak mendukung multipemain, tetapi layanan ini membuat langkah maju untuk menyelesaikan bagian teknis yang sulit. Layanan ini tidak pernah tersedia untuk umum jadi saya tidak dapat mengatakan seberapa baik layanan ini bekerja di dunia nyata, tetapi demonstrasinya terlihat bagus (seperti yang lainnya). Semua perusahaan platform AR utama berupaya meningkatkan relokasi mereka yang merupakan bagian dari ARKit, ARCore, Hololens, Snap, dan sebagainya. Hal ini terutama untuk membuat sistem pelacakan mereka lebih andal, tetapi pekerjaan ini juga dapat membantu multipemain.

VPS adalah contoh yang baik dari peta SLAM bersama yang dihosting di cloud. Namun, peta ini sepenuhnya terikat dengan algoritme dan struktur data SLAM Google, dan peta ini tidak akan digunakan oleh Apple, Microsoft, atau OEM SLAM lainnya (yang mungkin menginginkan sistem mereka sendiri atau bermitra dengan pihak ketiga yang netral).

Masalah besar yang dialami setiap platform utama dengan multipemain adalah bahwa paling banter mereka hanya dapat mengaktifkan multipemain dalam ekosistem mereka—ARCore ke ARCore, ARKit ke ARKit, dan seterusnya. Ini karena agar relokasi lintas platform dapat berfungsi, perlu ada peta SLAM umum di kedua sistem. Ini berarti bahwa Apple perlu memberi Google akses ke data SLAM mentahnya, dan sebaliknya (ditambah Holo-lensa, Magic Leap juga terbuka, dll.). Meskipun secara teknis memungkinkan, ini merupakan jembatan komersial yang terlalu jauh, karena pembeda 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 sepakat untuk membuka semua data mereka satu sama lain, pilihannya terbatas pada hal berikut:

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

Keyakinan pribadi saya adalah bahwa karena integrasi yang sangat erat antara algoritme relokasi SLAM dan struktur data, sistem khusus yang dibangun khusus akan mengungguli (dari aspek UX) sistem terbuka umum untuk beberapa waktu. Ini telah menjadi kasus selama bertahun-tahun dalam visi komputer platform terbuka seperti OpenCV atau berbagai sistem slam terbuka (orb slam, lsd slam, dll.) adalah sistem yang hebat, tetapi mereka tidak memberikan tingkat kinerja yang dioptimalkan yang sama dari sistem yang dikembangkan secara internal yang terfokus. Hingga saat ini, tidak ada perusahaan platform AR yang saya ketahui yang menjalankan atau mempertimbangkan untuk menjalankan sistem open slam, meskipun banyak teknik algoritmik serupa diterapkan dalam sistem milik sendiri 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 sebuah industri, kita memahami masalah AR skala besar dengan cukup baik untuk secara khusus mengatakan bahwa sistem ini perlu terbuka dibandingkan sistem itu perlu dioptimalkan semaksimal mungkin.

5.5 RELOKALISASI MULTIPLEMAIN

Bagian ini membahas mengapa multipemain sulit diterapkan untuk AR. Di bagian sebelumnya, kami menyentuh beberapa masalah, salah satunya adalah tantangan untuk menjadikan relokalisasi sebagai kelas konsumen. Seperti yang telah kami bahas, ada aspek lain yang sulit dibangun, tetapi semuanya merupakan masalah yang telah dipecahkan sebelumnya. Namun, relokalisasilah yang benar-benar penting, dan bukan sekadar multipemain. Berikut ini beberapa masalah yang harus kita atasi:

“Mulai dingin”

Ini merujuk pada saat pertama kali Anda meluncurkan aplikasi atau menyalakan HMD, dan perangkat harus mencari tahu di mana aplikasi itu berada. Umumnya, sistem saat ini bahkan tidak mau repot-repot mencoba menyelesaikan ini, mereka hanya memanggil tempat 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. Dengan melakukan relokasi sistem AR sebagai hal pertama yang dilakukannya, berarti aplikasi AR persisten dapat dibangun karena sistem koordinat akan konsisten dari sesi ke sesi. Jika Anda meletakkan Pokemon Anda di beberapa koordinat tertentu kemarin, saat Anda melakukan relokasi keesokan harinya setelah menyalakan perangkat, koordinat tersebut akan tetap digunakan hari ini dan Pokemon tersebut akan tetap ada di sana. Perhatikan bahwa koordinat ini dapat bersifat unik untuk sistem Anda, dan belum tentu merupakan koordinat global absolut (lintang dan bujur) yang dimiliki oleh semua orang (kecuali jika kita semua melokalisasi ke dalam sistem koordinat global yang sama, yang pada akhirnya akan menjadi tujuan akhir)

Koordinat absolut

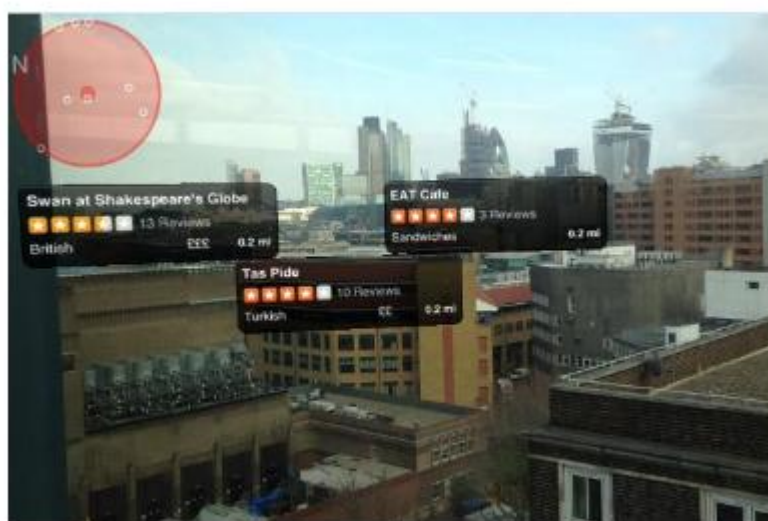
Ini mengacu pada pencarian koordinat Anda dalam hal lintang dan bujur ke tingkat akurasi yang "dapat digunakan AR", yang berarti bahwa koordinat tersebut akurat hingga

tingkat "subpiksel". Subpiksel berarti bahwa koordinat tersebut cukup akurat sehingga konten virtual akan digambar menggunakan piksel yang sama pada perangkat saya seperti pada perangkat Anda jika berada di tempat fisik yang sama persis. Biasanya subpiksel digunakan untuk pelacakan yang merujuk pada jitters/judder sehingga pose yang akurat subpiksel berarti konten tidak bergetar saat perangkat diam, karena pose bervariasi.

Ini juga merupakan angka yang tidak memiliki padanan metrik, karena setiap piksel dapat berkorespondensi 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 dari jarak yang jauh). Dalam praktiknya, memiliki akurasi subpiksel tidak diperlukan karena pengguna tidak dapat membedakan apakah konten tidak konsisten dengan beberapa sentimeter antara perangkat saya dan perangkat Anda. Mendapatkan koordinat lintang dan bujur yang akurat sangat penting untuk setiap 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.

Pelacakan hilang

Hal terakhir yang membuat relokasi penting adalah karena relokasi merupakan bagian penting dari pelacak. Meskipun akan lebih baik jika pelacak tidak pernah "kehilangan pelacakan", bahkan pelacak terbaik pun dapat menghadapi kasus-kasus tertentu yang membingungkan sensor; misalnya, masuk ke dalam kendaraan yang sedang bergerak akan membingungkan IMU dalam sistem VIO, dan dinding kosong dapat membingungkan sistem kamera. Saat pelacakan hilang, sistem perlu kembali dan membandingkan masukan sensor saat ini dengan peta SLAM untuk melakukan relokasi sehingga konten apa pun tetap konsisten dalam sesi aplikasi saat ini. Jika pelacakan tidak dapat dipulihkan, koordinat akan disetel ulang ke (0,0,0) lagi dan semua konten juga disetel ulang.



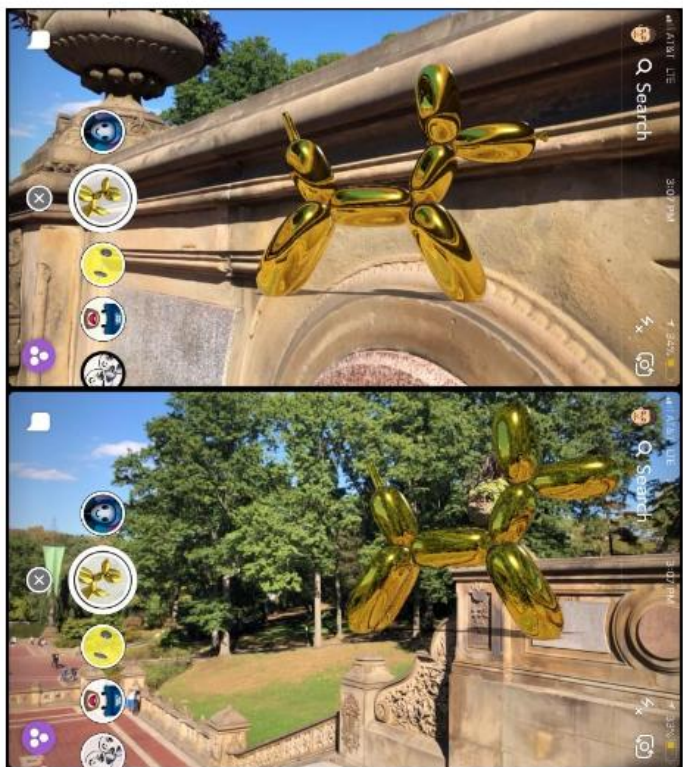
Gambar 5.17. Inilah yang Anda dapatkan saat Anda tidak memiliki koordinat absolut yang akurat (atau jaring 3D kota)

Bagaimana Relokasi Sebenarnya Dilakukan Saat Ini di Aplikasi?

Jawaban singkatnya? Buruk!

Secara umum, ada lima cara relokasi saat ini dilakukan untuk sistem pelacakan dalam-luar (mudah untuk luar-dalam, seperti HTC Vive karena kotak mercusuar eksternal memberikan koordinat umum ke semua perangkat yang dilacak). Berikut adalah uraian masing-masing:

- Andalkan GPS untuk kedua perangkat dan cukup gunakan lintang dan bujur sebagai sistem koordinat umum. Ini sederhana, tetapi objek umum yang ingin kita lihat akan ditempatkan di lokasi fisik yang berbeda untuk setiap ponsel, seperti yang ditunjukkan pada Gambar 5-18, hingga jumlah kesalahan di lokasi GPS (beberapa meter!). Beginilah cara Pokémon Go saat ini mendukung multipemain, tetapi karena back-end MMO masih cukup sederhana, sebenarnya lebih dekat dengan "beberapa orang memainkan permainan pemain tunggal yang sama di lokasi yang sama." Hal ini tidak sepenuhnya akurat, karena begitu Pokémon ditangkap, orang lain tidak dapat menangkapnya, jadi ada beberapa manajemen status sederhana yang dilakukan.



Gambar 5.18. Inilah yang terjadi jika Anda hanya mengandalkan GPS untuk relokasi—kita tidak melihat objek di tempat yang "seharusnya" berada, dan kita bahkan tidak melihatnya di tempat yang sama pada dua perangkat yang berbeda

- Mengandalkan gambar penanda pelacakan fisik yang sama (atau kode QR). Ini berarti kita mengarahkan ponsel kita ke penanda di atas meja di depan kita, seperti yang digambarkan pada Gambar 5-19, dan kedua aplikasi kita memperlakukan penanda tersebut sebagai koordinat asal (0,0,0). Ini berarti dunia nyata dan dunia virtual

- konsisten di kedua ponsel. Ini berfungsi cukup baik, hanya saja tidak ada yang akan membawa penanda tersebut, jadi ini jalan buntu untuk penggunaan di dunia nyata.
- Salin peta SLAM antarperangkat dan minta pengguna untuk berdiri berdampingan, lalu minta Pemain 2 memegang ponsel mereka sangat dekat dengan Pemain 1. Secara teknis, ini dapat berjalan dengan cukup baik; namun, UX hanyalah masalah utama yang harus diatasi pengguna. Beginilah cara kami melakukannya di Dekko for Tabletop Speed.
 - Coba tebak. Jika saya memulai aplikasi ARKit saya dengan berdiri di tempat tertentu, aplikasi saya akan meletakkan titik asal di koordinat awal. Anda dapat datang kemudian dan memulai aplikasi Anda dengan berdiri di tempat yang sama, dan berharap saja bahwa di mana pun sistem menetapkan titik asal Anda kira-kira berada di tempat fisik yang sama dengan titik asal saya. Secara teknis, ini jauh lebih sederhana daripada menyalin peta SLAM, dan rintangan UX-nya hampir sama, dan kesalahan di seluruh sistem koordinat kami tidak terlalu terlihat jika desain aplikasi tidak terlalu sensitif. Anda hanya perlu mengandalkan pengguna untuk melakukan hal yang benar.



Gambar 5.19. Aplikasi ini menggunakan gambar cetak yang digunakan semua perangkat untuk relokasi guna berbagi koordinat

- Batasi UX multipemain untuk menerima lokasi dengan akurasi rendah dan interaksi asinkron. Game jenis Ingress dan perburuan harta karun AR termasuk dalam kategori ini. Mencapai interaksi waktu nyata dengan akurasi tinggi adalah tantangannya. Saya yakin akan selalu ada kasus penggunaan hebat yang mengandalkan interaksi multipengguna asinkron, dan merupakan tugas desainer UX AR untuk mengungkapkannya. Perlu dicatat bahwa kelima solusi ini telah ada selama bertahun-tahun, namun jumlah aplikasi multipemain waktu nyata yang digunakan orang-orang hampir nol. Menurut pendapat saya, semua solusi tersebut termasuk dalam kategori seorang insinyur yang dapat berkata, "Lihat, ini berhasil, kami menyediakan multipemain!" tetapi pengguna akhir menganggapnya terlalu merepotkan dan tidak memberikan banyak manfaat.

Platform

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

Apple

Secara khusus, ARKit adalah sistem VIO, dengan beberapa deteksi bidang 2D sederhana. VIO melacak posisi relatif perangkat Anda di ruang angkasa (pose 6DOF Anda) secara real time; artinya, pose Anda dihitung ulang di antara setiap penyegaran bingkai pada 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 sistem mana dari kedua sistem tersebut yang memberikan estimasi terbaik dari posisi "nyata" Anda (kebenaran dasar) dan menerbitkan pembaruan pose tersebut melalui ARKit SDK. Sama seperti odometer di mobil Anda yang melacak jarak yang ditempuh mobil, sistem VIO melacak jarak yang ditempuh iPhone Anda dalam ruang 6D. 6D berarti 3D gerak XYZ (translasi), ditambah 3D pitch/yaw/roll (rotasi).



Gambar 5.20. ARKit Apple

Keuntungan besar yang dibawa VIO adalah pembacaan IMU dilakukan sekitar 1.000 kali per detik dan didasarkan pada percepatan (gerakan pengguna). Perhitungan mati digunakan untuk mengukur pergerakan perangkat di antara pembacaan IMU. Perhitungan mati lebih merupakan tebakan, sama seperti jika saya meminta Anda untuk melangkah dan memperkirakan berapa inci langkah itu. Kesalahan dalam sistem inersia terakumulasi dari

waktu ke waktu, jadi semakin lama waktu antara bingkai IMU atau semakin lama sistem inersia berjalan tanpa mendapatkan "pengaturan ulang" dari sistem visual, semakin jauh pelacakan akan menyimpang dari kebenaran dasar.

Pengukuran visual/optik dilakukan pada kecepatan bingkai kamera, jadi biasanya 30 bingkai per detik, dan didasarkan pada jarak (perubahan pemandangan di antara bingkai). Sistem optik biasanya mengakumulasi kesalahan dari jarak (dan waktu pada tingkat yang lebih rendah), jadi semakin jauh Anda bepergian, semakin besar kesalahannya.

Kabar baiknya adalah bahwa kekuatan masing-masing sistem membatalkan kelemahan yang lain. Jadi, sistem pelacakan visual dan inersia didasarkan pada sistem pengukuran yang sama sekali berbeda tanpa saling ketergantungan. Ini berarti bahwa kamera dapat ditutupi atau dapat melihat pemandangan dengan sedikit fitur optik (seperti dinding putih) dan sistem inersia dapat "memikul beban" selama beberapa bingkai. Atau, perangkat dapat diam dan sistem visual dapat memberikan pose yang lebih stabil daripada sistem inersia. Filter Kalman terus-menerus memilih pose dengan kualitas terbaik, dan hasilnya adalah pelacakan yang stabil.

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

Bagian utama kedua dari ARKit adalah deteksi bidang sederhana. Hal ini diperlukan agar Anda memiliki "landasan" untuk menempatkan konten Anda; jika tidak, konten tersebut akan terlihat seperti mengambang di luar angkasa. Hal ini dihitung dari fitur yang dideteksi oleh sistem optik (titik-titik kecil, atau titik-awan, yang Anda lihat dalam demonstrasi) dan algoritme tersebut hanya menghitung rata-ratanya karena tiga titik mana pun mendefinisikan sebuah bidang. Jika Anda melakukan ini cukup sering, Anda dapat memperkirakan di mana landasan sebenarnya. Titik-titik ini membentuk titik-awan yang jarang, yang telah kita bahas sebelumnya dalam bab ini, yang digunakan untuk pelacakan optik. Titik-awan yang jarang menggunakan lebih sedikit memori dan waktu CPU untuk melacak, dan dengan dukungan sistem inersia, sistem optik dapat bekerja dengan baik dengan sejumlah kecil titik untuk dilacak. Ini adalah jenis titik-awan yang berbeda dengan titik-awan padat, yang dapat terlihat mendekati fotorealisme (beberapa pelacak yang diteliti dapat menggunakan titik-awan padat untuk pelacakan, sehingga lebih membingungkan).

Beberapa Misteri Terungkap

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 penghilangan kesalahan IMU yang sangat baik (yaitu, membuat perkiraan perkiraan sangat akurat). Ketika Anda dapat melakukannya, berikut ini yang terjadi:

Untuk mendapatkan 3D dari satu lensa, Anda perlu memiliki dua tampilan pemandangan dari tempat yang berbeda, yang memungkinkan Anda melakukan kalkulasi stereoskopik posisi Anda. Ini mirip dengan cara mata kita melihat dalam 3D dan mengapa

beberapa pelacak mengandalkan kamera stereo. Mudah untuk menghitung jika Anda memiliki dua kamera karena Anda mengetahui jarak antara keduanya dan bingkai diambil pada saat yang sama. Untuk menghitung ini hanya dengan satu kamera, Anda perlu mengambil satu bingkai, lalu bergerak, lalu mengambil bingkai kedua. Dengan menggunakan perhitungan jarak IMU, Anda dapat menghitung jarak yang ditempuh antara dua frame dan kemudian melakukan perhitungan stereo seperti biasa (dalam praktiknya, Anda dapat melakukan perhitungan dari lebih dari dua frame untuk mendapatkan akurasi yang lebih baik). Jika IMU cukup akurat, "pergerakan" antara dua frame ini terdeteksi hanya oleh gerakan otot kecil yang Anda buat saat mencoba menahan tangan Anda agar tetap diam! Jadi, hasilnya tampak seperti sulap.

Untuk mendapatkan skala metrik, sistem ini juga mengandalkan perhitungan jarak IMU yang akurat. Dari pengukuran percepatan dan waktu yang disediakan IMU, Anda dapat melakukan integrasi mundur untuk menghitung kecepatan dan melakukan integrasi balik lagi untuk mendapatkan jarak yang ditempuh antara frame IMU. Perhitungannya tidak sulit. Yang sulit adalah menghilangkan kesalahan dari IMU untuk mendapatkan pengukuran percepatan yang hampir sempurna. Kesalahan kecil, yang terakumulasi 1.000 kali per detik selama beberapa detik yang diperlukan untuk menggerakkan ponsel, dapat berarti kesalahan skala metrik sebesar 30% atau lebih. Fakta bahwa Apple telah mengurangi kesalahan ini hingga persentase satu digit sungguh mengesankan.

Bukankah ARCore Hanya Tango-Lite?

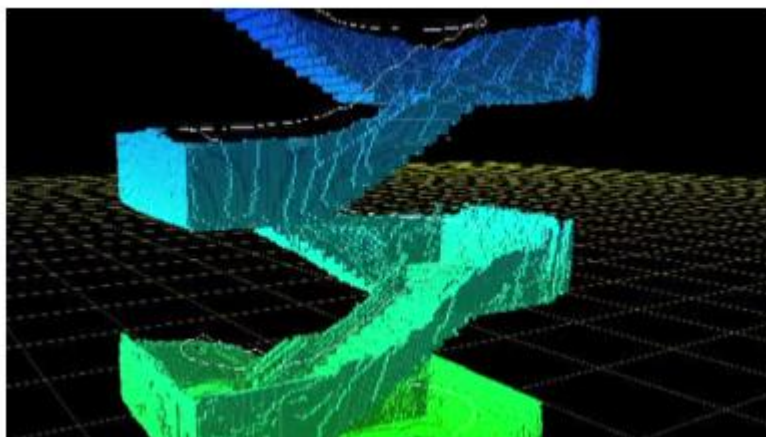
Seorang pengembang yang saya ajak bicara sekitar waktu peluncuran ARCore bercanda, "Saya baru saja melihat SDK ARCore, dan mereka benar-benar mengganti nama SDK Tango, mengomentari kode kamera kedalaman, dan mengubah tanda kompilasi." Saya menduga ada lebih dari itu, tetapi tidak lebih (ini bukan hal yang buruk!). Misalnya, peramban web baru yang mendukung ARCore sangat fantastis bagi 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 telah menyiapkan semua komponennya. Sekarang mereka telah melakukannya! Ini berita bagus, karena ini berarti ARCore adalah perangkat lunak yang sangat matang dan teruji dengan baik (perangkat lunak ini telah dikembangkan setidaknya dua tahun lebih lama di Google daripada ARKit di Apple meskipun pembelian Metaio dan Flyby membantu Apple mengejar ketertinggalan), dan ada peta jalan fitur yang lengkap yang disiapkan untuk Tango, yang tidak semuanya bergantung pada data kedalaman 3D, yang sekarang akan masuk ke ARCore.

Selain penamaannya, jika Anda menambahkan perangkat keras sensor kamera kedalaman ke ponsel yang menjalankan ARCore, Anda akan memiliki ponsel Tango. Sekarang Google memiliki jalur yang jauh lebih mudah untuk mendapatkan adopsi SDK secara luas dengan dapat mengirimkannya pada ponsel unggulan OEM. Tidak seorang pun akan menyerahkan ponsel Android yang hebat demi ponsel yang lebih buruk dengan AR (sama seperti tidak seorang pun akan menyerahkan ponsel hebat apa pun demi ponsel Windows dengan AR, jadi Microsoft tidak peduli; langsung beralih ke HMD). Sekarang orang akan

membeli ponsel yang akan mereka beli, dan ARCore akan ditarik secara gratis.

Banyak ide awal ditujukan untuk pemetaan dalam ruangan. Baru kemudian AR dan VR menjadi kasus penggunaan yang paling populer. Jika kita mempertimbangkan namanya, saya pikir menarik bahwa Tango selalu dideskripsikan sebagai "ponsel yang selalu mengetahui 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 hanyalah renungan (apakah Google melihatnya seperti itu masih bisa diperdebatkan). Dengan nama baru itu, semuanya adalah AR, sepanjang waktu, seperti yang ditunjukkan pada Gambar 5-22.



Gambar 5.21. Tango awalnya lebih berfokus pada pelacakan gerakan ponsel dalam ruang 3D



Gambar 5.22. Google ARCore merupakan evolusi dari 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-orang sangatlah menantang. Akan jauh lebih sedikit upaya untuk mempelajari cara membangun di ARKit atau ARCore daripada upaya untuk mempelajari apa yang harus dibangun. Ingat juga bahwa SDK ARKit/ARCore adalah versi 1.0. SDK tersebut benar-benar dasar (VIO, deteksi bidang, pencahayaan dasar) dan akan menjadi jauh lebih lengkap dalam beberapa tahun ke depan (pemahaman pemandangan 3D, oklusi, multipemain, persistensi konten, dll.). Ini akan menjadi kurva pembelajaran yang konstan bagi pengembang dan konsumen. Namun untuk saat ini, fokuslah pada pembelajaran hal yang sulit (aplikasi apa yang akan dibuat) dan berpeganglah pada apa yang Anda ketahui untuk teknologi yang mendasarinya (cara membuatnya: Android, IOS Xcode, dll.). Setelah Anda memahami apa yang membuat aplikasi bagus, buatlah keputusan mengenai platform terbaik untuk meluncurkan aplikasi terkait jangkauan pasar, dukungan fitur AR, monetisasi, dan sebagainya.

Bagaimana dengan Tango, Hololens, Vuforia, dan Lainnya?

Jadi, Tango adalah sebuah merek (sudah dimatikan oleh Google), bukan benar-benar sebuah produk. Tango terdiri dari desain referensi perangkat keras (RGB, fisheye, kamera kedalaman, dan beberapa spesifikasi CPU/GPU) dan tumpukan perangkat lunak yang menyediakan VIO (pelacakan gerakan), pemetaan renggang (pembelajaran area), dan rekonstruksi 3D padat (persepsi kedalaman). Hololens (dan Magic Leap) memiliki tumpukan perangkat lunak yang sama persis, tetapi mencakup beberapa chip pemrosesan sinyal digital (DSP) dasar, yang mereka sebut sebagai Unit Pemrosesan Holografik, untuk memindahkan pemrosesan dari CPU/GPU dan menghemat daya. Desain chip yang lebih baru dari Qualcomm akan memiliki fungsionalitas ini, sehingga menghilangkan kebutuhan untuk pemrograman DSP khusus dan mengurangi biaya perangkat keras di masa mendatang.

Vuforia pada dasarnya sama lagi, tetapi tidak bergantung pada perangkat keras. Masing-masing menggunakan jenis sistem VIO yang sama. Baik Hololens, Magic Leap, maupun Tango tidak menggunakan kamera kedalaman untuk pelacakan (meskipun saya yakin mereka mulai mengintegrasikannya untuk membantu dalam beberapa kasus khusus). Jadi mengapa ARKit begitu bagus?

Jawabannya adalah ARKit tidak lebih baik dari Hololens, tetapi perangkat keras Hololens tidak tersedia secara luas. Jadi, pada akhirnya, alasan ARKit lebih baik adalah karena Apple mampu melakukan pekerjaan untuk menggabungkan algoritma VIO dengan sensor dan menghabiskan banyak waktu untuk mengkalibrasinya guna menghilangkan kesalahan dan ketidakpastian dalam kalkulasi pose.

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

Pertimbangan Pengembangan Lainnya

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

Pencahayaan

Baik ARKit maupun ARCore memberikan perkiraan sederhana tentang pencahayaan alami dalam adegan, seperti yang ditunjukkan pada Gambar 5-23. Ini adalah satu perkiraan untuk adegan, terlepas dari apakah dunia nyata diterangi dengan cahaya sekitar atau penuh dengan lampu sorot yang tajam. ARKit menyerahkan kendali intensitas dan suhu warna kembali ke pengembang, sedangkan ARCore memberikan nilai intensitas piksel tunggal (API Android Studio) atau shader (API Unity). Kedua pendekatan tersebut tampaknya memberikan hasil yang serupa dari demonstrasi awal. Secara subjektif, 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 gerakan cahaya dunia nyata. Ini akan memberikan peningkatan besar dalam kehadiran di mana kita secara tidak sadar percaya konten tersebut "benar-benar ada".



Gambar 5.23. ARCore dan ARKit memberikan estimasi cahaya dalam adegan secara real-time (sederhana), sehingga pengembang dapat langsung menyesuaikan pencahayaan simulasi agar sesuai dengan dunia nyata (dan mungkin memicu animasi pada saat yang sama)

AR Multipemain Mengapa Cukup Sulit

Sebelumnya dalam bab ini, kita telah membahas apa yang membuat aplikasi AR ponsel pintar menjadi hebat dan mengapa ARKit dan ARCore telah memecahkan masalah teknis yang

sangat sulit (pelacakan dalam-luar 6DOF yang tangguh) dan menciptakan platform bagi AR untuk akhirnya mencapai penggunaan arus utama (masih perlu beberapa tahun lagi untuk diadopsi secara luas, tetapi menurut saya ada banyak ceruk besar untuk aplikasi saat ini). Pengembang sekarang berupaya untuk meningkatkan kurva pembelajaran dari aplikasi kentut menjadi aplikasi yang bermanfaat (meskipun putra saya yang berusia sembilan tahun menganggap aplikasi kentut cukup bermanfaat, terima kasih). Satu fitur yang paling banyak ditanyakan orang daripada fitur lainnya adalah multipemain. Istilah "multipemain" sebenarnya adalah istilah yang salah kaprah, karena yang kami maksud adalah kemampuan untuk berbagi pengalaman AR Anda dengan orang lain, atau banyak orang lain, secara real-time. Jadi, menyebutnya "multipengguna", "Sharing AR", "Social AR", dan "AR Communi- cation" juga merupakan istilah yang bagus, tetapi multipemain tampaknya masih populer saat ini, mungkin karena sebagian besar alat AR 3D berasal dari latar belakang permainan, dan itulah istilah yang digunakan oleh para pemain game. Perhatikan bahwa Anda dapat melakukan multipemain secara asinkron, tetapi itu seperti bermain catur dengan sahabat pena. Sebagai tambahan, saya tidak sabar menunggu alat-alat baru untuk AR yang selaras dengan alur kerja disiplin desain yang lebih tradisional (arsitek, desainer produk, desainer UX, dll.) karena saya pikir itu akan mendorong peningkatan besar pada utilitas aplikasi AR. Tetapi itu untuk buku lain.

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

Jadi, jika multipemain adalah fitur utama yang diminta orang, mengapa kita tidak memilikinya? Jawabannya, seperti banyak fungsi AR lainnya, berarti menyelami teknologi visi komputer yang memungkinkan AR. (Kita juga memerlukan jaringan latensi rendah, pemeliharaan model dunia yang konsisten, berbagi audio dan video, dan metafora interaksi kolaboratif, tetapi bagian ini berfokus pada tantangan visi komputer, yang belum benar-benar terpecahkan.) AR multipemain saat ini agak mirip dengan pelacakan posisi 6DOF beberapa tahun yang lalu. 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 digunakan untuk mengaktifkan multipemain, tetapi yang perlu diperhatikan adalah teman lama kita: relokasi. Aspek multipemain non-intuitif lainnya adalah ia memerlukan beberapa infrastruktur di cloud agar dapat berfungsi dengan baik.

Bagaimana Orang Terhubung Melalui AR?

Bagaimana kita mendukung banyak pengguna untuk berbagi pengalaman? Bagaimana kita melihat hal-hal virtual yang sama pada saat yang sama, tidak peduli perangkat apa yang kita pegang atau kenakan, saat kita berada di tempat yang sama (atau tidak)? Anda dapat memilih istilah yang familier untuk menggambarkan kemampuan ini berdasarkan apa yang

sudah Anda ketahui: aplikasi "multipemain" untuk para gamer, atau aplikasi "sosial" atau aplikasi "berkomunikasi". Semuanya adalah infrastruktur yang sama di balik layar dan dibangun di atas teknologi yang sama. Lokalisasi yang benar-benar tangguh, streaming pose 6DOF dan status sistem, penjahitan mesh 3D, dan pembaruan mesh yang bersumber dari banyak orang adalah semua masalah teknis yang harus dipecahkan di sini. Jangan lupakan tantangan tingkat aplikasi seperti hak akses, autentikasi, dan sebagainya (meskipun sebagian besar sekarang merupakan masalah teknis).



Gambar 5.24. Game “The Machines” yang didemonstrasikan Apple pada keynote-nya menggunakan sistem multipemain yang dikembangkan sendiri secara sederhana (demonstrasi yang bagus, tetapi bukan cloud AR)

Bagaimana Aplikasi AR Terhubung dengan Dunia dan Mengetahui Lokasi Mereka Sebenarnya?

GPS bukanlah solusi yang cukup baik—bahkan GPS yang akan datang yang akurat hingga satu kaki.



Gambar 5.25. Hal semacam ini tidak mungkin dilakukan tanpa cloud AR

Bagaimana Aplikasi AR Memahami dan Terhubung dengan Benda di Dunia Nyata?

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

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

Bagaimana kita membuat AR berfungsi di luar ruangan 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 mencapainya baik di dalam maupun di luar ruangan? Bagaimana kita memastikan konten tetap berada di tempatnya, bahkan beberapa hari atau tahun kemudian? Bagaimana kita mengelola begitu banyak data? Melokalisasi terhadap koordinat absolut adalah masalah teknis yang sangat sulit untuk dipecahkan di sini.



Gambar 5.26. Agar ponsel Anda dapat mengetahui hal ini saat Anda berjalan melewatinya sambil mengambil dan mengelola struktur data 3D yang terlibat, diperlukan cloud AR

Tepat saat Anda mengira telah memahami perbedaan antara AR, VR, dan MR, semuanya menjadi lebih mendalam. Vendor akan menggunakan istilah yang identik yang memiliki arti yang sama sekali berbeda, seperti berikut ini:

AR Multipemain

Ini dapat merujuk pada cara yang murni pada level permainan untuk melacak apa yang dilakukan setiap pemain dalam permainan itu sendiri tanpa visi komputer atau kesadaran spasial. Atau, ini dapat merujuk pada cara untuk memecahkan beberapa masalah pelokalan visi komputer yang sangat sulit. Atau, keduanya. Atau keduanya dapat memiliki arti yang sama sekali berbeda.

AR Luar Ruang

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

Pengenalan

Ini mungkin berarti mengonfigurasi penanda atau gambar tunggal yang dapat dikenali aplikasi Anda secara manual. Atau, ini mungkin berarti mesin klasifikasi objek 3D global yang didukung pembelajaran mesin serba guna dan waktu nyata.

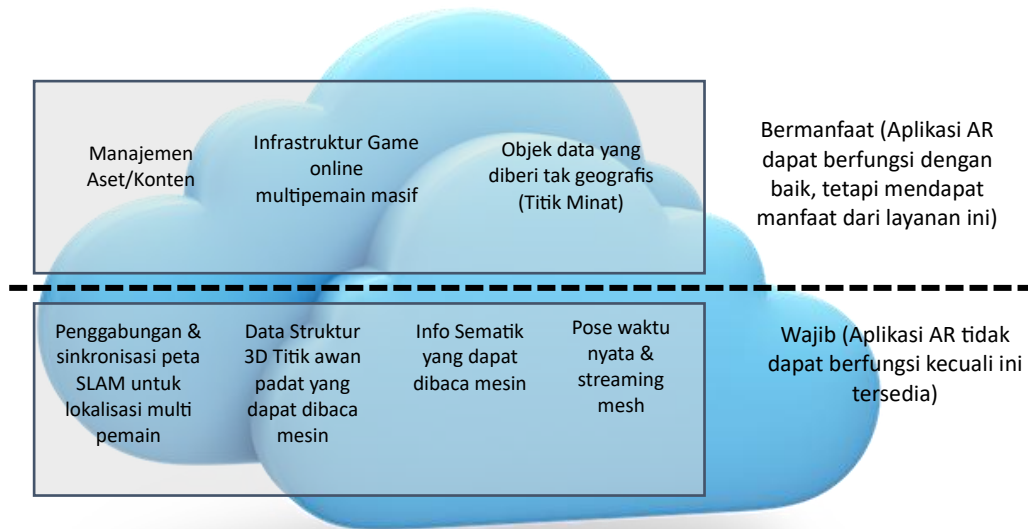
AR Cloud

Jika Anda memikirkan semua bagian aplikasi yang ada di cloud, saya cenderung membagi "cloud" secara horizontal dan memisahkan layanan tersebut menjadi hal-hal yang "bagus untuk dimiliki" di bagian atas, dan "harus dimiliki" di bagian bawah (Gambar 5-27). Hal-hal yang bagus untuk dimiliki umumnya terkait dengan aplikasi dan konten serta memudahkan pembuatan dan pengelolaan aplikasi dan pengguna.

Apa yang Saya Bayangkan Saat Memikirkan AR Cloud

Aplikasi AR Anda saat ini tanpa koneksi cloud AR seperti memiliki ponsel yang hanya dapat memainkan Snake.

Bagi saya, bagian bawah cloud adalah bagian yang menarik. Sistem AR, pada dasarnya, terlalu besar untuk sebuah perangkat. Dunia terlalu besar untuk muat di dalamnya, dan itu seperti mencoba memasukkan semua peta Google dan seluruh web di ponsel Anda (atau HMD). Wawasan utamanya adalah jika Anda ingin aplikasi AR Anda dapat berbagi pengalaman atau bekerja dengan baik (misalnya, dengan kesadaran akan dunia 3D tempatnya berada) di lokasi mana pun, aplikasi tersebut tidak dapat bekerja sama sekali tanpa akses ke layanan cloud ini. Layanan ini 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 berada di perangkat, dan sebagian berada di cloud. Layanan data jaringan dan cloud sama pentingnya bagi aplikasi AR seperti halnya jaringan untuk melakukan panggilan telepon seluler. Pikirkan kembali sebelum ada telepon pintar—ponsel Nokia lama Anda tanpa jaringan masih dapat menjadi kalkulator dan Anda dapat memainkan Snake, tetapi kegunaannya cukup terbatas. Jaringan dan cloud AR akan menjadi sama pentingnya bagi aplikasi AR. Saya yakin kita akan melihat aplikasi ARKit/ARCore saat ini sebagai padanan dari sekadar memiliki "Nokia Snake" offline dibandingkan telepon yang terhubung jaringan.



Gambar 5.27. Awan AR dapat dikelompokkan menjadi dua lapisan: bagian awan yang bagus untuk dimiliki yang membantu aplikasi, dan bagian yang harus dimiliki, yang tanpanya aplikasi tidak akan berfungsi sama sekali

Seberapa Penting Awan AR?

Jika Anda ditanya apa aset tunggal paling berharga dalam 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 aset-aset ini yang tidak ada saat ini. Mungkin lebih berharga jika Anda melihatnya dalam konteks aset sistem operasi Microsoft Windows (yang merupakan aset teknologi paling berharga pada tahun 1990-an) pada tahun 2017 dibandingkan dengan tahun 1997.

Akankah satu perusahaan pada akhirnya memiliki (bagian yang sangat menguntungkan) aset tersebut? Sejarah mengatakan mungkin. Apakah itu akan menjadi perusahaan baru? Mungkin juga. Sama seperti pada tahun 1997, mustahil untuk membayangkan Microsoft kehilangan posisinya, pada tahun 2019 tampaknya mustahil Google atau Facebook akan kehilangan posisi mereka. Namun, tidak ada yang pasti. Saya akan mencoba memaparkan argumen yang mendukung masing-masing dari tiga pihak yang bermain di sini (petahana, perusahaan rintisan, web terbuka) di bagian terakhir bab ini.

Sebelumnya, kita telah membahas cara kerja ARKit dan ARCore. Kita membahas apa yang tersedia saat ini dan bagaimana kita sampai di sini. Di bagian selanjutnya, kita akan melihat apa yang hilang dari ARKit dan ARCore dan bagaimana bagian-bagian yang hilang itu akan bekerja.

Jadi, Apa Sebenarnya AR Cloud Ini?

Untuk melampaui ARKit dan ARCore, kita perlu mulai berpikir melampaui diri kita sendiri. Bagaimana orang lain pada perangkat AR jenis lain bergabung dengan kita dan berkomunikasi dengan kita dalam AR? Bagaimana aplikasi kita bekerja di area yang lebih besar dari ruang keluarga kita? Bagaimana aplikasi kita memahami dan berinteraksi dengan dunia? Bagaimana kita dapat meninggalkan konten untuk ditemukan dan digunakan orang lain? Untuk

menghadirkan kemampuan ini, kita memerlukan infrastruktur perangkat lunak berbasis cloud untuk AR.

Awan AR dapat dianggap sebagai model dunia nyata berskala 1:1 yang dapat dibaca mesin. Perangkat AR kita adalah antarmuka waktu nyata ke dunia virtual paralel ini, yang dihiperkonfirmasi dengan sempurna ke dunia fisik.

Mengapa Semua "Biasa Saja" dari Pers untuk ARKit dan ARCore? Ketika ARKit diumumkan di WWDC tahun ini, kepala eksekutif Apple Tim Cook memuji augmented reality, dan mengatakan kepada para analis, "Ini adalah salah satu hal besar yang akan kita lihat kembali dan kagumi pada awalnya."

Beberapa bulan berlalu. Para pengembang bekerja keras pada hal besar berikutnya, tetapi reaksi terhadap ARKit pada acara peluncuran utama iPhone adalah, "biasa saja." Mengapa demikian?

Itu karena ARKit dan ARCore saat ini berada pada versi 1.0. Mereka hanya memberi pengembang tiga alat AR yang sangat sederhana:

- Pose 6DOF ponsel, dengan koordinat baru setiap sesi
- Bidang 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-kata "augmented" dan "reality" bersama-sama di atas panggung, kami mengabaikan bahwa Anda benar-benar tidak dapat membangun sesuatu yang terlalu mengesankan hanya dengan tiga alat tersebut. Masalah terbesar mereka adalah orang-orang mengharapkan aplikasi yang luar biasa sebelum perangkat lengkap untuk membangunnya tersedia. Namun, bukan soal apakah, tetapi kapan, yang keliru.

Apa yang Hilang untuk Membuat Aplikasi AR yang Hebat?

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

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

Apakah Cloud Seluler Saat Ini Mampu Berfungsi?

Ketika saya bekerja di infrastruktur telekomunikasi, ada sedikit kebenaran yang berbunyi, "Tidak ada cloud, itu hanya komputer orang lain." Kami selalu berakhir bekerja dengan pasangan tembaga atau untai serat (atau spektrum radio) yang secara fisik menghubungkan satu komputer ke komputer lain, bahkan di seluruh dunia. Itu bukan sihir, hanya sulit. Yang membuat infrastruktur cloud AR berbeda dari cloud saat ini, yang mendukung

aplikasi web dan seluler kita, adalah bahwa AR (seperti mobil, drone, dan robot yang dapat mengemudi sendiri) adalah sistem waktu nyata. Siapa pun yang pernah bekerja di bidang telekomunikasi (atau pada infrastruktur gim MMO fast-twitch) sangat memahami bahwa infrastruktur waktu nyata dan infrastruktur asinkron adalah dua hal yang sama sekali berbeda.

Jadi, meskipun banyak bagian dari cloud AR akan melibatkan hosting data besar dan penyediaan API web serta pelatihan model pembelajaran mesin—seperti cloud saat ini—perlu ada pemikiran ulang yang sangat besar tentang bagaimana kita mendukung aplikasi waktu nyata dan interaksi AR dalam skala besar. Kasus penggunaan AR dasar seperti streaming model 3D langsung dari ruangan kita saat kita melakukan "AR Skype"; memperbarui data dan aplikasi yang terhubung ke berbagai hal, yang disajikan saat saya lewat di transportasi umum; streaming data (grafis yang kaya) kepada saya yang berubah tergantung ke mana mata saya melihat, atau siapa yang berjalan di dekat saya; dan memelihara serta memperbarui status aplikasi waktu nyata setiap orang dan aplikasi di tengah kerumunan besar di sebuah konser. Tanpa jenis UX ini, AR tidak ada gunanya. Mari kita tetap menggunakan aplikasi telepon pintar. Mendukung hal ini untuk miliaran orang pada akhirnya akan menjadi peluang besar. Jaringan 5G akan memainkan peran besar dan dirancang khusus untuk kasus penggunaan ini. Jika sejarah menjadi petunjuk, beberapa, jika tidak sebagian besar, dari perusahaan mapan saat ini yang memiliki investasi besar dalam infrastruktur cloud saat ini tidak akan memanfaatkan 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 hal ini. "Tidak berguna" adalah pilihan kata yang provokatif. Sejauh ini, setelah satu bulan, berdasarkan metrik awal, pengguna cenderung memilih "hampir tidak berguna." Mungkin itu adalah hal baru yang menyenangkan yang membuat Anda tersenyum saat membagikannya. Mungkin jika Anda membeli sofa, Anda akan mencobanya terlebih dahulu. Namun, ini bukanlah aplikasi penting untuk penggunaan sehari-hari yang mendefinisikan platform baru. Untuk itu, kita memerlukan aplikasi asli AR. Aplikasi yang benar-benar terhubung dengan dunia nyata. Dan untuk menghubungkan aplikasi AR kita satu sama lain dan dunia, kita memerlukan infrastruktur yang ada untuk melakukannya. Kita memerlukan cloud AR.

Fajar AR Cloud

Sejak konferensi WWDC Apple pada tahun 2017, yang menjadi titik awal AR konsumen dengan peluncuran ARKit, kita telah melihat setiap platform besar mengumumkan strategi AR: ARCore milik Google; platform kamera milik Facebook; Amazon Sumerian; dan Microsoft yang terus membangun ekosistem realitas campurannya. Kita juga telah melihat ribuan pengembang bereksperimen dengan aplikasi AR tetapi sangat sedikit yang diterima oleh konsumen. Pada bulan September 2017, saya meramalkan bahwa aplikasi AR akan kesulitan untuk berinteraksi tanpa AR cloud, dan ini tentu saja terbukti demikian. Namun, kita sekarang menyaksikan fajar layanan cloud yang akan membuka kemampuan menarik bagi pengembang AR, tetapi hanya jika penyedia cloud mendapatkan UX yang tepat. Ini bukan tentang menjadi yang pertama di pasar, tetapi yang pertama untuk mencapai UX tingkat konsumen.

Apakah ada yang ingat AR sebelum ARKit dan ARCore? Secara teknis berhasil, tetapi

UX-nya kikuk. Anda memerlukan spidol cetak atau memegang dan menggerakkan ponsel dengan hati-hati untuk memulai, dan hasilnya cukup baik. Video demonstrasi yang bagus dibuat untuk menunjukkan pengalaman kerja akhir, yang membuat orang kagum. Hasilnya: tidak ada yang menggunakannya. Memecahkan masalah teknis (meskipun masalah teknis yang cukup sulit) ternyata sangat berbeda dengan mencapai UX yang dapat digunakan konsumen. Baru setelah ARKit diluncurkan, UX "yang berfungsi" untuk AR dasar tersedia (dan ini terjadi 10 tahun setelah Mobile SLAM ditemukan di Oxford Active Vision Lab, yang dipimpin oleh Victor Prisacariu, salah satu pendiri 6D.ai saya).

Kita memasuki masa 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 tersebut (dan, yang lebih penting, nilai dari penyelesaian masalah ini), kini kita melihat platform AR terbesar mulai mengadopsi bahasa ini sebagai pengakuan atas masalah yang sangat penting. Saya mendengar rumor kuat bahwa Google tidak akan menjadi perusahaan bernilai miliaran dolar terakhir yang mengadopsi bahasa cloud AR pada tahun 2018.

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

- Mengenali bahwa orang tidak akan melewati rintangan. Aplikasi tersebut tidak perlu meminta Pemain 2, 3, 4, dan seterusnya untuk "datang pertama dan berdiri di samping saya" atau "mengetik beberapa info." Sistem SLAM yang sinkron hanya perlu bekerja dari mana pun pengguna berdiri saat mereka ingin bergabung; yaitu, dari sudut atau jarak relatif apa pun di antara pemain.
- Hilangkan atau minimalkan "pemindaian awal", terutama jika pengguna tidak mengerti mengapa hal itu diperlukan atau menerima umpan balik yang diberikan tentang apakah mereka melakukannya dengan benar.
- Setelah sistem disinkronkan (yaitu, dilokalkan ulang ke dalam kumpulan koordinat dunia bersama), konten harus memiliki penyelarasan yang akurat. Ini berarti bahwa kedua sistem setuju bahwa titik virtual x,y,z yang sama cocok dengan titik yang sama persis di dunia nyata. Secara umum, perbedaan beberapa sentimeter antara perangkat dapat diterima dalam hal persepsi pengguna. Namun, ketika (akhirnya) jaring oklusi dibagikan, kesalahan penyelarasan apa pun sangat terlihat karena konten "dipotong" tepat sebelum melewati objek fisik. Penting untuk dicatat bahwa pelacak ARCore dan ARKit yang mendasarinya hanya akurat hingga sekitar tiga hingga lima sentimeter, jadi mendapatkan penyelarasan yang lebih baik daripada itu saat ini mustahil untuk sistem relocalizer multipemain mana pun.
- Pengguna tidak perlu menunggu. Sistem koordinat sinkronisasi harus instan dan tidak

memerlukan klik apa pun. Idealnya, instan berarti sepersekian detik, tetapi seperti yang akan dikatakan oleh setiap desainer aplikasi seluler, pengguna akan bersabar hingga dua hingga tiga detik sebelum merasa sistemnya terlalu lambat.

- Pengalaman multipemain harus bekerja lintas platform, dan UX harus konsisten di semua perangkat.
- Pengelolaan data penting. Pengelolaan mengacu pada "pengelolaan yang cermat dan bertanggung jawab atas sesuatu yang dipercayakan kepada seseorang," dan ini adalah kata yang kami gunakan di 6D.ai ketika kami berpikir tentang data cloud AR. Pengguna mempercayakannya kepada kami. Ini adalah masalah yang berkembang karena orang-orang mulai memahami bahwa data mereka yang tersimpan dapat digunakan untuk hal-hal yang tidak dijelaskan sebelumnya atau bahwa data tersebut dapat diretas dan digunakan secara kriminal.

Namun, orang-orang juga umumnya menerima tawaran bahwa "Saya akan membagikan beberapa data jika saya mendapat keuntungan sebagai imbalannya." Masalah muncul ketika perusahaan menyesatkan atau tidak kompeten sehubungan dengan tawaran ini daripada transparan. Jadi, selain semua aspek tingkat aplikasi dari UI multipemain (seperti tombol lobi dan daftar pemilih untuk memilih untuk bergabung dalam permainan), bagian SLAM-synch bukan sekadar kotak centang, tetapi UX itu sendiri. Jika UX itu tidak memberikan hasil "berfungsi dengan baik", pengguna bahkan tidak akan repot-repot mencoba aplikasi untuk kedua kalinya. Namun, mereka akan mencoba sekali karena penasaran, yang berarti bahwa pengamat pasar tidak boleh memperhatikan unduhan aplikasi AR atau pengguna terdaftar, tetapi untuk mengulangi penggunaan.

Memberikan kesempatan kepada pengembang untuk membangun aplikasi AR yang menarik adalah fokus perusahaan cloud AR, dengan memecahkan masalah teknis yang menantang untuk mengaktifkan aplikasi AR-First yang asli untuk AR. Ini berarti (seperti yang telah saya pelajari dengan susah payah beberapa kali) bahwa UX adalah yang utama. Meskipun kami adalah perusahaan visi komputer teknologi mendalam, UX dari cara kerja sistem visi komputer tersebut adalah yang terpenting, bukan apakah sistem tersebut berfungsi atau tidak.

5.6 PRIVASI DAN DATA AR CLOUD

Jika berbicara tentang Google Cloud Anchors, data gambar visual dikirim ke server Google. Asumsi yang cukup aman adalah bahwa data ini berpotensi direkayasa ulang menjadi gambar yang dapat diidentifikasi secara pribadi (Google tidak menjelaskan secara rinci, jadi saya berasumsi bahwa jika benar-benar anonim, mereka akan mengatakannya dengan jelas).

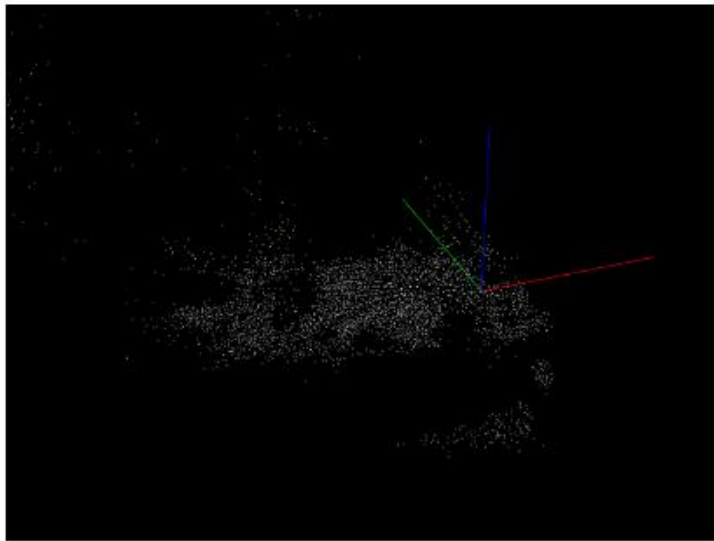
Ini adalah data gambar sumber yang tidak boleh meninggalkan ponsel, dan tidak boleh disimpan di ponsel atau disimpan dalam memori (Gambar 5.28). Ini adalah jenis data gambar visual yang dapat diidentifikasi secara pribadi yang tidak ingin Anda simpan atau pulihkan dari penyedia cloud AR. Google mengatakan bahwa mereka tidak mengunggah bingkai video, tetapi deskriptor poin fitur dapat direkayasa ulang menjadi gambar (lihat Gambar 5.29).



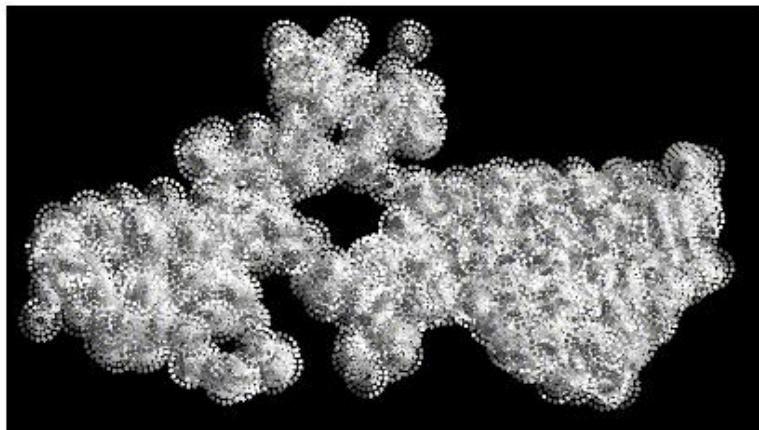
Gambar 5.28. Data gambar yang dapat dilihat oleh manusia tidak boleh meninggalkan ponsel

Demi masa depan kemampuan cloud AR dalam menghadirkan persistensi dan relokasi, data gambar visual tidak boleh meninggalkan ponsel, dan bahkan tidak boleh disimpan di ponsel. Menurut saya, semua pemrosesan yang diperlukan harus dijalankan di perangkat secara real time. Dengan izin pengguna, yang harus diunggah hanyalah peta titik renggang pascaproses dan deskriptor fitur, yang tidak dapat direkayasa balik. Tantangan menarik yang kami (dan yang lainnya) hadapi adalah bahwa seiring perangkat mengembangkan kemampuan untuk menangkap, menggabungkan, dan menyimpan titik-awan padat, jaring, dan tekstur fotorealistik, semakin banyak nilai dalam produk semakin "dapat dikenali" data yang ditangkap. Kami yakin ini akan memerlukan pendekatan semantik baru untuk segmentasi data 3D dan identifikasi spasial untuk memberi pengguna tingkat kontrol yang sesuai atas data mereka; ini adalah area yang sedang dieksplorasi oleh kelompok penelitian Oxford kami.

Gambar 5.29 menyajikan titik-awan yang jarang untuk pemandangan pada Gambar 5.28 (sistem kami memilih titik-titik yang jarang semi-acak, bukan sudut dan tepi geometris, yang tidak dapat dijalin menjadi ruang geometris yang dapat dikenali). Bagian kedua dari teka-teki ini adalah "deskripsi fitur," yang disimpan oleh kami dan juga Google di awan. Google sebelumnya telah mengatakan bahwa file Tango ADF, yang menjadi dasar ARCore, dapat memiliki deskriptor fitur visual yang direkayasa ulang dengan pembelajaran mendalam kembali menjadi gambar yang dapat dikenali manusia (Gambar 5.30) (dari dokumentasi Tango ADF — "pada prinsipnya mungkin untuk menulis algoritma yang dapat merekonstruksi gambar yang dapat dilihat"). Perlu dicatat bahwa saya tidak tahu apakah ARCore mengubah spesifikasi jangkar dari ADF Tango secara cukup untuk mengubah fakta ini, tetapi Google telah menjelaskan bahwa ARCore didasarkan pada Tango, dan mengubah struktur data deskriptor fitur merupakan perubahan yang cukup mendasar pada algoritme tersebut.



