

NORTHWESTERN UNIVERSITY

Compositional Possibilities of New Interactive and Immersive
Digital Formats

A DISSERTATION

SUBMITTED TO THE BIENEN SCHOOL OF MUSIC IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS

for the degree

DOCTOR OF MUSICAL ARTS

Program of Composition

By
Daniel R. Dehaan

EVANSTON, IL
(June 2019)

Abstract

From 2008 to 2019, a range of new interactive and immersive digital formats that present new possibilities for musical and artistic expression have become available. In order to begin the work of uncovering what new compositional and experiential possibilities are now possible, this document will examine each format's core concepts and tools, cataloging the current state of related technologies. It also provides a survey of each format's representative works, including a discussion of my original and evolving work for virtual reality, *Infinite Void*. The ultimate goal of this dissertation is to serve as a point of departure for composers interested in working with and influencing the direction that musical and creative expression will take in these immersive and interactive digital environments.

Acknowledgments

This document would not have been possible without countless individuals to whom I owe more than just the acknowledgements of this page.

To my committee members, Chris Mercer, Hans Thomalla, and Stephan Moore, who made themselves available from all corners of the globe and encouraged me to keep going even when it seemed like no end was in sight.

To Donna Su, who kept me on track and moving forward throughout my entire time at Northwestern.

To my readers, Nick Heinzmann and Caleb Cuzner, without whom I don't think I would have ever been able to finish.

To my partner, Brianna Tonner, who has supported and loved me through it all—even the Italian operas.

And finally, to my family, who has always encouraged my love for music.

Table of Contents

Chapter 1: Introduction	31
Chapter 2: Sound Localization and Spatialization	34
2.1 Spatial Perception.....	34
2.1.1 Lateral Perception.....	35
Amplitude.....	35
Timing.....	35
<i>Onset Timing</i>	35
<i>Interaural Phase Relationships</i>	36
2.1.2 Front/Back and Elevation.....	37
Direction Selective Filters	37
Head-Related Transfer Functions	37
Head Motion.....	38
2.1.3 Distance.....	38
Loudness	38
Initial Time Delay.....	39
The Ratio of Direct Sound to Reverberation	39
Motion Parallax	39
High-Frequency Attenuation	39
2.2 Audio Delivery Systems.....	40
2.2.1 Personal Delivery Systems	40
Closed-Back Headphones.....	40
Open-Back Headphones.....	41
Earbuds	41
In-Ear Monitors	41
2.2.2 External Delivery Systems	41
Standard Speaker Configurations.....	43
<i>Stereo</i>	43
<i>Quadraphonic</i>	43
<i>5.1 Surround Sound</i>	43
<i>Octaphonic</i>	44

Further Channel Expansion	44
2.3 Distribution Formats	44
2.3.1 Compressed vs. Uncompressed	45
Lossless	45
Lossy	46
File Container Formats	46
2.3.2 Discrete vs. Matrixed	47
2.3.3 Ambisonics	47
<i>Zeroth-Order</i>	48
<i>First-Order (FOA)</i>	48
<i>Second-Order (SOA)</i>	48
<i>Third-Order (TOA)</i>	48
<i>Higher-Orders (HOA)</i>	49
Ambisonic Formats	49
<i>A-Format</i>	49
<i>B-Format</i>	49
<i>C-Format (Consumer Format)</i>	49
<i>D-Format</i>	49
<i>G-Format (Geoffrey's Format)</i>	50
<i>Additional Formats</i>	50
Pros and Cons of Ambisonics	51
2.3.4 Binaural Audio	52
2.4 Spatialization Techniques	52
2.4.1 External Delivery Systems	53
Phantom Image	53
Wave Field Synthesis	53
Size/Width	54
Head shadowing, IID, and ITD	54
Spatial Effect Ideas for use with External Delivery Systems	54
<i>Spatial Filter Sweeps</i>	54
<i>Spatial Tremolo</i>	54
<i>Microtonal Pitch Inflections</i>	54
<i>Complex Delay Networks</i>	55
2.4.2 Personal Delivery Systems	55
<i>Creating Interaural Intensity Differences</i>	56
<i>Creating Interaural Time Differences</i>	57
<i>Head shadowing</i>	57

<i>Front, Back, and Elevation</i>	57
<i>Distance</i>	58
2.5 Environment Modeling	58
2.5.1 Direct Sound	59
2.5.2 Indirect Sound	59
Early Reflections.....	60
Diffusion	60
<i>Convolution Reverb</i>	61
<i>Algorithmic Reverb</i>	61
Occlusion and Obstruction	61
2.6 Sound Design Guidelines for Spatialized Audio	63
2.6.1 Mono	63
2.6.2 Complex Timbres	63
2.6.3 Head-locked vs. Spatialized	63
2.6.4 Visual Attachments	64
2.6.5 Keep it simple	64
Chapter 3: Survey of New Digital Formats 2008-2019	65
3.1 Established Formats	67
3.1.1 Massively Multiplayer Online Games	68
Linden Lab: <i>Second Life</i>	69
3.1.2 iOS Instruments	72
Smule: <i>Ocarina</i>	73
Herrmutt Lobby: <i>Playground</i>	74
Moog: Model 15	79
3.1.3 Album Apps	80
Jay-Z: <i>Magna Carta</i> - 2013	81
Björk: <i>Biophilia</i> - 2011	82
Brian Eno & Peter Chilvers	88
<i>Bloom</i> – 2008.....	88
<i>Air and Trop</i> – 2009.....	91
<i>REFLECTION</i> – 2017	94
<i>Bloom: 10 Worlds</i> – 2018.....	95
Others Album Apps	97
3.1.4 360 Videos	99
Björk: “Stonemilker” - 2015	99
Reeps One: “Does Not Exist” – 2016.....	100

Conservation International Documentaries.....	101
3.2 Emerging Formats.....	102
3.2.1 360+ Videos	103
MelodyVR – 2015.....	103
3.2.2 Augmented Reality Albums	104
Bluebrain: Location-Based Album Apps — 2011	105
TCW: <i>Year One</i> – 2017.....	105
Zach Lieberman: <i>AR Audio Recording Test</i> – 2017	107
Particle Inc.: <i>Fields</i> – 2018.....	108
3.2.3 Virtual Reality Instruments	109
Snake Productions: <i>Block Rocking Beats</i> – 2017.....	109
Aesthetic Interactive: <i>EXA: The Infinite Instrument</i> – 2018.....	111
AliveIn Technology: <i>AliveInVR</i> – 2018	113
3.2.4 Virtual Reality Albums	114
Hot Sugar: <i>Melody of Dust</i> – 2017.....	115
Saint Motel: <i>saintmotelevision</i> – 2018	116
I-Exist: <i>Consciousness</i> — The VR Concept Album (2018)	118
3.2.5 Virtual Reality Experiences.....	120
Novelab: Notes on Blindness – 2017	120
3.2.6 Outliers.....	123
Fernando Ramallo & David Kanaga: <i>Panoramical</i> – 2015	123
Plan8: <i>PolyDome</i> – 2016.....	132
Particle Inc.: <i>Drops: Rhythm Garden</i> – 2018	133
Robin Arnott: <i>Soundself</i> –2018.....	135
Chapter 4: The Current State of Technology	137
4.1 360 Video	138
4.1.1 Playback & Interaction.....	138
4.1.2 Compatible Video Players.....	139
4.1.3 Supported Audio Formats	140
4.2 Augmented Reality.....	140
4.2.1 Current Capabilities of Augmented Reality.....	141
Motion Tracking	141
Plane Detection.....	141
Light Estimation	142
Face Tracking	143
4.2.2 Areas of Current AR Development.....	144

Persistent Cloud-Based Mapping.....	144
Real-Time Mesh Construction.....	144
4.2.3 Supported Audio Formats	145
4.3 Virtual Reality.....	145
4.3.1 Head-Mounted Displays.....	145
PC-tethered.....	145
Mobile.....	146
Standalone.....	147
4.3.2 Degrees of Freedom.....	147
4.3.3 Outside-In vs. Inside-Out Tracking.....	148
4.3.4 Controllers.....	150
Gamepads.....	150
3DoF Controllers	151
6DoF Controllers	151
Specialized Controllers	152
4.3.5 Eye-Tracking.....	153
4.3.6 Supported Audio Formats	154
4.4 Digital Audio Workstations.....	154
4.4.1 Ableton: Live.....	155
4.4.2 Adobe: Audition	155
4.4.3 Apple: Logic Pro X.....	156
4.4.4 Avid: Pro Tools	156
4.4.5 Cockos: REAPER.....	156
4.4.6 MOTU: Digital Performer 10.....	157
4.4.7 Steinberg: Cubase 10 & Nuendo.....	157
4.5 Plug-ins.....	157
4.5.1 Facebook: FB360 Spatial Workstation.....	158
4.5.2 Ambisonic Tool Kit (ATK).....	163
4.5.3 Blue Ripple Sound.....	165
O3A Core	165
O3A Decoding.....	167
O3A Harpex Upsampler.....	168
O3A Manipulators	169
O3A Reverb.....	171
O3A Upmixers.....	171
Binaural Surround.....	172

O3A View	173
Rapture3D Advanced	176
4.5.4 Dear Reality.....	177
DAW Plug-ins	177
VR Mix Control.....	179
4.6 Game Engines	179
4.6.1 Sources and Listeners.....	180
4.6.2 Unity	180
Audio Source Component	181
Microphone Input	182
MIDI and OSC Input/Output.....	183
Audio Effects and Audio Effect Components.....	183
Mixing.....	184
Ambisonic Audio.....	186
Licensing.....	186
4.6.3 Unreal Engine.....	186
Supported Audio Files.....	187
Sound Wave and Sound Cue Assets.....	187
Timelines.....	189
Defining Acoustic Spaces.....	190
Licensing.....	190
4.7 Audio Middleware	191
4.7.1 Firelight Technologies: FMOD Studio.....	191
Events.....	191
Logic Markers.....	192
Conditional Triggers	193
Trigger Regions	194
Instruments	194
Event Automation and Modulation	195
Built-in Parameters	195
Banks	196
Audio Effects.....	196
Stereo and Spatial Audio.....	197
Mixing.....	198
Supported Audio File Formats.....	198
Licensing.....	199
4.7.2 Audiokinetic: Wwise.....	200

Project Hierarchy	200
<i>Actor-Mixer Hierarchy</i>	200
<i>Interactive Music Hierarchy</i>	200
<i>Master-Mixer Hierarchy</i>	200
Sound Objects	201
<i>Audio Source</i>	201
<i>Source Plug-ins</i>	202
Object Containers	203
Events	204
Game Syncs	205
Interactive Music	208
<i>Music Segment</i>	208
<i>Music Containers</i>	209
<i>Transitions</i>	210
<i>MIDI Sequencing</i>	211
Positioning, Spatialization, and Animation	211
<i>Speaker Panning</i>	212
<i>Listener-Relative Routing</i>	212
<i>Spatial Animation</i>	213
Effect Plug-ins	214
Acoustic Simulations	215
Supported File Formats	216
Licensing	216
4.8 Audio Software Development Kits	217
4.8.1 Steam Audio SDK	217
4.8.2 Google Resonance SDK	218
4.8.3 Oculus Audio SDK	218
4.9 What is on the horizon?	219
4.9.1 Multidimensional Polyphonic Expression and MIDI 2.0	219
4.9.2 Volumetric Capture	220
Chapter 5: New Compositional Techniques	222
5.1 New Possibilities	222
5.1.1 Social Connections	222
5.1.2 New Approachable and Accessible Instruments	222
5.1.3 Audience-Centered and Individualized Experiences	223
5.1.4 Transportable and Transformable Audiences	223

5.1.5	Shared Time Controls	223
5.1.6	The MP3 of Installation Art.....	224
5.1.7	Pure Algorithmic	224
5.1.8	Altered Physics	225
5.2	Specific Environmental Considerations	225
5.2.1	360 Videos	225
5.2.2	iOS Instruments and Album Apps	226
5.2.3	Augmented and Virtual Reality	227
5.3	Catalog of Techniques.....	229
5.3.1	Time Triggers.....	229
5.3.2	Gaze Activation	229
5.3.3	Motion Activation.....	229
5.3.4	Real-Time Player-Controlled Parameters.....	230
5.3.5	Group Controlled Parameters	230
5.3.6	Real-Time Microphone Input	230
5.3.7	Distance and Time-of-Arrival Phasing.....	231
5.3.8	Unrealistic Scale and Proportion	231
5.3.9	Proximity-Based Deconstructions.....	231
5.3.10	Updates and Evolving Projects	232
Chapter 6: <i>Infinite Void</i>—An infinite series of works for VR.....		233
6.1	A Framework for Future Development	233
6.2	Why Virtual Reality	234
6.3	The Learning Process	234
6.4	Why Unity?.....	235
6.5	Why Wwise?.....	235
6.6	User Locomotion.....	236
6.7	Triggerable Events	236
6.8	Scene Portals.....	237
6.9	Evolutions Orbiting.....	237
6.10	Granular Constellations	239
6.11	Plans for Future Development	243
Chapter 7: Conclusion.....		244
Works Cited		248

Bibliography	265
Appendix A	282
Appendix B	283
Appendix C	285
Appendix D	287
Appendix E	288
Appendix F	289

List of Figures

- Figure 2-1. Graphic showing the first three Ambisonic orders for both FuMa and ACN channel labeling systems. Graphic sourced from the Wwise User Documentation:
https://www.audiokinetic.com/library/edge/?source=Help&id=ambisonics_channel_ordering .50
- Figure 3-1. Morton Heilig’s 1962 *Sensorama*. Image sourced from
<http://www.mortonheilig.com/InventorVR.html>..... 65
- Figure 3-2. Ivan Sutherland's *Sword of Damocles*. 66
- Figure 3-3. An image showing U2 performing in *Second Life*, 2009. 68
- Figure 3-4. Image capture of the Avatar Orchestra Metaverse rehearsing. 70
- Figure 3-5. Early rehearsal of Pauline Oliveros' “The Heart Tones” with the Avatar Orchestra Metaverse in *Second Life*, 2007. Pictures sourced from
<http://avatarorchestra.blogspot.com/p/avatar-orchestra-compositions-and.html> 70
- Figure 3-6. Premier performance of Pauline Oliveros' 'The Heart Tones' with the Avatar Orchestra Metaverse in *Second Life*, 2008. Picture sourced from
<http://avatarorchestra.blogspot.com/p/avatar-orchestra-compositions-and.html> 71
- Figure 3-7. In-app screen capture of *Ocarina*’s main instrument controls..... 73
- Figure 3-8. In-app screen capture showing *Ocarina*'s global view and the notes of a live performance streaming in real-time..... 74
- Figure 3-9. In-app screen capture of *Playground*’s first user orientation stage: learning the basic gestures. 75
- Figure 3-10. In-app screen capture from the second stage of *Playground*’s user orientation: learning more complex gestures..... 76

Figure 3-11. In-app screen capture from the final stage of <i>Playground's</i> user orientation: learning object interaction possibilities.	76
Figure 3-12. In-app screen capture from "Naga Viper Breaks," a free playground within the Herrmutt Lobby <i>Playground</i> app.	77
Figure 3-13. In-app screen capture of Moog's <i>Model 15</i> app showing the home user interface complete with patch cables and rusty ¼" jacks.	79
Figure 3-14. Jay-Z <i>Magna Carta</i> album app.	81
Figure 3-15. In-App screen capture: Opening view of Björk's <i>Biophilia</i> Album App.....	83
Figure 3-16. In-app screen capture of the "Hallow" song landing page within Björk's <i>Biophilia</i> album app.	84
Figure 3-17. In-app screen capture of Björk's <i>Biophilia</i> —poetic statement by Björk regarding the inspiration behind the song “Hallow.”	85
Figure 3-18. In-app screen capture of Björk's <i>Biophilia</i> —a still frame from the Play app for the song "Hallow."	85
Figure 3-19. In-app screen capture of Björk's <i>Biophilia</i> —a still frame from the Play app for the song "Hallow."	86
Figure 3-20. In-app screen capture of Björk's <i>Biophilia</i> —a still frame from the Play app for the song "Hallow."	86
Figure 3-21. In-app screen capture of Björk's <i>Biophilia</i> —a still frame from the Hallow Machine app for the song "Hallow" where users can use enzymes and a “DNA Replicator” to sequence musical patterns.	86
Figure 3-22. In-app screen capture of Björk's <i>Biophilia</i> —a still frame from the Animate app for the song "Hallow."	87

Figure 3-23. In-app screen capture of Björk's <i>Biophilia</i> —a still frame from the Score app for the song "Hallow."	87
Figure 3-24. In-app screen capture of Björk's <i>Biophilia</i> —a still frame from the Credits app for the song "Hallow."	87
Figure 3-25. In-app screen capture of the <i>Bloom</i> app's home screen.....	89
Figure 3-26. In-app screen capture of the <i>Boom</i> app's "Listen" screen.	89
Figure 3-27. In-app screen capture of melody creation in <i>Bloom</i> . The x-axis controls the stereo placement and the y-axis controls the relative pitch of each note.	90
Figure 3-28. In-app screen capture of available setting to tailor <i>Bloom's</i> algorithmic results.	90
Figure 3-29. In-app screen capture showing the general settings for the "Bloom" app.	91
Figure 3-30. In-app screen capture showing the <i>Air</i> app's home screen.	92
Figure 3-31. In-app screen capture of the <i>Air</i> app's "Listen" / "Conduct" page.	92
Figure 3-32. In-app screen capture showing an example of the graphical notation used in the <i>Trop</i> app.	93
Figure 3-33. In-app screen capture showing the available setting and shapes for creating music with the <i>Trop</i> app.....	93
Figure 3-34. In-app screen capture showing the <i>Air</i> app's additional setting.	93
Figure 3-35. In-app screen capture of the <i>REFLECTION</i> app showing the launch splash screen and play/pause controls.....	94
Figure 3-36. In-app screen capture 30 seconds after launching the <i>REFLECTION</i> app.....	94
Figure 3-37. In-app screen capture 1 minute after launching the <i>REFLECTION</i> app.	94

Figure 3-38. In-app screen capture of the worlds overview page from the <i>Bloom: 10 Worlds</i> app.	95
Figure 3-39. Cropped in-app screen capture showing the simplified options within each of the 10 worlds of the <i>Bloom: 10 Worlds</i> app.	96
Figure 3-40. Preview images available on the iPhone App Store for the <i>Site: Nine Elms</i> app by The Holladay Brothers.	105
Figure 3-41. In-app screen capture of <i>Year One</i> : Step one of the calibration process.	106
Figure 3-42. In-app screen capture of <i>Year One</i> : Step two of the calibration process.	106
Figure 3-43. In-app screen capture of <i>Year One</i> : Step three of the calibration process.	106
Figure 3-44. In-app screen capture of <i>Year One</i> : Step four of the calibration process.	107
Figure 3-45. In-app screen capture of <i>Year One</i> : Step five of the calibration process.	107
Figure 3-46. In-app screen capture of <i>Year One</i> : Exploring the augmented sound space.	107
Figure 3-47. Screen capture of Zach Lieberman's original Instagram post demonstrating his AR audio test. Used with written permission from Zach Lieberman.	108
Figure 3-48. In-app screen capture showing the sound orbs of Particle Inc.'s "Fields."	109
Figure 3-49. In-app screen capture showing more sound orbs of Particle Inc.'s "Fields."	109
Figure 3-50. Image showing someone creating a drum pattern the "Block Rocking Beats" VR Instrument. Image sourced from http://www.blockrockingbeats.com/	110
Figure 3-51. Image showing three people performing together in "Block Rocking Beats." Image sourced from http://www.blockrockingbeats.com/	110
Figure 3-52. Multi-player performance in the <i>EXA: The Infinite Instrument</i> VR app. Still frame pulled from a promotional video found at https://store.steampowered.com/app/606920/EXA_The_Infinite_Instrument/	111

- Figure 3-53. The four basic "ringers" used for building instruments in *EXA: The Infinite Instrument*. Image sourced from a GIF found at https://store.steampowered.com/app/606920/EXA_The_Infinite_Instrument/ 112
- Figure 3-54. Image showing one example of the strange instruments that can be built inside *EXA: The Infinite Instrument* and the new performance techniques they inspire. Image sourced from a GIF found at https://store.steampowered.com/app/606920/EXA_The_Infinite_Instrument/ 112
- Figure 3-55. Image showing three sequencers inside *EXA: The Infinite Instrument* being chained together to create a more complex sequence. Image sourced from a GIF found at https://store.steampowered.com/app/606920/EXA_The_Infinite_Instrument/ 112
- Figure 3-56. Image showing a full performance in action within *EXA: The Infinite Instrument*. Image sourced from a GIF found at https://store.steampowered.com/app/606920/EXA_The_Infinite_Instrument/ 113
- Figure 3-57. Image showing two performers using *AliveInVR* to control and Ableton Live set. Image sourced from <https://steamcommunity.com/games/674180/announcements/detail/1651009712387237652..> 114
- Figure 3-58. Still from the VR experience *When We Land*. Image sourced from the apps Oculus Store page: <https://www.oculus.com/experiences/go/944889082266875/> 115
- Figure 3-59. Opening bedroom scene from *Melody of Dust*. Image sourced from the app's Steam home page: https://store.steampowered.com/app/527110/The_Melody_of_Dust/ 116
- Figure 3-60. Looking through the double door of the *Melody of Dust* bedroom out toward the fountain. Image sourced from the app's Steam home page: https://store.steampowered.com/app/527110/The_Melody_of_Dust/ 116
- Figure 3-61. In-app screen capture showing the TV preview and remote control for navigating and choosing a 360 video to watch..... 117
- Figure 3-62. In-app screen capture showing the portal into one of the 360 videos. 117

- Figure 3-63. In-app screen capture showing the 360 video selection screen for the VR version of the *saintmotelevision* app. 117
- Figure 3-64. Kickstarter campaign photo for I-Exist's VR concept album *Consciousness*. Image sourced from I-Exist's press release materials. 118
- Figure 3-65. "The Celestial Dither," one of the "conscious instruments" that will be playable within I-Exist's forthcoming VR concept album *Consciousness*. Image sourced from the project's Kickstarter campaign page:
<https://www.kickstarter.com/projects/844845263/consciousness-the-vr-concept-album/description> 119
- Figure 3-66. What appears to be a chair inside a listening station. Image sourced from promotional materials for I-Exist's forthcoming VR concept album *Consciousness*. 119
- Figure 3-67. Distant image of what appears to be a listening station. Image source from promotional materials for I-Exist's forthcoming VR concept album *Consciousness*. 120
- Figure 3-68. In-app screen capture of the opening park scene from the *Notes on Blindness* app. 122
- Figure 3-69. In-app screen capture of using wind to expose the world inside of the *Notes on Blindness* app. 122
- Figure 3-70. In-app screen capture showing the laptop controls for *Panoramical*. 124
- Figure 3-71. In-app screen capture showing the default state of a world in *Panoramical*. 125
- Figure 3-72. In-app screen capture showing a world in *Panoramical* when the "Q" control is at its maximum value. 125
- Figure 3-73. In-app screen capture showing a world in *Panoramical* when the "W" control is at its maximum value. 126
- Figure 3-74. In-app screen capture showing a world in *Panoramical* when the "E" control is at about half of its maximum value. 126

Figure 3-75. In-app screen capture showing a world in <i>Panoramical</i> when the "E" control is at its maximum value.....	127
Figure 3-76. In-app screen capture showing a world in <i>Panoramical</i> when the "A" control is at about half of its maximum value.	127
Figure 3-77. In-app screen capture showing a world in <i>Panoramical</i> when the "A" control is at its maximum value.....	128
Figure 3-78. In-app screen capture showing a world in <i>Panoramical</i> when the "S" control is at about half its maximum value.	128
Figure 3-79. In-app screen capture showing a world in <i>Panoramical</i> when the "D" control is at its maximum value.....	129
Figure 3-80. In-app screen capture showing a world in <i>Panoramical</i> when the "Z" control is at its maximum value.....	129
Figure 3-81. In-app screen capture showing a world in <i>Panoramical</i> when the "X" control is at its maximum value.....	130
Figure 3-82. In-app screen capture showing a world in <i>Panoramical</i> when the "C" control is at about half its maximum value.	130
Figure 3-83. In-app screen capture showing a world in <i>Panoramical</i> when the "C" control is at its maximum value.....	131
Figure 3-84. In-app screen capture showing one possible variation on a world in <i>Panoramical</i> when all nine controls are enabled.	131
Figure 3-85. In-app screen capture showing a stream of shot blue ball, the sound objects, and sequencer paths in <i>Polydome</i>	133
Figure 3-86. Still frame capture from the promotional video for <i>Drops: Rhythm Garden</i> showing one complete garden in action. Image sourced from the game's Steam Store page: https://store.steampowered.com/app/864960/Drops_Rhythm_Garden/	134

- Figure 3-87. In-app screen capture showing the palette for selecting objects to place into the *Drops: Rhythm Garden*. Image sourced from the game's Steam Store page:
https://store.steampowered.com/app/864960/Drops_Rhythm_Garden/ 134
- Figure 3-88. Still frame captured from a promotional video for *SoundSelf*. Image sourced from the games website: <https://www.soundself.com/> 135
- Figure 4-1. Image showing OmniVirt's hotspots and analytics capabilities for 360 videos. Image sourced from <https://www.omnivirt.com/blog/360-interactive-video/> 140
- Figure 4-2. Graphic showing how visual inertial odometry works. Graphic sourced from <https://developers.google.com/ar/discover/concepts> 141
- Figure 4-3. Graphic showing how plane detection works. Graphic sourced from <https://developers.google.com/ar/discover/concepts> 142
- Figure 4-4. Graphic showing how light estimation works. Graphic sourced from <https://developers.google.com/ar/discover/concepts> 142
- Figure 4-5. Image captured via infrared camera showing Apple's TrueDepth camera and facial recognition system in action. Image source from <https://www.idropnews.com/news/app-developer-access-to-truedepth-camera-worries-privacy-advocates/57116/> 143
- Figure 4-6. Image showing 6D.AI's SDK rendering a 3D mesh in real time. Image sourced from <https://www.techprior.com/6d-ai-revolutionizing-ar-technology-for-3d-mapping/> 144
- Figure 4-7. Graphic showing a typical PC-tethered VR setup with outside-in tracking. Image sourced from the HTC Vive PRE User Guide (HTC, 2016) 146
- Figure 4-8. Image showing Google Cardboard. Image sourced from https://store.google.com/product/google_cardboard 147
- Figure 4-9. Graphic showing the difference between 3DoF and 6DoF. Graphic sourced from <https://www.aniwaa.com/guide/vr-ar/vr-headset-buying-guide/> 148

Figure 4-10. Image of the HTC Vive Cosmos, a standalone HMD with inside-out tracking. Image sourced from https://www.vive.com/eu/cosmos/	149
Figure 4-11. Image showing the Gear VR's simple onboard controls. Image sourced from https://groovr.zendesk.com/hc/en-us/articles/115002822988-Using-GrooVR-GearVR	150
Figure 4-12. Image of the 3DoF controller included with the Oculus Go. Image sourced from https://www.oculus.com/go/accessories/	151
Figure 4-13. Image showing the HTC Vive Cosmos controller. Image sourced from https://www.vive.com/eu/cosmos/	152
Figure 4-14. Image of the HTC Vive Tracker. Image sourced from https://www.vive.com/us/vive-tracker/	152
Figure 4-15. Image of the Virtuix Omni movement controller. Image sourced from http://www.virtuix.com/product/omni-platform/	153
Figure 4-16. The three main stages of the spatial audio production process as outlined on the Ambisonic Tool Kit's website: http://www.ambisonictoolkit.net/documentation/workflow/ ...	158
Figure 4-17. Screen capture showing the FB360 Spatial Workstation Spatialiser audio plug-in	160
Figure 4-18. Screen capture showing the FB360 Spatial Workstation Converter audio plug-in.	161
Figure 4-19. Screen capture showing the FB360 Spatial Workstation Control audio plug-in. .	161
Figure 4-20. Screen capture showing the FB360 Spatial Workstation Meter audio plug-in.	162
Figure 4-21. Screen capture showing the FB360 Spatial Workstation Stereo Loudness audio plug-in.	162
Figure 4-22. Screen capture showing the FB360 Spatial Workstation Encoder application.	162

- Figure 4-23. Screen capture showing the ATK FOA Transform FocusPressPushZoom plug-in.
..... 163
- Figure 4-24. Screen capture showing the ATK FOA Transform Mirror plug-in..... 163
- Figure 4-25. Screen capture of O3A Harpex Upsampler plug-in by Blue Ripple Sound. 169
- Figure 4-26. Screen capture of Blue Ripple Sound's O3A View Sync plug-in. Image sourced
from: <https://www.blueripplesound.com/products/o3a-view> 174
- Figure 4-27. Screen capture of editing a spatial mix while viewing a 2D representation of a 360
video via Blue Ripple Sound's O3A View app. Image source from:
<https://www.blueripplesound.com/products/o3a-view> 175
- Figure 4-28. Screen capture of editing a spatial mix with a HMD and 6DoF controller via Blue
Ripple Sound's O3A View app. Image source from:
<https://www.blueripplesound.com/products/o3a-view> 175
- Figure 4-29. Screen capture of the configuration tool in Blue Ripple Sound's Rapture3D
Advanced. Image source from: <https://www.blueripplesound.com/products/rapture-3d-advanced>
..... 176
- Figure 4-30. Picture showing dearVR's free Ambi Micro plug-in. Image sourced from:
<https://www.dearvr.com/products/ambi-micro> 177
- Figure 4-31. Screen capture showing the dearVR Music plug-in. Image sourced from:
<https://www.dearvr.com/products/dearvr-music> 178
- Figure 4-32. Screen capture showing the dearVR Pro plug-in. Image sourced from:
<https://www.dearvr.com/products/dearvr-pro> 178
- Figure 4-33. Graphic showing the basic concept and functions of dearVR Spatial Connect.
Image sourced from: <https://www.dearvr.com/products/dearvr-spatial-connect> 179
- Figure 4-34. Screen capture of an empty audio component in Unity..... 182

- Figure 4-35. Graphic showing a possible signal flow from a sound source to a listener through Unity's mixer. Graphic sourced from Unity's developer documentation:
<https://docs.unity3d.com/Manual/AudioMixerOverview.html> 185
- Figure 4-36. Unity's mixer window. Image sourced from Unity's developer documentation:
<https://docs.unity3d.com/Manual/AudioMixer.html> 185
- Figure 4-37. Image showing a typical Blueprint in Unreal Engine. Image sourced from the Unreal Engine developer documentation: <https://docs.unrealengine.com/en-us/Engine/Blueprints>..... 187
- Figure 4-38. Image showing the Sound Cue Editor in Unreal Engine. Image sourced from the Unreal Engine developer documentation: <https://docs.unrealengine.com/en-us/Engine/Audio/SoundCues/Editor>..... 187
- Figure 4-39. Image showing a simple example of a Timeline in the Unreal Game Engine. Image sourced from Unreal's developer documentation: <https://docs.unrealengine.com/en-US/engine/blueprints/userguide/timelines/keysandcurves> 189
- Figure 4-40. Image of the Audio Volume component in the Unreal Game Engine. Image sourced from Unreal's developer documentation: <https://docs.unrealengine.com/en-us/Engine/Actors/Volumes/AudioVolumeDetails> 190
- Figure 4-41. Image of the Reverb Effect in the Unreal Game Engine. Image sourced from Unreal's developer documentation: <https://docs.unrealengine.com/en-us/Engine/Audio/Overview> 190
- Figure 4-42. Screen capture showing an event's timeline in FMOD Studio 1.10. 192
- Figure 4-43. Screen capture showing FMOD Studio's spatializer audio effect. 197
- Figure 4-44. Screen capture of the FMOD Studio's mixer window. 199
- Figure 4-45. Graphic depicting the Wwise project hierarchy. Graphic sourced from the Wwise documentation:

https://www.audiokinetic.com/library/edge/?source=WwiseFundamentalApproach&id=understanding_master_mixer_hierarchy.....201

Figure 4-46. Graphic showing the sound object, audio source, and audio file structure in Wwise.

Graphic sourced from the Wwise documentation:

https://www.audiokinetic.com/library/edge/?source=WwiseFundamentalApproach&id=audio_objects202

Figure 4-47. Graphic showing a single sound object that reference two audio files with several different audio source settings. Graphic sourced from the Wwise documentation:

https://www.audiokinetic.com/library/edge/?source=WwiseFundamentalApproach&id=audio_objects202

Figure 4-48. Graphic providing an example of an Actor-Mixer Hierarchy in Wwise. Graphic sourced from:

https://www.audiokinetic.com/library/edge/?source=Help&id=using_containers_and_actor_mixers_within_actor_mixer_hierarchy.....203

Figure 4-49. An example of a simple Wwise event that plays one sound object while stopping another. Graphic sourced from the Wwise documentation:

https://www.audiokinetic.com/library/edge/?source=WwiseFundamentalApproach&id=understanding_events_understanding_events204

Figure 4-50. Example of a Wwise event that plays one sound object, sets the volume of a bus track in the Master-Mixer Hierarchy, and pauses all events except for those that pass through the music and menu bus tracks of the Master-Mixer Hierarchy. Graphic sourced from the Wwise documentation:

https://www.audiokinetic.com/library/edge/?source=WwiseFundamentalApproach&id=understanding_events_understanding_events205

Figure 4-51. An example of how the State Game Sync could be used to change the sound of gunfire and grenades exploding when the player jumps or goes under water. Graphic sourced from the Wwise documentation:

https://www.audiokinetic.com/library/edge/?source=WwiseFundamentalApproach&id=understanding_states.....206

Figure 4-52. An example of how a Switch Game Sync can be used to switch between two groups of footstep sounds in random containers depending on the surface type the player is walking upon. Graphic sourced from the Wwise documentation:

https://www.audiokinetic.com/library/edge/?source=WwiseFundamentalApproach&id=understanding_switches206

Figure 4-53. An example of linking a car's speed to the volume of a Wwise event. Graphic sourced from the Wwise documentation:

https://www.audiokinetic.com/library/edge/?source=WwiseFundamentalApproach&id=understanding_rtpcs.....207

Figure 4-54. An example of a trigger used to start a short musical phrase when the player performs a special action. Graphic sourced from the Wwise documentation:

https://www.audiokinetic.com/library/edge/?source=Help&id=working_with_triggers_overview207

Figure 4-55. A graphic illustrating how Music Segments are sequenced and aligned at their markers. Graphic sourced from the Wwise documentation:

https://www.audiokinetic.com/library/edge/?source=Help&id=what_is_music_segment209

Figure 4-56. An example of an Interactive Music Hierarchy constructed using a Music Switch container. Graphic sourced from the Wwise documentation:

https://www.audiokinetic.com/library/edge/?source=Help&id=defining_contents_and_behaviors_of_music_switch_containers.....210

Figure 4-57. An example of an Interactive Music Hierarchy using a Music Playlist container. Graphic sourced from the Wwise documentation:

https://www.audiokinetic.com/library/edge/?source=Help&id=defining_contents_and_behaviors_of_music_switch_containers.....210

Figure 4-58. An example of how transitions can be used to move from one Music Playlist to another. Graphic sourced from the Wwise documentation: https://www.audiokinetic.com/library/edge/?source=Help&id=working_with_transitions_overview	211
Figure 4-59. Screen capture showing the Balance Control panel in Wwise.....	212
Figure 4-60. Screen capture of the Position tab in Wwise.....	213
Figure 4-61. Screen capture showing the Position Editor in Wwise.....	213
Figure 4-62. Image showing the use of raycasting to calculate the early reflections of a single sound source heard by a listener. Image source from: https://blog.audiokinetic.com/acoustic-simulation-with-wwise/	215
Figure 4-63. A graphic showing the room and portal system used by Wwise Spatial Audio. Graphic sourced from: https://www.audiokinetic.com/products/wwise-spatial-audio/	216
Figure 4-64. Screen capture of the Oculus Spatializer VST plug-in.	219
Figure 4-65. Photo of a volumetric capture stage. Photo courtesy of sound engineer/composer Joel Douek of ECCO VR: http://www.eccovr.com/	221
Figure 6-1. Screen capture of the "Evolutions Orbiting" from the initial version of <i>Infinite Void</i>	238
Figure 6-2. Screen capture of the first two subgroups of the Wwise music playlist container for the "Evolutions Orbiting" scene of <i>Infinite Void</i>	239
Figure 6-3. Screen capture of Wwise's SoundSeed Grain source plug-in as used in the "Granular Constellations" scene of <i>Infinite Void</i>	241
Figure 6-4. Screen capture showing the RTPC mapping for <i>Infinite Void's</i> "Granular Constellations."	241

Figure 6-5. Screen capture showing the attenuation curves used in *Infinite Void's* "Granular Constellations" scene242

List of Tables

Table 3-1. Listing of all Album Apps uncover by the author.	97
Table 4-1. A full list of the plug-in included in the Ambisonic Tool Kit.....	164
Table 4-2. A full list of the plug-ins included in Blue Ripple Sound’s O3A Core bundle. Information found on Blue Ripple Sound's website: https://www.blueripplesound.com/products/o3a-core	166
Table 4-3. A full list of the plug-ins included in Blue Ripple Sound’s O3A Decoding bundle. Information found on Blue Ripple Sound's website: https://www.blueripplesound.com/products/o3a-decoding	167
Table 4-4. A full list of the plug-ins included in Blue Ripple Sound’s O3A Manipulators bundle. Information found on Blue Ripple Sound's website: https://www.blueripplesound.com/products/o3a-manipulators	169
Table 4-5. A full list of the plug-ins included in Blue Ripple Sound’s O3A Manipulators bundle. Information found on Blue Ripple Sound's website: https://www.blueripplesound.com/products/o3a-reverb	171
Table 4-6. A full list of the plug-ins included in Blue Ripple Sound’s O3A Manipulators bundle. Information found on Blue Ripple Sound's website: https://www.blueripplesound.com/products/o3a-upmixers	171
Table 4-7. A full list of the plug-ins included in Blue Ripple Sound’s Binaural Surround bundle. Information found on Blue Ripple Sound's website: https://www.blueripplesound.com/products/binaural-surround	173
Table 4-8. A full list of the plug-ins currently controllable via Blue Ripple Sound’s O3A View app. Information found on Blue Ripple Sound's website: https://www.blueripplesound.com/products/o3a-view	174

Table 4-9. A list of the plug-ins included with Blue Ripple Sound's Rapture3D Advanced. Information found on Blue Ripple Sound's website: https://www.blueripplesound.com/products/rapture-3d-advanced	176
Table 4-10. List of Unity's supported audio file formats. Information gathered from Unity Developer Documentation: https://docs.unity3d.com/Manual/AudioFiles.html	181
Table 4-11. Unity's included audio effect components. Information gathered from Unity's developer documentation: https://docs.unity3d.com/Manual/class-AudioEffect.html	183
Table 4-12. Unity's included audio effects. Information gathered from Unity's developer documentation: https://docs.unity3d.com/Manual/class-AudioEffectMixer.html	183
Table 4-13. List of Unity's licensing options. Information gathered from Unity's website: www.unity3d.com	186
Table 4-14. List of all currently available Sound Cue Nodes in Unreal Engine's Sound Cue Editor. Information gathered from the Unreal's developer documentation: https://docs.unrealengine.com/en-us/Engine/Audio/SoundCues/NodeReference	188
Table 4-15. List of FMOD's logic markers. Information gathered from FMOD's user manual: https://www.fmod.com/resources/documentation-studio?version=2.0&page=authoring-events.html#timeline-logic	193
Table 4-16. Complete list of FMOD's included instrument types. Information gathered from FMOD's user manual: https://www.fmod.com/resources/documentation-studio?version=2.0&page=authoring-events.html#instruments	194
Table 4-17. Complete list of FMOD's included modulators. Information gathered from FMOD's user manual: https://www.fmod.com/resources/documentation-studio?version=2.0&page=authoring-events.html#automation-and-modulation	195
Table 4-18. List of FMOD's built-in parameters. Information gathered from FMOD's user manual: https://www.fmod.com/resources/documentation-studio?version=2.0&page=authoring-events.html#parameters	196

Table 4-19. A complete list of the audio effects included in FMOD.	197
Table 4-20. List of FMOD licensing options. Information gathered from FMOD's website: http://www.fmod.com	199
Table 4-21. A list of all the source plug-ins included in Wwise. Note: some sound source plug-ins require additional licensing.....	202
Table 4-22. Actor-Mixer Hierarchy container types. Information gathered from the Wwise documentation: https://www.audiokinetic.com/library/edge/?source=Help&id=building_actor_mixer_hierarchy_building_actor_mixer_hierarchy	203
Table 4-23. List of the Wwise Game Syncs. Information gathered from the Wwise user documentation: https://www.audiokinetic.com/library/edge/?source=WwiseFundamentalApproach&id=what_are_game_syncs	205
Table 4-24. A list of all Wwise's effect plug-ins. Information gathered from the Wwise documentation: https://www.audiokinetic.com/library/edge/?source=Help&id=effects	214
Table 4-25. List of Wwise licensing options: Information gathered from Audiokenetic's website: https://www.audiokinetic.com	217

Chapter 1: Introduction

Throughout music history, each new format for producing, performing, capturing, sharing, or consuming music has expanded the horizon of musical possibilities. But within the past 100 years, perhaps no other development has been as influential as the development of recording technology. As musicologist Mark Katz explains in his book “Capturing Sound,” in which he traces the history of recording technology and how it has changed the way people both make and listen to music, recordings do more than just capture reality—they also change it. He refers to this change as the “phonograph effect.”

Simply put, a phonograph effect is any change in musical behavior—whether listening, performing or composing—that has arisen in response to sound-recording technology. (Katz, 2010)

Revised in 2010, Katz’s history ends with the effects that web-based file sharing and the MP3 have had on music consumption, a topic more deeply explored several years later by Jonathan Sterne in his book “MP3: The Meaning of a Format.” Neither of these books, however, were published recently enough to include the range of new digital formats that have become available in the last decade through major technological advancements backed by both serious financial resources and rising consumer interest.

August 1, 2012, marked the launch of a Kickstarter campaign (Oculus, 2015) that ultimately raised more than \$2 million in pledges to support the development of a new virtual reality (VR) headset by then-teenager Palmer Luckey (Kickstarter, 2016). Two years later, shortly after Luckey delivered the first VR headsets, Facebook acquired his company, Oculus, for \$3

billion (Constine, 2014). On October 13, 2016, Sony Interactive Entertainment (SIE) released a head-mounted display called “PlayStation VR,” which a little over a year later had already sold more than 2 million units (Joyce, 2017). By that same time, the Oculus Rift and another VR headset, the HTC Vive, had both sold around 200,000 units each (Matney, 2017). All three of these VR headsets targeted the dedicated video gamer market. Recognizing this, Google’s Cardboard (2014), Samsung’s Gear VR (2015), and more recently Oculus’ Go (2018) offered cheaper, less functional VR systems that sought to entice more mainstream consumers to begin exploring what VR had to offer. According to an eMarketer study, there were at least 9.6 million active users of VR headsets in 2017 (Roettgers, 2017).

While virtual reality is undoubtedly the poster child for new interactive and immersive environments that are finding footholds in global entertainment markets, VR is not alone. On March 13, 2015, Google’s video-sharing service YouTube began supporting 360-degree videos (Hollister, YouTube's Ready To Blow Your Mind With 360-Degree Videos, 2015). Facebook, not wanting to be left behind, updated its platform to support 360-degree videos several months later (Maher, 2015). Looking to influence the future of these technologies, Apple announced at its 2018 Worldwide Developers Conference (WWDC) the release of its AR Kit 2, which aims to make integrating augmented reality (AR) features into apps easier for developers (Apple, 2018). And although the technology is still very much in its infancy, AR already has acquired a substantial user base: the same eMarketer study indicated there were 40 million active users of AR-enabled apps in 2017.

All of these indicators point to a rapid pace of advancements in entertainment technologies, many of which consumers are adopting faster than creators can even provide content for. But even though these new environments open vast worlds of creative possibilities, revealing

new and uncharted territories of artistic expression, most of the content that has already been created only mimics older practices.

Take Netflix's VR App, which places the experiencer on a couch in a virtual living room with a large, flat-screen television on the wall accompanied by a virtual remote to navigate and watch Netflix's content just as if they were sitting in their actual living room (Oculus, 2018). Contrast that with *Dispatch* (Oculus, 2017), a mini-series created by Ed Robles and starring Martin Starr. Each episode of *Dispatch* places the viewer inside the consciousness of police dispatcher Ted as he receives a stream of gut-wrenching emergency phone calls. The drama is conveyed almost entirely via the sounds that Ted hears through his phone, but as each scenario develops, ghost-like impressions representing Ted's imagination of the scene begin to fill in around the viewer. *Dispatch* is an excellent example of how the art of storytelling can be enhanced and expanded through these new digital environments.

Yet while there is a growing body of documentation for these new tools that support the creation of new digital spaces generally, as of yet there is no comprehensive collection of the fundamental concepts, representative works, and new compositional techniques emerging with these new digital formats. The purpose of this dissertation is to examine the concepts and tools at the core of these new formats and begin the work of understanding what new compositional and experiential possibilities they present. This dissertation will serve as both a foundation and a point of departure for composers interested in influencing the direction that musical and creative expression will take in these immersive and interactive digital environments.

Chapter 2: **Sound Localization and Spatialization**

When considering what techniques are available for positioning a sound in space, it is essential first to develop an understanding of how with only two ears, humans can perceive a sound's location in three-dimensions. While most of the spatialization techniques that have developed from replicating human spatial audio perception remain similar across audio delivery systems (e.g., stereo vs. surround, speakers vs. headphones), there are unique difficulties, as well as some compelling benefits and future potentials, of each system and the audio file formats associated with them. As a result, a discussion of possible delivery systems and their related file formats will help guide a creator in making choices and troubleshooting any issues that may arise when spatializing audio within each format. Finally, with the proliferation of powerful personal electronic devices that are making immersive, but more importantly, interactive virtual environments (e.g., YouTube/Facebook 360, augmented reality, virtual Reality) more widely accessible, replicating or modeling the ways a sound interacts with its surroundings is becoming more and more vital to giving the listener a more realistic sense of auditory presence.

2.1 Spatial Perception

Identifying a sound's spatial location requires listening for different cues in each of the four sonic dimensions: lateral, front/back, elevation, and distance. In each of these dimensions, the component frequencies or qualities of a sound will affect how, or if, localization is possible.

2.1.1 Lateral Perception

A listener can reliably identify the lateral location of a sound by perceiving differences in amplitude and timing between each ear.

Amplitude

In general, when a sound is louder in one ear than the other, the listener will perceive the sound as being on whichever side is louder. The bigger the differences in amplitude (*interaural intensity differences*, IID) the more to one side or the other it will appear to be. However, sounds below ~800 hertz (Hz) become increasingly difficult to discern IID because of the size of their wavelength relative to the listener's head. In this range, *interaural time differences* may help.

Timing

Human ears are on average 21.5 centimeters apart. Sound travels through air at approximately 343,000 centimeters per second which, depending on a sound's location can result in small timing differences in the range of 0 to ~625 microseconds. These differences are called interaural time differences, or ITD, and two aspects of ITD help a listener locate a sound: onset/arrival, and more subtly, phase relationship.

Onset Timing

A sound that originates from one side will arrive at the nearest ear first before arriving at the other. The greater the ITD of the sound's arrival, the more to one side or the other it will appear to be. If a sound has no clear onset quality, listening for interaural phase differences can aid in locating the sound.

Interaural Phase Relationships

From 0 to ~80 hertz the wavelengths are too large in relationship to most heads and both IID or interaural phase difference are too minuscule to be detected giving the sensation of an omnidirectional sound.

For frequencies from around 80 to ~800 Hz, IID are too challenging to identify, but differences in phase can be detected and relied upon for localization.

There is a sweet spot from ~800 to ~1600 Hz where both IID and ITD are both useful for sound localization but sounds above this range have wavelengths that are too small, completing more than half a cycle within 21.5 cm. While interaural phase difference can still be detected, they can no longer be used to locate a sound accurately. Much like the illusion of a forward rotating wheel appearing to rotate backward when captured on film, the interaural phase relationship of a sound above ~1600 Hz can be deceptive as to its origin direction.

The wavelength of frequencies which exceed ~1600 Hz become small enough to be obfuscated by the average human head. If a sound contains frequency above ~1600 Hz subtle high-frequency attenuation will be audible at the ear opposite from the sound origin. This phenomenon is known as *head shadowing*. Listening for head shadowing can help with localization.

2.1.2 Front/Back and Elevation

If a listener is not able to alter the axis relationship of their ears to a sound, they may not rely upon IID or ITD for determining if a sound is in front of, behind, above, or below them. Instead, they must listen for spectral modifications caused by their head, neck, torso, and pinnae (outer ear). While the lateral perceptions of a sound are already somewhat person-specific—where the distance between the listener’s ears determines the frequency crossovers between the different lateral localization modes—the way bodies spectrally modify sounds depending upon their spatial relationship to the sound is person- and even clothing -specific. The individual nature of how sound localization works becomes a significant hurdle for delivering 360° audio environments that accurately translate to each listener.