Gambar 5.29. Titik-awan ini didasarkan pada data gambar kantor yang ditunjukkan pada Gambar 5.28



Gambar 5.30. Ini adalah deskriptor fitur yang dihasilkan untuk setiap titik di titik awan (ini sejauh mana data yang dihosting awan 6D.ai dapat direkayasa ulang, berdasarkan penerapan ilmu pengetahuan terbaru yang tersedia saat ini bersama dengan sumber daya komputasi yang besar)

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

Realitas mengenai data awan AR ini menciptakan masalah pasar struktural untuk semua perusahaan platform AR utama saat ini, mengingat model bisnis Google dan Facebook (dan yang lainnya) dibangun berdasarkan penerapan data yang mereka kumpulkan untuk melayani iklan Anda dengan lebih baik. Platform seperti Apple dan Microsoft bersifat silo dan

dengan demikian tidak akan menawarkan solusi lintas platform. Mereka juga tidak akan memprioritaskan solusi cloud yang memungkinkan solusi P2P pada perangkat yang bersifat kepemilikan.

Satu faktor yang saya remehkan adalah bahwa pengembang dan mitra besar memahami dengan jelas nilai data yang dihasilkan oleh aplikasi mereka, dan mereka tidak ingin memberikan data tersebut kepada platform besar untuk dimonetisasi oleh organisasi tersebut. Mereka ingin membawa semuanya sendiri (seperti yang dilakukan Niantic) atau bekerja dengan mitra yang lebih kecil yang dapat memberikan kesetaraan teknologi dengan platform besar (bukan tugas kecil) dan yang juga dapat menjamin privasi dan keselarasan model bisnis. AR dianggap terlalu penting untuk memberikan fondasi data. Ini adalah keuntungan pasar struktural yang dimiliki oleh perusahaan rintisan cloud AR, dan ini merupakan tanda yang menggembirakan untuk masa depan kita yang dapat diperkirakan.

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

BAB 6

REALITAS VIRTUAL DAN REALITAS TERTAMBAH

Langsung terjun ke dalam membangun pengalaman realitas virtual (VR) bisa sangat menakutkan. Hal yang sama berlaku untuk realitas tertambah (AR), dan melakukan curah pendapat untuk membuat keputusan sederhana mengenai apakah pengalaman tersebut harus dilakukan dalam VR atau AR adalah latihan yang baik. Demi kesederhanaan, bab ini menjelaskan pengalaman VR dan AR secara bergantian sebagai "immersive" karena mayoritas pengembangan konten imersif menggunakan prinsip yang sama. Jika ada perbedaan yang bernuansa antara keduanya, kami merujuknya secara eksplisit.

Langkah pertama mempelajari pengembangan imersif bergantung pada perspektif dan tujuan Anda. Anda kemungkinan besar mengidentifikasi diri dengan satu atau beberapa pernyataan berikut:

- ❖ Saya tidak memiliki pengalaman pengembangan.
- ❖ Saya memiliki pengalaman pengembangan dalam grafis 3D.
- ❖ Saya memiliki ide aplikasi dalam pikiran untuk VR.
- ❖ Saya ingin mempelajari cara membangun VR sebelum memikirkan ide aplikasi.

Jika Anda memiliki ide tentang proyek yang ingin Anda bangun, ini merupakan keuntungan karena Anda kemudian dapat menargetkan tonggak tertentu sambil mempelajari pelajaran Anda sebagai dasar untuk produk yang lengkap. Atau, jika Anda ingin mempelajari semua alat terlebih dahulu, ini akan membantu dalam menyusun ide aplikasi pertama Anda karena Anda akan memahami fitur dan kendala VR sebelum berkomitmen pada ide yang tidak layak. Apa pun itu, tidak ada cara kerja yang salah atau lebih baik, kecuali menggunakan sistem yang paling sesuai untuk Anda.

Dalam kasus ini, kami menguraikan blok-blok penyusun untuk membangun pengalaman imersif dengan tujuan akhir yang direncanakan. Bab ini lebih berfokus pada pemikiran tingkat tinggi. Kami membahas mengapa lintas platform penting, memberikan pengantar tentang mesin permainan, dan menawarkan beberapa strategi untuk membangun kerangka kerja lintas platform.

6.1 MENGAPA LINTAS PLATFORM?

Menangani lintas platform dapat dilihat sebagai topik tingkat lanjut, tetapi sebenarnya ini adalah solusi desain mendasar yang memengaruhi seluruh arsitektur produk apa pun yang ingin menjadi imersif. Pada masa-masa awal VR dan AR ini, semuanya masih eksperimental—desain headset, desain pengontrol, skema aksesori, dan sebagainya. Kami dapat mengidentifikasi lebih dari 16 kombinasi headset dan pengontrol yang berbeda untuk VR dan AR, dengan lebih banyak lagi yang muncul setiap beberapa bulan. Hingga konsistensi tercapai, konten akan terpecah-pecah di seluruh ekosistem. Dengan VR dan AR, kami berada di ujung spektrum kompatibilitas. Di sisi dekat, ada perangkat televisi tradisional dan perangkat media seluler, yang berpotensi memutar semua konten layar datar terlepas dari produsennya. Di

tengah, kita memiliki konsol gim video, di mana setiap perangkat berbagi beberapa konten (misalnya, Fortnite, Minecraft) sekaligus memiliki beberapa konten eksklusif (misalnya, Uncharted di PlayStation, Mario di Nintendo, Halo di Xbox). Kemudian, di ujung terjauh, kita memiliki VR dan AR, yang kontennya berpusat pada platform pada masa-masa 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. Kategori ini dapat dibagi menjadi berikut:

- ✓ Headset yang ditambahkan dengan satu atau beberapa 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 kemudian pada tahun yang sama, menawarkan setidaknya tiga skema kontrol alternatif di luar mouse dan keyboard tradisional. Sementara Oculus Go hadir standar dengan pengontrol, platform Samsung GearVR dikirimkan sebagai headset drop-in dengan dukungan gamepad sebelum merilis opsi pengontrol yang dilacak setahun kemudian. Akibatnya, tidak dapat dijamin bahwa pemilik platform juga akan memiliki salah satu input yang awalnya tidak dikirimkan dengan platform tersebut. Untuk menjangkau audiens terluas, kita harus membahas input inti yang tersedia untuk setiap konsumen. Saat membangun kontrol input adaptif, ini akan membantu meningkatkan skala produk antara kontrol yang paling tersedia hingga yang paling canggih.

Hingga VR dan AR bergerak mendekati tengah dalam hal kompatibilitas, pesan yang disampaikan kepada konsumen umum bahwa "VR sudah siap" akan terbatas, karena konsumen tersebut tidak dapat membeli satu headset dan sekaligus mendapatkan sebagian besar konten terbaik yang tersedia. Hal yang sama berlaku untuk pemasaran: laba atas investasi untuk kampanye VR terbatas karena tingginya biaya pengembangan konten ditambah dengan terbatasnya basis audiens satu headset.

Meskipun adopsi umum merupakan 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 untuk membuat aplikasi bernilai tinggi untuk audiens yang lebih kecil. Ini dapat berhasil saat mengembangkan untuk pasar perusahaan atau di lingkungan yang sangat terkendali seperti arena permainan VR atau instalasi. Dalam semua kasus ini, berfokus hanya pada satu platform mungkin baik-baik saja, dan dengan demikian bab ini mungkin tidak memiliki nilai sebanyak itu.

Namun, mendesain untuk portabilitas memiliki keuntungan tambahan untuk menghindari ketergantungan pada vendor. Dengan mempertahankan desain yang sebagian besar bersifat platform agnostik, aplikasi akan dapat beradaptasi dengan platform perangkat keras alternatif dengan sangat cepat dengan biaya pengembangan yang lebih rendah. Hal ini

dapat menguntungkan ketika platform yang lebih cocok diluncurkan serta ketika peluang bisnis yang lebih baik mungkin muncul dengan perusahaan VR dan AR yang bersaing. Saat ini juga belum pasti platform perangkat keras mana yang akan bertahan dari gelombang industri VR dan AR yang terus berubah, jadi metode ini mengurangi risiko terikat pada satu platform yang mungkin akan mengambil pangsa pasar terkecil di masa mendatang.

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

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

6.2 PERAN MESIN PERMAINAN

Meskipun aplikasi VR dapat dikembangkan menggunakan C++, saat ini mesin permainan sangat populer untuk segala hal mulai dari membuat prototipe cepat sebuah konsep permainan dalam hitungan jam hingga membangun produk triple-A yang dirilis sepenuhnya. Apa itu mesin permainan? Ini adalah istilah industri untuk serangkaian perangkat lunak yang menerima serangkaian masukan (mouse, keyboard, layar sentuh, dll.), menerapkan logika padanya (misalnya, menggerakkan karakter, melompat, menembakkan senjata), 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 permainan terutama, dengan manfaat utamanya adalah banyak matematika kompleks dan logika kode tingkat rendah akan diprogram sebelumnya ke dalam sistem. Selain itu, mesin permainan juga pada akhirnya akan menjadi kompatibel dengan multiplatform, membangun serangkaian desain kode umum oleh pengembang sambil dapat digunakan untuk target platform yang berbeda. Keuntungan utama dari mesin permainan yang muncul adalah kemampuan untuk menargetkan semua jenis arsitektur sistem tanpa harus mempelajari banyak bahasa pemrograman dan API yang bergantung pada platform. Pengembang permainan dapat bekerja dengan bebas di mesin permainan yang mereka pilih untuk dipelajari dan kemudian menerapkannya ke sistem baru saat dipasarkan.

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

Meskipun Unreal Engine memiliki serangkaian manfaatnya sendiri, dalam bab ini kami fokus pada Unity. Awalnya dirilis pada tahun 2005, Unity telah membantu banyak pengembang di seluruh dunia untuk memulai membangun permainan tiga dimensi, termasuk semuanya mulai dari seluler hingga konsol hingga desktop. Unity telah menjadi tulang punggung pengembangan 3D bagi banyak pengembang sekaligus membina komunitas yang luar biasa selama bertahun-tahun karena perusahaan terus mengembangkan produknya untuk memenuhi kebutuhan fitur VR dan AR baru yang terus berubah.

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

Aplikasi mesin gim 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-contohnya, kami sarankan Anda mengunduh Unity IDE. Rilis publik Unity apa pun saat ini dapat digunakan, tetapi untuk tujuan kompatibilitas, kami menggunakan Unity 2018.1f1. Unity sangat tangguh dan fleksibel. Berbagai alat bawaan dan plug-in eksternalnya telah mengalami peningkatan signifikan dari waktu ke waktu, menanggapi masukan pengembang untuk mempertahankan komunitas yang kuat.

Di permukaan, Unity menangani pengembangan dan penerapan lintas platform, tetapi memanfaatkan fitur setiap platform memerlukan kemahiran yang melampaui pemetaan ulang kontrol dasar. Unity menangani banyak hal untuk pengembang dalam hal pekerjaan berat, tetapi masih ada latihan yang harus diselesaikan pengembang, yang akan kami bahas lebih lanjut di bagian ini. Belajar Unity dari awal berada di luar cakupan buku ini, tetapi banyak tutorial tersedia daring, langsung dari situs web Unity serta berbagai sumber daya yang tersedia di YouTube, Udemy, dan PluralSight, antara lain.

6.3 MEMAHAMI GRAFIK 3D

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

Kamera Virtual

Kamera virtual merupakan fondasi inti VR. Secara tradisional, di dunia nyata, kita

mengenai kamera sebagai perangkat mekanis atau elektronik yang mengambil gambar dan video. Dalam obrolan video antara dua ponsel, masing-masing orang memegang ponsel di dunia 3D nyata yang mentransmisikan apa yang dilihat ponsel, secara real time, ke perangkat layar datar milik orang lain. Kamera virtual di dalam mesin game Unity dapat dianggap dengan cara yang sama, tetapi alih-alih kamera menjadi perangkat nyata yang ditempatkan di dunia 3D nyata, kamera ditempatkan di lingkungan 3D virtual. Dengan demikian, umpan langsung disediakan 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.

Dalam VR, ada beberapa hal yang berubah di sini. Pertama, kamera menjadi melekat pada kepala pengguna, jadi alih-alih menggunakan tangan untuk mengubah sudut pandang, Anda cukup menggerakkan kepala. Kedua, tampilan ditampilkan dua kali: sekali untuk setiap mata, masing-masing diberi layarnya sendiri, dengan posisi kamera virtual masing-masing mata sedikit bergeser dari tengah sehingga penonton akan merasakan efek paralaks stereoskopik. Semua ini ditangani untuk penonton di Unity dengan sakelar sederhana, tetapi yang sebenarnya berarti bahwa mengembangkan untuk VR berarti mengembangkan untuk 3D terlebih dahulu. VR dapat dilakukan setelahnya, yang sangat penting untuk merencanakan strategi pembelian perangkat keras Anda. Untuk memulai, yang benar-benar Anda perlukan hanyalah laptop (atau desktop jika Anda bukan tipe pekerja keras yang portabel). Anda dapat mempelajari mekanisme dasar Unity dan menciptakan dunia 3D dengan input keyboard untuk menggerakkan kamera, lalu memasang VR di bagian akhir untuk masuk ke dalam pengalaman. Dari sana, Anda dapat beradaptasi sesuai keinginan.

Hampir sama halnya dengan AR, tentu saja dengan beberapa perbedaan. Pertama, dengan AR seluler yang menggunakan ponsel, kamera ponsel dipegang di tangan Anda, bukan dipasang di kepala Anda. Namun, umpan kamera virtual disajikan ke layar datar ponsel (tanpa pemisahan optik untuk VR) dengan cara yang hampir sama. Memahami kamera sangatlah penting karena perubahan kamera dari posisi yang relatif tetap dalam gim video menjadi dapat digerakkan sepenuhnya dalam VR merupakan perubahan besar yang memengaruhi cara gim dirancang dan dioptimalkan. Pertama, frame rate grafis harus berkinerja tinggi untuk mengurangi efek buruk pada otak, dan, kedua, trik dan teknik kinerja tertentu (seperti area dunia yang belum selesai yang tidak mungkin dilihat pengguna) tidak tersedia jika penonton memiliki kebebasan penuh untuk menjelajahi area tersebut. Tidak semua perangkat keras VR identik, bahkan pada 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, merujuk pada variasi gerakan yang tersedia untuk setiap objek yang dilacak. Objek yang dilacak adalah objek yang bergerak di ruang fisik dan melaporkan informasi posisi dan/atau rotasinya ke mesin permainan. Hal ini dilakukan melalui kombinasi data sensor, tetapi yang terpenting, posisi dan/atau orientasi objek yang dilacak di

dunia nyata dapat direpresentasikan di dunia virtual, yang menyinkronkan dunia nyata dan virtual.

Headset realitas virtual tersedia dalam dua jenis: tiga derajat kebebasan (3DOF) dan enam derajat kebebasan (6DOF). Jika Anda pernah 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 kabel yang ditenagai oleh telepon, kemungkinan besar Anda menggunakan headset 3DOF. Jika Anda dapat berbalik dan melihat ke belakang dalam VR, tetapi Anda tidak dapat merasakan berjalan ke arah objek di kejauhan, Anda mungkin menggunakan headset 3DOF. Hal yang sama berlaku jika Anda berjongkok dan pandangan Anda tidak berubah. Ini karena pelacakan 3DOF berarti bahwa rotasi objek yang dilacak dilaporkan ke perangkat lunak, tetapi posisinya tidak. Dengan pelacakan rotasi, mesin permainan akan memiliki informasi tentang yaw, pitch, dan roll headset (rotasi sepanjang sumbu x, y, dan z, tidak harus berturut-turut). Ini umumnya dikenal sebagai pelacakan 3DOF. Google Cardboard, Samsung GearVR, Google Daydream View, dan Oculus Go termasuk dalam kategori ini, dan pelacakan 3DOF dapat dilakukan menggunakan sensor akselerometer, giroskop, dan magnetometer internal yang ada di sebagian besar chipset ponsel.

3DOF yang tersisa dalam headset 6DOF adalah posisi x, y, dan z sepanjang sumbu x, y, dan z (masing-masing). Karena pengalaman 3DOF tidak dapat menggerakkan kamera hanya dengan kepala Anda, kamera virtual bergerak secara otomatis (seperti dalam pengalaman roller coaster) atau beberapa bentuk gerakan diimplementasikan melalui input kontrol. Ini dikenal sebagai lokomosi, dan kami membahasnya lebih lanjut di bagian berikutnya. Karena ketidaknyamanan otak dan tubuh, lokomosi memiliki sejumlah solusi, termasuk teleportasi dan kamera kabur untuk mengurangi efek samping mabuk perjalanan. Pengalaman 6DOF terasa lebih alami karena hubungan 1:1 yang lengkap antara menggerakkan kepala Anda ke segala arah dan mencocokkan pengalaman visual. Dalam pengalaman ini, Anda dapat berjongkok di lantai untuk mengambil sesuatu, berjinjit untuk mendapatkan sudut pandang yang lebih baik, atau melangkah ke samping untuk menangkap bola.

Namun, pelacakan posisi adalah masalah rumit yang memerlukan 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 pelacakan penuh memiliki kelebihan, hal itu juga dapat mengorbankan kebebasan bergerak. Saat ini, mayoritas solusi 6DOF dihubungkan ke komputer, dengan kabel yang menjuntai berfungsi sebagai tali pengikat yang dapat menyebabkan bahaya tersandung atau dapat terlilit atau tercabut dari mesin jika tidak dikelola dengan benar. Kemajuan 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 merupakan pengalaman 3DOF. Pengendali merupakan faktor lain untuk realitas virtual. Pengendali dapat dilacak atau tidak dilacak, dan, seperti headset, pelacakan tersedia dalam variasi 3DOF dan 6DOF. Tabel 6.1 mencantumkan platform VR utama yang tersedia pada pertengahan tahun 2018.

Tabel 6.1 Platform VR yang tersedia

Platform VR 3DOF	Metode masukan
Mobile VR: Cardboard/phone drop-in	Menatap dan menahan posisi
VR Seluler: Pemasangan pada karton/ponsel	Tidak terlacak
Oculus Mobile dengan panel sentuh headset Panel sentuh headset	tidak terlacak. Panel sentuh yang dapat diklik + 1 tombol
Oculus Mobile dengan Gamepad	Gamepad tidak terlacak. Panel arah digital, pengontrol analog, masing-masing 6 tombol muka + 4 tombol pemicu
Oculus Mobile dengan pengontrol	1 x pengontrol tangan 3DOF dengan panel sentuh yang dapat diklik + 2 tombol
Tampilan Google Daydream	1 x pengontrol 3DOF dengan panel sentuh yang dapat diklik + 1 tombol
Platform VR 6DOF	Metode input
Google Daydream: Mirage Solo	1 x pengontrol 3DOF dengan panel sentuh yang dapat diklik + 1 tombol
HTC Vive Focus	1 x pengontrol 3DOF dengan touchpad yang dapat diklik + 2 tombol
Oculus Rift dengan pengontrol Xbox	Gamepad, tidak terlacak. Panel arah digital, pengontrol analog, masing-masing 6 tombol muka + 4 tombol pemicu
Oculus Rift dengan remote	Remote, tidak terlacak. Papan penunjuk arah + 1 tombol
Oculus Rift dengan pengontrol Sentuh	Pengontrol 2 x 6DOF. Joystick yang dapat diklik, masing-masing 2 muka + pemicu variabel + pegangan
HTC Vive	Pengontrol 2 x 6DOF. Panel sentuh yang dapat diklik + 2 tombol + pemicu variabel, masing-masing
Headset realitas campuran Microsoft	Pengontrol 2 x 6DOF. Panel sentuh yang dapat diklik + 2 tombol + pemicu variabel, masing-masing
Platform AR 6DOF	
Microsoft Hololens	2 x posisi tangan 3DOF, mendeteksi tangan + ketukan tangan saja
Microsoft Hololens	1 x clicker dengan 1 tombol saja
AR Seluler: iPhone / Android	Layar sentuh

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 pengembangnya yang tertutup lebih sulit untuk dimasuki, dan semua platform dalam Tabel 6.1 dapat dibeli dan dikembangkan oleh rata-rata orang dalam hitungan minggu atau bahkan hari.) Menekankan kembali bahwa meskipun lintas platform merupakan tantangan teknis, ini merupakan tantangan desain yang bahkan lebih sulit. Ada alasan mengapa sebagian besar konten video 360 derajat muncul secara konsisten di setiap platform, dan itu karena sistem interaksi "duduk dan menatap, melihat-lihat" dapat dikaitkan dengan hampir setiap platform yang disebutkan di atas.

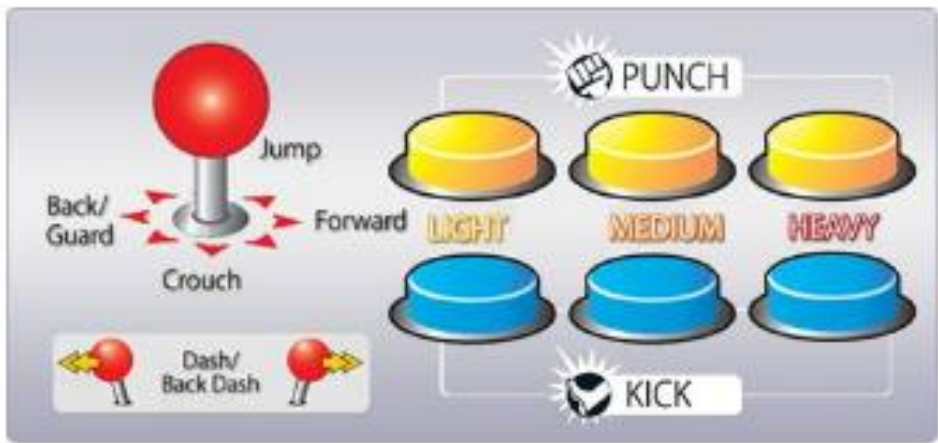
Melihat Tabel 6.1, tidak sulit untuk membayangkan bahwa mendukung 16 headset berbeda ditambah kombinasi input untuk VR dan AR dapat tampak cukup menakutkan.

Namun, ini adalah tantangan yang dipecahkan setiap hari oleh siapa pun yang melakukan perencanaan ruang di dunia nyata: menampung anak-anak, kereta dorong, kursi roda, orang-orang dengan ukuran besar dan kecil, pendek dan tinggi, gangguan pendengaran, gangguan penglihatan, dan sebagainya. Dunia nyata terus-menerus direkayasa untuk mengakomodasi audiens sebanyak mungkin dengan jalur kursi roda, braille, subtitel, dan banyak lagi. Dengan batasan yang jelas diterapkan jika perlu ("Anda harus setinggi ini untuk naik"), sudah ada istilah standar industri untuk ini: aksesibilitas. Prinsip desain aksesibilitas, jika diterapkan pada VR, dapat diadaptasi dengan indah.

Artinya, pertimbangkan pengalaman keseluruhan yang ingin Anda sampaikan. Apa yang Anda inginkan agar pemain lakukan di lingkungan virtual ini? Apakah ini pengalaman pasif atau aktif? Apa yang akan mereka anggap memuaskan? Jika pengalaman yang Anda rancang murni merupakan etalase untuk perangkat keras seperti pelacakan tangan atau antarmuka pengontrol baru, penekanannya pasti akan berpusat pada platform dan mungkin tidak dapat dipindahkan ke sistem lain. Menjaga konsep tingkat tinggi tetap bersifat eksperiensial daripada berbasis interaksi memungkinkan kebebasan yang lebih besar dalam desain pengalaman, dan sebagai tambahan fungsi pengontrol dapat diterapkan segera setelah pembangunan dunia diselesaikan.

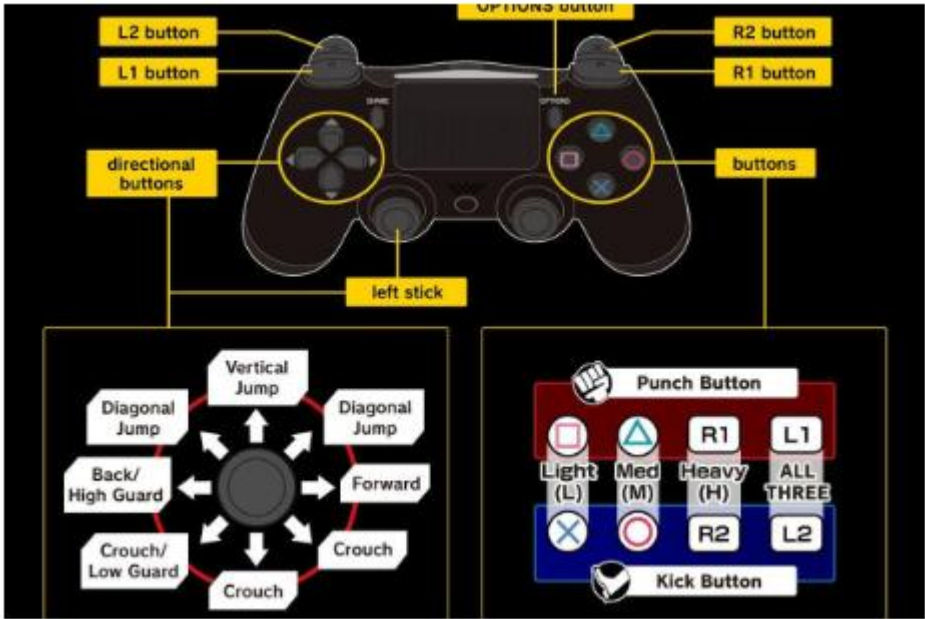
6.4 PORTABILITAS DARI DESAIN GIM VIDEO

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 tepat cenderung mengalami kendala karena tidak dirancang untuk kelas bawah selama desain produk awal. Selain masalah grafis, skema kontrol diganti dan diganti dalam upaya untuk mensimulasikan versi kelas atas dari pengalaman input, yang benar-benar dapat terlihat. Beberapa gim video telah beralih dari rilis konsol rumah asli ke format gim portabel. Salah satu contohnya adalah gim *Street Fighter II* asli, yang tata letak enam tombolnya 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 perkembangan rendah/sedang/tinggi dari kiri ke kanan.



Gambar 6.1 Pengalaman input arcade enam tombol asli untuk Street Fighter II

Setelah dirilis untuk PlayStation asli, tata letak empat tombol pada gamepad menggunakan dua tombol bahu untuk menggantikan dua tombol muka 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 berjalan 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.

Hal ini diatasi dalam iterasi waralaba berikutnya, seperti Marvel vs. Capcom 2. Game ini dirancang agar lebih ramah port, menghapus serangan sedang dan mengurangi tombol serangan utama menjadi ringan dan berat, dengan dua tombol yang tersisa dalam rangkaian gerakan bantuan mitra baru, sehingga berada 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 dapat ditempatkan sebagai dua tombol kanan dari kontrol enam tombol, seperti yang ditunjukkan pada Gambar 6-3, atau dapat dipetakan ke tombol bahu pengontrol PlayStation.



Gambar 6.3. Konfigurasi empat tombol plus dua tombol yang dapat diskalakan untuk Marvel versus Capcom 2 yang dapat disesuaikan dengan tata letak kontroler 3 x 2 dan 4 x 4 + 2

Hal utama yang perlu diingat: pikirkan tentang pengalaman tersebut sambil mempertimbangkan beberapa opsi adaptasi yang tersedia. Apa yang perlu terjadi dalam pengalaman tersebut, lalu bagaimana kita mengakomodasi skema input terbanyak untuk dimasukkan ke dalam sistem kontrol tersebut? Jawab pertanyaan tersebut untuk mengambil langkah pertama menuju desain input yang dapat diskalakan.

6.5 MENYEDERHANAKAN INPUT KONTROLER

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 tersebut adalah untuk mengeksplorasi keajaiban VR melalui serangkaian tugas yang sudah dikenal yang dilakukan dalam VR dengan pendekatan yang aneh dan menyenangkan sebagai elemen tambahan yang menyenangkan dan menyenangkan.

Para pengembang di Owlchemy Labs telah menawarkan beberapa wawasan dan ceramah tentang pelajaran yang dipetik di berbagai konferensi dan daring, dan salah satu poin utama yang mereka sampaikan adalah bahwa VR memungkinkan penghapusan banyak masukan pengontrol tradisional karena tindakan tersebut kini digantikan dengan menggunakan tubuh kita yang sebenarnya di dunia nyata. Sementara gim video tradisional mengendalikan gerakan dan tindakan protagonis dengan menggunakan satu atau bahkan dua joystick arah, semua gerakan tersebut kini dapat ditangani dengan menoleh dan benar-benar berjalan. Berjongkok atau melompat kini dapat dilakukan langsung oleh tubuh alih-alih menggunakan tombol pada pengontrol. Hal ini membebaskan skema kontrol rumit yang secara tradisional ditemukan pada gim konsol dan memungkinkan pengontrol tangan untuk fokus

hanya pada manipulasi item di dunia.

Dengan menetapkan satu tombol pemicu untuk menangani pengambilan, lemparan, dan manipulasi item, Job Simulator menjadi salah satu gim tercepat untuk dimainkan tanpa perlu persiapan apa pun. Yang lebih penting, hal ini berlaku untuk berbagai usia yang telah mencobanya, mulai dari usia 4 tahun hingga 80 tahun ke atas. Cara mereka mencapai hal ini adalah melalui satu pendekatan yang sangat jenius: menempatkan interaksi ke dalam dunia dan memungkinkan masukan tersebut dimanipulasi oleh tangan pemain. Menekan tombol, memutar kenop, menarik pegangan jaring lemari terbuka, dan melempar objek semuanya dilakukan dengan mensimulasikan dunia nyata, yang memungkinkan naluri alami seseorang untuk membimbing mereka melalui pengalaman tersebut. Alih-alih menetapkan serangkaian kontrol yang rumit ke pengontrol tangan, mereka menjadi perpanjangan dari tangan pemain itu sendiri, dan antarmuka pengguna untuk belajar semuanya ada di dalam dunia.

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

Langkah Pengembangan 1: Mendesain Antarmuka Dasar

Anda dapat merancang objek dalam VR dengan memikirkan cara kerja objek di dunia nyata. Apakah ada tombol yang terpasang padanya seperti remote control? Apakah ia terbuka dan tertutup seperti kotak? Apakah ia memiliki keberadaan fisik atau hanya penanda simbolis (seperti bola lampu)? Dapatkah ia diambil atau dipasang pada objek lain?

Pertimbangkan semua cara seseorang berinteraksi dengan suatu objek; misalnya, sakelar lampu. Saat ditekan, sakelar lampu berubah posisi dan tindakan dilakukan, dalam hal ini lampu dinyalakan atau dimatikan. Sakelar lampu tersebut dapat dimanipulasi dengan berbagai cara: sakelar dapat dinyalakan langsung dengan tangan di dekat sakelar lampu, atau Anda dapat mengambil tongkat, yang akan berfungsi sebagai perpanjangan tangan, sehingga menekan sakelar dari jarak jauh. Atau, daya bohlam lampu juga dapat dikontrol oleh sistem rumah pintar, sehingga gaya 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 pengontrol tangan 6DOF, Anda dapat langsung meletakkan tangan Anda di sebelah sakelar dan mengklik tombol untuk membalikinya.

Misalkan pengontrol itu 3DOF dan tidak dapat memindahkan posisinya di ruang angkasa tetapi dapat berputar. Penunjuk laser yang terpasang pada pengontrol dapat bertindak seperti ekstensi tongkat fisik, dan ketika laser menunjuk ke sakelar, mengklik tombol dapat mengaktifkan sakelar. Tanpa pengontrol, seperti pada platform kepala-tracked 3DOF seperti perangkat Cardboard, laser dapat dipasang ke kepala, dan melihat langsung ke sakelar dan mengetuk tombol dapat mengenai 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 beralih sendiri. Terakhir, pada perangkat seluler, tombol tersebut dapat disentuh langsung dari layar sentuh atau bahkan dapat ditarik ke atas dalam menu di layar dan dikontrol 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 yang merespons perintah sederhana seperti "nyalakan," "matikan," dan "alihkan." Anda dapat mengembangkannya lebih jauh. Sakelar lampu kita memiliki dua status: hidup dan mati. Bagaimana jika ada lebih banyak status, seperti intensitas cahaya dengan rentang 0 hingga 100? Sekarang, kita memerlukan cara untuk mengendalikan input variabel. Di dunia nyata, kita menangani ini dengan sakelar peredup. Alih-alih menjadi sakelar yang dibalik, peredup sering kali diimplementasikan sebagai bilah geser pada panel. Dengan demikian, sakelar lampu memerlukan kontrol tombol tambahan, tetapi yang ini akan memiliki fitur tambahan yang dapat diangkat dan dipindahkan.

Kemudian, saat tombol diklik, alih-alih sakelar yang mengaktifkan/menonaktifkan bohlam lampu, peredup akan mengaktifkan status terhubung ke kontrol atau terputus. Dalam keadaan terhubung, gerakan pengontrol (baik itu pengontrol tangan, penunjuk laser, atau kepala untuk pemasangan tatapan) kemudian akan menggerakkan slider dan mengubah nilai intensitas bola lampu secara real time hingga terputus. Dengan tombol atau layar sentuh, tindakan "turun/tekan" kemungkinan akan menghubungkan objek dengan tindakan "naik/lepas", memutuskan hubungan objek. Dalam pendekatan tatapan-dan-tatapan, setiap tindakan tatapan-dan-tatapan harus dilakukan sepenuhnya sekali untuk menghubungkan dan sekali lagi untuk memutuskan hubungan.

Tanpa pengontrol, Anda dapat menggunakan kepala untuk mengarahkannya dengan melihat sakelar, dan jika tidak ada pengontrol, tatapan yang lama dapat mengaktifkannya. Pada akhirnya sakelar lampu akan memiliki antarmuka jenis "tekan tombol" yang akan diaktifkan oleh sejumlah skema kontrol agar dapat berfungsi. Kami sekarang telah mendefinisikan dua antarmuka, satu untuk pemilihan dan satu untuk meraih, dan kami telah menunjukkan cara kerjanya di semua jenis masukan platform. Kita dapat memperluasnya ke konsep yang lebih kompleks seperti berikut:

1. Cara memasang objek di Unity: transformasi langsung, sambungan tetap, dan gaya fisika
2. Manipulasi objek dengan dua tangan untuk gerakan dan skala
3. Manipulasi objek sekunder, seperti mengisi ulang senjata atau membuka gabus botol
4. Membatasi gerakan objek yang terpasang, seperti memutar kenop pintu

Sayangnya, pembahasan mendalam tentang hal ini berada di luar cakupan bab ini. Namun, Anda dapat melihat bagaimana ada banyak cara untuk mengatasi tantangan antarmuka.

Dengan dua properti antarmuka dasar ini, Anda dapat mengisi lingkungan dunia dengan objek eksplorasi yang dapat berinteraksi dengannya, apa pun platform yang digunakan. Bagian terbaiknya adalah semua pekerjaan itu dapat dilakukan tanpa harus meninggalkan komputer dan mengenakan headset. Setelah siap, kita mulai menyiapkan cara

menangani integrasi platform.

Langkah Pengembangan 2: Integrasi Platform

Integrasi platform melibatkan pengambilan aplikasi dan pemasangan perangkat keras ke dalamnya. Hal ini dapat dilakukan secara langsung, tetapi desain lintas platform sedikit memperlambat kita karena kita menyiapkan lapisan abstraksi yang solid terlebih dahulu. Jika Anda telah mencoba membangun prototipe pada platform VR, bagian ini merupakan 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 mengintegrasikan dengan platform.

- Memasang kepala
- Memasang input kontrol

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

Memasang input kontrol bisa jauh lebih rumit berdasarkan berbagai sistem interaksi. Misalnya, sistem HTC Vive dapat menggunakan OpenVR atau plug-in SteamVR milik Valve, sedangkan kontroler Oculus Touch menggunakan OVRPlugin dari Oculus. Google memiliki kode sendiri untuk kontroler Daydream, yang meskipun memiliki konfigurasi yang mirip dengan kontroler Oculus Go, menggunakan skrip manajer input yang berbeda. Contoh yang diberikan untuk setiap platform cenderung bekerja secara langsung dengan kode SDK kustom untuk melampirkan fungsionalitas kontroler, yang bagus untuk pembelajaran, tetapi untuk tujuan kita, kita akan mengintegrasikan pada tingkat yang berbeda.

Untuk lebih jelasnya, fungsionalitas kontroler hadir dalam dua bagian: melacak posisi dan/atau orientasi objek, dan memantau tombol atau input sentuh. Melacak posisi/rotasi objek mudah dilakukan di Unity, dan kode kita akan dilampirkan ke objek virtual yang memetakan ke objek dunia nyata 1:1. Input tombol/sentuhanlah yang kita minati untuk latihan ini. Mari kita lihat dulu seperti apa jalur kode “Tekan tombol menu untuk membuka menu” pada contoh 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 melakukannya saja? Implementasi kode aktual dari baris Apakah pengguna mengklik tombol menu kontroler? kemungkinan besar bergantung pada platform dan dibuat berdasarkan API kontroler SteamVR atau Oculus Rift.

Kode ini menetapkan kontrol yang ditentukan ke tombol tertentu dalam kode, sehingga tidak dapat disesuaikan (misalkan pengguna individu ingin menggunakan tombol pemicu untuk membuka menu) dan tidak memiliki portabilitas (kontroler Daydream tidak akan memahami panggilan API.) Selain itu, kode ini ditulis secara khusus untuk menjalankan satu fungsi tunggal yaitu membuka menu. Jika fungsionalitas baru akan ditambahkan, fungsi tersebut juga perlu dikodekan secara kaku dalam urutan blok kode yang sama di sini.

Atau, mari kita abstrakkan 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 masukan kontrol. Ini akan menjadi objek yang mendefinisikan serangkaian fungsi yang dapat dilampirkan ke pengontrol ini. Dalam kasus ini, fungsi "buka menu" dapat diatur, tidak diatur, atau diubah. Kemudian, kami akan memetakan satu atau beberapa kontrol ke sana, dan kode ini harus berupa kode yang bergantung pada platform, seperti memetakan `button_menu` pada pengontrol SteamVR ke fungsi `open_menu`. Kemudian, alur alur jika ada kontrol yang dipetakan akan mencari objek `button_menu` pengontrol SteamVR dan memeriksa apakah objek tersebut dalam keadaan yang penting untuk fungsi ini. Keadaan ini dapat berupa "tombol turun," "tombol naik," atau "klik tombol." Kemudian, jika keadaan tombol itu mengirimkan keberhasilan, kontrol akan diaktifkan untuk membuka atau menutup menu.

Ada empat skrip yang digunakan di sini. Salah satunya adalah pemetaan pengontrol yang memeriksa interaksi pada setiap bingkai. Sebut ini `ControllerModule`. Ini sederhana dan menyediakan logika untuk menangani frame loop dan sepenuhnya independen terhadap platform. Kelas kedua adalah kelas skema kontrol; sebut saja `ControlScheme`. Ini mendefinisikan fungsionalitas khusus aplikasi yang tersedia, dan akan dibangun per aplikasi sesuai kebutuhan. Kelas ketiga adalah skema pemetaan control misalnya, `MappingMenu ButtonToViveController` yang akan dibuat sekali 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, tanpa memiliki pengetahuan tentang fungsionalitas aplikasi dan hanya berfungsi sebagai antarmuka ke pengontrol 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-kelas lainnya akan memiliki kelas akar fungsionalitas dasar, dengan kelas-kelas anak yang diimplementasikan sesuai kebutuhan.

ControlScheme akan menjadi implementasi konkret yang ditulis sekali per skema interaksi untuk sebuah aplikasi. Beberapa contoh selain interact adalah grab, draw, dan select. Semua ini dapat memiliki mode respons yang berbeda untuk kontroler. Grab akan menjadi pick up, draw akan mengeluarkan karya seni ke dunia, dan select dapat memilih objek untuk dimanipulasi.

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

Terakhir, implementasi kontroler akan ditulis sekali per jenis kontroler pada setiap platform. Ini akan menangani kode khusus platform untuk memantau pemacu tombol, input analog, dan sebagainya. Jadi, bagaimana semua pekerjaan ekstra ini membantu kita? Sekarang mari kita lampirkan fungsi "interaksi" ke pengontrol. Berikut diagram alir yang 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 melekat pada objek yang melayang. Skrip ini sekarang dapat mengaktifkan sakelar lampu jika tombol yang sesuai dipilih, atau mengaktifkan menu jika tidak. Perhatikan di sini bahwa peta kendali dapat menetapkan "berinteraksi" ke pemacu sementara "buka menu" tetap melekat pada tombol menu.

Mari kita lanjutkan dan tambahkan kemampuan untuk mengambil objek. Perhatikan bahwa

jika suatu objek diambil, kita mungkin ingin melakukan sesuatu dengan objek itu seperti melemparnya, memutarnya, dan seterusnya, sehingga logika untuk itu 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?
```

Dapatkah Anda melihat polanya? Kita dapat mengulangi skema kontrol dan beralih ke fungsionalitas kontroler dasar jika ada. Ini dapat diskalakan untuk membuat fungsionalitas kontroler.

Selain itu, mari kita lihat cara mengadaptasi kontroler yang berbeda untuk ini. Jika kita ingin memindahkan aplikasi ini ke Daydream, kita memerlukan skrip DaydreamController baru untuk menangani implementasi kontroler dan skrip MappingMenuButtonToDay dream baru untuk mengganti pemetaan lainnya. Selain itu, kita sudah selesai.

Contoh dunia nyata

Memindahkan prototipe dari Daydream ke Oculus Go menarik karena satu-satunya perbedaan dalam dua jenis kontroler tampaknya adalah tombol pemicu. Jadi, satu prototipe memerlukan dua tombol yang dapat disesuaikan: satu untuk melakukan fungsi utama dan satu untuk melakukan fungsi sekunder untuk mengubah mode fungsi utama. Kemudian, kita dapat beralih dari mode pengecatan ke mode pemilihan dengan satu tombol sambil menggunakan tombol utama untuk benar-benar melakukan pengecatan atau pemilihan.

Pada pengontrol Go, pemicu adalah tombol yang paling tepat untuk menangani fungsi utama, sehingga klik pada panel sentuh dapat digunakan untuk melakukan pertukaran mode sekunder. Namun, pada pengontrol Daydream, tidak ada pemicu, sehingga input utamanya adalah klik pada panel sentuh, yang merupakan input sekunder pada pengontrol Go! Tidak masuk akal untuk menggunakan pengontrol Go sebagai input sekunder, jadi pengontrol Daydream menggunakan tombol aplikasi untuk input sekunder guna mengubah mode input utama. Meskipun hal ini membuat platform berbeda, mengedit file pemetaan untuk bermain dengan opsi yang berbeda merupakan hal yang mudah.

Ringkasan

Teori yang diberikan dimaksudkan untuk membantu memulai pemikiran tingkat tinggi dalam desain VR dan AR dan untuk membantu memaksimalkan informasi teknis yang menyertainya. Dalam bab ini, kami menjelaskan dasar-dasar tentang mesin permainan dan kamera 3D, 3DOF versus 6DOF, dan mengapa pendekatan desain harus dipertimbangkan pada tingkat tinggi secara holistik berdasarkan tujuan pengalaman.

Konsep abstraksi lintas platform akan membantu 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 membantu:

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

Dalam bab berikutnya, Vasanth memberikan contoh yang lebih konkret tentang perancangan untuk VR dan AR, termasuk teknik penggerak serta informasi lebih lanjut tentang jenis kontrol yang tersedia. Sebagai mantan kapitalis ventura di bidang ini, 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 terelakkan. Yang tidak diketahui adalah berapa lama waktu yang dibutuhkan untuk mencapainya, baik 2 atau 20 tahun. Namun, setiap upaya yang dapat kita lakukan untuk mempercepat pertumbuhan itu akan membantu kita mencapai masa depan di mana kita secara alami berbagi keajaiban VR dan AR dengan lebih banyak teman dan keluarga kita. Itulah masa depan yang saya nanti-nantikan, dan saya berharap bersama-sama kita dapat mencapainya segera.

BAB 7

VIRTUAL REALITY TOOLKIT

VRTK adalah basis kode sumber terbuka yang memungkinkan pengguna untuk menarik dan melepas fungsionalitas. Dengan kemampuan untuk menarik dan melepas Aset ke dalam Unity 3D, dengan beberapa konfigurasi, mereka dapat segera membuka contoh adegan dengan mekanika permainan penting yang berfungsi penuh dan siap pakai seperti pergerakan, navmeshes, berbagai antarmuka pengguna, dan interaksi fisik yang dapat Anda gunakan untuk mulai membangun permainan Anda. Dengan manfaatnya sebagai sumber terbuka, siapa pun dapat segera mengurangi waktu penyiapan dan segera mulai menyesuaikan aset dan kode sumber untuk mewujudkan ide mereka di Unity untuk pembuatan prototipe cepat paling tidak.

Manfaat utama dari perangkat ini adalah bahwa perangkat ini satu-satunya yang dapat dengan mudah disesuaikan dengan perangkat keras apa pun yang akan Anda gunakan untuk mengembangkan: pengendali Oculus + Touch, HTC Vive, headset realitas campuran (MR), headset VR seluler. Karena aksesibilitas perangkat keras dapat menjadi penghalang bagi banyak calon atau kreator VR baru, penggunaan standarnya mencakup simulator VR. Simulator VR memungkinkan kreator untuk dapat membuat game imersif yang berfungsi penuh di Unity beserta pratinjau yang memungkinkan input keyboard untuk menggantikan penggunaan pengendali dalam menavigasi pengalaman.

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

Cara VRTK memberdayakan partisipasi yang begitu besar adalah dengan menjadi sumber terbuka (berdasarkan lisensi MIT) sepenuhnya gratis bagi siapa saja yang ingin menggunakannya, untuk alasan apa pun mereka ingin menggunakannya, baik untuk mempelajari seluk-beluk pengembangan VR atau menggunakannya untuk membuat gim VR terbaru, atau bahkan jika mereka adalah entitas komersial yang membuat solusi simulasi.

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

7.1 SEJARAH 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 dalam VR. Skrip tersebut, yang

hanya membutuhkan waktu sekitar dua hingga tiga jam untuk ditulis, adalah satu skrip untuk Unity3d yang diseret dan dijatuhkan ke dalam sebuah adegan dengan perangkat kamera SteamVR dan segera memberikan kemampuan untuk menyinari penunjuk laser di dalam adegan dan berteleportasi ke mana pun ujungnya menunjuk.

Skrip tersebut juga memberikan kemampuan untuk mengambil barang menggunakan sistem pengambilan berbasis sambungan tetap yang sederhana. Ini hebat: Anda dapat membangun sebuah adegan, dan dengan sekali seret dan jatuhkan skrip sumber terbuka, Anda dapat bergerak di sekitar adegan dan mengambil barang serta melemparkannya. Langkah selanjutnya adalah membagikannya kepada dunia, jadi di saluran YouTube yang tidak dikenal dengan tidak lebih dari 100 pelanggan dan hampir tidak ada penayangan rutin, sebuah video diunggah yang menunjukkan cara menggunakan skrip VR ini. Setelah beberapa hari, video tersebut ditonton ribuan kali, yang dengan jelas menunjukkan bahwa ada banyak orang dalam situasi yang sama, yang ingin sekali menemukan tutorial tentang cara membuat konten untuk VR. Kami segera menyadari bahwa jika ada kebutuhan dan keinginan untuk konten semacam ini, skrip dasar dan rapuh ini bukanlah cara terbaik bagi orang untuk maju. Skrip tersebut terlalu membatasi, terlalu bergantung (artinya satu skrip melakukan segalanya sehingga penyesuaiannya akan sangat merepotkan), dan itu bukanlah sesuatu yang dapat dikerjakan bersama-sama dalam sebuah komunitas.

Selamat datang di SteamVR Unity Toolkit

Setelah keberhasilan skrip tunggal asli ini, lahirlah SteamVR Unity Toolkit, yang merupakan upaya yang lebih terpadu untuk mencoba membuat kumpulan skrip yang dapat digunakan kembali dan diperluas yang membuat pembuatan untuk HTC Vive lebih mudah dan cepat tidak hanya bagi pengembang berpengalaman, tetapi juga bagi pemula yang ingin mencoba tetapi tidak tahu apakah mereka dapat melakukannya. SteamVR Unity Toolkit merupakan nama yang sangat tepat karena pada dasarnya merupakan perangkat skrip yang dibuat dalam Unity3d yang membantu saat menggunakan plug-in SteamVR, yang menawarkan sekumpulan solusi seperti teleportasi, pointer, grabbing, dan touchpad locomotion. Karena semuanya sepenuhnya gratis dan open source, perangkat ini mulai menarik perhatian orang-orang yang membuat konten yang menarik bagi mereka dalam VR, beberapa konten ini kemudian menjadi beberapa game VR yang paling terkenal.

Saat itu adalah era Wild West dalam pengembangan VR; tidak seorang pun benar-benar tahu masalah apa yang akan muncul dari media tersebut, dan tidak seorang pun benar-benar memiliki jawaban untuk solusinya. SteamVR Unity Toolkit menjadi repositori GitHub tempat orang-orang dapat berbagi ide untuk solusi dengan berkontribusi pada basis kode dan menyampaikan ide mereka kepada orang lain dalam upaya untuk memudahkan orang lain membangun pengalaman dan permainan VR mereka sendiri. Faktanya, pengembang di balik QuiVR menyumbangkan sejumlah fitur keren ke VRTK, terutama contoh adegan busur dan anak panah, yang menginspirasi sejumlah permainan busur dan anak panah yang menyenangkan untuk dibuat oleh pengembang pemula.

Seiring bertambahnya jumlah pengembang yang menggunakan SteamVR Unity Toolkit, semakin sulit untuk membantu orang-orang dengan masalah mereka masing-masing. 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 perangkat ini telah berkembang pesat dalam waktu yang singkat dan mereka membutuhkan tempat untuk berkomunikasi dan mengembangkan ide bersama secara bebas. Solusinya adalah saluran Slack yang dapat diikuti dan dikontribusikan oleh siapa saja, mencari bantuan, dan mengobrol tentang ide-ide yang pada akhirnya dapat menjadi fitur perangkat ini agar dapat digunakan oleh orang lain.

Saluran Slack masih menjadi jantung komunitas saat ini, dengan lebih dari 4.500 orang di seluruh dunia yang bekerja sama memecahkan masalah dan berbagi ide tentang cara membuat pengalaman VR lebih menarik bagi audiens mereka. Saluran ini telah menjadi tempat orang-orang menjalin ikatan komunitas yang nyata, persahabatan daring, atau kemitraan dalam usaha baru untuk membuat beberapa gim VR yang sangat keren. Orang-orang di komunitas merasa sangat bersemangat untuk berbagi ide-ide mereka, hal ini mengukuhkan SteamVR Unity Toolkit sebagai alat upaya komunitas, bukan pekerjaan satu orang, yang justru membantunya tumbuh dengan kecepatan yang terus meningkat dengan semakin banyaknya pengalaman yang diciptakan.

Saluran ini telah berkembang sedemikian rupa sehingga telah diperhatikan oleh beberapa perusahaan VR berpengalaman dengan perwakilan Oculus yang menanyakan apa yang dapat mereka lakukan agar alat ini berfungsi dengan headset mereka. Oculus cukup baik hati untuk menyediakan paket kontroler Oculus Rift dan Touch gratis sehingga Harvey dapat menjalankan perangkat tersebut pada headset selain Vive. Dalam beberapa hari, perangkat tersebut menjadi perangkat multiheadset dengan manfaat tambahan bahwa jika sesuatu dibuat untuk bekerja pada Vive, perangkat tersebut kini juga akan bekerja dengan lancar dengan Oculus Rift. Perangkat tersebut kini juga telah menjadi lapisan abstraksi perangkat pengembangan perangkat lunak (SDK) yang sangat kurang dari produk Unity3d.

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

Komunitas meneruskannya dengan membangun lebih banyak fitur keren seperti memanjat, gerakan mengayunkan lengan, berbagai jenis mekanika meraih, mulai dari yang menggunakan fisika untuk menggerakkan objek, hingga teknik yang lebih sederhana seperti menjadikan objek sebagai anak kontroler. Namun, masalah lain muncul: perangkat ini awalnya dibangun berdasarkan cara SteamVR disiapkan dan cara kerjanya. Integrasi Oculus SDK sebenarnya hanyalah lapisan abstraksi di atas cara kerja internal SteamVR. Harvey menyadari bahwa hal ini akan menyebabkan masalah yang lebih besar di kemudian hari saat headset dan teknologi lain dirilis. Semuanya tidak dapat bergantung pada dasar-dasar SteamVR, karena ini akan menjadi perbedaan mendasar dengan cara kerja headset lain di masa mendatang.

Saat ini, VRTK telah melakukan banyak hal dan banyak orang dengan senang hati menggunakannya untuk membangun berbagai hal yang luar biasa, tetapi jelas bahwa

perangkat ini perlu dipikirkan ulang, dibayangkan ulang, dan dibangun kembali dari awal dengan cara yang sangat berbeda yang tidak terikat pada teknologi tertentu.

Karena popularitas VRTK yang meledak, tidak banyak waktu untuk keputusan pemeriksaan akal sehat atau fondasi arsitektur, dan karena perangkat ini didasarkan pada warisan dukungan SteamVR, itu berarti basis kode terus berkembang berdasarkan konsep tersebut. Semakin banyak kode yang ditambahkan oleh kelompok kontributor yang terus berkembang, semakin sulit pula kode tersebut untuk dipelihara dan dikembangkan. Pada titik tersebut, jelas terlihat bahwa VRTK perlu ditulis ulang, hingga ke pertimbangan desain fundamentalnya.

VRTK v4

VRTK v4 akan menjadi pendekatan yang sama sekali baru untuk pengalaman menggunakan perangkat. Alih-alih skrip yang sudah dibuat sebelumnya yang melakukan hal tertentu, perangkat ini akan menjadi pola desain mendasar yang dapat disusun dalam berbagai konfigurasi untuk menyediakan fungsionalitas yang bermanfaat bagi VR (atau kasus penggunaan lainnya). Ini sangat penting karena berarti apa pun yang berubah dengan teknologi di masa mendatang, perangkat akan selalu ada untuk mendukungnya. Pengembang dapat dengan mudah membangun untuk perangkat keras baru apa pun yang mencoba untuk berhasil di pasar yang berarti lebih banyak permainan dan pengalaman dapat mendukungnya, yang hanya akan membantu keberhasilan setiap proses evolusi.

Pengerjaan VRTK v4 dimulai pada akhir April 2018 dengan pendekatan yang sama sekali baru terhadap cara kerja perangkat bagi pengembang. Selain itu, kurangnya ketergantungan pada fitur inti Unity3d berarti bahwa masa depan perangkat dapat, dan diharapkan, diperluas ke platform lain seperti Unreal Engine, WebVR, dan Godot, untuk menyebutkan beberapa di antaranya. Premis ini bahkan lebih menarik bagi potensi VRTK: jika seorang pengembang dapat memahami dasar-dasar VRTK dan cara membuat solusi menggunakannya di Unity3d, seharusnya tidak banyak yang menghalangi mereka untuk mentransfer pengetahuan itu ke platform lain. Satu-satunya penghambat adalah mempelajari cara menggunakan antarmuka platform lain, tetapi kemampuan untuk memilih dan memilah head-mounted display (HMD) dan engine untuk membangun akan sangat bermanfaat bagi semua pengembang.

Salah satu hasrat besar tim dan komunitas VRTK adalah memastikan bahwa VRTK dan pengembangan VR dapat diakses oleh sebanyak mungkin orang. VRTK telah digunakan di banyak hackathon, lokakarya, dan lembaga pendidikan seperti sekolah menengah dan universitas untuk mengajarkan pengembangan VR kepada gelombang baru kreator. Bagaimana VRTK v4 juga dapat menyelaraskan kekuatan perangkat 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. Namun, ada pertanyaan yang perlu dijawab: apakah mungkin untuk memiliki koleksi panduan, tutorial, video, dan materi pembelajaran yang membantu mengajarkan kekuatan VRTK tetapi juga secara konsisten dan menyediakan berbagai level yang, tergantung pada keahlian pengguna, dapat menjadi titik masuk yang layak dan dapat dipahami?

Kemampuan untuk menghadirkan media yang sama sekali baru ke dunia kreator baru

adalah peluang khusus. Munculnya komputasi rumahan pada tahun 80-an memungkinkan programmer kamar tidur untuk menciptakan industri gim video. Dapatkah VRTK membantu menghidupkan kembali gerakan seperti itu tanpa VR? Semangat untuk melakukannya tentu ada, dengan penekanan pada pendidikan yang setara dengan benar-benar membangun alat untuk melakukannya. Yang lebih penting adalah agar sebagian besar konten pendidikan juga gratis dan bersumber terbuka sehingga dapat dengan mudah digunakan dan disumbangkan oleh siapa pun di bidang pendidikan.

Masa Depan VRTK

Masa depan VRTK bukan hanya menyediakan platform bagi para pemula untuk memulai perjalanan pengembangan mereka, tetapi juga membantu dan meningkatkan proses pengembangan dengan cepat bagi para pengembang berpengalaman, dari pengembang indie hingga pengembang AAA. Menyediakan alat yang andal dan teruji untuk mencegah mereka harus menciptakan kembali roda berarti ide-ide baru dapat dibuat prototipenya dengan cepat untuk menentukan apakah mekanismenya berfungsi. Kemampuan untuk berfokus pada konten dan bukan mekanisme berarti para pengembang dapat berupaya lebih keras untuk menghasilkan konten yang sangat bagus yang akan meningkatkan minat pasar yang masih baru, tetapi juga dengan kekuatan dasar VRTK v4 berarti para pengembang ini dapat lebih menyesuaikan dan memperluas solusi dasar untuk memberikan pengalaman yang lebih unik.

Kemampuan untuk membuka aksesibilitas VR bagi perusahaan juga merupakan misi penting bagi VRTK. Untuk memungkinkan industri menguji coba solusi VR dengan cepat dan murah untuk masalah sehari-hari berarti bahwa penerimaan komersial VR akan lebih cepat, yang menghasilkan lebih banyak investasi bagi media baru yang luar biasa ini untuk berkembang dan makmur. Dengan harapan dan cinta dari komunitas VRTK, VRTK akan terus mendukung pengembangan VR sebagai media dan bahkan memperluasnya ke masa depan untuk mendukung sektor komputasi spasial lainnya seperti augmented reality (AR). Masa depan tampak cerah, dan semoga VRTK mampu membuatnya jauh lebih cerah.

Konsep di balik VRTK v3 adalah menyediakan satu skrip tunggal yang memberikan fungsionalitas tertentu. Meskipun hal ini memudahkan untuk menjalankan sesuatu hanya dengan menyeret dan meletakkan skrip, hal ini berarti bahwa penyesuaian komponen fungsionalitas akan memerlukan perluasan skrip dan bahkan kemungkinan penulisan potongan kode yang besar. VRTK v4 bertujuan untuk memecah fungsionalitas yang dapat digunakan menjadi komponen-komponen umum yang memiliki tanggung jawab untuk melakukan satu pekerjaan tertentu, dan komponen-komponen kecil ini kemudian digabungkan untuk membentuk prefab yang menjalankan fungsionalitas yang sama dengan yang dilakukan oleh skrip tunggal. Prefab yang berisi komponen-komponen yang relevan ini dihubungkan menggunakan peristiwa sehingga jika ada bagian dari jalur eksekusi yang perlu diubah atau diamandemen, ia dapat dengan mudah menghubungkan pendengar baru pada peristiwa tersebut, yang sering kali mengakibatkan tidak perlunya pengodean.

Manfaat dari subkomponen ini dalam VRTK v4 adalah banyaknya fungsionalitas yang dapat digunakan kembali yang dapat disebarkan ke berbagai kasus penggunaan, baik itu memindahkan objek atau mendeteksi tabrakan. Ini juga berarti bahwa kode inti yang

mendasarinya dalam VRTK v4 sama sekali tidak ada hubungannya dengan VR, sehingga dapat digunakan untuk tujuan apa pun, baik itu VR, AR, atau bahkan sekadar pengalaman desktop atau seluler. Prefab yang berada di atas kode inti menyediakan fungsionalitas khusus, yang memungkinkan persyaratan baru apa pun untuk 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 yang dibangun di sekitar cara kerja internal 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 sangatlah sulit. VRTK v4 tidak memiliki fondasi dalam SDK apa pun dan karenanya benar-benar generik dan harus dapat mendukung sejumlah perangkat dengan relatif mudah. Contoh yang bagus adalah sesuatu seperti menggerakkan pemain di sekitar suatu adegan; dalam 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 dalam kaitannya dengan touchpad atau yang setara. Ini berarti bahwa apa pun yang hanya ingin menyuntikkan informasi arah ke dalam skrip gerakan pemain harus melalui seluruh alur kerja SDK untuk mencapainya.

Dalam VRTK v4, karena tidak ada ketergantungan pada pengetahuan intrinsik semacam itu, sangat mudah untuk membuat "Action" yang memancarkan Vector2 yang berisi data arah dan kemudian operasi dapat dilakukan pada Vector2 untuk mengubahnya di sepanjang jalan, seperti mengalikan elemen-elemennya 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 meneruskan pesan antar subkomponen, jauh lebih mudah untuk membuat fungsionalitas kustom tanpa perlu menulis kode apa pun. Ada juga manfaat dari kemampuan untuk menggunakan alat skrip visual untuk membuat fungsionalitas menggunakan drag and drop sederhana. Ini adalah langkah maju yang hebat bagi mereka yang tidak berasal dari latar belakang pengkodean tetapi tetap ingin menciptakan pengalaman unik tanpa perlu mempelajari pengkodean yang mendasarinya.

7.2 KEBERHASILAN VRTK

Sejak kemunculan VRTK, perangkat ini telah diunduh lebih dari 30.000 kali dan telah digunakan dalam berbagai macam proyek mulai dari pengembang indie solo hingga studio game AAA. Gambar 7-1 menunjukkan sebagian kecil judul yang diterbitkan yang menganggap VRTK mampu mengurangi waktu pengembangan secara cepat menuju produksi yang tersedia di semua platform utama, termasuk Oculus Store dan Steam.



Gambar 7.1. Berikut ini beberapa proyek sukses yang menggunakan VRTK

Anda dapat menemukan daftar lengkap game yang dipublikasikan yang menggunakan VRTK secara daring.

Memulai dengan VRTK 4

VRTK adalah kumpulan skrip dan demonstrasi konsep yang bermanfaat untuk membantu membangun solusi VR dengan cepat dan mudah. Tujuannya adalah membuat pembangunan solusi VR di Unity3d menjadi cepat dan mudah bagi para pemula dan pengembang berpengalaman.

VRTK mencakup sejumlah solusi umum seperti berikut:

- Pergerakan 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

Menyiapkan proyek

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

1. Buat proyek baru di Unity3d 2018.1 atau yang lebih baru menggunakan Template 3D.
2. Pastikan kotak centang Didukung Realitas Virtual dipilih.
 - a. Di menu utama Unity3d, klik “Edit,” lalu “Project Settings,” lalu “Player.”
 - b. Di panel inspektur PlayerSettings, perluas XR Settings.
 - c. Pilih kotak centang opsi Virtual Reality Supported.
3. Perbarui proyek ke Versi Scripting Runtime yang didukung.
 - a. Di menu utama Unity3d, klik “Edit,” lalu “Project Settings,” lalu “Player.”
 - b. Di panel inspektur PlayerSettings, perluas Other Settings.
 - c. Ubah Versi Scripting Runtime ke .NET 4.x Equivalent.
 - d. Unity sekarang akan memulai ulang di runtime skrip yang didukung.

Mengkloning repositori

Berikut cara mengkloning repositori VRTK ke dalam proyek Anda:

1. Arahkan ke direktori Assets/ proyek.

2. Git clone submodule yang diperlukan ke dalam direktori Assets/:

```
git clone --recurse-submodules https://github.com/thestonefox/VRTK.git
git submodule init && git submodule update
```

Menjalankan pengujian

Buka adegan VRTK/Scenes/Internal/TestRunner:

1. Di menu utama Unity3d, klik “Window”, lalu “Test Runner”.
2. Pada tab EditMode, klik Run All.
3. Jika semua pengujian berhasil, instalasi Anda berhasil.

Menyiapkan lingkungan Anda

1. Unduh versi terbaru VRTK dari repositori GitHub (lihat Tabel 7-1) di www.vrta.io (atau www.github.com/thestonefox/VRTK).

Tabel 7.1. SDK yang didukung

SDK yang didukung	Tautan unduhan
Simulator VR	Termasuk
SteamVR	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 VR Headset, 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. 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 *Aset/VRTK/Contoh*, lalu buka Adegan Contoh mana pun, tekan Putar untuk melihat bagaimana interaksi terlihat di Adegan Game Anda.

Adegan contoh

Kumpulan adegan contoh 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 kalinya karena adegan tersebut juga berfungsi sebagai titik awal yang bagus untuk pembuatan prototipe cepat atau permulaan proyek dasar.

Adegan contoh adalah lingkungan yang siap disiapkan untuk fungsionalitas instan dengan SDK pilihan Anda. Setiap adegan ini diberi judul berdasarkan jenis fungsionalitas yang akan ditunjukkan oleh adegan tersebut. Adegan contoh dapat dengan mudah diduplikasi dan disesuaikan ke dalam proyek Anda dan adegan tersebut mendukung semua SDK VR yang didukung VRTK.

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

7.3 MEMERIKSA REPOSITORI 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. Jika Anda menggunakan Macintosh, buka Terminal.
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 hingga selesai.
6. Opsional: Unduh SDK untuk perangkat keras yang Anda inginkan.
7. Buka *Aset/VRTK/Contoh*, buka Adegan Contoh mana pun, lalu tekan putar untuk melihat bagaimana interaksi terlihat di Adegan Permainan Anda.

Berikut ini adalah daftar adegan contoh dan fitur interaksi terkini (saat tulisan ini dibuat):

Adegan input

Menampilkan informasi yang diberikan oleh kontroler atau input keyboard ke dalam game.

Adegan penunjuk objek

Menampilkan raycast, menggunakan laser hijau yang dipancarkan oleh kontroler. Anda dapat mengarahkan laser ke berbagai objek pada adegan dan melihat berbagai reaksi yang dapat ditimbulkan oleh penunjuk saat diarahkan ke objek tertentu.

Penunjuk lurus

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

Penunjuk Bezier

Emisi garis lengkung yang menunjuk ke tanah—ini bisa dibilang pengalaman pengguna terbaik untuk teleportasi.

Teleportasi tunjuk-dan-klik

Dengan menggunakan penunjuk, Anda dapat memilih area yang ingin Anda tuju dan dalam "kedipan" Anda akan diposisikan ulang ke lokasi tersebut.

Teleportasi instan

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

Dash teleport

Arahkan ke area yang ingin Anda teleportasi, dan ini akan mempercepat frame gerakan agar pengguna tiba di sana. Gerakan ini muncul dan terasa lebih alami daripada metode teleportasi instan.

Adekan teleportasi

Menampilkan berbagai area dan jenis area yang dapat Anda teleportasi di adegan dengan mengeklik Thumbstick untuk mengaktifkan teleportasi dan mengarahkan ke arah atau titik yang ingin Anda tuju dengan mengeklik tombol pemicu.

Objek yang dapat berinteraksi

Teleportasi di sekitar adegan ke berbagai objek. Gunakan tombol pegangan pada pengontrol untuk melihat berbagai jenis pegangan pada setiap objek. Berikut adalah beberapa contoh jenis interaksi yang tersedia pada saat rilis di folder aset VRTK:

Pegangan presisi

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

Gun grab

Mengambil objek berbentuk pistol akan selalu menempatkannya dalam posisi siap tembak di tangan saat diambil dari sudut mana pun.

Toggle grab

Memungkinkan Anda melepaskan tombol saat objek diambil dan mempertahankan posisi yang diambil hingga Anda menekan tombol lagi.

Two-handed hold

Menahan objek dalam posisi fungsional saat kedua tangan telah mengambilnya.

Pompa

Memungkinkan penggunaan aksi pemompaan pada objek untuk menghasilkan efek tertentu.

Hinge joint

Bergerak hanya pada sumbu tertentu—misalnya, pintu yang terbuka dan tertutup. Interaksi lainnya akan mencakup scaled grab, two-handed precision grab, basic joint, customizable joint, character joint, pick and grabs, dan masih banyak lagi. Untuk informasi lebih lanjut tentang rilis aset ini, kunjungi situs web kami!

Scene Switcher

Dalam mode permainan, Anda dapat memilih CameraRig Switcher untuk berganti-ganti antara berbagai jenis SDK yang Anda gunakan untuk pengalaman Anda atau memilih simulator. Hal ini sangatlah berguna, jika bukan penggunaan VRTK yang paling menguntungkan, karena Anda dapat merakit untuk beberapa headset dan melihat pratinjau tampilannya secara langsung selama tahap pengembangan untuk masing-masing headset hanya dengan sekali klik tombol.

Menyiapkan Proyek Vrtk Core Dari Awal

1. Di proyek Unity 3D 2018.1+ Anda, buka adegan kosong.
2. Pilih kotak centang Virtual Reality Supported.
3. Buka Project Settings → Player → Other Settings. Di bawah Configuration, ubah Scripting Runtime Version ke .Net 4x Equivalent.
4. Saat diminta, tekan Restart.
5. Unduh paket VRTK.Unity.Core dari GitHub, lalu seret dan letakkan Assets Folder ke proyek Unity 3D Anda.

Cara menyiapkan Unity CameraRig

1. Di adegan Anda, buka tab “Hierarchy”, lalu hapus Default Camera.
2. Seret VRTK Camera Rig ke adegan Anda.
 - a. Buka Assets → VRTK Unity.Core → CameraRig → [UnityXRCameraRig]. Seret [UnityXRCameraRig] Prefab ke tab Hierarki Anda.1
3. Tekan putar pada adegan untuk melihat pratinjau kamera.

Jangkar kepala

Objek permainan induk yang merujuk ke posisi headset.

Jangkar kiri

Objek permainan anak yang merujuk ke lensa mata kiri headset.

Jangkar kanan

Objek permainan anak yang merujuk ke lensa mata kanan headset.

Cara menyiapkan Alias Terlacak

1. Seret prefab “Alias Terlacak” ke Hierarki Anda.
2. Buka Aset → VRTK.Unity.Core → CameraRig → TrackedAlias.
3. Objek permainan alias terlacak adalah objek permainan anak yang dapat Anda sesuaikan untuk interaksi pengguna yang diwujudkan yang disesuaikan dengan sensor yang akan digunakan terkait dengan jenis perangkat keras.

Alias area bermain

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

Alias headset

Mengacu pada posisi headset

Alias pengontrol kiri

Mengacu pada pengontrol tangan kiri

Alias pengontrol kanan

Mengacu pada pengontrol tangan kanan

Kamera adegan

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

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

- Cara menyiapkan Simulator
- Pengantar Tindakan VRTK
- Cara menyiapkan penunjuk
- Cara menyiapkan teleportasi dengan penunjuk
- Cara menyiapkan objek yang dapat berinteraksi (interaktor/yang dapat berinteraksi)

BAB 8

PRAKTIK PENGEMBANGAN VR DAN AR

Pengembangan untuk Realitas Virtual dan Realitas Tertambah Itu Sulit Dan mungkin itulah alasan Anda membaca buku ini sejak awal. Namun, penting untuk memahami kompleksitasnya sebelum terjun ke pengembangan. Jadi, mari kita uraikan terlebih dahulu apa yang membuat pengembangan jauh lebih rumit daripada kebanyakan bidang lainnya.

Mari kita mulai dengan alat-alatnya. Sepanjang bab ini, kita bekerja dengan mesin permainan Unity. Awalnya dirilis pada tahun 2005, Unity telah membantu banyak pengembang di seluruh dunia untuk mulai membangun permainan tiga dimensi, mulai dari seluler hingga konsol hingga desktop. Dan meskipun telah menjadi tulang punggung pengembangan 3D bagi banyak orang dan telah menumbuhkan komunitas yang luar biasa selama bertahun-tahun, itu sama sekali tidak sempurna, terutama karena paradigma desain untuk realitas virtual (VR) dan realitas tertambah (AR) terus berkembang. Sejak kit pengembangan VR modern pertama kali dirilis pada tahun 2013, berbagai alat bawaan dan plug-in eksternal Unity telah mengalami peningkatan yang signifikan, tetapi tugas-tugas tertentu seperti pengembangan lintas platform dan multipemain masih belum semudah mengaktifkan sebuah tombol. Ingatlah hal itu saat Anda melanjutkan membaca bab ini.

Selanjutnya, perangkat keras. Lebih dari sekadar alat, jumlah perangkat keras yang berbeda dapat meningkatkan kompleksitas secara drastis. 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 tersebut. Meskipun ini bukan hal baru jika Anda berasal dari latar belakang grafis atau game, setiap perangkat memiliki serangkaian tombol dan persyaratan pelacakan unik yang perlu diintegrasikan ke dalam cara setiap aplikasi tertentu dikembangkan.

Dan, terakhir, pemeliharaan. Seperti yang disebutkan sebelumnya, VR dan AR adalah bidang yang terus 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 baru atau alat untuk headset yang sudah ada dapat berubah lebih cepat dari itu. Ini mengharuskan Anda untuk selalu memperbarui kode, terkadang bahkan sebelum merilis pengalaman Anda. Mungkin ini memakan waktu, tetapi penting untuk memastikan bahwa pengalaman Anda berjalan dengan sukses untuk semua orang.

Sekarang, saya tahu bahwa ada banyak masalah negatif di bidang ini, tetapi ada secercah harapan di ujung terowongan. Menurut saya, VR dan AR adalah pengembangan yang paling memuaskan yang dapat Anda lakukan. Mungkin tidak mudah dan pasti akan ada titik-titik yang membuat frustrasi di sepanjang jalan, tetapi ketika Anda melihat orang-orang mengenakan headset dengan senyum di wajah mereka, itu sangat memuaskan. Dan alasan mengapa saya memulai bab ini dengan pengantar yang panjang itu adalah untuk memperjelas apa saja batasan dalam pengembangan dan benar-benar menekankan kendala yang akan Anda hadapi. Ada solusi untuk sebagian besar masalah ini melalui perencanaan yang cermat

dan dengan bekerja sama dengan tim desain untuk menciptakan cakupan yang memberikan pengalaman yang menarik dan menyembunyikan semua masalah yang disebutkan di atas.

Pengembangan VR dan AR itu sulit, jadi beralihlah ke bab ini, mari kita pelajari bagaimana kita dapat memanfaatkan beberapa kiat dan trik untuk membuatnya lebih mudah. Dengan semua yang telah dikatakan, mari selami tiga praktik pengembangan yang dapat Anda gunakan dalam VR dan AR.

8.1 MENANGANI PERGERAKAN

Pertama-tama, kita akan melihat cara membangun beberapa jenis mekanika pergerakan untuk VR dan AR. Pergerakan bisa sangat sederhana tetapi merupakan mekanika yang sangat penting untuk pengalaman apa pun. Mekanika ini memungkinkan pengembang untuk mengambil dunia yang tak terbatas dan membuatnya dapat dilalui dalam ruang yang terbatas; misalnya, kamar Anda. Ada banyak cara untuk memecahkan masalah pergerakan, dan jenis pergerakan yang Anda pilih akan sering ditentukan oleh apa yang menurut audiens Anda paling mendalam. Di bagian ini, kita membangun tiga jenis pergerakan yang berbeda: gerakan linier, teleportasi, dan gerakan berskala.

Sebelum kita mulai membangun, saya ingin menyebutkan beberapa sistem gerak loko yang perlu diperhatikan yang tidak akan kita bangun tetapi dapat berguna untuk pembelajaran di masa mendatang:

Berjalan terarah

Teknik grafis yang sedikit mendistorsi gambar yang ditampilkan pada headset agar pengguna berpikir bahwa mereka berjalan dalam garis lurus, padahal sebenarnya mereka berjalan di jalur yang melengkung. Anda memerlukan area yang luas untuk mengelabui otak.

Berlari cepat

Menggerakkan pengguna dengan cepat ke tujuan mereka dalam interval pendek; misalnya, 0,5 detik. Keuntungannya adalah pengguna memiliki rasa keterlibatan yang lebih baik sambil mengurangi potensi mabuk simulasi.

Memanjat

Pengguna menggunakan tangan mereka untuk menarik diri ke arah yang diinginkan, sering kali dengan memegang objek virtual.

Controller Assisted On the Spot (CAOTS)

Menggunakan pengontrol yang dilacak secara posisional saat bergerak di tempat untuk bergerak secara virtual. Pelacakan ini digunakan dalam aplikasi VR Loko-gerak Freedom yang terdaftar gratis di Steam.

1 banding 1

Jika Anda dapat mengatur agar semua yang diperlukan untuk pengalaman Anda berada dalam jangkauan pemain, pergerakan mungkin tidak diperlukan sama sekali, seperti dalam Job Simulator. Dan ini dapat dilakukan selangkah lebih maju dengan ruang virtual yang beradaptasi dengan seberapa besar ruang pemain, sehingga pengalaman menjadi lebih mudah diakses.

8.2 PERGERAKAN DALAM VR

Sebelum kita mulai, penting untuk dicatat bahwa jenis pergerakan yang terintegrasi ke dalam aplikasi sangat bergantung pada aplikasi itu sendiri. Implementasi yang akan datang adalah yang paling umum di seluruh permainan VR, tetapi dengan demikian, implementasi tersebut mungkin tidak tepat untuk apa pun yang ingin Anda bangun. Meskipun demikian, implementasi ini penting untuk diketahui, terutama saat Anda membuat prototipe untuk melihat apa yang berhasil dan apa yang tidak.

Pergerakan linear (alias pergerakan trackpad)

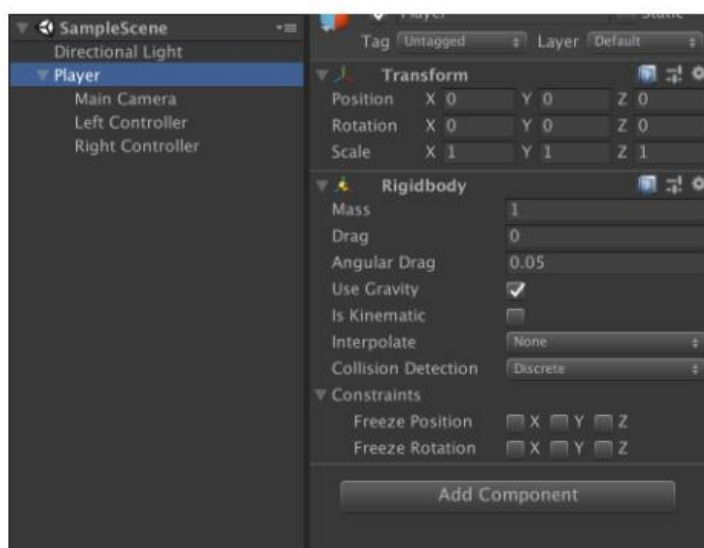
Selain Google Cardboard, semua sistem VR modern dilengkapi dengan semacam pengontrol, seperti yang ditunjukkan pada Gambar 8-1, dan semua pengontrol ini dilengkapi dengan joystick atau trackpad.



Gambar 8.1. Kontroler Oculus Rift (kiri) dan Kontroler HTC Vive (kanan)

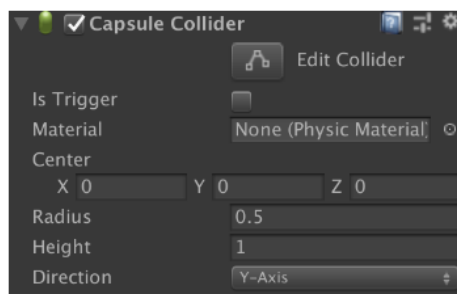
Dengan mengetahui hal ini, kita dapat membuat sistem gerakan 2D sederhana menggunakan joystick atau touchpad sebagai input. Jika Anda familier dengan gim first-person-shooter, mekanisme gerakan ini akan sangat mirip dengan gim-gim tersebut.

Untuk memulai, pertama-tama kita perlu menyiapkan pemain. Untuk ini, kita menggunakan simulasi fisika bawaan Unity, yang berarti kita perlu memasang komponen Rigidbody (lihat Gambar 8.2) ke pemain. Beberapa manfaat menggunakan Rigidbody meliputi penambahan gaya dan kecepatan ke objek serta simulasi tabrakan fisika antara dua objek. Ini sangat berguna bagi kita karena kita ingin menggunakan kecepatan untuk menggerakkan pemain secara linear serta mendeteksi saat pemain bertabrakan dengan tanah.



Gambar 8.2. Rigidbody pada pemutar VR kita

Selanjutnya, kita perlu menambahkan collider untuk menentukan batas pemutar kita. Di sini, saya punya kabar baik dan kabar buruk. Kabar baiknya adalah saat kita menentukan batas pemutar, batasnya tidak perlu sempurna. Collider kapsul sederhana sudah cukup, yang merupakan collider bawaan untuk Unity, seperti yang diilustrasikan pada Gambar 8.3, dan sangat dioptimalkan untuk performa.



Gambar 8.3. Kapsul penumbuk bawaan Unity (ditambahkan ke pemain)

Kabar buruknya adalah untuk VR (dan AR), tidak seperti mendefinisikan batas tumbukan pemain dalam gim video tradisional, tidak ada tinggi yang cocok untuk semua orang yang memainkan gim Anda. Ada beberapa perbaikan potensial untuk masalah ini:

- Minta pemain untuk berdiri diam sebelum memulai sehingga mereka dapat diukur selama sesi berlangsung
- Asumsikan bahwa tinggi pemain saat ini adalah tinggi maksimum mereka

Meskipun tidak ada perbaikan yang optimal, tergantung pada kasus penggunaan Anda, satu solusi mungkin lebih bermanfaat dibandingkan dengan yang lain. Satu hal yang perlu diperhatikan adalah bahwa di sebagian besar perangkat VR (SteamVR, VRTK, dll.), solusi kedua diimplementasikan secara default. Untuk melihat bagaimana mereka dapat mengimplementasikannya, lihat pseudocode ini (buka GitHub untuk implementasi

sebenarnya):

```
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
}
```

Setelah sistem fisika siap, kita sekarang dapat mulai menciptakan gerakan linear. Sama seperti pada kapsul penumbuk, 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 semua yang Anda butuhkan untuk menyiapkan beberapa gerakan linear sederhana. Meskipun ini adalah mekanisme gerakan yang cukup sederhana untuk disiapkan, ini sama sekali tidak cocok untuk setiap audiens. Yang kami sarankan adalah menyediakan sistem penggerak ini sebagai opsi bagi pengguna yang suka berpetualang (yang merupakan sebagian besar pengguna VR) dan kemudian menyertakan sistem penggerak kami berikutnya, teleportasi, sebagai sistem bagi mereka yang lebih sensitif terhadap mabuk simulasi.

8.3 PENGGERAKAN TELEPORTASI

Sejak Pengembangan Oculus pertama diluncurkan, teleportasi telah menjadi salah satu solusi paling sederhana, efektif, dan agak kontroversial untuk melintasi ruang virtual yang besar. Di satu sisi, ini menghindari banyak masalah yang dialami sistem penggerak lain dengan mabuk simulasi, menjadikannya yang paling mudah diakses. Namun, tergantung pada jenis pengalaman yang Anda bangun, ini juga dapat kehilangan rasa keterlibatan dengan sangat cepat. Meskipun demikian, ini adalah alat yang luar biasa untuk disimpan di ikat pinggang Anda karena lebih sering daripada tidak Anda akan ingin memasukkannya dalam pengalaman Anda. Jadi, mari kita bangun! Ada beberapa jenis teleportasi, 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 lintasan melengkung sering digunakan:

- Membatasi seberapa jauh pemain dapat bepergian, yang membatasi pemain untuk melakukan perjalanan di seluruh level.
- Mengurangi presisi yang dibutuhkan pemain untuk berakhir di lokasi yang diinginkan.



Gambar 8.4. Teleportasi melengkung

Untuk memulai, mari kita lakukan beberapa pengaturan terlebih dahulu. Kita ingin beberapa variabel untuk menyesuaikan teleportasi kita serta metode untuk merender teleportasi kita. Untungnya, Unity memiliki Komponen Line Renderer bawaan yang sangat dapat disesuaikan untuk mendapatkan tampilan dan nuansa yang Anda inginkan.

Dengan pengaturan tersebut, kita dapat fokus memeriksa masukan untuk memulai teleportasi kita. Bergantung pada platform tempat Anda mengembangkan, 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 ditekan pemain sesering

mungkin, seperti trackpad pada pengontrol Vive atau pemicu pada Oculus Rift. Setiap kali tombol itu ditahan, tunjukkan jalur melengkung untuk teleportasi, lalu saat dilepaskan, teleport ke lokasi tersebut.

Untuk kenyamanan lebih, Anda juga dapat memilih untuk memudahkan pandangan pemain menjadi hitam dengan sangat cepat untuk meningkatkan kenyamanan teleportasi (periksa repositori GitHub untuk solusi yang berfungsi):

```
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 utamanya adalah `CalculateCurvedPath`, yang mengambil titik awal, arah, dan efek gravitasi, lalu mengeluarkan jalur lengkung. Dengan masukan tersebut, kita dapat menjalankan simulasi fisika sederhana untuk menghitung berapa lama waktu yang dibutuhkan kurva untuk mendarat di tanah. Untuk menyederhanakannya, kita mengasumsikan kecepatan hanya sebagai arah yang dinormalkan, 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 position
        hypoPos=currPos+hypoVel+(gravityDirection*Time.fixedDeltaTime
        );
// calculate next point on curve
        hypoVel = hypoPos - currPos; //calculate 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 Raycast(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 lintasan dihitung, kita tinggal menetapkan semua titik ke perender garis, yang akan secara otomatis merender garis. Satu hal yang perlu diperhatikan di sini adalah Anda harus kembali ke kode input dan mengaktifkan serta menonaktifkan perender garis setiap kali tombol ditekan dan dilepas. Jika tidak, titik pertama yang ditetapkan ke lintasan akan selalu ada di sana, yang merupakan bug yang lucu tetapi mungkin tidak disengaja. Dan dengan itu, kita memiliki teleportasi, sedikit lebih rumit daripada gerakan linier, tetapi itu pasti sepadan dalam banyak pengalaman.

8.4 PENGGERAK DALAM AR

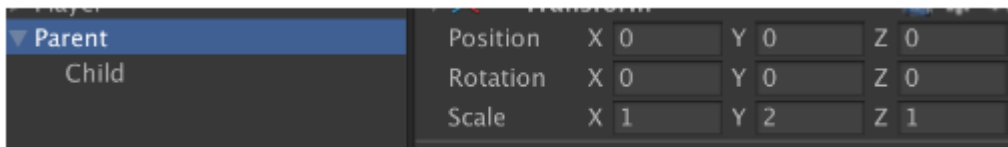
Kita baru saja membahas satu sistem penggerak yang sangat penting untuk VR. Sistem penggerak berikutnya bekerja dengan baik dalam VR dan AR, tetapi sangat berdampak dalam AR karena tidak banyak alat untuk penggerak mengingat akan tampak sedikit janggal jika objek virtual berubah posisi karena penggerak, sedangkan di dunia nyata objek tersebut tetap diam. Salah satu cara untuk mengatasi hal ini dengan AR adalah dengan menggunakan trik visual yang mengubah persepsi skala dan ini sebenarnya cukup mudah dicapai.

Mode Raksasa (atau Semut)

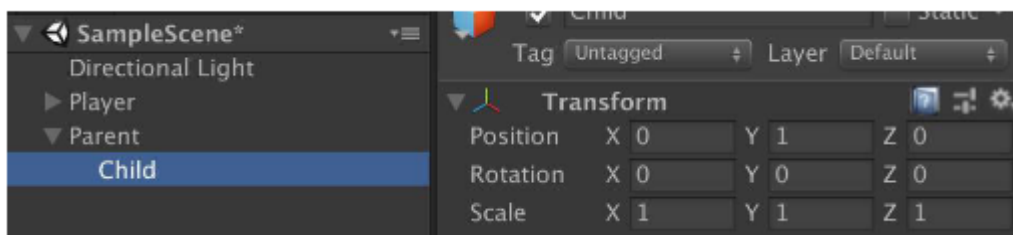
Salah satu nilai jual terbesar VR dan AR adalah kemampuan untuk mengalami hal-hal baru, dan nilai tambah lainnya adalah kemampuan untuk mengukur sesuatu dari sudut pandang yang benar-benar baru; yaitu, menjadi sangat besar atau sangat kecil.

Salah satu cara untuk mewujudkannya adalah dengan menambah atau mengurangi skala setiap objek individual dalam sebuah adegan—masalah dengan pendekatan ini adalah pemain masih berukuran manusia dan, sebagai hasilnya, jarak yang ditempuh pemain akan tetap berskala manusia.

Sebaliknya, untuk mencakup lebih banyak (atau lebih sedikit) wilayah dan menskalakan jarak yang ditempuh, sering kali lebih baik untuk menskalakan pemain. Cara kerja skala di Unity (dan di hampir semua mesin gim) adalah bahwa posisi dan nilai skala objek apa pun yang bersarang di bawah objek induk akan dikalikan. Untuk mengilustrasikan hal ini, lihat Gambar 8-5 dan 8-6.



Gambar 8.5. Parameter transformasi objek induk



Gambar 8.6. Parameter transformasi objek anak

Transformasi objek Anak akan menjadi:

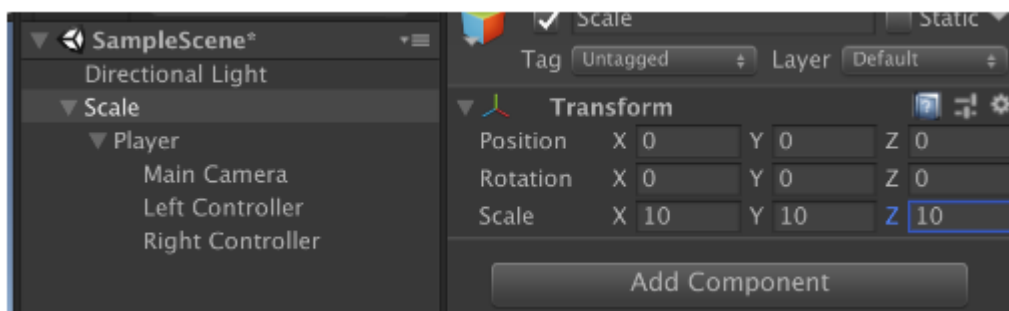
Posisi

$$(0, 1, 0) \text{ Anak dikalikan (produk titik) dengan } (1, 2, 1) \text{ Induk} = (0, 2, 0)$$

Skala

$$(1, 1, 1) \text{ Anak dikalikan (produk titik) dengan } (1, 2, 1) \text{ Induk} = (0, 2, 0)$$

Dengan menggunakan pengetahuan itu, kita dapat menerapkannya pada pemutar AR kita. 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

Dengan itu, kami telah menciptakan pemutar berskala raksasa jika skala tersebut ditetapkan lebih besar dari (1, 1, 1), dan semut jika skala tersebut kurang dari (1, 1, 1). Catatan penting yang perlu diingat adalah Anda dapat menetapkan skala untuk setiap sumbu ke nilai yang berbeda, tetapi ini akan menghasilkan pengalaman yang sangat membingungkan. Saya sangat menyarankan agar Anda menjaga semua skala tetap sama persis dan sebagai nilai positif untuk menghindari perilaku aneh.

Dan itu saja untuk penggerak. Seiring perubahan perangkat keras VR dan AR, berbagai jenis penggerak akan diujicobakan, dan ada kemungkinan beberapa sistem yang umum digunakan ini akan ketinggalan zaman seiring berjalannya waktu. Namun, mampu membangun semua sistem ini akan berguna untuk mulai melakukan curah pendapat tentang cara-cara baru untuk menangani perkembangan baru di bidang ini.

8.5 PENGGUNAAN AUDIO YANG EFEKTIF

Ketika Anda mendengar istilah “VR” atau “AR,” manakah dari kelima indra Anda yang terlintas dalam pikiran? Bagi kebanyakan orang, indra tersebut adalah penglihatan—kemampuan untuk melihat dunia dan bereaksi terhadapnya. Audio hampir selalu menempati posisi kedua atau ketiga (setelah sentuhan). Namun dalam realitas (dan realitas virtual), audio sama pentingnya, jika tidak lebih penting, daripada visual dunia untuk menciptakan suasana. Bayangkan ketiga skenario ini, Anda sedang berlayar di lautan virtual seperti yang ditunjukkan pada Gambar 8.8, ketika tiba-tiba Anda mendengar:

- ❖ Musik disko yang memukau
- ❖ Detak jantung dari belakang Anda
- ❖ Tidak ada apa-apa (pengembang sedang merasa malas)



Gambar 8.8. Lautan luas

Dalam setiap skenario, apa yang terjadi di dekat Anda tidak bergantung pada visual tetapi sangat bergantung pada apa yang Anda dengar atau tidak Anda dengar. Singkatnya, audio bukanlah sesuatu yang bisa dilupakan atau diabaikan, jadi mari kita lihat cara menerapkannya untuk mendapatkan hasil maksimal dari audio.

Audio dalam VR

Pada bagian ini, mari kita lihat lebih dekat bagaimana Unity memungkinkan pengembang untuk menerapkan audio ke dalam aplikasinya. Metode saat ini secara tradisional telah digunakan oleh pengembang gim 2D. Akibatnya, sebagai pengembang VR, kita perlu melakukan beberapa langkah tambahan untuk membuat audio di lingkungan VR terdengar lebih mendalam.

Audio ambient versus 3D versus spasial

Di dalam Unity, ada tiga jenis audio utama:

Audio ambient

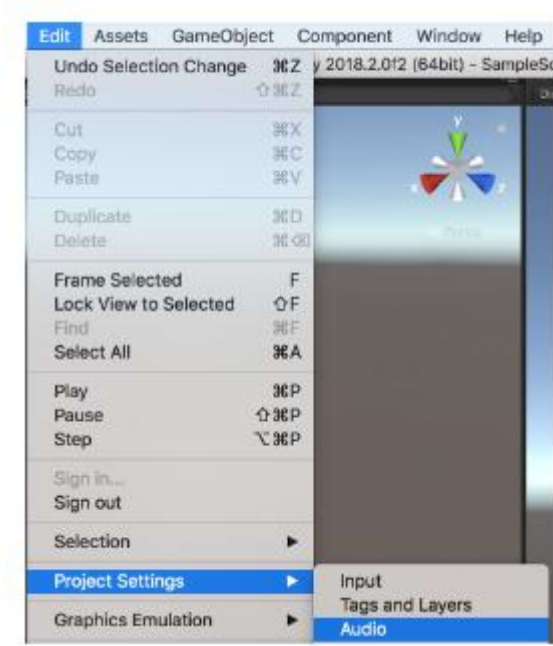
Audio 2D yang tidak bergantung pada lokasi pemain. Anggap saja ini seperti kicauan burung yang jauh di hutan.

Audio 3D

Audio yang terikat pada posisi 3D dan volumenya menurun semakin jauh pemain (pendengar audio) dari posisi tersebut.

Audio spasial (binaural)

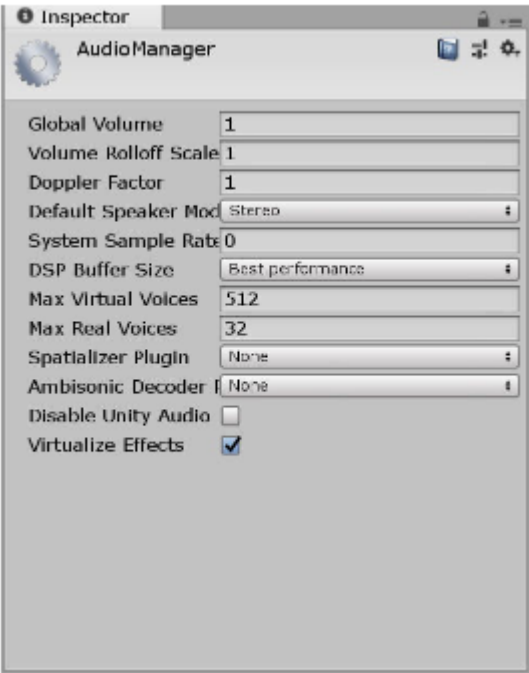
Mirip dengan audio 3D, perbedaan utamanya adalah intensitas saluran audio kiri dan kanan bervariasi tergantung pada seberapa jauh objek dari masing-masing telinga, seperti dalam kehidupan nyata.



Gambar 8.9. Jalur ke pengaturan audio Unity

Setiap jenis audio memiliki tempatnya sendiri 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 berfungsi lintas platform menggunakan Unity. Untuk mempermudah, di sini kami menggunakan plug-in Oculus Spatial Audio yang saat ini sudah terpasang di Unity. Sejak Unity 2108.2, Anda dapat menemukan pengaturan ini di tab Edit (Gambar 8.9), lalu klik Project Settings → Audio → Spatializer Plugin (Gambar 8-10).



Gambar 8.10. Pengelola audio Unity



Gambar 8.11. Komponen Sumber Audio bawaan Unity

Setelah Anda menyetel Spatializer ke Oculus, Anda dapat menambahkan Sumber Audio ke adegan Anda (idealnya, di dekat kamera Anda). Saat Anda melakukannya, opsi Spatialize akan aktif, seperti yang ditunjukkan pada Gambar 8.11.

Klik Spatialize pada sumber audio Anda, lalu impor klip audio yang Anda miliki untuk mengujinya. Untuk mencobanya, jalankan adegan Anda, lalu seret sumber audio Anda di dekat kamera. Jika Anda memakai headphone, Anda akan mendengar audio berpindah dari telinga kiri ke telinga kanan, dan sebaliknya. Jika Anda mengaktifkan VR, Anda akan dapat mengujinya hanya dengan memutar kepala! Cukup rapi, bukan?

Audio dalam AR

Dalam kebanyakan kasus, pengembangan untuk headset AR akan sangat mirip dengan AR seluler. Namun, dalam kasus 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 sarankan pendekatan yang sama sekali berbeda menggunakan audio mono tanpa spasi.

Mono versus stereo

Sebagian besar berkas audio yang Anda temukan daring akan berformat stereo, yang berarti bahwa audio direkam dalam dua saluran, satu untuk telinga kiri dan satu untuk telinga kanan, yang sangat cocok untuk pengguna headphone. Namun, untuk AR seluler, sebaiknya ambil berkas stereo dalam proyek Anda dan ubah menjadi mono; artinya, 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 menghabiskan lebih sedikit ruang, yang penting

pada perangkat seluler apa pun dan terutama penting saat mengembangkan menggunakan AR Studio atau Lens Studio.

Untuk memberikan pengalaman terbaik, saya sarankan Anda menggunakan audio mono dengan sumber audio 3D untuk mengubah audio hanya saat pemain bergerak mendekati suatu 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 mengekspor klip, Anda dapat menggunakannya dalam sumber audio Unity atau program AR apa pun yang Anda pilih.

8.6 PARADIGMA INTERAKSI UMUM

Hukum Ketiga Newton: Untuk setiap tindakan, ada reaksi yang sama dan berlawanan.

Saat Anda menerapkan ini pada VR dan AR, untuk setiap tindakan yang dilakukan pengguna, harus ada respons yang sesuai dengan harapan pengguna. Entah itu melempar objek dan melihatnya terbang, menarik tuas dan membuka pintu rahasia, atau sekadar menoleh dan dunia berubah sesuai dengan itu. Semua tindakan ini termasuk dalam kategori desain gim yang bagus, tetapi, yang lebih penting, semua tindakan ini akan membuat pengguna merasa benar-benar tenggelam dan terhindar dari potensi mabuk simulator.

Selain itu, pelajaran penting lainnya dari desain gim adalah untuk tidak pernah membanjiri pengguna dengan terlalu banyak opsi masukan. Dan konsekuensinya adalah pengembang harus selalu memandu pengguna untuk mempelajari semua opsi masukan sambil tetap membuat pengguna merasa seperti sedang menemukan alat baru. Singkatnya, artinya, tidak peduli apakah pengguna baru saja mengunduh aplikasi Anda atau telah menghabiskan 100 jam di dalamnya, mereka harus selalu tahu apa yang perlu mereka lakukan selanjutnya.

Apa yang kami bangun selanjutnya adalah beberapa interaksi input umum yang terlihat di seluruh judul VR dan AR: sistem inventaris untuk VR, dan raycast dunia nyata layar sentuh untuk AR seluler.

Inventaris untuk VR

Sistem inventaris sering kali menjadi keharusan saat menciptakan 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 luangkan sedikit ruang di depan pengontrol untuk menampung objek kita. Untuk mengaturnya, mari kita buat beberapa slot inventaris (diwakili oleh bola default, seperti yang ditunjukkan pada Gambar 8.12) dan tempatkan di bawah pengontrol kita (Gambar 8.13) sehingga objek selalu mengikuti pengontrol.



Gambar 8.12. Prototipe slot inventaris



Gambar 8.13. Pengaturan untuk inventaris yang menjadi induk pengontrol

Dengan pengaturan tersebut, kita dapat fokus pada skrip kita. Secara total, kita akan memerlukan tiga: Inventory Manager, InventorySlot, dan InventoryItem. Mari kita bahas masing-masing secara lebih rinci:

a. InventoryManager

Ini akan menjadi skrip yang menangani saat pengguna menekan tombol Menu Aplikasi di pengontrol kiri atau kanan untuk memungkinkan kita menyembunyikan atau menampilkan semua slot inventaris untuk pengontrol masing-masing.

b. InventoryItem

Ini akan menjadi skrip yang dilampirkan ke item apa pun yang dapat ditambahkan dan dihapus dari inventaris.

c. InventorySlot

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 ya, 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 disiapkan, kita dapat menambahkan kelas ini sebagai komponen ke setiap pengontrol 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}
}

```

Berikutnya, InventoryItem. Sebelum kita menulis skrip ini, mari tambahkan semua komponen yang kita butuhkan ke objek uji; yaitu, Komponen Rigidbody dan Collider. Untuk mengujinya, kita juga perlu menyertakan SDK, seperti VRTK, agar dapat mengambil objek, dan ini sudah akan disertakan dalam repositori GitHub.

Setelah itu disiapkan, saatnya menulis kode. Metode utama yang akan kita gunakan untuk memicu saat kode mulai berjalan adalah OnTriggerEnter, metode yang dipicu dari Rigidbody yang kita lampirkan sebelumnya. Rigidbody ini memeriksa saat collider yang terpasang pada objek kita bertabrakan dengan collider lain. Saat itu terjadi, objek dapat direferensikan untuk melihat apakah itu slot. Dalam contoh ini, kita memeriksa namanya, tetapi cara lain yang layak untuk memeriksa adalah dengan tag atau lapisan. Jika itu adalah slot, kita kemudian dapat menetapkan item kita ke slot inventaris, yang akan kita 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)}
}

```

Terakhir, tetapi yang terpenting adalah kode untuk setiap slot inventaris. Meskipun mungkin tampak seperti banyak kode, sebagian besar sebenarnya adalah mendapatkan dan menyetel 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 adegan Unity. Dan setelah setiap skrip ini ditetapkan ke Objek Game yang tepat, sistem inventaris kita seharusnya sudah 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, berikut adalah struktur dasar dari sistem inventaris. Pasti ada banyak perbaikan yang dapat kita lakukan (beberapa di antaranya akan dilakukan di GitHub), tetapi ini dimaksudkan sebagai titik awal menuju inventaris yang disesuaikan menurut pengalaman Anda.

8.7 RAYCAST REALITAS TERTAMBAH

Sebagai sebuah konsep, ini sangat penting untuk pengembangan dalam AR. Namun, dalam praktiknya, ini hanya memerlukan beberapa baris kode untuk diterapkan, tetapi ada

juga beberapa baris yang dapat sepenuhnya mengubah perilaku augmentasi Anda jika tidak tepat. Dalam grafik komputer dan juga dalam Unity, sinar didefinisikan oleh titik awal dan arah, lalu digunakan sebagai bagian dari perhitungan raycast untuk menemukan penumbuk fisika pertama (jika ada) yang berpotongan dengan sinar tersebut.

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 tersebut, SDK tersebut 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. Pengguna mengetuk layar untuk meletakkan 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
    RayrayCamera.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. 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
    Rayray=Camera.main.ScreenPointToRay(Input.touchPosit
ion);
//get location of touch
    if (Physics.Raycast(transform.position,transform.forw
ard,
```