Direction Selective Filters

Similar to head shadowing, the shape of a listener’s head, neck, torso, and pinnae filter the sound differently depending upon its relative location. For example, consider a sound that is directly behind a listener. The sound will arrive at both ears simultaneously, eliminating the use of both IID and ITD, but because the listener’s ears are angled slightly forward, some higher frequencies will be filtered out (reflected off the backs of the ears). The sound that is received by the listener's ears will have a different spectral characteristic than if that same sound source originated from in front of the listener.

Head-Related Transfer Functions

A person's Direction Selective Filters are captured in an anechoic chamber as Head-Related Impulse Responses (HIRR). This process involves placing small microphones in each of the listener's ears, and recording either white noise pulses or full-spectrum sine tone sweeps for a

higher signal-to-noise ratio (SNR) at as many points as possible in a one-meter sphere surrounding the listener. An HRIR records how the listener's body alters any frequency arriving from any of those recorded locations. A captured HRIR can then be applied to any sound as a Head-Related Transfer Function (HRTF) to deceive a listener into believing the sound is coming from some location as the applied HRTF.

Head Motion

Head motion is one final localization strategy. A listener can rotate their head to align their ears along any axis and turn any challenging perceptual axis into a lateral localization problem.

2.1.3 Distance

Detecting the distance to a sound is relatively simple, but as will be seen later, accurately recreating the sensation of distance in synthetic audio environments can be the key to successfully immersing the listener. Perceiving a sound's distance requires listening beyond IID, ITD, and HRTFs to cues given to us by the sound's environmental interactions.

Loudness

For sound sources that are familiar—a car engine, a musical instrument, someone speaking—one way to determine its distance with relative accuracy is by comparing previous experiences with similar sounds against the sound's current perceived volume. With strange sounds, listening for relative volume changes over time can reveal if the sound is at the very least moving closer or further away. If the sound is not moving, listening for other environmental cues can help determine its distance.

Initial Time Delay

If the environment has sonically-reflective surfaces, a sound's approximate distance can be discerned by comparing the arrival time of the direct sound against the arrival of the first reflections of that sound off the surrounding surfaces. The longer the time between the direct sound and the reflected sound, the closer the sound is perceived to be. This method is not an option in open environments that are void of any nearby reflective surfaces.

The Ratio of Direct Sound to Reverberation

The late reflections or diffusion of a sound within a reverberant space will eventually overpower the direct component of a sound when it gets too far away from the listener. A similar result may also occur if a wall or large object occludes the direct path from the sound to the listener.

Motion Parallax

How quickly a sound can shift perceptually from the left or right side can indicate how far away a sound is. Something that is very close will quickly transfer from one side to the other, while a sound that is further away will take longer.

High-Frequency Attenuation

Higher frequencies tend to be reflected and more easily absorbed by objects, or the medium (air) through which they are traveling, so as a sound gets further away its higher frequency content will decrease in amplitude more rapidly than its lower frequency components. The effect is very subtle and usually not audible until the sound is hundreds, if not thousands of feet away, but this is one final sonic distance cue.

2.2 Audio Delivery Systems

Before looking at how to spatialize a sound in artificial environments, a brief pause should be taken to consider the different systems currently available for delivering audio. Each system has its own set of benefits and limitations.

2.2.1 Personal Delivery Systems

For a growing portion of the population, the most common way to consume music is through a personal listening system, like over-ear headphones or earbuds. Personal listening systems are currently the ideal means for experiencing spatial audio environments. Typically, personal listening systems allow for better acoustic isolation from the external world, simplified implementation of head-tracking, avoidance of the many issues that may arise when applying HRTFs for sound localization, keeping the listener in the “sweet spot,” and maximum – to name a few.

Personal delivery systems come in a variety of forms.

Closed-Back Headphones

Closed-back headphones can offer great acoustic isolation and generally better low-frequency response but at the potential cost of long-term comfort and accurate sound reproduction due to internal resonance and pinnae deformations. Closed-back headphones can either be on-ear or over-ear. Over-ear closed-back headphones offer superior acoustic isolation and comfort, but on-ear closed-back headphones tend to be more portable and more affordable.

Open-Back Headphones

Open-back headphones can be more comfortable over extended listening sessions and tend to deliver a more natural sounding spatial representation at the cost though of less acoustic isolation from the external world. Almost all open-back headphones are designed to fit over the listener's ears.

Earbuds

As the cheapest and most accessible personal listening solution, earbuds also tend to offer the lowest audio fidelity, least acoustic isolation, and poorest bass response. Excellent low-frequency response is not necessarily relevant to audio spatialization, but can significantly raise the level of immersion or sense of presence the listener experiences.

In-Ear Monitors

In-Ear Monitors create the best situation for delivering high-quality spatialized audio experiences. They usually offer excellent frequency responses across the full range of human hearing and a high degree of acoustic isolation from the outside world. They also bypass any spectral modification from the listener's pinnae. However, they also bypass the ear canal which can prevent the proper receptions of HRTFs.

2.2.2 External Delivery Systems

Historically thought of as providing a better listening experience, current consumers are trending away from traditional hi-fi home stereo systems and towards more portable wireless solutions. However, both traditional and contemporary wireless home stereo systems are inferior to personal systems at delivering spatial audio. Discussing these systems and their specific

spatialization techniques is still important, but their limitations make them impractical for use in widely distributed interactive and immersive audio environments.

Today, probably the most used external delivery systems are the speakers built-in to our personal-electronic devices (e.g., mobile phones, laptops, hand-held game consoles). These speakers typically have very narrow frequency response ranges and little if any separation. Some listeners may use Bluetooth/wireless speakers that offer broader frequency responses but deliver poor channel separation and add audio/visual synchronization issues. While there are undoubtedly spatial translation issues when using personal delivery systems, many more issues can arise when using speakers: poor isolation, no head-tracking, narrow “sweet spots,” and room effects.

Two final and significant limitations of delivering spatial audio with external delivery systems are accessibility and portability. An ideal speaker setup would require two speakers to the front left and front right of the listener for delivering lateral stereo audio and two speakers to the rear left and rear right of the listener for front/back stereo audio.

Only two dimensions of audio are possible with four speakers. To complete the third dimension, another set of four speakers above the listener, and another set of four speakers below the listener, as well as the ability to suspend the listener in the exact center of all twelve of these speakers is needed. Moreover, this list does not even account for the additional equipment and cabling required to power and run this system. The cost, ongoing maintenance, and expertise required to properly install, calibrate, and operate such a system dramatically limits the availability and portability of an external system capable of delivering compelling spatial audio experiences.

Negatives aside, using external speaker systems for delivering spatial audio experiences make group experiences and real spatialization possible. Such systems can also simplify the computations and processing required to create complex spatial effects.

Standard Speaker Configurations

Many configurations are possible, but there are several standard surround-sound speaker setups. Each of these configurations will commonly have a subwoofer and a center channel added. The subwoofer is used to extend the frequency range and can be positioned in a variety of location because of the omnidirectional behavior of low frequencies. The center channel is usually reserved for dialog and placed directly in front of the listener in line with the front two speakers.

Stereo

Two speakers positioned in an equilateral triangle and at ear level with the listener.

Quadraphonic

A front stereo pair and a rear stereo pair of speakers. There are a variety of ways to arrange four speakers. Most computer/academic music positions the speakers in a perfect square with the listener at the very center and ear level.

5.1 Surround Sound

Commercial or home theaters typically expand on the quadraphonic setup by adding a center speaker in front of the listener and a subwoofer. The front speakers are then repositioned to be at ear level with a 20-30° angle towards the listener and the rear speakers are placed nearly alongside and slightly above ear level with a 90-110° orientation respectively.

Octaphonic

Implied by the name, octaphonic arrangements use eight equally spaced speakers at ear level in either a circle or a square with the listener at the center. Two conventional channel ordering commonly used: Stereo pairs equally spaced from the front to the rear, or a clockwise ring numbered sequentially starting directly in front of the listener.

Further Channel Expansion

Larger commercial movie theaters and specialized music venues/research facilities often expand upon these standard speaker arrangements by adding speakers along the horizontal and vertical planes.

Adding along the horizontal plane can increase lateral localization accuracy as well as expand the “sweet spot,” positioning more listeners in the ideal listening zone. Adding speakers above and below the audience not only allows for the vertical placement of sounds but can also substantially add to the audience’s sensation of envelopment in the sonic environment.

2.3 Distribution Formats

In addition to knowing the delivery system, knowing the audio delivery format can also dictate which spatialization techniques will be available and most effective. The following are some of the often-heard terms about audio file formats, and their significance in delivering spatialized audio.

2.3.1 Compressed vs. Uncompressed

Digital audio file types fall into two major categories: compressed and uncompressed. Most portable electronic devices default to compressed audio files because they require less storage space than uncompressed. Both compressed and uncompressed audio files can deliver spatial audio, but some compressed audio formats can introduce sonic degradation that may alter or confuse spatial effects.

The two primary uncompressed audio files are Waveform Audio File Format (.WAV) and Audio Interchange File Format (.AIFF).

Lossless

Lossless compression algorithms decrease that amount of storage a file requires without removing or losing any information.

As an example, we could take this series of numbers:

00000000 11111111 000 111 00 11 0 1

Which could be expressed it more efficiently as:

80 81 30 31 20 21 10 11

Compressing an audio file without losing any of the original data is a little bit more complicated than the aforementioned example, but hopefully, it provides the general concept of how lossless compression can work.

Some common lossless audio file formats are:

Free Lossless Audio Codec (FLAC) uses a linear prediction method for compressing files by 40% to 70%.

Apple Lossless Audio Codec (ALAC), which is commonly stored within an MPEG 4 Audio container file (M4A), also uses a linear prediction method capable of reducing files by as much as 60% of their original size.

Lossy

Lossy compression can be a significant concern even for non-spatialized audio because file sizes are reduced by removing information from the file.

The Motion Picture Experts Group Layer-3 (MP3) file was conceived at a time when exchanging and storing large amounts of data was still a costly task. The developers of the mp3 discovered they could reduce an audio file's size by removing information that was considered less perceptually relevant. As has been seen, however, accurate delivery and localization of spatial audio depends on the faithful reproduction of the entire audio signal. Removing or altering the audio in any way can distort or confuse the listener's ability to locate the sound. Some common lossy audio file formats are:

Motion Picture Experts Group Layer-3 Audio (MP3)

Advanced Audio Coding (AAC)

File Container Formats

Arising in the early 2000s, the OGG file is a free and accessible example of a flexible file container format that can carry a range of both lossless and lossy audio file formats along with

video files. Several other container formats are QuickTime's .MOV, the Motion Picture Experts Group (MPEG) file stream, and the Audio Video Interleave (AVI) file.

While file container formats can offer many conveniences for developers and audio professionals, they can be risky to work with because it can be difficult to determine which file formats they can contain.

2.3.2 Discrete vs. Matrixed

Surround audio delivers the information to each output channel as either a discrete or matrixed stream of data. For a discrete channel, the preferred and more precise of the two, all the audio information streaming to an output channel is unique and independent of all the others. A matrixed channel, on the other hand, borrows portions of another channel's stream of audio information to create a faux surround experience.

2.3.3 Ambisonics

Ambisonic audio has been around since its development in the 1970s by the British National Development Corporation and is perhaps the most useful, and yet misunderstood audio format for efficiently capturing, creating and distributing surround audio experiences. For most of its lifetime, Ambisonic audio has only been used by academics and audiophiles. However, with the recent arrival of more portable, visually immersive and interactive digital environments, Ambisonic audio is finally finding large-scale commercial applications.

Ambisonic Orders

Ambisonic audio ranks its spatial accuracy in terms of orders. The greater the order of Ambisonics, the higher the spatial resolution and the more channels of audio required to capture and store all the spatial information. Imagining an array of microphones with different pickup-patterns all capturing the sound arriving at a single point from different directions is the best way to explain how Ambisonic audio works and what differentiates its orders.

Zeroth-Order

Ambisonics would be the equivalent of listening to a room with an omnidirectional microphone – all sounds are heard but without any spatial information.

First-Order (FOA)

Ambisonics requires four channels of audio, referred to as W, X, Y, and Z. Continuing with the imagined array of microphones, First-Order Ambisonics maintains the omnidirectional microphone (W) but adds three bidirectional microphones oriented along the three-dimensional axes: left/right, front/back, and up/down.

Second-Order (SOA)

Ambisonics requires nine channels of audio (WXYZRSTUV) and adds five additional bidirectional microphones to the imagined array, each oriented along evenly distributed axis.

Third-Order (TOA)

Ambisonics requires 16 channels of audio (WXYZRSTUVKLMNOPQ) and adds another seven bidirectional microphones.

Higher-Orders (HOA)

Higher-Order Ambisonics (HOA) is possible but is currently uncommon because of the number of channels required. As an example, Sixth-Order Ambisonics would require 49 channels of audio.

Ambisonic Formats

Ambisonic audio has several different formats. Each format references the different states of Ambisonic audio from capture to playback. Rarely are any of these formats discussed except A- and B-Format.

A-Format

The direct signal captured by an Ambisonic microphone.

B-Format

The encoded audio signal stored as a multichannel-audio file ready for decoding/playback. Usually refers to First-Order (4-channel) files but can also be used to describe higher orders.

C-Format (Consumer Format)

A proposed format in the early days of Ambisonic audio that strove to make the playback of Ambisonic audio possible on home stereos. It was not widely adopted and is rarely used today.

D-Format

An Ambisonic audio stream decoded for any delivery system/configuration.

G-Format (Geoffrey's Format)

Named by its inventor Geoffrey Barton, was another attempt at driving the commercial adoption of Ambisonics by creating a default decoded format for 5.1 home theater setups.

Additional Formats

In addition to the lettered formats listed above, several conventions exist which define channel organization. FuMa (Furse-Malham) organize the channels as WXYZ and follows the MaxN normalization standard for all channels except W which is reduced by 3dB. AmbiX, on the other hand uses SN3D normalization and follows the ACN channel ordering (Hodges, Ambisonic Software, 2018).

Ambisonics formats may adopt new conventions as higher orders become more practical, so it will be essential to monitor how each new convention organizes the channels.

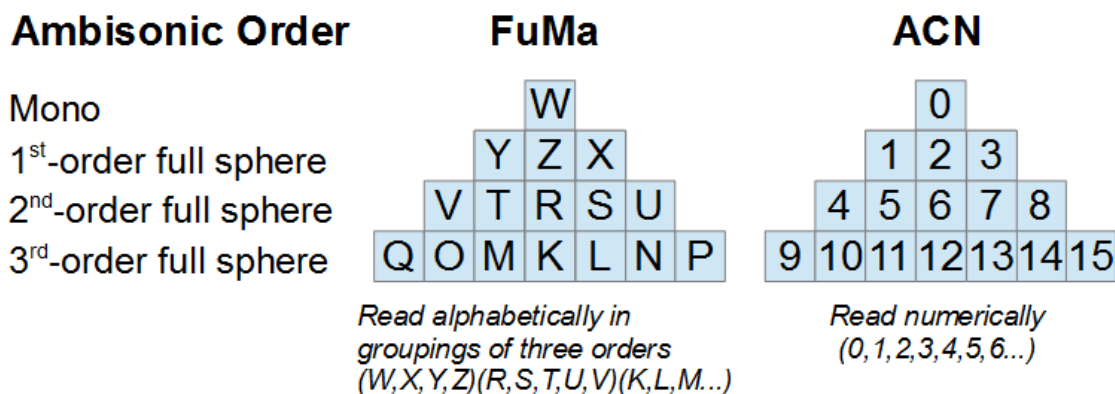


Figure 2-1. Graphic showing the first three Ambisonic orders for both FuMa and ACN channel labeling systems.
 Graphic sourced from the Wwise User Documentation:
https://www.audiokinetic.com/library/edge/?source=Help&id=ambisonics_channel_ordering

Pros and Cons of Ambisonics

There are positives and negatives to working with Ambisonic audio. The greatest strength of Ambisonic audio is undoubtedly the separation of its two primary states: encoded and decoded. With this separation, Ambisonic audio can take an audio source of any format (mono, stereo, quadraphonic, A-Format) and encode it to any order of Ambisonics. The encoded sound source can then be decoded to any given delivery system (stereo, headphones, quadraphonic, 5.1) and the spatial information will be preserved at a relatively high level of accuracy.

Ambisonic audio is also much more efficient at storing and delivering multichannel audio with fewer channels than would be required by a discrete multichannel format. For example, First-Order Ambisonics only requires four channels to deliver six directions of spatial information which could be decoded to any number of speakers.

A final selling point of Ambisonic audio is that the format is free of patents and there are many free tools for capturing, manipulating, encoding, and decoding Ambisonic audio available for all major operating systems.

Ambisonic audio is not without its weaknesses. One major shortcoming is that Ambisonic audio—despite a recent reinvigoration with the increased distribution of AR, VR, and 360 Videos—has not been widely adopted by audio professionals, likely because it is conceptually difficult to understand, and its perceptual results can be difficult to judge.

Sonically it is also not perfect for delivering spatial audio: the “sweet spot” is tiny, spatial confusion commonly occurs, the audio can be heavily colored from comb-filtering when played over speakers, and setting up an ideal delivery system can be extremely challenging even for experienced engineers.

2.3.4 Binaural Audio

A final format that is essential to making spatial audio easily consumable by a large audience using readily accessible delivery systems is *binaural audio*. The term mostly refers to a recording technique that mounts microphones either in a dummy's or an actual human's ears—similar to the recording technique used to measure and capture HRIRs. This arrangement positions microphones perfectly to capture sounds that are imprinted with the IIDs, ITDs, and other spectral modification (HRTFs) resulting from a subject's body, head, and ears. Then on playback, the consumer will have an aural experience similar to that heard at the original .

Because the crosstalk (sound emitted by one speaker which is then received by both ears) that would occur when playing binaural audio over speakers would distort the binaural reproduction, binaural audio can only be accurately played back using a personal-delivery system. Binaural audio makes an ideal partner to Ambisonics for mass delivery of immersive and interactive audio experiences to individual consumers. The creators of the audio experience can encode all the spatial audio as a B-format Ambisonic audio file. Then the consumer's playback system (a cellphone and cheap earbuds) can decode and render a binaural realization of the spatial audio in real time as determined by any interaction their playback system may afford them.

2.4 Spatialization Techniques

Audio spatialized for one delivery system will not translate accurately to another delivery system because of their different acoustic environments. Additionally, each environment must account for the listeners movements in different ways.

2.4.1 External Delivery Systems

From the perspective of spatialization techniques and processes, external delivery systems are the easiest to use. To make a sound to appear to come from a specific location, place a speaker at that location and play the sound through that speaker.

Phantom Image

When working with a fixed speaker arrangement (e.g., quadraphonic), a sound can be made to appear to come from a location precisely between two speakers by reducing the sound's volume by about 3.0 dB and playing it back from both speakers simultaneously. The listener will hear the sum of both speaker's outputs and will perceive the sound at its original volume and as if it is coming from a phantom source in the center of the two speakers. Using gradation of this technique enables a sound to be positioned anywhere between two speakers.

The problem with employing more sophisticated spatialization techniques like "phantom image" is that proper reception of the spatial audio relies on the listener remaining in the center, or "sweet spot," used to deliver that technique.

Wave Field Synthesis

There is one method for speaker-based sound spatialization that does not require the listener to stay within the "sweet spot." Rather than using psychoacoustic tricks, Wavefield Synthesis (WFS) relies upon a massive array of equally spaced full-range speakers to physically create the desired sound field. While WFS can create a much more realistic spatial sound environment, the number of speakers required makes it an extraordinarily impractical technique that will likely never see any successful consumer-level applications.

Size/Width

Balancing a sound across more than two speakers will make the sound appear to occupy more physical space. Subtle spectral alteration to the sound as it plays through each speaker can enhance this effect but taken too far can lead to localization confusion.

Head shadowing, IID, and ITD

The beautiful thing about working with external delivery systems is that they take care of all the person-specific spectral modification required for sound localization. However, playing with the timing and frequency differences of a sound as it simultaneously transmits from different speakers can create more intriguing and immersive sonic environments at the cost of confused localization.

Spatial Effect Ideas for use with External Delivery Systems

Spatial Filter Sweeps

Distribute a sound across multiple speakers and apply the same filter (lowpass, highpass, or bandpass) to all channels. Then modulate/automate the filter's cutoff frequency either in a synchronized or randomized fashion.

Spatial Tremolo

Distribute a sound across multiple speakers. Then modulate the amplitude of each channel in either a synchronized or randomized rate, somewhere between 0 and ~20 Hz. Synchronizing the rate of the tremolo across all channels and adjusting each channel's phase alignment can result in repeating movements around the space. Asynchronous rates will result in endless variations in spatial movements.

Microtonal Pitch Inflections

Using either fixed or time-varying processing, discretely raise or lower the pitch of a single spatially distributed sound. Small alterations in pitch will thicken the resulting sound and give it a “shimmering” quality. More significant alterations in pitch may cause the listener to no longer perceive them as being a single sound source.

Complex Delay Networks

To achieve a more advanced technique, send a single audio source through a complex network of delay lines with feedback. Send each channel of delayed sound directly to an output speaker as well as back into at least one other delay’s input. This technique works best with single impulse sounds, like the single strike of a piano key or drum hit. This technique can be taken further by inserting additional processing, such as a lowpass filter or reverb, into the feedback network. These effects will repeatedly process the sound as it echoes through the feedback network and around the space.

2.4.2 Personal Delivery Systems

Creating a realistic spatial audio environment inside of personal delivery systems requires additional work because it lacks the luxury of an acoustic environment for the necessary physical interactions to provide localization cues. For example, if a listener is wearing headphones and only the left headphone emits a sound, the listener will undoubtedly be able to identify that their left ear only hears the sound. It will, however, sound unnatural because they will not hear any portion of that sound with their right ear as they would if the same audio file was played back over speakers.

Of course, for creative purposes, this problem could be exploited. Considering what we know about the omnidirectional behavior of low frequencies, receiving completely different low-frequency content in each ear can be very unnerving and anxiety-inducing for most listeners. Cutting the low-frequency content entirely from one ear may disrupt a listener's equilibrium and induce nausea.

The interaural differences that we listen to for clues to a sound's lateral location can be easily created using audio tools bundled with most Digital Audio Workstations (DAWs). They will be discussed below in isolation but combining two or more of these processes can result in a more dramatic spatial effect.

Creating Interaural Intensity Differences

Most DAWs provide the user with either a balance or pan control in every audio track's mixer section. While on the surface these may appear to have the same effect, there are some critical differences.

On a mono audio source, balance and pan both produce the same results and can be used to set the amplitude distribution of a sound across two output (stereo) channels. On a stereo audio source, however, balance adjusts the independent level of the left and right channels of the audio source. If the source audio has different information on the left and the right channels, adjusting the balance control to either extreme will cause the loss of information in the negated channel. A stereo pan control, on the other hand, allows for the independent amplitude distribution of a sound between two output channels. In this case, no stereo information is lost when setting the pan control to either extreme. Panning has the added benefit of being able to control the stereo width. Setting both the left and

the right pan controls to the same value will result in folding the stereo information down to mono and placing it at that location within the stereo field.

Due to the aural separation of personal listening systems, applying only minor changes to amplitude is preserve a more natural spatial sensation.

Creating Interaural Time Differences

Most DAWs come bundled with a variety of delay effects. However, only a few come with audio effects capable of delaying a sound within the same timescales as ITDs ($< \sim 625 \mu\text{s}$). For example, Logic Pro X includes the “Sample Delay” audio effect that enables the user to independently set the delay time for the left and right channels by any number of samples.

If the sample rate of a Logic Pro X project is 48,000 samples per second, then one sample is equal to approximately two microseconds, and thirty samples of delay are equal to approximately 625 microseconds. This audio effect could be used to suggest a lateral location by adjusting either the left or the right delay to ≤ 30 samples.

Head shadowing

Applying a highshelf filter and attenuating frequencies above ~ 1600 Hz on either the left or the right side to will result in an the illusion of head shadowing.

Front, Back, and Elevation

Without tracking the listener’s head movement and allowing them to reorient their ears to check for front/back or elevation cues, the other dimension of spatial audio can be extremely difficult to accurately and universally mimic using the tools included in most DAWs.

In recent years, however, many freely available Ambisonic software plug-ins have been released that simplify the process of applying generic HRTFs and rendering the resulting spectral modification as a binaural audio file. Until there is a way for users to capture their HRIRs, using these specialized tools will not guarantee the accurate sound localization by all listeners.

Distance

Distance is quickly and commonly replicated using any DAW's reverb effect. Start by either selecting or designing the type of space to locate a sound within (cathedral, bathroom, or hallway). Then adjust the reverb effect's pre-delay and dry/wet parameters, as well as the overall level of the resulting sound.

- To achieve a sound that appears to be close in a very reverberant space, increase the pre-delay time and decrease the wet level.
- To achieve a sound that appears further away, decrease the predelay time, increase the wet level, and decrease the dry level.
- To create a more realistic sense of distance, calculating the reverb (the acoustic interaction between a sound and space) should result from the actual space the listener and sounds are within.

2.5 Environment Modeling

Something that is crucial to delivering an immersive and realistic aural experience is placing both the sounds and the audience in an acoustically responsive environment.

Now that computers, gaming consoles, mobile devices, and other content delivery systems have become powerful enough to handle the computational requirements of real-time acoustic-

environment modeling, the consumer demand for more realistic audio experiences has begun to grow. Many new tools have become available that make it easier for content creators to deploy these environmental modeling techniques and meet the demand. Modeling a three-dimensional acoustic environment boils down to three main components: direct sound, indirect sound, and occlusions.

2.5.1 Direct Sound

Direct sound is the sound heard without any transformations or modification applied to it by the space or surrounding objects. The direct sound carries with it the original sound plus all the information needed to detect its direction of arrival. All other iterations of a sound – the slap-back of a snare drum in a small club, an echo from deep within a canyon, the indecipherable roar that fills an excited stadium – is the indirect sound.

2.5.2 Indirect Sound

There are two parts to *indirect sound*: early reflections, and diffusion. Both of these elements are derived from the geometry and material composition of the space and the objects which inhabit it.

The size and shape of the room will determine, first, how long it will take the sound emitted from an internal location to reach each surface, and, second, with what trajectories the sound will reflect off each surface. Then, depending on the construction materials of each surface, the sound will be absorbed and spectrally altered to varying degrees. The harder the material, the

more reflective the surface will be. The softer and more porous the material, the more high-frequency absorption occurs.

Early Reflections

Early reflections can be replicated by first measuring the distance of all the potential paths from a sound source's location to the listener's ears that encounter only one reflection point. Second, by applying to each reflection both the spectral modification that the reflective surface imposes on it and all the necessary alteration to perceive from what direction the reflection is arriving. Finally, delay each reflection by the time it would take for the sound to travel the distance of its path and mix them all with the original (direct) sound.

The human ear is extremely sensitive to early reflections, and they play a considerable part in spatial awareness. Because of this, when a device's computational resources are in high demand by other processes, most of the available processing can be focused on calculating and rendering early reflections and less on a sound's spatial diffusion.

Diffusion

Depending on the material construction of the room, the sound will likely continue to reflect around the environment until its energy has either dissipated or completely diffused. Any space's complex diffusion network could be recreated using an infinite set of delays and filters. However, because diffusion is less critical to spatial awareness, two more generalized processes for fabricating a sound's spatial diffusion can work just as well: convolution or algorithmic reverb.

Convolution Reverb

Using a similar process to capturing HRTFs, convolution reverb begins with capturing an impulse response (IR) of a space by recording the results of either a short burst of white noise (all frequency at equal amplitudes) or a full-spectrum sine tone sweep. This recording can then be analyzed for how all the produced frequencies respond in the space over time. This IR can then be used to digitally simulate the resonance of any audio source produced within the captured space by multiplying (convolving) each sample of the audio source with each sample of the IR. Typically both *time* and *frequency domain* convolution methods are used to render the most detailed and realistic responses. The combination of these two methods require a lot more processing power in contrast to algorithmic reverb techniques, but this level of detail makes convolution reverb particularly well suited to simulating outdoor spaces.

Algorithmic Reverb

Algorithmic reverb, on the other hand, is more computationally efficient, but at the cost of realism. There are various methods for creating an algorithmic reverb, each with its unique characteristics and set of capabilities. Typically, these algorithms involve a network of delays for simulating the early reflections followed by a series of *allpass filters* which feed into a parallel bank of *feedback comb filters* that mix to form the reverb's diffusion. This popular algorithm was developed by Manfred Schroeder and Ben Logan in the early 1960s and is at the heart of most algorithmic reverb effects.

Occlusion and Obstruction

Acoustic occlusion occurs any time the path between a listener and a sound source is completely blocked. Acoustic obstruction is caused by smaller objects that do not completely

block the sound from the listener but are substantial enough to cause spectral modifications to the sound. The most significant difference between occlusion and obstruction is that an object that occludes a sound also occludes the reverberance of that sound, whereas an object that obstructs a sound alters only the direct component of the sound. In most real-world situations, interaction between a sound source and the surrounding environment will be much more complicated, and any object between the listener and a sound will cause occlusion and obstruction to varying degrees.

If, for example, a listener is standing in a large office space with a pillar very close on their right-hand side and a radio playing about 20' in front of them, the pillar will not obstruct the direct sound or the early reflections/diffusion from their left. It will, however, almost wholly occlude all the reflections/diffusion of the sound on their right side. It will also provide some fascinating reflections of both the direct and reflected sound.

Even if the listener and all the sound sources remain stationary, acoustic environment modeling involves monitoring and calculating many factors. Rendering the environment in real time becomes hugely complex if the listener and the sounds are moving. However, when done well this can be the difference between a listener losing themselves in the sonic world or hearing it as a rudimentary attempt at reality.

Along with the renewed interest in virtual reality, acoustic environment modeling is undergoing rapid research and development which has already yielded a wide variety of new tools and techniques. So rather than going into how to implement an acoustic modeling system, the reader is recommended to research what new tools are available and read their respective manuals.

2.6 Sound Design Guidelines for Spatialized Audio

Not all sounds spatialize easily. There are many concerns when designing or composing music and sounds for new sonic territories. The following are some general guidelines to keep in mind in the early stages of spatial audio projects.

2.6.1 Mono

It is easier to spatialize a sound that starts as a single channel (mono) audio source. There are two reasons for this:

- If a sound is already a multichannel audio file, any difference between the channels must be dealt with while also attempting to position it spatially.
- Most spatialization tools treat a sound as if it is coming from an infinitely small point in space and will fold any multichannel audio file down to mono without necessarily notifying the user.

If an audio source needs more than one channel, such as pre-rendered stereo or multichannel delay effects, spatialize it independently by breaking each channel out into a single file.

2.6.2 Complex Timbres

Listening for small interaural differences in the strength and timing of different frequencies is how most sound localization cues are perceived. If the sound has too pure of a timbre, like a sine tone, it can be challenging to locate it or detect movement accurately.

2.6.3 Head-locked vs. Spatialized

Most spatial audio mediums are capable of both head-locked and spatialized audio. Head-locked audio is the same as traditional stereo audio. It moves with the listener as they turn their head rather than remaining in some spatial location.

The term *Head-locked* is interchangeable with *2D Audio*, and the term *Spatialized Audio* is interchangeable with *3D*, *360*, and *Head-Track* audio.

One of the creative challenges of these new formats is choosing what will be spatialized audio and what to lock to the listener's head: The listener's virtual heartbeat, breath, and the narrator's voice might all be head-locked, while the sounds of lasers flashing past and exploding in specific spatial locations remain independent of the listener.

2.6.4 Visual Attachments

Attaching sounds to objects within the experiencer's field of vision can dramatically increase their ability to accurately localize a sound within an audio environment, even if the sonic cues are far from what they would expect to hear coming from that object.

2.6.5 Keep it simple

Not everything needs to be spatialized and moving all the time. Too much spatial activity may overwhelm the listener and cause them to disengage with the experience. Instead, if most things remain in a fixed location and only one or two elements are moving, the listener may be more intrigued.

Chapter 3: Survey of New Digital Formats 2008-2019

Composers looking to explore interactive and immersive digital experiences have few examples to draw on from the 20th century. Isolated endeavors such as Morton Heilig's 1962 *Sensorama* (Turi, 2014) (see Figure 3-1) and Ivan Sutherland's 1968 *Sword of Damocles* (Virtual Reality Society, 2015) (see Figure 3-2) constitute some of the earliest experiments with mixed and virtual realities. Within the last ten years (2008–2018), however, numerous composers have released works that push beyond the familiar formats of musical experiences into uncharted digital territories. Yet studying and creating in these new formats can be challenging for two reasons: the works are often obscure (and thus difficult to find) and experiencing them requires equipment that may not be readily available.



Figure 3-1. Morton Heilig's 1962 *Sensorama*. Image sourced from <http://www.mortonheilig.com/InventorVR.html>

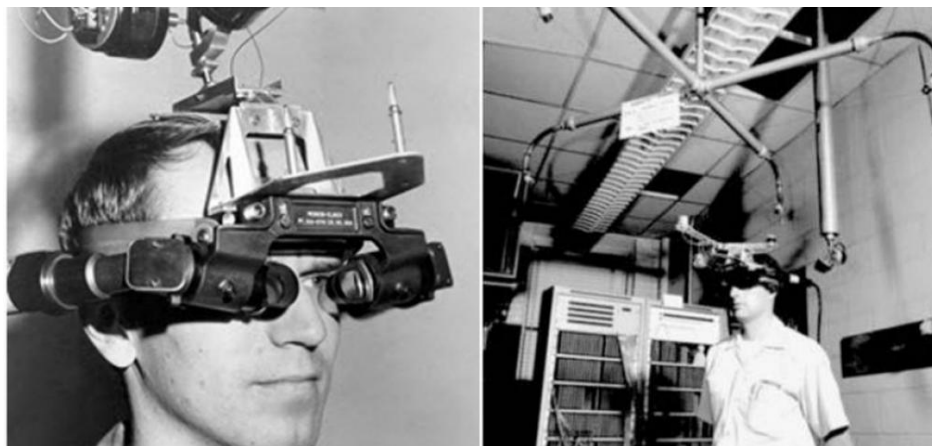


Figure 3-2. Ivan Sutherland's *Sword of Damocles*.

Image sourced from <https://vrroom.buzz/vr-news/guide-vr/sword-damocles-1st-head-mounted-display>

These issues share the same root cause—that is, the ever-increasing rapidity of technological development. Analyzing the number of patents issued by the US Patent Office provides one illustration of how quickly new technologies are emerging (Atlantic Re:Think, 2015). In its first 121 years of operation, the patent office issued one million patents. The next million patents came in only 25 years, and that trend has only continued to accelerate. As of 2018, the office had issued 10 million patents (Patel, 2018). While this indicates a thriving environment for innovation, the haste of technological development has also created an environment where new inventions quickly become incompatible, obsolete, and forgotten. This applies not only to hardware but also, and often more paralyzing, the software running on that hardware. As a result, some of the experiences included in this survey are no longer accessible in their original formats and can only be studied through surviving documentation. Thankfully, manufacturers like Oculus, a prominent manufacturer of VR headsets, have recognized this issue and are working to ensure future hardware and software updates will continue to support the content made today (Constine, Oculus VR Content Will Be Forward-Compatible with ITs Future Headsets, 2018).

The overarching question that guided this survey is, how are these formats different? Understanding the compositional possibilities made available through new digital formats requires examining the works collected here for how they align with or break from historical music formats? What qualities do they have in common with previously available musical experiences? And perhaps most important, do they offer anything new?

Each of these new formats claims to be different by positioning itself either partially or entirely in a new reality. In researching and experiencing these new realities, however, it became clear that both the works themselves and the developers of the software and hardware that enable these new realities seem to be investing most of their efforts in recreating our current realities. This raises several questions. Of the works included here, what aspects of our reality do they successfully copy? What elements do they leave behind or fail to adapt faithfully? Finally, do they push beyond the limits of our reality in any way? This survey will not concern itself with assigning value to these differences, only with cataloging them. Portions of the final chapter will reflect on these types of inquiries.

The final and most relevant element this survey seeks to uncover is how each of these formats changes the way music can be engaged. Ultimately, this document aims to uncover what composers can do within modern musical formats that has never been possible before.

3.1 Established Formats

Over the last ten years, several new formats have already amassed enough representative works to be considered established. Yet these formats did not appear spontaneously; rather, their lineage can be traced back nearly half a century to the rise of personal computers and the internet.

3.1.1 Massively Multiplayer Online Games

Created in the 1970s by students at MIT and the University of Essex (Bartle, 2016), massively multiplayer online games (MMO) represent some of the first alternative virtual worlds for human interaction and expression. As the internet sprouted and laid roots that inevitably connected the globe, MMOs grew along with it. Eventually, networks and players included more than just the computer science departments at universities, and people all over the world began taking up residence in every corner of these sprawling digital universes.

MMOs have spawned many subcategories, such as massively multiplayer online role-playing games (MMORPGs), massively multiplayer online bulletin board games (MMOBGs), massively multiplayer online first-person shooter games (MMOFPSs), and massively multiplayer online real-time strategy games (MMORTSs). These online games often foster rich internal cultures that even lend themselves to musical events. One prime example is *Second Life*, an MMORPG that has supported performances by the likes of Pauline Oliveros, U2 (see Figure 3-3), Duran Duran, and Suzanne Vega (Linden Lab, 2001).

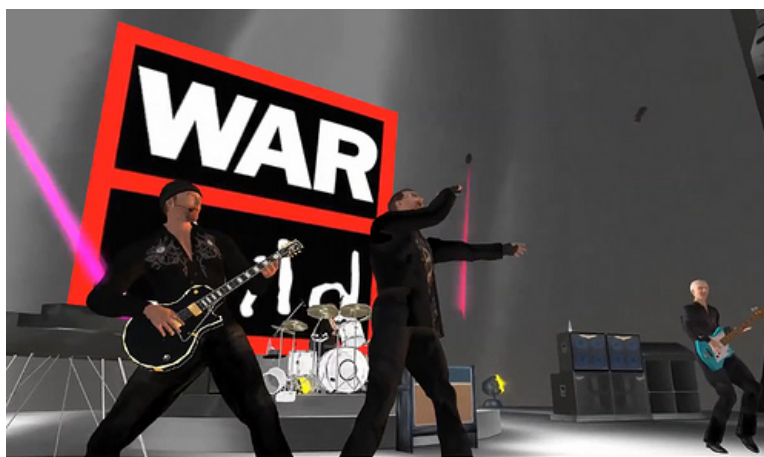


Figure 3-3. An image showing U2 performing in *Second Life*, 2009.

Image sourced from <http://www.secondlifeupdate.com/news-and-stuff/3d-concert-hd-video-virtual-u2-music-to-benefit-war-child-charity/>

Linden Lab: *Second Life*

Second Life is a free virtual world created by Linden Lab and first opened to the public in 2003. Players, known as “residents,” form active and industries communities, and the ability to exchange an in-game currency, called Linden Dollars, has in effect fostered a thriving virtual economy. In 2015, the economy of *Second Life* had an estimated gross domestic product (GDP) of \$500 million, with resident earnings averaging around \$60 million (Frankenfield, 2018). Similarly, the artistic expression of *Second Life*'s residents was just as productive.

Residents of *Second Life* are empowered by features like an internal scripting language, LSL (Linden Lab, 2007), which allows them to create and customize their worlds and digital selves however they want. From an audio perspective, *Second Life* is relatively limited and supports only playback or live streaming of mono/stereo audio files. Despite these simple capabilities, however, many musicians and composers still have found this new world inspiring.

In 2007, composers Hars Hefferman and Maximillian Nakamura founded the Avatar Orchestra Metaverse (AOM), an ensemble that performs exclusively in *Second Life* with members from all over the world. During each performance, AOM members stream audio separately from their homes to a predetermined space in *Second Life* where custom 3D visual and choreographed movements usually accompany the musical performance. AOM's website provides a full list of the group's commissioned works, performances, and members (Avatar Orchestra Metaverse, 2003).



Figure 3-4. Image capture of the Avatar Orchestra Metaverse rehearsing.
Image sourced from <http://www.avatarorchestra.org/how.html>

As part of the 2008 Voice++ Festival, Pauline Oliveros, who regularly explored new formats of musical expression throughout her five-decade-long career (Oliveros, 1995), premiered an ensemble adaptation of her composition “The Heart Tones” with AOM. The performance incorporated live-streamed trombones and voices with virtual instruments inside of *Second Life* (see Figure 3-5 and Figure 3-6).



Figure 3-5. Early rehearsal of Pauline Oliveros' “The Heart Tones” with the Avatar Orchestra Metaverse in *Second Life*, 2007.
Pictures sourced from <http://avatarorchestra.blogspot.com/p/avatar-orchestra-compositions-and.html>

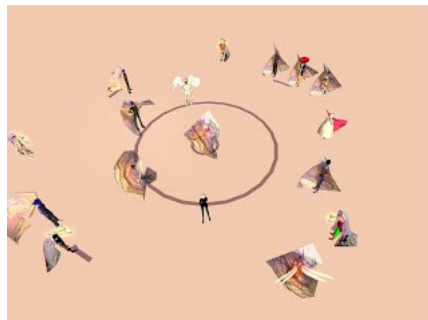


Figure 3-6. Premier performance of Pauline Oliveros' 'The Heart Tones' with the Avatar Orchestra Metaverse in *Second Life*, 2008. Picture sourced from <http://avatarorchestra.blogspot.com/p/avatar-orchestra-compositions-and.html>

Because the game positions itself as a virtual world akin to that of real life, *Second Life* attempts to recreate many elements of reality. When comparing the performance by AOM and other *Second Life* musicians to the history of real-world performance, there is little difference between a concert in *Second Life* and a concert in the physical world. The one aspect *Second Life* leaves behind, however, is what has made the game so popular, as well as what differentiates its musical possibilities the most from other realities: *Second Life* leaves behind the person's body and the identity carried with it.

In *Second Life*, people's avatars can be anything they want them to be. Players can opt to mirror their corporeal self or present themselves as something as unusual as a colorful mist. Considering the significant influence that a musician's body can have—not only on their commercial success or the way their music may be received but also the access that their physical body affords them—the option for artists in *Second Life* to completely modify and customize their virtual bodies and identities enables a substantial break from reality. Ignoring the technological knowledge and financial barriers to immersing oneself in this alternate reality, all members of *Second Life*, once admitted, become unchained from most of the limitations that their physical bodies have imposed upon them, allowing them to move freely in this new reality however they choose.

3.1.2 iOS Instruments

An iOS instrument is a mobile application designed to transform an iPhone, iPad, or iPod Touch into a musical instrument. Many iOS instruments are modeled after acoustic instruments, while others push into more unfamiliar territories. Instruments designed for other operating systems outside the Apple ecosystem exist, but iOS devices represent the most popular and fertile arena for instrument development (White, 2018).

With Apple's 6s model, iPhones began supporting a full range of modes for expressive inputs, including multi-touch, pressure sensitivity (like the aftertouch feature of some MIDI keyboards), audio input/output, and gyroscopic monitoring. The more recent iPhone models include facial recognition and tracking capabilities, the expressive possibilities of which have only been marginally investigated. Projects such as Chase Farmer's *ConductorAR* (Farmer, 2017) and his accompanying tutorial (Farmer, Making your face the effects automator with an iPhoneX, Unity, Max and Ableton, 2018) show the basics of building applications that can take advantage of face tracking functionality. Apple even saw the potential of this technology. In March 2018, the company gave users the ability to control effect parameters within the iOS version of GarageBand using their mouths (Deahl, 2018). But the technological and expressive possibilities iOS instruments afford are perhaps not their most notable characteristics. The two most significant areas of growth that iOS instruments have enabled for musical experiences have been establishing global connections by allowing people to collaborate across physical boundaries and diminishing technical and financial barriers to creating music.

Smule: *Ocarina*

Smule, a software developer that specializes in collaborative music-making applications for mobile, (Smule, 2001) released *Ocarina* (Smule, 2010), its first iOS instrument, on Apple's App Store on November 1, 2008. The main feature of *Ocarina* enables users to replicate the sound and simple mechanisms of the ancient flute-like instrument of the same name using simple touch features (see Figure 3-7). The most significant impact of this app came not from its instrumental capabilities but instead from its social features. By tapping the in-app globe icon, users could view an interactive map that displayed the approximate location of anyone in the world who was currently playing the app (see Figure 3-8.). Users can then navigate the world and listen to any performance in real time. Building upon the success of this feature, Smule has since released a suite of apps that all focus on how making music through personal devices can bring people together.

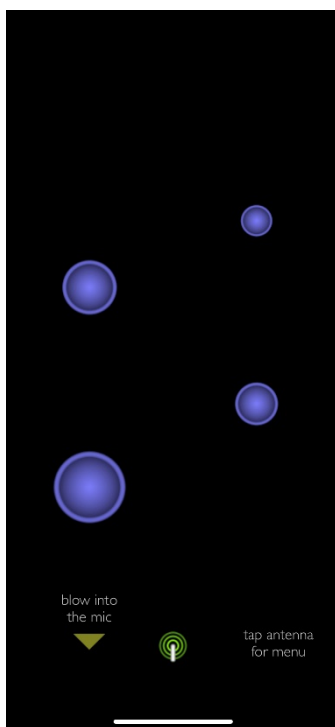


Figure 3-7. In-app screen capture of *Ocarina*'s main instrument controls.

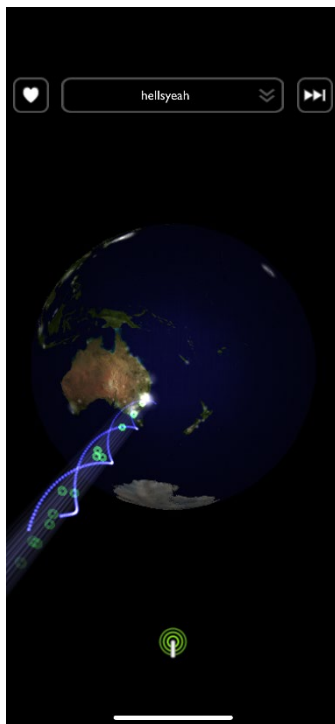


Figure 3-8. In-app screen capture showing *Ocarina's* global view and the notes of a live performance streaming in real-time.

Herrmutt Lobby: *Playground*

Playground, released in 2016 by a collective of artists that go by the name Herrmutt Lobby (Herrmutt Lobby, 2010), is a polyphonic gesture-based iOS instrument that defies a singular definition or function.

While this app can indeed be called an instrument, it could also be considered a score and an interactive environment for musical composition and performance. Recognizing the potential of the app's ambiguity, the creators have already commissioned nearly thirty artists to create original works that use *Playground* while also helping refine the capabilities of the format. A complete list of the artists can be found on the app's website (Herrmutt Lobby, 2015).

When players open *Playground* for the first time, they see a walkthrough guide that explains the basic concepts of how to interact and perform the musical works available within the

Playground format (see Figure 3-9). The app does not permit users to advance to the next stage of the orientation until they have successfully performed the current gesture. With each stage of the orientation, the patterns become more complex (see Figure 3-10) and users discover more sophisticated objects with unique functions, behaviors, and modes of interaction (see Figure 3-11).

Once they have completed the full orientation, users arrive at *Playground's* catalog of works, where they can choose from any of the free pieces or purchase additional music created by guest artists.

From a compositional perspective, the format of *Playground* is compelling because of how it merges the idea of score and instrument design. This design process begins with considering the layout, object responses, and interactive qualities or limitation.

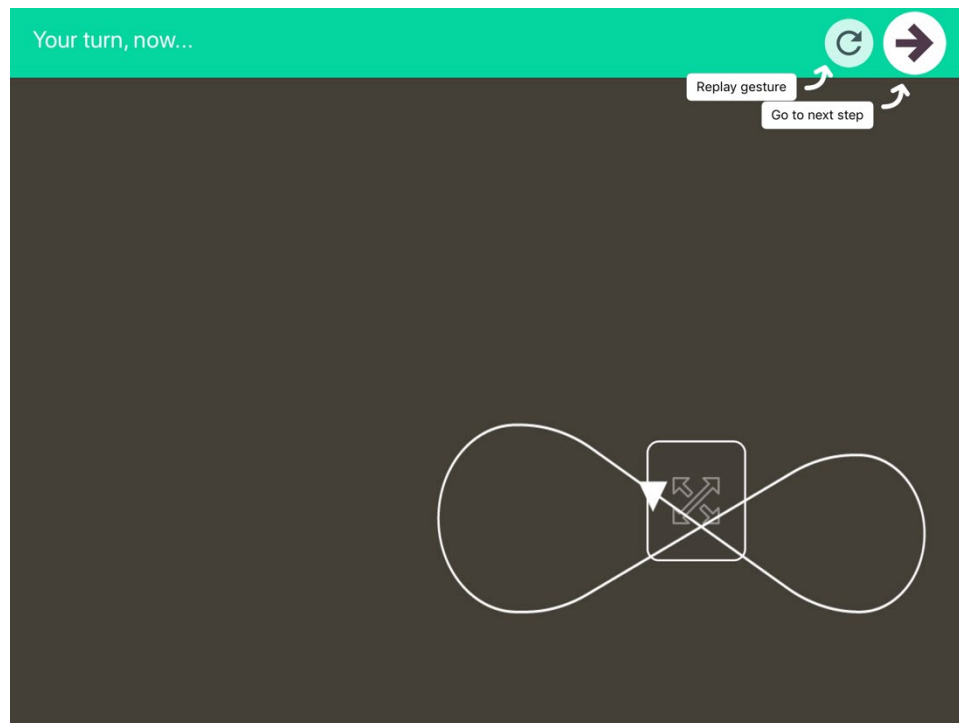


Figure 3-9. In-app screen capture of *Playground's* first user orientation stage: learning the basic gestures.

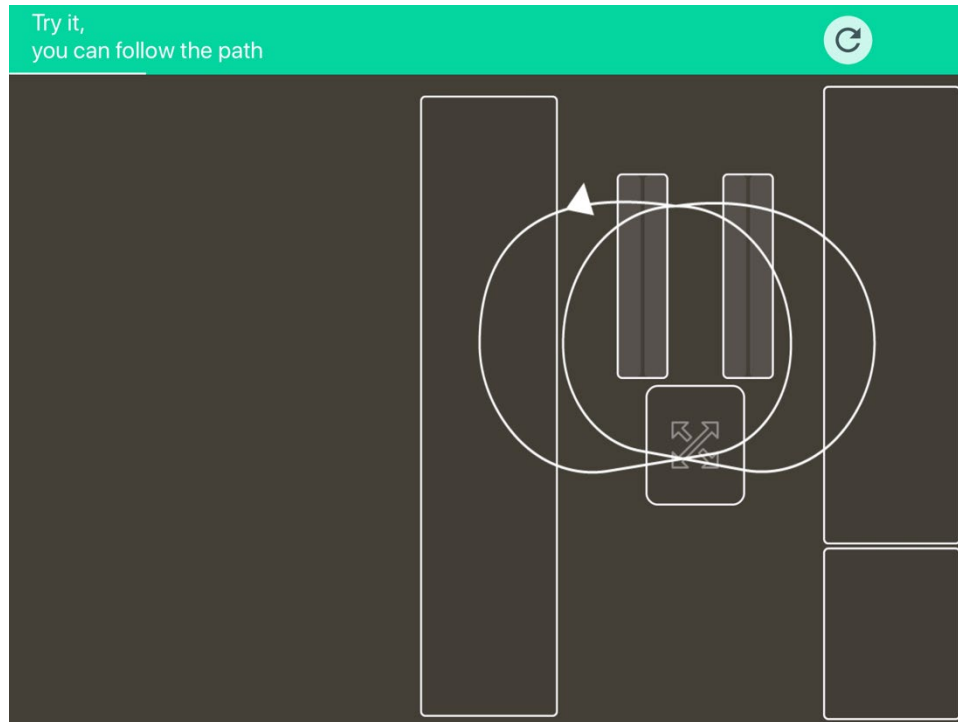


Figure 3-10. In-app screen capture from the second stage of *Playground's* user orientation: learning more complex gestures.

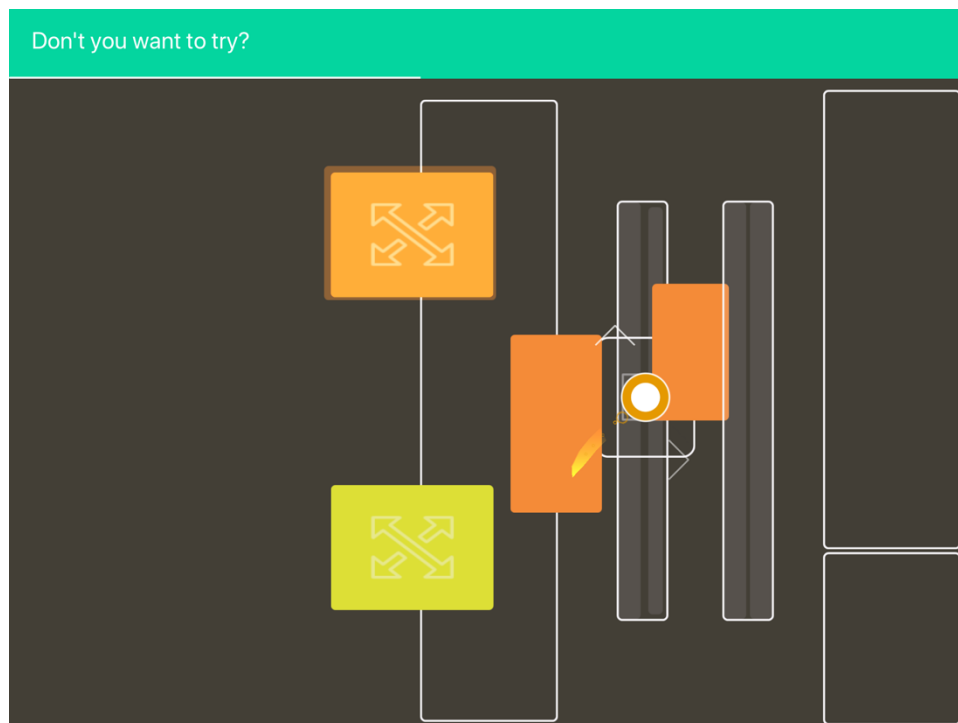


Figure 3-11. In-app screen capture from the final stage of *Playground's* user orientation: learning object interaction possibilities.

While user interaction feels smooth and free, the musical events that result from their interactions are quantized to musically relevant rhythms. The composer's spatial organization of the sounding objects in relationship to one another encourages or directs users toward specific physical gestures that result in repeatable musical motives and phrases. For formal control, the composer can use objects, often located near the edges of the playable space and isolated from the rest of the screen, to cause dramatic shifts in both the objects that appear and the resulting sounds.

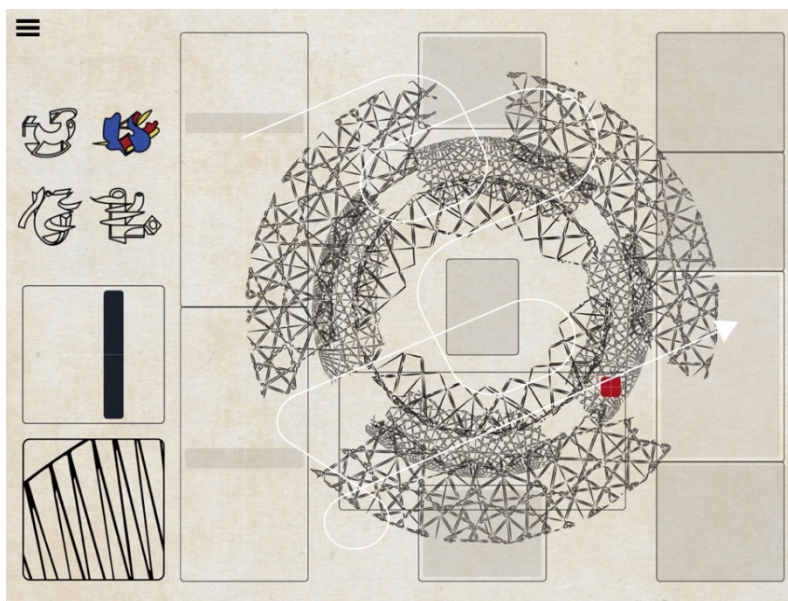


Figure 3-12. In-app screen capture from "Naga Viper Breaks," a free playground within the Herrmutt Lobby *Playground* app.

One example of these elements in action is in the free “Naga Viper Breaks” playground by Herrmutt Lobby (see Figure 3-12). The left side of the screen is devoted to formal and timbral controls. The four symbols near the top left of the screen switch between four different configurations. The square containing the zigzag pattern in the bottom left corner triggers minor layout and sound changes, while the square just above controls the amount of reverb and overall

sustain of the triggered sounds. The remainder of the screen can be considered the instrument or the performance space.

A central feature of this playground is a wheel-like structure that provides the sonic and rhythmic backbone. As soon as any part of the screen is touched, the wheel begins to spin clockwise. Placing a finger anywhere within the spokes of the wheel produces either a hi-hat or cowbell sound. Placing additional fingers at various locations within the spokes of the wheel create more complex patterns. Moving around the wheel counter to the direction of spin increases the complexity and density of the generated pattern. Moving in sync with the wheel has the opposite effect and reduces the pattern's density.

The bass, kick, snare, and clap patterns are articulated using the objects to the left and right of the wheel (see Figure 3-12). Organizing the kick and the bass on the left and the snare and clap sounds on the right inspires more typical drum patterns by keeping the two sound groups on alternating beats as the user travels around the wheel.

Following the same organizational concept, objects that trigger single plucked pitches are located behind and in the center of the wheel. As the user swirls around the wheel, these objects produce arpeggiated patterns that work in counterpoint with the hi-hats and fill the space between kick and snare drum hits.

From a listener's perspective, the lines between audience and performer are substantially blurred. The composer has significant control over the design of the musical playground, but ultimately someone must interact with it to produce any sound. Playground's level of expressive freedom and complexity of potential interaction propels users beyond merely listening to a performance.

Playground manages, through its simple introductory tutorial and playful design, to bypass traditional paths from consuming music to performing it. Within a few minutes of opening the app, anyone could use the Playground app as an instrument in a live performance, as one of Herrmutt Lobby's advertisements suggests (Herrmutt Lobby, 2016), or one could use it in a private setting to explore the sounds and music of the artists featured in the app.

Moog: Model 15

In addition to creating new methods and formats for composition, iOS instruments are also lowering barriers to musical tools and instruments that historically have been inaccessible for geographic or financial reasons. One example of this is Moog's release in 2016 of its Model 15 synthesizer as an iOS app (see Figure 3-13). This recreation costs only \$29.99, whereas the original retails for around \$10,000.

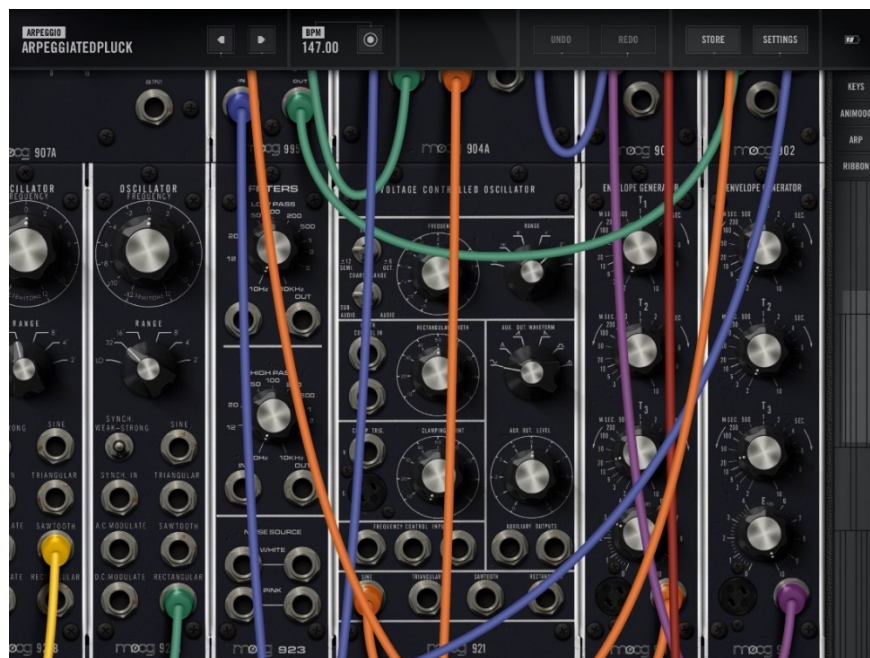


Figure 3-13. In-app screen capture of Moog's *Model 15* app showing the home user interface complete with patch cables and rusty 1/4" jacks.

The Model 15 app recreates each of the modules of the original system and adds two styles of on-screen keyboards: an analog-style arpeggiator and a ribbon controller. In order to make patching such an oversized synthesizer feasible on smaller screens, the app permits the use of two-finger gestures to zoom in and out and to navigate from module to module. And unlike an analog synthesizer, this one lets users save and recall as many patches as they'd like.

3.1.3 Album Apps

Album apps represent the digital evolution of richly colored trifold and box-set LP records. Album apps can give listeners a more interactive listening experience and features that no other album format has offered before. Unlike their historical analogs, however, album apps suffer from the planned obsolescence of their delivery software and hardware. Of the album apps in this survey, which were all released between 2008 and 2018, only a select few are still available. Unless the album app evolves as a standardized or more future-proof file format, artists will have to continually devote time, energy, and money toward maintaining their albums.

In a 2013 *Forbes* article, musician and producer Bobby Owsinski makes a series of arguments against album apps, suggesting that the format is inconvenient, requires too much development time to keep up with the rest of the industry, raises data privacy concerns, and ignores other music industry trends (Owsinski, 2013). Ultimately, he argues, album apps are doomed to become submerged in a sea of other non-musical apps. Owsinski concludes by suggesting that no one needs an album app. Those interested in making sound business decisions may want to carefully consider these critiques. Yet if one sets aside the economic considerations, the album app format still presents possibilities for musical experiences that are not achievable through other means.

Jay-Z: *Magna Carta* - 2013

An early and particularly contentious album app, *Magna Carta*, released by Jay-Z in 2013, challenged accepted standards and highlighted some of the marketing potential of the album app format. In a deal struck between Jay-Z and Samsung (Singleton, 2017), the maker of Galaxy smartphones agreed to purchase and distribute one million copies of the *Magna Carta* album to its users' devices as an app. While this created consumer and industry backlash, the deal also secured the album's platinum status before it was even officially released.

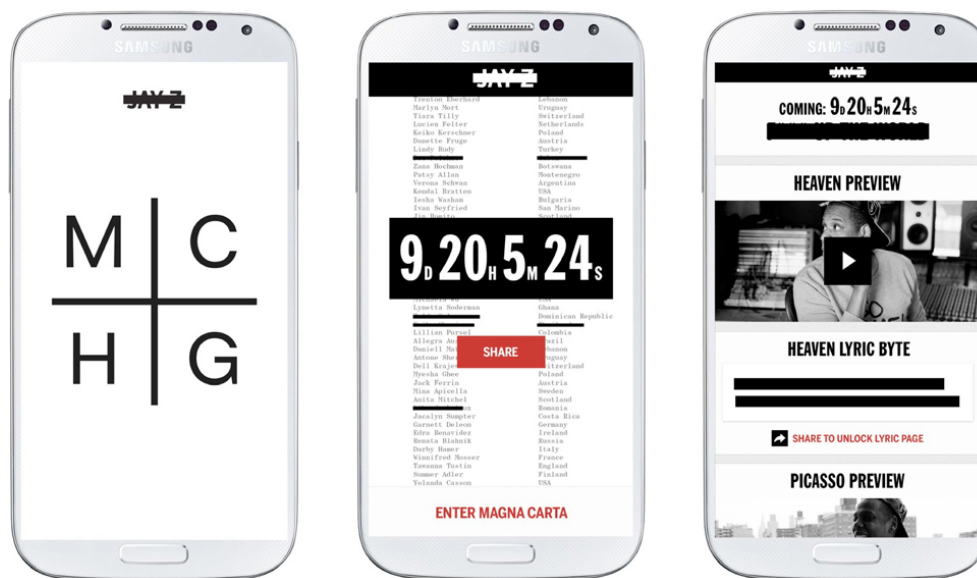


Figure 3-14. Jay-Z *Magna Carta* album app.
Image sourced from <https://allihayes.com/Magna-Carta-Holy-Grail-Launch>

The primary function of the app was to provide Samsung users with early access to Jay-Z's album. But Samsung also required users to make public posts to their social media account via the app in order to unlock songs and lyrics. This move proved effective from a marketing perspective, instantaneously forming an army of one million social media marketers. Yet many users felt uncomfortable with the amount of personal data and access that the app required to enjoy the music, as they received minimal rewards for promoting the album (Pareles, 2013).

Björk: *Biophilia* - 2011

Several years before Jay-Z's *Magna Carta*, Icelandic singer and composer Björk released a companion iPad experience for her 2011 album *Biophilia*. The app represents what is likely the first ever album app. Somewhat comically, the designer of the *Biophilia* Album App, Scott Snibbe, described his previous interactive apps in a talk he gave at the 2011 PICNIC Festival as “useless programs” (PICINIC, 2011), echoing Owsinski's assertion that no one needs an album app. Later in that same presentation, though, Snibbe expresses his and Björk's shared enthusiasm for the iPad and similar devices, contending that the artistic formats and experiences they make possible can help bring audiences closer to the meanings that artist are attempting to convey.

In a 2011 interview with Pitchfork (Dombal, 2011), Björk said her initial inspiration for the album was to take over a home somewhere in Iceland and turn each room into a song. From there her ideas ballooned to increasingly intricate levels—a giant pendulum bass instrument, virus cells singing in harmony, and a solar system suspended in an unseen choir. But after recognizing the impracticality of these ideas, Björk decided on the album app as the perfect format for realizing this concept and for spreading her curiosity.

David Attenborough, who had just finished writing and presenting the BBC's “Life” documentary series a few years prior, served as the narrator for the album app's introductory sequence (Björk, 2011). His voice functions perfectly for bringing the same wonder and amazement he evoked in the “Life” series to the opening scene of Björk's *Biophilia*.



Figure 3-15. In-App screen capture: Opening view of Björk's *Biophilia* Album App.

Upon launching the app, Attenborough's voice guides the user around the outskirts of Björk's universe and finally places them at the edge of a tangled and slowly spinning wireframed solar system (see Figure 3-15.) with a howling choir performing something evocative of a descending Shepard tone. The experience of transitioning from the serene Attenborough introduction to the unnerving and distant view of Björk's galaxy recalls reading about the mysteries of space and then startlingly feeling alone and exposed deep within a vast and unknowable universe.

Unlike Herrmutt Lobby's *Playground*, *Biophilia* provides no new user orientation, only subtle suggestion of ways to explore its universe. With a little experimentation, users will likely discover that their perspective on the solar system shifts in response to tilting the phone or can be spun by dragging a single finger across the screen. Using two fingers, the user can move the solar system left or right or zoom in and out by pulling their fingers together or apart.

Flying past the bright planets at light speed, flashes of new sounds can be heard emanating from the celestial spheres. This is the first example, of the two formats discussed so far, that uses both head-locked and spatialized audio. Here, the howling choir is head-locked and remains unaffected by the user's current perspective and spatial relationships. In contrast, the sounds

projecting from the planets are spatialized, their volume is relative to their distance, and their balance is dependent on where the planet appears from left to right. If users orbit closer to one of the planets, they can tap on it and get sucked inside.

Within each planet is a set of experiences designed around a single song. At the app's initial release, however, not all planets contained song experiences. These were added in one by one, building into the release of a traditionally formatted full-length album. Later, Björk even released a brief update of the "Solstice" song experience to celebrate the winter solstice (Nonesuch, 2012).

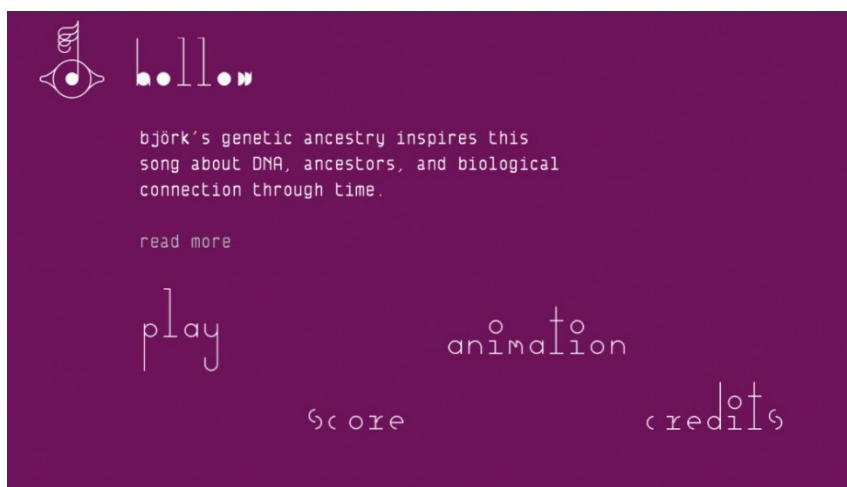


Figure 3-16. In-app screen capture of the "Hallow" song landing page within Björk's *Biophilia* album app.

Each song experience comprises four elements (see Figure 3-16): A text screen that usually contains three short readings, a poetic blurb from Björk on the song, a description of the app experience, and a musical analysis by Nikki Dibben (Flood, 2012).

Play (see Figures 21.0–3-21): A multi-featured animated and sometimes interactive version of the song that further explores core concepts from the song's conception.

Animate (see Figure 3-22): An animated playback of the song that looks similar to a graphic MIDI sequencer that includes text.

Score (see Figure 3-23): An interactive score that uses traditional notation and allows users to follow along while a simplified version of the song is played.

Credits (see Figure 3-24): A listing of everyone responsible for creating the song experience.

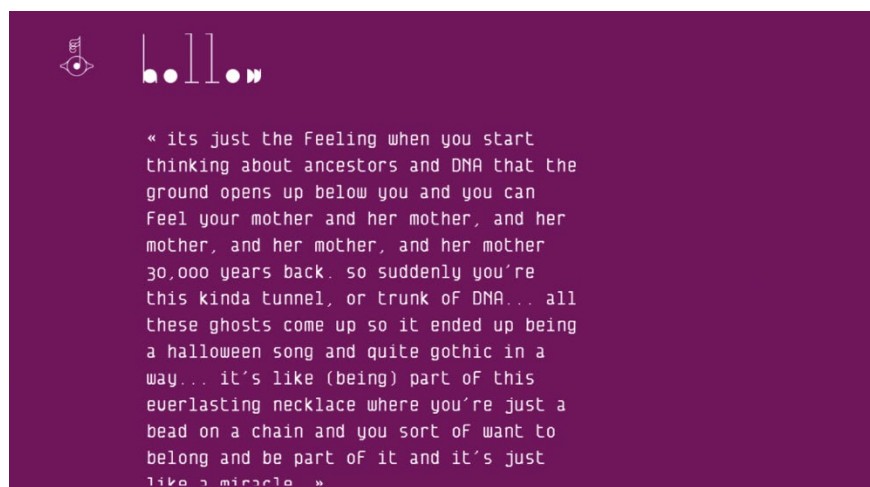


Figure 3-17. In-app screen capture of Björk's *Biophilia*—poetic statement by Björk regarding the inspiration behind the song "Hallow."

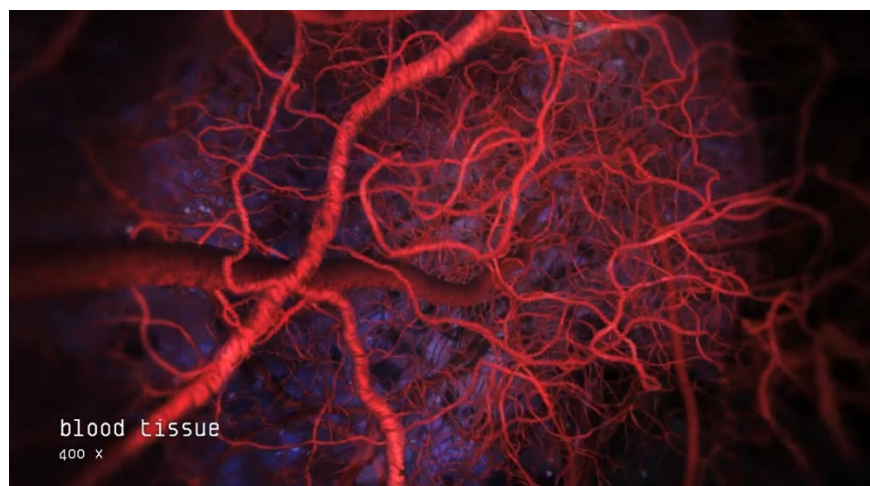


Figure 3-18. In-app screen capture of Björk's *Biophilia*—a still frame from the Play app for the song "Hallow."



Figure 3-19. In-app screen capture of Björk's *Biophilia*—a still frame from the Play app for the song "Hallow."

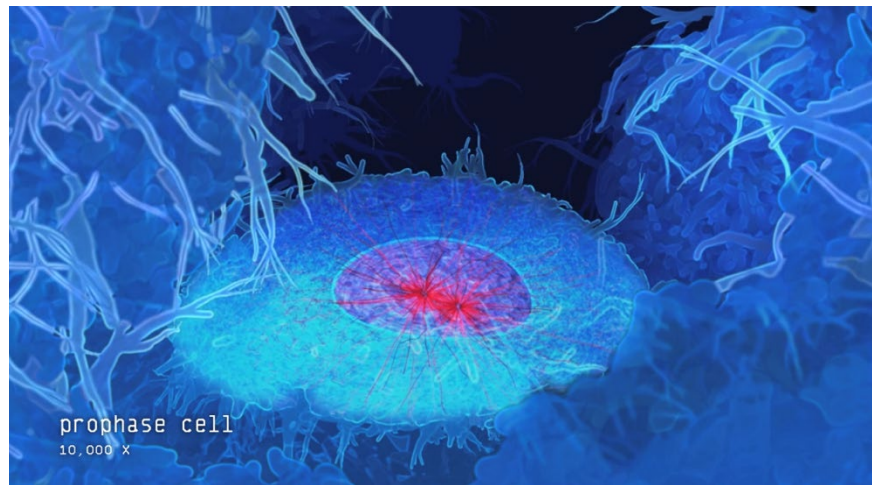


Figure 3-20. In-app screen capture of Björk's *Biophilia*—a still frame from the Play app for the song "Hallow."

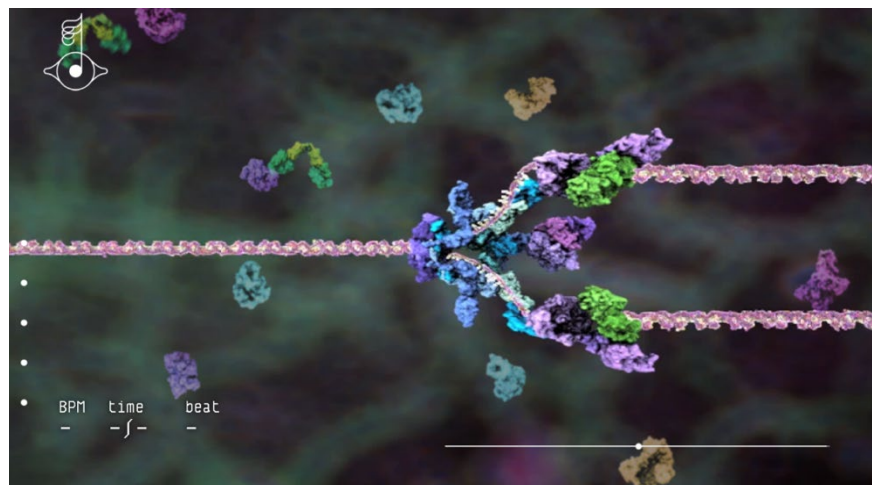


Figure 3-21. In-app screen capture of Björk's *Biophilia*—a still frame from the Hallow Machine app for the song "Hallow" where users can use enzymes and a "DNA Replicator" to sequence musical patterns.

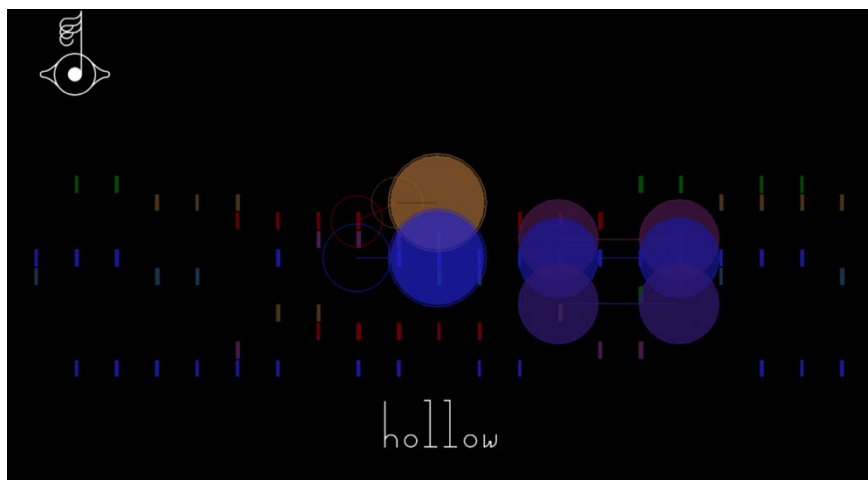


Figure 3-22. In-app screen capture of Björk's *Biophilia*—a still frame from the Animate app for the song "Hallow."

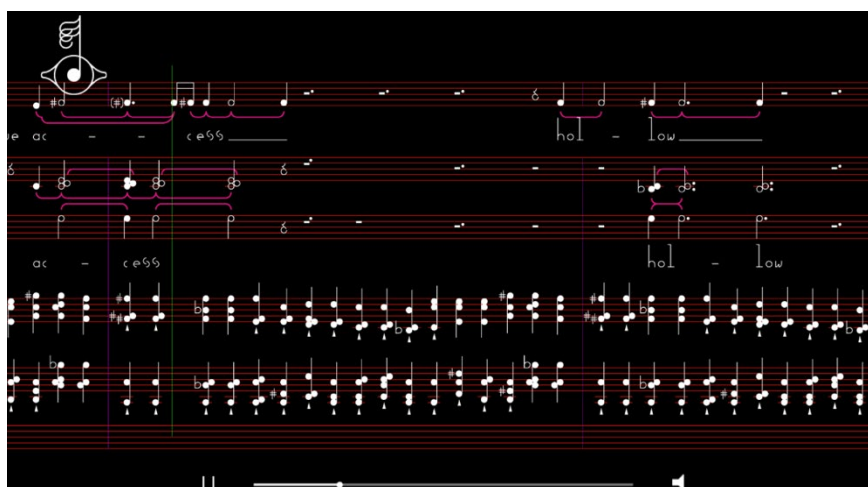


Figure 3-23. In-app screen capture of Björk's *Biophilia*—a still frame from the Score app for the song "Hallow."

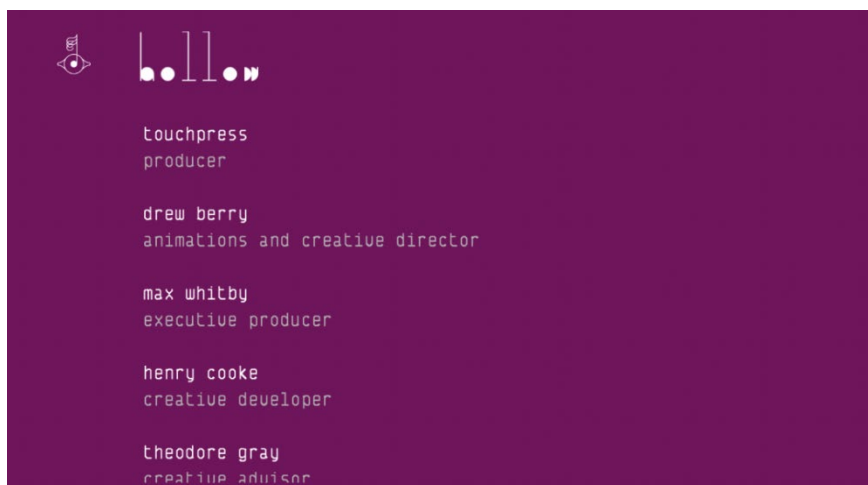


Figure 3-24. In-app screen capture of Björk's *Biophilia*—a still frame from the Credits app for the song "Hallow."

In contrast to Owsinski's dismissal of the format (Owsinski, 2013), *Biophilia* is the perfect example of how aspects that people love about liner notes and cover art can become augmented—that is, made more engaging and more meaningful—through the album app format.

Brian Eno & Peter Chilvers

In 2008, only a few months after Apple's App Store launched, Brian Eno and music technologist Peter Chilvers began releasing a series of generative music iOS applications. For Eno—a composer known for using algorithms to generate his music (Sterne, 2012)—the album app represented the opportunity to break free of the time constraints and fixed forms imposed by traditional formats (Sherburne, 2017).

Bloom – 2008

Bloom, the first in the series, introduced the template and fundamental concepts that Eno and Chilvers would use throughout all their apps. Upon opening the currently available version of *Bloom*, which has remained essentially unchanged since its initial release (markNine, 2008), users must first select between three basic modes (see Figure 3-25): classic, infinite, and freestyle.

The app describes the three modes as follows:

Classic: Classic mode recreates the sights, sounds, and smells of the original version of Bloom.

Infinite: In infinite mode, patterns evolve endlessly, and settings are shuffled during pauses.

Freestyle: Freestyle mode allows control over all parameters.

After choosing a mode, users must choose either to listen or to create before any sound begins. Due to the music's algorithmic nature, there seems to be little difference between choosing to listen or create. Both options take users to a softly colored screen (see Figure 3-26) and music composed of dynamic background textures and arrhythmic melodies.

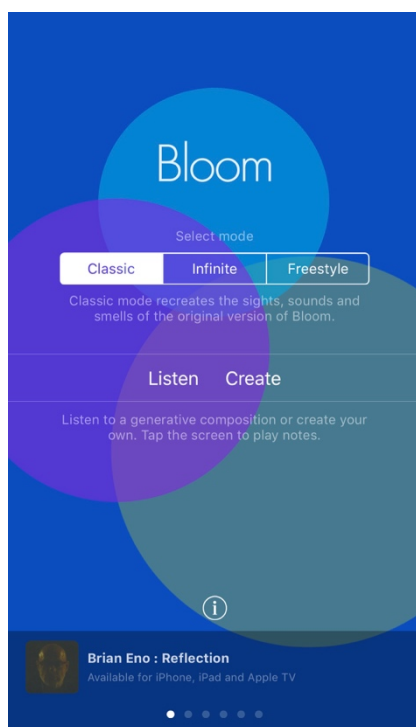


Figure 3-25. In-app screen capture of the *Bloom* app's home screen.



Figure 3-26. In-app screen capture of the *Bloom* app's "Listen" screen.

This view (see Figure 3-25) represents the main screen of the app, where the user can either sit back and enjoy the sights and sounds as they unfold or take a more hands-on approach to guiding the overall soundscape and musical development.

Tapping anywhere except the bottom right corner, the user can seed the algorithm with their melodies. Each note is represented by an expanding circle where the x-axis both

shows and controls the stereo placement of the notes from left to right, and the y-axis shows and controls the relative pitch of the note from low to high (see Figure 3-27).

Tapping on the small icon in the lower right corner exposes the setting and controls for tailoring the algorithmic results (see Figure 3-28). Here the user can choose from any of the essential oil-inspired musical scales (e.g., Neroli, Vetiver, Ylang, Bergamot), randomize all the settings by pressing “listen,” clear the current melody, freeze the algorithm to prevent the sounds from being further developed, and pause all sounds. The delay slider, the range of which varies from only once when set all the way to the left and nearly infinite when set all the way to the right, adjusts how many times the user-played melodic pattern will repeat. Finally, users can choose between three similar sounds for the melody: impact, blend, and bowl.

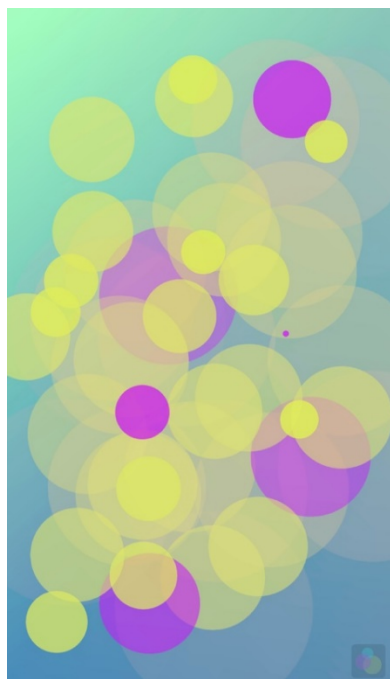


Figure 3-27. In-app screen capture of melody creation in *Bloom*. The x-axis controls the stereo placement and the y-axis controls the relative pitch of each note.

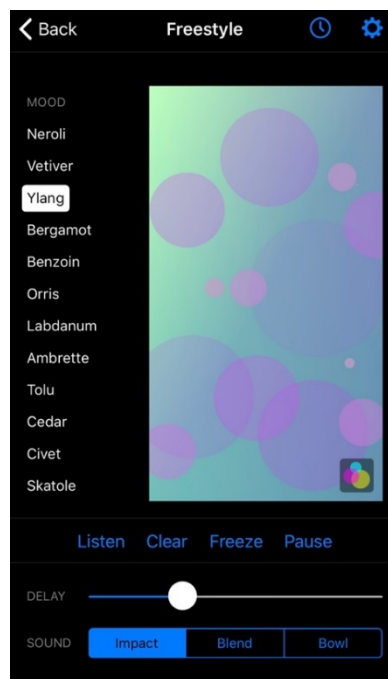


Figure 3-28. In-app screen capture of available settings to tailor *Bloom's* algorithmic results.

Users also can activate a sleep timer using the clock icon near the top right corner, along with several additional settings for defining the app’s general behaviors, such as enabling audio to play when using other apps on the device (see Figure 3-29).

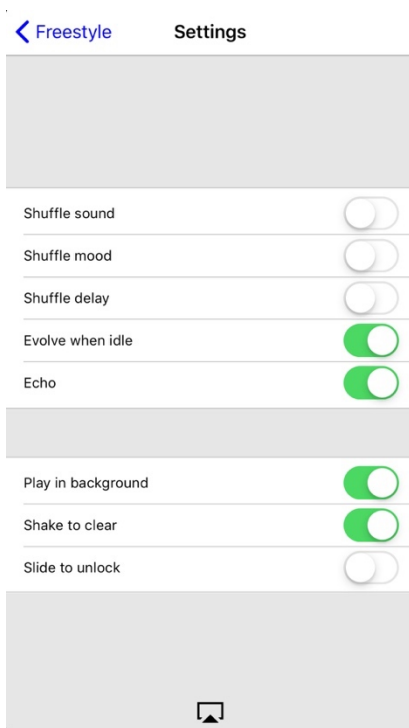


Figure 3-29. In-app screen capture showing the general settings for the "Bloom" app.

Air and Trop – 2009

A year after the debut of *Bloom*, Eno and Chilvers released two more generative music apps that followed in the same vein but that each explored two different interactive methods for creating music.

In *Air*, the screen for either listening to or, as the app calls it, “conduct[ing]” music is divided into twelve triangles (see Figure 3-31). Tapping any of the triangles triggers two notes of either a piano or voice sample (Sandra O’Neill) as selected on the app’s home screen (see Figure 3-30).

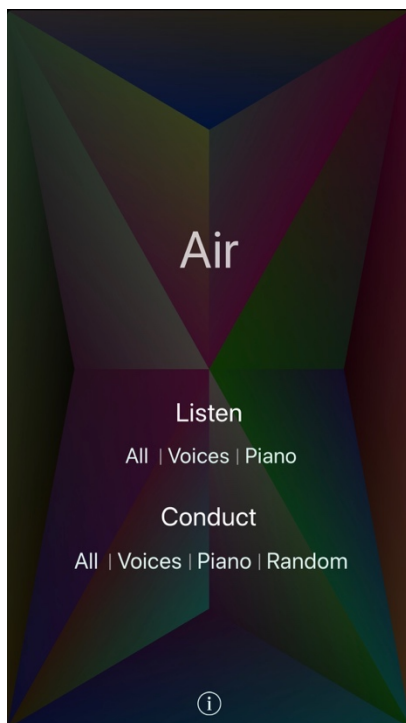


Figure 3-30. In-app screen capture showing the *Air* app's home screen.

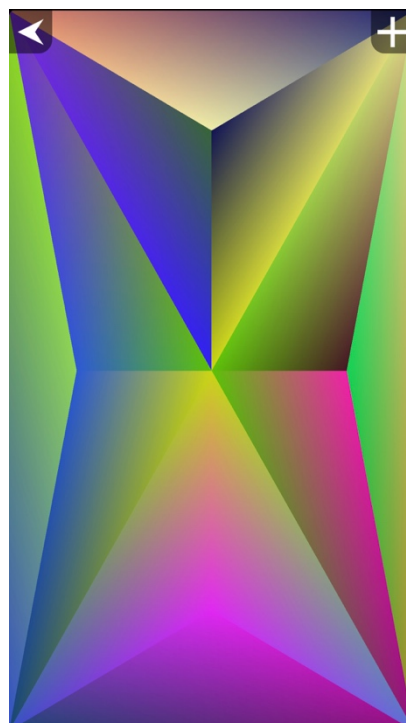


Figure 3-31. In-app screen capture of the *Air* app's "Listen" / "Conduct" page.

The *Trop* app offers an alternative method for creating music (see Figure 3-32) that is similar to the graphical transcription of György Ligeti's "Artikulations" by Rainer Wehinger in 1970 (Haus, Autumn, 1983). Like the original *Bloom* app, pitch and balance are controlled via the x- and y-axes and the user can tap the icon in the lower right corner of the screen to select different bacteria-inspired "moods," as well as choose between five shapes with which to paint their musical scores (see Figure 3-33).



Figure 3-32. In-app screen capture showing an example of the graphical notation used in the *Trop* app.

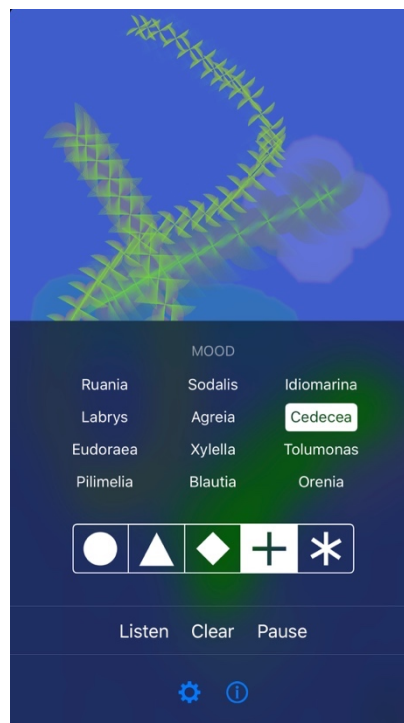


Figure 3-33. In-app screen capture showing the available setting and shapes for creating music with the *Trop* app.

A feature of *Air*, probably overlooked by most users, modifies the density of events produced by the app's algorithm based upon the number of devices currently running the app (see Figure 3-34). This is an early, low-tech example of this kind of feature. More recent iOS instrument and apps use a protocol developed by Ableton called Link, which enables apps to become aware of other Link-enabled apps on the local network and synchronize in various ways, such as indicating how many devices are producing sound (Goltz, 2018).

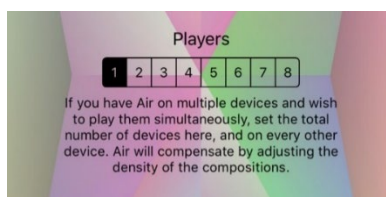


Figure 3-34. In-app screen capture showing the *Air* app's additional setting.

REFLECTION – 2017

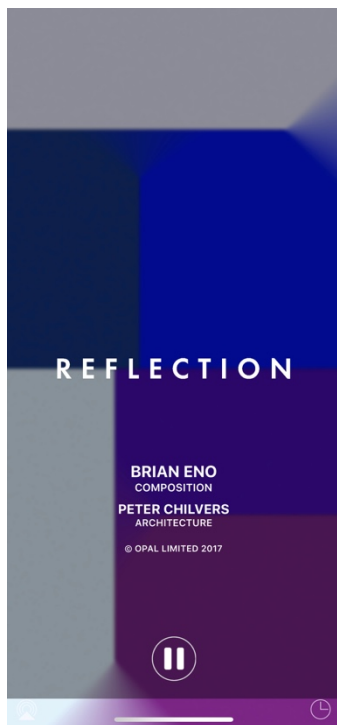


Figure 3-35. In-app screen capture of the *REFLECTION* app showing the launch splash screen and play/pause controls.



Figure 3-36. In-app screen capture 30 seconds after launching the *REFLECTION* app.

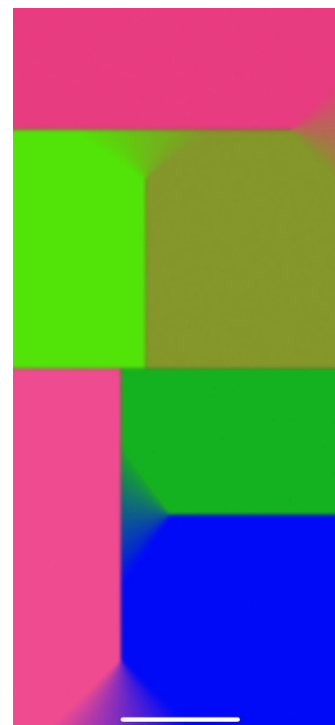


Figure 3-37. In-app screen capture 1 minute after launching the *REFLECTION* app.

In 2017, Eno and Chilvers broke from their interactive app format and released *REFLECTION*, which is available on the iPad, iPhone, Apple TV, and as a traditional album. However, as Brian Eno himself states in the App Store description for *REFLECTION*:

"My original intention with Ambient music was to make endless music, music that would be there as long as you wanted it to be. I wanted also that this music would unfold differently all the time - 'like sitting by a river': it's always the same river, but it's always changing. But recordings - whether vinyl, cassette or CD - are limited in length, and replay identically each time you listen to them. So in the past I was limited to making the systems which make the music, but then recording 30 minutes or an hour and releasing that. *REFLECTION* in its album form - on vinyl or CD - is like this. But the app by which *REFLECTION* is produced is

not restricted: it creates an endless and endlessly changing version of the piece of music.”

The app version of *REFLECTION* is designed solely to be listened to and not interacted with in any way. It features artwork that maintains a similar rectangular structure and fades through gradations of colors as Chilvers’ music system generates endless Eno-isms (see Figure 3-35, Figure 3-36, and Figure 3-37).

Bloom: 10 Worlds – 2018

Ten years after the release of the original version of *Bloom*, Eno and Chilvers joined forces once again to release *Bloom: 10 Worlds* (see Figure 3-38). In content promoting this new version, the creators analogize the relationship between the original *Bloom* and the new versions to the difference between a single and a full-length album (Eno & Chilvers, 2018).

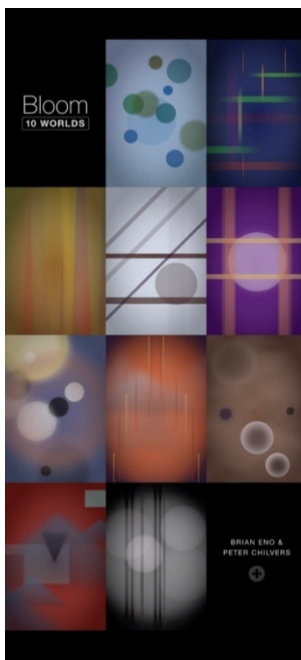


Figure 3-38. In-app screen capture of the worlds overview page from the *Bloom: 10 Worlds* app.

Bloom: 10 Worlds requires that its users experience all of the worlds in order. So to experience a world, the player must have opened all of the worlds that come before it. The first world inside of *Bloom: 10 Worlds* is called “Origin” and is, in essence, the same as the original *Bloom* app. Each of the remaining nine worlds becomes more complex both in terms of its sound pallet and the rules that guide and alter the music generated from user input.

The updated version feels more focused and refined. The same x- and y-axis relationship to stereo placement and pitch mostly has been preserved, but unlike their previous apps, Eno and Chilvers have removed any options for changing the mood or controlling the algorithm. The use of text is kept to a minimum as well as the few buttons found within the app feature symbols that intuitively indicate their function (see Figure 3-39).

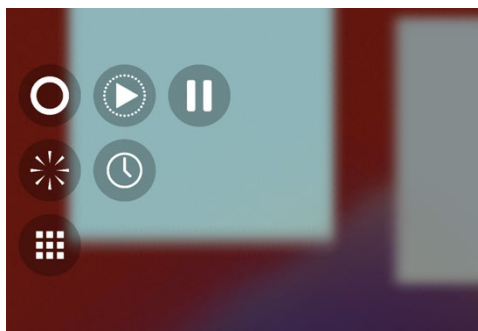


Figure 3-39. Cropped in-app screen capture showing the simplified options within each of the 10 worlds of the *Bloom: 10 Worlds* app.

Like the user interface elements, the music has also been purified, and the accompanying non-interactive musical elements found in the original *Bloom* have been stripped away, making it easier to listen for how the algorithm transforms the user's input over time. This was likely a conscious decision by the creators, as the response to user input within each world gradually changes in unpredictable ways the longer each world is

explored, which prolongs user interest and encourages the idea of music making as an act of play, not mastery.

Others Album Apps

Despite Owsinski's warning, a surprising number of artists have released album apps or app experiences of varying kinds in the ten years since the release of the Apple App Store. It seems that with each album app pulled from the mountain of digital refuse, several more are revealed beneath it. To begin the work of cataloging them and in hopes of preserving some record of their existence, what follows is a list of those identified for this survey.

This list is not assumed to be exhaustive. Many album apps are only available for short periods, sometimes by design but often because they are not given the care and attention necessary to sustain them as technological standards evolve.

Table 3-1. Listing of all Album Apps uncover by the author.