```
out hit, Mathf.Infinity)) {
```

3. Seorang pengguna ingin menyeret kursi virtual melintasi ruangan.

Dan yang terakhir inilah yang menjadi titik awal yang rumit. Dengan menyeret, Anda perlu menggabungkan penggunaan raycast fisika bawaan Unity dengan raycast AR. Secara khusus, dalam urutan pelaksanaan:

1. Pengguna mengetuk layar.
2. Menggunakan raycast fisika Unity, periksa apakah pengguna mengetuk objek untuk diseret.
3. Jika objek tersebut dapat diseret, lakukan raycast AR pada setiap panggilan Pembaruan untuk menyeret objek ke posisi baru yang diinginkan pengguna.

Semua ini merupakan interaksi AR yang cukup umum yang diharapkan pengguna dalam aplikasi mereka, jadi sangat membantu untuk mengingat raycast dan mengapa raycast tersebut mungkin tidak berfungsi sebagaimana mestinya saat melakukan debugging.

Kesimpulan

Kami baru saja membahas beberapa kiat yang penting untuk pengembangan VR. Namun sebelum menutup bab ini, saya ingin memberikan satu kiat terakhir sebelum Anda memulai perjalanan pengembangan untuk VR dan AR. Jika Anda pernah mengalami kebuntuan atau frustrasi, ingatlah PRE:

Gairah

Untuk siapa, untuk melakukan apa, dan mengapa Anda membangun proyek yang menakjubkan.

Sumber Daya

Ada banyak sekali sumber daya daring mulai dari komunitas daring di Facebook dan Slack hingga pertanyaan yang dijawab di Google. Kemungkinannya, seseorang juga mengalami hal serupa.

Pengalaman

Anda membangun sesuatu yang unik yang akan memungkinkan orang mengalami sesuatu yang tidak pernah bisa mereka alami. Hargai itu!

Ya, saya tahu akronim ini klise, tetapi jika membantu Anda menyelesaikan proyek, itu akan lebih berharga bagi semua orang. Dan ya, pengembangan VR dan AR memang sulit, tetapi sebagai penulis, kami semua bersemangat untuk melihat apa yang dapat Anda bangun dan kontribusikan kepada komunitas kami yang luar biasa!

BAB 9

DESAIN VISUALISASI DATA DALAM KOMPUTASI SPASIAL

9.1 PENDAHULUAN

Visualisasi data dan pembelajaran mesin mengubah masa depan tempat kerja. Membingkai prinsip desain sangatlah penting. Perusahaan seperti perusahaan yang didukung VR Fund, 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, dalam vertikal B2B fintech (teknologi keuangan), teknologi kesehatan, dan bioteknologi.

Kami memulai bab ini dengan membahas relevansi topik tersebut bagi pengguna yang mengalami aplikasi visualisasi data dan pembelajaran mesin. Kami kemudian menawarkan kerangka kerja untuk dipertimbangkan guna mengidentifikasi tujuan bermanfaat yang menjadikan topik tersebut 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, yang 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 atau desainer perangkat lunak berpengalaman yang sudah terbiasa bekerja pada platform web (Anda dapat dengan mudah menggunakan A-Frame dalam JavaScript atau kerangka kerja lain) atau dalam pengembangan asli, C# pada Unity. Ada beberapa contoh gambar yang dibuat menggunakan C++ dan Unreal Engine.

9.2 MEMAHAMI VISUALISASI DATA

Meskipun whitepaper seperti “Analisis Biaya-Manfaat Visualisasi dalam Lingkungan Virtual (VE)” IEEE mempertanyakan relevansi dan tujuan visualisasi dalam XR, dengan menanyakan “apakah kita benar-benar memerlukan visualisasi 3D untuk data 3D?” Secara sederhana, dasar bab ini mengasumsikan sejak awal bahwa penggunaan VE memungkinkan pemahaman yang lebih baik tentang data 3D, dengan konteks yang sesuai, desain yang cermat, dan pengembangan.

Pertama, kami menjelaskan dan mendefinisikan visualisasi data dalam XR dan apa yang membuatnya unik dibandingkan media sebelumnya. Kami melihat perbedaan antara visualisasi big data interaktif versus representasi infografis murni.

Dianggap sebagai bapak visualisasi data, ahli statistik Edward Tufte menulis bahwa selama berabad-abad pelukis, animator, dan arsitek telah berupaya untuk merepresentasikan

data (2D dan 3D) pada berbagai tampilan (terutama dalam ruang 2D), menggunakan perspektif dan gerakan. Perhatikan bahwa banyak infografis statis tidak memiliki gerakan atau pemahaman umum tentang perspektif dan dengan demikian tidak memenuhi syarat sebagai "visualisasi data yang baik" dalam komputasi spasial; pengalamannya sebagian besar pasif dan tidak memungkinkan rotasi atau prinsip lain yang dibahas nanti dalam bab ini. Visualisasi data dan pembelajaran mesin memungkinkan pengguna untuk melihat, menjelajahi, dan memahami data dengan lebih baik. Seperti yang dikatakan oleh Fernanda Viegas dan Matt Wattenberg (Peneliti Orang dan AI Google yang berfokus pada visualisasi data) di NeuralIPS 2018, visualisasi data mengubah data menjadi penyandian visual yang membantu pengguna mendidik, berkomunikasi, memberikan wawasan, dan menjelajahi data dengan lebih baik.

Tanpa visualisasi, data hanyalah angka mati di halaman. Dalam buku pentingnya, *The Visual Display of Quantitative Information* (Graphics Press, 2001), Tufte menulis bahwa visualisasi data membuat pemahaman manusia tentang set data besar lebih koheren. Ini melayani tujuan yang jelas untuk menggambarkan data yang direpresentasikan dalam berbagai bentuk; misalnya, sebagai abstraksi (diagram pai, diagram batang, dsb.) dan sering kali sebagai istilah untuk menggambarkan rekonstruksi data 3D sebagai objek dalam ruang 3D (misalnya, struktur anatomi yang direkonstruksi 3D seperti data otak, dan irisan datar berkas pencitraan resonansi magnetik [MRI] dalam lingkungan virtual dan augmented). Data itu sendiri bersifat komparatif, relasional, multivariabel, dan dapat memungkinkan pengguna untuk mengajukan pertanyaan tertentu atau menjelajahi data secara umum untuk mendapatkan pemahaman yang lebih baik tentang kualitasnya. Berikut adalah beberapa karakteristik utama visualisasi data interaktif dalam XR:

- Dapat memetakan 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).
- Melibatkan arsitektur informasi yang menunjukkannya sebagai dinamis dan menyediakan aktivitas interaktif bagi pengguna.
- Menekankan estetika untuk membantu pengguna memahami data melalui desain yang baik, bukan hanya untuk tujuan dekorasi.

Data jauh lebih sulit dipahami tanpa visualisasi. Seperti yang dikatakan oleh para peneliti pembelajaran mendalam, "visualisasi data dan analisis visual dapat digunakan untuk unggul dalam komunikasi pengetahuan dan penemuan wawasan dengan menggunakan penyandian untuk mengubah data abstrak menjadi representasi yang bermakna."

Interaktivitas dan animasi pada semua platform, desktop, seluler, dan komputasi spasial membantu pengguna membuat data lebih mudah diakses dan fleksibel dengan manipulasi langsung dengan berbagai set masukan dan kontrol.

Prinsip untuk Visualisasi Data dan Pembelajaran Mesin dalam Komputasi Spasial

Dengan merujuk pada kerangka kerja oleh para peneliti pembelajaran mendalam,⁶ kreator visualisasi data dan pembelajaran mesin dalam komputasi spasial harus mengeksplorasi lima W yang akan membantu memberi mereka dasar untuk menciptakan pengalaman aplikasi yang sukses dalam komputasi spasial. Pembuat konten harus

mempertimbangkan desain pengalaman pengguna mereka dengan memulai dengan hal-hal berikut: mengidentifikasi pengguna target mereka (Siapa) dan di mana visualisasi data tepat digunakan (Kapan), jenis visualisasi data yang dibuat (Apa), membenarkan keberadaannya sebagai yang optimal dalam komputasi spasial dan Mengapa sebelum mereka mengidentifikasi metode atau (jenis visualisasi mana) yang melibatkan atau tidak melibatkan pembelajaran mesin sebelum mereka mulai memilih Di mana 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 yang sebenarnya di akhir bab ini, di mana kami mempertimbangkan rekayasa data-ke-visualisasi holistik dan proses perancangan jalur pipa. Berikut adalah contoh prinsip-prinsip tersebut dalam praktik. Lebih khusus lagi, pembuat konten harus mempertimbangkan faktor-faktor ini agar disengaja tentang proses pembuatan visualisasi mereka:

Mengapa??

Identifikasi tujuan, tanyakan pada diri Anda sendiri mengapa visualisasi data atau pembelajaran mesin ini masuk akal dalam komputasi spasial dibandingkan komputasi lainnya. Pembuat harus mempertimbangkan interaksi sehingga pengguna dapat langsung memanipulasi dan membuka wawasan lain untuk memperoleh pengalaman visualisasi data efektif yang tidak mungkin dilakukan di media lain.

Siapa

Tentukan target pengguna akhir dari pengalaman/aplikasi komputasi spasial visualisasi pembelajaran mesin atau data 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 membahasnya secara terperinci 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 jenis data, seberapa besar data tersebut, dan seberapa besar keinginan mereka untuk memvisualisasikannya. Untuk data MRI khusus 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 konektomik, peneliti di Spanyol memilih untuk memvisualisasikan data multidimensi subset 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 bagian sumber daya di akhir bab ini) pada platform komputasi nonspasial (desktop dan seluler) jika memungkinkan. Pembuat harus memahami kompleksitas data sehingga mereka tahu apakah data tersebut harus diproses terlebih dahulu dan di mana data tersebut disimpan dan ditempatkan (di cloud menggunakan Amazon Web Services, dalam format JSON). Ini mungkin melibatkan atau tidak beberapa pembuatan prototipe (terkadang ini melibatkan penggunaan 3D dengan alat 2D), sebelum memuat dan memvisualisasikan data

sepenuhnya dalam XR.

Bagaimana??

Pilih metode apa yang akan digunakan saat Anda membuat data atau visualisasi pembelajaran mesin. Visualisasi dasar tidak memerlukan banyak pra-pemrosesan, tetapi untuk yang memerlukannya (sering kali yang menggunakan Python) seluruh alur kerja 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 (Unity, Unreal Engine, mesin permainan 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, yang mengarahkan mereka ke keputusan bisnis yang lebih baik. Yang lain mungkin adalah insinyur pembelajaran mesin, ilmuwan data, atau insinyur perangkat lunak yang berusaha menemukan teknik pengoptimalan, mengeksplorasi interpretasi model, dan dapat membuat penemuan ini melalui eksplorasi visualisasi komputasi spasial. Mereka dapat melihat lapisan yang mendasari data multidimensi yang lebih kompleks dalam komputasi spasial dengan lebih baik daripada media lain. Di sini visualisasi berfungsi sebagai solusi untuk "kutukan dimensionalitas," yang berarti memadatkan data multidimensi ke dalam format yang lebih mudah dipahami.

Di ujung spektrum yang lain, beberapa visualisasi sering kali salah dikategorikan sebagai visualisasi data dan sebenarnya termasuk dalam spektrum presentasi dan desain infografis serta dipandang sebagai karya eksperimen artistik yang "indah"; karya-karya tersebut dianggap dibuat untuk tujuan dekoratif semata, atau lebih untuk nilai dan apresiasi estetika daripada untuk penggunaan praktis. Kami membahasnya secara terperinci saat kami menjelaskan kategori data pada Gambar 9.4 di bab ini.

Mengapa Visualisasi Data dan Pembelajaran Mesin Berfungsi dalam Komputasi Spasial

Kami menyelidiki topik ini lebih dalam saat kami menjelaskan evolusi desain visualisasi data, bagaimana tujuannya telah ditingkatkan dan berevolusi 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 kreasi untuk menampilkan bukti. Ia menjelaskan lebih lanjut bahwa dasar desain visualisasi berasal dari prinsip dasar desain analitis yang mendasarinya, yang tidak bergantung pada "bahasa atau budaya atau abad atau teknologi tampilan informasi." Tufte menjelaskan:

Powerpoint seperti terjebak dalam gaya kartun dataran rendah Mesir kuno daripada menggunakan alat representasi visual Renaisans yang lebih efektif. Prinsip pemikiran analitis dan bukan dari adat istiadat setempat, mode intelektual, kenyamanan konsumen, pemasaran,

atau apa yang disediakan oleh teknologi tampilan.

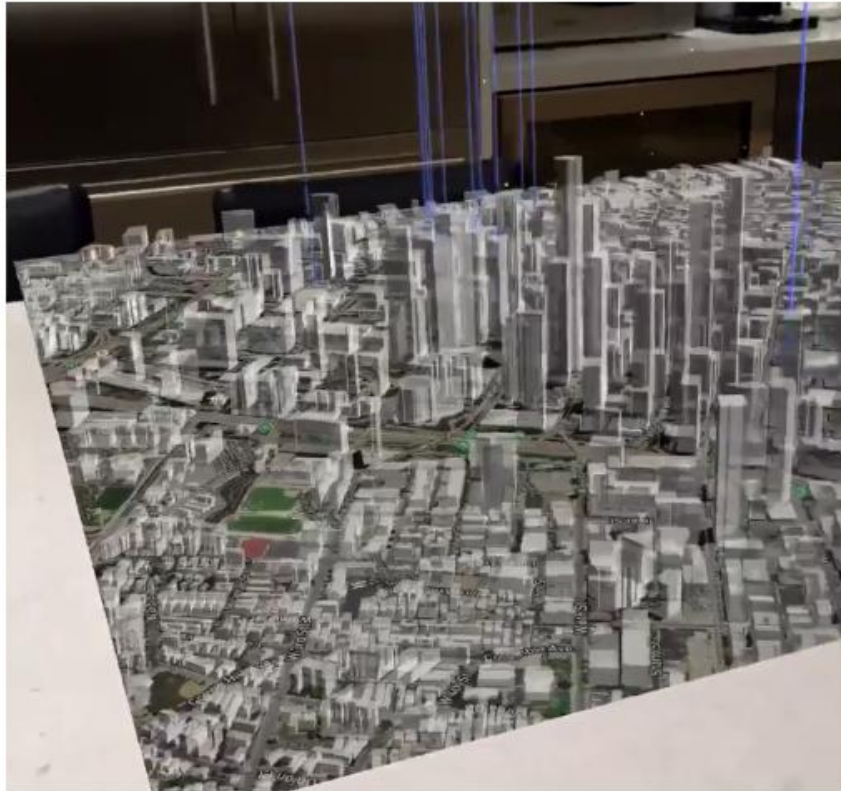
Meskipun ini benar, beberapa praktik terbaik desain visualisasi data (yang terutama dirancang untuk media kertas, desktop, dan seluler) dapat dianggap usang dan tidak semuanya berlaku langsung dalam media spektrum XR karena praktik tersebut memperhitungkan perancangan terutama hanya untuk ruang 2D dengan antarmuka pengguna (UI) datar, bahkan dengan data 3D, atau satu jendela atau layar. Hal 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 komputasional dan kecerdasan buatan (AI) yang menampilkan data multidimensi yang dapat dieksplorasi dengan lebih mudah dengan teknologi baru.



Gambar 9.1. Plot jumlah kali orang menunjukkan bahwa mereka tidak menyukai pembawa acara televisi Inggris Piers Morgan di wilayah tertentu, dibuat dalam kerangka webXR, A-Frame, yang dapat dilihat pada headset VR1

Tufte, seperti banyak ilmuwan data dan akademisi lainnya telah mengkritik visualisasi data 3D yang "buruk" seperti diagram lingkaran 3D dan sebaliknya menawarkan pendekatan yang lebih sederhana untuk visualisasi data, dengan menyatakan bahwa data geospasial 3D dengan peta sederhana di atas kertas 2D sudah cukup. Mereka menolak penggunaan visualisasi 3D sama sekali, tetapi ini salah arah dan terbelakang. Dengan diperkenalkannya AR dan VR, UI peta geospasial 3D telah berkembang pesat sejak saat Tufte menciptakan peta 3D kertas. Konsepsi data baru sekarang dikodekan ke dalam pengalaman aplikasi aktual yang meningkatkan interaksi pengguna dengan data mereka. Misalnya, dalam WebVR, peta cenderung lebih mirip permainan tempat pengguna dapat bergerak dalam bidang z, seperti yang digambarkan dalam Gambar 9.1. Visualisasi data dapat "digamifikasi" ke dalam platform VR seluler (lihat Gambar 9.2) di ARKit dan MapBox.

Ini menggunakan versi lama A-Frame (0.2.0). Beberapa kode mungkin tidak mencapai hasil yang diinginkan. Jika memungkinkan, tingkatkan ke versi terbaru A-Frame (yang saat tulisan ini dibuat adalah 0.9.0). Semua browser utama akan mulai mendukung spesifikasi WebXR pada tahun 2019. API webVR lama akan ditinggalkan.



Gambar 9-2. Pemetaan visualisasi data check-in FourSquare menggunakan ARKit dan MapBox oleh teknolog Aaron Ng

Lebih jauh, input AR dan VR memungkinkan paradigma interaksi baru yang sebelumnya tidak mungkin dalam ruang 2D, yang mendefinisikan ulang interaksi manusia-komputer (HCI) sebagai antarmuka pengguna melalui pengontrol, mulai memuat secara langsung, dan langsung memanipulasi data dengan suara yang lebih canggih (melalui evolusi pemrosesan bahasa alami [NLP]) dan kontrol sentuh (haptik karena keduanya juga terus berkembang).

Banyak yang telah ditulis tentang visualisasi data yang dibatasi pada bidang 2D atau data 3D yang telah terperangkap dalam media 2D (yang sangat membatasi untuk menggali wawasan lebih dalam bagi mereka yang bekerja di bidang bioteknologi dan teknologi kesehatan, dengan data yang berkisar dari anatomi manusia dalam mikroskopi pencitraan medis, visualisasi molekuler DNA, dan visualisasi protein) dalam data abstrak 2D dalam ruang 3D.

Dalam bukunya, *Fundamentals of Data Visualization* (O'Reilly Media, 2018), profesor Universitas Texas dan ahli biologi terlatih, Claus O. Wilke, menekankan posisi dalam XR sebagai bagian dari pembahasannya tentang visualisasi data. Ia berbicara tentang pentingnya berbagai elemen (warna dan garis) dan seterusnya. Fokus bab ini adalah pada posisi, mengingat kemampuan XR untuk menempatkan data sebagai objek pada rotasi sumbu z.

Data 2D dan 3D yang Diwakili dalam XR

Ada berbagai jenis data yang ditampilkan dalam visualisasi data dalam platform komputasi desktop, seluler, dan spasial. Data kategoris yang ditampilkan dalam Gambar 9-4 berkisar dari statis hingga dinamis pada berbagai platform. Jenis data yang sering ditampilkan

dalam XR meliputi berikut ini:

- Abstraksi data 2D yang terlihat dalam 3D dalam XR (sering terlihat sebagai diagram batang dan diagram garis)
- Data 3D dari data 2D (struktur anatomi seperti pencitraan fMRI otak yang direkonstruksi beberapa kali agar tampak 3D dan sesuai dengan ruang 2D)
- Data 3D yang ditampilkan dalam ruang 3D, dalam XR (visualisasi molekuler DNA yang terlihat dalam XR)

Setelah Anda memilih jenis data yang sedang Anda kerjakan, Anda dapat memvisualisasikan data tersebut.

Beberapa data kurang masuk akal untuk divisualisasikan daripada data lainnya. Misalnya, Tufte merujuk diagram pai 3D sebagai terbalik. Wilke melanjutkan ini dan secara khusus menunjuk ke arah visualisasi data yang efektif dalam XR yang semuanya tentang konteks.

Wilke berkata, *"...masuk akal untuk menggunakan visualisasi 3D ketika kita ingin menunjukkan objek 3D yang sebenarnya dan/atau data yang dipetakan ke dalamnya."*

9.4 VISUALISASI DATA 2D VERSUS VISUALISASI DATA 3D DALAM KOMPUTASI SPASIAL

Meskipun Tufte telah banyak dikutip menentang penggunaan visualisasi data 3D yang salah (yaitu data abstrak di luar konteks seperti diagram lingkaran, yang menurutnya tidak menambah perbedaan substansial daripada visualisasi 2D) dan animasi mewah, akademisi lain seperti Wilke menunjukkan bahwa ada beberapa nilai dalam visualisasi data 3D dan dalam 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 ia direpresentasikan, tidak cukup. Interaksi demi interaksi akan menghasilkan semacam karya seni, tetapi itu tidak akan selalu menghasilkan pengalaman visualisasi data dan tidak selalu melibatkan data. Harus ada pertimbangan yang cermat dalam pilihan desain di XR.

Namun, rekomendasi untuk menghindarinya sama sekali adalah picik. Jika kita menghindari visualisasi data 3D dalam ruang 2D dan 3D, kita akan merekomendasikan untuk tidak membuat visualisasi hebat seperti *distill.pub*, yang menunjukkan tujuan yang jelas dan meningkatkan pemahaman data oleh tim Google Brain. Kita 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 kompleksitasnya melalui "pengurangan dimensionalitas" disebut t-SNE, dan Analisis Komponen Utama (PCA) dipopulerkan.

Ini menunjukkan bagaimana insinyur dapat memperdalam pemahaman mereka tentang data oleh insinyur pembelajaran mesin terapan dan ilmuwan data. Meniadakan seluruh dimensi dan media karena visualisasi yang dirancang dengan buruk dan menghindarinya sama sekali menunjukkan kurangnya tantangan dalam berpikir. Jenis pemikiran ini membatasi pikiran kita untuk dibatasi pada perkakas terbelakang yang membuat

manusia menjauh dari teknologi dan semakin jauh dari data yang dapat membantu kemanusiaan. Sebaliknya, kita perlu memanfaatkan teknologi untuk lebih dekat dengan data kita dan memecahkan masalah manusia. Oleh karena itu, visualisasi data 3D dalam komputasi spasial harus didorong tetapi dengan tetap memperhatikan prinsip desain yang tepat. Seperti yang dikatakan Tufte, "Kita harus menjaga data melalui integritas." Hal ini dimungkinkan dalam media baru; kita hanya perlu mendorong batas-batas imajinasi pikiran kita dan membuatnya relevan dengan memperbarui desain, kegunaan, dan standar kita 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 demi seni, yang umum terjadi pada banyak desain infografis yang tidak memiliki interaksi dinamis; ini bersifat sekunder. Setiap pemikiran artistik seperti teori warna harus diterapkan untuk membantu membuat data lebih mudah dipahami (misalnya, ceramah Viegas yang mengingatkan kita untuk mempertimbangkan buta warna dan pilihan warna untuk mengkodifikasi berbagai kunci, peta, dan kategori). Affordance 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 hal lainnya." Pada saat Tufte menulis ini, komputasi spasial belum dapat digunakan sebagai teknologi untuk visualisasi data seperti sekarang dan belum tentu merupakan sesuatu yang akan diperhitungkannya.

Dalam ruang 3D, desainer dan insinyur perangkat lunak harus memikirkan substansi dari apa yang mereka ciptakan dan juga cara menciptakannya dengan media baru. Dengan memahami bagaimana visualisasi data terbentuk, kita dapat lebih memahami persepsi manusia, mengapa desain berhasil, dan menciptakan produk visualisasi data yang lebih efektif dalam media yang sama sekali baru. Pemirsa juga dapat menghargai bagaimana mereka dapat mengendalikan data mereka secara lebih intuitif dan bagaimana hal itu memperdalam pemahaman mereka tentang substansi karena metodologi estetika desain (desain grafis) dan teknologi produksi grafis (bagaimana data tersebut diciptakan dalam komputasi spasial).

Interaksi baru ini, yang hanya mungkin terjadi dalam komputasi spasial, membuka wawasan baru karena dapat melihat dan memanipulasi data dalam ruang 3D tidak seperti paradigma desain sebelumnya. Pembuatan konten 3D, misalnya, pada layar 2D sering kali setengah mundur, mengingat waktu tunda yang terlibat dalam memutar data 3D ke proyeksi layar yang di dunia nyata terkadang lebih baik untuk putaran umpan balik yang lebih cepat melalui pembuatan objek 3D yang sebenarnya (misalnya, media pahatan dan pembuatan prototipe kertas), dan visualisasi data dan mesin dalam komputasi spasial tidak terkecuali.

Komputasi spasial memungkinkan lebih banyak mekanisme untuk memanipulasi data secara langsung dan menawarkan kepada kreator komputasi spasial kemampuan untuk

mempelajari paradigma desain baru dalam HCI untuk lebih memungkinkan pemahaman yang lebih dalam tentang kognisi manusia serta meningkatkan kemampuan untuk memahami dan mengungkapkan wawasan baru tentang data bagi pemirsa atau pengguna.

Banyak peneliti dalam ilmu saraf, teknologi kesehatan, dan bioteknologi mengungkapkan bahwa mereka dapat memajukan pemahaman mereka tentang otak manusia dan penyakit neurodegeneratif melalui kemampuan mereka untuk merekonstruksi, memanipulasi, dan berinteraksi dengan data 3D. Data ini, ketika terkunci pada layar 2D, tidak memiliki sarana apa pun yang memungkinkan manipulasi yang lebih langsung. Ini masih merupakan metode interaksi yang lebih disukai daripada harus menggunakan abstraksi lain melalui keyboard, mouse, atau alat lain sebagai penghalang. Komputasi spasial membantu mempercepat produktivitas pengguna dan mengungkap wawasan baru yang tidak mungkin dilakukan jika mereka berinteraksi dengan data hanya dalam bentuk 2D dengan desktop dan perangkat seluler.

Animasi

Animasi tidak selalu penting untuk pengalaman yang bersifat interaktif setelah menerima masukan dari pengguna. Namun, kurangnya respons terhadap masukan dari pengguna berada di antara batas visualisasi data dan infografis. Animasi saja, dengan apa yang disebut Tufte sebagai "dekuantifikasi,"¹³ dan penghapusan data dan data kuantitatif akan kurang sejalan dengan visualisasi data dan infografis dan mungkin lebih merupakan karya seni yang sering digunakan dalam dunia eye candy dan sering dikelompokkan dengan visualisasi data, yang belum tentu merupakan deskripsi yang akurat atau tepat. Rekonstruksi 3D semata 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 UX responsif atau animasi yang signifikan tetap statis dan tidak menggunakan media komputasi spasial dan mungkin juga dianggap sebagai karya seni daripada visualisasi. Bagan garis data saham dalam komputasi spasial sering dikritik seperti bagan pai 3D yang dibenci Tufte. Jenis visualisasi ini dilihat sebagai peningkatan yang sangat kecil dalam komputasi spasial karena dianggap menambahkan dekorasi yang tidak perlu dan asing dalam komputasi spasial. Namun, cara bagan garis dirancang, ditampilkan, dan divisualisasikan harus dievaluasi sebagai ukuran keberhasilan visualisasi, bukan media yang harus diabaikan sepenuhnya atau jenis visualisasi itu sendiri yang harus dipertimbangkan. Kita juga harus mengevaluasi bagaimana pengguna berinteraksi dengan pengalaman tersebut untuk menemukannya berguna untuk lebih memahami wawasan data.

Pengabaian 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 memiliki pandangan pengguna dari semua sudut dan memutar data mereka secara lebih alami seperti yang akan mereka lakukan dengan objek 3D di dunia nyata. Visualisasi data yang berhasil dapat dicapai jika kita memikirkan dengan baik kedua fungsi yang bermanfaat bagi pengguna akhir, yang alur kerjanya ditingkatkan dalam ruang 3D dibandingkan dengan layar 2D dengan

interaksi manusia yang masuk akal bagi pengguna akhir.

Sistem Virtualitics dan 10K (Gambar 9.3) masing-masing memiliki peta yang dirancang dengan baik yang merupakan demonstrasi yang baik dari visualisasi data XR dalam konteks, seperti komputasi 3D lainnya yang tidak dalam komputasi spasial (deck.GL, dll.).



Gambar 9.3. Sistem 10K, misalnya, memiliki peta hebat yang memvisualisasikan data surya yang menunjukkan bagaimana visualisasi data VR bekerja secara efektif dalam konteks dan lebih dari sekadar konsep abstrak

Kegagalan dalam Desain Visualisasi Data

Beberapa insinyur dan desainer perangkat lunak baru telah menciptakan visualisasi data yang buruk dalam komputasi spasial yang telah menjauhkan 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 menunjukkan 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 dalam XR:

- Mereka tidak memuat data dengan cara yang koheren dan mudah dipahami.
- Mereka terutama menggunakan UI datar yang seharusnya 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 yang berlawanan dari apa yang dimaksudkan untuk memungkinkan visualisasi data.

Banyak visualisasi data dalam XR gagal karena mereka tidak menggunakan ruang dan media 3D yang sebenarnya. Ruang ini menyediakan kemampuan bagi pengguna yang tidak tersedia dengan UI 2D (misalnya, terbatas pada satu layar atau lembar kertas dalam kehidupan nyata). Visualisasi data yang buruk dalam komputasi spasial gagal menunjukkan tujuan yang jelas

mengapa data ini direpresentasikan dengan lebih baik dalam komputasi spasial dan tidak memungkinkan pengguna 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 dengan mudah mengontrol data mereka dengan interaksi pengontrol sentuh yang sederhana (suara saat ini masih terlalu dini, yang banyak diharapkan di masa mendatang akan berubah seiring kemajuan NLP dan AI dengan XR).

9.5 DESAIN VISUALISASI DATA YANG BAIK MENGOPTIMALKAN RUANG 3D

Banyak aplikasi visualisasi data dalam XR tidak memanfaatkan ruang 3D dan sering kali memiliki UI datar. Dalam Pedoman Antarmuka Manusia (HIG) Apple, disarankan untuk menggunakan ruang sepenuhnya dan menghindari kontrol UI yang terlalu rumit: Gunakan seluruh tampilan. Dedikasikan layar sebanyak mungkin untuk melihat dan menjelajahi dunia fisik dan objek virtual aplikasi Anda. Hindari mengacaukan layar dengan kontrol dan informasi yang mengurangi pengalaman mendalam.

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 gagal dalam konsep desain ini dan pengguna menjadi bingung. Alih-alih menghemat waktu pengguna untuk memperoleh pemahaman tentang data yang mereka alami, mereka justru kesulitan mencari 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 menghalangi pengguna untuk mengadopsi aplikasi data dalam teknologi yang baru muncul dan mendalam. Dalam buku John Maeda, *Laws of Simplicity* (MIT Press, 2006),⁸ ia menyatakan bahwa desain pada akhirnya harus menghemat waktu pengguna dengan UI yang 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 yang benar-benar baru dan visualisasi data yang baik menyampaikan desain yang cermat melalui pengalaman pengguna (UX) yang positif dalam aksesibilitas dan kemudahan penggunaannya serta kemampuannya untuk menambah kemampuan pengguna dalam memahami data dengan lebih baik.

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 pada berbagai platform dan bagaimana data tersebut 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 kelompok khusus: satu yang lebih abstrak, dan yang lainnya merupakan representasi data secara harfiah melalui rekonstruksi 3D, yang melibatkan langkah praproses yang lebih berat, terutama dalam pencitraan medis.

Jenis-jenis visualisasi data

Seperti yang telah disebutkan sebelumnya, ada berbagai jenis data yang memenuhi syarat sebagai visualisasi data, yang dapat kita kategorikan ke dalam beberapa area berikut:

- Abstraksi data diagram garis, diagram pai, grafik mulai dari data dasar (x, y) yang diplot dalam bidang z dari data dunia nyata
- Rekonstruksi 3D berdasarkan data kehidupan nyata (bioteknologi, teknologi kesehatan, rekonstruksi biologi manusia) dengan animasi dan interaktivitas

Banyak aplikasi lain dalam bidang AR, khususnya tampilan yang dipasang di kepala (HMD) Microsoft HoloLens mengambil rekonstruksi 3D dari suatu objek seperti MRI otak manusia dan menyatakan bahwa ini adalah visualisasi data. Meskipun ini adalah representasi data dari data dunia nyata dalam AR atau VR, bukan abstraksi data seperti diagram pai, diagram garis, grafik, dan sebagainya, ini sering kali disalahartikan dengan objek 3D itu sendiri yang dapat dengan mudah diunduh melalui TurboSquid atau Sketchfab.

Namun, hal ini tidak serta merta memenuhi syarat sebagai visualisasi data, karena beberapa di antaranya hanyalah interpretasi atau gambar anatomi manusia yang longgar tetapi tidak terhubung dengan data dunia nyata dan terkadang tanpa interaksi pengguna. Beberapa aset ini hanyalah rekonstruksi 3D dan, dengan demikian, tidak dapat dikategorikan (karena sering salah dikategorikan) sebagai visualisasi data; sebagian besar di antaranya hanyalah interpretasi seniman dari mata telanjang dan tidak selalu berdasarkan data dunia nyata. Perbedaan ini menarik garis antara keunggulan grafis dan insinyur yang memisahkan desain dan seni dari rekayasa. Visualisasi data sering kali merupakan hasil dari dua bidang yang menjadi penyebab produksi hebat dari pengalaman mengagumkan dalam komputasi spasial, tetapi kita juga harus membedakan antara "seni demi seni" dan terlihat cantik, sementara yang

lain benar-benar dipengaruhi oleh substansi dan konten 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.

Menetapkan Perbedaan dalam Visualisasi Data dan Visualisasi Big Data atau Machine Learning

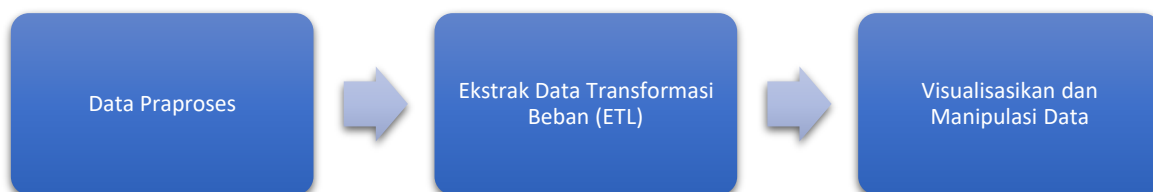
Visualisasi data mengungkap wawasan untuk memahami data tak terstruktur dan data terstruktur. Banyak perusahaan big data menangani kumpulan data besar, terkadang dalam skala terabyte hingga petabyte, seperti yang dijelaskan oleh tim Deck.GL sumber terbuka Uber dengan presentasi WebXR-nya di acara temu muka Silicon Valley Virtual Reality (SVVR) dan Konferensi At Scale Facebook.

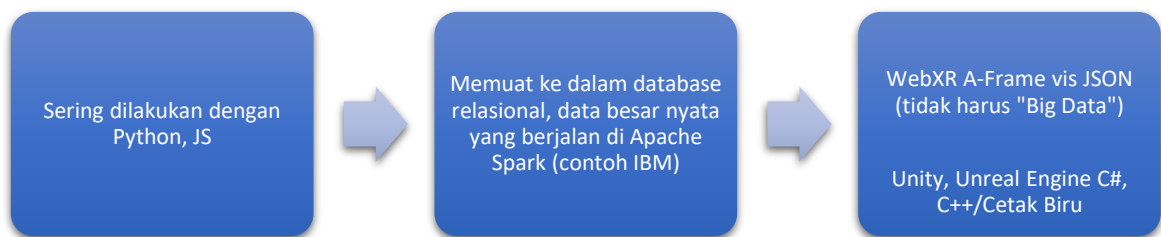
Aplikasi dan pengalaman yang melibatkan big data secara real time dan kompleksitas arsitektur informasi serta jumlah komputasi membuatnya berbeda dari diagram dan batang infografis sederhana yang bukan visualisasi big data, tetapi visualisasi atau representasi data sederhana yang dapat dibuat dengan mudah tanpa memperhatikan keterbatasan yang dapat dirender HMD secara real time. Visualisasi data 3D dalam ruang pembelajaran mendalam (cabang pembelajaran mesin) serupa tetapi juga berbeda dengan yang ada di ruang XR mengingat bahwa visualisasi tersebut dapat melibatkan sejumlah besar data yang divisualisasikan dan bersifat 3D dan tidak semua visualisasi data dalam XR dapat melibatkan visualisasi data berukuran besar atau cakupannya.

Selain itu, "big data" sebagai istilah itu sendiri sering digunakan secara bergantian untuk mendefinisikannya sebagai besar karena mengukur data kuantitatif oleh jutaan pengguna, tetapi tidak selalu demikian. Sebagian besar data teknologi kesehatan dianggap sebagai big data atau digunakan untuk visualisasi mesin dan data dan mungkin tidak selalu besar dalam jumlah pengguna, tetapi diukur berdasarkan ukurannya sendiri.

9.6 CARA MEMBUAT VISUALISASI DATA: ALUR PEMBUATAN VISUALISASI DATA

Alur pembuatan visualisasi data dari pengembangan asli dan pengembangan web melibatkan praproses data (yang sering kali merupakan pekerjaan membosankan yang dilakukan di antara para insinyur data, terutama untuk pemrosesan gambar di ruang medis). Hal ini dikenal sebagai Ekstraksi, Transformasi, dan Pemuatan (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 tersebut.

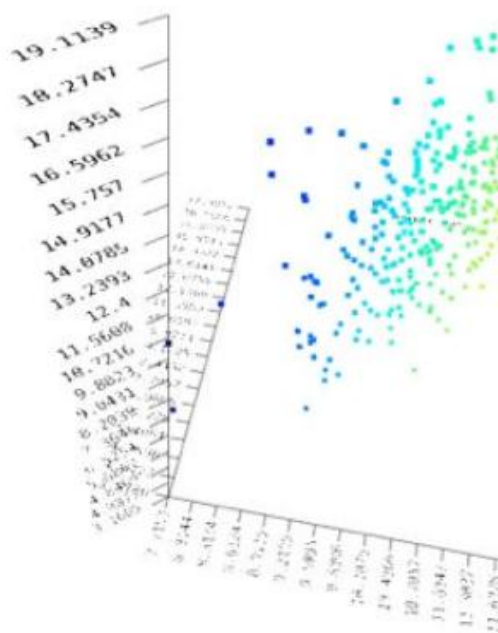




Gambar 9.5. Alur kerja visualisasi data, yang menunjukkan tempat data mentah dimuat, diubah, dan divisualisasikan

WebXR: Membangun Visualisasi Data untuk Web

WebXR telah memperkenalkan beberapa kerangka kerja web selama beberapa tahun terakhir, mulai dari React 360 milik Facebook (sebelumnya dikenal sebagai ReactVR, berdasarkan kerangka kerja frontend ReactJS yang terkemuka) dan A-Frame yang lebih populer, untuk membangun pengalaman realitas virtual. A-Frame, yang dibuat oleh Diego Marcos dan Kevin Ngo (dengan dukungan Mozilla), menjadi kerangka kerja yang populer di komunitas kursus terbuka. Pengembang web telah membuat visualisasi data dengan ReactJS, d3.JS (pustaka visualisasi data andalan 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 adegan A-Frame yang dimuat ke dalam halaman web. Hal ini tidak mesti dianggap sebagai “visualisasi data besar,” tetapi tetap saja merupakan visualisasi data yang dapat dilihat oleh pengguna melalui headset VR di dalam browser.



Gambar 9.6. Zac Canter memetakan data suhu permukaan laut di WebXR menggunakan A-Frame

Tantangan Visualisasi Data dalam XR

Di seluruh spektrum komputasi spasial, konsep input merupakan salah satu masalah desain yang paling menarik, unik, dan mutakhir. Suara kurang menjadi fokus karena kurangnya kematangan dengan NLP canggih dari Siri dan Cortana yang mampu menarik set data besar secara real time saat ini, yang banyak diantisipasi akan berubah dalam beberapa tahun mendatang. Keterbatasan saat ini (tetapi juga evolusi yang diantisipasi akan datang) dalam pengontrol, haptik, dan suara yang ingin diatasi oleh para insinyur dan desainer perangkat lunak dengan eksperimen dan solusi baru.

Meskipun fokus di sini kurang pada suara, pengontrol dalam VR dan kemajuan dalam teknik visi komputer untuk meningkatkan haptik dalam AR, keduanya kini menyediakan kemudahan bagi pengguna yang sebelumnya tidak mungkin dilakukan dalam VR selama beberapa dekade terakhir, dan AR selama beberapa tahun terakhir. Perkembangan HCI ini menimbulkan tantangan baru bagi para desainer dan insinyur perangkat lunak dalam teknologi imersif dan yang sedang berkembang, dengan berbagai kasus penggunaan untuk AR, VR, atau MR dalam spektrum XR.

Berikut ini beberapa tantangan di seluruh spektrum komputasi spasial yang khusus untuk setiap jenis platform mengingat keterbatasan dan interaksinya yang kami perkirakan akan berubah di masa mendatang seiring dengan kemajuan media secara keseluruhan:

AR

Ada dua jenis yang perlu dipertimbangkan di sini: MobileAR (yang hampir tidak memiliki interaksi, jadi tidak masuk akal untuk melakukan banyak hal di sini) dan PCAR (masih tidak nyaman secara ergonomis dan belum siap secara komersial untuk sebagian besar HMD).

VR

VR membatasi pengguna dari dunia nyata dan menempatkan mereka dalam lingkungan tertutup saat berada di HMD. Karena itu, banyak desainer UX dan konsumen merasa bahwa HMD tidak dapat digunakan dan tidak dapat diakses. Sebagian dari ini disebabkan oleh keterbatasan teknis dengan optik meskipun diklaim siap untuk konsumen. Karena tantangan aksesibilitas ini dalam ergonomi, banyak aplikasi memiliki durasi pengalaman yang lebih pendek untuk menghindari kelelahan.

Contoh Kasus Penggunaan Industri Visualisasi Data 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 masih memiliki keterbatasan desainnya sendiri. Gambar 9.7 menunjukkan visualisasi data IBM, yang memanfaatkan kerangka kerja data sumber terbuka, Apache Spark untuk memuat data (termasuk data sumber terbuka, khususnya analisis Sentimen Twitter) secara real-time.

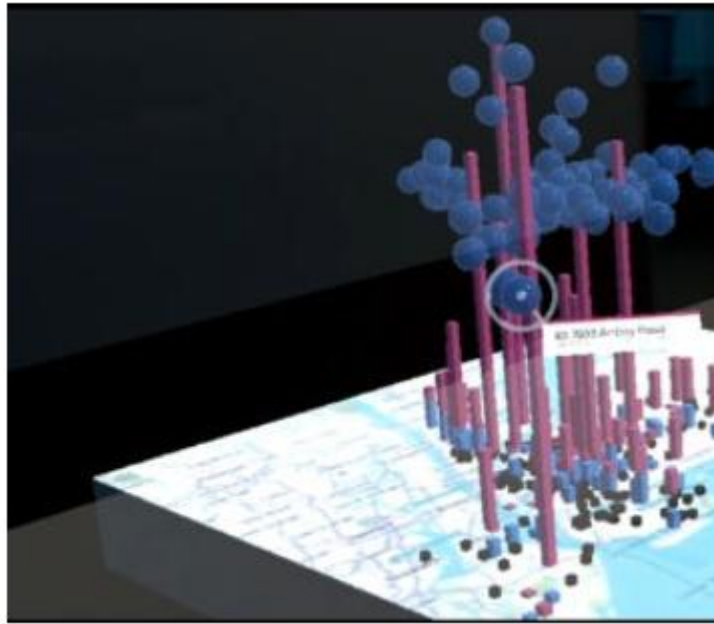


Gambar 9.7. Visualisasi data IBM di Microsoft HoloLens menggunakan kerangka kerja data sumber terbuka

Dalam ceramahnya di GDC 2017 “Immersive Data Visualization: AR in the Workplace,” Ros- stin Murphy, sebelumnya dari IBM, menyajikan penelitiannya tentang penggunaan AR untuk memvisualisasikan, menganalisis, dan memanipulasi big data di tempat kerja. Tujuannya adalah untuk melakukan hal berikut:

Gunakan AR untuk melengkapi kotak peralatan ilmuwan data dan meningkatkan kecepatan dan kedalaman analisis mereka. Menjelajahi interaksi VR dalam konteks bisnis menunjukkan hasil yang menjanjikan, tetapi mengungkap tantangan objektif terhadap VR di lingkungan kerja. Tantangan ini mencakup biaya waktu untuk beralih antara perangkat VR dan mouse serta keyboard, mengatur lingkungan virtual 3D di samping antarmuka desktop 2D konvensional, dan memilih model pengontrol perangkat keras dan algoritme interaksi 3D yang tepat. Teknologi AR, meskipun masih dalam tahap awal, memecahkan banyak masalah dengan elegan.

Lebih khusus lagi, dalam perjuangan Murphy dengan ergonomi dan kemudahan penggunaan saat ia beralih dari notebook Jupyter di desktop PC-nya secara nyata untuk mengubah nilai data dengan mengetik di keyboard sambil mengenakan headset HoloLens di kepalanya. Bobot HoloLens yang berat menjadi penghalang 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 status AR saat ini (yang kami harapkan akan berubah di masa mendatang) sulit diadopsi oleh pengguna.



Gambar 9.8. Contoh visualisasi data yang dibuat oleh Rosstin Murphy saat ia berada di IBM, yang melapisi data pada peta

Rekonstruksi 3D dan Manipulasi Langsung Data Dunia Nyata: Struktur Anatomi dalam XR. Banyak visualisasi data berupa rekonstruksi 3D yang memuat data secara real time yang berbentuk struktur anatomi, yang dapat langsung dimanipulasi dan diedit dalam komputasi spasial. Interaksi komputasi spasial yang unik ini meningkatkan efisiensi alur kerja secara keseluruhan untuk berbagai vertikal B2B.

Sebagai bagian dari University of California, San Francisco, pemimpin dan ahli saraf Adam Gazzaley dan Tim Mullen, di lab Gazzaley, Neuroscope, menciptakan visualisasi yang disebut "Glass Brain" yang sering digunakan dalam demonstrasi berbagai modul untuk Meta 2 AR HMD.

9.7 MELIHAT LEBIH DEKAT GLASS BRAIN

Dengan memanfaatkan Unity3D, visualisasi data Glass Brain terdiri dari struktur otak, baik arsitektur jaringan maupun serat, yang diperoleh dari pemindaian otak beresolusi tinggi (MRI dan Diffusion Tensor Imaging [MRI-DTI]). Aktivitas otak secara real-time dan interaksi fungsional antar jaringan ditumpangkan pada struktur otak menggunakan elektroensefalografi (EEG) berdensitas tinggi. Ini adalah contoh hebat pemuatan visualisasi data secara real-time. Visualisasi yang ditangkap, yang dapat Anda lihat pada Gambar 9-9, adalah visualisasi 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)

Modul VR Pencitraan Medis TVA Surg

Universitas Toronto juga memiliki citra Digital Imaging and Communications in Medicine (DICOM) (tanpa warna) yang direkonstruksi dari pemindaian MRI otak manusia dan citra medis lain dari struktur anatomi manusia.

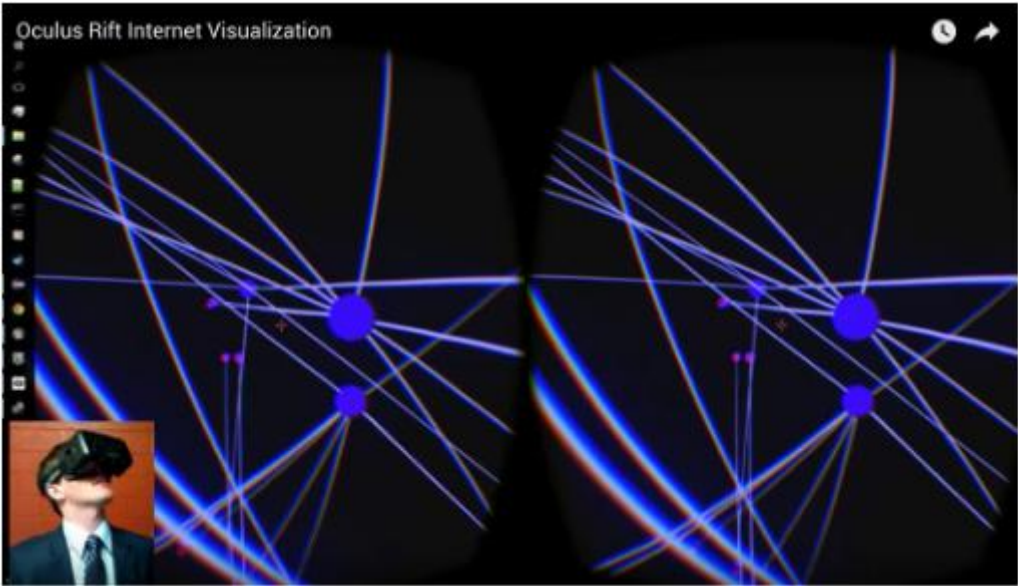
Holodek Medis—DICOM

Holodek Medis, didirikan oleh ahli radiologi Swiss yang menciptakan alat yang memungkinkan ahli radiologi dan profesional medis lainnya untuk mengubah citra DICOM mereka menjadi VR dan dapat mengurangi jumlah waktu yang dihabiskan untuk segmentasi citra (sering kali dilakukan secara manual melalui beberapa algoritme pembelajaran mesin dasar). Ini adalah tugas yang membosankan dalam keseluruhan alur kerja dalam rekayasa data yang melibatkan penguraian data dan rotasi irisan berbagai citra MRI di MATLAB, yang sebanding dengan kerning dan penyesuaian setiap perubahan mikro yang diperlukan dengan banyaknya ekosistem alat desktop 3D yang terpecah-pecah.

Medical Holodeck memungkinkan ahli radiologi, profesional medis, dan peneliti untuk lebih berfokus pada substansi masalah mereka dalam pekerjaan mereka yang melibatkan masalah medis aktual yang sedang mereka hadapi yang ingin mereka analisis, seperti menemukan tumor, membuat sayatan yang lebih tepat untuk operasi, menemukan korelasi lain dengan berbagai patologi, dan meningkatkan efisiensi untuk keseluruhan alur 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 didokumentasikan dengan fasih oleh Dilan Shah dalam Bab 11 buku ini.

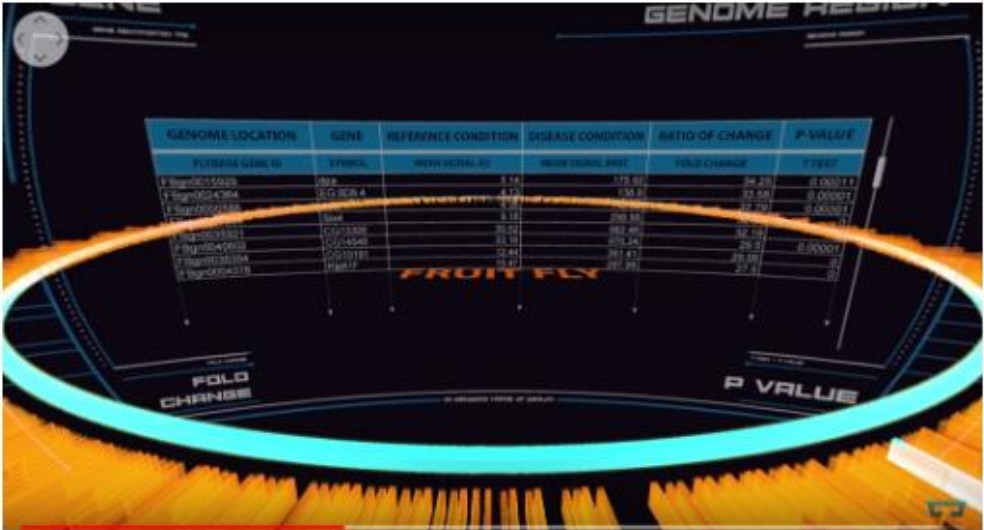
Visualisasi Data untuk Semua Orang: Visualisasi Data Berbasis Open Source dalam XR. Siapa pun dapat memanfaatkan data open source untuk membuat visualisasi data pertama mereka. Jika Anda baru memulai visualisasi data, tidak perlu merasa terintimidasi. Anda dapat

menemukan banyak sekali kumpulan data di Kaggle (sekarang dimiliki oleh Google) dan berbagai area lain (tergantung pada vertikal teknologi) yang memungkinkan pengembang baru membuat visualisasi data yang bermakna. Timothy Clancy juga berhasil membuat visualisasi datanya sendiri dengan mencoba mengindeks segmen berbagai potongan halaman di internet dan menciptakan visualisasi data yang hebat di Unreal Engine, yang dapat Anda lihat pada Gambar 9.10.



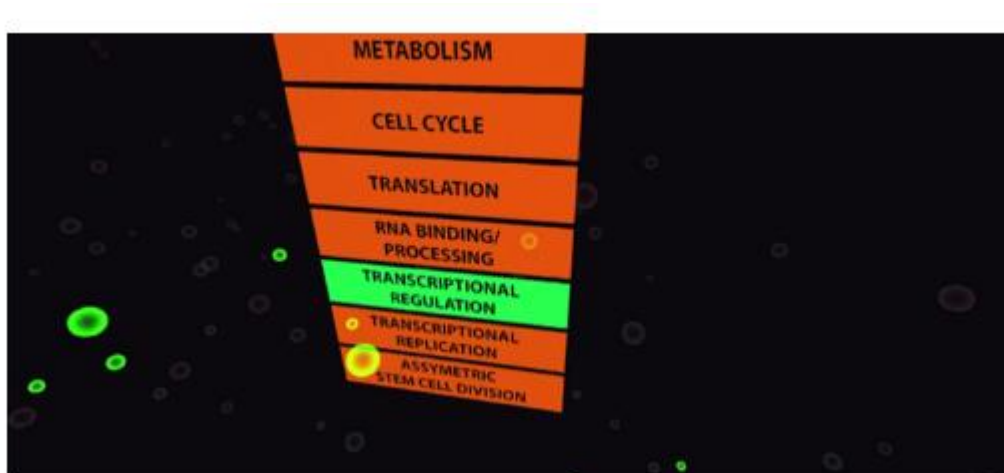
Gambar 9.10. Visualisasi data Timothy Clancy yang menunjukkan subsampel pengindeksan berbagai halaman internet menggunakan Unreal Engine

Pada tahun 2016, Unreal Engine juga menyelenggarakan hackathon dengan data Wellcome Trust dari Inggris (data bioteknologi), di mana pemenangnya, Hammer Head VR, membuat peramban VR yang menganalisis genom lalat buah, seperti yang digambarkan pada Gambar 9.11 dan 9.12.



Gambar 9.11. Visualisasi data ini dibuat dari proyek yang dibuat di hackathon Unreal Big

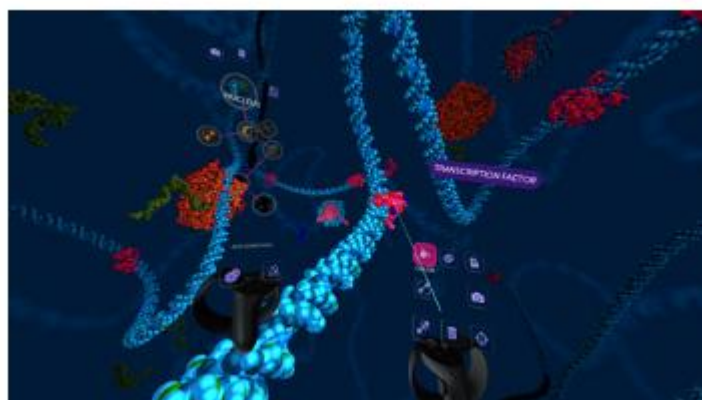
Data yang memiliki UI datar dengan tabel yang menampilkan data yang berasal dari genom lalat buah



Gambar 9.12. Visualisasi data yang dibuat dalam proyek hackathon Unreal Big Data oleh HammerHeadVR ini menggunakan bola-bola yang menyala hijau saat "diarahkan kursor" seperti di web, saat pengguna berinteraksi dengan menu melalui pengontrol VR

Visualisasi Data Protein

Platform 10K dari Dynamoid (lihat Gambar 9.13) juga menciptakan UI dinamis agar dapat melihat Protein Data Bank (PDB) sumber terbuka sebagai referensi untuk berbagai protein DNA dalam aplikasi VR-nya yang memungkinkan pengguna untuk dapat mencari, memasukkan, melihat, protein dengan beberapa kemampuan mengubah ukuran yang berjalan secara real time. Tidak seperti banyak aplikasi teknologi kesehatan dan bioteknologi lainnya yang sering kali berupa rekonstruksi data 3D statis tanpa interaksi atau seni 3D teknis sederhana (file OBJ atau FBX dari tampilan yang dibuat-buat dari pemindaian MRI atau protein DNA), sistem 10K secara dinamis menarik berdasarkan data nyata yang digunakan oleh praktisi di lapangan.

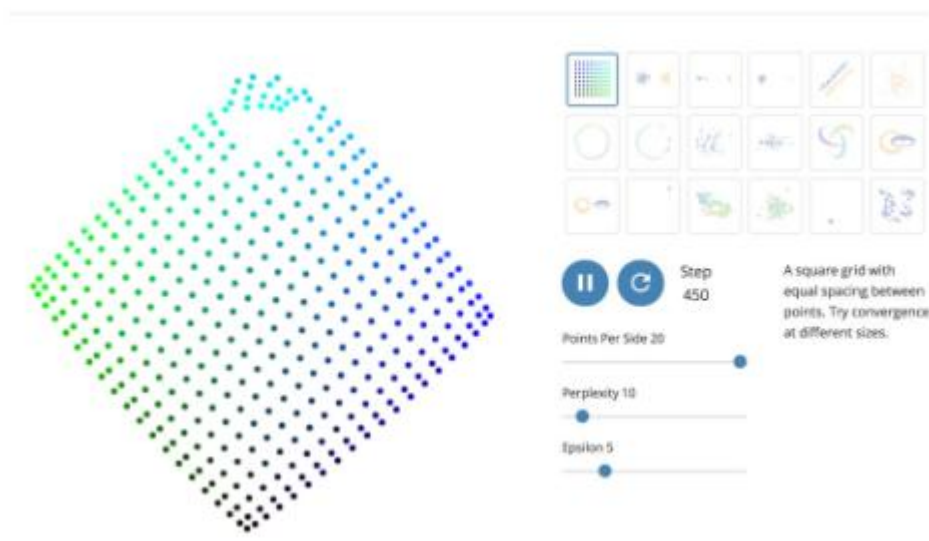


Gambar 9.13. Platform 10K yang didirikan oleh Laura Lynn Gonzalez dari Dynamoid, menunjukkan cara menampilkan visualisasi data dalam VR secara efektif dalam konteks

Tutorial Praktis: Cara Membuat Visualisasi Data dalam Komputasi Spasial

Sekarang setelah 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.

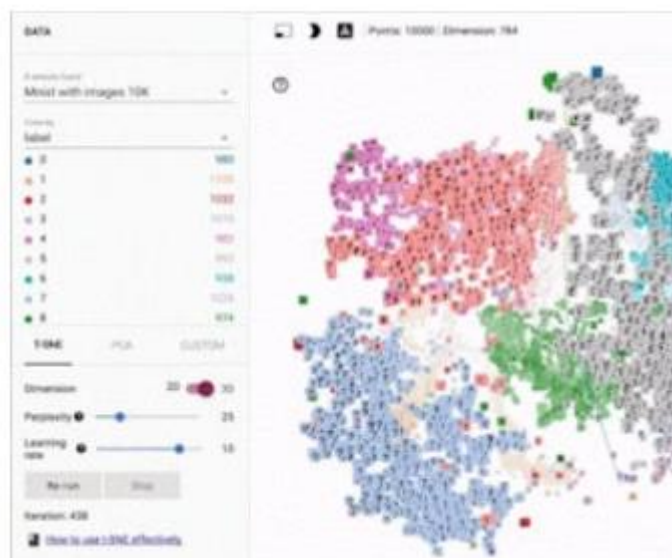
Sisa bab ini menjelaskan berbagai referensi oleh insinyur perangkat lunak baru dan berpengalaman tentang berbagai pendekatan tentang cara membuat visualisasi data dalam komputasi spasial mulai dari platform web dan asli, pada A-Frame, ReactJS, D3.JS (WebXR) menggunakan JavaScript (JS) dan Unity menggunakan C#.



Gambar 9.14. Jurnal pembelajaran mesin distill.pub memiliki banyak visualisasi pembelajaran mesin interaktif yang membantu peneliti lebih memahami data mereka

Untuk mendapatkan pemahaman yang lebih baik tentang visualisasi data menggunakan big data, beberapa contoh terbaik dapat ditemukan oleh ilmuwan data dan insinyur pembelajaran mesin terkemuka yang memamerkan 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 ada dalam komputasi spasial sebagai media, ini memberikan landasan yang kuat untuk memahami visualisasi kumpulan data multidimensi, kompleks, dan terkadang sangat besar, yang memberikan beberapa prinsip dasar kepada kreator baru yang dapat mereka terapkan untuk membuat visualisasi baru dalam komputasi spasial.

Untuk titik awal yang baik dalam memahami kedua jenis visualisasi data yang umum digunakan dalam komunitas pembelajaran mesin, lihat Projector TensorFlow milik Google dan karya Jurnal ML yang didukung YCombinator, distill.pub dengan karya yang juga dibuat oleh lan bersama dengan pemimpin pembelajaran mesin Google, Chris Olah.



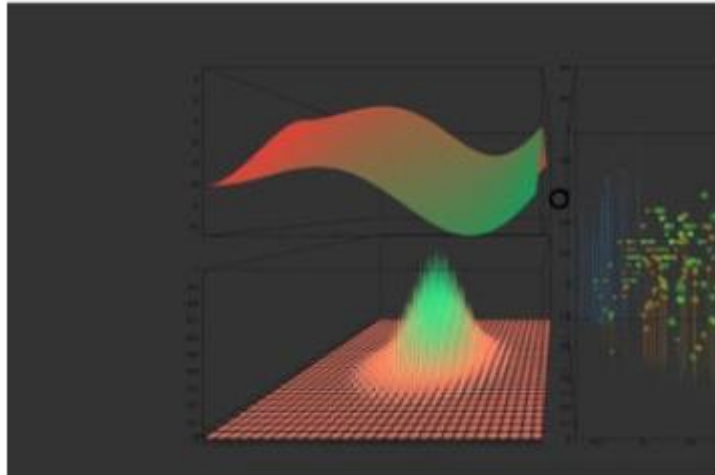
Gambar 9.15. Alat Fernanda Viegas dan Matt Wattenberg dengan kerangka kerja pembelajaran mesin Google TensorFlow, memvisualisasikan Principal Component Analysis (PCA) dan visualisasi lainnya

Di dunia Windows, C# adalah bahasa pilihan dan dengan demikian memindahkan visualisasi data ke Microsoft HoloLens AR HMD, yang umum 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 diperkenalkannya ARKit di iOS dan ARCore di Android dirinci dalam 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 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 Microsoft HoloLens, Magic Leap, Apple, dan lainnya dari Tiongkok dan Israel, dan bahkan yang lain yang mungkin belum kami ketahui yang sedang dalam proses yang diperkirakan akan dirilis pada tahun 2020 atau setelahnya (setelah tanggal rilis awal antologi ini).

Cara Membuat Visualisasi Data: Sumber Daya

Dengan meninjau kembali prinsip-prinsip di awal bab ini, kreator harus mempertimbangkan sejumlah langkah yang akan membantu mereka menentukan pendekatan terbaik untuk visualisasi data yang efektif dalam komputasi spasial. Salah satu contoh hebat untuk memulai WebXR datang dari Mustafee Saifee, yang menciptakan kerangka kerja yang menggabungkan A-Frame dengan React (untuk manipulasi DOM) dan D3 (untuk visualisasi data) untuk menghasilkan visualisasi dalam VR, salah satunya dapat Anda lihat pada Gambar 9.16.



Gambar 9.16. VR-Viz menyediakan komponen reaksi tingkat tinggi untuk menghasilkan visualisasi 3D dalam 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 desain yang solid untuk membangun visualisasi data komputasi spasial dengan sukses.

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 terutama berfokus pada visualisasi data dan hanya sedikit menyinggung visualisasi pembelajaran mesin karena keterbatasan panjang buku ini, sangat dianjurkan untuk melihat repositori GitHub tambahan dan referensi tutorial kami untuk melanjutkan dengan materi praktik untuk memulai pembuatan visualisasi data dan pembelajaran mesin dalam komputasi spasial.

BAB 10

KARAKTER AI DAN PERILAKU

10.1 PENDAHULUAN

Realitas virtual (VR) menghadirkan pengalaman sensorik yang lebih mendalam daripada bentuk hiburan digital sebelumnya (film, gim video, novel interaktif, dll.). Pengalaman yang lebih mendalam berarti pengalaman yang lebih emosional dan berdampak. Banyak dari kita mengingat pengalaman pertama mereka dalam VR dengan emosi. Banyak yang telah menyaksikan kejutan, keajaiban, dan antusiasme yang dapat ditimbulkan oleh pengalaman pertama dalam virtualitas pada orang lain. Dapat dikatakan tidak ada media lain yang dapat menghasilkan emosi dengan intensitas seperti itu hanya dalam beberapa detik. Fakta ini telah diakui oleh komunitas yang kini memanfaatkan kekuatan emosional ini untuk menghasilkan aplikasi VR dalam domain yang sensitif seperti pelatihan pribadi dan terapi.

Dengan menggunakan geolokasi dan menggabungkan elemen realitas dengan elemen virtualitas, augmented reality (AR) dan mixed reality (MR) mendefinisikan taman bermain baru bagi para seniman, pendongeng, dan pengembang gim untuk dijelajahi. Ini sebagian besar masih merupakan wilayah yang belum dipetakan, tetapi kegilaan yang diamati terhadap 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 merupakan bentuk primitifnya) atau mempertahankan jalan Anda dari pemain dari tim lain dengan menempatkan baterai artileri di atap rumah Anda sendiri pasti akan menarik bagi banyak pemain dan non-pemain. Aplikasi profesional AR juga hadir dalam jumlah banyak, mulai dari solusi pengganti desktop, hingga alat operasi lapangan dan alat visualisasi in situ. Kegembiraan seputar AR dapat diukur dari jumlah perusahaan rintisan 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 yang dikumpulkan di bawah akronim “XR” merupakan 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. Hal ini didorong oleh keberhasilan pembelajaran mesin secara umum dan subdivisinya, pembelajaran mendalam, dalam domain penambangan data dan persepsi robotik (penglihatan komputer, pemrosesan bahasa alami [NLP], pengenalan gerakan, dll.) serta dalam pembuatan konten (misalnya, gambar, suara, animasi). Hal ini memperbarui minat terhadap disiplin ini melampaui impian terliar banyak profesional jangka panjang. Kemungkinan aplikasi AI dan pembelajaran mesin dalam XR sangat banyak, dan bab-bab sebelumnya telah membahas beberapa di antaranya (misalnya, Bab 5 oleh Matt Miesnieks dan Profesor Victor Prisacariu, yang membahas tentang penglihatan dan pengenalan). Tujuan bab ini bukan untuk membahas semua potensi aplikasi AI dan pembelajaran mesin untuk pengembangan aplikasi XR. Sebaliknya, kami mengadopsi pendekatan yang berlawanan yang dimulai dari suatu masalah

menghasilkan perilaku dalam lingkungan virtual dan semi-virtual lalu meninjau pendekatan teknis yang tersedia.

Secara historis, penelitian dan pengembangan dalam domain XR sebagian besar telah dipromosikan oleh komunitas game. Hal 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. Di antaranya 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 mewujudkan kreasi mereka. Demikian pula, kami memulai studi kami dengan melihat teknik AI game dan membahas kemampuannya untuk mengatasi tantangan XR. Ini akan membawa kami untuk menekankan beberapa batasannya dan kemudian membahas alternatif yang ada dan upaya penelitian terkini untuk membawa AI game ke tingkat berikutnya. Dalam perjalanan ini, kami harus mengungkapkan pendapat tentang pendekatan yang paling menjanjikan dan ke mana upaya penelitian harus diarahkan. Pendapat ini diarahkan oleh beberapa tahun pengalaman dalam menerapkan sistem pengambilan keputusan dan pembelajaran mesin di bidang industri lain, terutama dalam robotika otonom. Namun seperti pendapat lainnya, pendapat ini dapat menimbulkan perselisihan.

Bab ini membahas pendekatan teknis yang tersedia untuk menghasilkan perilaku dalam video game dan XR. Sebelum memasuki inti masalah, kami membingkai diskusi dengan membahas secara singkat gagasan 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 dikaitkan dengan berbagai jenis entitas: jelas pada satu karakter yang tidak dapat dimainkan (NPC), tetapi juga pada sekelompok NPC atau pada seluruh dunia permainan saat kita mempertimbangkan penceritaan dan narasi interaktif. Anda akan melihat bahwa skala di mana kita mempertimbangkan perilaku memiliki pengaruh lebih besar pada sifat masalah yang kita hadapi dan juga pada pendekatan teknis yang akan diadopsi—daripada objek tempat kita menerapkan perilaku tersebut.

Bagian utama pertama dari bab ini, "Praktik Saat Ini: AI Reaktif," mensurvei teknik-teknik mutakhir yang saat ini mendukung sebagian besar AI permainan. Praktik saat ini sederhana: pengembang permainan harus menulis "dengan tangan" semua perilaku yang ditunjukkan entitas yang dikendalikan saat runtime. Tingkat kecanggihan terendah adalah urutan tindakan yang sepenuhnya ditulis dan tidak dapat diubah yang dieksekusi 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 berbagai peristiwa permainan yang memicu perilaku yang berbeda. Meskipun demikian, pengembang perlu merancang dan mengimplementasikan hubungan dari pengamatan ke

tindakan sepenuhnya dengan tangan. Melihat yang pertama (urutan tindakan tetap) sebagai kasus khusus dari yang terakhir (tindakan yang bergantung pada pengamatan), kami mengumpulkan keduanya dengan nama mesin berstatus terbatas (FSM) dan pohon perilaku (BT) untuk membantu pengembang mengatur pengetahuan dengan cara yang lebih mudah dibaca dan dipelihara daripada kode biasa. Keduanya hanyalah bahasa pemrograman visual. Alasan utama mengapa paradigma ini mendominasi pengembangan game adalah karena paradigma ini memungkinkan kontrol penuh oleh pengembang atas perilaku yang ditampilkan dalam game.

Seperti yang ditekankan di banyak tempat, AI reaktif sangat bagus dalam menjelaskan cara melakukan sesuatu. Namun, hal itu tidak memberikan bantuan apa pun dalam memutuskan apa yang harus kita lakukan. Kami percaya bahwa, karena berbagai alasan, baik AR maupun VR mengajukan tantangan pada model ini. Sebagai konsekuensinya, kita perlu memeriksa alternatif lain yang tersedia dalam AI akademis 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, ia hanya mengulang apa yang telah diperintahkan untuk dilakukan. Mengingat hal ini, kita dapat menulis AI reaktif hanya dalam kode biasa (misalnya, C#, C++, Java). Namun, perilaku biasanya merupakan struktur kompleks yang ingin atau perlu kita uji, debug, tingkatkan, dan tingkatkan saat aplikasi sedang dikembangkan. Kode biasa bersifat samar, dan perancang AI mungkin bukan pembuat kode yang ahli. Karena alasan ini, berbagai alat telah dikembangkan untuk membantu perancang menyusun perilaku.

Hal ini membawa kita ke bagian besar kedua dari bab ini, “Lebih Banyak Kecerdasan dalam Sistem: AI yang Disengaja,” yang membahas tentang AI yang disengaja 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 setiap aspek perilaku secara manual. Pendekatan ini menghindari banyak batasan AI reaktif. Secara khusus, pendekatan ini sangat baik dalam memutuskan apa yang harus dilakukan. Meskipun umum dalam domain robotika otonom, pendekatan ini sebagian besar diabaikan oleh komunitas AI game, dan hanya sebagian kecil produk komersial yang menggunakan teknologi ini saat ini.

Hal ini dapat dijelaskan oleh hilangnya kendali yang menyertai pendekatan ini. Lebih dari sekadar pesaing, kami percaya bahwa pendekatan ini harus dilihat sebagai pelengkap AI reaktif. Perencanaan otomatis harus digunakan untuk memecahkan masalah keputusan yang sulit, sedangkan AI reaktif harus digunakan untuk menggambarkan detail tentang cara mengeksekusi keputusan. Penjelasan lain untuk kurangnya minat ini adalah banyaknya pekerjaan yang diperlukan untuk membuat sistem AI reaktif yang berfungsi, yang membuat seluruh pendekatan tidak layak secara ekonomi bagi banyak studio kecil. Dalam hal itu, upaya yang dilakukan di Unity Technologies untuk menyediakan alat AI deliberatif yang siap pakai

dapat berdampak penting pada pasar. AI deliberatif bekerja dengan memecahkan masalah, yang setara dengan mencari ruang solusi yang besar untuk solusi optimal. Setiap algoritme pencarian memiliki batasan dalam hal ukuran masalah yang dapat ditanganinya sebelum memengaruhi fluiditas permainan/XR dengan menghabiskan terlalu banyak waktu. Untuk menghilangkan batasan ini, kita dapat beralih ke paradigma AI ketiga: pembelajaran mesin.

Pembelajaran mesin menjadi pusat sebagian besar perbincangan seputar AI saat ini, khususnya cabang yang disebut pembelajaran mendalam. Pencapaian spektakuler teknologi ini dalam domain penambangan data dan persepsi robotic memahami keluaran 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 pembentukan perilaku disebut pembelajaran penguatan. Dalam paradigma ini, AI belajar untuk melakukan perilaku yang benar melalui uji coba dan kesalahan, berinteraksi dengan gim video.

Dalam proses ini, AI dipandu oleh hadiah virtual yang diberikan setiap kali AI mencapai tujuan. Hubungan antara disiplin ilmu ini dan gim video secara historis sangat erat. Komunitas peneliti telah mengakui relevansi gim video sebagai tempat uji coba untuk algoritme pembelajaran penguatan. Gim video arkade sederhana kini umum digunakan untuk mengevaluasi kinerja algoritme pembelajaran penguatan baru dalam publikasi akademis. Hal ini tampaknya menunjukkan masa depan yang cerah bagi teknik pembelajaran mesin yang digunakan untuk menghasilkan perilaku dalam gim dan XR. Tujuan bab ini bukanlah untuk memberikan survei yang menyeluruh dan mendalam tentang semua teknik AI permainan. Satu bab saja tidak akan cukup.

Bagian Referensi dari bab ini mencakup survei bidang AI permainan, dan penelaahan mendalam terhadap penelitian terkini. Tujuan bab ini adalah untuk memberikan survei tingkat tinggi tentang pendekatan yang tersedia untuk mengatasi tantangan XR dan untuk menekankan kekuatan dan kelemahannya. Selain itu, kami tidak membatasi studi kami pada "AI permainan resmi", tetapi juga meminjam konsep dari AI akademis. Kami berharap visi ini akan memandu para pengembang XR awal untuk membuat pilihan desain yang tepat dalam hal merancang perilaku.

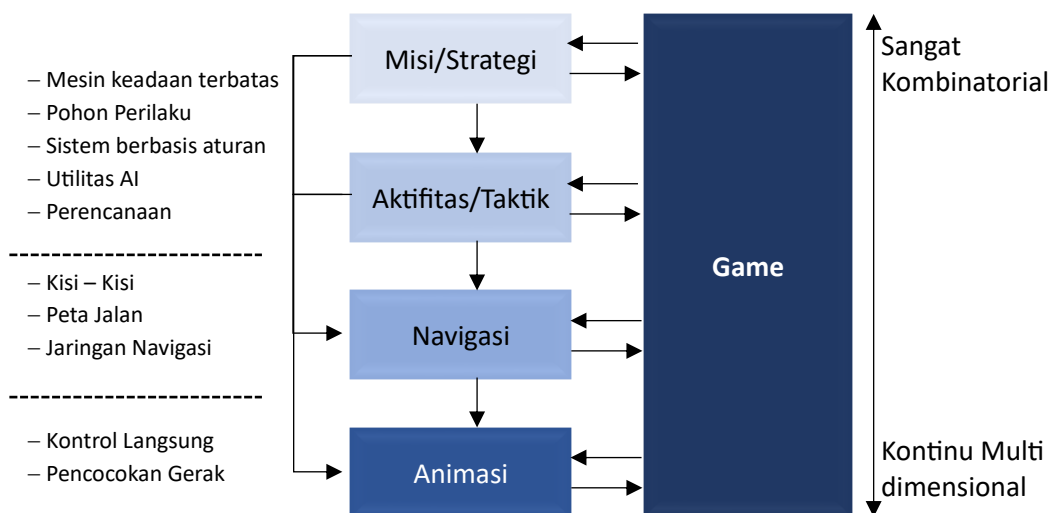
10.2 PERILAKU

Ketika kita mencari kata "perilaku", kita dengan cepat menemukan definisi berikut: "Cara seseorang bertindak atau berperilaku (...)." Jadi, perilaku tersebut 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 untuk memahami masukan dari rangkaian sensor yang dilengkapi dengan robot, dan untuk membangun model dunia yang terkonsolidasi darinya; keputusan, yang memiliki tujuan untuk

memutuskan tindakan berikutnya yang akan dilakukan; dan kontrol, yang mencoba untuk melaksanakan keputusan yang diambil setepat mungkin.

Bab ini berfokus pada masalah yang terkait dengan pengembangan lapisan keputusan. Lapisan keputusan itu sendiri terbagi 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 yang akan dipahami di sini dalam pengertian yang lebih terbatas daripada di bagian lain bab ini yang memutuskan tindakan taktis seperti perubahan jalur dan menunggu di rambu berhenti; dan perencanaan gerakan, yang mencoba mengemudikan mobil sambil tetap berada di jalan dan menghindari rintangan seperti lubang jalan. Arsitektur serupa dapat (dan harus) digunakan untuk NPC dalam gim video. 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 gim video atau aplikasi XR (kiri: teknik yang umum digunakan untuk menangani berbagai tingkat pengambilan keputusan; kanan: sifat ruang masalah berevolusi dari optimasi berkelanjutan ke kombinatorial saat kita menaiki hierarki)

Setiap modul keputusan pada Gambar 10.1 bekerja pada skala tertentu dalam hal waktu dan ruang (skala menurun saat kita beralih dari misi 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 arsitekturnya, semakin rendah frekuensi pengambilan keputusan dan revisi. Dalam gim video atau XR, hanya animasi yang harus diputuskan pada 60 bingkai per detik karena pose karakter yang berbeda harus ditampilkan pada setiap bingkai. Bisa dibayangkan tidak banyak yang tertarik untuk menjalankan sistem navigasi lebih cepat dari 10 bingkai per detik: pengguna tidak akan menyadari bahwa NPC

merencanakan ulang rutanya ke tujuannya 10 kali per detik, bukan 60 kali. Demikian pula, modul keputusan tingkat tinggi dapat bekerja pada frekuensi yang menurun saat kita naik ke atas dalam arsitektur.

Ini adalah fenomena yang sangat alami yang kita semua patuhi: frekuensi kita merevisi keputusan dan membuat rencana baru menurun seiring dengan skala penalaran kita. Kita tidak merevisi rencana karier kita setiap menit, tetapi kita merencanakan ulang tindakan kita beberapa kali per menit untuk beradaptasi dengan situasi saat kita baru saja berkemas untuk berangkat kerja. 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 terkenal dan umum 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 semua jenis keputusan pada frekuensi yang sama saat game dirender.

Hal penting lainnya tentang hierarki pengambilan keputusan seperti yang direpresentasikan dalam Gambar 10.1 adalah bahwa sifat masalah yang kita hadapi untuk menghasilkan perilaku yang baik berubah saat kita bergerak melalui arsitektur. Tugas tingkat rendah yang dekat dengan keterampilan sensori-motorik biasanya dibingkai dalam ruang kontinu multidimensi. Perencanaan gerakan dalam robotika dan pembuatan animasi dalam video game dan XR melibatkan sejumlah besar variabel kontinu dan dapat dibingkai sebagai masalah pengoptimalan kontinu berdimensi tinggi. Sebaliknya, perencanaan aktivitas tingkat tinggi sebagian besar tentang pencarian ruang diskrit (nonkontinu) yang sangat kombinatorial. Mereka biasanya melibatkan kumpulan objek, lokasi, dan/atau konsep diskrit, dan kesulitan utamanya adalah banyaknya cara untuk menggabungkannya.

Misalnya, jika kita memutuskan untuk menjatuhkan sebuah objek dari ransel dan menyimpannya di suatu lokasi untuk digunakan nanti (mungkin karena kita membutuhkan ruang di dalam tas), mungkin ada sejumlah besar kombinasi objek dan lokasi penyimpanan yang perlu dipertimbangkan. Keadaan menjadi lebih buruk jika waktu memengaruhi kapasitas penyimpanan, atau jika objek yang akan ditambahkan ke dalam tas harus diperhitungkan dalam keputusan tersebut. Setiap variabel yang ditambahkan akan melipatgandakan kompleksitas dengan jumlah opsi yang tersedia untuk variabel ini, sehingga menciptakan pertumbuhan eksponensial. Fenomena ini umumnya disebut ledakan kombinatorial. Pergeseran dari ruang kontinu multidimensi ke ruang diskrit yang sangat kombinatorial seiring dengan peningkatan skala dapat diamati dalam banyak domain aktivitas manusia.

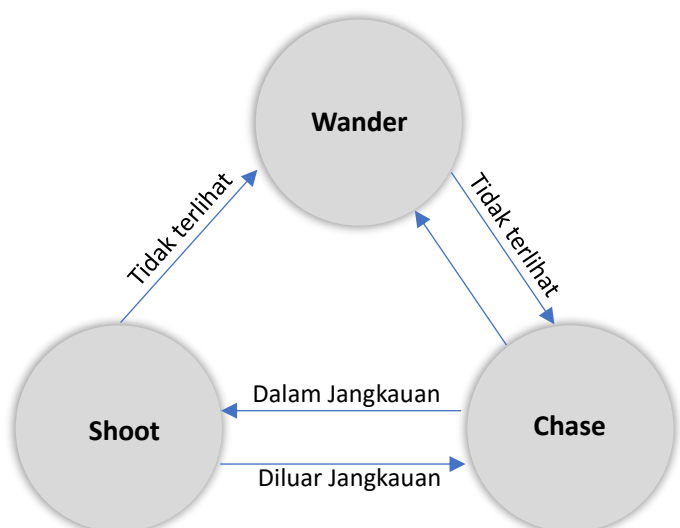
Sebelum menutup bagian pengantar ini, kami juga menekankan bahwa subjek perilaku yaitu, entitas yang menunjukkan perilaku dapat mengambil banyak bentuk. Kasus yang paling jelas adalah NPC dalam permainan peran (RPG). Namun, Anda juga dapat mempertimbangkan pasukan NPC musuh dalam gim first-person-shooter (FPS), atau seluruh pasukan musuh dalam gim strategi waktu nyata (RTS), sebagai subjek perilaku. Lebih jauh lagi, kita melihat penceritaan sebagai serangkaian tindakan yang dilakukan oleh seluruh dunia gim (atau, jika Anda lebih suka, modul manajemen skenario yang dapat bertindak pada skala seluruh dunia

gim). Memang, sebagian besar alat dan pendekatan teknis yang tersedia untuk narasi interaktif mirip dengan yang digunakan untuk NPC individual. Akhirnya, setiap aktor menghasilkan tindakan dan dengan demikian memiliki perilaku. Jadi, jika objek gim yang mewakili objek dunia nyata yang biasanya tidak bernyawa dapat mengambil tindakan, itu berkaitan dengan konten bab ini.