App Name	Artist	Year	Developer
Bloom	Brian Eno & Peter Chilvers	2008	Opal Limited
Trope	Brian Eno & Peter Chilvers	2009	Opal Limited
Air	Brian Eno, Peter Chilvers, & Sandra O'Neill	2009	Opal Limited
Steve Roach: Immersion I	Steve Roach	2010	Eric Freeman
Bjork: Biophilia	Björk	2011	Björk
Third Coast Percussion: John Cage	Third Coast Percussion & Joseph Genden	2012	Joseph Genden
PASSION PIT	Passion Pit	2012	Scott Snibbe
Steve Roach: Immersion II	Steve Roach	2012	Eric Freeman
Calvin Harris - 18 Months	Calvin Harris	2012	We Make Awesome Sh.it
Gwilym Gold's Tender Metal	Gwilym Gold	2012	Bronze Music LLP:
Jay-Z: Magna Carta	Jay-Z	2013	Björk

Philip Glass: REWORK_ - 2013	Philip Glass	2013	Scott Snibbe
Steve Roach: Immersion III	Steve Roach	2013	Eric Freeman
John Lennon: The Bermuda Tapes	John Lennon	2013	Design I/O
Lady Gaga - ARTPOP	Lady Gaga	2013	Relative Wave & TechHaus
Skrillex - Recess / Alien Ride app	Skrillex	2014	Motim Technologies Limited
Bernhoft Islander	Bernhoft	2014	HD360
Piano Ombre	Frànçois & The Atlas Mountains	2014	Domino Recording Co.
Paul McCartney - Band on the Run	Paul McCartney	2014	Concord Music Group
Paul McCartney - RAM	Paul McCartney	2014	Concord Music Group
Paul McCartney - McCartney	Paul McCartney	2014	Concord Music Group
Paul McCartney - McCartney II	Paul McCartney	2014	Concord Music Group
Paul McCartney - Wings Over America	Paul McCartney	2014	Concord Music Group
Metronomy – "I'm Aquarius" / Night Sky app	Paul McCartney	2014	Concord Music Group
Björk: Stonemilker VR	Björk	2015	Björk
Loscil: Adrift	Loscil	2015	Scott Morgan
Steve Roach: Immersion IV	Steve Roach	2015	Eric Freeman
Red Planet EP	Daisy and the Dark	2015	RT Sixty Ltd
Brian Eno: Reflektion	Brian Eno & Peter Chilvers	2017	Opal Limited
Coldplay : Hypnotised	Coldplay & Peter Chilvers	2017	Opal Limited
TCP: Resounding Earth	Third Coast Percussion & Joseph Genden	2017	Joseph Genden
Third Coast Percussion: the Music of Steve Reich	Third Coast Percussion & Joseph Genden	2017	Joseph Genden
Bloom: 10 Worlds	Brian Eno & Peter Chilvers	2018	Opal Limited
U2 eXPERIENCE	U2	2018	Lusional Limited
Asympt Man - Chasing Infinity	Asympt Man	2018	RT Sixty Ltd
Plan - Defabl & Vanessa Knight	Vanessa Knight	2018	RT Sixty Ltd

3.1.4 360 Videos

Primarily driven and supported by Facebook and YouTube, 360 videos enable creators to distribute simple and yet visually and aurally immersive content to any consumer that has access to either platform. When viewing on a mobile device, users can look around by merely tilting their displays in different directions. When viewing on a computer, clicking and dragging with the mouse alters viewers' orientations.

Despite the video's 360-degree view, audio in this format is commonly delivered in stereo. Facebook and YouTube do, however, support 360 videos with spatial audio in the form of First-Order Ambisonics and have also developed free tools to enable creators to generate their own content. Facebook 360 Videos also support Second-Order Ambisonic audio and the option to embed two separate audio files, one for head-locked audio and the other for spatialized audio.

360 videos are novel and therefore more captivating for consumers than traditional videos, but from an aural and musical perspective, the environment does not enable any wildly new approaches. Creators are still locked into a linear timeline and gain only several new audio spatialization capabilities that the experiencer may not even notice. Still, artists are using the format to express themselves in new and unique ways. 360 videos can also be embedded within other formats as a computationally efficient way to create backdrops or cutscenes.

Björk: "Stonemilker" - 2015

Björk's music video "Stonemilker," which was shot as an uncut 360 video by director Andrew Thomas Huang, was released, in true Björk fashion, the same month that YouTube began supporting 360 Video (Björk & Huang, 2015). The video was originally available only at Björk's

2015 retrospective held by New York City's Museum of Modern Art but three months later became publicly available via YouTube.

In an interview on the Song Exploder podcast (Hirway, 2015), Björk explains her motivation for recording “Stonemilker” as an intimate 360 video shot on a beach. The lyrics came to her while she was walking in circles on a beach and feeling a sense of emotional clarity. At this moment, she wondered how she could maintain such a state. In the music video, the audience assumes the position of Björk's own interior space as Björk herself becomes her own inner dialog circling and singing back at herself.

Through more complicated editing, similar roles could have been shown with traditional video recording techniques, but recording this song as a 360 video captures the immediacy and clarity of concepts that Björk conveys both in the lyrics and in the musical accompaniment. This is, nevertheless, as far as she has explored the potential of the 360 video format.

The audio is delivered in standard stereo with no spatial information of any kind. Using spatialized audio could have been unnecessarily distracting, but had Björk's voice moved around with her as she circled the camera, and had the instruments remained stationary, this could have further emphasized the cyclical concepts found elsewhere within the song.

Reeps One: “Does Not Exist” – 2016

In Reeps One's 360-music video “Does Not Exist” (Reeps One, 2016), the musical and video-graphic elements are so conceptually matched it is difficult to imagine which came first. Throughout the nearly two-and-a-half-minute video there is an almost seamless parallel between what is seen and what is heard. The video opens with Harry Yeff, the given name of performer Reeps One, a London-born beatboxer, warming up his voice against a depthless backdrop. If

viewers are streaming the video via a mobile device, they soon realize that they can look anywhere in the frame they desire. If they look away from Yeff while wearing headphones, however, they hear that his vocal sounds remain stationary. Even on repeat viewings, this is still a startling effect. Just before counting himself off, Yeff pronounces a final snare hit, which bounces around the studio along with clones of Yeff at each of the positions from which the sound echoes back.

The next scene cuts to an open desert with the sound of a soft breeze and Yeff far off in the distance. Then, in a rapid four-note gesture, Yeff flashes forward, with each note both sounding and appearing closer until he is back directly in front of the viewer, singing a gentle three-part harmony along with two additional copies of himself just over each of the viewer's shoulders. The video continues in this same fashion, complete with a few frame-rattling bass notes and a glass bottle that gets smashed inside a small abandoned cabin. The 360 video for “Does Not Exist” is indeed a must-see that illustrates precisely how immersive 360 Videos can be.

Conservation International Documentaries

360 videos and spatial audio are being used not only for music videos but also, and perhaps more effectively, for documentaries. Conservation International, a non-profit environmental organization, has now produced several 360 documentaries that use both head-locked and spatial audio (Conservation International, 2008).

A unique challenge of capturing footage and recording spatial audio for 360 documentaries is how to keep all of the crew and equipment out of the frame so that the audience can have an uninterrupted sense of presence. Composers Joel Douek and Benedict Green of EccoVR (EccoVR, 2016) have described the challenge of hiding microphones under large leaves

and behind branches at the on-site shoots for several of Conservation International's 360 documentaries.

The 360 video format seems ideal for documentaries because it places the viewer in the center of the experience, which studies have shown heighten the viewer's levels of empathy and the likelihood that they will take action of some kind after viewing (Archer & Finger, 2018). It has yet to be proven, though, whether these new formats will make any meaningful difference in furthering a filmmaker's cause.

3.2 Emerging Formats

Over the last several years, a wide range of less codified, more experimental and untested formats have begun to bubble up from the digital sludge, thanks primarily to the advent of virtual reality and more powerful portable devices. Some formats are on the verge of establishing themselves, while others are unclassifiable and stand as the lone example of a possible format for others to explore. Most of these emerging formats have sprung forth from renewed interest in virtual reality (VR).

Whereas the formats discussed so far have enabled creators to introduce new elements into the real world, VR empowers creators to wholly remove their audience from the real world and place them in a new universe of their design. Despite the recent hype surrounding virtual reality, the technology still exists in a largely inaccessible format because it requires dedicated hardware to experience. The current budget option for entering virtual reality is either a specially designed case for the user's cellphone that transforms the device into a head-mounted display (HMD)—

such as the Samsung Gear VR (approximately \$20)—or a dedicated VR HMD, such as the Oculus Go (\$200).

3.2.1 360+ Videos

360+ videos (my term) are collections of 360 videos that capture a single event from multiple perspectives, which creators then organize into a VR environment. In the final product, viewers can freely switch from one perspective to another and interact with any additional content. At this point, due to the sheer number of investors and startups exploring this category, 360+ videos could be regarded as an established format. Time and consumers always have the final say, however, and according to Cherie Hu, a contributing writer to Forbes magazine, companies and creators in the 360+ video market have a long way to go before their bottom line catches up with all of the hype they have created (Hu, 2018).

MelodyVR – 2015

Several new content creators, such as Anthony Matchett’s MelodyVR, which bills itself as a VR music company, have already begun building substantial libraries of live musical performances by the likes of Bebe Rexha, Fall Out Boy, The Script, The Chainsmokers, Kygo, Sean Paul, Kiss, Imagine Dragons, Wiz Khalifa, the London Symphony Orchestra, The Who, and more (MelodyVR, 2015).

So far, 360+ video experiences have only elaborated on the 360 video format by providing viewers the option of multiple perspectives. The audio is usually delivered as a traditional stereo mix that is not affected by a change of perspective. Creators have likely not prioritized more

immersive aural experiences because while looking out at a stadium from the drum riser can be visually thrilling, the experience would probably be subpar sonically.

Most 360+ video content creators seem focused on large public events like concerts, sports matches, or festivals. This is understandable from a financial standpoint: the equipment required to capture, process, edit, and deliver this format is still prohibitively expensive and thus remains accessible only to extremely well-funded projects. As this technology becomes more affordable, however, it will be interesting to see how this format evolves and how new voices explore the medium. Perhaps the inevitable decline in hype for VR startups and the subsequent retreat of venture capital from this sector will provide a more welcoming environment for others to share their visions for the future of this format.

3.2.2 Augmented Reality Albums

The augmented reality (AR) album is an emerging evolution of the album app. AR is a form of mixed reality where digital objects and information are superimposed on the real world. At its 2017 Worldwide Developers Conference, Apple released ARKit, a set of tools that made it easier for developers to build AR features into their apps. AR-enabled apps are accessible to anyone with a smartphone; they also can be viewed through dedicated HMDs like Magic Leap's One (Magic Leap, 2018) or Microsoft's HoloLens (Microsoft, 2017).

The spatial awareness that AR adds to album apps enables creators to manipulate the experience of their content based upon their audience's relationship to the world that surrounds them, as well as to overlay or place virtual objects into the viewer's world.

Bluebrain: Location-Based Album Apps — 2011

Well before Apple's release of its AR Kit, Bluebrain, a Washington, D.C.,-based group that now that now creates under the name The Holladay Brothers, released one of the earliest examples of how user data available through smartphones could, even without AR capabilities, augment the album app format.

Bluebrain's 2011 album *The National Mall* debuted through the Apple App Store, but listeners could only hear the album and navigate its contents by walking around the National Mall in Washington, D.C. Each area, statue, and monument served as a geographic trigger to activate different musical pieces and textures. The Holladay Brothers went on to create several other location-based album apps. Their most recent release is *Site: Nine Elms* (see Figure 3-40), which commemorates the 2018 opening of the new U.S. Embassy in London by musically exploring its grounds and surrounding areas (Bluebrain, 2018).

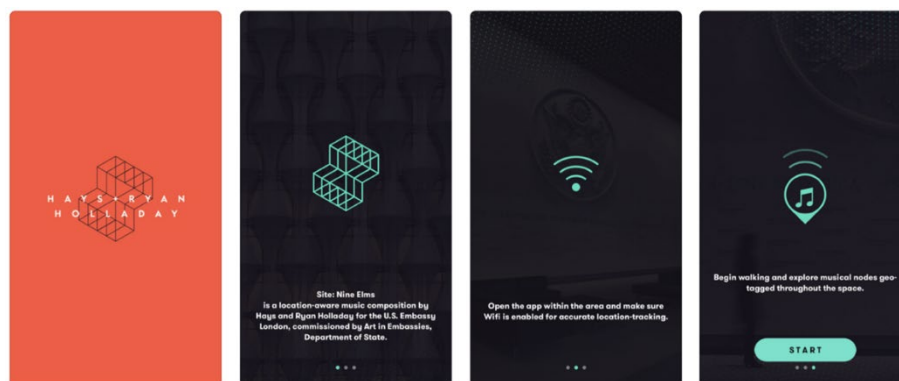


Figure 3-40. Preview images available on the iPhone App Store for the *Site: Nine Elms* app by The Holladay Brothers.

TCW: *Year One* – 2017

Under constant development and regularly updated, *Year One* is an AR album by Washington, D.C.,-based group and label TCW designed explicitly for the new augmented reality capabilities of the iPhone. On TCW's website (www.tcwav.com), the creators of *Year One* are

documenting their progress and publishing PDF guides so that others can begin to create their own AR albums.

Conceptually and functionally, *Year One* is simple and comparable to the Holladay Brothers' work. The Holladay Brothers' albums used GPS data to track user locations, which does provide the same level of spatial accuracy that AR can support. AR can track user movements to around a few inches (Olszacki, 2017), whereas GPS currently has an accuracy range of around nine to sixteen feet (Kastrenakes, 2017).

After a brief startup and calibration (see Figure 3-41–43), the *Year One* app displays a bird's-eye view of the user's augmented space. The user is represented as a small "+" and differently colored circles show the locations of different sounds (see Figure 3-44–46). The user can either move around physically to seek out specific circles of interest or remain stationary and allow the programmed movements of the spheres to dictate the sonic adventure.

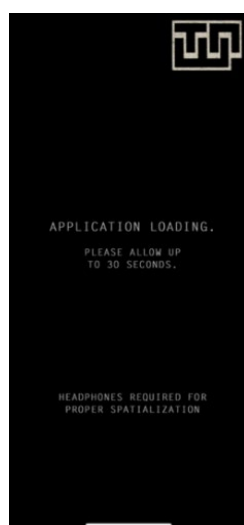


Figure 3-41. In-app screen capture of *Year One*: Step one of the calibration process.



Figure 3-42. In-app screen capture of *Year One*: Step two of the calibration process.

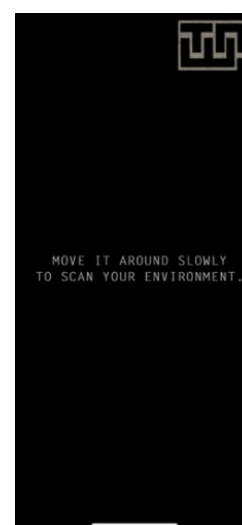


Figure 3-43. In-app screen capture of *Year One*: Step three of the calibration process.

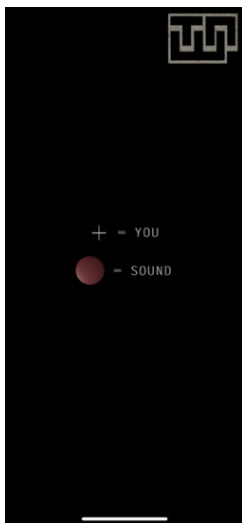


Figure 3-44. In-app screen capture of *Year One*: Step four of the calibration process.



Figure 3-45. In-app screen capture of *Year One*: Step five of the calibration process.

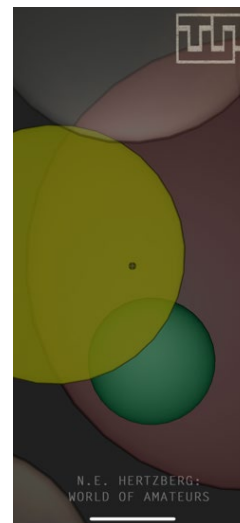


Figure 3-46. In-app screen capture of *Year One*: Exploring the augmented sound space.

Allowing the sound-circles to move produces similar sonic results regardless of whether the user is moving or lying on the floor, thus deemphasizing the importance of exploring the augmented world. Also, the musical materials obfuscate any spatial audio that may be present. Hopefully this is just the beginning for TCW and they will continue to develop and document ways for creators to further explore AR.

Zach Lieberman: *AR Audio Recording Test* – 2017

On September 6, 2017, Zach Lieberman (Lieberman, Homepage, 2015), a New York-based artist and educator who creates art through code, posted a brief video (Lieberman, Instagram, 2017) to his Instagram account demonstrating an AR app that enabled him to record, visualize, and play back audio over space rather than time (see Figure 3-47). The app was never made publicly available, but it has served as an inspiration for many.

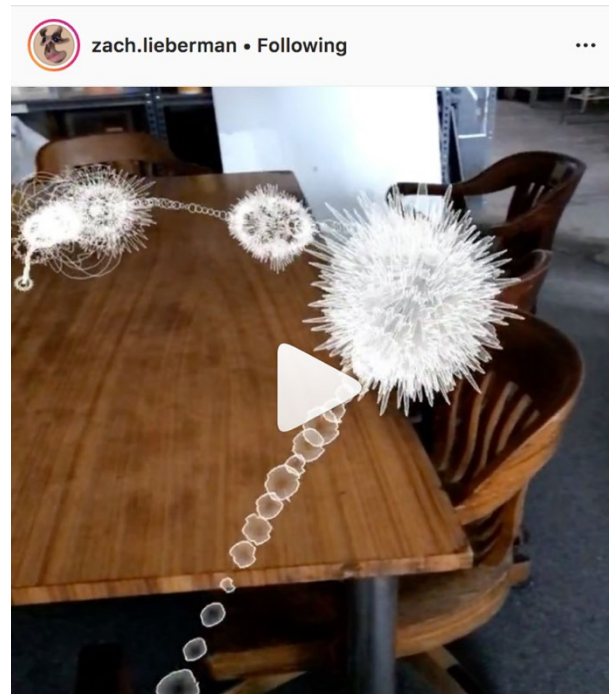


Figure 3-47. Screen capture of Zach Lieberman's original Instagram post demonstrating his AR audio test. Used with written permission from Zach Lieberman.

Particle Inc.: *Fields* – 2018

Fields (see Figure 3-48–49), an AR App by Particle Inc., enables users to create and explore sonic sculptures in augmented reality. The app connects to the microphone of users' mobile phones. They can then capture any desired sound and leave it suspended in space wherever a phone was at the time of recording. Unlike Lieberman's app, which uses a granular time-stretching algorithm to scrub the audio by moving the phone back through the spatial recording (Roads, *Microsounds*, 2001), *Fields* simply loops the recording infinitely and attenuates the volume based upon its distance from the viewer. *Fields* also allows users to save, record, and share their in-app creations, and features sonic sculptures by guest artists such as Matmos and Robert Lippok.

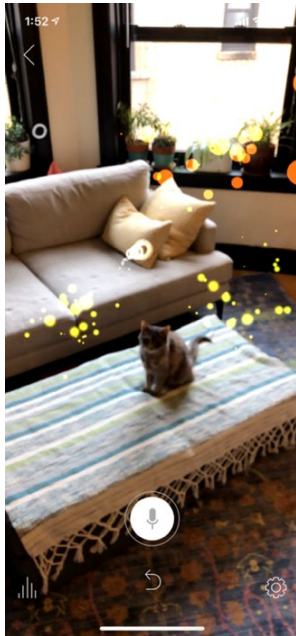


Figure 3-48. In-app screen capture showing the sound orbs of Particle Inc.'s "Fields."



Figure 3-49. In-app screen capture showing more sound orbs of Particle Inc.'s "Fields."

3.2.3 Virtual Reality Instruments

With few examples but still showing signs of a future, virtual reality instruments are beginning to carve out their own space in the virtual ecosystem. Most seem to be inspired by scenes from the 1982 movie *Tron* and place the user at the center of sizeable console-style step sequencer that commands a lead synth, bass, and electronic drum kit.

Snake Productions: *Block Rocking Beats* – 2017

Snake Productions' 2017 *Block Rocking Beats*, for the HTC Vive and Oculus Rift, is one example of the console-style VR instrument (see Figure 3-50). The setup allows users to sequence their music using royalty-free samples that are updated monthly. *Block Rocking Beats* also

supports jamming with up to three other people simultaneously and the ability to live-stream performances via sites like YouTube, Facebook, and Twitch (see Figure 3-50).



Figure 3-50. Image showing someone creating a drum pattern the "Block Rocking Beats" VR Instrument. Image sourced from <http://www.blockrockingbeats.com/>

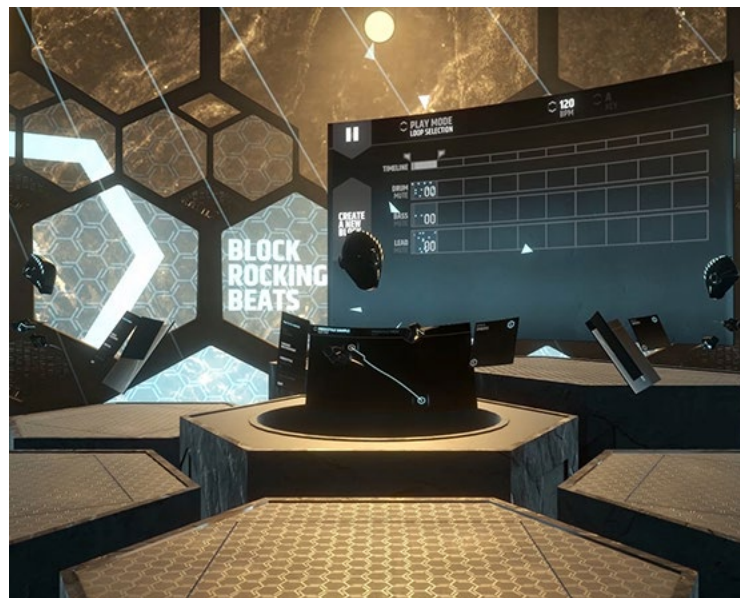


Figure 3-51. Image showing three people performing together in "Block Rocking Beats." Image sourced from <http://www.blockrockingbeats.com/>

Aesthetic Interactive: *EXA: The Infinite Instrument* – 2018

Under continuous development and with many of its features still considered to be in beta, Aesthetic Interactive’s *EXA: The Infinite Instrument* is a robust multi-player music performance environment (see Figure 3-52). Users can create their instruments using objects referred to as “ringers” (see Figure 3-53), which are sensitive to both the velocity and the angle from which they are played using a virtual mallet, bow, or proximity sensor. A ringer can be easily assigned to play any sample or note. Ringers do not behave like real-world objects and, as a result, the instruments built out of them do not have to either.

Instruments built in EXA encourage new instrument designs and performance techniques—for example, performers can create a spoke-like instrument that allows them to wave a mallet in a circle pattern and emit an arpeggiated pattern based upon the rate of revolutions (see Figure 3-54).



Figure 3-52. Multi-player performance in the *EXA: The Infinite Instrument* VR app. Still frame pulled from a promotional video found at https://store.steampowered.com/app/606920/EXA_The_Infinite_Instrument/

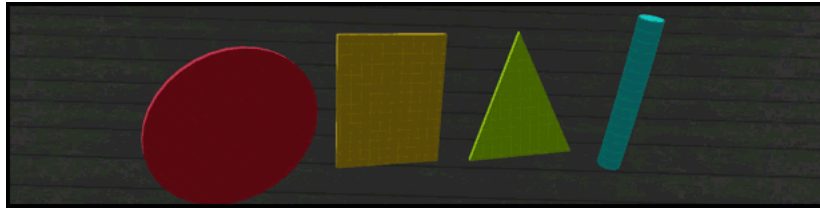


Figure 3-53. The four basic "ringers" used for building instruments in *EXA: The Infinite Instrument*. Image sourced from a GIF found at https://store.steampowered.com/app/606920/EXA_The_Infinite_Instrument/



Figure 3-54. Image showing one example of the strange instruments that can be built inside *EXA: The Infinite Instrument* and the new performance techniques they inspire. Image sourced from a GIF found at https://store.steampowered.com/app/606920/EXA_The_Infinite_Instrument/

Like many other music-making environments, EXA enables users to create musical loops. But rather than merely looping the recorded MIDI or audio, EXA also records the user's body movements. Once a loop has been captured, an optional “robot” performer appears and continues to perform the loop in the user’s place.

To enable more complex musical creations and instruments, EXA also supports simple MIDI-style sequencers that can be changed together to create highly complex patterns and musical structures (see Figure 3-55).

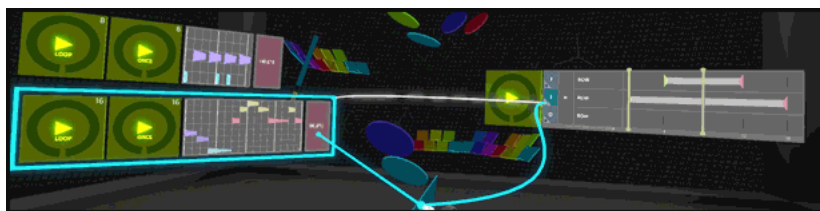


Figure 3-55. Image showing three sequencers inside *EXA: The Infinite Instrument* being chained together to create a more complex sequence. Image sourced from a GIF found at https://store.steampowered.com/app/606920/EXA_The_Infinite_Instrument/

With endless multiplayer jamming, live streaming, custom instruments, live looping, MIDI input and output, and support for third-party developers to integrate EXA with their virtual worlds or apps, *EXA: The Infinite Instrument* shows a lot of promise and vision for what musical performance spaces in virtual reality can do and be.



Figure 3-56. Image showing a full performance in action within *EXA: The Infinite Instrument*. Image sourced from a GIF found at https://store.steampowered.com/app/606920/EXA_The_Infinite_Instrument/

AliveIn Technology: *AliveInVR* – 2018

Several developers, such as AliveIn Technology, have begun investigating how VR can be used as a controller for existing musical software such as Ableton’s Live. One example of this is *AliveInVR*, which connects to a computer running Ableton Live via MIDI and can control most of Ableton using a virtual grid of interactive cubes (see Figure 3-57). Specifically, users can access:

- Session clips and scenes
- Mixer track levels, sends, panning and recording
- Real-time triggering of drum racks and instruments
- Step sequencer
- Continuous controllers and macros
- Loop record, arrange record, and global transport controls

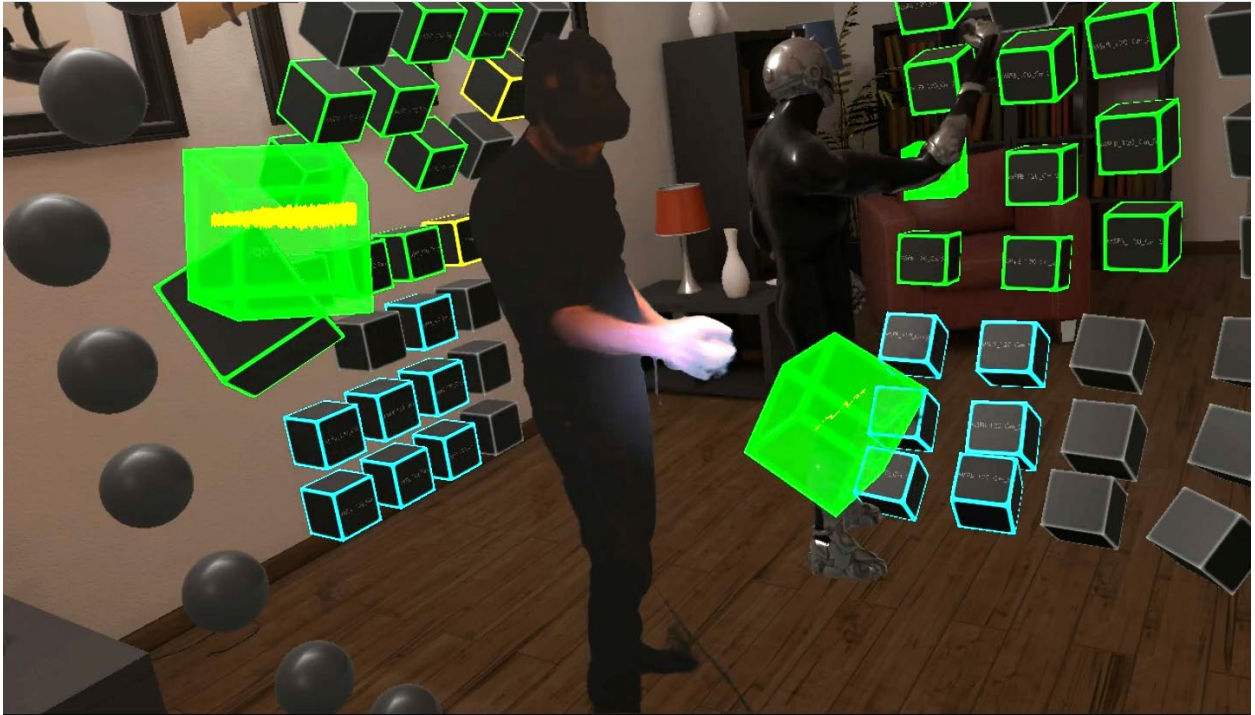


Figure 3-57. Image showing two performers using *AliveInVR* to control an Ableton Live set. Image sourced from <https://steamcommunity.com/games/674180/announcements/detail/1651009712387237652>

3.2.4 Virtual Reality Albums

There are plenty of examples of VR musical experiences that provide an experience similar to that of 360 music videos, such as *When We Land* by Phenomena Labs and Young Yosef (see Figure 3-58). But there also are several recently released works or projects in development that begin to carefully evaluate how the concept of an album can be expanded into new realities.



Figure 3-58. Still from the VR experience *When We Land*. Image sourced from the apps Oculus Store page: <https://www.oculus.com/experiences/go/944889082266875/>

Hot Sugar: *Melody of Dust* – 2017

One of the first VR albums, *Melody of Dust*, was created in 2017 by the artist Hot Sugar. Upon entering the experience, users watch a quick overview of the controls and then find themselves in a grandiose bedroom littered with assorted items of old luxury (see Figure 3-59). Behind a blue double door is a marbled fountain with a tornado swirling above it. Users eventually discover that they can collect items from the bedroom and throw them into the fountain (see Figure 3-60). Specific combinations of items unlock different musical soundscapes and melodies. The game contains 84 melodies in total, but often a player will only discover a dozen or so before reaching a climatic finish and seeing a screen indicating how many melodies they have uncovered so far.



Figure 3-59. Opening bedroom scene from *Melody of Dust*. Image sourced from the app's Steam home page: https://store.steampowered.com/app/527110/The_Melody_of_Dust/



Figure 3-60. Looking through the double door of the *Melody of Dust* bedroom out toward the fountain. Image sourced from the app's Steam home page: https://store.steampowered.com/app/527110/The_Melody_of_Dust/

Saint Motel: *saintmotelevision* – 2018

Released as both an AR and VR experience bundled in one app, Saint Motel's 2018 album *saintmotelevision* is a collection of twelve songs with accompanying 360 music videos. Both the AR and VR version present the user with slightly different methods for navigating and selecting which 360 video to watch.

In the AR version of the experience, an old tube television magically appears on the floor, and the user's phone becomes the remote (see Figure 3-61). Once users have selected a video, tapping the name of the song transforms the television into a portal that they must walk through to start the related 360 video (see Figure 3-62).



Figure 3-61. In-app screen capture showing the TV preview and remote control for navigating and choosing a 360 video to watch.



Figure 3-62. In-app screen capture showing the portal into one of the 360 videos.

In the VR version, users stand poolside with floating screens representing each of the 360 videos. To choose a video, users look at one and then select between streaming or downloading the video to their device (see Figure 3-63).

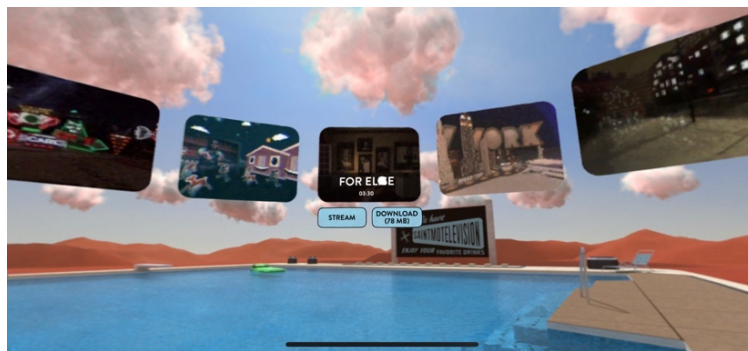


Figure 3-63. In-app screen capture showing the 360 video selection screen for the VR version of the *saintmotelevision* app.

Beyond this, each experience is the same, no longer an AR or VR experience but simply a 360 video. While this certainly does not break free of some of the established formats, it does present a practical recipe that is both efficient and effective.

I-Exist: *Consciousness* — The VR Concept Album (2018)

The music duo I-Exist, seeing the potential of virtual reality, began learning about the capabilities and development process in 2016. Two years later, they launched a Kickstarter campaign to fund a project with two objectives: First, to create and release what they are calling the world’s first VR concept album called *Consciousness* (see Figure 3-64), and second, to produce a website with tutorials and documentation showing how they built their VR concept album so that others can create similar works on their own (iExist, 2018).



Figure 3-64. Kickstarter campaign photo for I-Exist's VR concept album *Consciousness*. Image sourced from I-Exist's press release materials.

On the project’s Kickstarter page, I-Exist outlines some of the thought processes behind their VR album. One feature is an example of what the duo refers to as “conscious instruments” — new virtual instruments that couldn’t exist in the “real world.” These would be playable within the experience, such as one conceptual instrument called “The Celestial Dither” (see Figure 3-65).

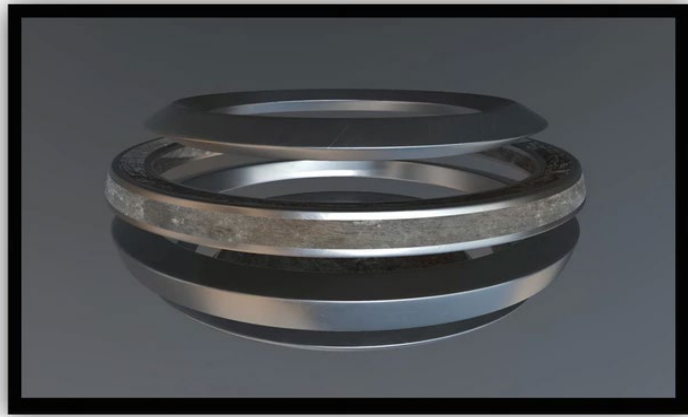


Figure 3-65. "The Celestial Dither," one of the "conscious instruments" that will be playable within I-Exist's forthcoming VR concept album *Consciousness*. Image sourced from the project's Kickstarter campaign page: <https://www.kickstarter.com/projects/844845263/consciousness-the-vr-concept-album/description>

A few other photos found on the project's Kickstarter page and promotional videos further detail what could be a valuable model for the future of the VR album (see Figure 3-66 and Figure 3-67).



Figure 3-66. What appears to be a chair inside a listening station. Image sourced from promotional materials for I-Exist's forthcoming VR concept album *Consciousness*.

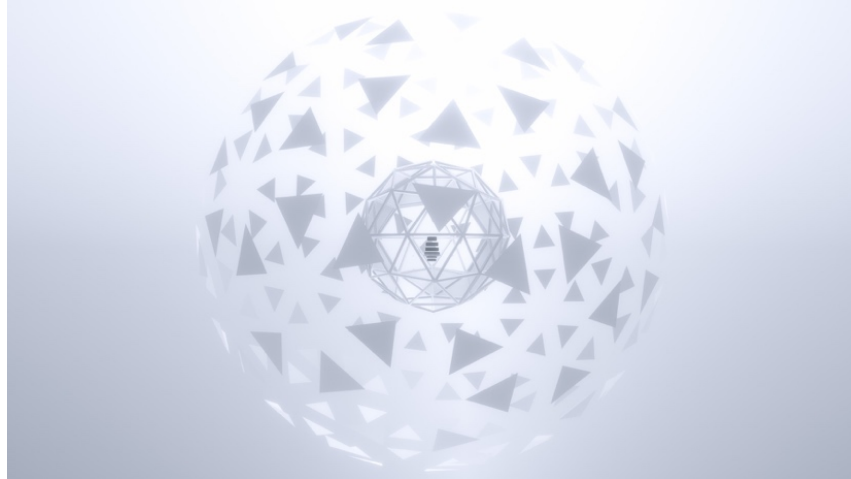


Figure 3-67. Distant image of what appears to be a listening station. Image source from promotional materials for I-Exist's forthcoming VR concept album *Consciousness*.

3.2.5 Virtual Reality Experiences

Virtual reality experiences represent a majority of the content being created for VR consumers. Until further subcategories are formulated that better classify virtual reality content, any work that is not a video game, 360+ video, or VR album should be called a VR experience. Like the example of the Netflix Oculus app mentioned in the introduction to this document and other VR formats already discussed, creators of VR experiences are struggling to realize the format's potential. In contrast, *Notes on Blindness*, discussed below, stands ahead of other VR experiences as an excellent example of the format's unique potential—that is, placing the audience in someone else's shoes.

Novelab: Notes on Blindness – 2017

Novelab's *Notes on Blindness* presents the recorded journal entries of writer John Hull, who began going blind in 1983 just before the birth of his son. In a stunning and interactive visual world, the VR experience features both head-locked and carefully spatialized audio that bring the sonic landscapes to life and give the viewer a strong sense of presence.

The journey begins with the viewer in near-total blackness; only the slightest impressions of a public park are visible (see Figure 3-68). Gradually, as Hull describes the park in one of his recorded entries, the scene comes to life, and the void fills with activity.

The experience is divided into several scenes, each of which explores a unique aspect of being blind. Of the handful presented, two scenes stand out: Hull's observations on how the sighted and blind experience weather differently. For the sighted, a sunny day with no wind is considered a beautiful day. But for the blind, a blustery and rainy day might be considered beautiful because of how the weather sonically activates the world.

In the first of these two weather-focused scenes, viewers can use their hand-held controllers to focus the wind and apply it as a tool to see the world that surrounds them. As the wind shifts, the sounds and silhouettes of a bird, trees, grass, a swing set, and windchimes appear out of the void and eventually form a kind of "mental image" of standing on a front porch and looking out over a suburban yard (see Figure 3-69).

Similarly, in the second weather-related scene, just after being outside and listening to a fully spatialized thunderstorm, viewers find themselves inside of a small cabin with a roof. Viewers can focus on an area within the scene to direct where the drops of water coming through the roof will fall. The sound of the water drop hitting the object functions as sonar to help the listener see the room and the objects within it. This represents one of Hull's fantasies: He loved it when it rained outside and when the thunder rolled because it placed a roof over his head and revealed the world around him. He always wished it would rain inside for the same reason. Both scenes help the viewer understand the sonic cues that the blind listen for in order to make sense of the world.



Figure 3-68. In-app screen capture of the opening park scene from the *Notes on Blindness* app.



Figure 3-69. In-app screen capture of using wind to expose the world inside of the *Notes on Blindness* app.

Notes on Blindness is one of the best currently available VR experiences. It often gives viewers the urge to close their eyes and listen, which may seem like odd impulse in a space and technological medium that generally emphasizes visual experiences over aural ones.

Just after the final scene—a fully spatialized choir concert in a cathedral where the sound is so enchanting that the visuals become entirely unnecessary and almost distracting—Hull recalls a memory of someone describing the appearance of the choir to him after the concert. He says that while he remembers politely listening to their description, by this point, Hull had become so comfortable with blindness that the sounds were enough for him and he no longer missed his sight.

The VR experience lasts only around 20 minutes, but within this short time frame, *Notes on Blindness* takes viewers from simply expecting an immersive visual experiencing to feeling akin to how John Hull felt, enchanted and fulfilled by listening.

3.2.6 Outliers

Some of the most compelling works in new interactive and immersive digital formats fall between the cracks, not fitting into any one format. Their designers seem obsessed with building worlds that glorify the pure pursuit of beautiful moments where visuals, sounds, and the player's presence coalesce into unique and unrepeatable moments.

Fernando Ramallo & David Kanaga: *Panoramical* – 2015

Panoramical, available on a wide range of systems, is a collection of malleable worlds where users are given nine controls that each simultaneously affect an aspect of the audio and

visuals distinctly within each world. For example, when experiencing *Panoramical* on a laptop, the nine controls can be adjusted by holding down specific keys, either individually or in combination while moving the trackpad (see Figure 3-70).

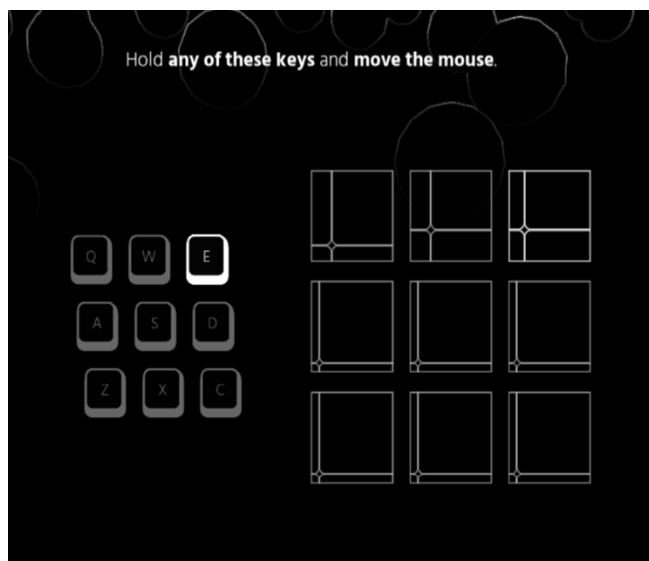


Figure 3-70. In-app screen capture showing the laptop controls for *Panoramical*.

In one world, holding “Q” and sliding from the lower left corner of the trackpad to the upper right corner narrows the gap between the perceived floor and ceiling of the world, while also raising the volume and increasing the level of activity of an arrhythmic low-string gesture (see Figure 3-71 and Figure 3-72). Increasing the “W” control’s value adds high-string tremolos and visually agitates the floor and ceiling of the world (see Figure 3-73). Each of the remaining controls add additional auditory and visual elements to the world, but not always on scales of silent to loud or invisible to visible (see Figure 3-74 – Figure 3-84). Often, each control will have multiple stages of sonic and visual influence upon the world. In the following images, the current control values are represented by the nine squares in the lower left corner of each image.



Figure 3-71. In-app screen capture showing the default state of a world in *Panoramical*.



Figure 3-72. In-app screen capture showing a world in *Panoramical* when the "Q" control is at its maximum value.



Figure 3-73. In-app screen capture showing a world in *Panoramical* when the "W" control is at its maximum value.



Figure 3-74. In-app screen capture showing a world in *Panoramical* when the "E" control is at about half of its maximum value.

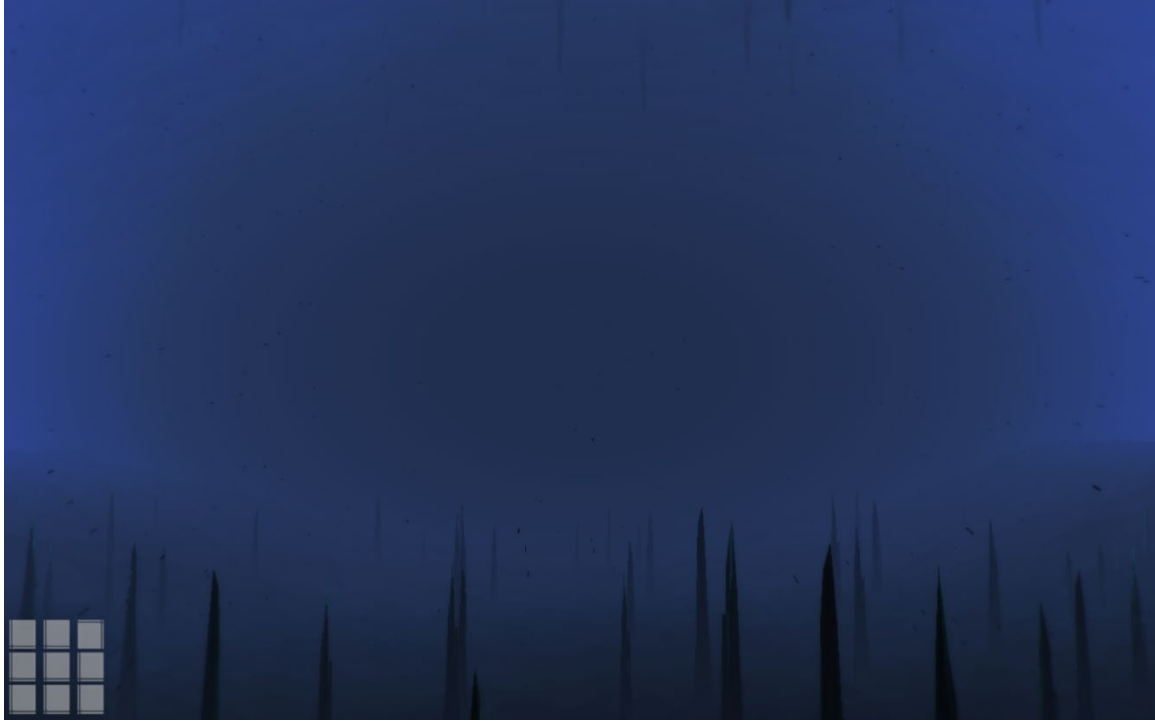


Figure 3-75. In-app screen capture showing a world in *Panoramical* when the "E" control is at its maximum value.



Figure 3-76. In-app screen capture showing a world in *Panoramical* when the "A" control is at about half of its maximum value.



Figure 3-77. In-app screen capture showing a world in *Panoramical* when the "A" control is at its maximum value.



Figure 3-78. In-app screen capture showing a world in *Panoramical* when the "S" control is at about half its maximum value.



Figure 3-79. In-app screen capture showing a world in *Panoramical* when the "D" control is at its maximum value.

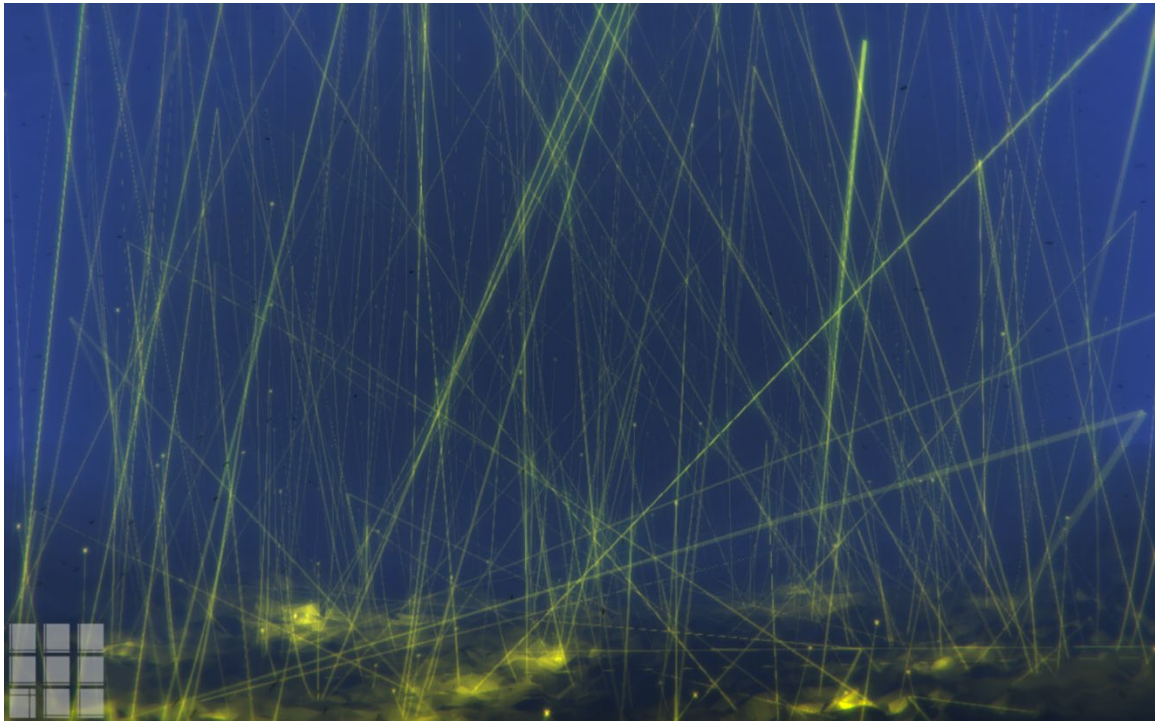


Figure 3-80. In-app screen capture showing a world in *Panoramical* when the "Z" control is at its maximum value.



Figure 3-81. In-app screen capture showing a world in *Panoramical* when the "X" control is at its maximum value.