Praktik Saat Ini: AI Reaktif

Jika Anda membuka buku tentang AI game, atau menghadiri AI Summit dari *Game Developer Conference* (GDC), Anda akan banyak membaca dan mendengar tentang FSM, BT, dan sistem berbasis aturan. Ketiga teknik ini, dan berbagai kombinasinya, menggerakkan AI di balik sebagian besar produksi saat ini dalam video game. Semuanya termasuk dalam paradigma AI yang sama yang disebut AI reaktif, yang telah kami singgung sebelumnya dan bahas lebih lanjut di bagian ini.

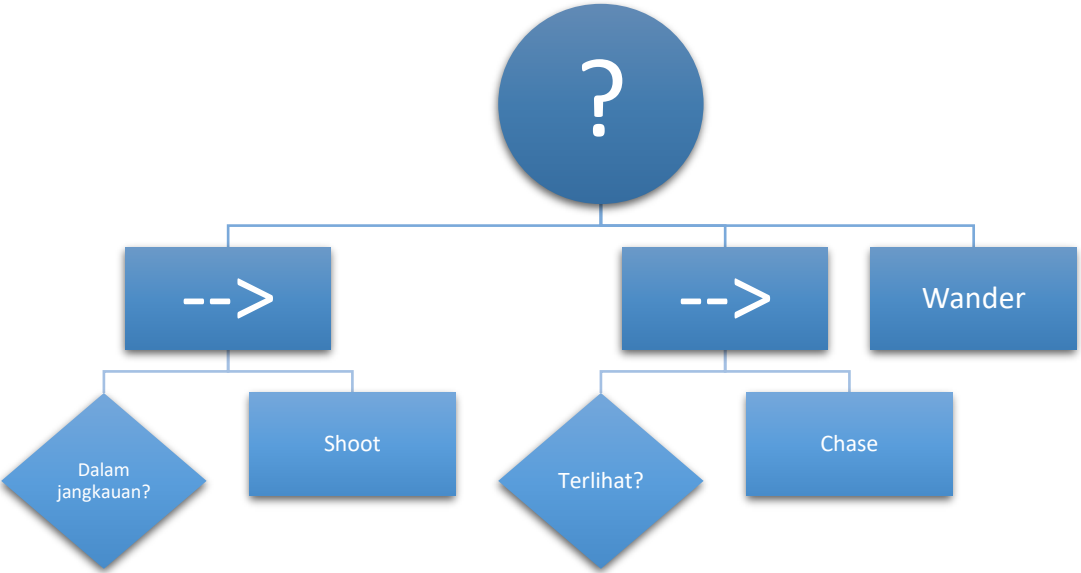
Anda dapat menemukan survei alat AI reaktif di akhir bab ini. Gambar 10.2 dan 10.3 menunjukkan bagaimana perilaku musuh sederhana yang disebut wander-chase-shoot diimplementasikan menggunakan dua teknik yang paling populer: FSM dan BT. Inti dari teknik ini dirangkum dalam kalimat terakhir: perilaku diimplementasikan menggunakan alat ini. Artinya, terserah pengembang untuk merancang setiap aspek perilaku yang akan ditunjukkan AI. Alat-alat ini ada untuk membantu mengatur pengetahuan dengan cara yang lebih atau kurang grafis dan membuat aturan keputusan lebih mudah dipahami manusia daripada kode biasa. Namun, alat-alat ini tidak memiliki kemampuan pemecahan masalah yang nyata dan tidak membantu dalam keputusan desain apa pun. Dalam hal ini, alat-alat ini hanyalah bahasa pemrograman visual. Kelebihannya adalah menyediakan cara untuk mengimplementasikan tindakan kompleks dengan cara yang mudah dipahami.



Gambar 10.2. Mesin berstatus terbatas yang mengimplementasikan perilaku mengembara-mengejar-menembak

Sistem yang ditunjukkan pada Gambar 10.2 dimulai dalam status Mengembara, di mana ia menjelajahi lingkungan secara acak untuk mencari musuhnya; yaitu, pemain. Saat

pemain terlihat, AI beralih ke status Mengejar di mana ia mencoba untuk sedekat mungkin dengan pemain. Saat pemain berada dalam jangkauan, AI bergerak ke status Menembak di mana ia menyerang pemain. Jika salah satu kondisi—pemain dalam pandangan dan pemain dalam jangkauan menjadi salah, AI kembali ke status yang sesuai (Mengembara dan Mengejar, masing-masing).



Gambar 10.3. Pohon perilaku yang mengimplementasikan perilaku mengembara-mengejar-menembak

Sistem yang ditunjukkan pada 10.3 dimulai di simpul akar pohon (simpul putaran atas). Simpul ini adalah simpul Pilih. Dengan demikian, ia mengeksekusi semua anaknya secara bergantian, dari kiri ke kanan, hingga salah satu dari mereka mengembalikan keberhasilan. Jika tidak ada anak yang berhasil, ia mengembalikan kegagalan; jika tidak, ia mengembalikan sukses (singkatnya, ia mengimplementasikan logika OR dan aturan prioritas). Node persegi panjang yang berisi panah gelap adalah node Urutan. Mereka juga mencoba mengeksekusi semua anak mereka dari kiri ke kanan, tetapi mereka berhasil hanya jika mereka dapat mencapai akhir urutan tanpa menemui kegagalan anak (jadi, mereka mengimplementasikan logika AND dan aturan urutan). Node berbentuk berlian adalah node Kondisi. Mereka memeriksa kondisi yang diberikan dan mengembalikan sukses hanya jika kondisinya ternyata benar. Akhirnya, node persegi panjang abu-abu adalah tindakan primitif dari perilaku mengembara-mengejar-menembak.

Alasan industri gim sangat mementingkan paradigma ini adalah karena paradigma ini menyediakan pengendalian AI secara total. Banyak orang termasuk kita melihat desain gim sebagai suatu bentuk seni. Seperti seniman lainnya, pengembang gim ingin memiliki kendali atas kreasi mereka. Karena AI reaktif menyediakan kendali total, AI ini sebagian besar diadopsi sebagai solusi default. Namun, kendali total memiliki beberapa kelemahan yang akan kita bahas di subbagian berikut.

Kemampuan beradaptasi

Poin pertama adalah, karena dirancang dengan tangan, perilaku reaktif sering kali dikhususkan untuk situasi tertentu dan tidak mudah beradaptasi dengan perubahan penting. Perluasan gim video yang menghadirkan elemen permainan baru yang dapat atau seharusnya memengaruhi perilaku NPC mungkin memerlukan penulisan ulang AI secara mendalam. AI utilitas adalah teknik yang diperkenalkan sebagian untuk mengimbangi hal ini. AI utilitas terdiri dari penerapan beberapa keputusan dasar berdasarkan komputasi numerik, bukan aturan tetap. Pertimbangkan, misalnya, simpul Select dari BT (Gambar 10.3). Ia mengodekan preferensi yang ketat dan selalu dihormati di antara opsi yang tersedia (anak kiri selalu lebih diutamakan daripada anak kanan). AI utilitas 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 dalam adegan, jarak ke musuh terdekat, jumlah amunisi yang tersedia, keberadaan perlindungan yang dapat dijangkau, dan sebagainya.

Intinya adalah bahwa faktor-faktor ini berkelanjutan dan hampir tidak dapat dikontrol dan diamati. Akibatnya, AI akan lebih mudah beradaptasi dan kurang dapat diprediksi. Perhatikan bahwa perhitungan utilitas yaitu, aturan keputusan inti masih diimplementasikan oleh perancang. Kita masih berada dalam domain AI reaktif, tetapi keputusan dapat beradaptasi 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 setiap tindakan yang tersedia, dan terserah kepada pengembang untuk memberikan perkiraan yang akurat. Dalam artikelnya “Simulating behavior trees: A behavior tree/planner hybrid approach,” Daniel Hilburn menjelaskan bagaimana simulasi digunakan sebagai pengganti komputasi yang ditentukan pengguna untuk mengevaluasi alternatif yang tersedia.

10.3 KOMPLEKSITAS DAN UNIVERSALITAS

Otak manusia tidak dapat menangani kompleksitas yang ada. Jumlah pengetahuan yang harus dimasukkan ke dalam AI untuk mencapai perilaku kompleks yang memuaskan sering kali sangat besar. Salah satu aspek tersulit dalam merancang sistem AI reaktif adalah memastikan bahwa semua situasi yang akan dihadapi sistem saat dijalankan tercakup oleh aturan perilaku yang tepat. Pertimbangkan, misalnya, menulis AI untuk mengemudikan mobil melalui persimpangan empat arah (masalah yang dipinjam dari robotika otonom). Merancang aturan dasar itu mudah: tunggu sampai Anda memiliki hak jalan dan persimpangan sudah aman, lalu lanjutkan untuk menyeberangi persimpangan.

Sekarang, kita harus membahas kasus-kasus tertentu: jika seorang pengemudi menunjukkan perilaku agresif dan mencoba menipu rambu berhenti, berhati-hatilah dan biarkan dia lewat; jika ada truk pemadam kebakaran yang mendekat dari belakang, dengan kecepatan penuh dan semua sirene menyala, cobalah untuk membebaskan jalannya dengan terus melaju dengan hati-hati melewati persimpangan. Kemudian, kita dapat bertanya pada diri sendiri: apa yang harus saya lakukan jika ada orang yang melanggar rambu berhenti dan

truk pemadam kebakaran yang datang? Menjawab pertanyaan ini mungkin memerlukan pertimbangan faktor-faktor lain; yaitu, mendefinisikan kasus-kasus yang lebih khusus yang memerlukan aturan-aturan khusus. Ini adalah contoh ledakan kombinatorial yang menghancurkan AI reaktif dalam aplikasi dunia nyata.

Kelayakan

Ini adalah batasan terkuat dari pendekatan AI reaktif. Mendesain semua aspek perilaku secara manual memerlukan pengetahuan tentang 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; diperlukan sejumlah pemecahan masalah. Pertimbangkan, misalnya, navigasi; yaitu, tugas menemukan jalur terpendek dari lokasi asal ke tujuan. Hal ini umumnya diselesaikan menggunakan algoritma jalur terpendek seperti A*, yang merupakan contoh sistem perencanaan yang dibahas di bagian berikutnya. Jalur terpendek diselesaikan dengan mengeksplorasi kemungkinan masa depan, memprediksi hasil dari rangkaian tindakan yang berbeda untuk dapat memilih yang terbaik. Penalaran ini bergantung pada keadaan lingkungan saat ini dan tujuan AI, dan melibatkan sejumlah pemecahan masalah yang wajar.

Tidaklah layak untuk memecahkan masalah jalur terpendek dengan cara umum dengan menggunakan serangkaian aturan yang tetap dan telah ditetapkan sebelumnya. Dalam praktiknya, pengembang gim menggunakan perangkat 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 penalaran lain di luar navigasi. Contoh lain diberikan 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 gim video ke XR. Karena berbagai alasan, baik VR maupun AR lebih menantang paradigma ini daripada gim.

Dalam AR, kesulitannya berasal dari lingkungan yang tidak dapat diprediksi. Adegan AR dibangun berdasarkan adegan dunia nyata, dengan menambahkan elemen virtual di dalamnya; oleh karena itu, adegan tersebut tidak dapat dikontrol dan tidak dapat diprediksi. Tidak diragukan lagi bahwa saat aplikasi AR menjadi umum, aplikasi tersebut akan diuji di lingkungan paling orisinal oleh pengguna. Hal ini sangat kontras dengan adegan dalam gim video yang sepenuhnya dirancang dengan tangan dan dengan demikian sepenuhnya dapat dikontrol dan diprediksi (jika kita mengecualikan kasus terbatas level gim yang dibuat secara prosedural).

Jelas, lebih mudah untuk merancang dengan tangan AI yang disesuaikan dengan adegan yang sepenuhnya kita kendalikan daripada sebagian. Dengan kata lain, AR lebih menantang batasan AI reaktif dalam hal kemampuan beradaptasi, kompleksitas, dan universalitas daripada gim video. Dalam kasus VR, argumennya berbeda. Seperti adegan gim video, adegan VR sepenuhnya dikendalikan oleh perancang. Masalahnya di sini ada pada ekspektasi pengguna.

Karena pengalaman sensorik dan imersi ditingkatkan secara signifikan saat kita beralih dari gim 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 gim video. Namun, pengembang gim tampaknya takut untuk mengulangi kesalahan ini di dunia VR. Hal ini tercermin dari rendahnya jumlah NPC aktual yang ditemui dalam permainan VR (selain NPC musuh yang hanya merupakan target bergerak untuk ditembak pemain).

Sebelum memeriksa alternatif yang ada untuk AI reaktif, kami mencatat bahwa, sejauh ini, bagian ini hanya membahas aspek perilaku tingkat tertinggi dan skala terbesar (lihat kembali bagian "Perilaku"). Kami telah menekankan bahwa tugas navigasi umumnya diselesaikan menggunakan algoritme jalur terpendek khusus yang tidak termasuk dalam paradigma AI reaktif. Bagaimana dengan level terendah dalam hierarki Gambar 10-1; yaitu, pembuatan animasi? Faktanya, situasinya sangat mirip dengan perilaku skala besar. Praktik saat ini adalah menggunakan klip animasi berdurasi beberapa detik dan mewakili satu gaya berjalan. Klip-klip disusun ke dalam pengontrol animasi besar, yang merupakan FSM dengan satu klip yang terpasang pada setiap status. Transisi antar klip menggambarkan perubahan gaya berjalan dan dipicu oleh masukan pemain dalam kasus karakter yang dapat dimainkan (PC) atau oleh modul perilaku tingkat tinggi dalam kasus NPC. Transisi dari satu animasi ke animasi lain dikelola melalui pencampuran animasi, yang melibatkan beberapa parameter numerik. Praktik yang paling umum adalah mengatur dan menyetel parameter-parameter ini secara manual, yang merupakan tugas yang berat.

Hal ini memang termasuk dalam paradigma AI reaktif: sistem animasi tidak mengantisipasi lebih dari satu bingkai ke depan di masa mendatang, dan semua aturan keputusan dirancang oleh pengembang manusia. Baru-baru ini, pendekatan berbasis tujuan telah diusulkan di mana AI berhak memutuskan pose karakter berikutnya berdasarkan pose saat ini dan tujuannya. Pendekatan berbasis tujuan serupa dari pembuatan perilaku umum dibahas di bagian berikutnya.

10.4 LEBIH BANYAK KECERDASAN DALAM SISTEM

Pertukaran antara kemampuan pengendalian dan otonomi menjadi pusat pembahasan seputar AI permainan: AI yang menampilkan beberapa bentuk kekuatan keputusan (jelas) kurang dapat dikendalikan daripada AI yang sepenuhnya dikodekan secara manual. Kita melihat di bagian sebelumnya bahwa praktik yang paling umum adalah mengorbankan otonomi demi kemampuan pengendalian, dengan menggunakan alat AI reaktif. Kita juga melihat bahwa hal itu tidak dapat diterapkan pada semua masalah yang dapat dihadapi agen buatan. Beberapa keputusan memerlukan prediksi dan antisipasi efek dari rangkaian tindakan yang berbeda. Ini adalah kasus navigasi yang tidak dapat dilakukan tanpa algoritme jalur terpendek; dengan kata lain, pemecah masalah. Contoh lain dari masalah "sulit" meliputi yang berikut ini:

Manajemen sumber daya

Terutama jika dicampur dengan masalah navigasi. Misalnya, agen harus menavigasi ke tujuan, tetapi gerakan menghabiskan sumber daya yang jumlahnya terbatas. Sumber daya dapat diisi ulang di berbagai lokasi lingkungan. Jadi, agen perlu mengintegrasikan beberapa

pemberhentian untuk mengisi ulang sumber daya dalam perjalanannya ke tujuan. Menemukan jalur terpendek ke tujuan termasuk membuat jalan memutar terpendek untuk mengisi ulang sumber daya. Penalaran ini harus mengintegrasikan struktur lingkungan, termasuk lokasi stasiun pengisian ulang sumber daya dan tujuan. Sangat sulit jika memungkinkan, sama sekali untuk merancang aturan keputusan umum secara manual untuk masalah ini.

Eksplorasi cerdas

Mengintai karakter musuh memerlukan penalaran tentang status pengetahuan apa yang diketahui dan tidak diketahui pada saat ini dan merencanakan bagaimana gerakan eksplorasi akan mengubah status ini. Misalnya, AI akan memutuskan untuk membuat jalan memutar melalui bukit di dekatnya karena pemandangan dari puncak 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. Hal ini tidak dapat diselesaikan dengan baik oleh aturan keputusan tetap.

Perencanaan taktis

Contohnya termasuk mengelola regu NPC sehingga mereka mencoba menjebak pemain, menghalangi semua jalur keluar mereka dari adegan permainan. Hal ini lagi-lagi terlalu bergantung pada konfigurasi masalah untuk menggunakan aturan yang telah ditetapkan sebelumnya.

Tugas-tugas ini (dan yang lainnya) memerlukan beberapa bentuk kemampuan pencarian dan pemecahan masalah dalam AI. Tugas-tugas ini ditangani oleh alat perencanaan otomatis, yang menerapkan paradigma AI deliberatif yang dibahas dalam bagian ini.

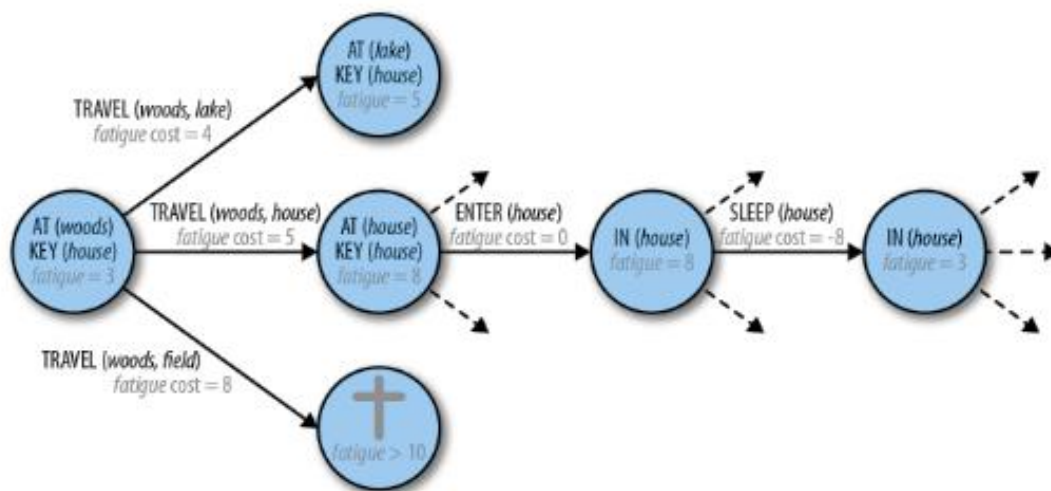
Berikut adalah dua poin utama dari paradigma AI deliberatif:

- Kami berfokus pada pembuatan urutan tindakan daripada keputusan sekali jalan. Hal ini sesuai dengan jenis masalah keputusan yang disebutkan sebelumnya. Masalah-masalah ini dicirikan oleh fakta bahwa suatu tindakan hanya menarik sehubungan dengan tindakan-tindakan yang akan mengikutinya. Untuk memahami dengan jelas perbedaannya dengan AI reaktif, pertimbangkan masalah navigasi ke suatu tujuan. Jika kita selalu bergerak lurus ke arah tujuan, kita dapat terjebak pada 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 tujuan dan melewati rintangan. Langkah ini menarik hanya sebagai langkah pertama dari rangkaian yang dapat membawa kita ke tujuan. Jika diambil secara terpisah, langkah ini tidak mencapai tujuan apa pun. Output sebenarnya dari algoritme adalah jalur lengkap menuju tujuan; yaitu, rencana, yang keputusan yang dieksekusi hanyalah langkah pertama.
- Keputusan dibuat secara otomatis, dengan memecahkan masalah yang terdefinisi dengan baik, bukan dikodekan secara kaku oleh pengembang. Di dunia lain, AI memiliki kekuatan pengambilan keputusan yang nyata, yang didukung oleh kemampuan pemecahan masalah. Dalam hal ini, AI merupakan pelengkap yang sempurna untuk AI

reaktif karena AI membantu memutuskan apa yang harus dilakukan dalam situasi tertentu. AI juga membantu menghindari sebagian kerumitan yang muncul dari kebutuhan universalitas yang dibahas sebelumnya. Perencana otomatis menggunakan model masalah pengambilan keputusan yang mereka hadapi. Untuk menguraikannya, dengan menggunakan contoh navigasi lagi, algoritme jalur terpendek dapat memodelkan masalah navigasi sebagai peta jalan; yaitu, jaringan diskrit (atau grafik) di mana simpul disebut titik jalan, konektor antara simpul mewakili kemungkinan gerakan, 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) model domain mereka.

Seperti yang kami katakan, contoh perencana yang paling umum adalah sistem navigasi yang menghitung jalur terpendek antara dua lokasi di dunia permainan. Setiap buku pengantar AI permainan memberikan survei tentang algoritme ini; lihat misalnya [3, 30]. Prinsip dasarnya disebut pencarian. Diberikan lokasi awal, yang direpresentasikan misalnya sebagai titik jalan di peta jalan, mereka memperluas beberapa lintasan masa depan yang mungkin menuju tujuan dan memilih yang terbaik. Trik algoritme memungkinkan Anda melakukan pencarian ini secara efisien dan menghindari mempertimbangkan urutan tindakan yang dapat terbukti tidak optimal. Namun 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 status perencanaan umum. Keadaan perencanaan merangkum semua informasi yang relevan dengan masalah keputusan pada waktu tertentu. Dalam masalah navigasi, keadaan 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 jalan digeneralisasikan oleh tindakan perencanaan.



- Keputusan dibuat secara otomatis, dengan memecahkan masalah yang terdefinisi dengan baik, bukan dikodekan secara kaku oleh pengembang. Di dunia lain, AI memiliki kekuatan pengambilan keputusan yang nyata, yang didukung oleh kemampuan pemecahan masalah. Dalam hal ini, AI merupakan pelengkap yang sempurna untuk AI reaktif karena AI membantu memutuskan apa yang harus dilakukan dalam situasi tertentu. AI juga membantu menghindari sebagian kerumitan yang muncul dari kebutuhan universalitas yang dibahas sebelumnya. Perencana otomatis menggunakan model masalah pengambilan keputusan yang mereka hadapi. Untuk menguraikannya, dengan menggunakan contoh navigasi lagi, algoritme jalur terpendek dapat memodelkan masalah navigasi sebagai peta jalan; yaitu, jaringan diskrit (atau grafik) di mana simpul disebut titik jalan, konektor antara simpul mewakili kemungkinan gerakan, 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) model domain mereka.

Seperti yang kami katakan, contoh perencana yang paling umum adalah sistem navigasi yang menghitung jalur terpendek antara dua lokasi di dunia permainan. Setiap buku pengantar AI permainan memberikan survei tentang algoritme ini. Prinsip dasarnya disebut pencarian. Diberikan lokasi awal, yang direpresentasikan misalnya sebagai titik jalan di peta jalan, mereka memperluas beberapa lintasan masa depan yang mungkin menuju tujuan dan memilih yang terbaik. Trik algoritme memungkinkan Anda melakukan pencarian ini secara efisien dan menghindari mempertimbangkan urutan tindakan yang dapat terbukti tidak optimal. Namun 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 status perencanaan umum. Keadaan perencanaan merangkum semua informasi yang relevan dengan masalah

keputusan pada waktu tertentu. Dalam masalah navigasi, keadaan 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 jalan digeneralisasikan oleh tindakan perencana.

Perencanaan otomatis adalah pendekatan berbasis tujuan yang sistematis untuk menghasilkan perilaku. Lihat bagian Referensi untuk survei bidang ini. Ada banyak variasi pada skema dasar yang baru saja kami jelaskan, termasuk yang berikut:

Perencanaan temporal

Ini memungkinkan penalaran yang baik tentang durasi tindakan, dan pelaksanaan simultan dari beberapa tindakan yang tidak saling bertentangan.

Perencanaan dalam ketidakpastian

Ini memodelkan dan menalar tentang ketidakpastian dalam efek tindakan. Alih-alih satu set efek, suatu tindakan memiliki beberapa set efek dengan probabilitas berbeda yang menyertainya. Misalnya, menyerang musuh mungkin memiliki beberapa 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 menganggap bahwa setiap serangan selalu berhasil atau selalu gagal. Dalam kedua kasus, hal itu akan mengarah pada perilaku yang buruk (terlalu percaya diri dalam kasus pertama, dan terlalu konservatif dalam kasus kedua). Perilaku yang tepat diperoleh hanya dengan mempertimbangkan berbagai kemungkinan hasil tindakan, dan menyeimbangkan risiko dengan manfaat dari berbagai pilihan.

Perencanaan urutan parsial

Ini menghasilkan rencana di mana tindakan tidak sepenuhnya diurutkan dalam urutan. Misalnya, cabang teknik menghasilkan rencana dalam bentuk urutan serangkaian tindakan (berlawanan dengan urutan tindakan). Rencana ini dijalankan dengan cara berikut: pertama semua tindakan dalam rangkaian pertama dijalankan dalam urutan apa pun yang sesuai, kemudian semua tindakan dalam rangkaian kedua dijalankan dalam urutan apa pun, dan seterusnya. Perencana menjamin bahwa tindakan dalam rangkaian yang sama dapat dijalankan dalam urutan apa pun tanpa memengaruhi hasilnya.

Perencanaan hierarkis

Ini mencoba memecahkan beberapa lapisan pengambilan keputusan dalam satu alat (lihat Gambar 10.1). Alat yang paling sering digunakan dalam pendekatan ini disebut jaringan tugas hierarkis (HTN). Alat ini memungkinkan dekomposisi perilaku hierarkis seperti yang diamati dalam BT, sambil mengotomatiskan semua pengambilan keputusan inti menggunakan perencanaan.

Perhatikan bahwa teknik di balik ekstensi ini dapat berbeda secara signifikan dari

pencarian ruang-keadaan dasar yang diuraikan di sini. Game pertama yang menggunakan perencanaan otomatis untuk tingkat tertinggi hierarki perilaku adalah F.E.A.R. Perencana tersebut disebut Perencanaan Tindakan Berorientasi Sasaran (GOAP).³⁰ Alat ini digunakan untuk AI musuh, dan memberikan kesan yang kuat pada komunitas gamer. Meskipun demikian, seluruh pendekatan AI deliberatif memiliki penetrasi yang sangat dangkal dalam industri game. Tidak mengherankan, pendekatan ini sebagian besar telah diterapkan pada game taktis dan strategis yang berisi masalah keputusan dan pengoptimalan yang sulit yang solusinya memerlukan beberapa bentuk pertimbangan. HTN mendapat perhatian dari masyarakat, terutama karena kemiripannya dengan BT yang sangat populer. Meskipun ada upaya-upaya ini, paradigma AI deliberatif masih mewakili minoritas kecil dari semua AI permainan komersial. Ada beberapa alasan untuk menjelaskan hal ini:

Kemampuan Kontrol

Kita sudah membahas poin ini secara luas. AI Reaktif menyediakan kontrol total, yang sangat bagus untuk membuat game dan XR (meskipun sulit). Perencana kurang mudah dikendalikan, tetapi dapat memecahkan masalah sulit yang tidak dapat diselesaikan dengan aturan tertulis. Daripada harus memilih antara satu pendekatan atau yang lain, kami sangat menyarankan untuk menggunakan campuran keduanya. Kami mendorong pengembang untuk merancang arsitektur modular yang bersih dan menggunakan teknik serta 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 yang setiap tindakannya diimplementasikan oleh BT tertentu adalah arsitektur yang dapat sangat masuk akal dalam banyak situasi. Demikian pula, BT dapat berisi simpul perencanaan yang membahas submasalah yang terbatas tetapi sulit diputuskan.

Kompleksitas waktu proses

Salah satu masalah utama dengan teknik pemecahan masalah seperti pencarian ruang-keadaan yang dijelaskan sebelumnya adalah waktu pelaksanaannya. Kompleksitas meningkat seiring dengan ukuran masalah (ukuran peta dalam masalah navigasi, jumlah NPC yang harus dikendalikan dalam penembak taktis, dll.). Bergantung 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 menembus batas ini. Dengan kata lain, beberapa masalah terlalu besar untuk dipecahkan pada frekuensi yang diinginkan. Kami mencatat bahwa ketika mereka terlalu lambat untuk digunakan dalam waktu nyata, Anda dapat menggunakan perencana secara luring pada saat mengembangkan permainan atau aplikasi XR untuk menghasilkan rencana terlebih dahulu yang nantinya dapat diubah menjadi aturan keputusan reaktif untuk digunakan pada waktu proses. Kami membahas di bagian berikutnya bagaimana Anda dapat menggunakan pembelajaran mesin untuk lebih meningkatkan kekuatan pemecahan masalah perencana saat digunakan secara luring.

Kesulitan untuk diimplementasikan

Perencanaan otomatis menjanjikan perilaku AI yang hebat, tetapi ada harganya. Algoritme perencanaan memiliki kompleksitas tertentu dan memerlukan sejumlah sumber daya untuk dikembangkan, sehingga tidak dapat dijangkau oleh tim kecil tanpa programmer AI yang ahli. Di dunia akademis, bahasa deskripsi domain perencanaan telah dikembangkan untuk memungkinkan penggunaan kembali perencana/penyelesai yang sama di domain yang berbeda. Idanya adalah untuk mendefinisikan bahasa yang dapat digunakan untuk mengekspresikan (merencanakan) masalah dengan sifat yang berbeda, lalu membuat penyelesaian yang dapat menangani masalah apa pun yang diekspresikan dalam bahasa tersebut (dengan cara yang sama seperti algoritme jalur terpendek dapat menangani masalah apa pun yang direpresentasikan sebagai peta jalan). Pendekatan ini banyak digunakan oleh komunitas robotika otonom, terutama oleh NASA, yang mengendalikan beberapa perangkat (semi-)otonom dengan sifat dan ukuran yang sangat berbeda menggunakan perencana yang sama. Kami yakin ini adalah kunci untuk menghasilkan alat AI deliberatif yang umum dan dapat digunakan kembali untuk gim video dan XR, dan layak mendapat perhatian lebih dari komunitas ini.

Kesulitan untuk mengadopsi

Beralih dari AI reaktif ke AI deliberatif merupakan 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 langsung mengedit perilaku tetapi harus melalui tugas pemodelan domain, banyak orang akan merasa seperti lapisan kompleksitas tambahan telah ditambahkan di sepanjang jalan. Namun, ada argumen teoretis yang menunjukkan fakta bahwa setidaknya untuk masalah yang kompleks AI deklaratif mungkin sebenarnya lebih mudah digunakan daripada AI reaktif.

Singkatnya, argumen ini berbunyi sebagai berikut. Dunia memiliki struktur, setidaknya bagi mata manusia. Ketika kita diminta untuk menjelaskan suatu masalah, kita biasanya dapat melakukannya dengan cara yang relatif ringkas dan terstruktur. Khususnya, kita dapat menilai beberapa hipotesis independensi antara berbagai variabel. Misalnya, pertimbangkan masalah yang dihadapi oleh teknisi perbaikan mesin fotokopi yang perlu merancang jadwal untuk satu hari. Ia perlu memutuskan pelanggan mana yang akan dikunjunginya dan operasi apa yang akan dilakukannya pada mesin fotokopi milik pelanggannya. Ketika kita menyatakan masalah itu, kita dapat mengasumsikan banyak hubungan independensi antara variabel dan tindakan.

Jelas, setiap operasi perbaikan atau diagnostik yang dilakukan teknisi servis pada mesin fotokopi tidak memengaruhi mesin lain milik pelanggan yang sama atau pelanggan lain. Bahkan, kita dapat dengan mudah membagi masalah tersebut menjadi beberapa submasalah, satu untuk setiap mesin yang perlu diservis. Ada banyak variabel status yang dikaitkan dengan satu mesin (misalnya, status terkini dari berbagai komponen, hasil pengujian yang telah dilakukan, riwayat perbaikan pada mesin ini), tetapi variabel-variabel ini independen dari satu mesin ke mesin lainnya.

Permasalahan ini secara alami dimodelkan sebagai arsitektur dua lapis. Pada level tertinggi, kita memiliki masalah umum dalam memutuskan pelanggan mana yang akan kita

kunjungi dan dalam urutan apa (masalah yang mengandung komponen kuat perencanaan jalur terpendek). Level terendah dibentuk oleh beberapa subproses, satu untuk setiap mesin fotokopi yang perlu diperbaiki. Pada level yang lebih rendah, submasalah lokal memiliki banyak variabel dan tindakan yang dapat dianggap sebagai privat bagi subproses ini: menjalankan tindakan privat hanya memengaruhi variabel privat dari proses ini.

Subproses-subproses tersebut diikat bersama pada tingkat tinggi oleh sekumpulan kecil variabel bersama yang mewakili sumber daya bersama: total waktu yang dapat dihabiskan teknisi untuk mengerjakan mesin fotokopi pada hari itu, sekumpulan suku cadang (terbatas) yang tersedia untuknya, dan lokasi agen (yang dapat dilihat sebagai sumber daya bersama tertentu). Sebagai konsekuensinya, selama kita hanya peduli dengan mendeskripsikan masalah, kita dapat dengan mudah menambahkan, menghapus, atau memodifikasi mesin fotokopi/submasalah. Modifikasi satu subproses tidak memengaruhi proses lainnya dengan cara apa pun. Mesin fotokopi baru dapat disambungkan ke arsitektur umum seperti kepingan Lego, variabel bersama memainkan peran sebagai tiang yang menjaga konstruksi tetap utuh.

Masalah memiliki struktur alami yang membuat deskripsinya padat dan modifikasinya mudah dan bertahap. Sayangnya, struktur ini lenyap saat 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 akan memengaruhi rencana kita saat kita mengerjakan mesin fotokopi lain. Namun, ini tidak benar.

Untuk melihatnya, bayangkan bahwa kita saat ini sedang menyelesaikan masalah pelanggan tertentu dan kita berencana untuk mengembalikan mesin pelanggan tersebut sepenuhnya ke kondisi semula sebelum beralih ke pelanggan berikutnya. Ini adalah kebijakan lokal kita saat ini untuk pelanggan ini: selesaikan semua pekerjaan di sini sebelum pergi. Sekarang, kita tiba-tiba menambahkan mesin fotokopi baru yang bersaing dengan tugas kita saat ini untuk beberapa suku cadang tertentu. Jika kita menambahkan bahwa mesin fotokopi baru ini milik pelanggan prioritas utama kita dan bahwa kita berkomitmen untuk menyelesaikan masalah pelanggan tersebut terlebih dahulu, kita mungkin memutuskan untuk menghentikan pekerjaan kita saat ini, menyimpan suku cadang untuk pelanggan yang paling penting, dan segera beralih ke pelanggan prioritas utama. 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 mudah dipahami yang membantu menjelaskannya (dan sebagian besar masalah nyata memang demikian), perilaku optimal untuk memecahkan masalah ini tidak boleh memiliki struktur sama sekali. Oleh karena itu, pada prinsipnya, ada tingkat kerumitan tertentu yang lebih baik jika kita bekerja di ruang masalah daripada bekerja di ruang perilaku. Seperti yang kita lihat, beberapa batasan ini memiliki kemungkinan solusi yang layak diteliti dalam domain gim video dan XR.

Seperti yang kita katakan, setiap perencana memiliki batasan dalam hal kerumitan masalah yang dapat ditanganinya sesuai anggaran atau bahkan dalam waktu yang wajar.

Dalam penelitian akademis, kekuatan AI yang disengaja telah didorong ke 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.

10.5 PEMBELAJARAN MESIN

Sejauh ini, kita telah membahas perilaku dalam konteks algoritme dan metode yang menghasilkan kebijakan tetap untuk tindakan. Pendekatan ini bergantung pada penulisan perilaku oleh manusia, baik secara langsung dengan memberikan aturan eksplisit untuk perilaku (di bagian sebelumnya), atau secara tidak langsung dengan memberikan model dinamika simulasi dan mekanisme untuk perencanaan menggunakan model ini. Berbeda dengan metode tersebut, kini kita beralih ke pendekatan yang didasarkan pada pembelajaran perilaku dari data. Pembelajaran dari data dapat menarik karena sangat mengurangi jumlah pengetahuan yang dibutuhkan untuk membangun AI. Kita mungkin dapat membangun AI untuk masalah yang tidak kita pahami dengan baik atau yang tidak kita kuasai. AI melakukannya dengan mengorbankan waktu komputasi dan kumpulan data yang besar. AI juga menarik dalam situasi di mana kita ingin perilaku yang dipelajari digeneralisasi ke keadaan yang tidak terlihat.

Pendekatan ini termasuk dalam cakupan luas pembelajaran mesin, yang hadir dalam tiga bentuk berbeda:

Pembelajaran terbimbing

Ini berkaitan dengan pembelajaran pemetaan dari himpunan X ke himpunan Y melalui 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 masukan dan mengeluarkan label seperti "kucing," "anjing," "burung," dan seterusnya. Selanjutnya, kami mendefinisikan fungsi kerugian yang menetapkan nilai numerik untuk perbedaan antara hasil yang diinginkan dan keluaran yang diamati dari model. Singkatnya, kami mengukur seberapa baik kinerja kami dengan nilai numerik. Dengan menggunakan kumpulan data besar contoh 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 mempelajari struktur dalam data tanpa pemetaan yang jelas dari satu set ke set lainnya sebagai tujuan. Ini berguna, misalnya, untuk memahami atau mengompresi data.

Pembelajaran penguatan

Ini adalah 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 penghargaan tidak secara langsung menggambarkan seberapa baik kinerja model, tetapi diberikan saat agen memasuki kondisi tertentu, dan sesuai dengan keinginan kondisi tersebut. Dengan kata lain, tujuannya bukanlah untuk mempelajari fungsi penghargaan (yang akan menjadi kasus untuk pembelajaran

terbimbing), tetapi untuk menemukan perilaku yang mengarah pada pencapaian kondisi yang paling diinginkan paling sering. Fungsi ini sangat terkait dengan otomatis karena kita berkepentingan dengan produksi urutan tindakan alih-alih keputusan sekali jalan. Memang, model yang mendasari sebagian besar algoritme pembelajaran penguatan adalah model perencanaan yang parameternya awalnya tidak diketahui dan harus dipelajari dengan berinteraksi dengan simulasi.

Relevansi pembelajaran penguatan untuk masalah yang dibahas dalam bab ini mudah dipahami. Tidak mengherankan, hubungan antara disiplin ilmu ini dan permainan video telah berkembang menjadi sangat kuat. Komunitas peneliti telah mengakui relevansi permainan video sebagai tempat uji coba untuk algoritme pembelajaran penguatan. Permainan video arkade sederhana kini menjadi tempat uji coba favorit masyarakat untuk mengevaluasi kinerja algoritme pembelajaran penguatan baru.

Ada dua komponen utama untuk penerapan pembelajaran mesin: data dan algoritme. Kita dapat mempertimbangkan keduanya dalam konteks spesifik perilaku yang dibuat dalam lingkungan simulasi (video game dan XR). Pembelajaran penguatan mendapatkan datanya dari interaksi dengan simulasi itu sendiri. Jenis aplikasi lain belajar dari demonstrasi manusia; yaitu, data pengguna (ahli). Pendekatan ini disebut pembelajaran imitasi dan dibahas kemudian dalam bab ini.

Pembelajaran Penguatan

Pembelajaran penguatan adalah pendekatan dalam pembelajaran mesin untuk mencapai perilaku yang diinginkan, yang kami sebut kebijakan. Pemetaan yang kami minati dalam pembelajaran mesin adalah antara status s dan tindakan a . Dalam beberapa kasus, pemetaan ini bersifat probabilistik dan berbentuk $p(a | s)$. Dalam banyak keadaan, agen mungkin tidak memiliki akses ke definisi lengkap status simulasi. Dalam kasus seperti itu, kami katakan bahwa agen hanya memiliki akses ke pengamatan, yang terbatas dan berasal dari status sebenarnya. Contoh sederhananya adalah dengan mempertimbangkan simulasi kota besar. Dalam simulasi ini, status terdiri dari seluruh posisi dan lintasan semua mobil di jalan virtual. Kita dapat membayangkan bahwa agen di dalam salah satu mobil virtual ini mungkin memiliki akses ke pandangan orang pertama dari kendaraan itu terhadap mobil lain di depannya. Kumpulan informasi terbatas ini kemudian sesuai dengan pengamatan. Masalah pembelajaran penguatan adalah mempelajari pemetaan ($o \mapsto a$), atau probabilitas $p(a | o)$, yang memaksimalkan fungsi imbalan dari waktu ke waktu.

Dibandingkan dengan pendekatan berbasis rencana yang dibahas di bagian sebelumnya, pembelajaran penguatan dapat terjadi tanpa adanya model maju dari dinamika simulasi. Metode-metode ini disebut bebas model. Meskipun memerlukan interaksi yang jauh lebih banyak dengan lingkungan, metode-metode ini sangat umum karena tidak membuat asumsi atau persyaratan pada hal-hal spesifik dari lingkungan.

Ada dua kategori besar metode yang dirancang untuk memecahkan masalah pembelajaran penguatan. Metode-metode ini adalah metode berbasis nilai dan berbasis kebijakan. Dalam metode berbasis nilai, agen mencoba mempelajari estimasi nilai setiap keadaan $V(s)$, atau nilai setiap pasangan keadaan-tindakan $Q(s, a)$. Nilai ini mewakili jumlah

diskonto imbalan masa depan yang diharapkan, seperti yang ditunjukkan dalam Persamaan 10.1.

Persamaan 10.1.

$$V(s) = E\left[\sum_{t=0}^{\infty} \gamma^t r(t) \mid s(0) = s\right],$$

$$Q(s, a) = E\left[\sum_{t=0}^{\infty} \gamma^t r(t) \mid s(0) = s \wedge a(0) = a\right],$$

di mana $s(t)$ 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 diskonto. Jumlah imbalan yang didiskontokan biasanya digunakan untuk membatasi agen pada kebijakan selama periode waktu yang terbatas. Hal ini juga memudahkan untuk memungkinkan adanya trade-off antara keuntungan jangka pendek (jangka waktu diskonto yang lebih kecil) dan keuntungan jangka panjang (jangka waktu diskonto yang lebih besar). Dalam pendekatan ini, fungsi nilai dipelajari melalui interaksi dengan lingkungan itu sendiri. Setelah estimasi nilai yang baik dipelajari, agen dapat menggunakan argmax atas nilai-Q dalam keadaan tertentu sebagai kebijakan, seperti yang ditunjukkan dalam Persamaan 10.2.

Persamaan 10.2.

$$a^*(s) = \arg \max [Q(s, a)]$$

Ini adalah tindakan optimal dalam status 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 status atau pasangan status-tindakan dapat disebutkan secara eksplisit.

Selain metode berbasis nilai, ada kelas metode berbasis kebijakan. Di sini, alih-alih mempelajari serangkaian estimasi nilai, kita langsung mempelajari kebijakan untuk bertindak. Kebijakan ini disebut sebagai $\pi(a \mid s)$ dan menyediakan serangkaian probabilitas atas tindakan a yang dikondisikan pada status s . Kebijakan ini dapat ditingkatkan menggunakan algoritme gradien kebijakan. Intuisi di balik pendekatan ini adalah menggunakan imbalan diskon yang diamati yang diperoleh oleh suatu kebijakan selama evaluasi sebagai sarana untuk meningkatkan kebijakan secara langsung. Untuk kasus-kasus di mana hasilnya lebih baik dari yang diharapkan, kita meningkatkan probabilitas tindakan yang terkait dengan hasil tersebut. Untuk kasus-kasus di mana hasilnya lebih buruk dari yang diharapkan, kita menurunkan probabilitasnya. Algoritma gradien kebijakan dikembangkan untuk digunakan dalam kasus perkiraan fungsi linear.

Pembelajaran Penguatan Mendalam

Metode yang baru saja kita bahas bekerja dengan baik untuk ruang keadaan kecil, yang probabilitas tindakan atau estimasi nilai dapat dihitung untuk semua keadaan dan direpresentasikan dalam memori sebagai larik dan matriks sederhana dari nilai floating-point. Namun, dalam sebagian besar simulasi yang menarik, ini bukanlah asumsi yang dapat dibuat.

Jika kita kembali ke contoh simulasi kota sebelumnya, kemungkinan kombinasi kendaraan dan pejalan kaki jauh melebihi apa yang mungkin dihitung. Lebih jauh, jika kita menggunakan piksel mentah yang tersedia untuk agen dari dalam mobil virtual, bahkan ruang observasi ini tidak dapat diatasi. Di sini, kita memerlukan metode yang memungkinkan perkiraan fungsi kompleks untuk merepresentasikan fungsi nilai $V(s)$ atau $Q(s, a)$, atau probabilitas tindakan $\pi(a|s)$.

Dalam banyak kasus, aproksimator fungsi pilihan adalah jaringan saraf dengan beberapa lapisan tersembunyi, yang mengarah ke teknik yang dikenal sebagai pembelajaran penguatan mendalam. "Mendalam" mengacu pada beberapa lapisan inferensi yang dilakukan oleh jaringan saraf ini. Beberapa lapisan ini sering kali diperlukan ketika tidak ada pemetaan linier sederhana antara pengamatan dan tindakan. Dalam kasus gambar mentah sebagai input, ini hampir selalu terjadi, kecuali untuk gambar yang paling sederhana. Pendekatan penerapan aproksimasi fungsi pada pembelajaran penguatan telah sangat sukses dalam beberapa tahun terakhir.

Dimulai pada tahun 2013 dengan DeepMind yang menunjukkan bahwa Deep Q-Network jaringan saraf dalam yang digunakan untuk memperkirakan nilai-Q dari gambar mentah dapat mempelajari kebijakan untuk memainkan gim Atari lebih baik daripada manusia, telah ada keberhasilan setiap tahun yang mendorong keadaan bidang tersebut lebih jauh. Kini, dimungkinkan untuk mempelajari kebijakan menggunakan deep reinforcement learning untuk melakukan segalanya mulai dari pergerakan, hingga memainkan permainan strategi waktu nyata, hingga menyelesaikan lusinan tugas menggunakan satu jaringan. Elemen algoritmik utama untuk memungkinkan keberhasilan dalam deep reinforcement learning ini difokuskan pada perolehan keuntungan dari penggunaan jaringan saraf sebagai aproksimator fungsi sambil mengurangi kerugian dari pendekatan semacam itu. Ini berarti memanfaatkan kemampuan mereka untuk memodelkan fungsi nonlinier yang kompleks tanpa jatuh ke dalam ketidakstabilan bawaan 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 bootstrapping, bukan yang paling mutakhir. Ini adalah pendekatan yang diambil dalam Deep Q-Network, dan telah diadopsi pada sebagian besar pendekatan pembelajaran mendalam berbasis nilai berikutnya. Dalam kasus metode berbasis kebijakan, ini berarti membatasi divergensi kebijakan baru dari yang lama menggunakan berbagai metode, yang sering kali didasarkan pada divergensi KL dalam ruang tindakan kebijakan. Dua pendekatan yang paling populer adalah Trust Region Policy, yang memberlakukan batasan KL yang keras, dan Proximal Policy Optimization, yang memberlakukan batasan lunak.

Pembelajaran Imitasi

Sejauh ini, kita telah membahas pembelajaran perilaku dari awal hanya dengan menggunakan interaksi dengan simulasi/permainan/XR. Namun, dalam kebanyakan kasus, hal ini dapat menjadi tidak efisien dalam hal sampel dan waktu karena pembelajaran berlangsung melalui uji coba. Perilaku yang diinginkan juga harus ditentukan melalui fungsi penghargaan. Sayangnya, fungsi penghargaan ini sering kali sulit ditentukan dengan cara yang sepenuhnya selaras dengan perilaku yang diinginkan. Misalnya, jika perilaku yang diinginkan adalah agen

melakukan jungkir balik, penghargaan apa yang harus diberikan untuk mendorong perilaku tersebut dengan cara uji coba? Dalam banyak kasus, akan jauh lebih intuitif untuk sekadar memberikan serangkaian demonstrasi perilaku yang diinginkan. Demonstrasi ini kemudian dapat digunakan untuk mempelajari model perilaku.

Ada beberapa cara untuk melakukannya. Yang pertama adalah demonstrasi dapat berfungsi sebagai masukan dan keluaran kumpulan data yang akan 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. Pertimbangkan lagi contoh agen virtual yang mengemudi jalan-jalan di kota yang disimulasikan. Mungkin ada beberapa titik tertentu di mana terdapat percabangan jalan. Jika data demonstrasi berisi contoh yang sama tentang belok kiri di percabangan dan belok kanan, model yang belajar dengan kloning perilaku kemungkinan akan belajar untuk melewati bagian tengah! Kloning perilaku juga mengalami kesalahan majemuk dari waktu ke waktu karena perilaku agen menyebabkannya menjauh dari ruang keadaan demonstrasi yang diberikan selama proses pembelajaran.

Ada sejumlah pendekatan yang mencoba mengatasi kesulitan ini. Pendekatan tersebut sebagian besar berada di bawah domain pembelajaran penguatan terbalik. Dalam pendekatan ini, algoritme mencoba mengungkap fungsi penghargaan yang diikuti oleh demonstran dan menggunakan fungsi penghargaan tersebut untuk memandu model yang dipelajari. Pendekatan seperti ini memungkinkan yang terbaik dari kedua dunia karena agen belajar melalui fungsi penghargaan padat yang mencakup seluruh ruang keadaan serta fungsi penghargaan yang ditentukan dengan tepat, yang akan mendorong perilaku yang diinginkan. Salah satu pendekatan kontemporer khususnya adalah pembelajaran imitasi adversarial generatif, yang menggunakan diskriminator yang dipelajari untuk memberikan sinyal penghargaan kepada agen pembelajaran.

11.6 MENGGABUNGKAN PERENCANAAN OTOMATIS DAN PEMBELAJARAN MESIN

Bagian sebelumnya difokuskan pada metode yang menggunakan model dunia/masalah yang dihadapi oleh AI dan memanfaatkan model ini untuk perencanaan perilaku. Sebaliknya, kami sekarang berfokus pada pembelajaran penguatan 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 muncul dari penggabungan keduanya secara cerdas. Contoh terbaru yang paling terkenal adalah keberhasilan Deep Mind dalam bermain Go menggunakan sistem AlphaGo-nya.

AlphaGo didasarkan pada model permainan Go yang dapat digunakan untuk memprediksi hasil dari berbagai urutan keputusan mengikuti mekanisme pencarian ruang-keadaan yang diuraikan sebelumnya. Namun, AlphaGo melengkapi sistem perencanaan ini menggunakan jaringan saraf dalam yang dilatih untuk bertindak sebagai penaksir nilai dan kebijakan. Penaksir nilai mengaitkan nilai dengan status. Penaksir ini digunakan untuk memangkas kedalaman pohon pencarian perencanaan, membatasi seberapa jauh proses

pencarian harus dilakukan di masa mendatang. Jaringan kebijakan mengaitkan probabilitas dengan pasangan status-tindakan: jaringan ini memperkirakan probabilitas tindakan yang optimal dalam status tertentu. Jaringan ini digunakan untuk memangkas lebar pohon pencarian, memungkinkan proses pencarian untuk fokus hanya pada simpul yang dianggap lebih mungkin berdasarkan kebijakan optimal.

Algoritme ini mematuhi protokol berikut: hasil dari proses perencanaan digunakan sebagai data pelatihan untuk jaringan saraf, menyediakan pemetaan nilai dan 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 ditemukan. Kebijakan yang ditingkatkan ini diumpungkan kembali ke pembelajaran mesin untuk meningkatkan keakuratan nilai yang dipelajari dan distribusi tindakan. Proses ini diulang berkali-kali, dengan sistem perencanaan dan pembelajaran saling memberi data yang semakin akurat. Dengan cara ini, kedua sistem dapat secara berulang meningkatkan dan memperbaiki satu sama lain dalam prosesnya. Pendekatan ini memungkinkan AlphaGo mengalahkan juara dunia dalam permainan tersebut.

Selain memainkan permainan papan dengan kapasitas super-manusia, menggabungkan kedua metode ini dapat memungkinkan pengembang untuk berkompromi secara real time antara keakuratan (yang disediakan oleh perencanaan tradisional) dan kecepatan (yang disediakan oleh aproksimator fungsi jaringan saraf) dalam pengambilan keputusan. Peningkatan kecepatan selama evaluasi diperoleh dengan mengorbankan waktu pelatihan yang lebih lama sebelumnya. Namun, dalam banyak kasus, ini merupakan pilihan yang dapat diterima, dan mirip dengan pilihan yang dibuat saat mempertimbangkan waktu pengembangan prarilis pada perilaku apa pun dalam simulasi. Menggabungkan metode ini juga penting untuk situasi di mana sejumlah besar keputusan harus dibuat dalam simulasi, beberapa dengan fidelitas lebih besar daripada yang lain.

Aplikasi

Pembelajaran penguatan dan pembelajaran imitasi membawa banyak janji AI hebat untuk permainan dan XR. Pembelajaran penguatan membuka jalan untuk menciptakan agen yang dapat memecahkan masalah yang tidak kita pahami sepenuhnya. Ini sangat kontras dengan pendekatan sebelumnya yang mengharuskan kemampuan untuk menggambarkan masalah dengan sempurna (AI delibatif 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 terkini dalam pemecahan masalah dan pengambilan keputusan.

Hebatnya, komunitas penelitian pembelajaran penguatan telah mengembangkan hubungan yang kuat dengan budaya video game dengan mengadopsi game sebagai tolok ukur favorit mereka. Hal ini kontras dengan fakta praktis bahwa hanya sebagian kecil game komersial yang menggunakan konsep pembelajaran mesin saat ini. Namun, tidak diragukan lagi bahwa ketertarikan komunitas penelitian pembelajaran penguatan terhadap game akan segera kembali ke industri ini. Memang, pembelajaran mesin membawa solusi untuk masalah

praktis yang dihadapi saat merancang AI game. Jadi, mari kita periksa apa yang saat ini dapat dilakukan, dan apa yang menurut kami kemungkinan besar akan terjadi dalam waktu dekat.

Kembali ke pembahasan di bagian "Perilaku" dalam pengantar Gambar 10.1, kami tekankan bahwa dampak terkuat dari pembelajaran mesin ada di lapisan persepsi sistem otonom: memahami data numerik kompleks yang berasal dari sensor (yang penambangannya dapat dilihat sebagai contoh). Keberhasilan pertama pembelajaran mendalam ada di domain seperti visi komputer, pengenalan gerakan, dan NLP. Ketika kita sampai pada pengambilan keputusan dan pembuatan perilaku, pembelajaran mesin telah terbukti sangat berharga untuk menyelesaikan tugas di level terendah arsitektur pada Gambar 10.1.

Pembelajaran penguatan mendalam unggul dalam menyelesaikan gim arkade Atari, yang lebih didasarkan pada refleks dan koordinasi yang baik daripada pemodelan dan pemecahan masalah yang sulit. Balapan, pertarungan, dan simulasi olahraga adalah domain aplikasi pembelajaran penguatan yang hebat. Ini tidak terlalu mengejutkan. Sebelumnya, kami tekankan bahwa tugas motorik sensorik tingkat rendah sebagian besar adalah masalah pengoptimalan berkelanjutan multidimensi (sebagai lawan dari pengoptimalan kombinatorial diskrit). Pada saat yang sama, masalah persepsi juga memiliki sifat multidimensi yang berkelanjutan. Hal ini juga selaras dengan hasil pembelajaran mesin yang mengesankan dalam pembuatan animasi, domain (sangat) berkelanjutan lainnya. Dalam praktiknya, pembelajaran penguatan merupakan kandidat yang hebat untuk mengendalikan agen pada level motorik sensorik terendah.

Apa yang kita pelajari dari studi akademis juga merupakan kesulitan teknik pembelajaran penguatan, dalam kondisi saat ini, untuk mengatasi level tertinggi hierarki Gambar 10.1. Salah satu gim arcade tersulit untuk pembelajaran penguatan mendalam adalah Montezuma's Revenge. Gim ini melibatkan pemecahan teka-teki dengan mengurutkan serangkaian tindakan yang panjang seperti mengambil kunci di satu ruangan untuk membuka pintu di ruangan lain. Menjalankan rencana ini dapat berlangsung hingga beberapa menit waktu nyata, yang sangat kontras dengan beberapa detik perencanaan paling lama yang diperlukan untuk memecahkan Space Invaders atau Breakout.

Kesimpulan

AI merupakan bidang yang kaya yang mengusulkan pendekatan yang berbeda terhadap masalah pembuatan perilaku. Daripada melihat keduanya sebagai sesuatu yang saling bersaing, kami lebih suka menekankan sifat saling melengkapi dari pendekatan-pendekatan ini. Kami percaya bahwa desain sistem AI perilaku untuk permainan video atau aplikasi XR harus dimulai dengan penguraian yang jelas dari masalah umum menjadi beberapa subtugas dan memahami kendala yang ada pada 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 penyelesaian masalah yang sulit seperti jalur terpendek, manajemen sumber daya, atau eksplorasi cerdas, AI reaktif adalah kandidat yang hebat. Namun, harus diharapkan bahwa pengembangan AI akan menjadi proses yang membosankan

yang memerlukan banyak percobaan dan kesalahan untuk memperbaiki semua kasus tertentu yang perlu dicakup oleh perilaku tersebut.

- Jika perilaku yang ingin kita hasilkan mencakup pemecahan masalah yang sulit dan kita tahu betul cara menjelaskan masalah ini, atau jika kita tidak memiliki sumber daya untuk memperbaiki AI reaktif kasus per kasus, AI deliberatif harus lebih disukai. Namun, AI deliberatif memerlukan keterampilan teknis untuk mengimplementasikan mesin perencanaan.
- Jika masalah yang harus kita selesaikan terlalu sulit atau kita tidak tahu persis cara 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.

BAB 11

EKOSISTEM TEKNOLOGI KESEHATAN REALITAS VIRTUAL DAN TERTAMBAH

Bab ini membahas berbagai isu yang terkait dengan desain pengalaman realitas virtual (VR) dan realitas tertambah (AR) yang diterapkan dalam konteks perawatan kesehatan, dan menyediakan tutorial untuk menggunakan data gerakan dari pengendali guna mengurangi getaran yang tampak pada pasien Parkinson dalam lingkungan virtual. Saat ini, prospek perawatan kesehatan global ditentukan oleh serangkaian kebijakan, langkah-langkah kesehatan masyarakat, metode pemberian layanan, penelitian klinis berbasis komunitas, terapi, dan inovasi teknologi yang terus berkembang. Tidak ada satu teknologi pun yang dapat mengatasi semua masalah perawatan kesehatan sendirian, dan kini, mulai 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 (misalnya, fMRI) hingga yang sederhana (misalnya, penjadwalan janji temu yang efisien) semuanya memiliki peran dalam pemberian layanan. Teknologi VR dan AR tergolong baru dan belum dianggap sebagai konvensi, apalagi standar perawatan, dalam domain kesehatan mana pun. Ruang-ruang yang bermasalah meliputi pengurangan rasa sakit, perawatan gangguan stres pascatrauma (PTSD) dengan terapi paparan, dan perawatan ambliopia. Ruang-ruang ini telah terbukti cocok untuk VR sebagai teknologi pemberian 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, rincian substansial tentang organisasi formal, badan pengawas, dan proses persetujuan dihilangkan.

Sebaliknya, bab ini membahas lebih lanjut tentang upaya tingkat tinggi yang dapat dilakukan untuk merancang teknologi kesehatan yang lebih baik menggunakan VR dan AR. Penting untuk diketahui bahwa pasien harus memberikan persetujuan sebelum mereka dapat mencoba aplikasi atau eksperimen apa pun, dan ada dewan peninjau yang secara tegas ditujukan untuk tujuan tersebut. Terakhir, bab ini membahas 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

Untuk membuat aplikasi VR dan AR, pengembang harus mempertimbangkan lingkungan fisik tempat pengguna akan berada saat menggunakan teknologi. Misalnya, apakah pengguna adalah pasien di ruang pra-operasi, sendirian sebelum prosedur, atau pengguna adalah anggota keluarga yang berada di kamar pasien.

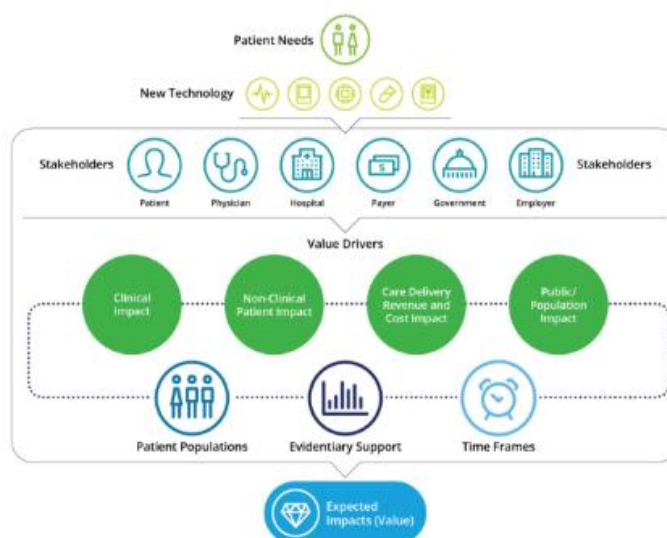
Proses desain harus mencakup waktu untuk memahami lingkungan ini dan apa yang terjadi selama skenario umum. Kita dapat mewawancarai dokter, praktisi perawat (NP), staf, dan personel lain yang terlibat untuk menjawab pertanyaan seperti berikut:

- Jika pasien adalah pengguna, apakah keluarga akan tertarik untuk menghabiskan waktu berkualitas dengan pasien di lingkungan X atau apakah pasien sendirian?
- Sejauh mana lingkungan tersebut memerlukan gangguan pada pengalaman virtual?
- Bagaimana mobilitas pengguna?
- Berapa lama pengalaman tersebut?
- Dapatkah pengguna memakai headphone?

Beberapa pertimbangan tingkat kedua mungkin adalah:

- Apakah perangkat VR dan/atau AR akan higienis dan bagaimana?
- Setelah pengalaman tersebut, apa proses untuk menjaganya tetap seperti itu?
- Siapa yang akan memfasilitasi pengalaman tersebut dan berapa lama waktu yang dibutuhkan?
- Apakah pengguna merasa aman?

Sekali lagi, cakupan 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 ruang ini. Menambahkan penelitian pemangku kepentingan yang tepat dan pemikiran ke depan 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 bagi 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 Kerja untuk Penilaian Komprehensif Teknologi Medis: Menentukan Nilai dalam Ekosistem Perawatan Kesehatan Baru,” yang dikembangkan bersama dengan Deloitte Consulting LLP

11.2 PENGALAMAN PENGALAMAN PENGGUNA STANDAR TIDAKLAH INTUITIF

Desain aplikasi VR dan AR telah berevolusi secara signifikan dalam beberapa tahun terakhir saja, menstabilkan cara pembuatan berbagai sistem dan penyelesaian tugas. Tidak perlu mencari lebih jauh lagi selain Virtual Reality Toolkit (VRTK) (lihat Bab 7). Namun, jika kita melihat “pergerakan” sebagai contoh yang sering ditemukan dalam aplikasi VR, pergerakan adalah cara pengguna dapat bergerak melalui lingkungan virtual ada banyak bentuk, mulai dari kurva Bézier hingga “memilih orientasi” hingga teleportasi berbasis titik arah dalam kanon pengalaman pengguna VR (UX).

Namun, meskipun demikian, ketika digunakan di luar konteks permainan atau mereka yang berkecimpung dalam industri teknologi imersif, jelas bahwa hal-hal yang mungkin tampak biasa dan mudah dipahami sebenarnya tidaklah demikian. Hal ini bahkan berlaku untuk gerakan input dasar untuk perangkat keras AR karena banyak yang belum pernah menggunakan sistem ini sebelumnya. Oleh karena itu, bekerjalah dengan cerdas tanpa antarmuka pengguna jika memungkinkan; misalnya, dengan membuka langsung ke lingkungan virtual utama tempat aktivitas inti berlangsung.

Jenis input umum lainnya yang terlihat dalam aplikasi tertentu adalah pelacakan tangan, yang memungkinkan pengguna melihat tangan virtual yang mencerminkan gerakan mereka sendiri melalui teknologi seperti Leap Motion. Kita dapat melihat ini dalam salah satu pengalaman pembelajaran yang diwujudkan baru yang berfokus pada pasien kanker dari Embodied Labs, yang merupakan bentuk untuk membuat pengguna tetap terlibat tetapi juga bentuk untuk memerangi keterputusan yang dirasakan pengguna ketika mereka tidak melihat tangan mereka dalam VR. Yang perlu diperhatikan secara 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 asing. Lingkungan virtual yang perlu "diatur ulang" atau "kedaluwarsa berdasarkan waktu" harus diulang atau diatur sedemikian rupa sehingga tidak bergantung pada klik untuk memulai ulang.

Dalam contoh proyek yang akan datang, Insight, untuk meminimalkan klik yang tidak disengaja, pengguna harus menahan pengontrol selama waktu minimal agar tindakan dapat diterima. Komponen ini, yang dikenal sebagai penambahan 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 menjadi titik frustrasi, jangan biarkan hal itu bertahan lama bagi pengguna ubahlah dengan cepat.

Pilih Lingkungan yang Tenang

Dengan proyek Insight, berbagai langkah diambil untuk menciptakan lingkungan yang kontras dengan lingkungan pasien atau lingkungan belajar pada umumnya. Penempatan di dekat air dan audio dalam bentuk lonceng angin yang lembut mengundang rasa rileks.

Penggunaan VR dalam perawatan paliatif meningkatkan kualitas hidup melalui lingkungan yang indah. Hal ini tidak diragukan lagi sebagian karena kemauan pengembang untuk berpikir dalam konteks "ruang dunia" bukan "ruang layar" dan adegan papan cerita dari sudut pandang atas untuk memastikan pemirsa tertarik. Menggunakan plug-in audio spasial

untuk isyarat relaksasi (contohnya termasuk gemerisik lembut, angin bertiup melalui beberapa lonceng, atau suara ombak) akan membantu menarik pengguna ke dalam lingkungan dan meningkatkan kredibilitas.

Kenyamanan

Cara lain VR dan AR digunakan dalam organisasi kesehatan adalah menciptakan efisiensi ekonomi. Untuk menghemat waktu dokter dan perawat di klinik, misalnya, Augmedix menawarkan platform otomasi dokumentasi yang didukung oleh pakar dan perangkat lunak manusia. Meskipun platform pilihannya adalah Google Glass (dan karenanya tidak sepenuhnya AR), penyampaiannya 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 sebesar 30%.

Di bagian berikutnya, kami membahas cara mengotomatiskan penilaian sentuhan jari-hidung untuk tremor visuomotor, yang juga merupakan tujuan dari makalah yang diserahkan ke Arxiv pada tahun 2018 oleh tim di balik Insight. Karena uji sentuhan jari-hidung sebagian besar dilakukan melalui logika permainan yang dibangun dalam aplikasi berbasis Unity, hal ini memungkinkan dokter untuk melanjutkan tugas-tugas lain dan memberikan kemudahan.

Tutorial: Eksperimen Insight tentang Parkinson

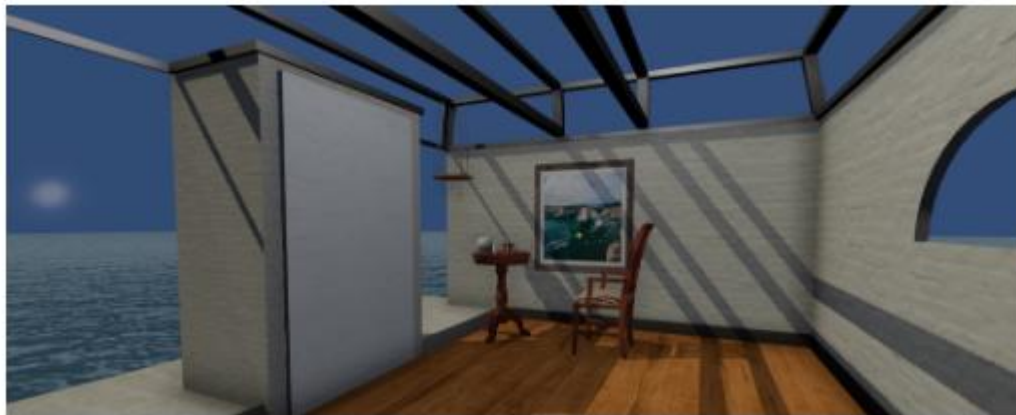
Penyakit Parkinson adalah gangguan neurodegeneratif yang perkembangannya lambat dengan gejala-gejala seperti tremor, kekakuan anggota tubuh, gerakan yang lambat, dan masalah keseimbangan. Perkembangan penyakit ini dapat sangat memengaruhi kualitas hidup, mulai dari ketidakmampuan fisik hingga depresi. Insight adalah platform VR yang berpusat pada pasien untuk pasien penyakit Parkinson dan keluarga mereka. Insight dibangun berdasarkan pengamatan mendasar bahwa pengendali VR normal dapat mengirimkan data posisi dan orientasi frekuensi tinggi.

Apa yang Dilakukan Insight

Insight berfungsi sebagai alat penilaian VR, manajemen, dan aplikasi pendidikan kesehatan. Pasien biasanya menemui dokter untuk memantau perkembangan penyakit dan menyesuaikan pengobatan serta terapi rehabilitasi pada kunjungan klinik interval tertentu berdasarkan gejala. Insight tetap bersama pasien sepanjang hidup mereka, terus menilai pengguna di antara kunjungan dan membantu penyedia layanan mereka di klinik. Platform ini memanfaatkan perawatan terkini melalui informasi kesehatan pihak ketiga seperti catatan medis dan data gerakan yang dikumpulkan dalam VR untuk membuat penilaian status kesehatan pasien, menyediakan latihan rehabilitasi yang dipersonalisasi, dan memandu tim dokter dalam pengambilan keputusan berdasarkan data.

Sebelum pasien memulai latihan rehabilitasi, mereka menyentuh seperangkat 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 yang dipersonalisasi berdasarkan bukti yang selain

meningkatkan fungsi fisik juga mengumpulkan data untuk penilaian perkembangan penyakit. Di akhir penilaian, pasien menerima ikhtisar status kesehatan mereka saat ini, termasuk pengobatan, 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 yang lembut agar pasien dapat lebih rileks saat melakukan penilaian jangkauan

Insight menyediakan platform untuk pengumpulan data sukarela.

Cara Pembuatannya

Platform Data Pasien Insight dibuat dengan menggunakan kombinasi mesin permainan Unity 2017.3.0f3 dan alat analisis data MATLAB dan Python.

Filter low-pass untuk tremor tangan

Bagian terpenting dari proyek ini melibatkan transformasi cara seseorang yang mengalami tremor bergerak dibandingkan dengan cara mereka bergerak saat melihat tangan mereka sendiri melalui perangkat VR. Dibuat dengan menggunakan filter low-pass, atau moving average, Smoothed Hand C# Script yang dilampirkan ke model tangan pengguna menangkap data posisi dan rotasi transformasi dari objek input yang dilacak VR sebagai input dan mengeluarkan data yang dihaluskan untuk transformasi model tangan.

Lingkungan

Inspirasi untuk lingkungan sangat dipengaruhi oleh seorang 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. Dengan menggunakan sistem medan Unity, medan dibuat dengan menggunakan peta ketinggian dan kemudian pohon-pohon ditempatkan.

Analisis dan pelaporan data

Pada dasarnya, tim mengumpulkan nilai tremor X,Y,Z dari posisi pengontrol VR, dan setelah itu analisis data memberikan rincian yang relevan bagi pasien dan pengasuh tentang tremor.

Berikut pseudocode-nya:

Impor untuk mendukung analisis data dan fungsi

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
import os.path
```