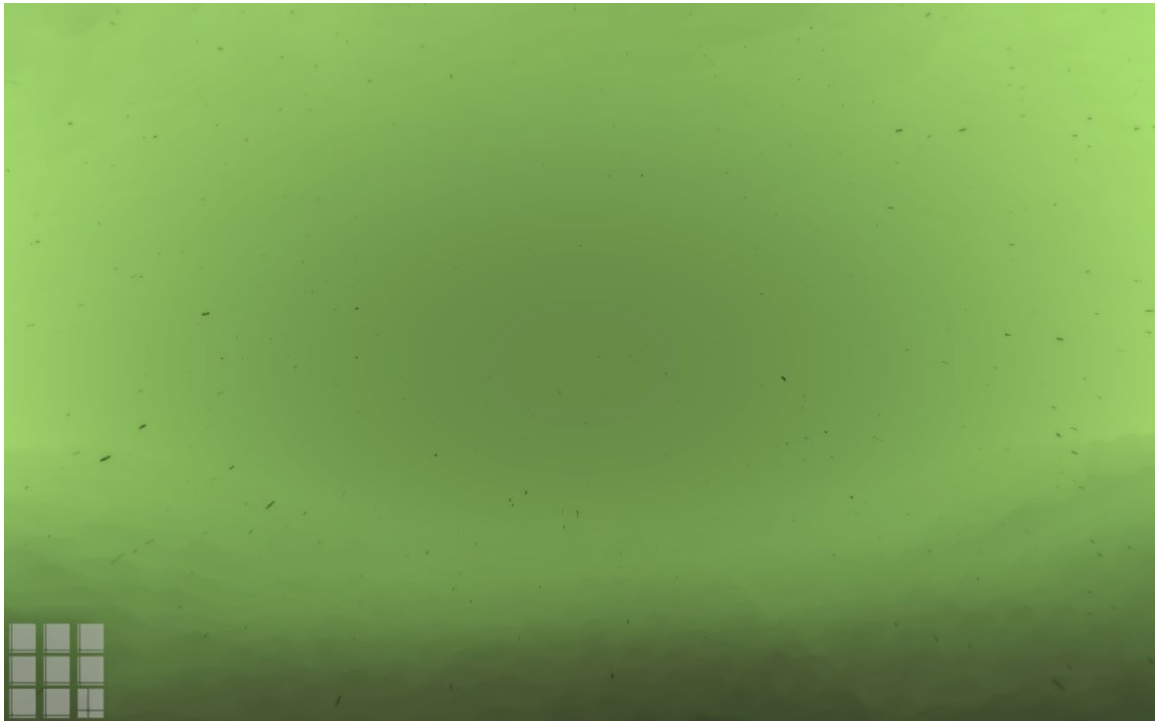


Figure 3-82. In-app screen capture showing a world in *Panoramical* when the "C" control is at about half its maximum value.



Figure 3-83. In-app screen capture showing a world in *Panoramical* when the "C" control is at its maximum value.

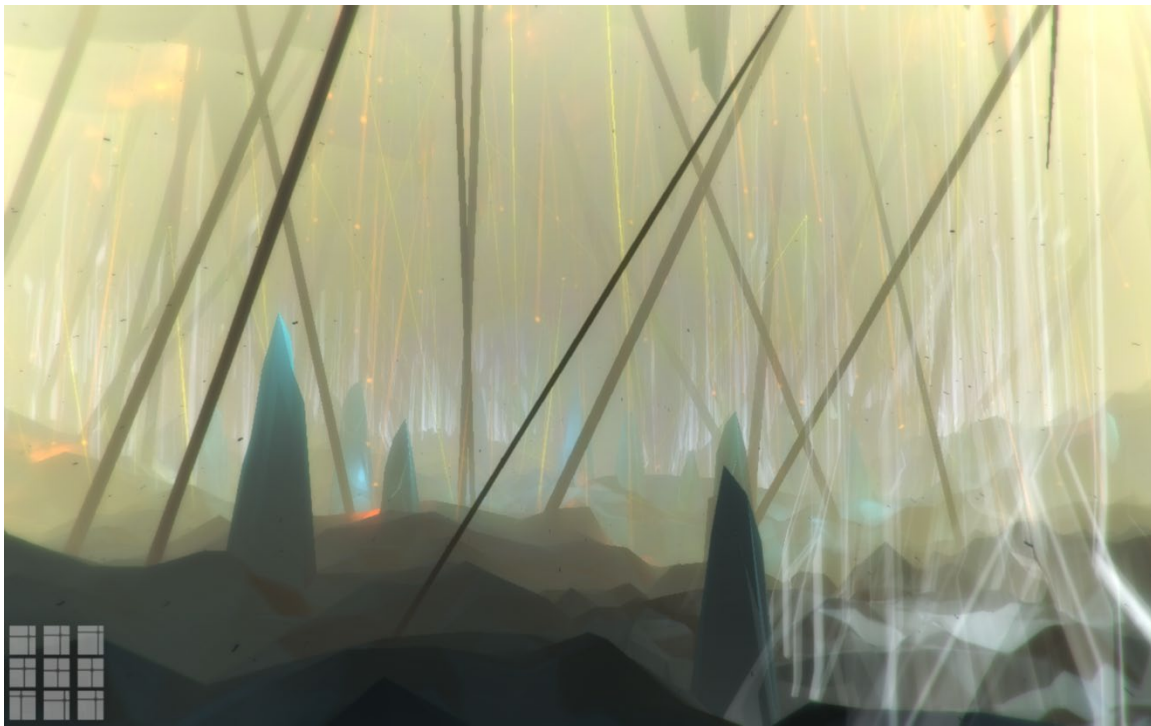


Figure 3-84. In-app screen capture showing one possible variation on a world in *Panoramical* when all nine controls are enabled.

Many examples of the resulting sound effects from altering these controls can be found by searching “Panoramical Gameplay” on YouTube. YouTuber Defunct Games’ video “Hands on Panoramical (Steam)” is an excellent overview. The video begins with the introductory tutorial and then narrates the experience of exploring several worlds (Defunct Games, 2015).

Although *Panoramical* is available through major video game platforms like Steam, it is not a game—at least not in the typical sense of the word. There is no winning or losing; there is just experiencing. This is what sets *Panoramical* apart. There are plenty of examples of video games with equally captivating visuals and reactive soundtracks but few like *Panoramical* where experiencing is the sole focus of the game.

Plan8: *PolyDome* – 2016

PolyDome, by Plan8, is another environment where there is no goal other than to enjoy the experience. Available in a variety of formats but most immersive when played in VR, *PolyDome* is something like a planetary sequencer. The bodiless player can fly around inside a giant dome where they will find orbiting objects of different shapes. The objects remain silent until the player hits them by shooting blue spheres, which add the activated object and its contained sound to the sequence (see Figure 3-85). The activated objects and sequence are visualized by laser-like lines that connect the activated objects. Each time a player launches a ball, a synthesized pluck sound is produced and functions as the melody for the soundscape.

Sonically, *PolyDome* never becomes much more than a smooth electronic jam and sounds mostly the same once more than a handful of objects have been added to the sequence. The sounds are fully spatialized so certain sounds can be focused on by flying around the space, but this does not provide enough variation to maintain the player's interest for very long.

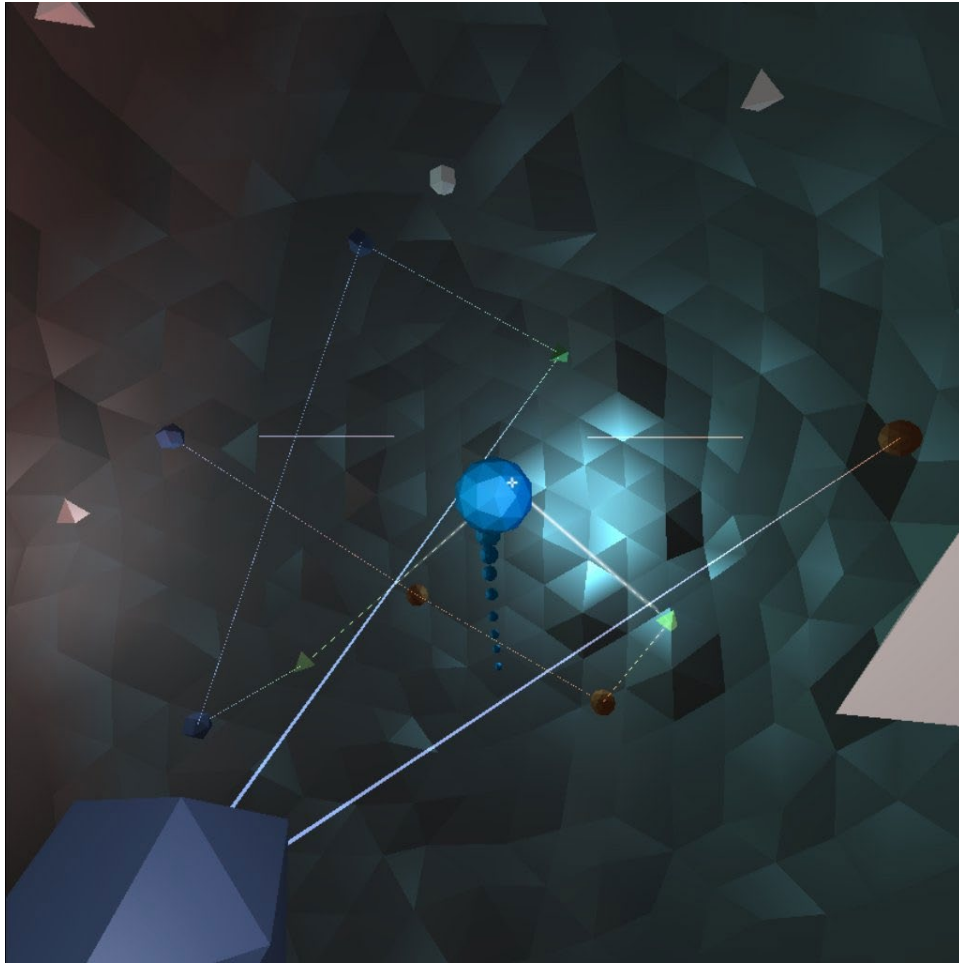


Figure 3-85. In-app screen capture showing a stream of shot blue ball, the sound objects, and sequencer paths in *Polydome*.

Particle Inc.: *Drops: Rhythm Garden* – 2018

Drops: Rhythm Garden, by Particle Inc., also reimagines what a sequencer in a gravity-based virtual reality could be. Unlike *PolyDome*, however, the player builds—or, as the title of the game implies, nurtures—their garden of rhythms to life (see Figure 3-86).

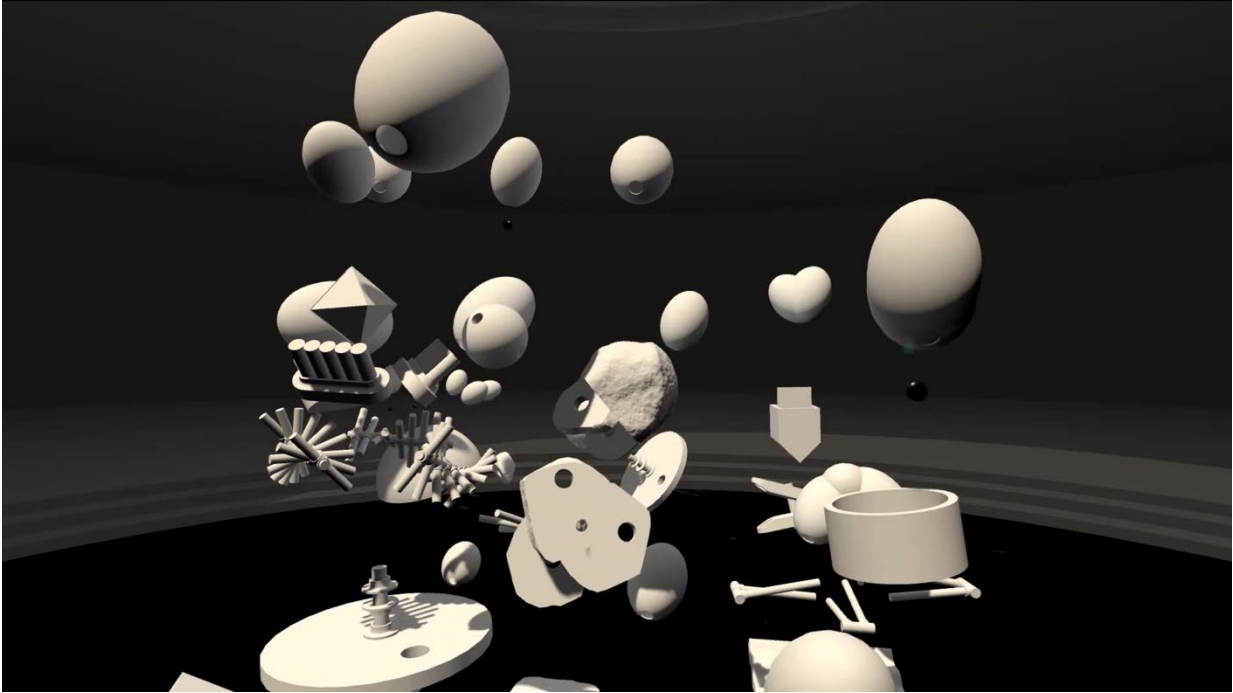


Figure 3-86. Still frame capture from the promotional video for *Drops: Rhythm Garden* showing one complete garden in action. Image sourced from the game's Steam Store page: https://store.steampowered.com/app/864960/Drops_Rhythm_Garden/



Figure 3-87. In-app screen capture showing the palette for selecting objects to place into the *Drops: Rhythm Garden*. Image sourced from the game's Steam Store page: https://store.steampowered.com/app/864960/Drops_Rhythm_Garden/

In their left hand, players hold a palette of objects that they can resize and position anywhere within their garden using their right hand (see Figure 3-87). These objects divide into two main categories: those that spawn a controllable stream of black marbles, and objects for those marbles to fall upon. The player can animate all of the objects within their garden and alter the sound produced when a marble interacts with each object. Finally, the game allows players to save and edit as many gardens as they would like.

Robin Arnott: *Soundself*—2018

Soundself comes from the fusion of creator Robin Arnott's love of video games and his meditation practice. Around 2010, Arnott became focused on what he described as his dual selves: his disembodied self in the video games he had played since a child and his bodied self outside of those games.

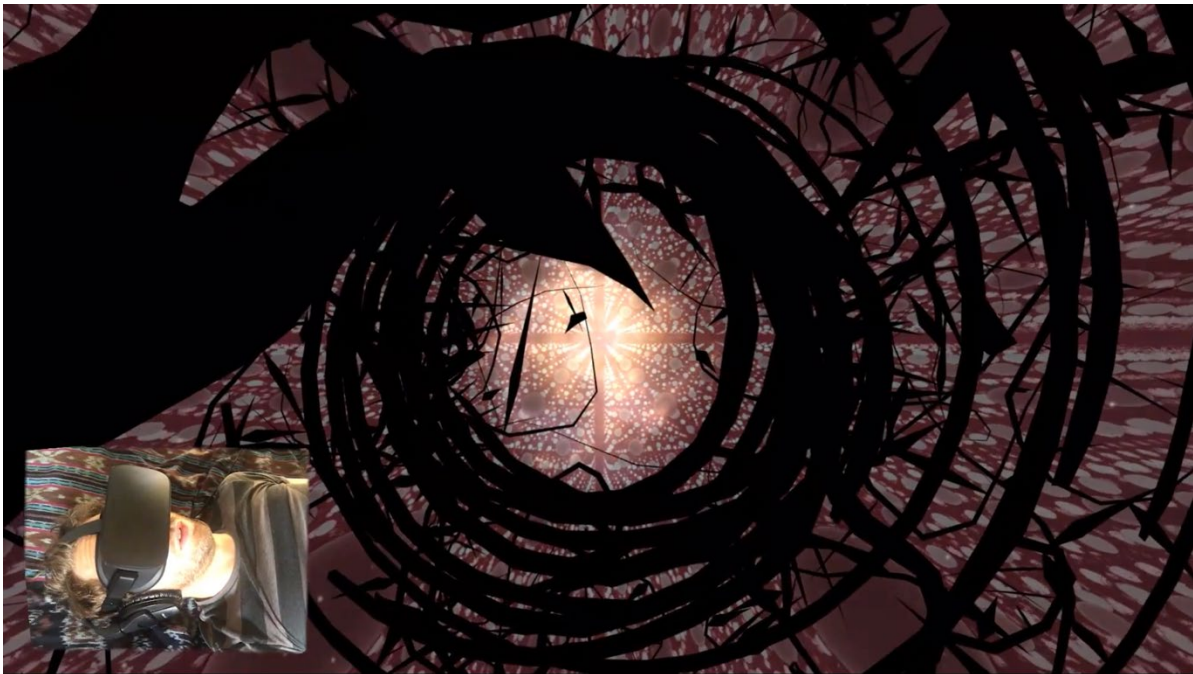


Figure 3-88. Still frame captured from a promotional video for *SoundSelf*. Image sourced from the games website: <https://www.soundself.com/>

As a game developer, Arnott, like any craftsman, makes sense of the world through the tools of his trade. With *Soundself*, he wanted to explore how video games could be used as a tool for human transcendence. Rather than keep the worlds of the video game self and the physical self separate, he encourages players to bring them together. Arnott realized through his meditation experiences that the human voice could serve as a tool to pass the embodied self through to the disembodied world of video games. This realization is what inspired *Soundself*, a VR meditation space that positions a microphone on an HMD to visually and sonically respond to the players' sung tones (see Figure 3-88). The full experience lasts an hour and differs each time depending on the breath and sung tones of the player.

Chapter 4: The Current State of Technology

Although creators have experimented with 360 videos, AR, and VR for years, these technologies started to become more widely available in 2018. Notable milestones included the release by Apple and Google of AR software development kits (SDKs), the release of the first standalone virtual reality headset, and the incorporation of spatial audio workflows into most major DAWs. In addition to these technological advancements, tools like video game engines and their related audio middleware, which already were well equipped for AR and VR content creation, started to focus on workflows and features for these formats.

This chapter introduces readers to the essential tools and concepts necessary to begin working with the digital formats discussed in this document. It is assumed that the reader is already familiar with traditional music production tools like DAWs and plug-ins, and thus only information about current options and developments in those areas that better support or integrate with these formats is examined. Tools with which composers may not be as familiar, however, such as FMOD and Wwise, two popular middleware programs for video game sound design, are discussed in greater detail, as developing a strong understanding of these programs' features and capabilities will allow creators to make better educated choices about which tools to select for future projects.

Due to the ongoing and rapid developments made in these areas, a secondary goal of this chapter is to provide a historical snapshot of these technologies in late 2018 and early 2019. Many

of the sources cited in this document, such as the developer resources provide by Oculus (Oculus, 2014), will likely remain useful references for future readers seeking more current information.

4.1 360 Video

YouTube (Hollister, YouTube's Ready To Blow Your Mind With 360-Degree Videos, 2015) and Facebook (Maher, 2015) brought 360 video to the masses in 2015 when both companies announced support for the format on their platforms. Playback initially was possible on only a handful of Android and desktop devices (Bonnington, 2015; Pagliery, 2015), but that quickly changed, and by 2018, 360 videos could be viewed on most devices (Rieger, 2018).

4.1.1 Playback & Interaction

When viewing 360 videos on a mobile device, viewers can tilt or turn their devices to look around the captured scene. In conditions where it is impossible or impractical for a viewer to rotate their devices a full 360 degrees, viewers can also drag a finger across the display to change their perspective without moving the device. In addition, developers can enable focus detection to callout visual and sonic elements based upon the area of the image the viewer is looking at.

This same finger-drag functionality allows desktop computer users to view and navigate 360 videos by clicking and dragging with the mouse. While watching 360 videos on a mobile device provides a more intuitive and immersive viewing experience, the availability of the format on either platform makes 360 videos the most accessible of all the new digital media formats discussed in this document.

4.1.2 Compatible Video Players

In addition to Facebook and YouTube, a handful of additional online and offline players now support 360 videos. In 2017, Vimeo, one of YouTube's competitors, announced basic support for 360 videos (Ingraham, 2017). VLC, a cross-platform media player app, added support for viewing 360 videos offline in early 2018 (Ellis, 2018). The most recent and feature-rich option for delivering 360-video experiences, however, is not a social video-sharing platform. Rather, it is OmniVirt, a toolset for developing, hosting, and monetizing 360 video content (Omnivirt, 2013).

Originally named AdsOptimal (Ha, 2016), OmniVirt expands the interactive advertising possibilities of 360 videos by adding the ability to link to other content through clickable hotspots (see Figure 4-1) anchored to objects within the scene (Omnivirt, 2017). OmniVirt also supports AR and VR environments, which means content hosted with OmniVirt tools can smoothly transition between all three media formats.

While OmniVirt is currently a leader in enhancing the interactivity of 360 videos, the company has several competitors (Ha, 2016), including Facebook, YouTube, and Vimeo. Most of these companies are focused on developing tools for advertising, but hopefully this competition will accelerate the evolution of 360 video functionality generally.

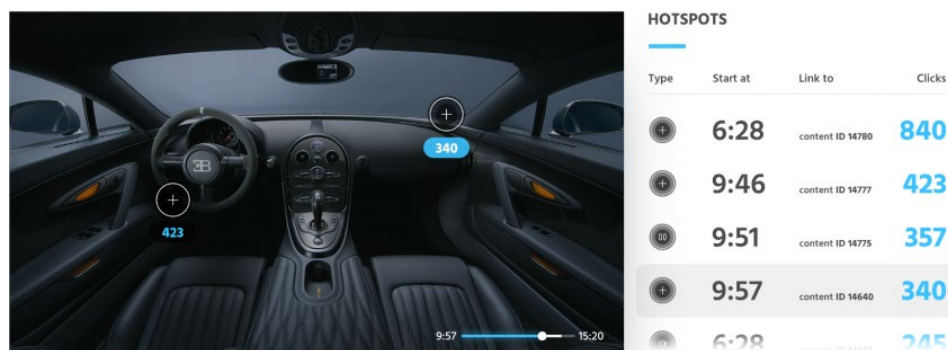


Figure 4-1. Image showing OmniVirt's hotspots and analytics capabilities for 360 videos. Image sourced from <https://www.omnivirt.com/blog/360-interactive-video/>

4.1.3 Supported Audio Formats

All platforms for viewing 360 videos support standard stereo audio playback, but an increasing number have begun supporting more immersive audio formats, as well. Facebook and YouTube, for example, support two different spatial audio formats: FOA, and FOA with an additional stereo track for head-locked audio. The latter enables creators to embed videos with one audio stream that remains spatially independent from the viewer's head movements along with another audio stream that behaves like traditional stereo audio (Google, 2016). In addition to these formats, Facebook 360 videos support the use of SOA (Facebook, 2016). Other playback solutions, which are not as widely accessible as YouTube or Facebook, have adopted a range of spatial and surround audio formats (Rieger, 2018), from binaural to HOA and even a specialized form of Dolby Atmos for VR (Dolby, 2016).

4.2 Augmented Reality

Two major SDKs exist for developing AR applications and experiences: Apple's ARKit (Apple, 2017) and Google's ARCore (Gosalia, 2018). Both SDKs provide developers with the

same essential functions required to create interactive AR content and differ only in several small details (Blum, 2018; Vieyra Software, 2018).

4.2.1 Current Capabilities of Augmented Reality

Motion Tracking

The foundation of AR rests upon tracking objects in the real world through visual inertial odometry (VIO). VIO monitors points of interest detected within a camera's field of view and compares each point's transformation across consecutive video frames to additional data gathered from internal motion sensors (Zhu, Atanasov, & Daniilidis, 2017). VIO makes it possible for virtual objects to remain at fixed points while the viewer navigates around them by anchoring their location relative to these points of interest (see Figure 4-2).

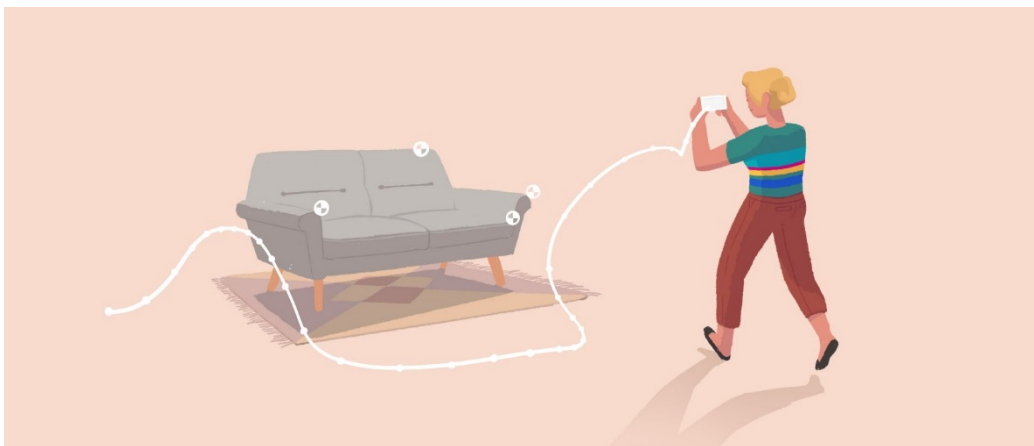


Figure 4-2. Graphic showing how visual inertial odometry works. Graphic sourced from <https://developers.google.com/ar/discover/concepts>

Plane Detection

Once the device's motion tracking system has identified enough points of interest, it will begin searching for horizontal or vertical relationships to determine where features such as the

ground, walls, tables, or other similar reference points reside within the space. Plane detection allows the placement of virtual objects on surfaces in the real world (see Figure 4-3). At this point, both ARKit and ARCore can only detect simple geometric shapes or identify known 3D objects (Apple, 2018), but forthcoming SDKs, such as one from 6D.AI, claim to have more advanced object-detection capabilities (6D.AI, 2018).

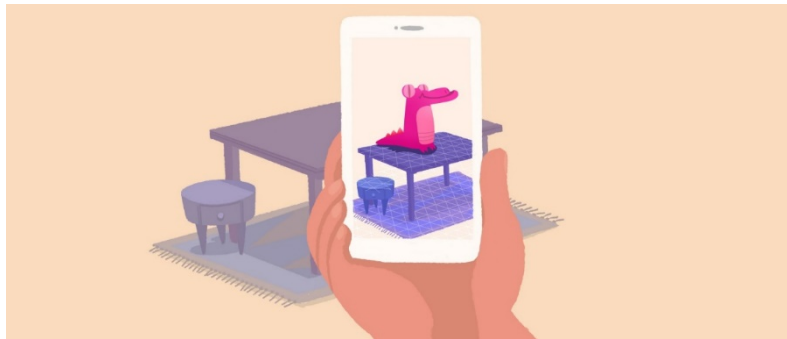


Figure 4-3. Graphic showing how plane detection works. Graphic sourced from <https://developers.google.com/ar/discover/concepts>

Light Estimation

The final stage of creating a believable AR experience is to adjust the overall lighting of virtual objects according to the estimated light in the real world (see Figure 4-4). Neither ARKit nor ARCore is currently able to determine the direction of a light source, which often results in inaccurate shadows. Instead, both SDKs track the global luminance of the scene (Google, 2018).



Figure 4-4. Graphic showing how light estimation works. Graphic sourced from <https://developers.google.com/ar/discover/concepts>

Face Tracking

Face tracking, an AR capability made famous by apps like Snapchat, is enabled through several algorithms that identify patterns or landmark features with a variety of sensors. Geometric face tracking, one of the earliest examples, compares a library of known facial images against the live feed from the user's camera looking for previously identified indicators and geometric relationships (Starovoitov & Samal, 1999). The latest iPhones rely on Apple's TrueDepth camera for facial tracking and recognition (see Figure 4-5). The TrueDepth camera projects and analyzes 30,000 infrared dots on the user's face (Apple, 2017). In contrast to geometric face tracking, Apple's TrueDepth camera provides a much more accurate and three-dimensional map of the user's face.



Figure 4-5. Image captured via infrared camera showing Apple's TrueDepth camera and facial recognition system in action.
Image source from <https://www.idropnews.com/news/app-developer-access-to-truedepth-camera-worries-privacy-advocates/57116/>

4.2.2 Areas of Current AR Development

Persistent Cloud-Based Mapping

One of the current problems with AR is that any captured environment is stored on the local device, which makes managing large maps, revisiting locations, or sharing maps in multiplayer AR situations challenging. Developer 6D.AI offers one solution, as any apps built with its AR SDK can elect to automatically pull from and merge any user-generated map data with the 6D.AI AR Cloud. This cloud-based AR map will grow over time as more consumers use AR apps built with 6D.AI's SDK (6D.AI, 2018).

Real-Time Mesh Construction

Another current shortfall of both ARKit and ARCore is that neither can detect unknown or complex 3D shapes. The 6D.AI SDK, however, can achieve this and even enable a virtual object to interact with more complicated 3D objects in the real world. It does this by creating a 3D mesh of the environment around the viewer in real time (6D.AI, 2018) (see Figure 4-6).

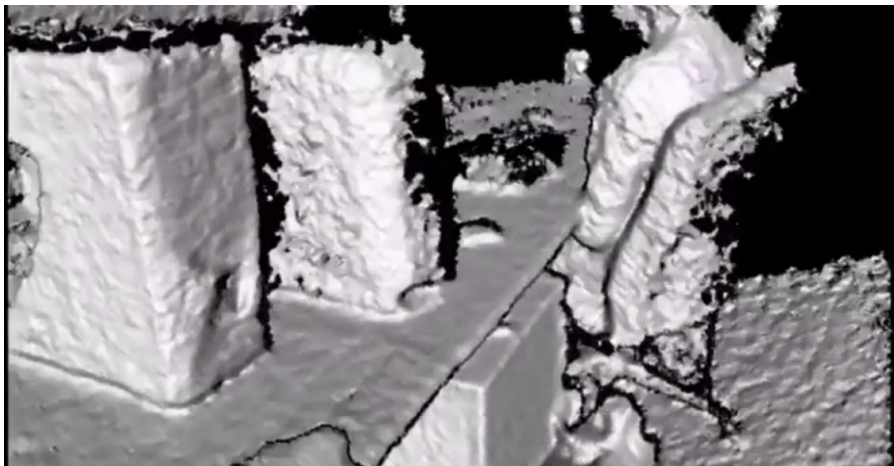


Figure 4-6. Image showing 6D.AI's SDK rendering a 3D mesh in real time. Image sourced from <https://www.techprior.com/6d-ai-revolutionizing-ar-technology-for-3d-mapping/>

4.2.3 Supported Audio Formats

As seen in Chapter 3, AR is typically packaged as a custom-built app experience. The result is that developers typically decide what audio file types and formats to support in their apps. For a more thorough description, see the sections below on game engines, audio middleware, and spatial audio SDKs.

4.3 Virtual Reality

After countless attempts (Taylor, 2016), VR finally may have overcome its biggest technological obstacles and secured a position in the technological mainstream. In 2018, VR companies solidified the three major form factors for VR systems, managed to package motion tracking into two unified solutions, and created controllers that inspire play.

4.3.1 Head-Mounted Displays

Functioning as the portal between the real and virtual worlds, head-mounted displays (HMDs) fall into three categories: PC-tethered, mobile, and standalone.

PC-tethered

PC-tethered HMDs were among the first HMDs. They also can offer the most robust VR experience, because the computer to which the HMD is tethered handles all of the processing (see Figure 4-7). Putting the processing burden on the PC, however, comes with a price: the space a user can explore is restricted to the length of the connecting cables. For those that do not already own a gaming PC, tethered HMDs are also the most expensive option. Gaming laptops with

enough processing power to properly support VR experiences cost a minimum of \$1,000 (Buzzi, 2018).

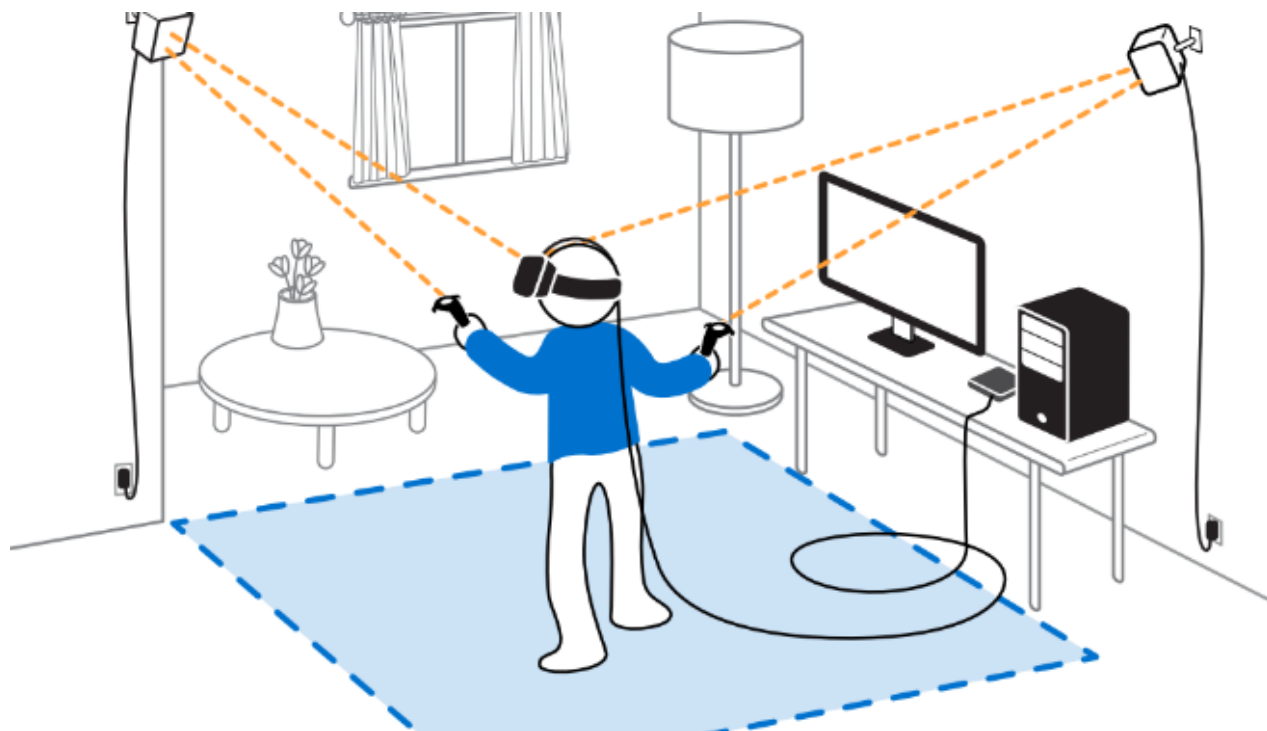


Figure 4-7. Graphic showing a typical PC-tethered VR setup with outside-in tracking. Image sourced from the HTC Vive PRE User Guide (HTC, 2016)

Mobile

Mobile HMDs work in tandem with a specially designed case that allows consumers to use their higher-end smartphones as both the computer and display for a virtual-reality headset. Mobile VR solutions dramatically reduce VR's entry cost and make content more readily available. Crafty consumers can build a cardboard headset for free (see Figure 4-8) (Google, 2016) or purchase a ready-made case like Samsung's Gear VR for around \$40 (Samsung, 2016). Mobile HMDs are cheaper, but they do not offer all of the features or high-end performance of a PC-tethered HMD. Accessories, such as Nolo's CV1 (Nolo, 2018), can add some of the missing features but increase the cost of mobile VR to that of PC-tethered systems.



Figure 4-8. Image showing Google Cardboard. Image sourced from https://store.google.com/product/google_cardboard

Standalone

Standalone systems, like the Oculus Go (Oculus, 2017), are the near future of VR. Standalone VR solutions package everything VR requires into a single HMD. The Oculus Go was the first standalone HMD released in 2018 and, like mobile HMDs, offered only basic VR capabilities. Recently announced HMDs, however, like the HTC Vive Cosmos (HTC, 2019) or the Oculus Quest (Oculus, 2018), manage to pack all of the current and cutting-edge features of VR into a completely standalone HMD.

4.3.2 Degrees of Freedom

Not all HMDs provide the same amount of freedom to move in VR. Using internal sensors like those used in AR, mobile HMDs and early standalone HMDs kept the viewer's perspective locked to a single point in space and offered only three degrees of freedom (3DoF): roll, yaw, and pitch (see Figure 4-9). PC-tethered, accessorized mobile, and newer standalone HMDs allow the viewer to move in an additional three dimensions (6DoF): forward/backward, left/right, and up/down, bringing all of the modes of movement experienced in reality to virtual reality (see Figure 4-9). These three additional dimensions require monitoring the user's actions with either outside-in or inside-out tracking.

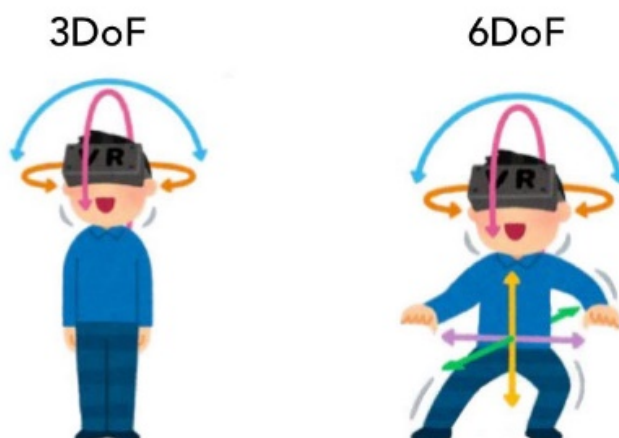


Figure 4-9. Graphic showing the difference between 3DoF and 6DoF. Graphic sourced from <https://www.aniwaa.com/guide/vr-ar/vr-headset-buying-guide/>

4.3.3 Outside-In vs. Inside-Out Tracking

The Oculus Rift and other early 6DoF HMDs use outside-in tracking, where external sensors track the player's movements (see Figure 4-7). In the case of the Oculus Rift, two cameras detect infrared light emitted from the HMD to triangulate the player's position (Durbin, 2017). The HTC Vive reverses this idea, using photosensors on the HMD to capture flashes of non-visible light emitted from two lighthouses. The HTC Vive then calculates the timing difference between different spectrums of received light to determine the player's three-dimensional location (Buckley, 2015).

Outside-in tracking delivers low latency and accurate position tracking even with just two external devices. Adding additional sensors can improve a system's accuracy (Langley, 2017). Yet outside-in tracking also suffers from two significant limitations: occlusions and limited play areas. Outside-in tracking relies upon a direct line of sight between the HMD and the external devices. If a player ducks behind a couch, for example, the outside-in tracking system will fail.

Similarly, if the player moves outside of the trackable area, the outside-in tracking system will fail. These problems are what inspired manufacturers to explore inside-out tracking techniques.

Inside-out tracking systems, like those found on the new HTC Vive Cosmos (see Figure 4-10) or Oculus Quest, are based upon the same principles as AR motion tracking (Langley, 2017). Mounted along the outside surface of the HMD are several ultra-wide-angle cameras that, when paired with a computer vision algorithm, identify landmarks and surfaces in the real world as anchor points for tracking the user's movements (Oculus, 2018). While outside-in tracking likely will still be used for tethered VR systems to provide the user with the best and most accurate experience, inside-out tracking gives a viewer total freedom and can be deployed in both AR and VR, which enables a more seamless flow in mixed reality experiences.



Figure 4-10. Image of the HTC Vive Cosmos, a standalone HMD with inside-out tracking. Image sourced from <https://www.vive.com/eu/cosmos/>

4.3.4 Controllers

Looking around in a virtual world is visually stimulating, but the world won't begin to feel real until a player can reach out and grab a virtual object. Similar to HMDs, controllers are also organized into categories based upon their degrees of freedom: gamepads, 3DoF, and 6DoF.

Gamepads

Mobile HMDs, like the Gear VR, often feature a simple collection of buttons and touchpads for navigating on-screen menus or fundamental in-game interactions (see Figure 4-11). Alternatively, consumers can choose to pair any HMD with any Bluetooth-enabled gamepad, like an Xbox 360 controller. Both controllers work well for playing more traditional video games in a VR setting, but since the gamepad's location and orientation are unknown to the HMD, intuitive interaction with virtual objects using the gamepad is impossible.



Figure 4-11. Image showing the Gear VR's simple onboard controls. Image sourced from <https://groovr.zendesk.com/hc/en-us/articles/115002822988-Using-GrooVR-GearVR>

3DoF Controllers

3DoF controllers, like the one included with the Oculus Go (see Figure 4-12), enable slightly more intuitive virtual interaction by reporting their yaw, pitch, and roll to the HMD. With just these three dimensions, the controller functions similar to a laser pointer, allowing the user to move to specific locations or interact with objects by pointing at them. Usually, these controllers also incorporate touchpads, buttons, and triggers like those found on gamepads.



Figure 4-12. Image of the 3DoF controller included with the Oculus Go. Image sourced from <https://www.oculus.com/go/accessories/>

6DoF Controllers

The controllers that come with the HTC Vive Cosmos are great examples of the current capabilities of VR controllers. Using a combination of each controller's internal sensors and the HMD's inside-out tracking, the Cosmos' controllers have six degrees of freedom and a full set of ergonomically arranged buttons, joysticks, and triggers (see Figure 4-13).



Figure 4-13. Image showing the HTC Vive Cosmos controller. Image sourced from <https://www.vive.com/eu/cosmos/>

Specialized Controllers

In addition to the controllers that typically come with a VR system, creators and consumers can also choose from an ever-expanding field of specialized controllers. Two prime examples are the HTC Vive Tracker and Virtuix Omni.

Initially available only to developers, the HTC Vive Tracker is a small device that can transform anything into a 6DoF VR controller (see Figure 4-14). Already the Tracker has been used to convert tennis rackets, arcade guns, and baseball bats into 6DoF controllers (HTC, 2017).



Figure 4-14. Image of the HTC Vive Tracker. Image sourced from <https://www.vive.com/us/vive-tracker/>

On the opposite end of the spectrum, the Virtuix Omni is a giant controller that allows users to walk and run through a virtual world while staying stationary in the real world (see Figure 4-15). The device locks a user's entire body inside a protective ring and requires him to wear special slip pads on his shoes to reduce friction. Then, by merely walking or running, the devices translates the user's real-world body movements into equivalents in the virtual world.



Figure 4-15. Image of the Virtuix Omni movement controller. Image sourced from <http://www.virtuix.com/product/omni-platform/>

4.3.5 Eye-Tracking

Eye-tracking is a feature that many manufacturers are fixated on, because of the many ways it could streamline resource allocation, improve the overall VR user experience, and create a goldmine of user data. Current HMDs equally distribute their processing power across the entire display. The human eye, on the other hand, focuses only on a small area, leaving what lies in peripheral vision slightly out of focus. Tracking a user's eye movement would allow a HMD to render just the zone that the viewer is focused on, a technique called foveated rendering (Langer,

2018). Foveated rendering could substantially improve device performance, especially on mobile VR systems, where processing power is limited. Additional benefits of eye-tracking could include user identification, automatic display calibration, user intent analytics, and healthcare research (Langer, 2018).

4.3.6 Supported Audio Formats

As seen in Chapter 3, VR is typically packaged as a custom-built app experience. Because of this approach, developers decide what audio file types and formats are supported by their app, and thus there is no widespread standard. The average VR consumer likely will use either an HMD's built-in speakers or headphones. In some individual cases, such as with location-based entertainment (Sag, 2019) or truly dedicated VR consumers, a more sophisticated setup of personal- or external-delivery systems can be used. For a more thorough description, see the sections below on game engines, audio middleware, and spatial audio SDKs.

4.4 Digital Audio Workstations

Digital audio workstations (DAWs) do not necessarily need to evolve to continue to be a part of content production for interactive and immersive digital formats. At this point, game engines, audio middleware, and audio spatialization tools do not support recording, synthesizing, and creating music (or sound design) anywhere near as robustly as DAWs. Furthermore, mono or stereo audio files, which DAWs can efficiently produce, are the preferred source formats for crafting spatial audio.

For those interested in creating spatial audio within a DAW, any DAW that supports mixing in 5.1 surround can handle FOA and basic spatial audio. In addition, many DAW manufacturers have recognized the growing spatial audio needs of their users and have begun adding native support for HOA and spatial audio workflows.

4.4.1 Ableton: Live

Ableton's 2017 acquisition of Cycling '74, the makers of Max/MSP, a graphical programming environment for digital artists, which is now just called Max, came as no surprise to users of the companies' products (Kirn, 2017). Since 2010, Ableton and Cycling '74 have supported a version of Max designed to run inside of Live called Max for Live. Live 10, the first significant update since the acquisition, added new capabilities to Max for Live that simplified the process of working with HOA or any surround audio format.

One example of this is the Envelop for Live pack, a complete suite of audio effects and control devices for creating and performing with HOA (Kirn, 2018). The open source nature of Max patches and Live's distinctive take on the traditional functions and features of a DAW make Live 10 a promising environment for spatial audio experimentation.

4.4.2 Adobe: Audition

Adobe Audition offers many features that make it an attractive environment for editing audio, such as its robust spectral editing tools and noise reduction algorithms. Unfortunately, Audition does not officially support Ambisonics, and like other DAWs, it limits track output

options to mono through 5.1 surround. Audition does allow users to edit audio files that contain up to 32 channels of audio, but it offers no solution for spatializing audio in HOA.

4.4.3 Apple: Logic Pro X

Logic Pro X (10) supports mixing in 5.1 and 7.1 surround, so it should be possible to work with FOA, but no official support has been announced. At this time, those working in Logic have to import their sounds into another DAW to complete their spatial mixes.

4.4.4 Avid: Pro Tools

Beginning with Pro Tools HD 12.8.2, Avid added additional bus and track types with the required number of channels to support up to TOA. The standard version of Pro Tools, however, supports only mono and stereo audio. Like Cockos, Avid also has yet to release Ambisonic plug-ins and instead includes Facebook's free Spatial Workstation plug-ins as part of its standard installation package (Sherbourne, 2017).

4.4.5 Cockos: REAPER

REAPER's unique singular track type, which supports up to 64 channels of audio input and output, makes it the most flexible DAW, as well as the only major application that supports up to seventh-order Ambisonics. Although REAPER does not have any native effects for working with Ambisonic audio it is the go-to DAW for most audio professionals working with HOA.

4.4.6 MOTU: Digital Performer 10

MOTU's recently updated DAW, Digital Performer 10 (DP10), added several significant updates that returned DP10 to its roots as a live performance software. MOTU has not, however, announced any plans for spatial audio beyond DP10's current support for mixing in 10.2 surround. A step above other DAW's 5.1 surround sound, DP10's support for 10.2 surround mixing allows for 12 discrete channels of audio per track, which is enough for SOA but not enough for TOA.

4.4.7 Steinberg: Cubase 10 & Nuendo

With features like Game Audio Connect, which streamlines the process of moving files between Nuendo and Audiokinetic's Wwise (for more information on Wwise, see the dedicated section below), Steinberg has established Nuendo as the premier DAW for spatial and game audio workflows. In 2018, Steinberg added many of the Ambisonic and VR features already available in Nuendo 8 to Cubase 10—like bus tracks for TOA, a binauralizer, a spatial panner, and the option to connect an HMD for in-DAW head-tracking.

4.5 Plug-ins

Recognizing the spatial-audio void left by DAWs, many third-party developers already have released a range of plug-ins for working with Ambisonic audio. The plug-ins examined below, while only a small slice of what is currently available, have been selected based upon their popularity and unique features.

Most Ambisonic plug-ins separate the spatial audio production process into three categories: authoring, imaging, and monitoring (see Figure 4-16). These three stages can be loosely equated with encoding, transforming, and decoding. As a result, Ambisonic plug-ins of different developers can be mixed and matched within each category as long as they use the same Ambisonic order and format.

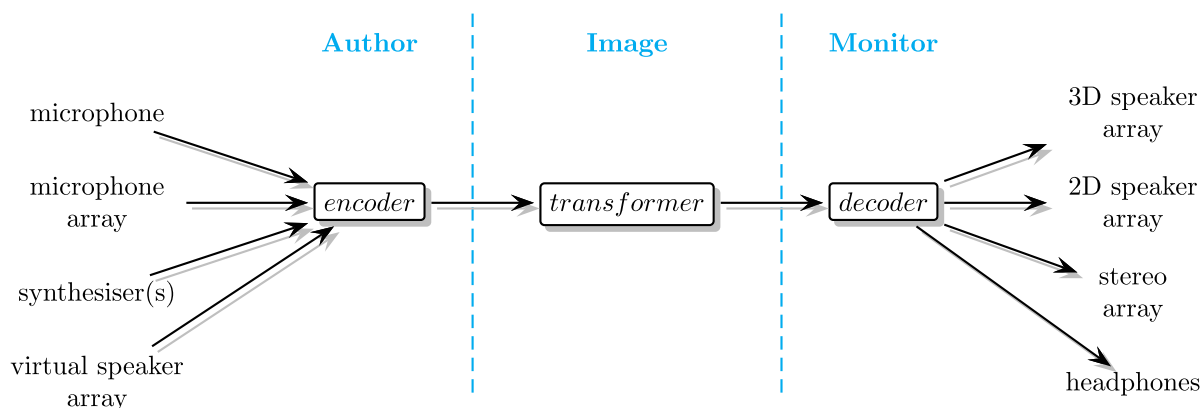


Figure 4-16. The three main stages of the spatial audio production process as outlined on the Ambisonic Tool Kit's website: <http://www.ambisonictoolkit.net/documentation/workflow/>

4.5.1 Facebook: FB360 Spatial Workstation

The Facebook 360 Spatial Workstation is a software suite of free audio plug-ins initially developed by Two Big Ears (Matney, 2016). The tools are capable of decoding, encoding, and embedding head-locked audio and TOA. While the suite is designed to help content creators add spatial audio to Facebook 360 Videos, any of the individual plug-ins could be used with any other third-order ambiX plug-in. The FB360 Spatial Workstation suite is focused around three main plug-ins: the Spatialiser (see Figure 4-17), the Converter (see Figure 4-18), and the Control (see Figure 4-19).

The Spatialiser (see Figure 4-17) works with a range of input formats: mono, stereo, 4.0 through 7.0 surround, first-order B-format (ambiX and FuMA), second-order B-format (ambiX), and third-order B-format (ambiX). The Spatialiser plug-in conveniently shows the spatial placement of all other tracks with the same plug-in and serves as the 360 video host, allowing users to visualize the placement of each sound over the video.

The output of each Spatialiser is routed to a single track that hosts the FB360 Converter plug-in (see Figure 4-18). The Converter transforms the audio from B-format second- or third-order ambiX, or Facebook's 8-channel Spatial Workstation format, to either binaural, first-, second- or Third-Order ambiX, or Spatial Workstation.

The Control plug-in (see Figure 4-19) sets the room dimensions and enables or disables a preview of Mix Focus, an optional feature of Facebook 360 videos and VR applications that reduces the level of sounds positioned outside a definable field of view. More important, however, the Control plug-in synchronizes the 360 video to the DAW's timeline. The remaining two plug-ins are for monitoring the overall loudness of both the spatial and head-locked mixes (see Figure 4-20 and Figure 4-21).

Included as a separate application, the FB360 Encoder (see Figure 4-22) is used in the final stage of preparing a 360 video, where the user can merge the spatial audio, head-locked audio, and video into one file.

For more information on the Facebook Spatial Workstation, see the website (<https://facebook360.fb.com/spatial-workstation/>) or join the active Facebook group (<https://www.facebook.com/groups/1812020965695437/>).

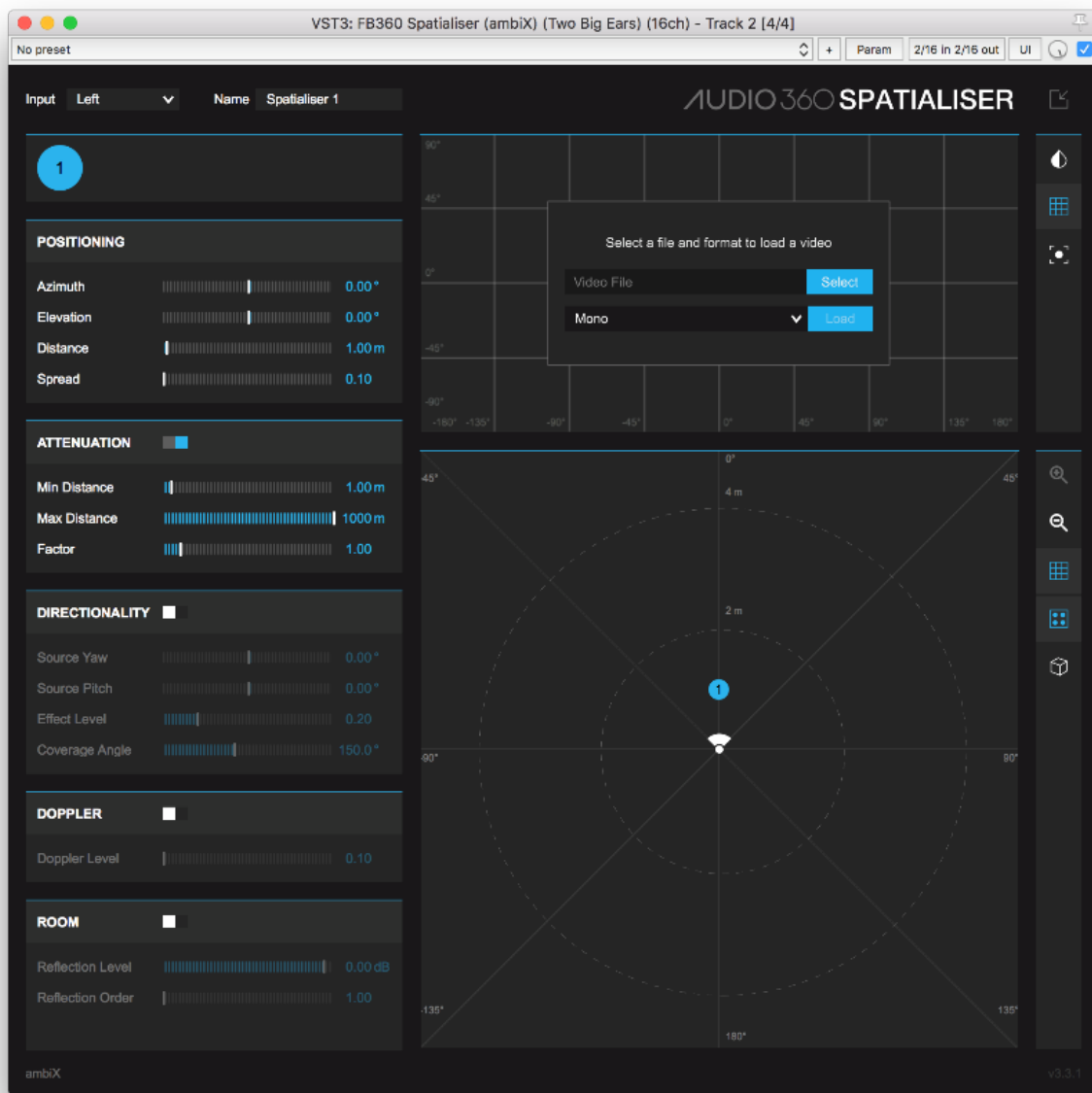


Figure 4-17. Screen capture showing the FB360 Spatial Workstation Spatialiser audio plug-in

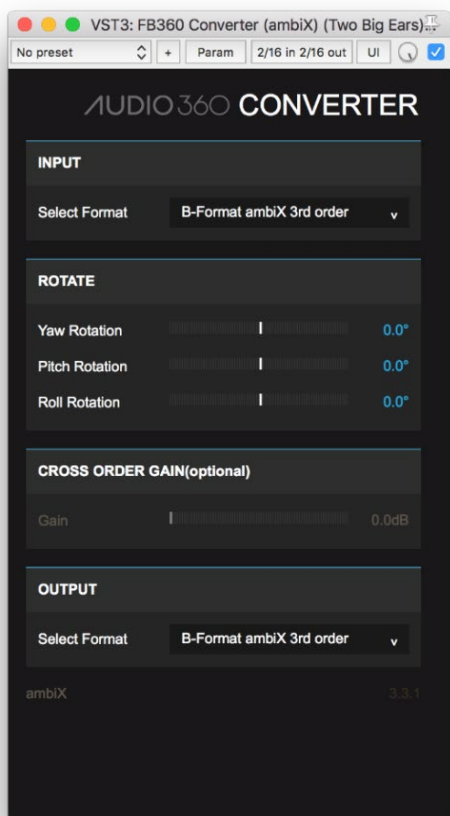


Figure 4-18. Screen capture showing the FB360 Spatial Workstation Converter audio plug-in.

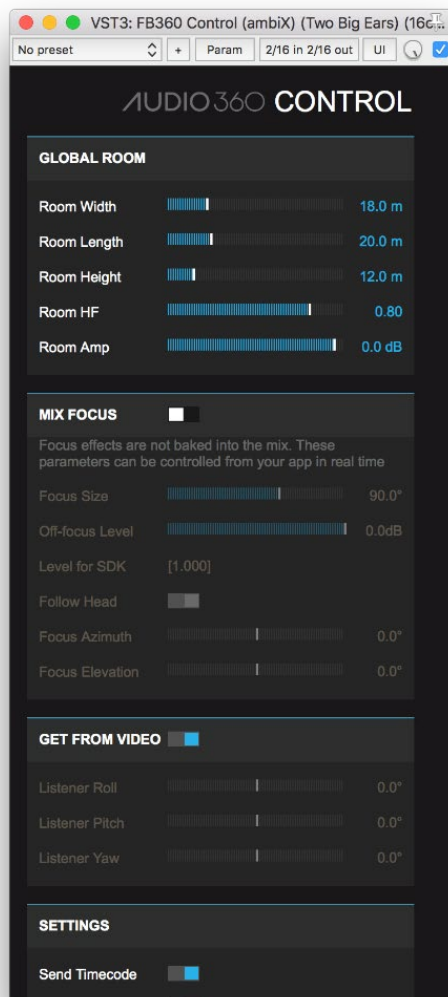


Figure 4-19. Screen capture showing the FB360 Spatial Workstation Control audio plug-in.

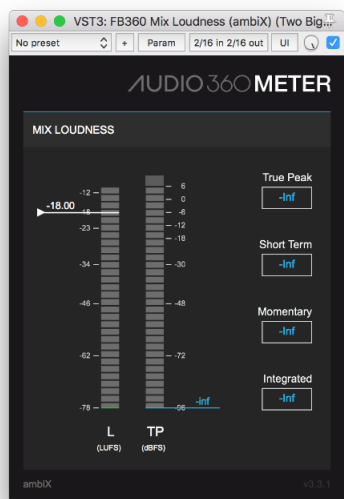


Figure 4-20. Screen capture showing the FB360 Spatial Workstation Meter audio plug-in.

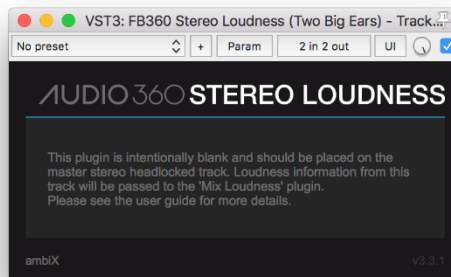


Figure 4-21. Screen capture showing the FB360 Spatial Workstation Stereo Loudness audio plug-in.

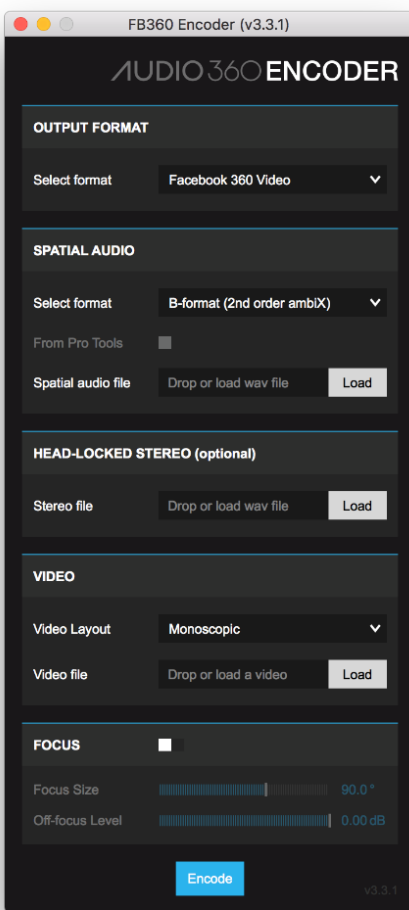


Figure 4-22. Screen capture showing the FB360 Spatial Workstation Encoder application.

4.5.2 Ambisonic Tool Kit (ATK)

The free Ambisonic Tool Kit (ATK) was developed for REAPER and SuperCollider using the Cokkos' JSFX scripting language (Cokkos, 2009). This toolkit provides a variety of encoding, transforming, and decoding functionality for first-order Ambisonics (see Table 4-1).

Some of the most enticing tools within the kit are the transforming plug-ins, like the ATK FOA Transform FocusPressPushZoom plug-in, which provides a unified interface for manipulating the spatial emphasis of an FOA audio recording (see Figure 4-23), or the ATK FOA Transform Mirror plug-in, which can reflect an Ambisonic field across an arbitrary plane (see Figure 4-24).

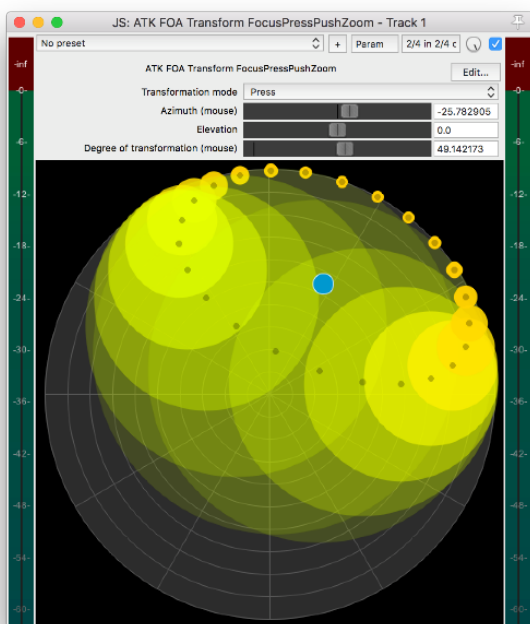


Figure 4-23. Screen capture showing the ATK FOA Transform FocusPressPushZoom plug-in.

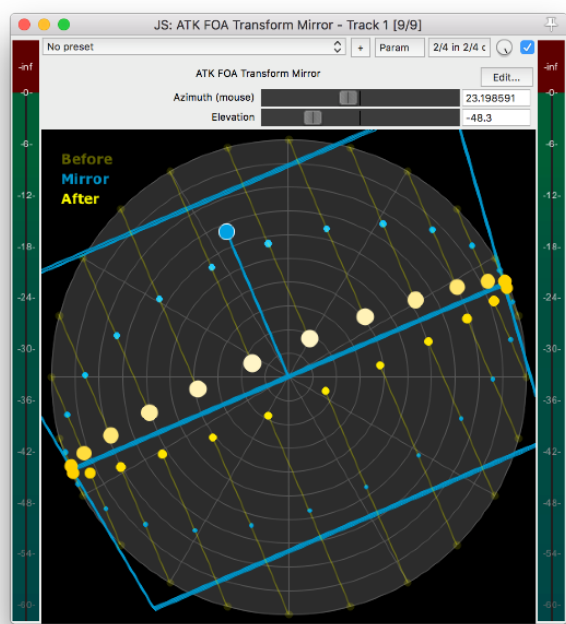


Figure 4-24. Screen capture showing the ATK FOA Transform Mirror plug-in.

Table 4-1. A full list of the plug-in included in the Ambisonic Tool Kit.
 Information found on the Ambisonic Tool Kit website: <http://www.ambisonictoolkit.net/documentation/reaper/>

Encoders	
Mono Signals	
Omni	Encodes a mono signal as an omnidirectional sound field.
Planewave	Provides classic encoding of a mono source as a planewave, where the arrival direction (azimuth and elevation) can be set.
Spread	Encodes a monophonic signal by smoothly rotating the signal across the sound field by frequency.
Diffuse	Randomizes the phase of the incoming monophonic signal to create a diffuse field.
PseudoInverse	Provides a considerable flexibility and can be used with both microphone arrays and synthetic signals. In the absence of a soundfield microphone, this encoding technique gives the opportunity to deploy real-world microphone arrays (omni, cardioid, etc.) to capture natural sound fields. Zoom H2 adapts PseudoInverse for this popular portable recorder. Not yet implemented.
Stereo Signals	
Stereo	The Stereo plugin encodes stereo left and right channels as two plane waves coming from left and right; the angular spread between the two waves is parameterized.
SuperStereo	SuperStereo is the classic “super stereo” method for encoding stereophonic signals.
UHJ	The UHJ encoder offers access to numerous published recordings for periphonic (2D) audition.
Multichannel Signals	
Plane wave transcoders	Plane wave transcoders are provided for a number of fixed-channel distribution formats: quad, 5.0 and 7.0 for the standard ITU layouts, and pantophonic for 2D and periphonic for 3D arrays. Periphonic remains to be implemented.
AtoB	Provides a powerful method for constructing full, complex sound fields via A-format encoding. The user supplies four separate but related signals to be distributed equally throughout the three dimensions of the sound field. These signals may either be synthesized or captured by microphones.
Transformers	
Imaging transforms	
Direct	Adjusts the sound field directivity across a plane specified by the user.
DirectO	Adjusts the directivity of an FOA sound field across the origin, functioning as a spatial low-pass filter; with an increasing degree of transformation, the sound field becomes less directional, eventually becoming omnipresent.
FocusPressPushZoom	Unified interface for four different spatial transforms. Focus and Zoom are dominance-related transforms and “emphasize” elements in the direction of interest. Press and Push differ. Rather than emphasizing elements in a target direction, all are “pressed” or “pushed” toward the direction of interest. With increasing spatial transform, all four transforms collapse the sound field to a plane wave arriving from the direction of interest specified by azimuth and elevation.
Dominance	Increases the gain of elements in the direction of interest while decreasing the gain of elements in the opposite direction.
Mirror	Mirrors the sound field across an arbitrary plane.
MirrorO	Mirrors the sound field through the origin.
RotateTiltTumble	Multi-axes FOA rotations.
Near-field Effects	
Proximity	Introduces the proximity effect to encoded signals.
Nearfield	Reduction or removal of the proximity effect from encoded signals.
Decoders	
Mono decoder	

Mono	This virtual microphone decoder returns a single channel and can be used to “listen in” to the sound field at the specified azimuth and elevation.
Stereo decoders	
Stereo	Returns a pair of virtual microphones.
UHJ	While the virtual microphone stereophonic decoder is very easy and convenient, for production work we advise using the UHJ decoder.
Multichannel decoders	
Quad	Optimized quadraphonic decoder with variable loudspeaker angle.
5.0	Uses Bruce Wiggins optimized ITU 5.0 decoders.
Pantophonic2D	A regular 2D polygon decoder.
Periphonic	A regular cylindrical decoder for 3D dual ring. Not yet implemented.
Diametric	Gerzon’s classic decoder suitable for varied periphonic and pantophonic loudspeaker arrays. Not yet implemented.
BtoA	Decodes B to A-format. Can be used with AtoB and intermittent sound effects for sound design.
Binaural decoder	
Binaural	For headphone listening.
Utilities	
4channels	Extract 4 channel A- or B-format signal from multichannel recording.
MuteSoloChannels	Mute or solo individual channels of a four-channel track.

Visit the Ambisonic Tool Kit website (<http://www.ambisonic toolkit.net/>) for more information on the full line of FOA plugins.

4.5.3 Blue Ripple Sound

Blue Ripple Sound makes several plug-in bundles in both AAX (Avid Pro Tools) and VST2 formats that address the full range of general and specific spatial audio needs.

O3A Core

Blue Ripple’s O3A Core is a free bundle of TOA plug-ins offering the essential functions necessary to create a spatial audio mix. The bundle offers each plug-in for both the AAX and VST2 formats.

Table 4-2. A full list of the plug-ins included in Blue Ripple Sound's O3A Core bundle. Information found on Blue Ripple Sound's website: <https://www.blueripplesound.com/products/o3a-core>

O3A Panner	I/O: 1 in, 16 out This plug-in places a sound in a particular direction in a 3D O3A mix.
O3A Panner — Classic	I/O: 1 in, 16 out This plug-in places a sound in a particular direction in a 3D O3A mix. It uses a simple GUI.
O3A Panner — Hemisphere	I/O: 1 in, 16 out This plug-in places a sound in a particular direction in a 3D O3A mix. It uses a non-standard GUI that some composers prefer.
O3A Panner — Large	I/O: 1 in, 16 out This plug-in places a sound in a particular direction in a 3D O3A mix, using a large control surface.
O3A Panner — Two Channel	I/O: 2 in, 16 out This plug-in places two sounds in particular directions in a 3D O3A mix.
O3A Panner — Eight Channel	I/O: 8 in, 16 out This plug-in places eight sounds in particular directions in a 3D O3A mix.
O3A Decoder — Mono	I/O: 16 in, 1 out This plug-in renders simple mono from 3D O3A mixes.
O3A Decoder — Stereo	I/O: 16 in, 2 out This plug-in renders robust, high-quality stereo from 3D O3A mixes. Please note that this is not a binaural decoder.
O3A Decoder — 5.1 Basic	I/O: 16 in, 6 out This plug-in renders robust, high-quality 5.1 surround mixes from 3D O3A mixes.
O3A Decoder — FuMa	I/O: 16 in, 16 out This plug-in converts an O3A Ambisonic mix to third-order FuMa Ambisonic audio. FuMa is compatible with "classic" B-Format and was used by this software prior to version two.
O3A Injector — FuMa	I/O: 16 in, 16 out This plug-in converts third-order FuMa Ambisonic audio to an O3A Ambisonic mix. FuMa is compatible with "classic" B-Format and was used by this software prior to version two.
O3A Beamer	I/O: 16 in, 1 out This plug-in extracts mono sound in a particular direction from a 3D O3A mix.
O3A Virtual Microphone	I/O: 16 in, 2 out This plug-in simulates a conventional stereo microphone placed within a 3D O3A sound scene.
O3A Gain	I/O: 16 in, 16 out This simple plug-in applies a gain to an O3A stream.
O3A Look	I/O: 16 in, 16 out This plug-in rotates the 3D O3A sound scene as if the listener's head has turned to look in a particular direction.
O3A Rotation	I/O: 16 in, 16 out This plug-in allows a 3D O3A mix to be rotated by any angle.
O3A Spatial Delay	I/O: 16 in, 16 out This plug-in applies a delay effect to an entire O3A scene. The delay includes a feedback loop with scene reflection, rotation, and filtering.
O3A Meter	I/O: 16 in, 16 out This metering plug-in presents real-time information about the sound energy in a O3A mix.
O3A Meter — Signal	I/O: 16 in, 16 out This simple diagnostic plug-in indicates if component signals are present in a O3A mix.

O3A Visualiser	I/O: 16 in, 16 out This plug-in presents a real-time view of a O3A mix, so you can see what you're doing.
O3A Visualiser — Hemisphere	I/O: 16 in, 16 out This plug-in presents a real-time view of a O3A mix, so you can see what you're doing.
O3A Colourizer	I/O: 16 in, 16 out This plug-in presents a real-time view of a O3A mix, so you can see what you're doing. Colors depend on frequency content.
O3A Flare	I/O: 16 in, 16 out This plug-in presents a real-time view of a O3A mix using an active analysis. It can be used with the View/ViewVR applications.

O3A Decoding

Blue Ripple Sound's O3A Decoding provides a bundle of plug-ins for various TOA decoding situations, as well as a TOA limiter and stereo crosstalk cancelation reducer. The bundle retails for £199.00 and is available in both AAX and VST2 formats. Blue Ripple Sound also offers a similar bundle for FOA.

Table 4-3. A full list of the plug-ins included in Blue Ripple Sound's O3A Decoding bundle. Information found on Blue Ripple Sound's website: <https://www.blueripplesound.com/products/o3a-decoding>

O3A Decoder — 5.1	I/O: 16 in, 6 out This plug-in renders high-quality 5.1 surround from 3D O3A mixes. The corner speakers can be set out in a square or at standard ITU angles.
O3A Decoder — 7.1	I/O: 16 in, 8 out This plug-in renders high quality 7.1 surround from 3D O3A mixes.
O3A Decoder — 7.1.2 (Dolby Atmos)	I/O: 16 in, 8 out This plug-in renders high quality 7.1.2 surround from 3D O3A mixes.
O3A Decoder — 3D7.1	I/O: 16 in, 8 out AAX: mapped to 7.1 This plug-in renders high-quality 3D7.1 surround from 3D O3A mixes. This layout uses 7.1 hardware but arranges it differently.
O3A Decoder — Auro-3D 9.1, 10.1, 11.1, 13.1	I/O: 16 in, 10, 11, 12 or 14 out AAX: only 9.1 supported, mapped to 7.1.2 These plug-ins render high-quality Auro-3D 9.1, Auro-3D 10.1, Auro-3D 11.1, or Auro-3D 13.1 surround from 3D O3A mixes. Note the number of channels that are output by this plug-in. These plug-ins do not perform Auro-3D compression.
O3A Decoder — IMAX 5.0, 6.0, 12.0	I/O: 16 in, 5, 6 or 12 out AAX: IMAX 12.0 is not supported. IMAX 6.0 is mapped to conventional 6.0. These plug-ins render high quality IMAX 5.0, IMAX 6.0 or IMAX 12.0 surround from 3D O3A mixes.

O3A Decoder — 22.2	I/O: 16 in, 24 out AAX: not supported This plug-in renders high-quality Hamasaki 22.2 surround from 3D O3A mixes.
O3A Decoder — Cube	I/O: 16 in, 8 out AAX: mapped to 7.1 This plug-in renders high-quality eight-channel surround for a cube array from 3D O3A mixes.
O3A Decoder — Hexagon	I/O: 16 in, 6 out AAX: mapped to 6.0 This plug-in renders high-quality six-channel surround for a hexagon array from 3D O3A mixes.
O3A Decoder — Octagon	I/O: 16 in, 8 out AAX: mapped to 7.1 This plug-in renders high-quality eight-channel surround for an octagon array from 3D O3A mixes.
O3A Decoder — Quad	I/O: 16 in, 4 out This plug-in renders high quality four-channel surround for a quad array from 3D O3A mixes.
O3A Decoder — Headphones	I/O: 16 in, 2 out This plug-in synthesizes high quality binaural headphone audio from 3D O3A mixes.
O3A Decoder — Quad Binaural	I/O: 16 in, 2 out This plug-in produces simplified binaural audio for use with Quad Binaural systems.
O3A Decoder — UHJ Stereo	I/O: 16 in, 2 out This plug-in renders classic Ambisonic UHJ stereo from 3D O3A mixes.
O3A Decoder — N3D	I/O: 16 in, 16 out This plug-in converts from the SN3D Ambisonic format used by O3A to the N3D Ambisonic convention (using ACN).
O3A Meter — Karma	I/O: 16 in, 16 out This plug-in estimates the perceived loudness of a O3A mix in a style similar to that used in BS.1770, R128, and A/85.
O3A Brickwall	I/O: 16 in, 16 out This plug-in acts as a "brickwall" limiter, keeping the signal peak below a chosen level.
Stereo Crosstalk Cancellation	This plug-in reduces stereo crosstalk at a small sweet spot to increase the perceived stereo width there. This can produce a surprisingly wide stereo image (at a rather small sweet spot) from stereo speakers.

O3A Harpex Upsampler

The O3A Harpex Upsampler is specifically designed to resynthesize FOA B-format files as TOA based upon the estimated direction of a sound in the FOA file. The plug-in, which retails for £119.00, offers meters showing the spatial energy before and after the translation process (see Figure 4-25) and is available in both AAX and VST2 formats.



Figure 4-25. Screen capture of O3A Harpex Upsampler plug-in by Blue Ripple Sound.

O3A Manipulators

Similar to the ATK's Transformer, Blue Ripple Sound's O3A Manipulators bundle moves beyond basic TOA functions to more creative space and sound design tools. The bundle retails for £399.00 and is available in both VST2 and AAX, although not all of the plug-ins in the bundle are AAX compatible.

Table 4-4. A full list of the plug-ins included in Blue Ripple Sound's O3A Manipulators bundle. Information found on Blue Ripple Sound's website: <https://www.blueripplesound.com/products/o3a-manipulators>

O3A B->A20 and A20->B Converters	I/O: 16/20 in, 20/16 out AAX: not supported These plug-ins convert a 3D O3A mix to a spatially sampled 20-channel format suitable for non-linear processing and back again.
O3A Compressor	I/O: 16 in, 16 out This plug-in reduces the dynamic range of an O3A mix. Its directivity control allows different directions to be compressed together or separately.
O3A Diffuser	I/O: 16 in, 16 out This plug-in smudges 3D O3A mixes in time and space to increase their diffuseness. It can also be used to place a mono sound into a O3A mix without giving it a definite direction.
O3A Directional Emphasis	I/O: 16 in, 16 out This plug-in emphasizes sounds in a particular direction in a 3D O3A mix using an algorithm also used in Rapture3D v3.
O3A Directional Mask	I/O: 16 in, 16 out This plug-in emphasizes or de-emphasizes sounds in a particular direction in a 3D O3A mix using simple controls.
O3A Graphic Equalizer	I/O: 16 in, 16 out This plug-in applies a 31-band 1/3-octave graphic equalizer to an entire 3D O3A mix.
O3A Join	I/O: 32 in, 16 out AAX: not supported This plug-in reassembles "Split" audio (e.g. from O3A Spatial Mask Split or O3A Spotlight Split) into a single 3D O3A mix.
O3A Line	I/O: 1 in, 16 out

	This plug-in converts mono to third-order Ambisonics (O3A), spreading sound out in a line.
O3A Low/High-Pass Filter	I/O: 16 in, 16 out This plug-in applies a low- or high-pass Butterworth filter to an entire O3A scene.
O3A Move	I/O: 16 in, 16 out This plug-in warps a 3D O3A mix to move sounds around.
O3A Order Amplifier	I/O: 16 in, 16 out This plug-in allows gains to be applied to each Ambisonic order in a 3D mix.
O3A Panner LS	I/O: 1 in, 16 out This 3D panner simulates large sources and distance.
O3A Panner LS XYZ	I/O: 1 in, 16 out This 3D panner simulates large sources and distance. This plug-in is essentially the same as the O3A Panner LS, except it exposes X, Y, and Z automation parameters rather than azimuth, elevation, and distance.
O3A Parametric Equalizer	I/O: 16 in, 16 out This plug-in applies a simple parametric equalizer to an entire 3D O3A mix.
O3A Portal	I/O: 16 in, 16 out This plug-in shifts a region in an audio scene to another direction.
O3A Reflection	I/O: 16 in, 16 out This plug-in moves all sounds in a 3D O3A sound field from one side to another.
O3A Screen Resizer	I/O: 16 in, 16 out This plug-in warps a O3A scene, which matches a screen image to compensate for a change in screen size.
O3A Spatial Equalizer	I/O: 16 in, 16 out This plug-in applies a 31-band 1/3-octave graphic equalizer to particular directions in a 3D O3A mix.
O3A Spatial Mask	I/O: 16 in, 16 out This plug-in applies a level change to particular directions in a 3D O3A mix.
O3A Spatial Mask Split	I/O: 16 in, 32 out AAX: not supported This plug-in splits directions into two separate 3D O3A mixes that can be reassembled with O3A Join.
O3A Spotlight	I/O: 16 in, 16 out This plug-in keeps sounds in the "beam" of the spotlight and makes others disappear.
O3A Spotlight Split	I/O: 16 in, 32 out AAX: not supported This plug-in splits directions into two separate 3D O3A mixes that can be reassembled with O3A Join.
O3A Zoom	I/O: 16 in, 16 out This plug-in allows an entire O3A scene to be moved around, for instance to give the impression that the audience is entering a space.
O3A Zoom XYZ	I/O: 16 in, 16 out This plug-in allows an entire O3A scene to be moved around. This plugin is essentially the same as O3A Zoom, except it exposes X, Y, and Z automation parameters rather than azimuth, elevation, and distance.

O3A Reverb

The O3A Reverb bundle offers several modular and complete TOA reverb solutions with both algorithmic and convolution options. The bundle retails for £349.00 and is available in both VST2 and AAX formats.

Table 4-5. A full list of the plug-ins included in Blue Ripple Sound's O3A Manipulators bundle. Information found on Blue Ripple Sound's website: <https://www.blueripplesound.com/products/o3a-reverb>

O3A Early Reflection	I/O: 16 in, 16 out This plug-in simulates a single early reflection but applies it to a complete 3D O3A audio scene.
O3A Reverb — MS3	I/O: 16 in, 16 out This plug-in adds high-quality synthetic reverb to a 3D O3A sound field. It uses the same algorithms as the MS5 reverb but has a simplified control interface.
O3A Reverb — MS5	I/O: 16 in, 16 out This plug-in adds high-quality synthetic reverb to a 3D O3A sound field using a highly configurable, directionally dependent early reflection model and a rich late-reflection model.
O3A Reverb — Convolution	I/O: 16 in (but mono input to the convolution core), 16 out This plug-in performs convolution reverb based on a WAV file with up to sixteen channels.
O3A Reverb — Shaped Convolution	I/O: 16 in, 16 out This plug-in performs convolution reverb based on a WAV file with up to sixteen channels. The reverb output can be shaped by the input directivity, so the spatial output is not determined only by the WAV file.
O3A Shoebox	I/O: 1 in, 16 out The O3A Shoebox operates like a panner but simulates up to 128 dynamic acoustic reflections off the walls of a virtual shoebox-shaped room.

O3A Upmixers

In contrast to Blue Ripple Sound's O3A Decoding bundle, the O3A Upmixer TOA bundle provides a variety of tools for transforming audio from mono, stereo, or surround formats to TOA. The bundle retails for £299.00 and is available in both AAX and VST2 formats.

Table 4-6. A full list of the plug-ins included in Blue Ripple Sound's O3A Manipulators bundle. Information found on Blue Ripple Sound's website: <https://www.blueripplesound.com/products/o3a-upmixers>

O3A Upmixer — 5.1	I/O: 6 in, 16 out This plug-in converts 5.1 surround into a 2D O3A mix.
O3A Upmixer — 6.1	I/O: 7 in, 16 out

	This plug-in converts 6.1 surround into a 2D O3A mix.
O3A Upmixer — 7.1	I/O: 8 in, 16 out This plug-in converts 7.1 surround into a 2D O3A mix.
O3A Upmixer — 7.1.2 (Dolby Atmos)	I/O: 10 in, 16 out This plug-in converts 7.1.2 surround into a 3D O3A mix.
O3A Upmixers — Auro-3D 9.1, 10.1, 11.1, 13.1	I/O: 10, 11, 12 or 14 in, 16 out AAX: only 9.1 supported, mapped to 7.1.2 This plug-in converts Auro-3D surround into a 3D O3A mix. Note the number of channels that need to be input to this plug-in. The plug-in does not unpack compressed Auro-3D audio.
O3A Upmixers — IMAX 5.0, 6.0, 12.0	I/O: 5, 6 or 12 in, 16 out AAX: IMAX 12.0 is not supported. IMAX 6.0 is mapped to conventional 6.0 This plug-in converts IMAX surround into a 3D O3A mix.
O3A Upmixer — 22.2	I/O: 24 in, 16 out AAX: not supported This plug-in converts Hamasaki 22.2 surround into a 3D O3A mix. Note that this plug-in needs even more channels than usual.
O3A Upmixer — Quad	I/O: 4 in, 16 out This plug-in converts quad surround into a 2D O3A mix.
O3A Upmixer — Stereo	I/O: 2 in, 16 out This plug-in spreads a stereo image dynamically over a region of space.
O3A Upmixer — Coincident Microphone	I/O: 4 in, 16 out AAX: input mapped to quad This plug-in processes recordings from 2D coincident microphone systems with simple polar patterns, such as the Zoom H2 or Milab SRND 360, or Double M/S systems. It converts to first-order Ambisonics and then to O3A, applying some gentle processing if required. The results from this plug-in are less sharp than from proper O3A panning, and these systems do not record height information. However, this can be a good way to add ambient field recordings into a O3A mix.
O3A Injector — First Order	I/O: 4 in, 16 out This plug-in applies some gentle processing to first-order Ambisonic material that sometimes helps when injecting it into a third-order O3A mix.
O3A Injector — N3D	I/O: 16 in, 16 out This plug-in converts higher-order Ambisonic (HOA) material using the N3D convention (in ACN) to the SN3D convention used by O3A.
O3A Mono Ambience	I/O: 1 in, 16 out This plug-in spreads a mono signal in all directions.
O3A Stereo Ambience	I/O: 2 in, 16 out This plug-in spreads a stereo signal in all directions, or over regions on the left and right.

Binaural Surround

A surround speaker system is not always available while working on a surround mix. Because of this, and the various other reasons someone may need to translate a surround mix to a headphone (binaural) mix, Blue Ripple Sound offers the Binaural Surround plug-in bundle. The bundle retails for £399.00 and is available in AAX and VST2 formats.

Table 4-7. A full list of the plug-ins included in Blue Ripple Sound's Binaural Surround bundle. Information found on Blue Ripple Sound's website: <https://www.blueripplesound.com/products/binaural-surround>

Binaural Surround — 5.1	I/O: 6 in, 2 out This plug-in converts 5.1 into a 2D binaural headphone mix.
Binaural Surround — 6.1	I/O: 7 in, 2 out This plug-in converts 6.1 into a 2D binaural headphone mix.
Binaural Surround — 7.1	I/O: 8 in, 2 out This plug-in converts 7.1 into a 2D binaural headphone mix.
Binaural Surround — 7.1.2	I/O: 10 in, 2 out This plug-in converts 7.1.2 into a 2D binaural headphone mix. Please note that this is not a Dolby Atmos renderer.
Binaural Surround — Auro-3D 9.1, 10.1, 11.1 and 13.1	I/O: 10, 11, 12 or 14 in, 2 out AAX: only 9.1 supported, mapped to 7.1.2 This plug-in converts Auro-3D into a 3D binaural headphone mix. Note the number of channels that need to be input to this plug-in. The plug-in does not unpack compressed Auro-3D audio.
Binaural Surround — 22.2	I/O: 24 in, 2 out AAX: not supported This plug-in converts 22.2 into a 3D binaural headphone mix. Please note the huge input channel requirement here.
Stereo Crosstalk Cancellation	I/O: 2 in, 2 out This plug-in reduces stereo crosstalk at a (rather small) sweet spot to increase the perceived stereo width there. This can produce a surprisingly wide stereo image from stereo speakers.

O3A View

Blue Ripple Sound's O3A View is a 360 Video/VR companion app that synchronizes to a DAW over a local network connection via the O3A View Sync plug-in (see Figure 4-26). The View app allows users to control many of the other O3A plug-ins (see Table 4-8) while either watching a 360 video (see Figure 4-27) on the same or networked computer, or in VR while wearing a HMD (see Figure 4-28).

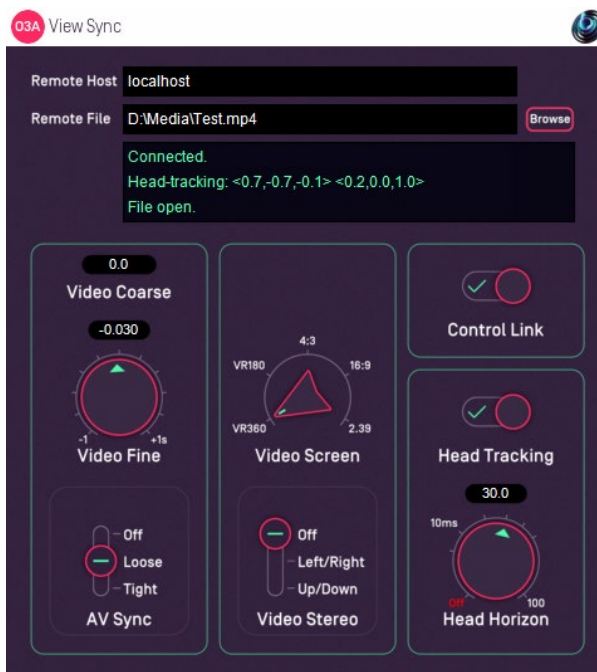


Figure 4-26. Screen capture of Blue Ripple Sound's O3A View Sync plug-in. Image sourced from: <https://www.blueripplesound.com/products/o3a-view>

Table 4-8. A full list of the plug-ins currently controllable via Blue Ripple Sound's O3A View app. Information found on Blue Ripple Sound's website: <https://www.blueripplesound.com/products/o3a-view>

O3A Core	O3A Beamer, O3A Flare, O3A Look, O3A Panner, O3A Panner — Classic, O3A Panner — Hemisphere, O3A Panner — Large, O3A Panner — Two Channel, O3A Panner — Eight Channel, O3A Rotation
O3A Manipulators	O3A Directional Emphasis, O3A Directional Mask, O3A Move, O3A Panner LS, O3A Panner LS XYZ, O3A Portal, O3A Reflection, O3A Spotlight, O3A Spotlight Split, O3A Zoom, O3A Zoom XYZ
O3A Reverb	O3A Shoebox
O3A Upmixers	O3A Upmixer — Stereo
O3A View	O3A Panner 16, O3A Panner 32



Figure 4-27. Screen capture of editing a spatial mix while viewing a 2D representation of a 360 video via Blue Ripple Sound’s O3A View app. Image source from: <https://www.blueripplesound.com/products/o3a-view>

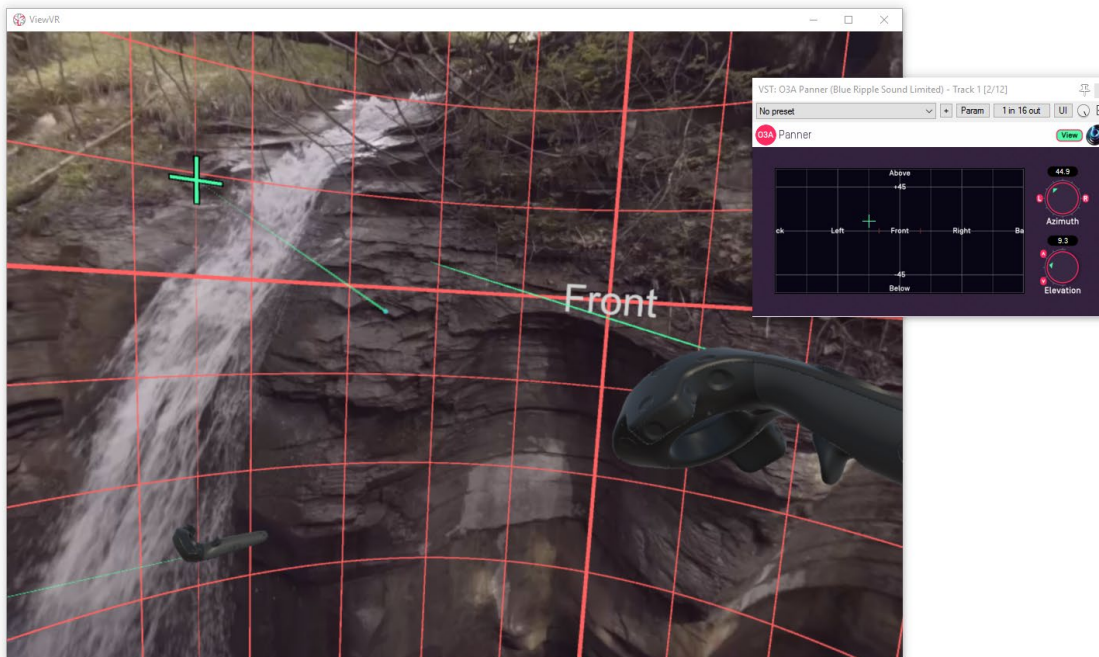


Figure 4-28. Screen capture of editing a spatial mix with a HMD and 6DoF controller via Blue Ripple Sound’s O3A View app. Image source from: <https://www.blueripplesound.com/products/o3a-view>

Rapture3D Advanced

Retailing at £599.00, Blue Ripple Sound's Rapture3D Advanced offers a configuration tool (see Figure 4-29) and two plug-ins for those that need to decode spatial audio to custom speaker arrangements and sound systems.

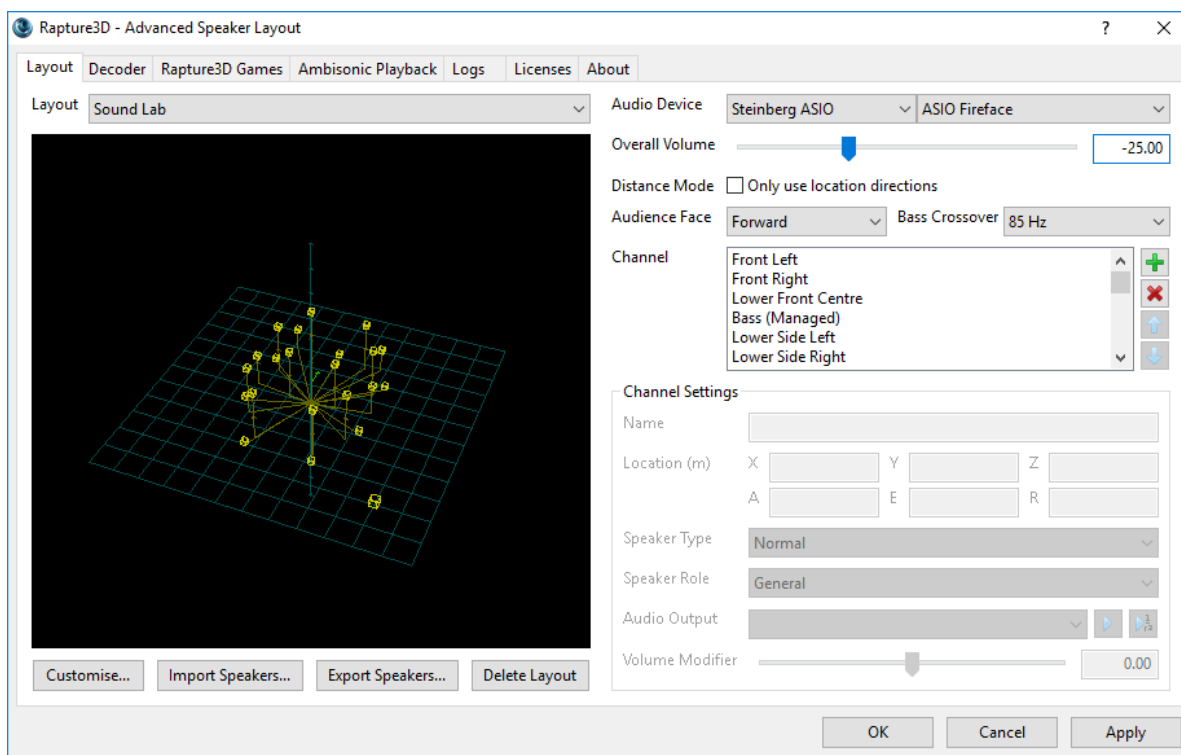


Figure 4-29. Screen capture of the configuration tool in Blue Ripple Sound's Rapture3D Advanced. Image source from: <https://www.blueripplesound.com/products/rapture-3d-advanced>

Table 4-9. A list of the plug-ins included with Blue Ripple Sound's Rapture3D Advanced. Information found on Blue Ripple Sound's website: <https://www.blueripplesound.com/products/rapture-3d-advanced>

O3A Decoder — Rapture3D Advanced	I/O: 16 in, 16, 32, 64 or 128 out AAX: only 16-channel output supported, mapped to third-order Ambisonics This plug-in renders high-quality multichannel audio from a first-, second- or third-order Ambisonic input. It is configured in a separate application.
O3A Decoder Block — Rapture3D Advanced	I/O: 16 in, 8 out AAX: output mapped to 7.1 This plug-in renders high-quality multichannel audio from a first-, second- or third-order Ambisonic input. It is configured in a separate application. Multiple plug-ins can be used together to achieve high channel counts.

4.5.4 Dear Reality

One of the newer spatial audio plug-in developers, Dear Reality has a line of products targeting DAWs, audio middleware, game engines, and VR mixing controllers.

DAW Plug-ins

Available in AU, VST, and AAX plug-in formats, Dear Reality offers three versions of its dearVR plugin. The free Ambi Micro version (see Figure 4-30) is the simplest, and while it only has a limited set of features, it still offers four binaural output modes, head-tracking when coupled with Dear Reality's Spatial Connect, and an A-to B-format converter. The mid-tier version, called dearVR Music, retails for \$199 and adds more robust position capabilities, Ambisonic output options, and an Ambisonic reverb effect (see Figure 4-31). The Pro version of the plug-in retails for \$349 and includes additional reverb controls, more output and input options, and occlusion modeling (see Figure 4-32).



Figure 4-30. Picture showing dearVR's free Ambi Micro plug-in. Image sourced from: <https://www.dearvr.com/products/ambi-micro>



Figure 4-31. Screen capture showing the dearVR Music plug-in. Image sourced from: <https://www.dearvr.com/products/dearvr-music>



Figure 4-32. Screen capture showing the dearVR Pro plug-in. Image sourced from: <https://www.dearvr.com/products/dearvr-pro>

Unlike the other plug-ins discussed so far, dearVR packages the encoding, transformation, and decoding stages into a single plug-in that makes it compatible with any DAW, as long as the final decoded output is for an audio format supported by the DAW (e.g., mono, stereo, 5.1 surround).

VR Mix Control

Similar to Blue Ripple Sound's O3A View, Dear Reality's dearVR Spatial Connect allows users to create spatial mixes in a HMD and 6DoF controllers with any DAW (see Figure 4-33). The controller app lets users control a DAW's transport, automation modes, plug-in presets, and mixer settings.



Figure 4-33. Graphic showing the basic concept and functions of dearVR Spatial Connect. Image sourced from: <https://www.dearvr.com/products/dearvr-spatial-connect>

4.6 Game Engines

Game engines provide developers with the tools and workflows necessary to build anything from simple 2D games to real-time filmmaking for cinematic experiences (Unity, 2018). Unity and Unreal Engine are the two most popular game engines (TNW, 2016), with Unity claiming to power 50% of mobile games, 60% of AR/VR content, and 70% of the interactive content featured at the Sundance Film Festival's *New Frontier* exhibits (Unity, 2014). Unity and

Unreal Engine both organize the creative process by breaking projects into scenes and populating those scenes with objects that consist of a variety of asset (file) types.

A modern game engine must provide developers with high-end and real-time graphic design and rendering tools, support for online interactions and content, the ability to define physics-based simulations, audio and music controls, and, finally, a scripting environment for developing custom logic and object behavior.

4.6.1 Sources and Listeners

Following a game object paradigm, a scene usually contains two types of sonic game objects: sources and listeners. Players are often both listeners and sound sources, hearing the sounds emitted from other players or game objects and producing sounds of their own. Attaching a sound component to a game object enables the game engine's audio system to control and spatialize the sound relative to the associated game object. Each game engine handles this process differently but typically the attached audio component can be either 2D or 3D. A 2D sound is the equivalent of head-locked or traditional stereo audio. A 3D sound, on the other hand, opens up a few more possibilities for manipulating the perception of the sound as a listener moves around it.