Memuat 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 seputar 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 tremor dan inialisasi 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 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 diekstraksi menjadi variabel x, y, dan z, masing-masing, 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 bagian pertama (0 hingga pi dalam domain frekuensi ternormalisasi) didistribusikan ke variabel Xr, Yr, dan Zr, secara berurutan.
- ❖ Respons frekuensi dari setiap komponen (sebagai fungsi dari frekuensi ternormalisasi: 0 hingga pi) diplot (lihat Gambar 11.1).
- ❖ Skor yang mencerminkan deviasi standar (getaran) dari sinyal yang direkam dihitung—getaran yang lebih banyak akan menghasilkan skor yang lebih rendah (`skor=int((1-(1.07*(x2.std()+y2.std()+z2.std())))*100)`).
- ❖ Deviasi standar dari setiap sumbu (x, y, z) dihitung dan diplot dalam baris 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 telah sedikit berubah (direvisi).

Untuk singkatnya, kode yang tersisa untuk menyusun laporan PDF tentang pergerakan pasien dihilangkan; untuk melihat kode tersebut, buka repositori GitHub untuk buku ini. Diperlukan string dan nama file bersarang untuk media dalam blok kode pemformatan yang disediakan oleh pustaka ReportLab. Posting Devpost berisi video yang memamerkan aplikasi yang dihasilkan dalam tindakan, dan kode tersebut ditautkan ke dalam repositori GitHub. Perangkat keras yang digunakan:

- HTC Vive Aset eksternal yang digunakan:
- SteamVR

- Frames Pack
- Post-Processing Stack

Alat yang digunakan:

- Untuk analisis, paket yang digunakan meliputi NumPy dan Pandas
- Untuk visualisasi, Matplotlib Textures untuk model:
- CGTextures

11.3 PERUSAHAAN

Bagian berikut membahas perusahaan yang menggunakan VR dan AR untuk membantu orang dalam berbagai cara dalam perawatan kesehatan. Sebagai permulaan, Profesor Radiologi, Teknik Elektro, dan Bioteknologi Universitas Stanford, dan salah satu direktur IMMERS, Brian Hargreaves, PhD, telah mengartikulasikan uraian yang bagus tentang di mana letak nilai di sepanjang spektrum teknologi imersif di klinik. Sebagai latar belakang, IMMERS adalah inkubator untuk realitas campuran medis (MR) dan realitas eXtended (XR) di Universitas Stanford. MR atau AR berguna di area yang memerlukan informasi tambahan tentang pasien, seperti perencanaan, panduan, dan penilaian. Meskipun VR digunakan untuk komponen imersifnya, hal itu mungkin memudahkan dokter yang sedang menjalani pelatihan untuk memahami topik medis atau menjelaskan topik tersebut kepada pasien.

Perencanaan dan Panduan

Perencanaan dan panduan secara khas terkait dengan pembedahan dalam kasus penggunaan teknologi kesehatan VR dan AR, tetapi beberapa, termasuk perangkat lunak penyelarasan 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 itu sendiri. Misalnya, ahli bedah, pasien, dan keluarga dapat berdiri dengan arteri di sebelah kanan mereka, struktur dasar tengkorak tulang di kaki mereka, dan, dengan melihat dari balik bahu mereka, mereka dapat mengamati tumor atau patologi vaskular. Pengalaman imersif ini memungkinkan mereka untuk memahami patologi dan rencana pembedahan mereka.

Osso VR

Osso VR adalah platform pelatihan bedah VR tervalidasi terkemuka yang dirancang untuk dokter bedah, tim penjualan, dan staf rumah sakit dari semua tingkat keterampilan. Produk perusahaan ini menawarkan interaksi berbasis tangan yang sangat realistis dalam lingkungan pelatihan imersif yang berisi prosedur dan teknologi mutakhir dan terkini.

Archform

Archform, sebuah perusahaan perangkat lunak yang menyediakan alat koreksi gigi intuitif bagi dokter gigi ortodontis, menunjukkan bahwa bagi penggunanya, daya tarik 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 mengendalikan orientasi model dengan

cepat dan melihatnya dalam VR.

Pengalaman yang Dirancang untuk Pendidikan Kedokteran

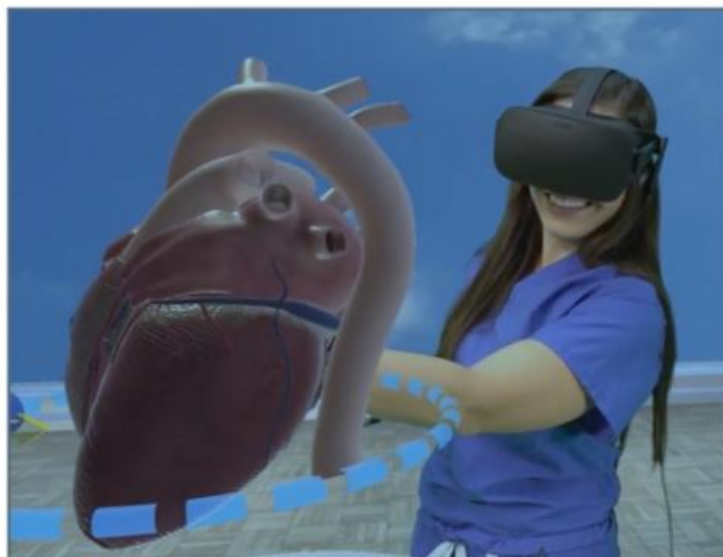
Semua pengalaman berikut ini sebanding, meskipun beberapa seperti Stanford Virtual Heart Project atau pengalaman Embodied Labs mungkin bermanfaat bagi jenis pengasuh tertentu.

Embodied Labs

Penceritaan dalam VR membutuhkan perhatian yang sangat besar terhadap detail, dan, khususnya jika tujuannya adalah 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 terkini menggunakan pelacakan suara dan tangan, yang memungkinkan pengguna untuk berperan sebagai pasien di hadapan anggota keluarga selama peristiwa penting selama tahap kritis berbagai penyakit.

Lighthouse

Lighthouse, sebuah perusahaan teknologi San Francisco Bay Area, dan David Axelrod, MD di Rumah Sakit Anak Stanford Lucille Packard, seorang ahli jantung anak, berkolaborasi dalam sebuah proyek yang disebut Stanford Virtual Heart Project (SVHP). Digunakan oleh mahasiswa dan praktisi, SVHP dibuat untuk menguraikan berbagai kelainan jantung anak dan prosedur yang diperlukan untuk mengatasinya dalam pengalaman VR interaktif. Proyek ini membuka perpustakaan berisi hati, dan sebagai pengguna, Anda dapat menarik masing-masing hati ke area utama untuk interaktivitas. Pemirsa dapat memutar hati, seperti yang digambarkan pada Gambar 11.3, dan melihat prosedur yang diperlukan untuk menyembuhkan penyakit.



Stanford Virtual Heart Project

Gambar 11.3. Seorang pengguna yang mengenakan Oculus Rift memutar jantung virtual di tempatnya saat berdetak

MVI Health

MVI Health, usaha patungan antara Penumbra dan Sixense, adalah perangkat keras VR yang dirancang agar dapat dihapus dengan pengontrol yang diperlengkapi 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 (trombosit) dari pembuluh darah.

Kemudahan untuk dapat mengatur ulang semua peralatan dengan mengklik tombol, menghindari kekacauan, dan memungkinkan tinjauan kinerja menjadikan ini contoh yang jelas mengapa pendidikan medis memerlukan penawaran produk MVI Health daripada metode pelatihan lain untuk prosedur ini.

Better Lab

Better Lab, yang berpusat di University of California, San Francisco, menerapkan pemikiran desain pada masalah yang berpusat pada pasien. Saat ini, proyek VR perusahaan yang mencakup pasien trauma dan didanai oleh hibah HEARTS hampir selesai untuk headset mandiri tahun depan. Zuckerberg San Francisco General (ZSFG) adalah satu-satunya pusat trauma Level Satu di San Francisco. Dari 255 kasus trauma yang dirawat setiap bulan, 90 di antaranya adalah "aktivasi 900" tingkat tinggi yang memerlukan kecepatan dan koordinasi intensif di berbagai departemen. Setiap konfigurasi tim trauma baru karena penyedia dan staf berganti-ganti berdasarkan shift dan bulan. Untuk memperhitungkan 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 diberikan di fasilitas trauma Level Satu.

Pengalaman yang Dirancang untuk Digunakan oleh Pasien

Perusahaan-perusahaan berikut menerapkan AR dan VR dengan cara yang memberikan manfaat langsung kepada 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 pemandangan 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 keduanya lebih mudah bekerja sama. Seiring perkembangan pasien, tujuannya adalah agar tidak perlu lagi modifikasi gambar untuk menggabungkannya dan melihat lebih dalam sepanjang waktu. Setiap minggu, pasien membutuhkan sedikit bantuan, sehingga perbedaan antara kedua mata menjadi semakin kecil. Dengan latihan, kedua mata belajar cara bekerja sama dan bekerja sama.

VRHealth

VRHealth mengkhususkan diri dalam mengembangkan alat dan konten medis sambil memberikan analisis waktu nyata. Produknya, VRPhysio, terdaftar di FDA sebagai alat latihan dan penilaian rentang gerak. Untuk memulai, VRPhysio dibuka dengan penilaian Rentang Gerak (ROM), seperti yang ditunjukkan pada Gambar 11-4. Seorang terapis fisik virtual

memperagakan gerakan-gerakan tersebut dan aplikasi tersebut memungkinkan penyesuaian sesi-sesi pasien sesuai dengan penilaian ROM dan rencana perawatan mereka.



Gambar 11.4 Bagian dalam lingkungan virtual dan avatar untuk membantu pelaksanaan uji ROM

Kemudian, mereka memilih video pendek dari berbagai konten—klip musik, Ceramah TED, film pendek, dan banyak lagi. Terakhir, laporan ringkasan terperinci dibuat untuk setiap sesi pelatihan.

USC ICT Bravemind (Atas izin USC Institute of Creative Technologies)

Bravemind adalah aplikasi untuk dokter yang mengkhususkan diri dalam menangani PTSD. Aplikasi ini menyediakan alternatif untuk melakukan paparan di zona perang dan/atau membuat trauma ulang pada individu dengan PTSD terkait pertempuran. Dua lingkungan virtual utama meliputi Irak dan Afganistan. Pasien dapat terlibat dalam patroli jalan kaki, konvoi, evakuasi medis melalui helikopter dalam berbagai skenario. Setiap skenario memungkinkan dokter untuk menyesuaikan lingkungan agar mencakup ledakan, baku tembak, serangan pemberontak, dan bom pinggir jalan.

Luasnya pasukan koalisi dan cedera warga sipil, kerusakan pada kendaraan (jika skenario konvoi digunakan) dan arah ledakan dapat diubah. Efek suara mencakup suara khas zona pertempuran (misalnya, tembakan senjata), suara bising kota (misalnya, panggilan untuk salat, serangga berdengung), obrolan radio, pesawat di atas kepala, dan banyak lagi. Umpan balik vibrotaktil memberikan sensasi yang biasanya dikaitkan dengan gemuruh mesin, ledakan, baku tembak, dan suara bising di sekitar yang sesuai. Mesin aroma dapat digunakan untuk memberikan aroma yang relevan dengan situasi (misalnya, mesiu, bahan bakar diesel, sampah, bubuk mesiu).

Firsthand Technology, 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 membangun bidang pengendalian rasa sakit VR dan membantu membangun aplikasi pereda nyeri VR pertama, SnowWorld.

Penelitian pengendalian nyeri VR bermula dari Ramachandran dan Rogers-Ramachandran (1996) yang menemukan hubungan antara gambar visual sintetis dan nyeri fisik saat mereka menggunakan "kotak realitas virtual" berteknologi rendah yang terbuat dari cermin untuk meredakan nyeri anggota tubuh semu pada pasien 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, yang menunjukkan bahwa VR yang dihasilkan komputer dapat mengurangi nyeri pasien secara signifikan. Sejumlah penelitian selanjutnya yang menggunakan SnowWorld menemukan bahwa VR secara signifikan lebih efektif daripada pengalihan lain seperti film dan permainan komputer berbasis layar.

Firsthand Technology telah menyusun daftar referensi utama dan artikel jurnal tentang penelitian pereda nyeri VR dalam VR Pain Relief Bibliography-nya.

11.4 KESEHATAN PROAKTIF

Ketika memikirkan perawatan kesehatan di Amerika Serikat, sering kali dikaitkan dengan konotasi reaktif. Seseorang jatuh sakit dan kemudian mencari antibiotik, seseorang mengalami serangan jantung dan kemudian membutuhkan statin dan obat kolesterol, dan seterusnya. Kesehatan proaktif atau preventif didefinisikan sebagai pengoptimalan untuk kesehatan pribadi saat kesehatan Anda relatif stabil. Misalnya, gagasan untuk berolahraga termasuk dalam kesehatan proaktif karena dapat membantu mengurangi faktor risiko berbagai penyakit jika dilakukan secara konsisten. Perusahaan-perusahaan berikut berperan dalam kesehatan proaktif menggunakan VR atau AR.

Black Box VR

Black Box VR mengambil VR dan memadukannya dengan ilmu olahraga dan penelitian perubahan perilaku selama puluhan tahun untuk menemukan kembali konsep pusat kebugaran fisik. Didirikan oleh mantan CEO Bodybuilding.com dan beberapa orang lainnya, Black Box VR menggabungkan konsep permainan VR dengan mesin resistensi dengan beberapa contoh termasuk mesin kabel yang dapat secara otomatis menyesuaikan diri untuk memenuhi kriteria berat dan tinggi pemain.

YUR, Inc.

YUR menggunakan komputasi spasial (baik AR maupun VR) untuk membuat individu lebih aktif, terlibat, dan terinformasi. Dari data yang dikumpulkan dari beberapa individu yang menggunakan judul-judul aktif terkenal seperti Beat Saber, YUR telah menemukan bahwa pembakaran kalori dari penggunaan permainan VR dapat menjadi signifikan. Peran YUR adalah untuk menunjukkan data kesehatan kepada pengguna yang didasarkan pada masukan VR saja. VR sebagai alat penurunan berat badan yang efektif secara umum tidak disengaja. VR memikat indera dan tubuh sehingga menyebabkan penggunaanya cukup banyak bergerak sehingga penurunan berat badan dapat menjadi manfaat tambahan. Ini adalah perubahan paradigma yang bersejarah karena kebugaran terkenal karena kegagalannya dalam merangsang pikiran. YUR melihat kombinasi antara sifat menghibur dari permainan dengan manfaat fisik dari olahraga sebagai gerakan sejati menuju perubahan format kebugaran yang stereotip.

Studi Kasus dari Institusi Akademik Terkemuka

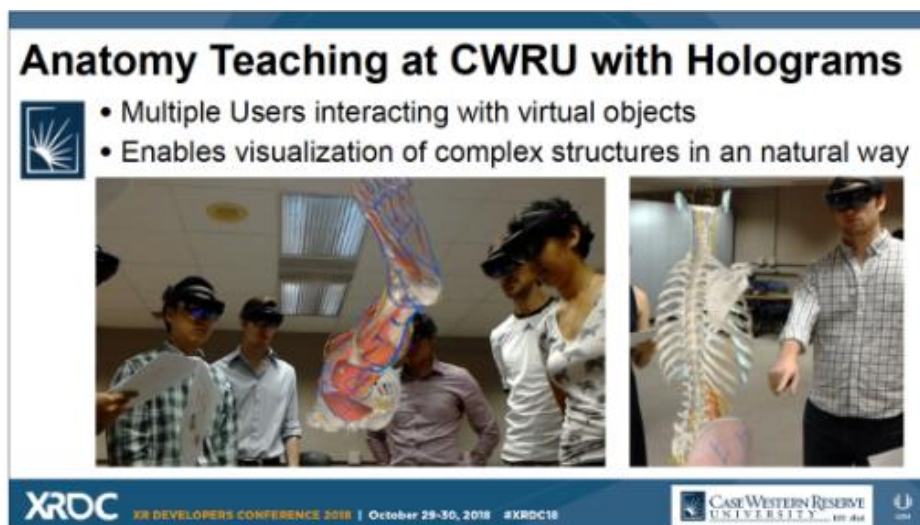
Meskipun institusi akademis yang dibahas di sini hanya mencakup University of California, San Francisco, Stanford, dan Case Western Reserve University, banyak institusi akademis lain yang berupaya untuk memungkinkan dan memberlakukan solusi atas tantangan nyata dalam perawatan kesehatan melalui kolaborasi kelompok AR dan VR.

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 memperlihatkan letak lesi pada payudara pasien untuk melapisi lesi pada lokasi sebenarnya, seperti yang diilustrasikan pada Gambar 11.5.



Gambar 11.5. Citra MRI payudara dan lesion (terlentang) yang disejajarkan dengan pasien dan dilihat menggunakan Microsoft HoloLens



Gambar 11.6. Dalam gambar ini, siswa menggunakan HoloLens dengan tangan mereka untuk berinteraksi dengan model

Ringkasan hasil studi tersebut mencakup peningkatan awal dalam semua ukuran; namun, menyelaraskan tampilan AR dengan pasien masih menjadi tantangan. Peningkatan penyelarasan di masa mendatang berasal dari bidang seperti visi komputer dan pelacakan tanpa penanda. Departemen Radiologi Stanford Medicine telah mengilustrasikan kemungkinan ini dengan kamera RealSense Intel. Gambar 11.6 menunjukkan perbandingan aplikasi HoloLens dengan prosedur standar untuk memperkirakan lokasi tumor yang teraba.

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 banyak orang yang melihat model ini dapat membantu siswa menyelesaikan kesalahpahaman bersama dengan cepat.

Memajukan pemetaan citra MRI menggunakan AR selangkah lebih maju adalah alur kerja Case Western University untuk MRI waktu nyata dan rendering HoloLens (Gambar 11.7). Hal ini memungkinkan penggunaan tampilan HoloLens yang intuitif dari data MRI volumetrik karena data tersebut diperoleh dengan waktu tunggu yang sangat sedikit.



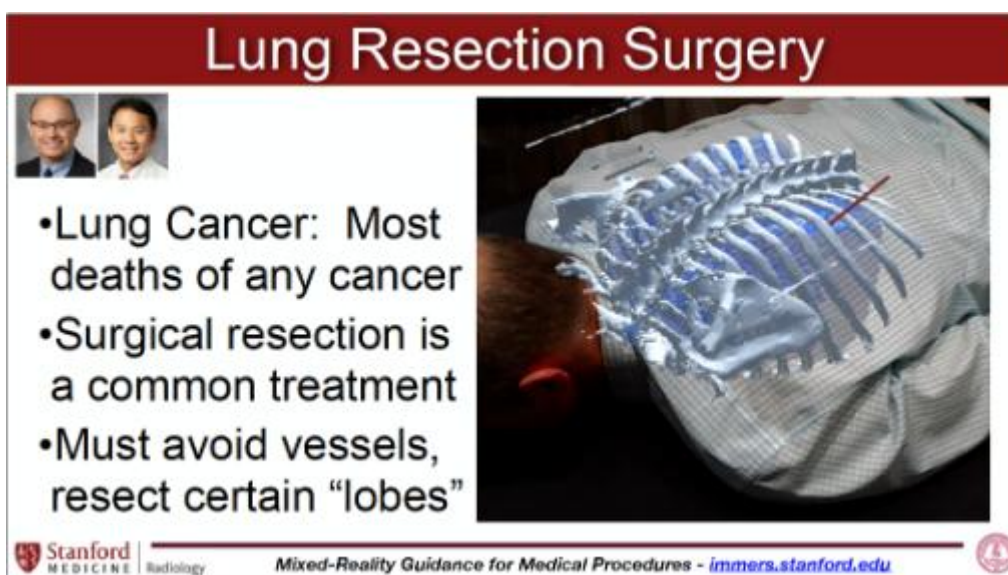
Gambar 11.7. Alur kerja rekonstruksi MRI berbasis Unity (konduktor MRI mengenakan Microsoft HoloLens dan dari sudut pandang mereka, memungkinkan untuk melihat data MRI pasien secara waktu nyata)

Stanford juga memiliki beberapa aplikasi lain yang menggunakan HoloLens untuk memanfaatkan lapisan AR pada pasien demi perawatan yang lebih baik, di mana objek yang digunakan dalam perawatan standar rutin dilacak posisinya. Dua contoh objek ini adalah tongkat ultrasonografi dan jarum, seperti yang diilustrasikan pada Gambar 11-8. Dalam kedua kasus tersebut, seorang praktisi dapat menggunakan data yang dilacak untuk melihat area tubuh atau membuat garis dengan lebih tepat.



Gambar 11.8. Seorang praktisi mendapatkan lapisan data gambar ultrasonografi secara langsung di atas lengan pasien (biasanya, praktisi perlu melihat layar terpisah tanpa MRI)

Dampak dari penambahan di tempat belum benar-benar dijelaskan, tetapi menarik untuk membayangkan bagaimana hal ini dapat memengaruhi, misalnya, kecepatan pemberian perawatan atau faktor lainnya. Apakah format tampilan ini akan meningkatkan kemampuan diagnostik dan mengurangi kesalahan?

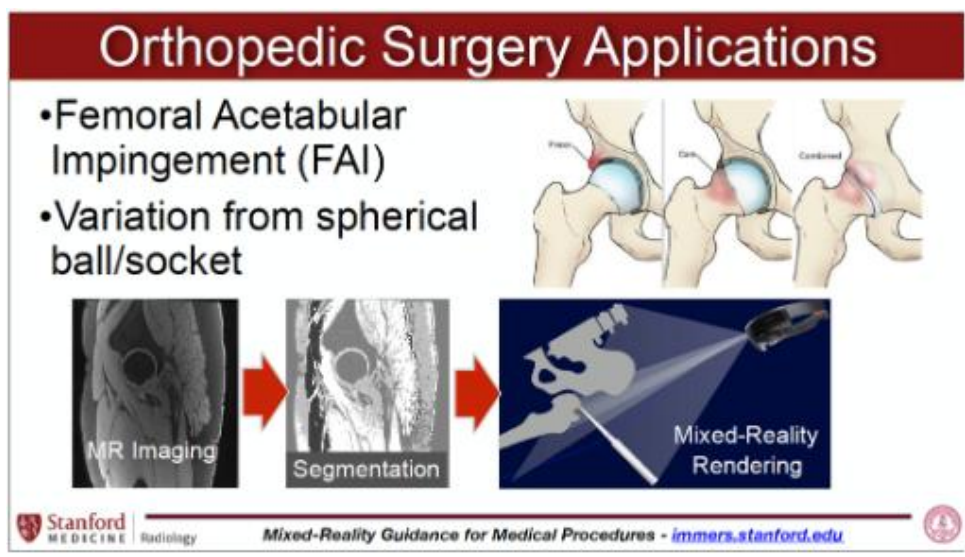


Gambar 11.9. Di sini, pasien diperbesar dengan paru-paru virtual saat dokter mempersiapkan perawatan reseksi paru-paru

Ada juga aplikasi MR untuk perencanaan dari (lihat Gambar 11-9) Stanford dalam bidang bedah termasuk transplantasi ginjal, reseksi paru-paru karena kanker paru-paru, dan bedah ortopedi. Ini biasanya merupakan area di mana sayatan pada pembuluh darah atau "lobus" tertentu dapat menjadi masalah, dan penggunaan MR dapat memberi dokter UI baru,

lebih modern, dan lebih membantu untuk tugas perencanaan masing-masing di mana pasien sebenarnya diperbesar.

Upaya yang sedang berlangsung di Case Western Reserve University dan Stanford mungkin memimpin lembaga pendidikan tinggi kedokteran dalam membekali fakultas dan mahasiswa dengan teknologi VR dan AR. Contoh terakhir, yang ditunjukkan pada Gambar 11.10, yang diberikan oleh Stanford melibatkan penggunaan MR dalam berbagai konteks ortopedi.