4.6.2 Unity

The Unity game engine provides a DAW-like workflow to authoring audio experiences. Creators can design immersive and reactive sonic worlds without needing to dive too deeply into Unity's C# scripting capabilities. To truly take advantage of all of Unity's audio functionality,

however, users need to be able to write at least basic custom scripts, even if working with an audio middleware like FMOD or Wwise (see section 4.7 on audio middleware).

Audio Source Component

Unity supports audio sources of a wide range of file formats (see Table 4-10) that can be added to a game object and customized for specific uses via the available parameters (see Figure 4-34). For 3D or spatialized sounds, Unity automatically links a listener's distance to the audio source's volume, spatial blend, spread, a low-pass filter, and the amount of reverb for specially defined zones (Unity, 2014) (see Figure 4-34).

Table 4-10. List of Unity's supported audio file formats. Information gathered from Unity Developer Documentation: <https://docs.unity3d.com/Manual/AudioFiles.html>

Format	Extensions
MPEG layer 3	.mp3
Ogg Vorbis	.ogg
Microsoft Wave	.wav
Audio Interchange File Format	.aiff/.aif
Ultimate Soundtracker module	.mod
Impulse Tracker module	.it
Scream Tracker module	.s3m
FastTracker 2 module	.xm

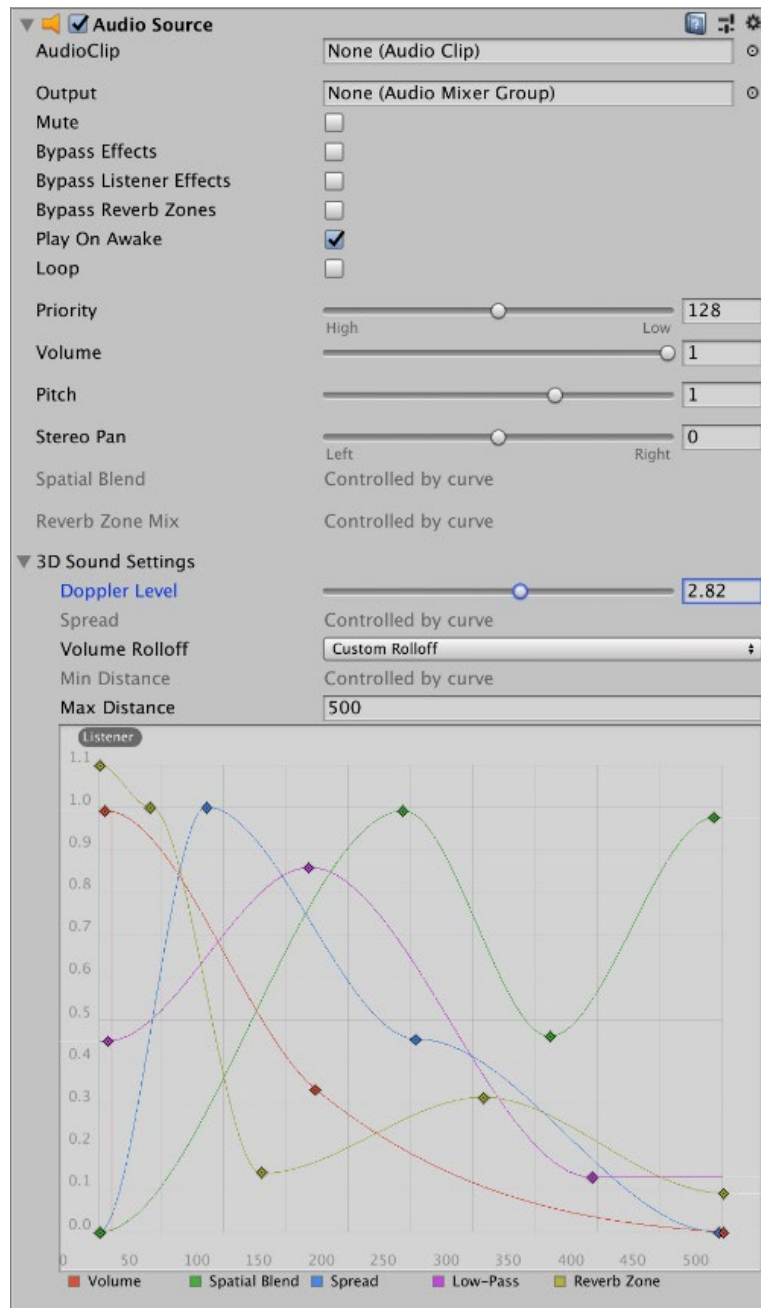


Figure 4-34. Screen capture of an empty audio component in Unity.

Microphone Input

As of version 2018.3, Unity can access a device's microphone, but the microphone can only be activated and controlled only C# scripting (Unity, 2014).

MIDI and OSC Input/Output

Unity offers no native support for MIDI or Open Sound Control (OSC) input or output. Several third-party libraries, however, such as Atsushi Eno's *midi-manage* (Eno, 2010) or Thomas Fredericks's *UnityOSC* (Fredrick, 2018), can extend Unity's scripting capabilities to include these types of connections.

Audio Effects and Audio Effect Components

Unity allows audio effects to be inserted either as an audio effect component directly on a game object that hosts an audio source component (see Table 4-11) or as an insert on a track within Unity's included audio mixer (see Table 4-12).

Table 4-11. Unity's included audio effect components. Information gathered from Unity's developer documentation: <https://docs.unity3d.com/Manual/class-AudioEffect.html>

Audio Low-Pass Filter	The Audio Low-Pass Filter passes low frequencies of an audio source or all sound reaching an audio listener while removing frequencies higher than the cutoff frequency.
Audio High-Pass Filter	The Audio High-Pass Filter passes high frequencies of an audio source and cuts off signals with frequencies lower than the cutoff frequency.
Audio Echo Filter	The Audio Echo Filter repeats a sound after a given delay, attenuating the repetitions based on the decay ratio.
Audio Distortion Filter	The Audio Distortion Filter distorts the sound from an audio source or sounds reaching the audio listener.
Audio Reverb Filter	The Audio Reverb Filter takes an audio clip and distorts it to create a custom reverb effect.
Audio Chorus Filter	The Audio Chorus Filter takes an audio clip and processes it, creating a chorus effect.

Table 4-12. Unity's included audio effects. Information gathered from Unity's developer documentation: <https://docs.unity3d.com/Manual/class-AudioEffectMixer.html>

Audio Low-Pass Effect	The Audio Low-Pass Effect passes low frequencies of an audio mixer group while removing frequencies higher than the cutoff frequency.
Audio High-Pass Effect	The Audio High-Pass Effect passes high frequencies of an audio mixer group and cuts off signals with frequencies lower than the cutoff frequency.

Audio Echo Effect	The Audio Echo Effect repeats a sound after a given delay, attenuating the repetitions based on the Decay Ratio.
Audio Flange Effect	The Audio Flange Effect is used to create the audio effect produced by mixing two identical signals together, one signal delayed by a small and gradually changing period, usually smaller than 20 milliseconds.
Audio Distortion Effect	The Distortion Effect distorts the sound from an audio mixer group.
Audio Normalize Effect	The Audio Normalize Effect applies a constant amount of gain to an audio stream to bring the average or peak amplitude to a target level.
Audio Parametric Equalizer Effect	The Audio Param EQ Effect is used to alter the frequency response of an audio system using linear filters.
Audio Pitch Shifter Effect	The Audio Pitch Shifter Effect is used to shift a signal up or down in pitch.
Audio Chorus Effect	The Audio Chorus Effect takes an audio mixer group output and processes it, creating a chorus effect.
Audio Compressor Effect	The Audio Compressor Effect reduces the volume of loud sounds or amplifies quiet sounds by narrowing or “compressing” an audio signal’s dynamic range.
Audio SFX Reverb Effect	The SFX Reverb Effect takes the output of an Audio Mixer group and distorts it to create a custom reverb effect.
Audio Low-Pass Simple Effect	The Audio Low-Pass Simple Effect passes low frequencies of an audio Mixer group while removing frequencies higher than the cutoff frequency.
Audio High-Pass Simple Effect	The Audio High-Pass Simple Effect passes high frequencies of an audio mixer group and cuts off signals with frequencies lower than the cutoff frequency.

Mixing

An audio component can be routed directly to any track established in Unity’s flexible audio mixer (see Figure 4-35). Signal flow through the mixer follows a parent-child hierarchy constructed along the left edge of the Mixer Window (see Figure 4-36). In addition to the audio effects already mentioned, multiple send and receive audio effects can be inserted on any track to allow for typical reverb or volume ducking applications. Finally, creators can save for each mixer an infinite number of Snapshots, which can be recalled during run-time via dedicated script callbacks.

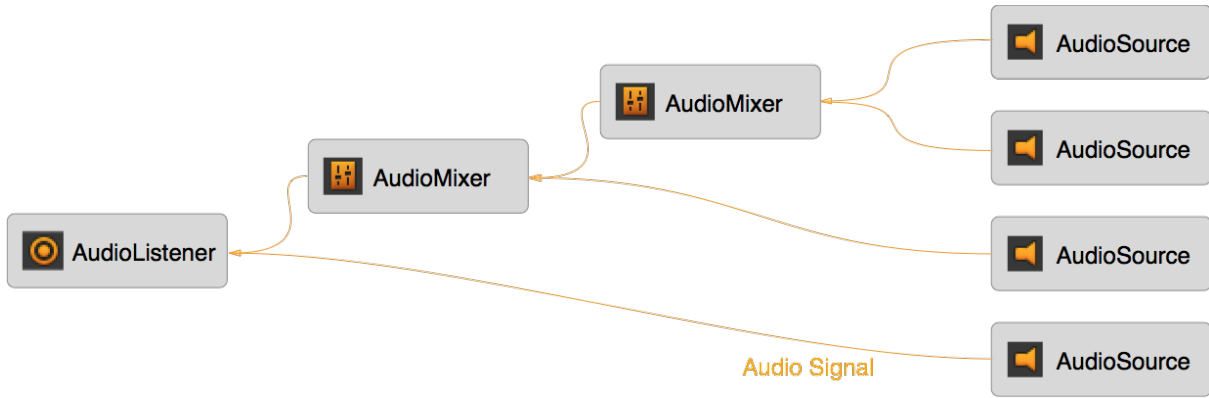


Figure 4-35. Graphic showing a possible signal flow from a sound source to a listener through Unity's mixer. Graphic sourced from Unity's developer documentation: <https://docs.unity3d.com/Manual/AudioMixerOverview.html>

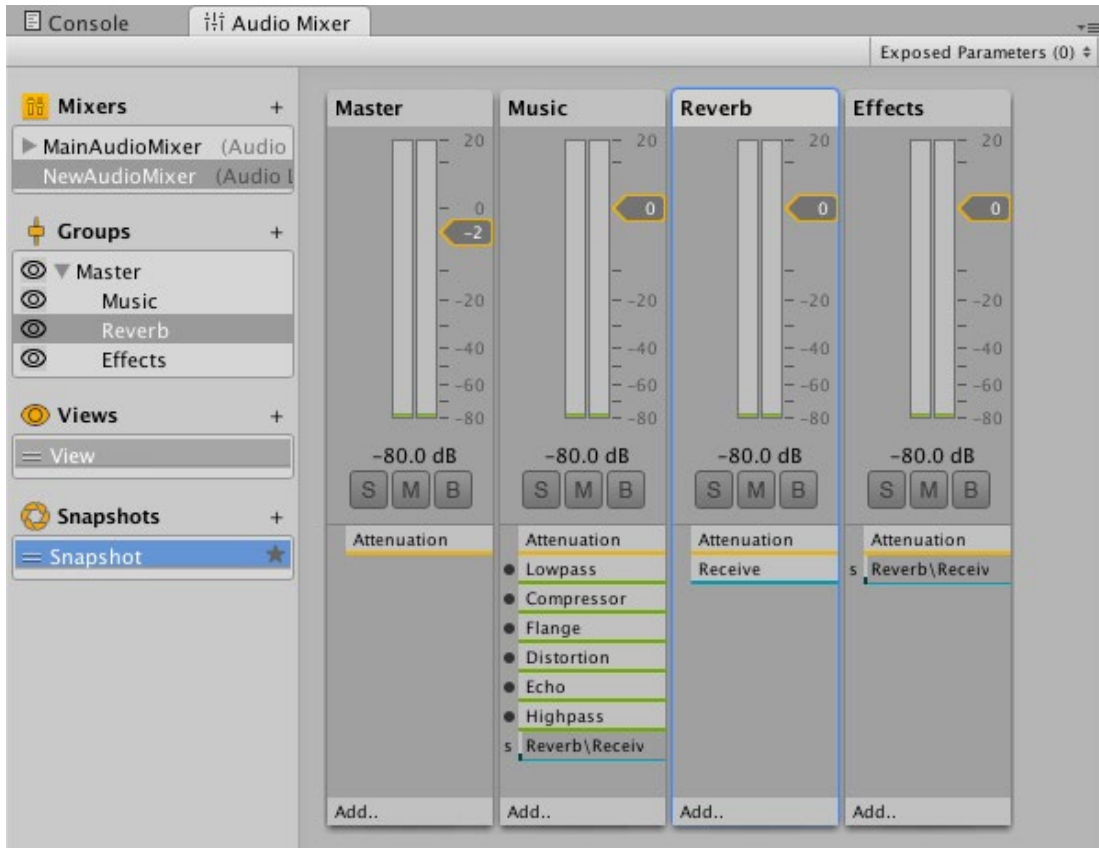


Figure 4-36. Unity's mixer window. Image sourced from Unity's developer documentation: <https://docs.unity3d.com/Manual/AudioMixer.html>

Ambisonic Audio

Unity currently can be configured to play back first- and second-order AmbiX Ambisonic files; however, no Ambisonic decoder is included with Unity's standard installation files. To use a third-party decoder like those offered by Google or Oculus, it must be installed and selected in the project's audio settings.

Licensing

Unity offers three tiers of licensing, all of which are royalty free and come with the full version of the game engine.

Table 4-13. List of Unity's licensing options. Information gathered from Unity's website: www.unity3d.com

Personal	A free and fully functional version of Unity is available for educational purposes or projects with less than \$100,000 in funding.
Plus	Unity's Plus plan is \$25 per month and grants users or projects that generate less than \$200,000 per year access to online training, support, storage, and assets.
Pro	Unity's Pro plan costs \$125 per month and is for projects that generate more than \$200,000 a year, or those that want a limited-edition t-shirt and full access to Unity's online training, storage, cloud synchronization, user metrics, and dedicated tech support.

4.6.3 Unreal Engine

Although Unreal Engine's Lead Audio Programmer Aaron McLeran has alluded to significant future enhancements to the developer's audio engine (McLeran, 2017), for the time being Unreal Engine's audio capabilities are similar to Unity's, minus any audio effects or macro-level mixing functionality. For those familiar with environments like Max/MSP or Pure Data, however, Unreal Engine's scripting environment, Blueprints (see Figure 4-37), may make customizing the behavior of audio more approachable than Unity's C# scripts (Gatheral, 2017).

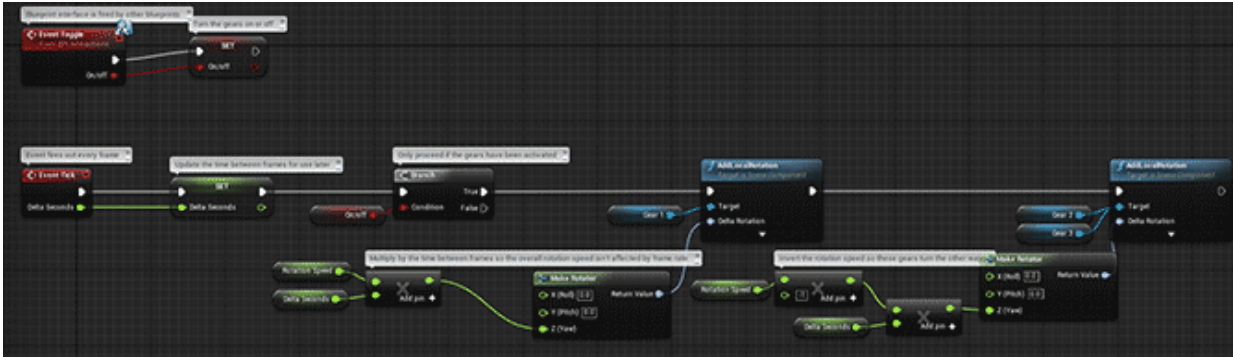


Figure 4-37. Image showing a typical Blueprint in Unreal Engine. Image sourced from the Unreal Engine developer documentation: <https://docs.unrealengine.com/en-us/Engine/Blueprints>

Supported Audio Files

Unreal Engine supports 16 bit .WAV files ranging from mono up to 7.1 surround. Like Unity, Unreal requires a third-party decoder when working with Ambisonic audio and only supports FOA (Unreal, 2018).

Sound Wave and Sound Cue Assets

An audio file can be directly imported into a scene or attached to an existing game object as a Sound Wave or a Sound Cue which can combine multiple Sound Wave assets. The playback behavior of a Sound Cue can be further defined in the Sound Cue Editor (see Figure 4-38) using the 21 available Sound Cue Nodes (see Table 4-13).

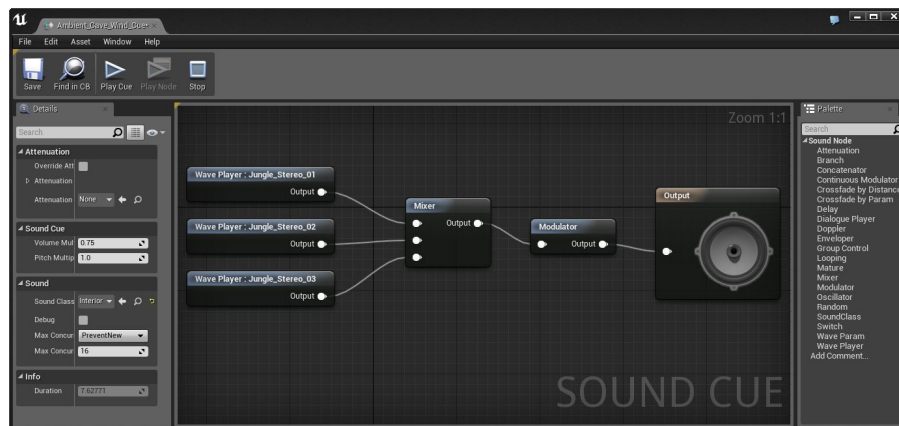


Figure 4-38. Image showing the Sound Cue Editor in Unreal Engine. Image sourced from the Unreal Engine developer documentation: <https://docs.unrealengine.com/en-us/Engine/Audio/SoundCues/Editor>

Table 4-14. List of all currently available Sound Cue Nodes in Unreal Engine's Sound Cue Editor. Information gathered from the Unreal's developer documentation: <https://docs.unrealengine.com/en-us/Engine/Audio/SoundCues/NodeReference>

Attenuation Node	The Attenuation node is used to control spatialization, attenuation, and radii properties.
Branch Node	The Branch node selects a child node based on the value of a Boolean parameter.
Concatenator Node	The Concatenator node is used to play sounds sequentially (e.g., creating a simple BGM playlist that plays through a series of specified tracks). While this node handles sequential playback, it does not blend from one sound asset to the next.
Continuous Modulator Node	The Continuous Modulator node allows gameplay parameters to control volume and pitch modulation in real time. A typical example is using the velocity of a vehicle to correspond to the pitch of an engine sound. This modulator needs to be hooked up in code for desired purposes and does not function on its own within the Sound Cue Editor.
Crossfade by Distance Node	The Crossfade by Distance node provides the ability to fade between two or more input sounds based on the distance from the Sound Cue's origin. By default, there are two inputs on this node, but you can add more inputs by right-clicking on the Crossfade by Distance node and clicking Add Input on the context menu. Sound nodes can be connected directly to the Crossfade by Distance node inputs, but you can also add nodes between them for independent control per layer.
Crossfade by Param Node	The Crossfade by Param node works similarly to the Crossfade by Distance node, except that the Crossfade by Param node allows you to manipulate the value of the parameter that controls the crossfade programmatically, with code or Blueprints.
Delay Node	The Delay node inserts a delay into the audio node chain to cause a pause before the input sound is passed to the output. The amount of the delay is a random value between the Delay Min and Delay Max values specified in the Details panel.
DialoguePlayer Node	The DialoguePlayer node sets up a Dialogue Wave sound asset to be played under specific conditions. Since Dialogue Waves are expected to be used purely for spoken dialogue, the DialoguePlayer node requires the user to specify both the speaker and the targets of the dialogue being played. This allows the Dialogue Wave to play the correct piece of audio according to the speaker's voice and any gender/plural/honorific handling that needs to occur for localization purposes.
Doppler Node	The Doppler node simulates the Doppler effect, in which sounds increase in pitch as you approach them at speed and decrease as you move away.
Envelope Node	The Envelope node allows you to create an envelope with optional looping ability to control the volume and pitch of a sound over time via distribution curves.
Group Control Node	The Group Control node determines which Sound Wave to play, depending on what sounds in the defined groups are already playing and their proximity to the listener.
Looping Node	The Looping node is used to set a Sound Wave to looping. You can loop multiple audio files independently when used in conjunction with the Mixer node.
Mature Node	The Mature node enables you to label a specific node or Sound Cue as containing content that would be labeled "Mature" by the ESRB rating system. You can use this node to filter out Mature-rated content in certain versions of your game.
Mixer Node	The Mixer node is used to trigger multiple Sound Wave nodes simultaneously. Sound nodes can be connected directly to the Mixer node inputs, but you can also add nodes between them for independent control per layer.
Modulator Node	The Modulator node is used to add random volume and pitch modulation. Each has a Max and Min value to determine the range of randomization. When the Sound Cue is triggered, it randomly selects a value within that range. You can use the Modulator node to adjust Volume + Pitch consistently by setting the Min and Max to the same values. It can also be used to adjust relative volumes of multiple Sound Wave nodes contained within a Sound Cue. If the Modulator node is used in conjunction with the

	Looping node, it will randomly select upon Sound Cue retrigger not per loop revolution.
Oscillator Node	The Oscillator node is used to add continuous pitch and volume oscillation over time. This is mostly useful on looping sounds to create a greater degree of ongoing movement.
Random Node	The Random node is used to randomly trigger a Sound Wave node from within a group of possible Sound Wave nodes. Weight controls the probability of a Sound Wave node being triggered relative to other Sound Waves nodes in the Sound Cue. The checkbox RandomWithoutReplacement will exhaust the entire list of possible sound nodes before repetition. Inputs are added for each audio file by right-clicking on the Random node and selecting Add Input. Sound Wave nodes can be connected directly to a Random node, but you can also add nodes between them for additional control.
SoundClass Node	The SoundClass node allows you to remap which Sound Class asset you are using with the Sound Wave input node.
Switch Node	The Switch node selects an input node based on the value of an integer parameter.
Wave Param Node	The Wave Param node takes a runtime parameter to determine whether to play a different Sound Wave than the one feeding into the input pin. If the specified parameter name is set to a valid Sound Wave asset, that Sound Wave asset is played instead of the Sound Wave input node.
Wave Player Node	The Wave Player node contains a Sound Wave asset from the Content Browser.

Timelines

Unreal's Timelines are an extremely useful tool that can be added to a Blueprint and can fulfill similar functions to DAW automations, ADSR envelopes, or multisegmented line generators (see Figure 4-39).

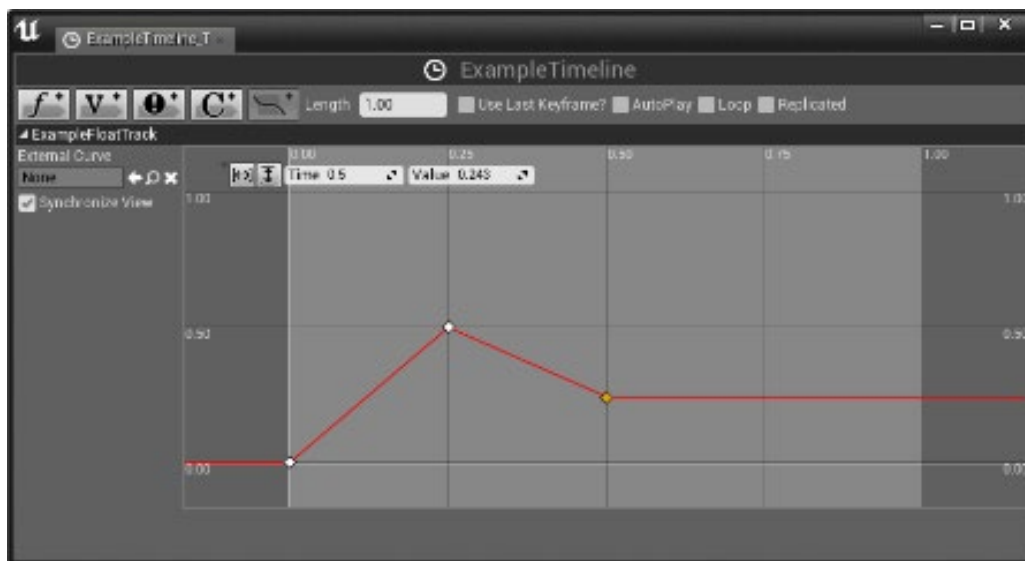


Figure 4-39. Image showing a simple example of a Timeline in the Unreal Game Engine. Image sourced from Unreal's developer documentation: <https://docs.unrealengine.com/en-US/engine/blueprints/userguide/timelines/keysandcurves>

Defining Acoustic Spaces

Dedicated game objects, called Audio Volume Actors, work in tandem with Unreal Engine's included reverb effect to define the acoustic characteristics of a three-dimensional area (Unreal, 2018).

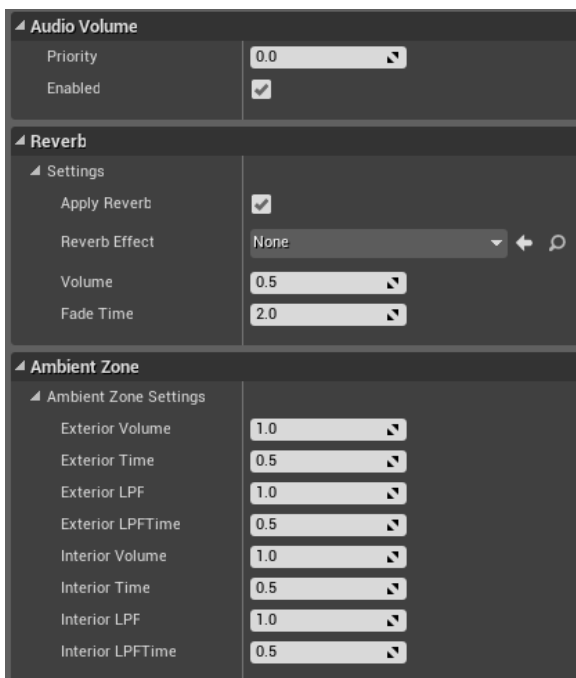


Figure 4-40. Image of the Audio Volume component in the Unreal Game Engine. Image sourced from Unreal's developer documentation: <https://docs.unrealengine.com/en-us/Engine/Actors/Volumes/AudioVolumeDetails>

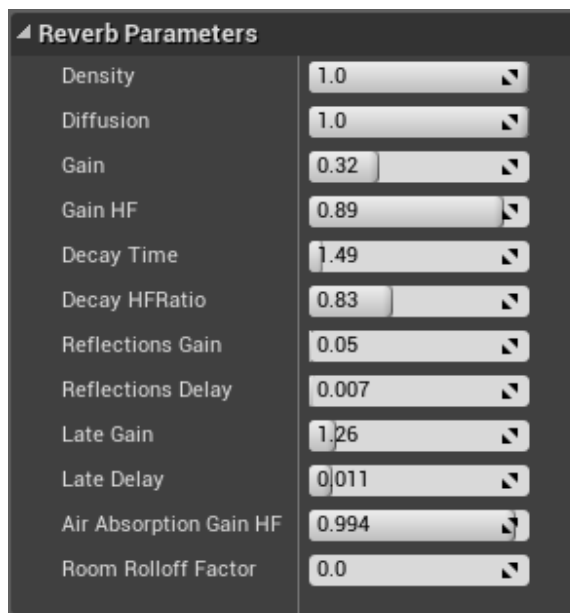


Figure 4-41. Image of the Reverb Effect in the Unreal Game Engine. Image sourced from Unreal's developer documentation: <https://docs.unrealengine.com/en-us/Engine/Audio/Overview>

Licensing

Unreal Engine offers a simplified licensing scheme of 5% of all revenue streams after the first \$3,000. Larger projects that want to limit potential royalty costs can negotiate custom terms directly with Unreal.

4.7 Audio Middleware

Audio middleware is specially designed software that works in conjunction with game engines to either augment or replace the sound and music systems. The two most popular audio middleware programs are FMOD Studio (Firelight Technologies, 2017) and Wwise (Audiokenetic, 2017). Both programs follow a hierarchical approach to authoring interactive sound and music systems and offer support for a variety of spatial audio formats. These two programs will be discussed in greater detail than some of the other technologies referenced in this chapter, as they are the environments which composers will focus the bulk of their efforts on because these audio middleware are currently the most powerful platforms for composers to explore the possibilities offered by new digital formats.

4.7.1 Firelight Technologies: FMOD Studio

FMOD Studio appears simple at first but is a competent and sophisticated tool. For creators familiar with a DAW like Ableton Live, FMOD Studio's layout will feel immediately familiar. FMOD Studio projects organize assets (audio files) and actions into trigger regions within events that ultimately are compiled into banks and loaded into a game engine.

Events

An event can be as simple as a single audio file or as complex as a complete interactive music system that should begin playing when called by the game engine. Each event in FMOD contains one multitrack timeline (see Figure 4-42) and an optional number of parameter sheets. An event's game parameter sheet shares the same tracks as the event's timeline but can have a

different collection and arrangement of trigger regions (see Trigger Regions) and parameter automations. Notably, however, trigger regions on a game parameter sheet are triggered by the current value of the associated game parameter rather than the playhead's location in the event's timeline.

For more on working with FMOD events, see chapter 4 of the FMOD Studio User Manual 1.10 (Firelight Technologies, 2018).



Figure 4-42. Screen capture showing an event's timeline in FMOD Studio 1.10.

Logic Markers

Without any logic markers, an event's timeline begins to play when called by the game engine and will run until the end of the last trigger region is reached. Optionally, a variety of logic markers can be added to control the timeline's playback behavior and the response of its trigger regions (see Table 4-14).

Table 4-15. List of FMOD's logic markers. Information gathered from FMOD's user manual:
<https://www.fmod.com/resources/documentation-studio?version=2.0&page=authoring-events.html#timeline-logic>

Tempo Marker	Tempo markers set the tempo and subsequent quantization behavior for tempo-synchronized trigger regions and logic markers.
Transition Marker	Transition markers identify a single point along the timeline where a different section could be immediately transitioned to.
Transition Region	Similar to a transition marker, a transition region identifies and equally divides an area of time into a number of potential transition points.
Transition Timelines	Double-clicking on any transition marker and region exposes its transition timeline that will be played before the destination material. Transition timelines can be any duration but do not allow any logic markers.
Destination Marker	A destination marker identifies a single point along the timeline the could be transitioned to using either a transition marker or region.
Loop Region	A loop region identifies a portion of the timeline for looping.
Sustain Point	A sustain point sets a location along the timeline that may be paused at until a later event is called to resume playback.

For more on working with logic markers in FMOD Studio, see section 4.7 of the FMOD Studio User Manual 1.10 (Firelight Technologies, 2018).

Conditional Triggers

Two types of conditional triggers, probability and parameter conditions, determine the active state of a specific transition marker, transition region, or loop region.

Probability, the most basic conditional trigger, sets the likelihood that the action of the logic marker will occur. Parameter conditions set the value range that either a single or collection of game parameters must fall within for a logic marker to become active. A typical example of this would be transitioning from the fight music loop to the victory music only when the player wins.

For more on conditional triggers, see section 4.7.1 of the FMOD Studio User Manual 1.10 (Firelight Technologies, 2018).

Trigger Regions

Events consist of trigger regions, which contain either an instrument or snapshot and are placed on either the event's timeline or game parameter sheet. Just like logic markers, trigger regions can also have conditional triggers applied to them.

For more on trigger regions, see section 4.3 of the FMOD Studio User Manual 1.10 (Firelight Technologies, 2018).

Instruments

Instruments are the sound sources found within the trigger region(s) of every FMOD event and can be either synchronous or asynchronous with the event's timeline. FMOD provides for a variety of basic instrument types to speed the authoring of sophisticated sound and music systems (see Table 4-15). For more on conditional instruments, see section 4.3 of the FMOD Studio User Manual 1.10 (Firelight Technologies, 2018).

Table 4-16. Complete list of FMOD's included instrument types. Information gathered from FMOD's user manual: <https://www.fmod.com/resources/documentation-studio?version=2.0&page=authoring-events.html#instruments>

Single Instrument	The simplest FMOD instrument, which plays only a single audio file.
Programmer Instrument	The programmer instrument does not make a sound. Instead, it produces a callback in the game engine, which can be useful in a variety of situations.
Event Instrument	There are two types of event instruments: nested and referenced. A referenced event instrument is the same as a regular event but just called by a trigger region within another FMOD event. The referenced event preserves its mixer and output settings and can still be triggered as a separate event or by other events. A nested instrument, alternatively, can only be called by the event it is nested within and does not preserve its own mixer. Both can be useful in a situation where more complex event interactions are needed.
Multi-Instrument	As its name implies, a multi-instrument can contain an assortment of single, programmer, or event instruments that can be played in order or randomly.
Scatterer Instrument	Scatterer instruments are ideal for ambiances or soundscapes because their 2D or 3D sound is produced at a random location relative to the game object the FMOD event is associated with each time they are triggered.
Plug-in Instrument	Plug-in instruments allow the use of custom sound generators built by third parties using FMOD's plug-in API (FMOD, 2018), such as AudioGaming's AudioWeather and AudioMotors plug-ins.
Snapshot Instrument	Snapshot instruments, like programmer instruments, do not make any sound but are used to recall specific mixer and event setting. Snapshot instruments are ideal

	for creating unique acoustic spaces such as reverb zones or echo chambers that are triggered by the player entering or exiting a specific location in the game.
--	---

Event Automation and Modulation

As an event plays, almost every parameter can be controlled with automations or by several modulation sources. A parameter can be automated across the timeline or game parameter sheets. In addition to automation, one of each modulator type can be applied to any given parameter (see Table 4-16). For more on modulators, see section 4.9.2 of the FMOD Studio User Manual 1.10 (Firelight Technologies, 2018).

Table 4-17. Complete list of FMOD's included modulators. Information gathered from FMOD's user manual: <https://www.fmod.com/resources/documentation-studio?version=2.0&page=authoring-events.html#automation-and-modulation>

Random Modulator	Applies a random value to the parameter each time the event or instrument is triggered.
AHDSR Modulator	Applies a basic attack, hold, decay, sustain, and release envelope to the parameter each time the event or instrument is triggered.
Sidechain Modulator	Modulates the value of a parameter based upon the audio signal received by the sidechain effects inserted at other location within the same event or mixer.
Autopitch Modulator	The autopitch is capable of modulating the pitch parameter of an instrument that is not on an event's timeline based upon the current value of the parameter sheet's associated value.

Built-in Parameters

Sound designers can define their custom game parameters for controlling an event's associated parameter sheet, but by default, FMOD supports several useful game parameters that do not require game developers to write any custom code (see Table 4-17). For more on FMOD's built-in parameters, see chapter 17 of the FMOD Studio User Manual 1.10 (Firelight Technologies, 2018).

Table 4-18. List of FMOD's built-in parameters. Information gathered from FMOD's user manual:
<https://www.fmod.com/resources/documentation-studio?version=2.0&page=authoring-events.html#parameters>

Distance	Reports the distance between the FMOD event emitter and the player. The distance parameter can be useful for applying more reverb to an event as the player gets farther away.
Direction	Reports the angle between the player's horizontal orientation and the location of the FMOD event emitter. In simpler terms, direction says if the player is looking at the event emitter.
Elevation	Similar to direction, the elevation parameter reports the vertical angle between the direction the player is facing and the location of the FMOD event emitter.
Event Cone Angle	In contrast to direction, the event cone angle parameter reports the horizontal relationship between the orientation of the FMOD event emitter and the player from the perspective of the event emitter.
Event Orientation	Also, from the perspective of the FMOD event emitter, event orientation reports the vertical relationship between the event emitter and the player.

Banks

Completed FMOD events and all of the assets that they require are packed into banks. The game engine must load a bank before any of its contained events can be called. Organizing events into banks allows the game engine to more efficiently manage system resources by loading only the audio assets required at that time.

For more on banks, see chapter 7 of the FMOD Studio User Manual 1.10 (Firelight Technologies, 2018).

Audio Effects

FMOD Studio comes bundled with a full suite of sound effects that can be applied, adjusted, automated, and modulated during real-time gameplay rather than prerendered like audio created within a DAW (see Table 4-18). Also, developers can create custom effects for use in FMOD with the FMOD Plug-in API (FMOD, 2018). Effects are added at any stage or order within FMOD's signal flow.

Table 4-19. A complete list of the audio effects included in FMOD.

3-EQ	A simple 3-band equalizer with an adjustable slope and frequency crossovers.
Channel Mix	The channel mix automatically detects the number of received audio channels and allows the independent volume control of each channel and the option to mix down to a specific speaker configuration.
Chorus	A simple chorus effect that includes a low-frequency oscillator (LFO) with independent rate and depth controls.
Compressor	A simple stereo compressor that features channel link and sidechain options.
Convolution Reverb	A stereo convolution reverb effect that supports importing any .WAV file as an impulse response.
Delay	A simple feedback delay effect.
Distortion	A general distortion effect.
Flanger	A simple flanger effect that includes an LFO with independent rate and depth controls.
Gain	A useful utility for adjusting a signal's volume.
Limiter	A traditional limiter with an input gain control, channel link option, and adjustable release time.
Loudness Meter	A meter for monitoring the loudness of a signal at any point along its path.
Multi-band EQ	A five-band graphic equalizer where each band is optionally assigned to a variety of low-pass, high-pass, shelving, peak, notch, bandpass, or all-pass filter types.
Pitch Shifter	An FFT-based pitch shifter capable of transposing an audio signal up or down by an octave without altering its rate.
Reverb	An algorithmic reverb with adjustable setting for reverb time, input filtering, reflections, and diffusion.
Transceiver	Working in groups of two or more, the transceiver acts as a send and return track, but rather than sending a signal to a specific track, the signal is streamed to any transceiver tuned to receive on the same channel.
Tremolo	A tremolo effect with the option for creating unique stereo panning patterns.

Stereo and Spatial Audio

FMOD distinguishes between two-dimensional (2D, stereo, or head-locked) and three-dimensional (3D, or spatial) events. Both event types support spatial audio, but, by default, 2D events do not have a spatializer audio effect (see Figure 4-43) placed on the master track. Instead, 2D events default to the content's channel format (see the section below on FMOD's supported audio file formats for more information).



Figure 4-43. Screen capture showing FMOD Studio's spatializer audio effect.

FMOD's spatializer audio effect relies upon its built-in parameters (see the section above on built-in parameters for more information) to adjust the sound's volume and output distribution in relation to the listener. The spatializer transforms all of the audio that passes through it to 3D audio. For more on the spatializer audio effect, see section 27.85 of the FMOD Studio User Manual 1.10 (Firelight Technologies, 2018).

Mixing

Each FMOD event features all of the elements found in a traditional DAW mixer—audio tracks, auxiliary (sends/returns) tracks, group tracks for creating sub-mixes, and a master track—allowing for total control over the event's signal flow. The output of each event's master track passes along to FMOD's mixer, which is the final stage of signal control before reaching the listener (see Figure 4-44). In the mixer, creators can route events to specific group tracks, set up auxiliary tracks for adding effects (e.g., reverb), and apply mastering effects to the master bus. FMOD's mixer also supports VCA faders and saving mixer snapshots that are recalled from the game engine. For more on mixing in FMOD, see chapter 8 of the of the FMOD Studio User Manual 1.10 (Firelight Technologies, 2018).

Supported Audio File Formats

The FMOD Studio API documentation explicitly states that it supports WAV, MP3, and OGG Vorbis formats but also suggests that it supports other “obscure” formats like AIFF and FLAC files, as well as sequenced formats like MIDI, MOD, S3M, XM, and IT. Nowhere in the accompanying user manual (Firelight Technologies, 2018), however, does it affirm this support, nor does it discuss how files like MIDI can be used (Firelight Technologies, 2018). The user manual does declare support for mono through 7.1.4 (seven speakers surround the audience at ear level, one subwoofer, and four additional speakers above the listener) speaker configurations.



Figure 4-44. Screen capture of the FMOD Studio's mixer window.

Licensing

Firelight Technologies offers three tiers of licensing (see Table 4-19) for FMOD based upon use and project budget (Firelight Technologies, 2017).

Table 4-20. List of FMOD licensing options. Information gathered from FMOD's website: <http://www.fmod.com>

Indie	Free for use on one project per year if the overall budget is less than \$500,000. Each additional license will cost \$2,000.
Basic	Each license will cost \$5,000 for any release with a budget of more than \$500,000 and less than \$1.5 million.
Premium	For release built with a budget of more than \$1.5 million, licensing FMOD will cost \$15,000.

4.7.2 Audiokinetic: Wwise

The user interface of Wwise features few, if any, similar landmarks to the tools most composers may be accustomed to working with. Once users have found their bearing and taken advantage of the excellent educational and training materials available on Audiokinetic's website, however, the incredible power and flexibility that comes with breaking free of a DAW approach to authoring musical experiences starts to reveal itself.

Project Hierarchy

Wwise projects are broken into three separate hierarchies: the Actor-Mixer Hierarchy and the Interactive Music Hierarchy, both of which output their audio signals to the Master-Mixer Hierarchy (see Figure 4-45).

Actor-Mixer Hierarchy

The Actor-Mixer Hierarchy is where all sound objects are authored and organized into object containers.

Interactive Music Hierarchy

The Interactive Music Hierarchy is where all music segments are authored and organized into music containers.

Master-Mixer Hierarchy

The Master-Mixer Hierarchy is similar in function to a DAW's mixer and is where all sound objects and music segments are bussed and mixed.

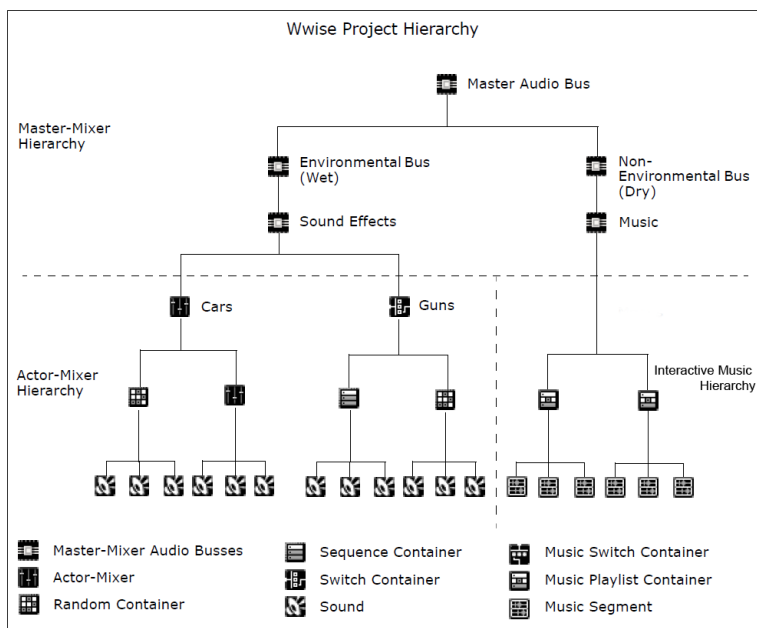


Figure 4-45. Graphic depicting the Wwise project hierarchy. Graphic sourced from the Wwise documentation: https://www.audiokinetic.com/library/edge/?source=WwiseFundamentalApproach&id=understanding_master_mixer_hierarchy

Organizing assets into these three hierarchies establishes multi-level parent-child relationships. Children objects can inherit settings from their parents, which makes adjusting parameters for multiple objects easy and efficient.

Sound Objects

A sound object is the base container for assets added to the Actor-Mixer Hierarchy. A sound object can contain two basic source types: audio source and source plug-ins.

Audio Source

An audio source is the abstracted layer between the sound object and the audio file associated with that audio source (see Figure 4-46). This added layer between the sound object and the audio file makes it easier for a sound object to reference the same audio file while using different setting (see Figure 4-47).

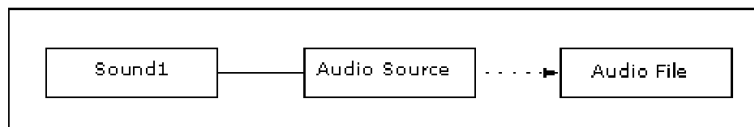


Figure 4-46. Graphic showing the sound object, audio source, and audio file structure in Wwise. Graphic sourced from the Wwise documentation:

https://www.audiokinetic.com/library/edge/?source=WwiseFundamentalApproach&id=audio_objects

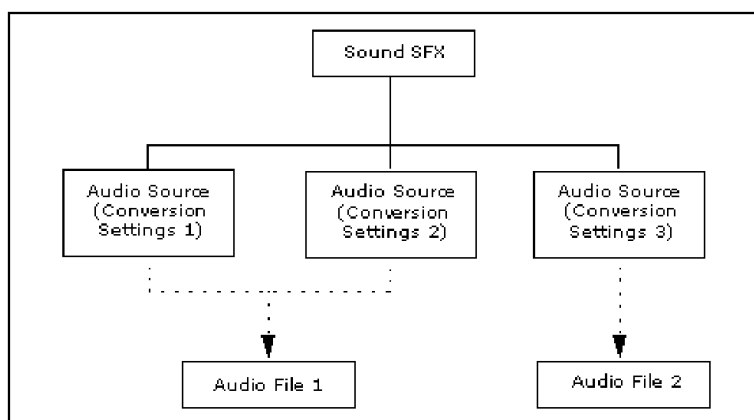


Figure 4-47. Graphic showing a single sound object that reference two audio files with several different audio source settings. Graphic sourced from the Wwise documentation:

https://www.audiokinetic.com/library/edge/?source=WwiseFundamentalApproach&id=audio_objects

Source Plug-ins

Unlike audio sources, which reference prerendered audio files, source plug-ins (see Table 4-20) are similar to the plug-in instruments found in most DAWs and are capable of synthesizing sound in real time.

Table 4-21. A list of all the source plug-ins included in Wwise. Note: some sound source plug-ins require additional licensing.

WWise Audio Input	Allows game audio, such as the user's microphone, to be input and processed in Wwise.
Wwise Motion Source	Generates low-frequency patterns for activating haptic feedback devices.
Wwise Silence	Generates digital silence. Useful for inserting duration of silence between other audio events.
Wwise Sine	A simple sine tone generator.
Wwise Synth One	A two-oscillator synthesizer with options to either sum, ring modulate, or frequency modulate oscillator one with the output of oscillator two. Also includes white noise generator.
Wwise Tone Generator	A bandlimited sine, saw, square, triangle, white noise, or pink noise tone generator.

Wwise External Source	Enables a variety of user-specific audio files to be loaded and processed by Wwise at runtime.
SoundSeed Air Wind	Generates wind sounds based upon real-time simulation of weather and other environmental conditions.
SoundSeed Air Woosh	Generates the “woosh” sound of an object moving through the air.
SoundSeed Grain	A full-featured granular synthesizer for real-time audio generation.

Object Containers

Sound objects can be grouped into a variety of containers for controlling the properties and behaviors of multiple sound objects (see Figure 4-48). Wwise has several object container types that can be arranged in almost any way (see Table 4-21).

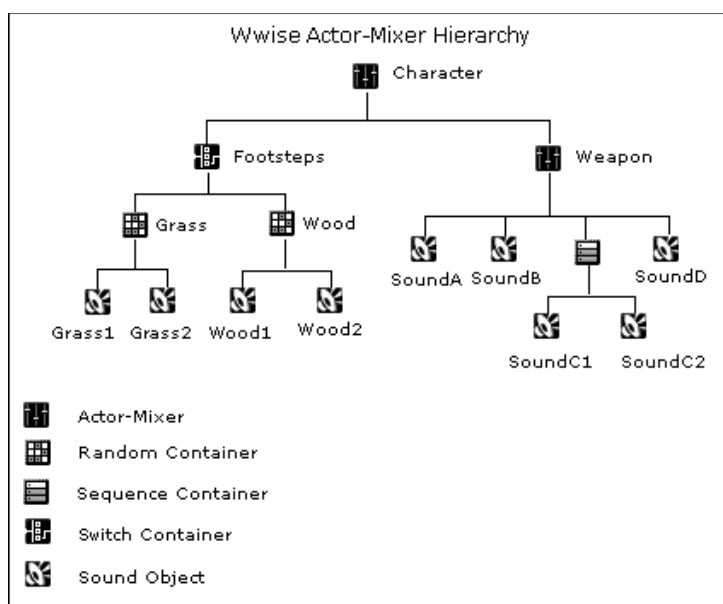









Figure 4-48. Graphic providing an example of an Actor-Mixer Hierarchy in Wwise. Graphic sourced from: https://www.audiokinetic.com/library/edge/?source=Help&id=using_containers_and_actor_mixers_within_actor_mixer_hierarchy

y

Table 4-22. Actor-Mixer Hierarchy container types. Information gathered from the Wwise documentation: https://www.audiokinetic.com/library/edge/?source=Help&id=building_actor_mixer_hierarchy_building_actor_mixer_hierarchy

y

Icon	Type	Description
	Work Units	High-level elements used to divide up a project so that different people can work on the project concurrently. Work Units contain the assets within your project hierarchy, as well as other Wwise elements, such as States, Effects, and SoundBanks.