Gambar 11.10. Aspek aplikasi bedah ortopedi mencakup representasi permukaan dan model 3D, peralatan (di dalam tubuh), reseksi, gerakan, dan simulasi benturan virtual

Sebagai penutup, aplikasi teknologi kesehatan VR dan AR serta ruang masalah mencakup perencanaan dan panduan, pendidikan kedokteran, terapi yang digunakan dalam lingkungan klinis, dan kesehatan proaktif, antara lain. Aplikasi ini menyatukan tim peneliti, programmer permainan, seniman, dan dokter dan memungkinkan terobosan yang berpotensi berarti bagi orang-orang dengan disabilitas motorik, seperti yang ditunjukkan dalam tutorial kode Parkinson's Insight dalam bab ini. Untuk proyek Parkinson's Insight, VR menyediakan sarana untuk mengukur uji sentuhan jari-hidung yang analog. Dalam tinjauannya, penggunaan VR dan AR dalam teknologi kesehatan akan berubah seiring perkembangan teknologi dan peningkatan kemampuan serta kematangan perangkat keras.

BAB 12

PENGALAMAN PENGGEMAR: SPORTSXR

12.1 PENDAHULUAN

Ini benar-benar saat 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 luar biasa. Olahraga telah menjadi salah satu kategori konten yang paling banyak dikonsumsi di pasar media digital global dan memajukan teknologi untuk lebih banyak pengalaman AR dan VR olahraga.

Berikut adalah aturan dasar yang perlu diketahui pengembang:

- Olahraga adalah acara yang aturannya ditetapkan, kontesannya berkompetisi, dan ada hasilnya.
- AR dan VR menggunakan teknologi untuk membuat dan menyempurnakan konten. Contoh terbaiknya adalah "lini First-and-Ten" dari Sportvision pada tahun 1998.
- Aksi langsung itu penting, aksi langsung menciptakan rasa takjub, antisipasi, keinginan untuk tidak ketinggalan.

Untuk mengeksplorasi raksasa ini lebih menyeluruh, bab ini dibagi menjadi tiga bagian:

- ✓ Lima prinsip utama AR dan VR untuk olahraga
- ✓ Evolusi berikutnya dari pengalaman olahraga
- ✓ Menciptakan masa depan

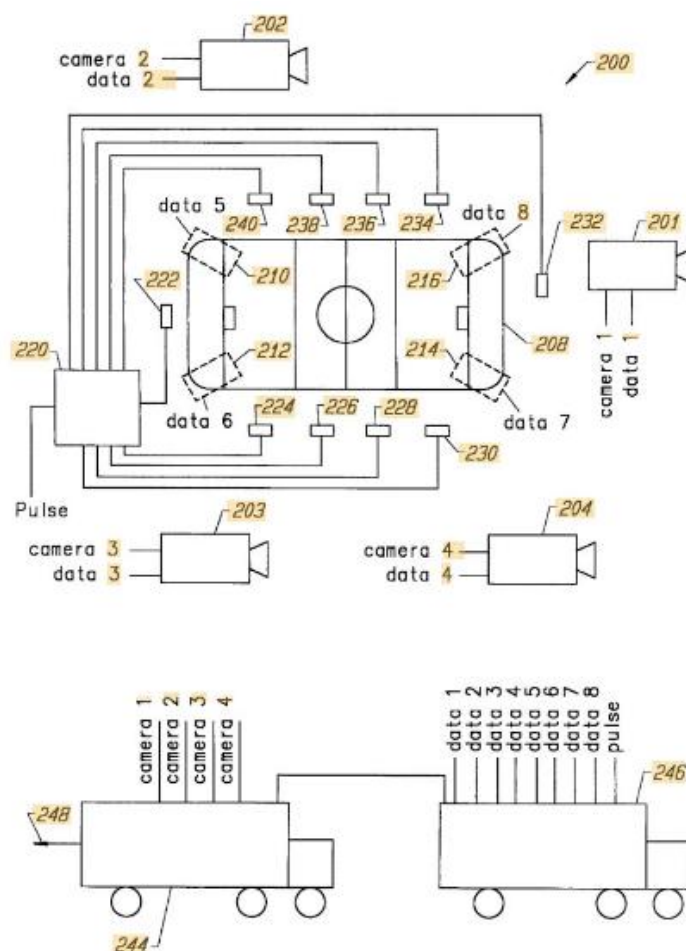
Pertama, perkenalan yang tepat. Saya Marc Rowley, dan saya menganggap diri saya sebagai pelopor AR/VR, telah bekerja pada AR langsung dalam olahraga selama lebih dari 20 tahun. Saya telah meraih lima penghargaan Emmy, beberapa penghargaan inovasi global, dan telah mendirikan beberapa perusahaan AR. Menurut saya, momen terbaik bagi seorang pendongeng adalah melihat reaksi audiens saat Anda menunjukkan sesuatu yang belum pernah mereka lihat.

Sasaran saat Anda membuat produk untuk penggemar olahraga adalah membuat sesuatu yang ajaib. Jika Anda melakukan ini, orang akan kembali lagi untuk menontonnya. Garis First-and-Ten untuk sepak bola Amerika dan garis offside dalam sepak bola/futbol/soccer adalah momen ajaib terbaik dalam olahraga. Garis-garis itu menunjukkan sesuatu yang Anda tahu ada di sana, tetapi Anda tidak dapat melihatnya tanpa teknologi.

Saya menghabiskan 18 tahun di ESPN untuk menciptakan teknologi baru seperti grafik rundown, buku pedoman virtual, dan tiang kamera multiview pertama. Saya meninggalkan jabatan ini di ESPN pada tahun 2017 untuk mengerjakan gelombang AR dan VR berikutnya. Saat ini, saya adalah CEO Live CGI. Tim saya telah menciptakan pemutar AR siaran langsung digital lengkap dari acara langsung dalam CGI (computer generated images) dengan kemampuan streaming simultan ke semua perangkat.

Sekarang, sebelum kita mulai, kita perlu mencapai garis dasar. Riset itu penting. Ya,

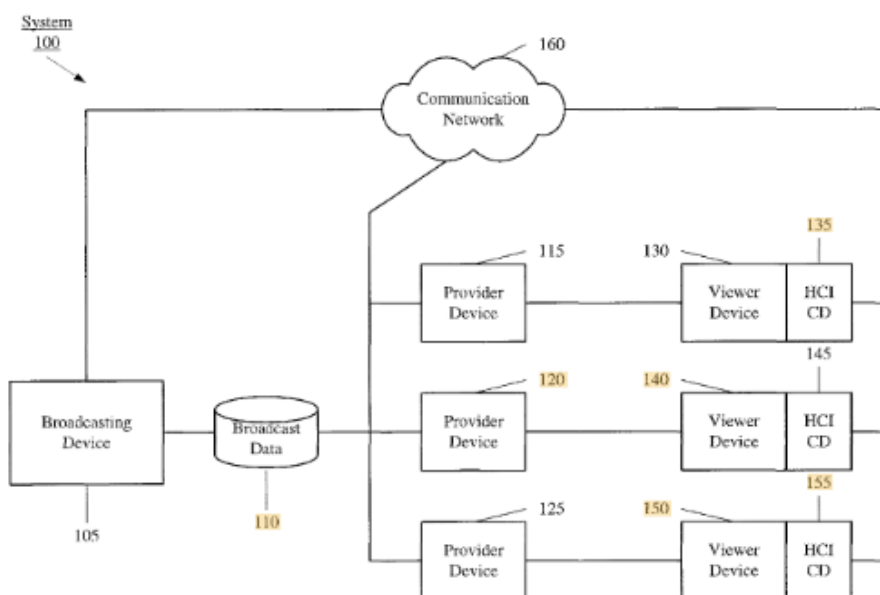
kami tahu Anda pernah mendengarnya sebelumnya—ketika Anda berurusan dengan produk olahraga, gandakanlah. Kenyataannya adalah ini, sangat mungkin seseorang telah memikirkan ide Anda. Mereka mungkin telah mengerjakannya atau memulainya sebelum Anda. Namun, jika Anda percaya dan dapat membuktikan bahwa masalah itu ada dan solusi Anda istimewa, ada dorongan luar biasa yang menanti Anda. Ketika Anda menciptakan dan menyampaikan produk yang mengubah cara orang melihat permainan mereka, Anda akan merasakan kegembiraan yang tiada tara. Untungnya, ide-ide terbaik dalam olahraga bersifat publik. Yang perlu Anda lakukan adalah memulai dengan pencarian paten.



Gambar 12.1. Jalur penggunaan kamera untuk menentukan posisi relatif terhadap lapangan permainan

Gambar 12.1 dan 12.2 menyajikan dua contoh cepat dari dua paten yang membentuk cara kita menonton olahraga langsung saat ini. Gambar 12.1 adalah paten untuk sistem untuk meningkatkan penyajian objek di televisi dalam acara olahraga, dan Gambar 12.2 adalah paten untuk menyajikan konten dan melengkapi siaran. Paten mungkin tampak tidak menarik untuk dibaca pada awalnya, tetapi jika Anda meluangkan waktu, Anda akan menemukan bahwa paten tersebut menciptakan jalur yang jelas menuju masalah dan solusi. Anda tidak perlu menghabiskan waktu berminggu-minggu, tetapi Anda harus meluangkan waktu setidaknya 40 jam untuk memastikan Anda memiliki dasar yang baik tentang apa yang telah Anda lalui

sebelumnya. Ini akan membantu meningkatkan peluang Anda untuk berhasil.



Gambar 12.2. Alur kerja tentang cara data dapat diproses untuk membuat ulang hasil permainan, yang merupakan kunci untuk memahami cara elemen AR dibuat

Sekarang, mari kita bahas apa yang perlu Anda ketahui saat membuat produk untuk olahraga.

12.2 LIMA PRINSIP UTAMA AR DAN VR UNTUK OLAHRAGA

Baiklah, panggung telah disiapkan untuk lima prinsip AR/VR dalam olahraga:

- ✓ Momen adalah segalanya
- ✓ Tidak ada yang langsung
- ✓ Gambar datar/rasa kehadiran
- ✓ Aturan 20/80
- ✓ Waktu itu penting

Mari kita mulai dengan mengajukan dua pertanyaan.

Mengapa olahraga istimewa? Seperti apa olahraga tanpa AR dan VR?

Jawaban untuk pertanyaan pertama mudah—momen; momen adalah segalanya. Olahraga memiliki tiga fase: pra-acara, acara itu sendiri, dan pasca-acara. Namun, setiap pertandingan, setiap acara, dapat didefinisikan oleh momen-momen penting. Momen-momen inilah yang paling didambakan orang. Mereka ingin berada di sana secara langsung, berada di momen itu, dibalut dalam semua kejayaan, rasa sakit, dan kemenangan. Olahraga adalah salah satu pelarian terhebat dari kenyataan yang pernah ada. Selama beberapa jam, Anda dapat meninggalkan dunia Anda dan hidup di dunia yang diciptakan untuk Anda agar dapat mengalami momen-momen menakjubkan.

Sekarang, mari kita bahas pertanyaan kedua.

Pertimbangkan tenis, di mana wasit membiarkan algoritma dan kamera virtual memutuskan apakah bola masuk atau keluar batas (Hawkeye). Anda mungkin berpikir tentang

siaran sepak bola Amerika di mana ada garis kuning garis first-down yang ditumpangkan di layar yang sebenarnya tidak ada di dunia nyata. Semua pilihan bagus, tetapi bukan di mana kita perlu pergi. Kita perlu mendekonstruksi esensi olahraga.

Mari kita mulai dengan menghilangkan tayangan ulang. Tidak ada lagi tayangan ulang. Itu sedikit mengubahnya, bukan?

Kemudian, hilangkan skor langsung dan grafik jam. Sekarang yang Anda miliki hanyalah umpan video dari sebuah pertandingan dan Anda tidak tahu siapa tim yang bermain dan berapa skornya. Sekarang, hilangkan siaran video, audio, dan pembaruan teks langsung.

Yang tersisa adalah arena tempat orang-orang bertanding, dan orang-orang datang untuk menonton. Itulah dasar yang perlu kita mulai dan tempat kita berada sekitar seratus tahun yang lalu. Semua lapisan yang baru saja kita kupas adalah AR. Faktanya, siaran itu sendiri adalah representasi VR. Ini perlu diulang, apa yang Anda lihat tidaklah nyata. Itu semua adalah presentasi yang dibuat untuk Anda.

Anda mungkin berpikir, tunggu sebentar. Saya dapat menonton olahraga di TV saya tanpa menggunakan headset VR. Ya, Anda bisa, dan apa yang Anda tonton bukanlah siaran langsung. Itu adalah representasi dari suatu acara. Seorang sutradara menunjukkan sudut kamera, dan di otak Anda, Anda membuat peta arena dan otak Anda menghitung perubahan dan mengisi kekosongan. Itu benar; kamera hanya menunjukkan sekitar 90% aksi dan kemudian sutradara membuat otak Anda membuat sisanya dengan potongan cepat dan grafik serta musik berirama cepat.

Ini adalah poin penting: jika kita tersesat di sini, sisa bab ini akan gagal.

Tidak Ada yang Langsung

Langsung adalah keyakinan kita bahwa apa yang kita proses di otak kita sedang terjadi pada saat kita melihatnya. Faktanya, jika Anda berada di lapangan menonton pertandingan, otak Anda memproses gambar dalam 13 milidetik. Ini berarti bahwa apa yang Anda lihat sebenarnya terjadi 0,0013 detik yang lalu. Namun bagi Anda, itu terasa seperti baru saja terjadi. Bagi Anda, itu benar-benar langsung. Namun, jika kami mencoba menjelaskan hal ini kepada konsumen rata-rata, mereka akan memiliki ekspresi yang sangat bingung di wajah mereka. Dalam olahraga, "langsung" secara umum diterima memiliki tiga tahap berbeda: Langsung, Langsung Langsung, dan Langsung untuk merekam atau langsung untuk merekam (LTT).

Bagi banyak orang, konten langsung dapat berupa apa saja dalam waktu satu menit setelah aksi terjadi. Misalnya, Anda mungkin menonton konten di rumah di Chromecast, tetapi Anda berbicara dengan seorang teman di pertandingan dan Anda mengetahui ada penundaan 40 detik untuk sampai ke Anda. Itu masih disebut langsung, tetapi aturannya sedikit dilanggar.

Live Live ditujukan untuk konten yang berdurasi empat hingga lima detik

Jika ditelusuri lebih jauh, jika Anda berada di sebuah apartemen di New York City dan menonton pertandingan bisbol di Bronx, kemungkinan besar Anda menontonnya dengan jeda 5 hingga 60 detik, tergantung pada layanan yang Anda gunakan. Hal ini terjadi karena sinyal video memiliki data. Data tersebut perlu ditransmisikan dari kamera ke pengalih produksi, lalu ke hub transmisi, yang kemudian mengirimkannya ke area pusat siaran, lalu dialihkan ke sinyal

TV atau sinyal internet, dipecah menjadi paket-paket, lalu dikirim melalui berbagai pengalih jaringan untuk dipasang di perangkat yang Anda gunakan. Inilah sebabnya mengapa orang yang berbeda dapat melihat video yang sama di waktu yang berbeda. Live adalah sesuatu yang kami yakini ada dalam olahraga, media berita, dan banyak bidang lainnya. Ketahuilah bahwa ini semua adalah konstruksi virtual yang kami pilih untuk disetujui sebagai live dan itu pun mungkin akan berubah sedikit.

Satu cerita yang ingin kami ceritakan dari akhir tahun 1990-an terjadi ketika data otomatis mulai ditampilkan di layar TV dan dapat diakses melalui internet. Konsumen mulai menyadari bahwa skor akan berubah sebelum mereka melihat videonya. Dalam satu contoh, seorang penjudi yang marah menelepon penyiar besar untuk mencari tahu bagaimana mereka dapat memprediksi skor dengan sangat akurat. Lucunya, semuanya bermuara pada matematika. File video lebih besar dan membutuhkan waktu lebih lama untuk dikirim. File data seperti jam dan skor lebih kecil dan membutuhkan waktu lebih sedikit. Jadi, Anda mungkin melihat pembaruan papan skor digital sebelum videonya. Sebagian besar penyiar besar sekarang memiliki kode untuk menangani hal ini, tetapi pada saat itu itu adalah kejutan besar. Hal ini sebagian besar telah hilang karena latensi dalam video telah berkurang, tetapi di beberapa tempat, Anda masih dapat melihatnya. Siaran langsung adalah cara Anda melihatnya.

Di seluruh dunia, penyiar menempatkan kamera di lokasi untuk menangkap gambar datar dari suatu acara yang dimainkan dalam ruang tiga dimensi. Mereka kemudian mengambil beberapa kamera dan memotongnya untuk memberi orang persepsi kedalaman, waktu, dan ruang penyiar menciptakan realitas virtual yang diterima sebagian besar dunia sebagai acara nyata. Namun, dalam bisnis, kita tahu bahwa metode ini memiliki keterbatasan.

Kamera hanya sebagus lokasi penempatannya, titik fokus, iris, bayangan, pencahayaan, kekuatan sinyal, piksel kembali ke ujung kepala produksi, dan, yang terpenting, tim yang menyusun representasi ini. Namun, jangan salah, Anda sedang menonton representasi datar dari dunia nyata.

Memproduksi acara olahraga datar memerlukan beberapa komponen. Operator kamera, sakelar, audio, operasi, kabel, dan masih banyak lagi. Mereka semua bekerja untuk menciptakan representasi virtual dari permainan dengan berbagai dimensi di layar datar Anda. Selama 80 tahun terakhir, beginilah cara orang menonton olahraga. Dan seperti halnya industri apa pun yang semakin matang, orang menemukan cara untuk menambahkan elemen dan meningkatkannya. Seperti halnya kategori media apa pun, 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 olahraga langsung:

1. Transmisi langsung
2. Audio penyiar
3. Video
4. Tayangan ulang
5. Grafik skor

Dan berikut adalah daftar lima fitur AR dan VR olahraga berikutnya dalam 20 hingga 30 tahun terakhir:

- Grafik penceritaan langsung di layar
- Sorotan
- Grafik aturan tambahan
- Media streaming langsung ke perangkat internet
- Interaksi media sosial

Masing-masing elemen ini telah membantu membentuk cara pengguna melihat dan merasakan olahraga. Semuanya dengan satu tujuan tunggal: memberi konsumen rasa kehadiran untuk mewujudkan rasa heran, takjub, dan antisipasi, semuanya untuk satu momen dalam olahraga yang penting, titik kritis di mana hasilnya diragukan dan konsumen condong ke depan menginginkan lebih.

Pikirkan tentang hal ini, bagaimana jika tidak ada sorotan, bagaimana Anda dapat menonton sebuah pertandingan, untuk mendapatkan ringkasan 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 First-and-Ten dalam pertandingan sepak bola? Bagaimana jika Anda tidak dapat melihat pesan sosial instan dari pelatih atau pemain favorit Anda? Masing-masing elemen ini melengkapi pengalaman olahraga kita.

Namun, yang akan datang berikutnya adalah yang benar-benar revolusioner. Pada akhir tahun 2000-an, saya bekerja di sebuah tim di ESPN yang berupaya mengambil teknologi First-and-Ten AR dan menggunakannya untuk meningkatkan penceritaan agar pemirsa menonton lebih lama. Itulah inti dari bisnis penyiaran. Anda berbisnis untuk menghasilkan uang sebanyak mungkin selama mungkin. Dalam olahraga, Anda menghasilkan uang itu dengan melakukan transaksi dengan pelanggan. Baik secara langsung dari dompet mereka kepada Anda atau secara tidak langsung dengan menjual perhatian mereka kepada mitra (pengiklan). Kita dapat melakukan ini melalui tayangan, pemirsa, dan cara lain. Yang menjadi terkenal selama bertahun-tahun adalah satu metrik: waktu yang dihabiskan. Waktu yang dihabiskan adalah faktor yang memotivasi banyak mitra. 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 punya pepatah yang sekarang menjadi kenyataan. Konsumsi konten bermuara pada aturan 20/80. Aturan 20/80 seperti ini. Jika Anda mengambil total audiens pengguna yang mengonsumsi dan menghabiskan waktu dengan produk Anda (konten/platform), 20% pengguna akan menjadi yang terberat. Mereka akan mencapai 80% dari tindakan Anda. 80% pengguna lainnya akan mencapai 20% dari konsumsi Anda.

Ini terjadi setelah audiens Anda matang. Pada bulan-bulan pertama produk Anda, mungkin akan menjadi gila atau mungkin perlahan-lahan berkembang, apa pun cara Anda dalam olahraga, aturan 20/80 telah terbukti menjadi metrik yang bagus. Tujuan saya adalah selalu menjaga 20% itu tetap tertarik dan melihat apakah saya dapat meningkatkannya menjadi 21% atau 22%. Alasannya adalah Anda tidak akan membuat konsumen biasa melakukan perubahan besar dalam hidup mereka. Hal itu tidak akan terjadi, tetapi jika Anda dapat membuat konsumen garis keras berubah, Anda akan melihat peningkatan pada

konsumen biasa. Namun, ini tidak berarti Anda selalu dapat menjangkau seluruh audiens. Lihat saja kebangkitan Esports. Konsumen ada di sana dan mereka kurang terlayani, sekarang seiring dengan meningkatnya konsumsi mereka, kita melihat pola yang sama. Hanya saja, pola Esports berkembang dengan kecepatan yang jauh lebih cepat karena konsumen beralih dari satu judul game ke judul game lain untuk mencari hal terbaik berikutnya.

Apa hubungannya ini dengan AR dan VR? Semuanya. Ini adalah dasar bagi 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 sedang membangun aplikasi untuk anak-anak di rumah sakit atau mengadakan Super Bowl, waktu adalah tujuan Anda. Jika Anda membuat pengguna meluangkan waktu, Anda dapat membuat dampak sosial, finansial, atau pendidikan—atau bahkan semuanya. Waktu itu penting.

Sekarang kita punya seperangkat aturan dasar, jadi mari kita rangkum lima poin prinsip utama:

1. Momen adalah segalanya
2. Tidak ada yang nyata
3. Gambar datar/rasa kehadiran
4. Aturan 20/80
5. Waktu itu penting

12.3 EVOLUSI BERIKUTNYA DALAM PENGALAMAN OLAHRAGA

Dalam beberapa tahun ke depan, kita akan melihat produk yang memenuhi kelima poin ini; produk tersebut akan menjadi panduan saat pengalaman olahraga didefinisikan ulang. Perubahan ini akan terjadi dalam skala global yang sebelumnya dianggap mustahil hanya karena menjamurnya perangkat internet dan minimnya kurva pembelajaran bagi konsumen baru.

Perubahan akan berlangsung cepat dan akan berfokus pada tiga area utama:

- ❖ Koneksi
- ❖ Tampilan
- ❖ Interaksi

Tingkat koneksi di dunia telah berubah dengan cepat seperti halnya sistem kompresi untuk mengirimkan data. Dulu, 3G adalah ciri khasnya, sekarang digantikan oleh 4G, 5G, LTE, dan setiap peningkatan lainnya yang akan hadir. Artinya, pada dasarnya, jumlah data yang dapat Anda bawa 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 aksi 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 berskala besar pada acara

olahraga yang tergantung di lapangan melaju 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 kabel serat optik ke konverter yang mengirimkan 20.000.000 bit data ke pengalih yang mengubah data tersebut menjadi 10.000.000 bit data. Akhirnya, data tersebut dicampur kembali menjadi 20.000.000 bit dan dikirim ke konsumen. Yaitu 9 gigabyte, menjadi 20 meg, menjadi 10 meg, dan kembali lagi menjadi 20 meg data per detik. Alasan mengapa data tersebut ditulis dengan semua angka nol adalah untuk memberikan cakupan penuh. 9.000.000.000 bit data setiap detik sungguh luar biasa, tidak ada perangkat seluler yang dapat menanganinya hari ini tetapi besok mungkin bisa.

Hal ini membawa kita kepada para inovator di perusahaan kecil dan besar yang berupaya memecahkan masalah kompresi dan kecepatan sinyal untuk menciptakan pengalaman AR dan VR yang berjalan pada layar baru. Headset VR lengkap, kacamata AR, dan perangkat lain semuanya memerlukan umpan data. Dan seperti yang telah kita bahas sebelumnya, kita semua memerlukan koneksi cepat agar konten dapat berfungsi dengan baik dalam olahraga. Yang terpenting adalah siaran langsung.

Dalam hal tampilan, masa depan akan dipetakan oleh prosesor grafis dan kemudian perangkat keras optik. Sebagian orang mungkin berpikir bahwa perangkat keras visual adalah yang utama; namun, Apple dan Google mengubahnya dengan meluncurkan ARKit dan ARCore. Unity, Unreal, dan lainnya telah menciptakan kerangka kerja untuk menciptakan pengalaman yang menakjubkan. 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 kerasnya.

Prosesor grafis umumnya 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. Meningkatnya "penambangan" Bitcoin telah mendorong kecepatan prosesor ke tingkat yang sangat tinggi karena para penambang bersaing untuk mendapatkan lebih banyak Bitcoin. Hal ini memberikan dorongan yang tak terduga bagi pekerjaan VR dan AR. Hal ini meningkatkan kekuatan untuk menciptakan pengalaman yang benar-benar mendalam dengan mendorong kecepatan prosesor. Perubahan terkini lainnya adalah kemajuan dalam ray tracing.

Semua faktor teknologi ini hebat, tetapi tidak ada artinya tanpa cerita. Cerita adalah segalanya. Orang tidak membeli teknologi, mereka membeli cerita, mereka membeli barang, dan menggunakan barang yang menceritakan kisah kepada mereka. Pelajaran tertua dalam pengalaman massal adalah ceritakan sebuah kisah kepada saya.

Evolusi ketiga dari pengalaman olahraga akan mengubah cara konten dibuat. Saat ini, anak-anak tumbuh dengan kontrol khusus dalam permainan, mereka mendapatkan umpan dari media sosial yang dipersonalisasi untuk mereka. Namun, ketika mereka menonton siaran,

mereka memiliki satu produser, satu sutradara, dan beberapa kamera yang diarahkan kepada mereka. Ini akan berubah ketika pemegang hak mendorong kreator dan distributor untuk memberi pengguna lebih banyak kontrol. Harinya akan tiba ketika sutradara yang digerakkan oleh kecerdasan buatan secara otomatis memotong urutan untuk menceritakan kisah langsung dalam paket yang diproduksi untuk setiap konsumen individu dari permainan yang sama. Perubahan besar pertama untuk ini adalah Esports, di mana pengguna dapat berkomentar, berinteraksi, dan mengendalikan pengalaman mereka. Dinding antara penggemar dan atlet mulai runtuh. Meskipun penyiar tradisional awalnya menolak hal ini, mereka perlahan berubah pikiran ketika angka pendapatan menunjukkan kenyataan yang kita semua tahu betul: konsumen selalu menang.

Ini adalah kumpulan data penting yang perlu dipertimbangkan saat Anda membuat suatu produk. Telah terbukti berkali-kali bahwa jika Anda mencoba menghentikan laju konsumsi konten, kekosongan dapat dan kemungkinan akan tercipta. Dalam kasus ini, ketika perusahaan media lama menghentikan interaktivitas Esports, mereka membiarkan Twitch.tv menarik banyak pemirsa.

Pikirkan tentang hal ini. Tidak ada yang menghentikan Sky Sports di Inggris Raya, ESPN di Amerika Serikat, atau FOX Sports, atau entitas lain mana pun untuk membuat Twitch; mereka hanya melewatkannya, dan sekarang Twitch.tv menjadi bagian dari Amazon dan menarik banyak pemirsa global.

Meskipun Twitch.tv memiliki berbagai jenis konten, dua yang menonjol di sini adalah streamer yang bermain game dan streaming properti Esports di platform mereka. Menurut Esports Observer pada tahun 2018, empat entitas streaming Twitch teratas menyumbang hampir 500 juta jam. Streamer 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. Jangan pernah lupakan aturan ini: konsumen selalu menang, dan mulai sekarang, ketahuilah bahwa semua konsumen menginginkan kendali.

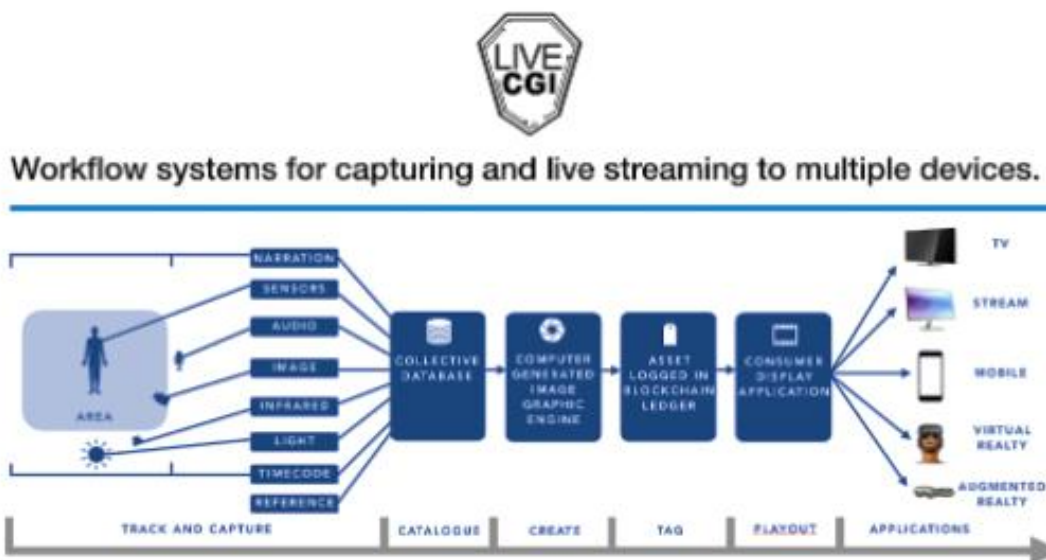
12.4 MENCIPTAKAN MASA DEPAN

Masa depan dibangun di atas inovasi masa lalu. Masa depan AR dan VR olahraga datang dengan kecepatan yang sangat cepat. Berkat proliferasi perangkat, fokus pada latensi, dan perubahan perilaku konsumen, tidak akan butuh waktu 80 tahun untuk pergeseran kuantum berikutnya. Sekaranglah saatnya untuk mendefinisikan ulang olahraga, sekaranglah saatnya untuk mengubah AR dan VR. Para pengembang yang membangun produk yang cepat, bercerita, dan menyediakan kontrol audiens akan menciptakan produk yang berhasil.

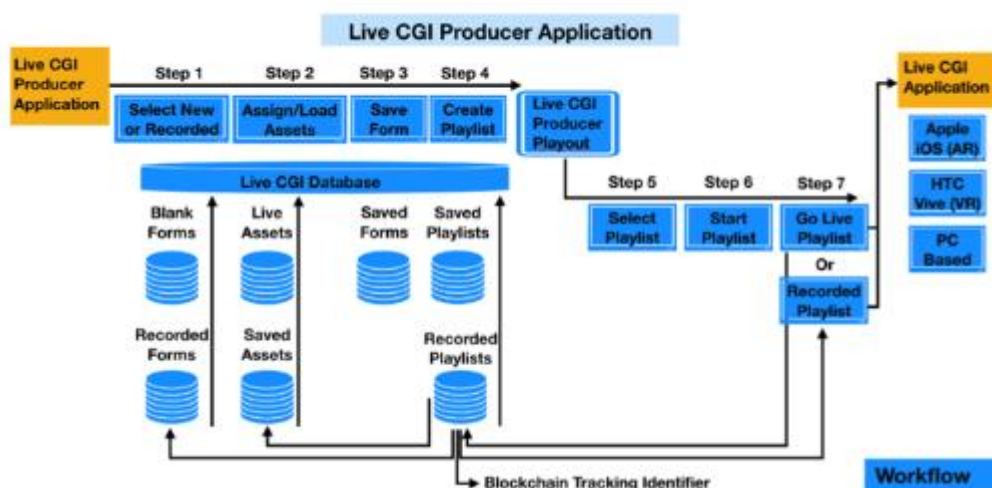
Saat Anda duduk untuk membangun masa depan ini, Anda harus memikirkan alur kerja Anda. Bagaimana konten dimasukkan, ke mana konten itu akan ditempatkan? Apa tampilan makro keseluruhan dan apa tampilan mikro? Untuk melakukan ini, Anda perlu membuat dua dokumen untuk tim Anda. Satu yang hanya menguraikan 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 formasi I atau shotgun; dalam bisbol, ini adalah perubahan; dalam Esports, ini adalah seberapa lama Anda melakukan jung- gle atau siapa pendukung Anda.

Kemudian, Anda perlu membuat tampilan mikro. Ini adalah naskah permainan: ke mana setiap aset akan ditempatkan; siapa yang akan menghalangi 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 lebih banyak detail. Dalam kebanyakan kasus, Anda tidak ingin membagikannya, karena ini adalah bagian dari saus khusus Anda untuk bagaimana produk Anda dibuat; namun, setelah Anda mengajukan paten, dan informasinya menjadi publik, itu adalah permainan yang adil bagi siapa pun. Jadi, seperti halnya segala sesuatu, semuanya bermuara pada eksekusi. Hal terbaik tentang olahraga adalah bahwa lapangan bermainnya bersifat publik, tidak seperti itu dengan semua produk. Dalam olahraga, seperti kata pepatah, "Anda adalah apa yang dikatakan catatan Anda."



Gambar 12.3. Alur kerja untuk menangkap tujuh set data langsung, menambahkan set data yang direkam, dan menyederhanakan semuanya menjadi proses presentasi visual—dengan mendorong semua titik data ke mesin CGI, narasi diubah menjadi asli untuk setiap kemampuan rendering platform masing-masing



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)

Sekarang ke detailnya. 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—fokuslah untuk menjadi cepat. Ini juga berarti Anda harus akurat. Tidak ada gunanya bagi siapa pun untuk menjadi cepat dan salah.

Selanjutnya, ceritakan sebuah kisah. Semua pengguna ingin dihibur. Ceritakan sebuah kisah tentang acara tersebut dan ceritakan kisah yang menarik bagi orang-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 untuk lebih banyak lagi. Sebagai pengembang, Anda tidak perlu menjadi pendongeng yang hebat, tetapi Anda perlu berbicara dengan seorang pendongeng. Anda perlu menemukan seorang pendongeng, seorang penulis, seseorang dengan bakat berbicara, dan memastikan bahwa produk Anda menceritakan sebuah kisah.

Panduan terakhir adalah memberi konsumen kendali. Berikan konsumen apa yang mereka inginkan dan biarkan preferensi, kesukaan, dan ketidaksukaan mereka menciptakan pengalaman.

Anda mungkin bertanya, "Baiklah, tetapi bagaimana cara mengujinya?" Nah, Anda dapat menguji latensi dengan matematika sederhana. Berapa lama waktu yang dibutuhkan untuk berpindah 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 kendali yang Anda berikan kepada mereka dengan memeriksa gerakan mereka. Anda dapat mengukur semua hal ini, tetapi dari mana Anda memulainya? Bagi tim saya, aturan 10 selalu berlaku. 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 hingga empat kali, Anda mungkin menunjukkannya kepada teman. Lima hingga sembilan kali, dan Anda menyukainya tetapi tidak membutuhkannya. 10 kali atau lebih dan Anda harus memilikinya.

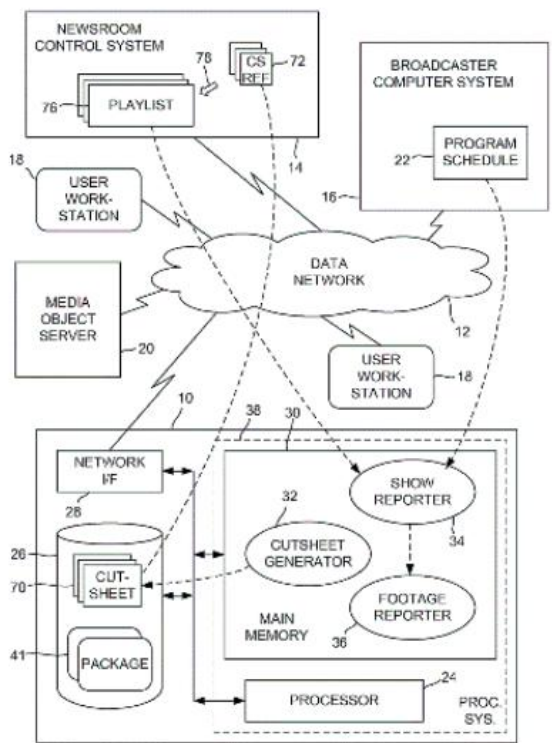
Contoh bagus dari hal ini terjadi dalam pertandingan sepak bola pada tahun 2008. New Orleans Saints memiliki pemain hebat bernama Reggie Bush, dan produk baru yang diluncurkan untuk pertandingan tersebut menunjukkan kecepatannya. Mereka menangkannya dengan kecepatan 22 mil per jam (mph). Meskipun ini sangat mengesankan, para penonton menghitungnya dengan mobil dan 22 mph tidak terdengar mengesankan. Namun, jika Anda memikirkannya, 22 mph dikonversi menjadi 32 kaki per detik, atau 10 yard per detik. Itu tidak penting; angka itu tidak beresonansi dengan penonton. Pikirkan tentang ini saat Anda membangun produk Anda, apakah itu akan beresonansi, apakah orang peduli, apakah itu memberi dampak?

Kepemilikan

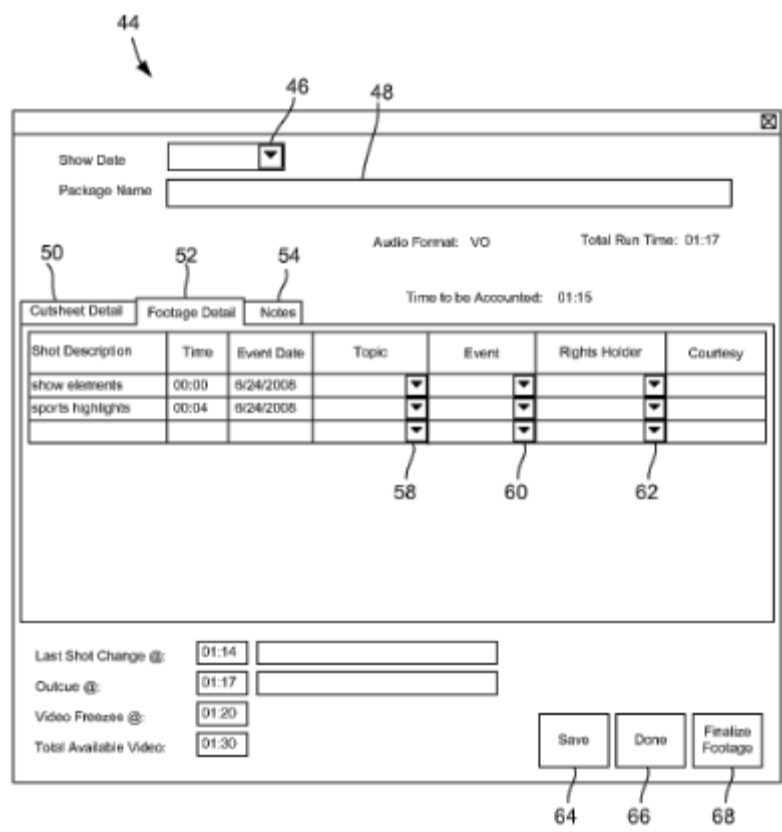
Sebelum kita mengakhiri bab ini, kita perlu berbicara tentang uang dalam olahraga. Ada area yang lembek dalam olahraga di mana orang berpikir liga besar menjual permainan atau acara kepada pemegang tiket. Meskipun ada beberapa pendapatan yang diperoleh dari titik penjualan individu itu, sebagian besar pendapatan selama lebih dari 50 tahun terakhir berasal dari hak media, lisensi, dan bundel konten. Ini berarti bahwa liga tidak terlalu banyak tentang olahraga dan lebih banyak tentang kekayaan intelektual dan konten. Namun, di permukaan, banyak orang mungkin memperdebatkan hal ini. Namun, ketika Anda melihat kontrak liga, melihat apa yang dibeli oleh perusahaan media, dan melihat nilainya, jawabannya sudah jelas. Olahraga adalah tentang konten. Perubahan ini telah memengaruhi cara produk dibuat dan menciptakan arus masuk uang tunai yang besar ke pasar.

Misalnya, jika Anda membuat penampil interaktif untuk rugby, Anda perlu melacak menit, jam, dan bagaimana konten tersebut digunakan. Ini adalah kunci bagaimana liga dan pemain memonetisasi kinerja mereka. Ini berlaku dalam Esports dan juga biliard. Penggerak bisnis utama untuk banyak transaksi bernilai miliaran dolar adalah waktu.

Ya, Anda perlu memikirkan latensi, Anda perlu menceritakan sebuah kisah, dan Anda perlu melepaskan kendali. Dan, Anda perlu melacak semuanya untuk dapat memberikan akuntansi atas pengalaman tersebut. Alasan mengapa ini tercantum terakhir di bagian ini adalah untuk menegaskan betapa pentingnya hal itu. Jika Anda membangun semuanya dengan luar biasa tetapi kehilangan akuntansi dasar atas pengalaman tersebut, Anda membatasi kemungkinan keberhasilan Anda serta potensi pertumbuhan Anda. Anda perlu memastikan 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 pelacakan sederhana untuk menangkap rekaman saat sedang dibuat dan kemudian mengatalkannya sebelum dikirim melalui streaming untuk konsumen. Sistem sederhana ini mendefinisikan ulang bagaimana kesepakatan hak rekaman dinegosiasikan pada akhir tahun 2000-an. Sistem ini tidak perlu rumit, hanya perlu melacak apa yang dikonsumsi dan bagaimana.



Gambar 12-5. Alur kerja tentang bagaimana sistem siaran menggunakan data yang diberi tag dan lembar potong untuk mengaitkan data dengan video



Gambar 12-6. Entri data untuk lembar potong, yang merupakan tempat produser individu menetapkan data prasetel ke video baru yang mereka buat (Keindahan sistem ini adalah

semua poin kesepakatan, batasan, dan proses bisnis individual disembunyikan dari pengguna, yang hanya diminta untuk memasukkan topik, acara, lokasi, dan kesopanan. Keempat item tersebut membuat katalog rumit yang membantu bisnis memonetisasi konten)

Pikiran Akhir

Anda mungkin tidak menyukai ini, tetapi begini: tidak ada yang peduli dengan perjuangan Anda sampai Anda menjadi bagian dari momen yang luar biasa. Ini mungkin terdengar kasar, tetapi semakin cepat Anda menerimanya, semakin cepat Anda bisa meraih kesuksesan. Dalam olahraga, kita belajar bahwa tidak seorang pun peduli dengan perjuangan atlet sampai mereka berhasil.

Saya memiliki angka di dinding kantor saya selama beberapa tahun, dan hanya sedikit orang yang pernah menebak apa artinya. Angka itu adalah 1.184. Itulah jumlah pemain yang merupakan atlet hebat 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 bagi banyak dari mereka, semuanya sudah berakhir. Mereka semua melakukan latihan selama 15 jam untuk bersiap menghadapi sepak bola profesional, hanya untuk mendapatkan kesempatan di kamp pelatihan. Kemudian, dalam sekejap, mereka diputus kontraknya.

Pikirkan ini saat Anda memulai produk Anda. Saat Anda membangun, saat Anda berkreasi untuk olahraga, fokuslah untuk membawa penggemar ke dalam perjalanan atlet yang bangkit untuk menghadapi momen tersebut. Bisa berupa pertandingan bisbol fantasi, aplikasi pelatihan, atau siaran langsung apa pun yang Anda lakukan, ciptakan sesuatu yang membuat orang ingin terus menontonnya lagi dan lagi.

Jangan bagikan bisnis Anda. Kesalahan terbesar yang dapat Anda lakukan saat penggemar terpicat pada produk Anda saat konsumen siap untuk momen itu adalah menunjukkan model bisnis Anda atau menunggu sesuatu terjadi.

Kesimpulan

Ini adalah waktu yang 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, siaran langsung yang sesungguhnya memengaruhi sebagian besar kehidupan orang. Saat Anda membangun produk, fokuslah pada momen, fokus pada latensi, fokus pada cerita, dan berikan kendali kepada konsumen.

BAB 13

KASUS PENGGUNAAN PELATIHAN PERUSAHAAN REALITAS VIRTUAL

13.1 PENTINGNYA PELATIHAN PERUSAHAAN

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 diperkirakan orang; pada tahun 2017, \$121,7 miliar dihabiskan untuk bermain game, tetapi \$362,2 miliar dihabiskan untuk pelatihan.

Pada tahun 2018, STRIVR mengirimkan 17.000 tampilan yang dipasang di kepala (HMD) Oculus Go ke Wal-mart. Itu berarti ada beberapa HMD di setiap toko Walmart di Amerika Serikat, dengan lebih dari satu juta karyawan Walmart memiliki akses ke pelatihan perusahaan dalam VR setiap hari. Itulah dampaknya. Gambar 13-1 menggambarkan gudang darurat STRIVR, tempat semua orang bekerja sama untuk melakukan kontrol kualitas pada setiap headset.



Gambar 13.1. 17.000 HMD sedang dipersiapkan untuk pengiriman (© STRIVR 2018)

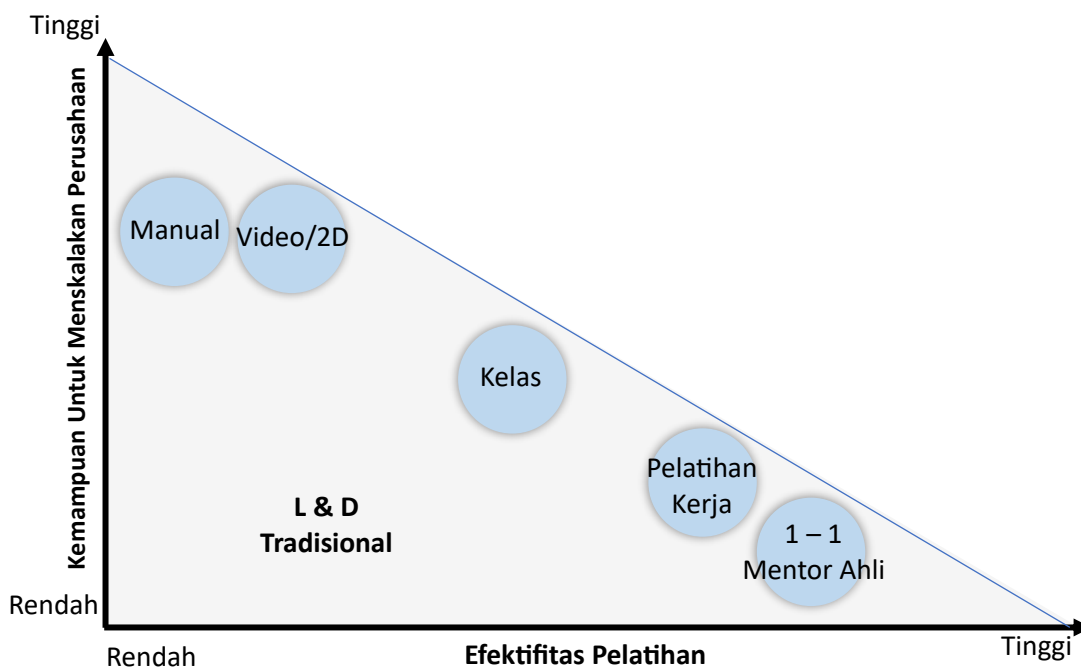
Agar VR berhasil, ia perlu memecahkan satu masalah spesifik dalam skala besar dan melakukannya lebih baik daripada teknologi lainnya. Pelatihan perusahaan adalah industri yang siap untuk ditransformasikan. Pelatihan perusahaan adalah masalah itu. Bab ini memaparkan kasus penggunaan, tantangan, dan pendekatan untuk membangun konten dan meningkatkan skala pelanggan pelatihan VR, dengan fokus pada video sferis sebagai media pelatihan.

Apakah Pelatihan VR Berhasil?

Cara terbaik untuk mempelajari sesuatu adalah dengan melakukannya. Untuk tugas-tugas seperti belajar menerbangkan pesawat terbang 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 menemukannya.

pisau bedah ke kulit. Membaca manual tentang suatu tugas adalah salah satu cara belajar yang paling tidak mendalam, sedangkan dipandu melalui suatu tugas oleh seorang ahli adalah salah satu cara yang paling mendalam. Pelatihan dalam VR saat ini tidak dapat menyamai instruktur manusia yang memandu Anda melalui suatu tugas secara langsung, tetapi pelatihan ini dapat mendekati meskipun jauh lebih murah dan lebih dapat diskalakan.

Gambar 13.2 menunjukkan diagram sebar dengan satu sumbu yang mewakili biaya dan skalabilitas, dan sumbu lainnya mewakili efektivitas. Di satu sisi, pertimbangkan buku panduan pelatihan. Anda dapat mengirimkannya ke mana saja, mencetaknya sesuai permintaan, atau membacanya di layar, tetapi buku panduan tersebut bukanlah alat pengajaran yang sangat efektif, terutama jika Anda mempertimbangkan untuk mengajarkan tugas fisik seperti mengikat simpul. Buku panduan pelatihan sangat dapat diskalakan, tetapi tidak terlalu efektif.



Gambar 13.2. Skalabilitas versus efektivitas opsi pelatihan (© STRIVR 2018)

Di sisi lain, ada bimbingan ahli secara langsung, bentuk pelatihan yang paling efektif. Seorang instruktur manusia mengetahui segala hal tentang subjeknya dan dapat memandu Anda selangkah demi selangkah, melibatkan Anda, membimbing Anda, menantang Anda, dan menanggapi kemajuan Anda. Namun, ini membutuhkan waktu yang berharga dari seorang ahli yang dibayar mahal. Bentuk pelatihan ini sangat efektif tetapi mahal dan sulit untuk diskalakan.

Janji VR adalah membangun sesuatu yang semurah teks digital untuk didistribusikan, 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 mengenakan HMD dengan pelacakan skala ruangan ditempatkan dalam lingkungan grafis komputer 3D tempat mereka

digantung pada ketinggian yang sangat tinggi di atas kota. Pada kenyataannya, pengguna berdiri di atas balok kayu yang bersandar di lantai, tetapi dari sudut pandang pengguna, mereka berada di ambang kematian. Hanya sedikit pengguna yang mencoba pengalaman ini yang dapat menyangkal efek fisik yang mendalam yang ditimbulkannya pada Anda: keseimbangan Anda goyah, kaki Anda lemas, dan setiap langkah maju membuat jantung Anda berdebar kencang. Anda belajar dengan 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 realitas daripada media pelatihan lainnya, dengan risiko dan biaya yang lebih sedikit.

Pelatihan VR sangat cocok untuk kebutuhan pelajar dewasa. Dalam *The Adult Learner*, Malcolm S. Knowles berpendapat bahwa orang dewasa memiliki kebutuhan belajar yang berbeda dengan anak-anak. Ketika orang dewasa belajar, mereka termotivasi oleh masalah-masalah praktis. Mengapa saya mempelajarinya? Bagaimana hal ini dapat bermanfaat? Dalam situasi kehidupan nyata seperti apa saya dapat menerapkan pengetahuan ini? Pelatihan VR memberikan banyak manfaat dibandingkan pembelajaran non-eksperiensial:

Keterlibatan

VR adalah lingkungan yang kaya akan interaksi, tempat pelajar terus-menerus diminta untuk terlibat. Hanya dengan mengenakan headset dan melihat sekeliling berarti Anda sudah berinteraksi. Dalam skenario interaksi "Pelatihan Perampokan Toko", misalnya, para 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 tersebut.

Konteks

Pelatihan VR yang baik menempatkan pelajar dalam lingkungan yang realistis, tempat keterampilan yang mereka pelajari akan berguna. Dalam skenario "Pelatihan Rumah Banjir", perbedaan antara air Kategori 1 dan Kategori 3 bukanlah sesuatu yang akademis; perbedaannya adalah antara mencabut dan mengganti semua lantai di rumah dibandingkan dengan sekadar mengeringkannya.

Motivasi

Pembelajar dapat melihat konsekuensi dari tindakan mereka. Menerapkan keterampilan baru secara efektif akan menunjukkan hasil yang baik, dan gagal menerapkan pembelajaran akan mengakibatkan bahaya. Misalnya, dalam skenario "Pelatihan Kabel Putus", gagal mengomunikasikan bahaya kabel putus mengakibatkan anjing peliharaan tersengat listrik.

Di STRIVR, ada peluang untuk melakukan studi kecil tentang kemanjuran pelatihan VR. Di bagian berikutnya, kita melihat kasus penggunaan di mana kemanjuran metode pelatihan VR diuji terhadap bimbingan ahli satu lawan satu.

Kasus Penggunaan: Pelatihan Rumah Banjir

Rumah banjir adalah rumah sungguhan yang sengaja dibanjiri beberapa kali dalam

setahun, sehingga para profesional asuransi dapat berlatih. Ada sekitar 30 rumah seperti 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 sesuai dengan kenyataan.

Membangun rumah, membanjirinya, dan kemudian memperbaikinya, tentu saja, mahal. Kerusakan yang terjadi pada rumah itu nyata dan mahal. Ditambah lagi, karena hanya ada beberapa rumah yang terendam banjir di setiap negara bagian, para peserta pelatihan harus diterbangkan ke lokasi tersebut.

Namun, pengeluaran biaya ini sepadan, mengingat besarnya jumlah uang yang dipertaruhkan. Misalnya, Badai Florence pada tahun 2018 menyebabkan kerugian asuransi sebesar tiga hingga lima miliar dolar. Tidak semua klaim asuransi dibuat dengan niat yang jujur, dan penipuan menyumbang sekitar 10% dari klaim. Memiliki profesional asuransi yang terlatih dengan baik yang dapat mengurangi jumlah tersebut sangat penting bagi perusahaan asuransi. Tetapi bagaimana jika mereka bisa mendapatkan hasil yang sama dengan biaya yang lebih sedikit?

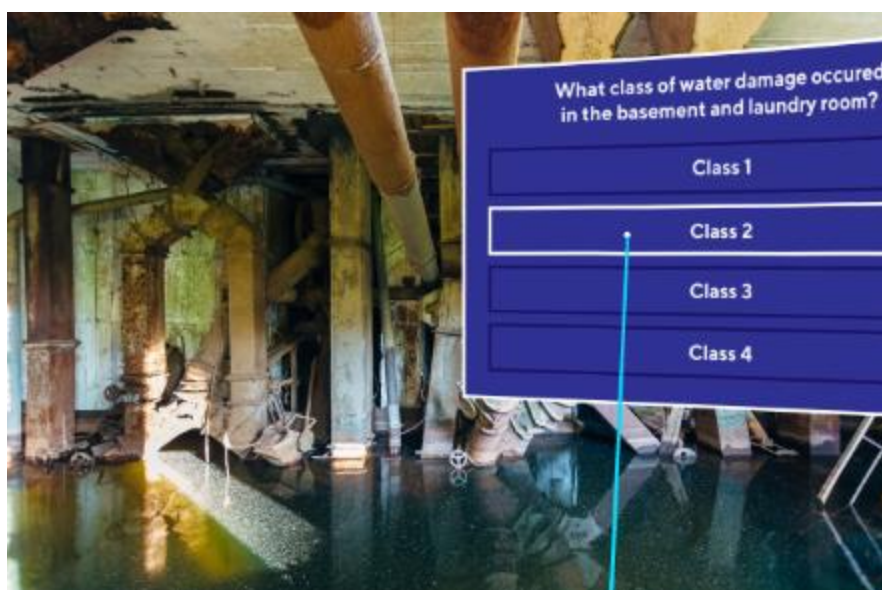
STRIVR bermaksud membuat versi VR dari kursus pelatihan rumah banjir, bekerja sama erat dengan instruktur ahli. Kru kamera merekam video bulat di rumah dari sudut pandang siswa yang diajari satu lawan satu, dan kemudian desainer membuat modul pelatihan VR menggunakan video tersebut. Dalam hal ini, modul pelatihan VR dibuat untuk mencakup secara komprehensif semua hal yang akan diajarkan selama kelas, terlepas dari apakah setiap aspek pelatihan "cocok" untuk VR.

Setelah modul pelatihan VR dibuat, STRIVR mengambil kelas yang terdiri dari 60 siswa dan mengajak setengah dari mereka ke rumah banjir sungguhan, dan setengah lainnya ke modul pelatihan VR. Tim data STRIVR kemudian menilai perbedaan efektivitas antara pelatihan rumah banjir sungguhan 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).

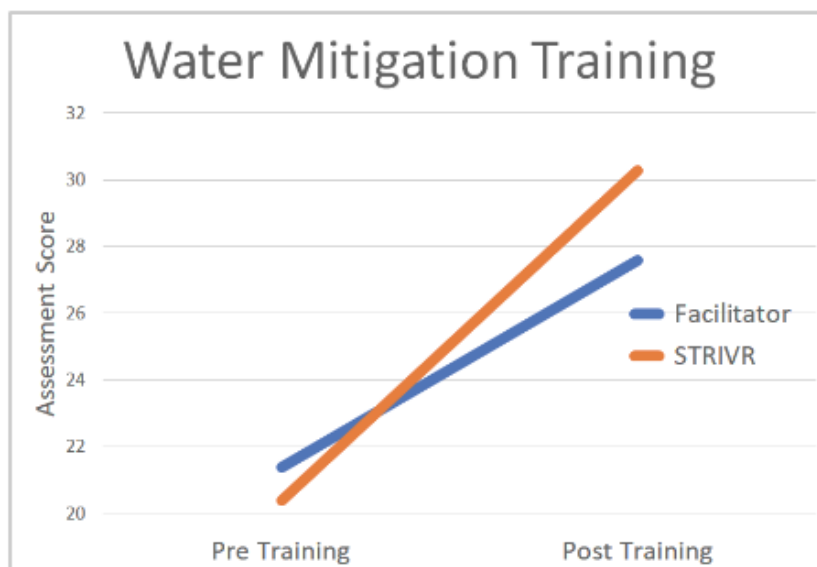


Gambar 13.3. Pertanyaan penanda digunakan untuk membuat peserta didik tetap terlibat dan berinteraksi selama Pelajaran



Gambar 13.4. Pertanyaan pilihan ganda tentang kategori kerusakan akibat air

STRIVR menguji setiap kelompok peserta pelatihan mengenai pengetahuan mereka sebelum dan sesudah setiap pelatihan, untuk melihat seberapa besar peningkatan yang mereka peroleh. Mereka menemukan bahwa kedua kelompok mengalami peningkatan yang hampir sama dalam pelatihan Mitigasi Air dan pelatihan Peningkatan. Hal ini menunjukkan adanya perbandingan; pengalaman dalam VR hampir sama dengan diterbangkan ke rumah banjir yang sebenarnya tetapi dengan biaya yang jauh lebih murah. Kedua kelompok mengalami peningkatan, seperti yang diilustrasikan dalam Gambar 13-5, dan tidak ada perbedaan yang signifikan secara statistik antara hasil mereka.



Gambar 13.5. Pelatihan VR dan pelatihan rumah banjir di dunia nyata menghasilkan peningkatan yang sebanding (© STRIVR 2018)

Pengumpulan data juga merupakan manfaat dari pelatihan VR. STRIVR melacak data gerakan pengguna dengan mencatat gerakan kepala dan tangan mereka. Satu pertanyaan khususnya sangat sulit bagi peserta pelatihan, dengan lebih dari separuh menjawab 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 saksama. Karena ukuran sampelnya kecil, tidak ada kesimpulan pasti yang dapat diambil, tetapi seiring dengan peningkatan alat, wawasan 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. Penting untuk membagi konten sehingga pengguna dapat beristirahat. 20 menit adalah patokan yang baik untuk sesi pelatihan, jadi satu pelajaran individu seharusnya jauh di bawah jumlah waktu ini.

Tidak semua konten “cocok” untuk VR

Karena STRIVR ingin menyertakan semua konten dari pelatihan di tempat, STRIVR membuat kesalahan dengan menyertakan konten yang tidak cocok untuk VR. Misalnya, dalam satu bagian, peserta pelatihan 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 kelas: pembelajaran ini memerlukan alat dari luar, dan tidak memanfaatkan kekuatan media VR.



Gambar 13.6. Meminta peserta didik untuk melakukan perhitungan tanpa konteks bukanlah "cara yang tepat" untuk VR

13.2 MANFAAT PELATIHAN VR? R.I.D.E.

VR tidak cocok untuk setiap kasus penggunaan. STRIVR menggunakan akronim "RIDE" untuk menentukan tempat terbaik untuk menggunakan VR:

- *Rare* / Langka
- *Impossible* / Tidak mungkin
- *Danger* / Berbahaya
- *Expensive* / Mahal

Berikut adalah beberapa contoh situasi yang sesuai dengan kriteria ini:

Langka

Black Friday jarang terjadi, hanya terjadi sekali setiap tahun, namun merupakan momen keuangan yang penting bagi pengecer. Pergantian karyawan yang tinggi berarti tidak cukup banyak karyawan yang meneruskan pengetahuan dan pengalaman dari tahun ke tahun.

Tidak mungkin

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 yang tidak mematuhi prosedur keselamatan dengan benar berbahaya untuk dilakukan, tetapi sangat penting bagi karyawan untuk dapat mengenali dan memperbaiki kesalahan dengan cepat.

Mahal

Seperti yang kita lihat pada kasus penggunaan sebelumnya dalam bab ini, profesional asuransi dilatih di "rumah banjir," yang menggunakan rumah dan air sungguhan untuk menggambarkan kondisi banjir secara realistis. Meskipun pelatihan ini realistis, biayanya juga

mahal.

Saat ini, pelatihan VR berorientasi spasial paling baik untuk 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-tugas ini dapat dilakukan dengan lebih mudah melalui antarmuka komputer umum. Karena sifat pekerjaan modern dan pekerjaan kantor, hal ini menghilangkan sejumlah kemungkinan kasus penggunaan.

12.3 APA YANG MEMBUAT PELATIHAN VR BAIK?

Pelatihan VR yang baik haruslah sebagai berikut:

- ✓ Spasial
- ✓ Sederhana dan mudah diakses
- ✓ Singkat
- ✓ Berorientasi pada tujuan
- ✓ Dapat diskalakan

Mari kita lihat lebih dekat setiap karakteristik:

Spasial

Pelatihan VR harus bersifat spasial untuk memanfaatkan sifat 3D VR, yang menyebutkan lokasi di atas, di belakang, dan di bawah pengguna. Hal ini menekankan perwujudan pengguna dan membantu meningkatkan daya ingat.

Sederhana dan mudah diakses

Keuntungan besar VR adalah aksesibilitas, dan skema kontrol Anda harus mencerminkan hal itu. Gim video modern menggunakan metode kontrol yang rumit dan tidak intuitif yang bergantung pada pengalaman dan keakraban pemain dengan genre tersebut; Anda tahu ini jika Anda pernah melihat seseorang yang tidak terbiasa dengan gim tembak-menembak orang pertama mencoba bergerak, menembak, dan melihat sekeliling pada saat yang bersamaan. Sebagian besar perangkat keras VR mendukung pengontrol tangan tunjuk-dan-klik yang sederhana. Tunjuk-dan-klik sangat bagus karena secara konseptual mirip dengan menggunakan tetikus komputer atau penunjuk laser. Hindari membuat pengguna mempelajari berbagai tombol dan antarmuka. Anda mencoba mengajarkan keterampilan kehidupan nyata, dan semakin Anda menonjolkan sifat pengontrol, semakin sedikit pengalaman Anda akan sesuai dengan kenyataan.

Singkat

Sesi harus singkat. 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 terbagi dengan baik, pengguna akan merasa nyaman untuk langsung memulai dan melakukan pelatihan, karena mereka tahu bahwa mereka akan segera dapat memutuskan apakah akan melanjutkan atau kembali ke tugas lain di dunia nyata. Dengan hambatan yang rendah untuk masuk dan mengikuti pelatihan, pengguna akan lebih sering masuk.

Berorientasi pada tujuan

Karena waktu sesi harus singkat, dan waktu peserta didik sangat penting, VR paling baik digunakan untuk 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 kotak pasir mungkin akan lebih banyak digunakan.)

Dapat diskalakan

Keunggulan VR dibandingkan media pelatihan lainnya adalah kualitasnya yang tinggi dan dapat diskalakan. Ingatlah hal ini saat Anda membangun platform dan konten. Konten baru harus mudah dibuat atau konten yang dibuat harus dapat digunakan kembali oleh banyak pengguna.

13.4 VIDEO SFERIS

Video sferis adalah rekaman yang menangkap setiap arah pandang sekaligus, yang memungkinkan penonton untuk menoleh secara fisik untuk melihat berbagai aspek video. Efek keseluruhannya seolah-olah penonton hadir secara fisik 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 sepenuhnya realistis.

Namun, agar kerangka kerja pelatihan dapat diskalakan dan efisien, kontennya harus cepat dan mudah dibuat. Anda memerlukan 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 tampak seperti mesin permainan berfitur lengkap.

Ingat, keuntungan utama VR untuk pelatihan adalah skalabilitas. Membuat pelatihan untuk banyak orang harus murah dan cepat. Semakin padat karya dalam membuat pelatihan, semakin jarang Anda dapat memperbaruinya, dan semakin sedikit konten yang dapat Anda hasilkan. Meskipun permainan video telah ada selama beberapa dekade, teknologi terkini untuk pelatihan perusahaan masih berupa video 2D, karena video mudah dibuat, dirawat, diperbarui, dan diganti.

Video sferis mudah direkam seperti video 2D konvensional, sekaligus menawarkan sejumlah manfaat terkait interaktivitas.

Manfaat Video Sferis

Berikut ini beberapa manfaat penggunaan video sferis:

- Dapat diskalakan
- Mudah menghasilkan konten
- Murah

- Lebih interaktif daripada video 2D

Saat membuat pelatihan VR, Anda perlu menciptakan kembali lingkungan pelatihan. Dengan video sferis, Anda hanya perlu merekam lingkungan tersebut di tempat. Ini menjamin bahwa hasil rekaman Anda benar-benar realistis dan sama persis dengan lingkungan sebenarnya.

Jika manusia merupakan bagian penting dari pelatihan, instruktur dan karyawan sungguhan dapat membantu. Orang-orang ini akan memiliki seragam yang tepat, mengetahui prosedur yang diajarkan, dan memiliki pengalaman dalam mendemonstrasikannya. (Peringatan: seperti yang akan kita bahas nanti, sebaiknya Anda selalu menyewa aktor sungguhan saat merekam peran yang membutuhkan penggambaran emosi atau ketenangan di depan kamera.) Dengan tingkat teknologi dan grafik komputer saat ini, saat ini belum ada grafik yang mendukung VR yang dapat mendekati tingkat realisme video.

Tantangan Video Sferis

Namun, ada juga tantangan dalam menggunakan video sferis untuk membuat konten. Beberapa situasi sulit dipentaskan, bahkan sekali untuk kamera. Untuk situasi seperti ini, kita dapat menggunakan penyuntingan video untuk menciptakan efek yang diperlukan, tetapi ini dapat menghasilkan hasil yang tidak realistis.

Tantangan lainnya adalah bahwa karyawan dan instruktur di lokasi tersebut mungkin bukan aktor alami yang dapat tampil baik di film. Jika pelatihan tersebut untuk sesuatu seperti perampokan toko, aktor sungguhan mungkin diperlukan, yang akan menimbulkan pengeluaran tambahan. Sangat penting untuk mendapatkan konten yang tepat sejak awal karena kembali ke lokasi untuk merekam ulang atau membersihkan video secara digital akan menghabiskan banyak biaya dan waktu. Namun, tantangan terbesar video sferis adalah interaktivitas dalam pengalaman tersebut.

Interaksi dengan Video Bulat

Saat mempertimbangkan apa yang dapat dicapai dengan video bulat, salah satu tempat pertama untuk mencari inspirasi adalah video 2D. Bentuk interaksi apa yang mungkin dilakukan dengan video 2D, dan apa keuntungan yang dibawa video bulat dibandingkan dengan video bulat?

Reaksi

Seperti halnya media apa pun, sebagian interaksi pengguna terjadi di dalam kepala pengguna itu sendiri, saat mereka menonton dan mendengarkan informasi yang disampaikan, lalu memprosesnya. Bahkan tanpa meminta interaksi langsung, pengguna berinteraksi dengan media dengan harapan, pikiran, dan pertanyaan internal mereka. Dengan cara ini, video 2D dan video sferis memiliki persyaratan yang sama, yaitu kontennya harus menarik, diproduksi dengan baik, dan relevan. Saat memproduksi video sferis untuk pelatihan, latar belakang dalam teknik yang digunakan untuk membuat video 2D menghibur sangatlah penting.

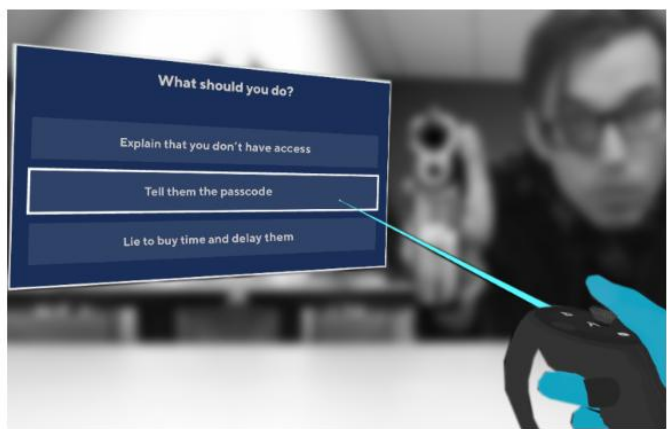
Tatapan

Tatapan adalah bentuk interaksi yang jauh lebih kaya dalam video sferis daripada dalam video 2D. Dalam video 2D, penonton menonton layar yang dibatasi. Pergerakan kamera dan transisi bidikan merupakan pilihan yang disengaja oleh editor untuk memfokuskan penonton pada tempat dan waktu tertentu. Namun, dalam video sferis, pengeditan harus dibatasi

semaksimal mungkin untuk menghindari kebingungan pengguna. Ini berarti bahwa tuntutan yang lebih besar diberikan kepada penonton; penonton adalah kameranya, dan mereka harus mengambil bagian aktif dalam melihat-lihat dan menyerap informasi. Dalam pengalaman "Pelatihan Perampokan Toko" STRIVR, misalnya, peserta didik pertama kali didekati oleh perampok dari belakang dan harus menoleh untuk melihat mereka. Dengan cara ini, tatapan dapat menjadi alat interaktif yang ampuh dalam video sferis.

Pertanyaan pilihan ganda

Dalam lingkungan pembelajaran elektronik kontemporer, video sering kali diawali dengan pertanyaan pilihan ganda. Dalam video sferis, pertanyaan pilihan ganda dapat disajikan dari dalam pengalaman, bukan di luarnya, yang menambahkan konteks pada keputusan pengguna. Misalnya, pada Gambar 13.7, salah satu perampok mengarahkan senjata ke arah Anda dan meminta Anda memberinya akses ke brankas. Melihat waktu berhenti 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 sferis daripada di layar komputer. Video sferis juga memberi Anda kesempatan untuk menggunakan lokasi atau objek dalam video sebagai "jawaban" untuk sebuah pertanyaan. Pada Gambar 13.8, seorang quarterback sepak bola diminta untuk memilih lokasi "runfit" setelah mengamati beberapa detik pertama dari suatu down.



Gambar 13.7. Soal pilihan ganda telah menambahkan konteks dalam lingkungan VR tempat



Gambar 13.8. Seorang pemain sepak bola harus mengidentifikasi "kecocokan larinya" dengan memilih lokasi yang benar dalam video

Titik-titik Minat

Video sferis memiliki alat interaksi yang kuat yang jarang digunakan dengan video 2D: menunjuk dan memilih lokasi dalam video. Ini mungkin bentuk interaksi terbaik dengan video sferis karena memberikan kesempatan kepada pelajar untuk berinteraksi langsung dengan media. Titik-titik Minat (PoI) dapat digunakan sebagai informasi, untuk menarik perhatian pada objek-objek utama dalam video, seperti pada Gambar 13-9.



Gambar 13.9. Titik-titik Minat dapat digunakan sebagai informasi untuk menyorot area-area minat utama dalam video



Gambar 13.10. "Perburuan adegan" adalah latihan yang melibatkan pencarian kesalahan atau item tersembunyi lainnya di lingkungan sekitar

Pol juga dapat digunakan dalam konteks "permainan objek tersembunyi," di mana pengguna hanya diperlihatkan video dan diminta untuk mengidentifikasi kelas item atau kesalahan. Tidak ada yang terlihat pada video selain perintah, dan pengguna harus mengeklik di berbagai tempat, mencari item atau lokasi yang sesuai dengan target mereka. Ini adalah pelatihan yang hebat karena peserta didik dipaksa untuk melihat dengan saksama lingkungan sekitar 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. Yang digambarkan dalam Gambar 13.10 adalah skenario di mana pengguna telah menemukan lima Pol dan memiliki waktu sekitar satu menit tersisa untuk mengidentifikasi 16 sisanya.

Pilih petualangan Anda sendiri

Terakhir, ada cara untuk "mencurangi" dan berinteraksi lebih langsung dengan video. Gim arkade laserdisc tahun 1983 *Dragon's Lair* adalah salah satu contoh paling awal. Dalam gim ini, pemain memasukkan perintah sederhana sebagai respons terhadap kartun animasi. Jika mereka memasukkan arah yang benar pada waktu yang tepat, video akan terus berlanjut. Jika gagal, karakter akan ditampilkan sekarat. Meskipun video tidak dapat berinteraksi langsung dengan pengguna seperti halnya pengalaman grafis komputer interaktif, ada kemungkinan untuk merekam beberapa video yang menggambarkan hasil yang berbeda, yang dapat menunjukkan kepada peserta pelatihan serangkaian tindakan salah yang dapat mereka lakukan.

Dalam pengalaman "Pelatihan Kabel Putus" STRIVR, misalnya, jika peserta didik gagal memberi tahu penelepon tentang bahaya kabel yang putus bagi hewan peliharaan, peserta didik akan mendengar anjing penelepon tersengat listrik. Untuk menghindari ledakan kombinatorial, sebaiknya ikuti pendekatan "untaian mutiara" untuk merancang konten yang bercabang; hanya izinkan peserta didik untuk melakukan penyimpangan terbatas dari skenario yang Anda ajarkan sebelum membimbing mereka kembali ke jalur yang benar.

Kasus Penggunaan: Pelatihan di Lantai Pabrik

Lantai pabrik sangat besar, bising, kacau, dan penuh dengan bahaya tersembunyi. Dalam satu studi, kesalahan keselamatan di 10 pabrik merugikan produsen makanan besar hingga lima digit pada bulan April 2015. Lima digit. Bukan uang, melainkan jari.

Benang merah untuk pelatihan pabrik adalah identifikasi kesalahan. Dengan video lantai pabrik yang berbentuk bulat, dapatkan karyawan mengidentifikasi pelanggaran keselamatan di sekitar mereka?

Lantai pabrik memberikan contoh yang jelas tentang manfaat video berbentuk bulat dibandingkan video 2D. Mudah untuk melihat slide PowerPoint dengan gambar kendaraan pemuatan di atasnya, dan memahami bahwa kendaraan pemuatan itu berbahaya. Namun, kendaraan pemuatan yang menabrak Anda bukanlah kendaraan yang Anda lihat datang. Kendaraan yang menabrak Anda adalah kendaraan yang tidak Anda lihat, kendaraan yang datang dari belakang. Video sferis, seperti realitas, hadir dari semua sudut.

Untuk memenuhi kebutuhan klien, STRIVR membuat fitur yang disebut perburuan

adegan, di mana peserta pelatihan harus mengidentifikasi titik-titik tersembunyi yang menarik dalam video. Di sekeliling adegan terdapat "hitbox" tersembunyi, yang hanya dapat ditemukan oleh peserta pelatihan dalam waktu terbatas, seperti yang ditunjukkan pada Gambar 13-11.



Gambar 13.11. Dalam "perburuan adegan", pembelajar harus menemukan kesalahan dalam batas waktu dan memilih area yang mengandung kesalahan, agar kesalahan tersebut terungkap.

Jika pengguna tidak menemukan semua hotspot tepat waktu, mereka akan diberi tahu tentang apa yang terlewat, seperti yang ditunjukkan pada Gambar 13.12.



Gambar 13.12. Setelah waktu habis, kesalahan yang tidak ditemukan peserta didik akan terungkap dan harus diinteraksikan

Salah satu hal hebat tentang pelatihan semacam ini adalah kemudahan dalam merekam dan menyusun pelatihan. Pabrik biasanya memiliki catatan kecelakaan masa lalu, karyawan veteran yang berpengalaman luas, dan manual yang mengkategorikan potensi kesalahan. STRIVR menemukan bahwa mudah untuk mengidentifikasi kesalahan nyata di lantai pabrik atau membuatnya dengan aman dengan bantuan instruktur.

Peran Narasi

Anda akan kesulitan membuat peserta didik berlatih jika pengalaman pelatihannya membosankan. Salah satu alasan utama STRIVR adalah karena keadaan terkini untuk pelatihan...yah, payah. Dan alasan utamanya adalah karena membosankan. Gamifikasi telah

menjadi kata kunci dalam industri selama bertahun-tahun, tetapi fokusnya adalah pada poin, bonus harian, dan lonjakan dopamin. Mengapa orang bermain gim? Karena gim menghibur. Mereka menyenangkan, apik, terpoles, dan, biasanya, memberi Anda alasan untuk peduli.

Saat Anda bermain gim, Anda tidak hanya belajar menerbangkan pesawat tempur antariksa untuk bersenang-senang. Anda belajar menerbangkan pesawat tempur antariksa agar dapat mengalahkan kaisar jahat yang membunuh ayah Anda. Khususnya bagi pelajar dewasa, memberi sasaran dan konteks pada pembelajaran mereka akan sangat membantu untuk membuatnya lebih mendalam dan menarik. Memberi sasaran pada tugas akan menambah dimensi ekstra dan membuka sedikit tambahan dari otak pelajar.

Jika Anda ingin membangun pengalaman pelatihan VR yang benar-benar akan diselesaikan oleh peserta pelatihan, Anda perlu memiliki narasi. Anda memerlukan pembelajaran yang berorientasi pada tugas. Mungkin sulit untuk mewujudkan pengguna dalam video bulat. Kamera tidak memiliki kehadiran fisik, yang dapat membuat pengguna merasa seperti hantu. Daripada membiarkan pengguna fokus pada kekurangan tubuh mereka, Anda harus merekam video Anda dengan karakter yang berbicara dengan pengguna, berinteraksi dengan mereka, dan membimbing mereka. Manusia adalah makhluk sosial, dan otak kita bekerja untuk menganalisis wajah dan membaca isyarat sosial. Melatih pengguna dengan meminta mereka berinteraksi dengan instruktur manusia merupakan peningkatan yang sangat besar dibandingkan dengan menggunakan suara tanpa tubuh. Bimbingan ahli satu lawan satu merupakan skenario pelatihan yang ideal, dan merekam instruktur merupakan hal yang paling mendekati hal tersebut dalam konteks video sferis.

Ini berarti bahwa merekam video sferis untuk pelatihan jauh lebih mirip dengan merekam film daripada yang Anda kira. Anda perlu memiliki naskah yang menarik dan memikat, dengan karakter dan alur cerita. Jika Anda ingin membuat pelatihan VR sebaik mungkin, itu juga berarti Anda perlu menyewa aktor.

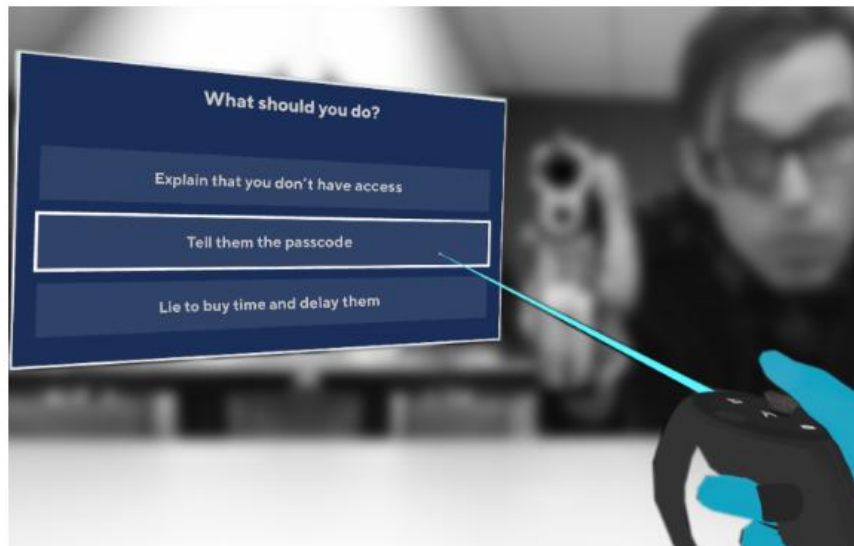
Kasus Penggunaan: Pelatihan Perampokan Toko

Perampokan toko merupakan contoh skenario yang hampir mustahil untuk dilatih secara normal. Tanpa cara apa pun untuk memprediksi perampokan, skenario pelatihan yang paling mendekati adalah lokakarya atau permainan peran yang sangat rumit.

Dalam “Pelatihan Perampokan Toko,” STRIVR membangun pengalaman yang digerakkan oleh narasi untuk menguji peserta pelatihan tentang apa yang telah mereka pelajari tentang respons yang tepat terhadap perampokan toko. Alih-alih membangun pelatihan VR ini untuk mengajarkan semua keterampilan yang diperlukan, Pelatihan Perampokan Toko dibangun sebagai ujian pengantar dan ujian akhir untuk melengkapi kelas atau pelatihan daring.⁸ Hasilnya, perancang instruksional STRIVR dapat lebih fokus pada hal-hal yang berjalan baik dalam VR.

Pelatihan Perampokan Toko menempatkan peserta didik pada posisi seorang manajer toko yang membuka toko untuk bekerja seharian dengan rekan kerjanya. Saat peserta didik memperhatikan pintu, perampok datang dari belakang, mengejutkan rekan kerja peserta didik dan memaksa peserta didik untuk berbalik untuk melihat apa yang terjadi. Peserta didik kemudian dipaksa untuk menghadapi situasi tersebut dan berinteraksi dengan menjawab

pertanyaan pilihan ganda, yang temanya adalah bekerja sama dengan dan bukan memusuhi perampok. Toko tersebut memiliki asuransi dan kontinjensi, dan peran manajer toko adalah untuk mencegah terjadinya bahaya apa pun pada karyawan mereka atau siapa pun di dalam toko. Dalam pengalaman ini, jika pelajar memilih jawaban yang salah, perintah suara akan memberi tahu mereka tentang tindakan yang benar dan alasannya. Meskipun STRIVR telah menggunakan pertanyaan pilihan ganda dalam banyak pengalaman sebelumnya, ini adalah salah satu kali pertama pertanyaan tersebut digunakan dengan cara yang begitu mendalam. Ketika perampok menanyai Anda, waktu melambat, layar menjadi kabur dan berubah menjadi hitam putih, dan suara detak jantung memberikan perasaan yang intens. Gambar 13-13 menunjukkan efek waktu terhenti, yang mengikat pertanyaan pilihan ganda dalam konteks yang mendalam.



Gambar 13.13. Saat perampok menanyai Anda, waktu melambat, lingkungan sekitar kabur, dan suara detak jantung menambah rasa intensitas

Pelatihan Perampokan Toko adalah contoh bagus 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 tersebut, peserta didik tidak akan siap untuk bereaksi, dan pengalaman tersebut akan menjadi komedi daripada menakutkan.

Nilai dari pelatihan ini adalah kemampuannya untuk mempersiapkan peserta didik secara mental. Pelatihan memaksa peserta didik untuk memvisualisasikan cara bereaksi dan apa yang harus dilakukan. Jika situasi tersebut terjadi dalam kenyataan, peserta didik dapat kembali ke model perilaku yang telah mereka latih secara virtual.

Saat menyusun pelatihan ini, desainer STRIVR memperhatikan betapa kuatnya pandangan peserta didik tertarik pada karakter manusia, dan mereka mulai menggunakan instruktur manusia yang berwujud untuk pelatihan mendatang, daripada suara yang tidak berwujud. Memiliki aktor yang berinteraksi dengan peserta didik menawarkan banyak

keuntungan. Para peserta didik 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 sesuai dengan Anda membantu Anda merasa lebih hadir dalam adegan tersebut. Manfaat lainnya adalah dalam arah pandangan. Daripada meletakkan anak panah atau tanda di sekitar adegan, terasa sangat alami bagi karakter manusia untuk berjalan atau menunjuk untuk mengarahkan pandangan peserta didik.

Satu hal terakhir yang perlu disebutkan: ketika membangun pengalaman yang berpotensi menjadi traumatis atau memicu, penting untuk memberi tahu peserta didik bahwa mereka dapat membatalkannya kapan saja. Untuk pelatihan ini, kami memastikan bahwa peserta didik menyadari bahwa mereka dapat menjeda, menekan tombol Beranda Oculus, atau melepaskan headset secara fisik kapan saja jika mereka merasa tidak nyaman.

13.5 MASA DEPAN PELATIHAN XR

Dalam bab ini, fokusnya adalah pada VR, dan khususnya video bulat, sebagai media pelatihan. Video bulat mencapai titik yang tepat karena mudah ditangkap dan memberikan hasil yang realistis. Akan tetapi, kami juga telah membahas bahwa video sferis dapat dibatasi dalam interaktivitas dan fidelitasnya.

Bagaimana dengan teknologi lain? Ke mana arah pelatihan XR di masa mendatang? Di sisa bab ini, kami akan membahas berbagai peningkatan dan alternatif untuk video sferis.

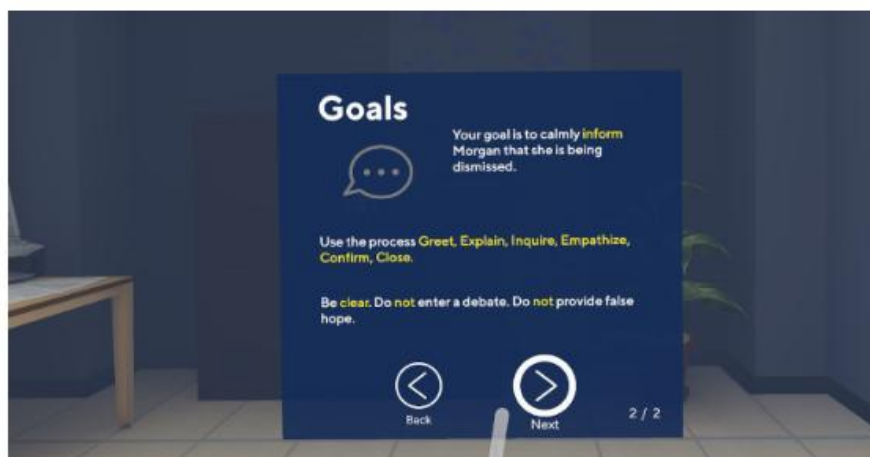
Grafik Komputer

Pilihan utama lainnya untuk menggambarkan lingkungan pelatihan adalah grafis komputer (CG). CG memberikan manfaat besar dalam hal interaktivitas. Model 3D dapat bergerak secara dinamis dalam suatu lingkungan tanpa perlu difilmkan dalam berbagai posisi yang tak terbatas.

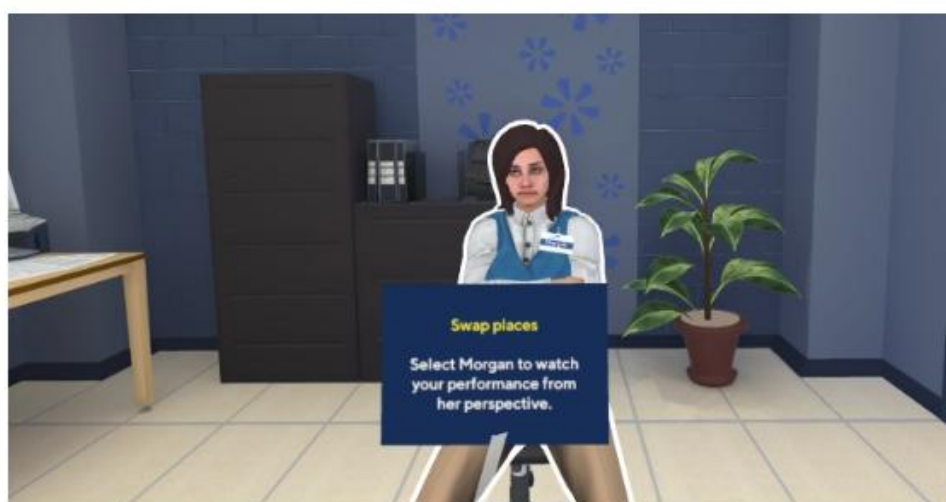
Namun, grafis 3D perlu dimodelkan, dianimasikan, dan diberi pencahayaan. Membangun aset 3D dan skenario interaktif memakan waktu. Rockstar Studios membutuhkan waktu lebih dari tiga tahun untuk mengembangkan Red Dead Redemption 2; jangka waktu seperti ini biasa terjadi pada perusahaan game. Karena pertimbangan ini, saat membangun pengalaman pelatihan CG, pelatihan haruslah penting dan berkelanjutan.

Kasus Penggunaan: Pelatihan Keterampilan Lunak

Kasus penggunaan ini menyoroti kerangka kerja pelatihan VR yang dibangun dengan CG. Soft Skills menggunakan manusia virtual untuk mensimulasikan percakapan yang sulit, seperti memberikan penilaian kinerja negatif kepada karyawan. Peserta pelatihan pertama-tama memilih avatar untuk mewakili diri mereka sendiri. Mereka diperkenalkan dengan skenario tersebut dan kemudian diminta untuk mengatakan apa. Gambar 13.14 menunjukkan contoh perintah. Manusia virtual bereaksi, dan percakapan berlanjut. Meskipun pengalaman tersebut sudah ditentukan sebelumnya, sebagian besar pengguna percaya bahwa manusia virtual tersebut bereaksi dan beradaptasi dengan apa yang mereka katakan.



Gambar 13.14. Peserta didik dipandu melalui garis besar tentang apa yang harus mereka sampaikan di setiap langkah (© STRIVR 2018)



Gambar 13.15. Setelah percakapan, peserta didik menonton pemutaran ulang kata-kata mereka sendiri dari sudut pandang lawan bicara mereka (© STRIVR 2018)

Dampak nyata dari pelatihan Soft Skills terjadi setelah percakapan. Selama Anda mengikuti perintah dan berbicara dengan karyawan virtual, suara dan gerakan Anda direkam. Di akhir pengalaman, semuanya diputar ulang dengan peran yang dibalik. Pada Gambar 13-15, peserta didik akan bertukar tempat dengan Morgan, karyawan yang bermasalah. Duduk berhadapan dengan diri sendiri dan menyaksikan kata-kata Anda keluar dari mulut orang lain adalah pengalaman yang luar biasa. Ketika saya mengikuti pelatihan ini pertama kali, saya mengatakan banyak hal yang kemudian saya sadari terdengar kasar. Misalnya, di awal percakapan, saya mengatakan bahwa rapat itu "tidak masalah." Mendengarnya kembali di seberang meja, saya bereaksi secara mendalam terhadap betapa tidak tulusnya hal itu terdengar.

Soft Skills adalah contoh bagus dari aplikasi yang dapat digunakan kembali yang dapat memperoleh manfaat dari investasi dalam CG. Pelatihan Soft Skills adalah kerangka kerja yang dapat digunakan kembali untuk keterampilan penting: komunikasi. Mayoritas 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 berjalannya waktu.

CG dalam konteks ini memiliki banyak manfaat dibandingkan video sferis. Misalnya, satu suara dapat digunakan dengan banyak avatar yang disesuaikan untuk mengendalikan penampilan manusia virtual. Animasi dapat digunakan kembali di seluruh avatar. Lingkungan dan avatar dapat dicampur dan dicocokkan. Skenario percabangan juga lebih mudah dibangun dengan CG, karena manusia virtual dapat bereaksi secara dinamis terhadap pengguna seperti karakter gim video.

Masa Depan: Fotogrametri

Fotogrametri adalah teknologi yang menarik untuk menangkap lingkungan, objek, atau orang di dunia nyata dan membuatnya menjadi model 3D. Ribuan foto diambil dari subjek dari setiap sudut, lalu gambar-gambar ini digabungkan menjadi model 3D.

Namun, sulit untuk membangun model 3D yang sempurna hanya dari teknik fotogrametri. Banyak pekerjaan pembersihan yang harus dilakukan untuk menutup celah-celah kecil pada model dan memperbaiki cacat lainnya.⁹ Di STRIVR, kami menggunakan teknik ini untuk membangun dinding basah toko kelontong virtual, dan seniman 3D kami Tyrone harus menghabiskan lebih banyak waktu daripada yang kami perkirakan untuk membersihkan lingkungan yang ditangkap dan elemen yang dapat berinteraksi. Gambar 13.16 menunjukkan sepotong brokoli yang sangat merepotkan.

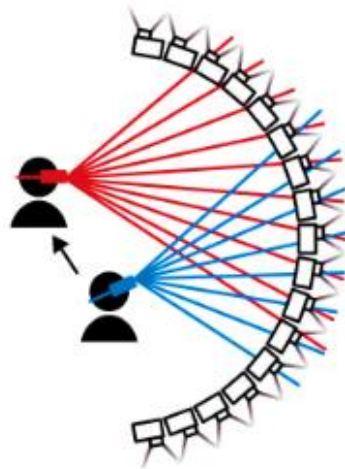
Fotogrametri adalah teknologi yang perlu dipertimbangkan, tetapi perlu diingat tantangannya. Ini adalah teknologi yang perlu diperhatikan seiring dengan peningkatan teknik.



Gambar 13.16. Menggunakan fotogrametri untuk membuat model objek di dunia nyata masih membutuhkan banyak tenaga kerja (© STRIVR 2018)

Masa Depan: Medan Cahaya

Lapangan cahaya adalah teknologi baru yang menarik yang menggunakan cincin kamera yang berputar untuk menangkap semua cahaya yang memasuki area bulat. Ini menghasilkan area yang ditangkap di mana pemirsa memiliki paralaks penuh. Sementara dengan video mono sferis, pemirsa dibatasi pada satu perspektif, medan cahaya memungkinkan pengguna untuk menggerakkan kepala mereka untuk melihat berbagai sudut pemandangan,5 seperti yang ditunjukkan pada Gambar 13-17.



Gambar 13.17. Medan cahaya memungkinkan perubahan posisi kepala untuk memberikan perspektif yang berbeda

Paralaks penuh sangat meningkatkan rasa kehadiran pemirsa, kunci untuk menciptakan pengalaman pelatihan yang lebih berkualitas. Jika Anda belum sempat mencobanya, cobalah demonstrasi Google “Welcome To Light Fields”. Hasilnya sangat mencolok.

Hingga tahun 2018, masih terlalu banyak keterbatasan 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 tersebut. Kecepatan rotasi perangkat kamera juga berarti bahwa hanya gambar diam yang dapat ditangkap; sesuai untuk beberapa pengalaman pelatihan yang difokuskan pada lingkungan statis tetapi kurang berguna untuk apa pun yang melibatkan aktor atau gerakan dinamis. Namun, medan cahaya adalah teknologi penangkapan yang patut diperhatikan.

Masa Depan: Pelatihan 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 yang ada; mengambil realitas dan kemudian melengkapinya.

VR bersifat transportasi; AR bersifat 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 bermanfaat, AR harus mampu melakukan sesuatu yang tidak dapat dilakukan otak manusia, dengan akurasi yang cukup agar aman dan bermanfaat. Kita semakin dekat. Kita memiliki perangkat lunak yang dapat menerjemahkan teks dan menggantinya dengan cepat, dan pengenalan gambar yang dapat menukar wajah. Namun, akurasi 99% tidak cukup baik untuk banyak kasus penggunaan; kita perlu mencapai 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 contoh rumah banjir, lalu bayangkan lingkungan pelatihan tempat AR mengajar dan melatih Anda di lingkungan nyata. Ini akan memberikan manfaat untuk menjelajahi ruang pelatihan fisik nyata dengan kemampuan 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 yang patut dicermati adalah VOID,¹² yang menggunakan campuran VR yang dapat dikenakan, pelacakan lokasi, dan ruang fisik untuk menciptakan pengalaman hiburan yang mendalam.

Masa Depan: Pengenalan Suara

Antarmuka suara akan menjadi tambahan yang sangat penting bagi perangkat pelatihan saat kita dapat melewati ambang batas akurasi akhir tersebut. Aspek penting dari pelatihan yang kita lakukan adalah interpersonal: pelatihan tentang layanan pelanggan atau penyelesaian konflik interpersonal. Pengenalan suara dapat menyediakan media dan skema kontrol yang sempurna segera setelah terintegrasi ke dalam perangkat HMD, tetapi pertamanya pengenalan suara harus cukup andal untuk bekerja dalam semua kondisi, termasuk ruang belakang ritel yang bising.

Masa Depan: Skenario Pelatihan Ideal

Bayangkan sejenak masa depan. Dunia tanpa batas, dengan lensa kontak AR dan AI yang kuat. Seperti apa pelatihan itu?

Mary Poppins. Jangan tertawa!

Mary Poppins muncul dari langit dengan payung ajaib dan mengubah segerombolan anak-anak yang mengerikan menjadi warga negara teladan. Anak-anak itu bahkan tidak menyadari bahwa mereka sedang dilatih, karena pelatihan itu menyenangkan. Mary Poppins mencapai keseimbangan sempurna antara membimbing, menantang, dan membina anak didiknya.

Inilah yang dibutuhkan untuk mendapatkan pengalaman pelatihan yang sempurna. AI yang kuat yang dapat memahami siapa peserta didik dan melatih mereka berdasarkan kebutuhan mereka. Skenario semacam ini sangat mustahil. Pada tahun 2018, lensa kontak AR dan AI yang kuat belum ada. Namun, mempertimbangkan dunia yang ideal dan memikirkan ke mana arah pelatihan di masa mendatang dapat menjadi alat yang hebat untuk mencari tahu cara mencapainya, terutama saat mempertimbangkan teknologi apa yang akan diinvestasikan dan bagaimana mempersiapkannya.

DAFTAR PUSTAKA

- Azuma, R. T. (1997). A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4), 355-385. <https://doi.org/10.1162/pres.1997.6.4.355>
- Billinghurst, M., & Kato, H. (2002). Collaborative augmented reality. *Proceedings of the 4th IEEE International Symposium on Wearable Computers*, 199-200. <https://doi.org/10.1109/ISWC.2002.1167231>
- Bimber, O., & Raskar, R. (2005). *Spatial augmented reality: Merging real and virtual worlds*. Springer. <https://doi.org/10.1007/3-540-27797-7>
- Boulanger, A., & Marier, M. (2018). Augmented reality and virtual reality in art and culture. *International Journal of Arts and Technology*, 11(3), 210-227. <https://doi.org/10.1504/IJART.2018.090121>
- Burdea, G., & Coiffet, P. (2003). *Virtual reality technology*. John Wiley & Sons.
- Cagiltay, K., & Yaman, M. (2017). Augmented and virtual reality in education: A review of recent research. *Journal of Educational Technology*, 44(6), 142-149. <https://doi.org/10.1109/JET.2017.8228426>
- Chen, M., Gaither, K., John, N. W., & McCann, B. (2017). Cost benefit analysis of virtual environments. *EuroVis*. Retrieved from <https://arxiv.org/pdf/1802.09012.pdf>
- Clancy, T. (2015, September 7). Oculus Rift internet visualization [Video]. YouTube. <https://youtu.be/GpFVWFUHLcl>
- Cummings, J. J., & Bailenson, J. N. (2016). Immersive virtual reality and the impact of social presence on self-reported empathy. *Computers in Human Behavior*, 58, 235-245. <https://doi.org/10.1016/j.chb.2015.12.038>
- Dede, C. (2009). Immersive interfaces for engagement and learning. *Science*, 323(5910), 66-69. <https://doi.org/10.1126/science.1167311>
- Fellner, D. W., & Schlömer, T. (2018). Augmented and virtual reality in healthcare applications: The potential and the challenges. *Journal of Healthcare Engineering*, 2018, 1-9. <https://doi.org/10.1155/2018/9518930>
- Freeman, D., & Reavley, N. (2018). Virtual reality in the treatment of psychiatric disorders. *Current Psychiatry Reports*, 20(6), 43. <https://doi.org/10.1007/s11920-018-0914-2>
- Gabbard, J. L., Hix, D., & Swan, J. E. (1999). Usability engineering for virtual environments: Designing virtual experience. *International Journal of Human-Computer Interaction*, 11(1), 157-188. https://doi.org/10.1207/s15327590ijhc1101_5
- Gonzales, L. L. (n.d.). 10KS [Video]. *STEAM*. Retrieved from <http://bit.ly/2XCi2LW>

- Harrell, D. F., & Buckner, S. (2018). Augmented reality and virtual reality in education. *Emerging Technologies for Education*, 22(1), 15-29. https://doi.org/10.1007/978-3-030-15597-1_2
- Heilig, M. F. (1965). El cine del futuro. *Proceedings of the International Conference on Virtual Reality and Augmented Reality*, 1-9. <https://doi.org/10.1016/j.jvcir.2019.101634>
- Hernández, J., & Moreno, A. (2017). Virtual reality and augmented reality for education. *International Journal of Emerging Technologies in Learning (IJET)*, 12(11), 44-50. <https://doi.org/10.3991/ijet.v12i11.7490>
- Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning*. Retrieved from <http://bit.ly/2SHKnws>
- Hohman, F., Kahng, M., Pienta, R., & Chau, D. H. (2018). Visual analytics in deep learning: An interrogative survey for the next frontiers. *Institute of Electrical and Electronics Engineers (IEEE). Transactions on Visualization and Computer Graphics*.
- Holscher, C., & Kiefer, H. (2019). Interaction design for virtual and augmented reality. *Virtual Reality and Augmented Reality in Education*, 1(1), 34-43. <https://doi.org/10.1016/j.jvra.2019.01.004>
- Itti, L., & Baldi, P. (2009). Bayesian surprise attracts human attention. *Vision Research*, 49(10), 1295-1306. <https://doi.org/10.1016/j.visres.2008.09.007>
- Johnson, I. (2016, October 13). How to use t-SNE effectively. *Distill.pub*. <https://distill.pub/2016/misread-tsne/>
- Jones, L. (2017). *Virtual reality and augmented reality in business and industry*. IGI Global. <https://doi.org/10.4018/978-1-5225-2057-0>
- Kato, H., & Billinghurst, M. (1999). The art of collaborative augmented reality. *Proceedings of the 2nd International Symposium on Virtual Reality*, 17-22. <https://doi.org/10.1109/VR.1999.756919>
- Kim, J., & Park, S. (2020). Augmented reality applications in tourism and hospitality industries. *Journal of Tourism & Hospitality*, 7(2), 1-8. <https://doi.org/10.1007/s11842-020-00798-1>
- Kourouthanassis, P., & Giaglis, G. M. (2015). Exploring the use of augmented reality in tourism. *Tourism Management Perspectives*, 13, 1-10. <https://doi.org/10.1016/j.tmp.2014.07.005>
- Latosińska, J., & Szmigiera, M. (2020). Virtual and augmented reality applications in medicine. *European Journal of Medical Research*, 25(1), 45. <https://doi.org/10.1186/s40001-020-00425-7>
- Lee, S., & Lee, D. (2016). Virtual reality and augmented reality applications in medical education and training. *International Journal of Engineering Education*, 32(3), 1059-1066.

- Maeda, J. (2006). *Law of simplicity: Simplicity, design, technology, business, life*. MIT Press.
- Mann, S. (2015). Wearable computing and augmented reality: An introduction. *Proceedings of the IEEE International Conference on Human-Computer Interaction*, 1-7. <https://doi.org/10.1109/HCI.2015.7600350>
- Milgram, P., & Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE Transactions on Information and Systems*, 77(12), 1321-1329. <https://doi.org/10.1109/50.248405>
- Mullen, P. (2018). Virtual and augmented reality applications in gaming. *Game Studies*, 18(2), 22-37. <https://doi.org/10.1007/s10941-018-0813-7>
- Murphy, R. (2016, November 2). Immersive data visualization: AR in the workplace. *Game Developer Conference*. Retrieved from <http://bit.ly/2NET0a8>
- Nah, F. F.-H., & Esche, S. A. (2017). Virtual reality, augmented reality, and mixed reality in education. *Handbook of Research on Mobile Learning in Contemporary Classrooms*, 23(1), 317-337. <https://doi.org/10.4018/978-1-5225-1040-3>
- Ng, A. (2017, July 13). Downtown LA. My July 4th weekend FourSquare check-ins visualized in AR. (ARKit + Unity + Mapbox + Swarm) [Tweet]. Twitter. Retrieved from <http://bit.ly/2SlmGnC>
- Papachristodoulou, P. (2014). Sonification of large datasets in a 3D immersive environment: A neuroscience case study. *The Seventh International Conference in Advances in Computer-Human Interactions (ACHI)*.
- Patel, M., & Patel, S. (2020). Virtual reality applications for enhancing learning experiences in education. *International Journal of Emerging Technologies*, 1(2), 56-63. <https://doi.org/10.1007/s20240-019-00927-w>
- Pimentel, K., & Teixeira, A. (2021). Virtual reality in education: A practical guide. *Journal of Education Technology*, 33(4), 210-222. <https://doi.org/10.1109/JET.2021.3052907>
- Polys, N. F., & McManus, R. (2016). Augmented reality and interactive visualizations for architectural design. *Computers in Industry*, 81, 101-110. <https://doi.org/10.1016/j.compind.2015.11.010>
- Ragan, E. D., & Bowman, D. A. (2019). Enhancing human performance in virtual and augmented reality: The role of cognitive load. *Virtual Reality*, 23(4), 429-441. <https://doi.org/10.1007/s10055-019-00378-6>
- Repetto, C., & Lizio, A. (2018). Applications of augmented reality for elderly people: A comprehensive review. *International Journal of Human-Computer Interaction*, 34(10), 907-918. <https://doi.org/10.1080/10447318.2018.1484449>
- Rizzo, A. S., & Koenig, S. T. (2017). Virtual reality and neuropsychological interventions: The potential for VR in the assessment and treatment of psychological disorders. *Psychology Research and Behavior Management*, 10, 17-23. <https://doi.org/10.2147/PRBM.S106678>

- Romero, D., & Soler, M. (2019). Exploring the applications of virtual and augmented reality for improving learning outcomes in education. *Journal of Educational Computing Research*, 57(8), 1843-1859. <https://doi.org/10.1177/0735633119874772>
- Santos, A., & Oliveira, F. (2020). Augmented reality in healthcare: An overview. *Journal of Health Informatics*, 15(2), 35-42. <https://doi.org/10.1177/0037549819894283>
- Schmalsteig, D. (2018, October 23). When visualization met augmented reality [Keynote]. *IEEE Vis Conference*, Berlin, Germany. <https://www.youtube.com/watch?v=qtar1Q2ZPYM>
- Shahriaree, R., & Hossain, M. (2018). Virtual and augmented reality in e-commerce: Current trends and future directions. *Computers in Human Behavior*, 87, 278-292. <https://doi.org/10.1016/j.chb.2018.06.016>
- Slater, M., & Sanchez-Vives, M. V. (2016). Enhancing our lives with immersive virtual reality. *Frontiers in Robotics and AI*, 3, 74. <https://doi.org/10.3389/frobt.2016.00074>
- Slater, M., & Wilbur, S. (1997). A framework for immersive virtual environments (FIVE): Speculations on the role of presence in virtual environments. *Presence: Teleoperators and Virtual Environments*, 6(6), 603-616. <https://doi.org/10.1162/pres.1997.6.6.603>
- Steuer, J. (1992). Defining virtual reality: Dimensions determining telepresence. *Journal of Communication*, 42(4), 73-93. <https://doi.org/10.1111/j.1460-2466.1992.tb00812.x>
- Tang, A., & Rawlins, P. (2017). The development of virtual and augmented reality in gaming. *Proceedings of the ACM Virtual Reality Conference*, 207-211. <https://doi.org/10.1145/3051851.3051857>
- Thomas, L., & Reiser, J. (2020). The role of virtual reality in teaching spatial understanding. *Journal of Educational Technology*, 34(6), 154-162. <https://doi.org/10.1080/10447318.2020.1827067>
- Tufte, E. (2001). *Visualizing display of quantitative information*. Graphics Press. Retrieved from <http://bit.ly/2TjKDap>
- Tufte, E. (2006). *Beautiful evidence*. Graphics Press.
- Tyndall, L. S., & O'Neill, H. (2017). Virtual and augmented reality in architectural design. *Journal of Virtual and Augmented Reality in Architecture*, 19(1), 8-17. <https://doi.org/10.1109/JVAR.2017.8010459>
- Ueda, H., & Takeda, H. (2019). The integration of virtual reality and augmented reality in education. *International Journal of Advanced Computer Science and Applications*, 10(11), 171-178. <https://doi.org/10.14569/IJACSA.2019.0101113>
- Vassileva, J., & Deters, R. (2018). *Virtual and augmented reality applications in industry*. Springer. <https://doi.org/10.1007/978-3-030-11948-4>
- Viegas, F., & Wattenberg, M. (2018). Visualization for machine learning. *Neural Information Processing Systems (NeurIPS)*. Retrieved from <http://bit.ly/2TncTYT>

- Waltemate, T., & Hesse, F. (2018). Augmented reality in education: A systematic review. *Educational Technology Research and Development*, 66(4), 747-770. <https://doi.org/10.1007/s11423-018-9570-1>
- Wang, Y., & Wang, S. (2019). The future of virtual and augmented reality in healthcare: Implications for medical education and patient care. *Medical Image Analysis*, 56, 123-130. <https://doi.org/10.1016/j.media.2019.02.003>
- Wiederhold, B. K., & Wiederhold, M. D. (2019). Virtual reality and telemedicine in mental health care. *Cyberpsychology, Behavior, and Social Networking*, 22(6), 404-411. <https://doi.org/10.1089/cyber.2018.0406>
- Wilhelmsen, J. S., & Dunbar, N. (2017). Augmented and virtual reality in military applications. *Journal of Defense Modeling and Simulation*, 14(3), 248-264. <https://doi.org/10.1177/1548512917693152>
- Wittenberg, M., & Braun, S. (2018). Virtual reality as a tool for cognitive rehabilitation: Current trends and future directions. *Journal of Neuroengineering and Rehabilitation*, 15(1), 3. <https://doi.org/10.1186/s12984-018-0393-4>
- Xie, S., & Li, H. (2020). The potential applications of virtual and augmented reality in product design. *Design Studies*, 44(2), 65-77. <https://doi.org/10.1016/j.destud.2019.10.006>
- Youngblut, C. (2018). Virtual reality and augmented reality in defense. *Virtual Reality*, 23(1), 121-130. <https://doi.org/10.1007/s10055-018-0337-2>
- Yuen, S., & Yaoyuneyong, G. (2013). Augmented reality and its application in education. *Journal of Educational Technology Development and Exchange (JETDE)*, 6(2), 99-112. <https://doi.org/10.18785/jetde.0602.06>
- Zaharias, P., & Pappas, I. O. (2020). Virtual reality applications in education: A review. *Journal of Educational Computing Research*, 58(3), 628-648. <https://doi.org/10.1177/0735633118780477>
- Zeng, X., & Lee, H. (2019). A review of augmented reality and virtual reality in education and healthcare. *International Journal of Human-Computer Interaction*, 35(2), 97-104. <https://doi.org/10.1080/10447318.2018.1492786>
- Zhang, F., & Chen, T. (2020). Mixed reality systems: Overview, applications, and challenges. *IEEE Transactions on Virtual Reality*, 26(4), 591-601. <https://doi.org/10.1109/TVCG.2020.2965643>
- Zhang, H., & Liu, Z. (2017). Exploring augmented reality technologies and applications. *Proceedings of the IEEE Virtual Reality Conference*, 233-239. <https://doi.org/10.1109/VR.2017.7892242>
- Zhang, R., & Zhang, Z. (2018). Design and implementation of augmented reality in retail and marketing. *Journal of Retailing and Consumer Services*, 45, 63-72. <https://doi.org/10.1016/j.jretconser.2018.08.005>

- Zhao, Q., & Xu, M. (2020). Applications of augmented and virtual reality in educational environments. *Journal of Educational Computing Research*, 58(2), 254-272. <https://doi.org/10.1177/0735633118755382>
- Zhi, J., & Huo, W. (2020). Augmented reality technology and its applications in industrial production. *Computers in Industry*, 116, 35-45. <https://doi.org/10.1016/j.compind.2019.08.003>
- Zhou, F., & Duh, H. B. (2019). Augmented and virtual reality technologies in engineering. *Journal of Engineering Education*, 108(1), 50-62. <https://doi.org/10.1002/jee.20245>
- Ziegler, D., & Koch, A. (2018). Mixed reality: Integrating virtual and augmented reality. *International Journal of Human-Computer Studies*, 118, 1-10. <https://doi.org/10.1016/j.ijhcs.2018.04.004>



Teori & Praktik AR dan VR (Augmented Reality dan Virtual Reality)

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



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