	Physical Folders	High-level elements into which you can group other physical folders or Work Units in a project. Physical folders cannot be child objects for containers or sounds.
	Virtual Folders	High-level elements into which you can place other objects. Virtual folders cannot be child objects for containers or sounds.
	Actor-Mixer	High level objects into which you can group other objects. Properties that are applied to an Actor-Mixer affect the properties of the objects grouped under it. You can also group objects within an Actor-Mixer using Virtual Folders.
	Random	A group of one or more objects and/or containers that are played back in a random order.
	Sequence	A group of one or more objects and/or containers that are played back according to a specific order or playlist.
	Switch	A group of objects and/or containers that are organized into a series of Switches that correspond to the different alternatives that exist for a particular element in the game. The particular objects that are played depend on which Switch is active in the game.
	Blend	A group of one or more objects and/or containers that are played back simultaneously. The different objects within this container can be grouped into blend tracks where object properties are mapped to Game Parameter values using RTPCs. Crossfades can also be applied between objects within a blend track based on the value of a Game Parameter.

Events

A Wwise event is called by the game engine to play, stop, or otherwise alter a sound object or container. An event can perform multiple actions in sequence or simultaneously (see Figure 4-49 and Figure 4-50). Without at least one event to play a sound object, a Wwise project would not make a sound. Every event is associated with an object in the game engine. Wwise monitors these game objects for information about their location, movement, and environment, using these data to alter the sound objects and music segments controlled by the event.

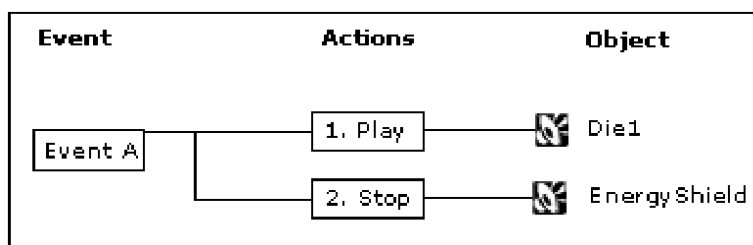


Figure 4-49. An example of a simple Wwise event that plays one sound object while stopping another. Graphic sourced from the Wwise documentation:

https://www.audiokinetic.com/library/edge/?source=WwiseFundamentalApproach&id=understanding_events_understanding_events

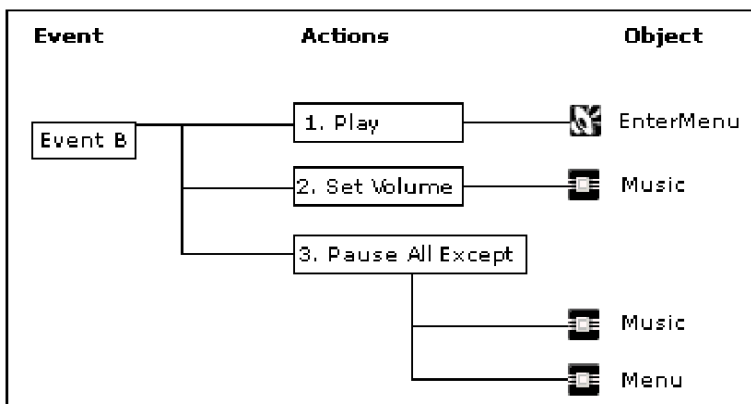


Figure 4-50. Example of a Wwise event that plays one sound object, sets the volume of a bus track in the Master-Mixer Hierarchy, and pauses all events except for those that pass through the music and menu bus tracks of the Master-Mixer Hierarchy. Graphic sourced from the Wwise documentation:

https://www.audiokinetic.com/library/edge/?source=WwiseFundamentalApproach&id=understanding_events_understanding_events

Game Syncs

Game Syncs are streams of data or event calls sent from the game engine to Wwise that can be used to control the music and sound systems designed within Wwise.

Table 4-23. List of the Wwise Game Syncs. Information gathered from the Wwise user documentation: https://www.audiokinetic.com/library/edge/?source=WwiseFundamentalApproach&id=what_are_game_syncs

States	Similar to mixer snapshots, States are used to recall global shifts in sound qualities or processing (see Figure 4-51).
Switches	A Wwise sound object may have several alternative versions and switches are used to determine which version will be played (see Figure 4-52).
Real-Time Parameter Controls (RTPCs)	Game engine and Wwise parameters can be linked in real time using RTPCs (see Figure 4-53).
Triggers	Triggers are calls from the game that launch short musical segments to accent key moments or actions (see Figure 4-54).

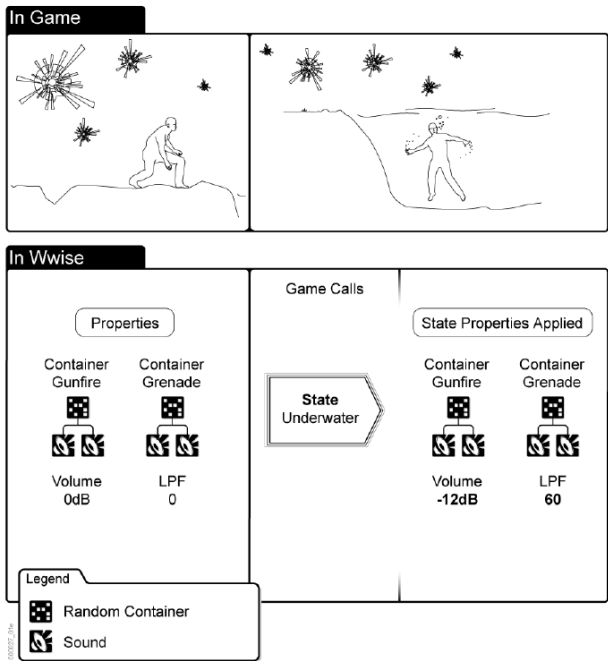


Figure 4-51. An example of how the State Game Sync could be used to change the sound of gunfire and grenades exploding when the player jumps or goes under water. Graphic sourced from the Wwise documentation: https://www.audiokinetic.com/library/edge/?source=WwiseFundamentalApproach&id=understanding_states

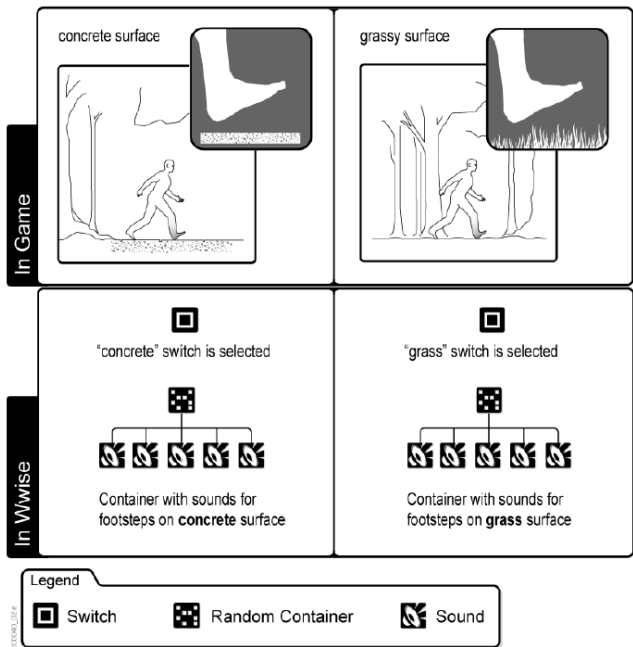


Figure 4-52. An example of how a Switch Game Sync can be used to switch between two groups of footstep sounds in random containers depending on the surface type the player is walking upon. Graphic sourced from the Wwise documentation: https://www.audiokinetic.com/library/edge/?source=WwiseFundamentalApproach&id=understanding_switches

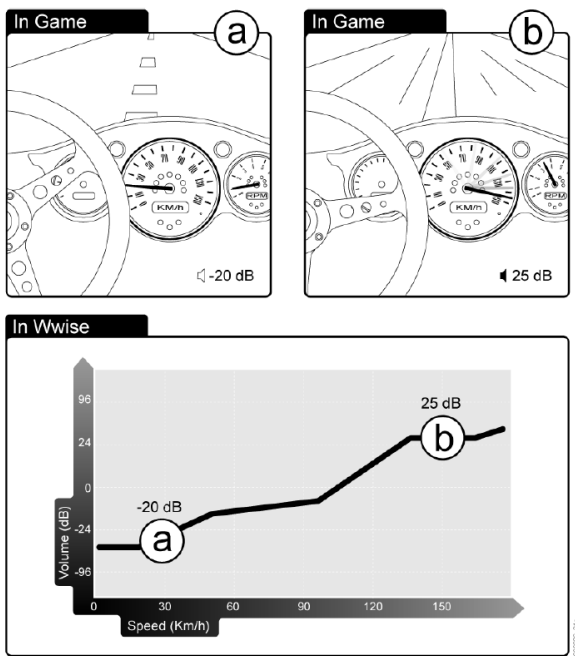


Figure 4-53. An example of linking a car's speed to the volume of a Wwise event. Graphic sourced from the Wwise documentation:
https://www.audiokinetic.com/library/edge/?source=WwiseFundamentalApproach&id=understanding_rtpcs

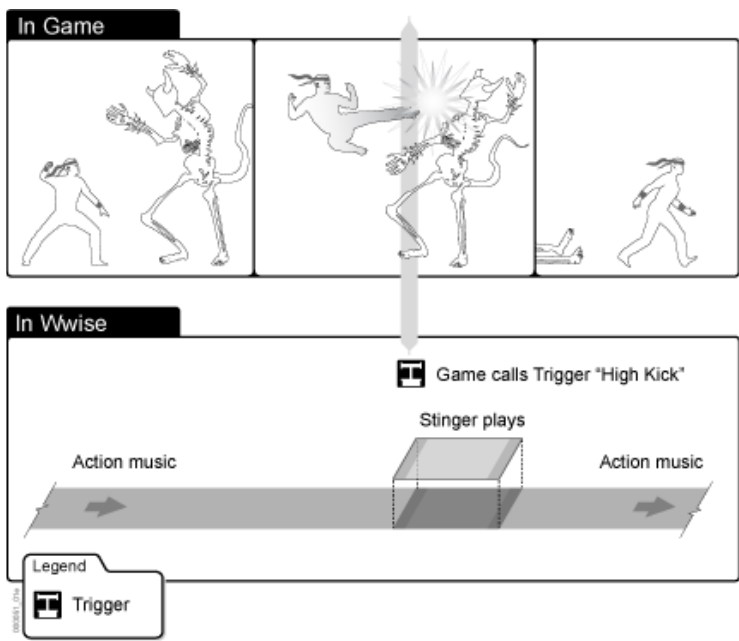


Figure 4-54. An example of a trigger used to start a short musical phrase when the player performs a special action. Graphic sourced from the Wwise documentation:
https://www.audiokinetic.com/library/edge/?source=Help&id=working_with_triggers_overview

Interactive Music

Music systems built in Wwise's Interactive-Music Hierarchy are driven at every level by internal structures and Game Sync controls.

Music Segment

Like the sound object at the base of the Actor-Mixer Hierarchy, the Music Segment is the core object of the Interactive-Music Hierarchy and is created by organizing MIDI or audio files onto its contained tracks and sub-tracks. An individual tempo, meter, and duration is given to each Music Segment, and entrance and exit markers define the points that each segment should align, including any pre- and post-roll material that has been identified by leaving material outside of the entrance and exit makers.

To provide greater variety and control, each track within a Music Segment can be divided into several types of sub-tracks: Random Step, Sequence step, or Switch. If a track has been subdivided into several Random Steps, a random sub track will play each time the music segment is played. The Sequence Step, on the other hand, will play sequentially through each of the sub-tracks each time the music segment is played. Alternatively, sub-tracks can be associated with a Switch Game Sync to determine which sub-track will play.

Completed Music Segments are then sequenced one after the other during runtime to generate complete musical performances (see Figure 4-55).

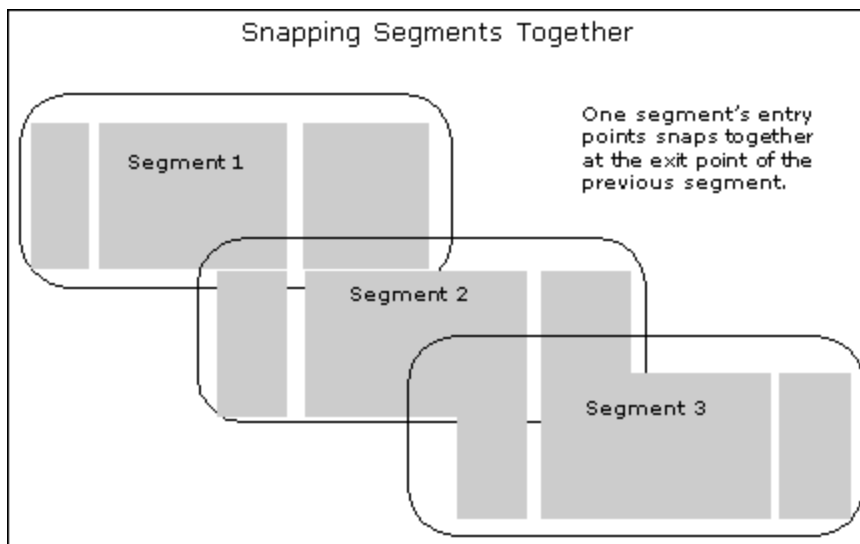


Figure 4-55. A graphic illustrating how Music Segments are sequenced and aligned at their markers. Graphic sourced from the Wwise documentation:

https://www.audiokinetic.com/library/edge/?source=Help&id=what_is_music_segment

Music Containers

Wwise offers two types of containers to organize Music Segments into the Interactive Music Hierarchy and control how and when music segments are played. Like the containers of the Actor-Mixer Hierarchy, Music Containers can be nested to create simple or complex structures.

Music Switch

The Music Switch Container allows branches within the Interactive Music Hierarchy to be selected based upon associated Game Syncs.

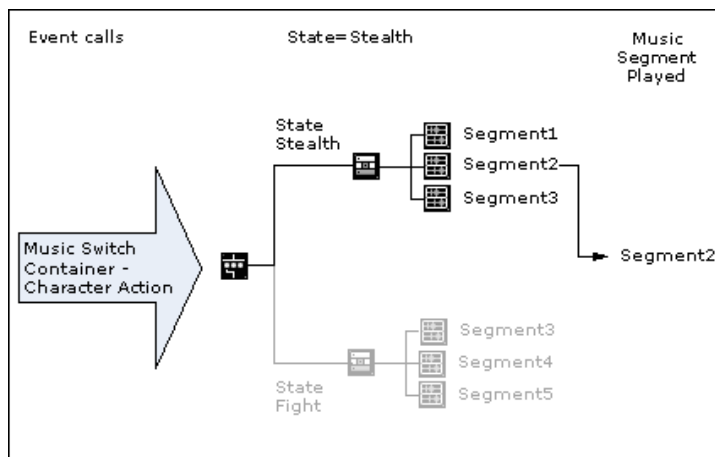


Figure 4-56. An example of an Interactive Music Hierarchy constructed using a Music Switch container.

Graphic sourced from the Wwise documentation:

https://www.audiokinetic.com/library/edge/?source=Help&id=defining_contents_and_behaviors_of_music_switch_containers

Music Playlist

The Music Playlist container allows Music Segments to be played a definable amount of times either randomly or in sequence (see Figure 4-57).

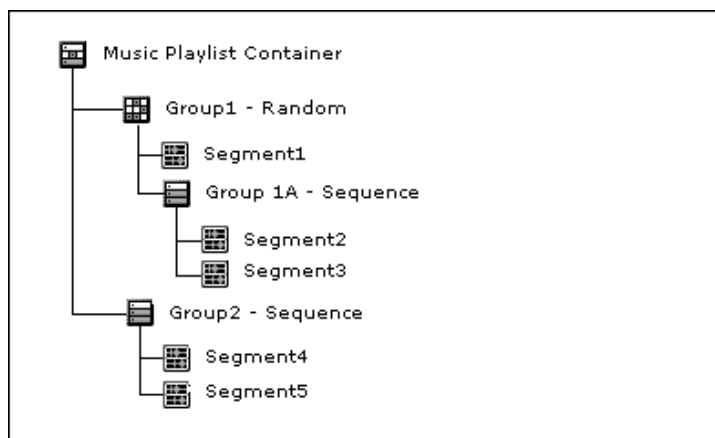


Figure 4-57. An example of an Interactive Music Hierarchy using a Music Playlist container. Graphic sourced from the Wwise documentation:

https://www.audiokinetic.com/library/edge/?source=Help&id=defining_contents_and_behaviors_of_music_switch_containers

Transitions

In addition to the pre- and post-rolls of individual Music Segments, multiple Music Segments or entire branch of Music Containers can be used to smooth the transition from

one musical construction to another (see Figure 4-58). Just like everything else in the Interactive Music Hierarchy, the transition that will play can also be randomized, determined by Game Syncs, or fixed.

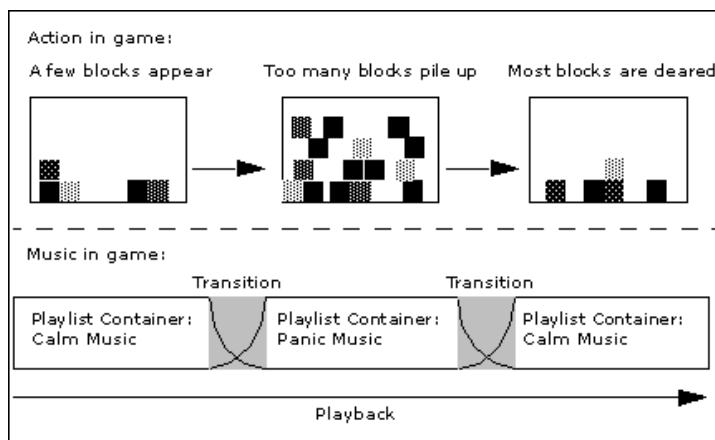


Figure 4-58. An example of how transitions can be used to move from one Music Playlist to another. Graphic sourced from the Wwise documentation:

https://www.audiokinetic.com/library/edge/?source=Help&id=working_with_transitions_overview

MIDI Sequencing

Sound objects in the Actor-Mixer Hierarchy can be customized to act similar to the sampler instruments typically found in DAW and triggered by MIDI files imported to tracks within Musical Segments.

For more on working with and creating MIDI instruments, see Chapter 26 of the Wwise documentation:

https://www.audiokinetic.com/library/edge/?source=Help&id=working_with_midi

Positioning, Spatialization, and Animation

Wwise offers several options for spatializing and animating a mono, stereo, multichannel, or even Ambisonic audio file.

Speaker Panning

All sound and music objects in Wwise default to speaker panning (head-locked audio) and can either be assigned directly to a speaker based upon their file format or rebalanced across any typical surround format using the surround balance control panel (see Figure 4-59).

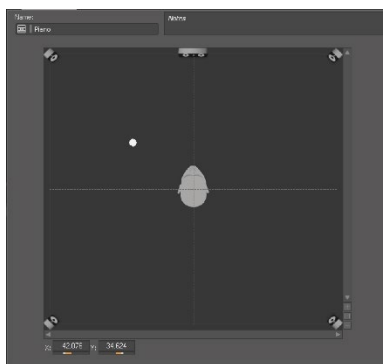


Figure 4-59. Screen capture showing the Balance Control panel in Wwise.

Listener-Relative Routing

Checking a small box on a sound object's Position tab (see Figure 4-60) switches from speaker panning, to listener-relative routing, where the spatial processing applied to the sound is responsive to the position or orientation of the listener and the sound's associated game object. In this mode, additional RTPC can be assigned to control various parameters of the sound. For example, the listener's distance from the associated game object commonly controls the volume of the sound so that as the listener gets closer the sound gets louder.

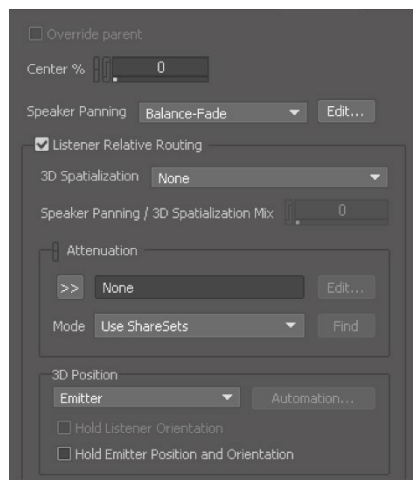


Figure 4-60. Screen capture of the Position tab in Wwise.

Spatial Animation

A listener-relative sound can also be spatially animated using the Position Editor panel (see Figure 4-61). The Position Editor allows for several paths for a sound to be charted across both time and space, which can then be played back in a variety of random and sequential modes.

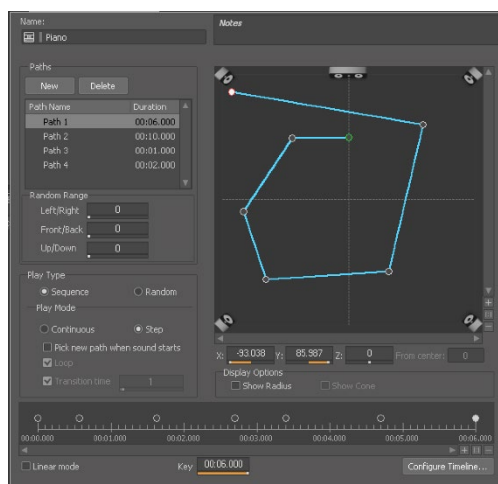


Figure 4-61. Screen capture showing the Position Editor in Wwise.

Effect Plug-ins

Wwise comes bundled with a full collection of audio effects (see Table 4-23) and supports many third-party plug-ins. Audio effects are inserted at a variety of points along the Wwise signal path and can also be controlled at runtime by RTPCs.

Table 4-24. A list of all Wwise's effect plug-ins. Information gathered from the Wwise documentation:
<https://www.audiokinetic.com/library/edge/?source=Help&id=effects>

McDSP FutzBox Lo-Fi Distortion	A distortion and noise generator for creating low-fidelity versions of audio signals.
McDSP ML1 Mastering Limiter	A mastering limiter for reducing signal peaks and increasing overall levels.
Wwise Compressor	A basic compressor for controlling the dynamic range of a signal.
Wwise Convolution Reverb	A convolution reverb plug-in compatible with a variety of spatial impulse response files, including TOA.
Wwise Delay	A simple feedback delay effect with a range of 0.001 to 1000 milliseconds.
Wwise Expander	Increases the dynamic range in the opposite fashion of a compressor by reducing any signal that falls below an adjustable threshold.
Wwise Flanger	A basic flanging effect created by mixing the original signal with a second copy delayed by a varying amount of time.
Wwise Gain	A simple volume control effect.
Wwise Guitar Distortion	A waveshaper-based distorting effect with a variety of distortion algorithms.
Wwise Harmonizer	An FFT-based pitch shifter for harmonizing and input signal.
Wwise Matrix Reverb	An algorithmic reverb designed to be easily scalable to meet varying CPU restrictions.
Wwise Meter	A level meter used for visualizing or side-chaining through RTPCs.
Wwise Parametric EQ	A three-band parametric equalizer. Each band has adjustable frequency, gain, and resonance, and can be assigned to any of the following filter types: low pass, high pass, band pass, notch, low shelf, high shelf, and peaking.
Wwise Peak Limiter	A peak limiter with look-ahead used for controlling the dynamic range of a signal by reducing any moment that exceeds an adjustable ceiling.
Wwise Pitch Shifter	A pitch shifter for transposing a signal without altering its duration that features a signal filter for processing the transposed signal.
Wwise Recorder	Used for capturing and saving in real time an audio recording of a signal at any point within Wwise and gameplay. Supports recording a variety of audio formats, including TOA.
Wwise Reflect	A multi-tap delay with filters for the purposes of simulating early reflections.
Wwise RoomVerb	A robust and processor-intensive algorithmic reverb for simulating a variety of room acoustics.
Wwise SoundSeed® Impact	An impact modeling synthesizer.
Wwise Stereo Delay	A stereo delay with feedback and cross-feed functionality.
Wwise Time Stretch	Enables stretching or compressing a sound's duration without altering its pitch.
Wwise Tremolo	A variable waveshape stereo amplitude modulation plug-in for creating a range of pulsing rhythmic patterns.

Acoustic Simulations

Under ongoing research, development, and implementation, Wwise Spatial Audio (Audiokinetic, 2019) currently uses the Wwise Reflect and Convolution plug-ins to render real-time acoustic simulation of virtual spaces. The simulations work by calculating the geometrical relationships and material properties of the emitter, listener, and surrounding world to dynamically compute a sound's early reflections, spectral transformations, and diffusion through a process called raycasting (see Figure 4-62).

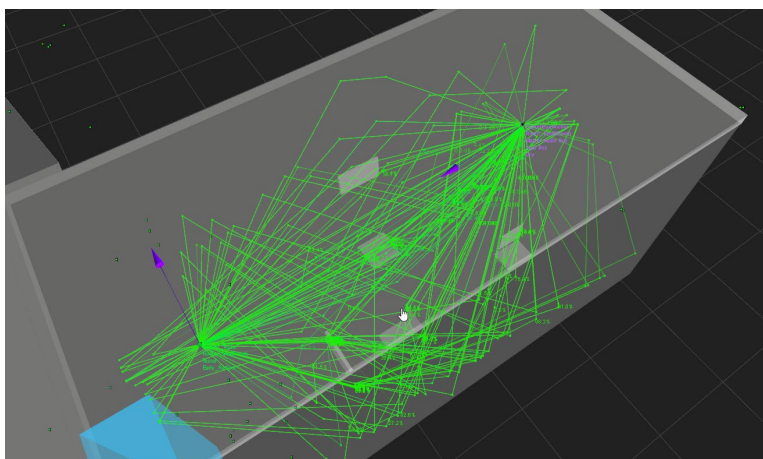


Figure 4-62. Image showing the use of raycasting to calculate the early reflections of a single sound source heard by a listener.
Image source from: <https://blog.audiokinetic.com/acoustic-simulation-with-wwise/>

To reduce the processing demands that a complete and real-time acoustic simulation would place on a playback device, Wwise Spatial Audio uses a room and portal system. This limits calculations to the area (space) a listener is currently within and the acoustic openings (portals) that space might have that would expose it to a selection of external sound emitters (see Figure 4-63).

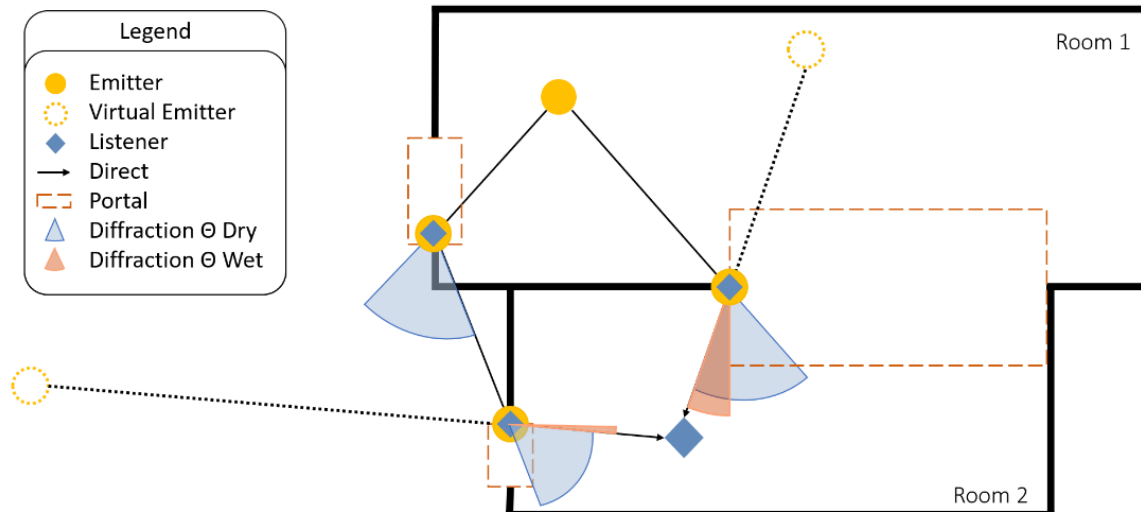


Figure 4-63. A graphic showing the room and portal system used by Wwise Spatial Audio. Graphic sourced from: <https://www.audiokinetic.com/products/wwise-spatial-audio/>

As seen in Figure 4-63, the Wwise Spatial Audio system can also account for any occlusion or obstruction of a sound emitter.

Several of the other developers currently working on more realistic and computationally efficient ways to render real-time acoustic simulations will be discussed in the section below on audio software development kits (see section 4.8).

Supported File Formats

Wwise supports 16 or 24-bit .WAV files with sample rates of up to 96kHz. Imported files can be mono, stereo, surround formats of up to 13.1, and up to TOA. Wwise also supports importing MIDI files for use in Music Segments.

Licensing

Like FMOD, licensing for Wwise is available at several levels dependent upon user needs and project budgets (see Table 4-24). Additional licensing information for the various source and effects plug-ins can be found at: <https://www.audiokinetic.com/pricing/for-games/>

Table 4-25. List of Wwise licensing options: Information gathered from Audiokinetic's website: <https://www.audiokinetic.com>

Starter, Academic, or Non-Commercial Projects	Wwise is completely free if the project will also be free or used for educational purposes. Limits the user to 500 media assets and offers only basic tech support.
Pro	Cost \$750 per project, unlocks the use of unlimited media assets and allows the commercial use of Wwise on projects with budgets between \$0 and \$1.5 million. Includes several tech support options for additional charges.
Premium	At \$6,000 for the first product and \$3,000 for each additional product, the Premium plan comes with full tech support and is for projects with a budget that exceed \$1.5 million.
Enterprise	For companies releasing multiple products a year and with massive budgets, the Enterprise plan offers negotiable licensing fees and support plans to better suit the client's needs.

4.8 Audio Software Development Kits

Several audio software development kits (SKDs) have emerged to further augment or replace the spatialization functionality of game engines or audio middleware. The list of SDKs presented here is not exhaustive but instead includes the most widely used offerings that represent the current capabilities of audio spatialization-focused SDKs.

With the exception of Wwise's acoustic simulation capabilities (see the Acoustic Simulations subsection of 4.7.2), most game engines and audio middleware rely on standard volume panning and distance-based attenuation to create a rudimentary sensation of spatialized sound sources. These Spatial Audio SDKs, on the other hand, work to emulate the real-world behavior of sounds within a space.

4.8.1 Steam Audio SDK

Initially released on February 23, 2017, the Steam Audio SDK 2.0 Beta offered a unified solution for 3D audio with real-time and baked sound propagation modeling. The latest version,

released on February 21, 2019, can now be integrated directly with Unity, Unreal Engine, FMOD, and a C API, with Wwise support coming soon. Steam Audio can calculate in real-time sound propagation, occlusion, and reflection for hundreds of sources and render the results for either HRTF-based binaural or HOA audio output (Valve, 2017).

4.8.2 Google Resonance SDK

Initially released on November 3, 2017, Google Resonance is functionally similar to Valve's Steam Audio SDK, offering customizable audio source spread and directivity, near-field effects, geometry-based reverb, occlusion modeling, and Ambisonic file rendering options. Unlike Steam Audio, however, Google Resonance is already capable of integrating with Wwise, in addition to FMOD, Unity, and Unreal Engine. Google also offers a version of the SDK for web audio and a simple VST plugin for decoding Ambisonic audio within a DAW (Google, 2019).

4.8.3 Oculus Audio SDK

Oculus' Audio SDK offers the most robust toolset for working with spatialized audio, which includes a VST/AAX DAW spatializer plug-in (see Figure 4-64); however, it is currently compatible only with Unity, FMOD, and Wwise (Oculus, 2016). In all three environments, the Oculus Audio SDK supports real-time reverb and occlusion simulation, volumetric source definition, and near-field effects. The SDK also features additional audio utilities for synchronizing an avatar's mouth animation with voice recordings, an audio effects processor, an

audio profiler for real-time statistics for tracking audio system performance, and a 3D loudness meter.

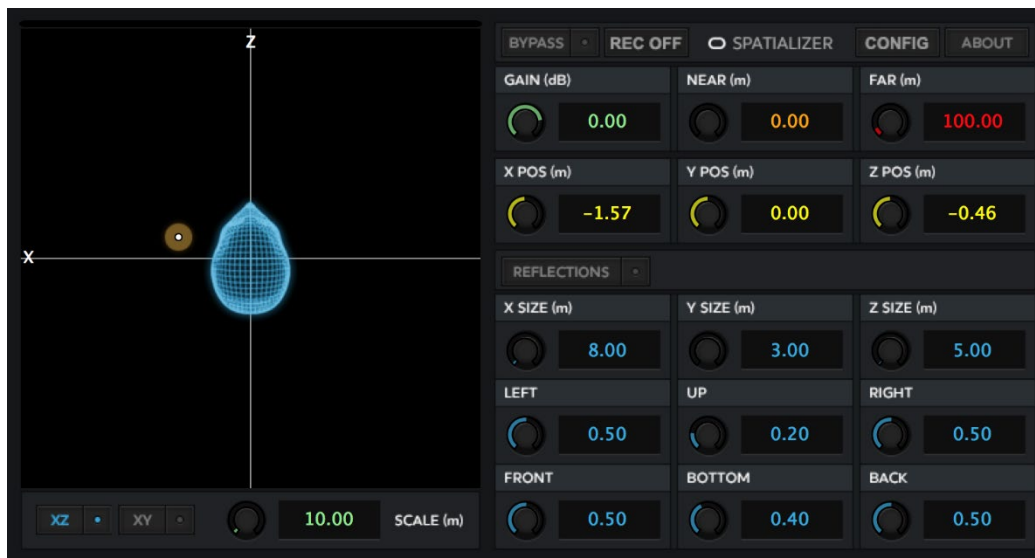


Figure 4-64. Screen capture of the Oculus Spatializer VST plug-in.

4.9 What is on the horizon?

At the end of the 2010s, two additional developments occurred that have yet to find large-scale adoption by AR/VR workflows and production techniques but that are still worth mentioning. The first is an evolution of old technology, and the second is a new recording technique that will help blur the boundaries between traditional and newer, more immersive media formats.

4.9.1 Multidimensional Polyphonic Expression and MIDI 2.0

In 2018 the MIDI Manufacturers Association (MMA) officially adopted the new MIDI Polyphonic Expression (MPE) enhancement (The MIDI Association, 2019). Later, at the National

Association of Music Merchants (NAMM) 2019 tradeshow, MMA announced its development and release plans for the new MIDI 2.0 standard (The MIDI Association, 2019). Both adoptions represent significant evolution of the MIDI protocol, which has remained fairly stagnant since its initial formation in the 1980s (The Holladay Brothers, 2018). MPE and MIDI 2.0 reorganize and expand MIDI's expressive capabilities, which will not only allows for new physical instruments and controllers, such as the Roli Seaboard Rise (Roli, 2017), but also present interesting possibilities for how the data-rich environments of AR and VR could rely upon MIDI to communicate musical expression in ways never before experienced.

4.9.2 Volumetric Capture

Research and development around volumetric capture is being conducted for both visuals and audio. Unlike 360 video or Ambisonic microphones, which capture from a single location, the volumetric capture attempts to record a complete scene or object in a way that could later be navigated through and experienced from any perspective. This approach requires surrounding the subject with meticulously positioned cameras and microphones to capture a performance, after which the separate feeds are stitched together in post-production to create a single 3D object (see Figure 4-65).



Figure 4-65. Photo of a volumetric capture stage. Photo courtesy of sound engineer/composer Joel Douek of ECCO VR:
<http://www.eccovr.com/>

Chapter 5: New Compositional Techniques

5.1 New Possibilities

5.1.1 Social Connections

As shown by Smule's *Ocarina*, mobile electronic devices can be used to create experiences that foster social connections across boundaries: geographic, cultural, economic, or, to a certain extent, even time (Ling, 2015). Yet the social benefits of game-like formats are not limited to forging new connections. One recent study, for example, also found that a significant percentage of teens who play online games feel more connected to their existing friends and gaming community, reinforcing a sense of closeness (Lenhart, 2015).

5.1.2 New Approachable and Accessible Instruments

As seen in iOS instruments such as those discussed in section 3.1.2, composers can now craft musical experiences through customized iOS interfaces. Composers can work anywhere along the continuum between articulating the score to designing the instrument(s) and guiding the performance. The created instruments can mimic established musical instrument design paradigms or develop an interactive language of their own (Vasquez, Tahiroglu, & Kildal, 2017). Furthermore, these instruments can be delivered over electronic devices that the public already has access to, potentially removing actual or perceived barriers to musical consumption and performances.

5.1.3 Audience-Centered and Individualized Experiences

In traditional concert or media experiences, audience members are grouped as one entity, fed similar content, and placed in a fixed position throughout the performance. With formats such as 360 videos, however, each audience member can be placed anywhere within the experience, as well as given agency over his or her unique experience. Composers creating works for AR or VR can choose when and how to grant listeners varying degrees of control over their interaction, reception, and experience of the work.

5.1.4 Transportable and Transformable Audiences

The location of a concert and the body the audience experiences it from is now completely transformable or transportable. Reeps One's "Does Not Exist" 360 video, in which the audience is transported to multiple locations with the performer, provides a basic example of a transportable concert hall. Robin Arnott's *SoundSelf*, on the other hand, demonstrates how the audience's physical presence can be altered or emphasized in ways previously impossible. VR, currently best accessed through a HMD, discards a listener's body upon entry and allows the creator to determine how much or in what way the listener's body will be translated into the new digital world.

5.1.5 Shared Time Controls

While 360 videos maintain a linear timeline and leave control of time's passage solely in the hands of the creator, all other new formats discussed in this document present the option to

break from this limitation. Of course, creators still must draw out the laws that will govern their work; they may choose to construct a world where inhabitants have varying degrees of control or influence over their experience of time. Bluebrain's location-based album apps show one potential model for other composers to adopt, whereas Plan8's *PolyDome* and Particle Inc.'s *Drops* demonstrate a more sculptural approach.

5.1.6 The MP3 of Installation Art

Similar to the possibilities that the MP3 created for music recordings, the most encapsulating benefit that these new formats bring is their ability to make interactive and immersive experiences highly portable and transferable. Works that previously had required dedicated and specialized locations, equipment, and technology to be performed can now be reimaged and bundled as a single digital document that is shareable across a range of general-use electronic devices.

5.1.7 Pure Algorithmic

As noted by Brian Eno (see page 94), composers of algorithmic music may now package their algorithms into apps rather than release fragmented recordings of resulting music. Furthermore, certain parameters of an algorithm can be exposed to the audience, allowing listeners to engage with, alter, or guide the results.

5.1.8 Altered Physics

The virtual environments of manufactured digital universes often mirror the laws that govern the physical world. But this is a choice, not a requirement. Theoretically, these worlds could be mapped out in ways that allowed audiences to engage in experiences not otherwise possible or practical in the real world. One simple example would be to alter the way that sounds interact with spaces. The speed of sound could remain faithful to the real world but the natural attenuation resulting from the distance between the source and the listener could be negated. This could breathe new life into the phasing ideas often present in composer Steve Reich's music. New and more inventive ideas will likely arise as creators and listeners begin to free their minds from the limitation of the natural world.

5.2 Specific Environmental Considerations

5.2.1 360 Videos

360 videos solve many of the issues that previously prevented spatial audio from becoming a widely distributed format. Today's 360 video players are compatible with or are already installed on most mobile devices (see section 4.1.2), head-tracking (see section 2.2) is managed by the playback device, and the videos themselves can be embedded with both stereo or Ambisonic (see section 2.3.3) audio files for delivering head-locked (see section 2.2) and spatial audio.

Composing for 360 videos is approached similarly to other fixed or linear formats. Composers can even continue working in their DAW of choice as long as it supports the requisite number of channels (see section 4.4) for the targeted Ambisonic output order (see section 2.3.3) and the necessary Ambisonic plug-ins (see section 4.5) for editing and spatializing their audio.

Beyond traditional spatial audio, 360 videos offer composers the ability to craft music or sounds that can be delivered as either head-locked or spatialized audio. For example, a sound designer can embed the sounds of the listener's heartbeat and breath in the head-locked track, while attaching the music to a radio that is visible within the scene. By doing this, the music will remain spatially fixed as viewers look around the 360-degree scene, and their heartbeat and breath sounds will move with them.

Despite the growing popularity of 360 videos, creators must still consider that their viewers may not be familiar with 360 videos or listening to spatial audio. On-boarding, or taking time at the beginning of the video to provide the audience with instructive visual and sonic cues, may help them understand how to navigate and what to expect from this new format. In addition, keeping the amount of audio relatively minimal may allow the listener to get a stronger sense of the different components that make up the spatialized soundscapes.

5.2.2 iOS Instruments and Album Apps

As the examples from chapter 3 demonstrate, iOS instruments and album apps are a vibrant format for musical expression and exploration. Through iOS apps, composers can now create musical experiences that completely blur the lines between score, instrument, performer,

listener, album, and performance. The iOS platform can meld both linear and non-linear approaches, as well as support pre-recorded and real-time sound synthesis/processing.

From a sonic perspective, however, iOS instruments and albums apps are, and perhaps should remain, the most two dimensional of all formats discussed in this document. This is because, unlike the other formats, iOS instruments and album apps (except those that incorporate AR elements) do not pretend to project virtual objects beyond the screen nor around the listener—everything exists exclusively on the screen. What’s more, binaural audio is still most commonly rendered via a generalized set of HRTFs (see section 2.3.4), which position a sound above, below, or behind the listener. This makes it challenging for the listener to determine the location of a sound without head-tracking or the ability to associate the sound with a visible object.

A final consideration for authoring audio for iOS instruments or apps is the audio delivery system. For optimal sound, a simple on-boarding sequence could recommend, or even insist, that listeners use headphones as opposed to the device’s built-in speakers, which often have significantly reduced frequency ranges and mono summation of all sound sources. (For more on delivery systems, see section 2.2.)

5.2.3 Augmented and Virtual Reality

AR and VR open similar worlds of sonic and experiential possibilities. The major differentiator between the two formats is that AR is an expansion of reality while VR is a departure from it. This is an essential concept for composers to consider when creating works for either format. Works for AR should consider or respond to the spaces and objects present in the audience’s immediate real-world surroundings. VR creators, on the contrary, are theoretically free

to design worlds full of sounds, objects, and laws that behave in ways completely foreign to our reality, as well as ones that are completely disassociated from anything in the viewer's physical surroundings.

Despite these new possibilities, however, VR may be momentarily in evolutionary limbo, stuck somewhere between this land of total creative freedoms and the expectations a person develops from life in the real world. Sounds, on the one hand, were liberated from the real world over half a century ago by recording technologies, and modern listeners are not regularly surprised by or consciously aware of the many unlikely sonic scenarios implied by the recordings they consume. Leap Motion, on the other hand, warns against sensory conflicts and suggests that VR design should be informed by the behavioral expectations users develop from their experiences in the real world (Leap Motion, 2015). Yet many current game sound designers and spatial audio SDKs (see section 4.8) have rejected the audio norms of the real world. A common example of this is how whenever a distant enemy shoots at the player in a video game, the flash and sound of the gunshot occur simultaneously from the player's perspective.

Contending with a vast field of options and the previously established expectations of audiences is not a new problem for composers. Composers in each new generation receive a wider pallet of resources with which to express their creative ideas, but the act of creation has always been one rooted in curation. Though working in AR and VR introduces a multitude of new possibilities, composers should continue to curate their material options in ways that are considerate of their audiences' expectations.

5.3 Catalog of Techniques

The resulting flood of user-related data and the flexibility of these new immersive digital formats present a variety of new compositional techniques and approaches. Below the methods are cataloged individually, but in practice they could be deployed in more sophisticated configurations to achieve more responsive systems.

5.3.1 Time Triggers

After a certain amount of time elapses following a monitored event, something happens. For example, if the audience remains within a space for a predetermined amount of time, a new musical element is introduced.

5.3.2 Gaze Activation

Via an HMD (see section 4.3.1), eye tracking (see section 4.3.5), or a 360 video with focus detection enabled (see section 4.1.1), musical cues could be taken from where or what the viewer is looking at.

5.3.3 Motion Activation

Similar to motion-activated lights, sonic or musical cues could respond to whether the audience is in motion or at rest.

5.3.4 Real-Time Player-Controlled Parameters

Any additional position, motion, or invented player parameters could be used to control aspects of the audience's experience. For example, their direction or speed of movement, the distance between their hands, or their in-game health.

5.3.5 Group Controlled Parameters

In a group experience, such as an MMO game, data extracted from the relationship between the audience members could be used to trigger or govern musical parameters. For example, if the entire audience is looking at the same object, a different sound results than if only one person is looking at the object.

5.3.6 Real-Time Microphone Input

Except for 360 videos, all other formats discussed in this document can support real-time input from a device's microphone. This could be used for either live-streaming musical performances in real-time or, as demonstrated by Robin Arnott's *SoundSelf*, for more complex audio control and world manipulation.

5.3.7 Distance and Time-of-Arrival Phasing

At the moment, none of the game engines, audio middleware, or SDKs discussed can alter a sound's arrival time based upon its distance from a listener, but a simple script could be written to mimic or even exaggerate this behavior. This would allow for phasing ideas, similar to those of Steve Reich, to be explored through an object's physical distance from the listener and each other.

5.3.8 Unrealistic Scale and Proportion

Creations in the real world are limited by practical concerns. The virtual world, in contrast, is limited only by the hosting device's resources (e.g., memory, processor, battery life). As a result, ideas can be realized at scales and proportions that would be impractical in the real world—for example, a multichannel work for 4,096 speakers that are animated over a void that stretches several hundred miles in all directions.

5.3.9 Proximity-Based Deconstructions

VR enables a viewer to travel through spaces and into objects in ways not possible in the real world, which opens many new ways for composers to conceive of musical forms and gestural development. For example, a sphere floating in a virtual world could be the source of a single note of a piano that contributes to a denser texture emitting from a network of surrounding

spheres. As the viewer approaches the sphere, the piano note grows louder and begins to spectrally deconstruct. Then, if the viewer moves too close, he or she gets sucked inside the sphere and into an entirely new musical texture made of the spectral components of the piano note.

5.3.10 Updates and Evolving Projects

Works built for these formats may require regular updates to stay compatible and functional. Mobile app users also have come to expect regular updates for their applications, based on their experiences with consumer software. This may seem like a nuisance, but it also presents interesting possibilities to develop and grow a work over time.

Chapter 6: *Infinite Void*—An infinite series of works for VR

6.1 A Framework for Future Development

An aspect of digital formats that has attracted me to media such as mobile apps or AR/VR experiences is the option for creators to push updates to their audiences as frequently as desired. Users of modern software often expect semi-regular updates to their applications that improve or expand various components of the experience. This ability to provide continual updates inspired the initial goal behind my VR work, *Infinite Void*, which accompanies this dissertation—to create a framework and environment that would allow for future revision or additions to the worlds and experiences it contains. Furthermore, rather than outsourcing many of the tasks not usually delegated to the composer, I wanted to create everything myself. While I am not opposed to collaboration and hope to work with visual artists/programmers in the future, I believe that acquiring the full range of skills needed to create this work will allow me to independently pursue my ideas, as well as make me a better collaborator with performers in other disciplines.

The framework I developed for *Infinite Void* using Unity and Wwise established some rudimentary functionality: basic user locomotion, button input from the Oculus Go controller, the ability to trigger events from a variety of user or environmental stimuli, and the option to identify certain objects or zones as portals that will transport the user from one scene to another (see section 4.6). On top of this basic framework, *Infinite Void* relies upon Wwise and its spatial audio capabilities to handle all in-world music and sound design.

6.2 Why Virtual Reality

I decided to construct *Infinite Void* in VR, as opposed to the other formats referenced in this document, for two reasons. First, VR encapsulates nearly all of the possibilities and capabilities of these new formats, so by learning how to construct and work within virtual worlds, I believed I would subsequently gain the knowledge and skills necessary to create works for the other formats. Second, many of my previous works sought to completely immerse the audience in the musical experience, and of all the new formats, VR is the most immersive, allowing me to realize a long-held aspiration.

6.3 The Learning Process

Learning to create this work has required me to acquire many new skills. Thankfully, most VR-related hardware and software manufacturers, recognizing that they need competent content creators to use their platforms for them to be successful, have produced numerous educational resources. For this project, I most heavily relied on Unity's online tutorials (Unity, 2019), Audiokinetic's online Wwise certification program (Audiokinetic, 2017), and Oculus' Developer Documentation (Oculus, 2014). In addition, Marco Gillies' article "Setting up Unity for Oculus GO development" (Gillies, 2018) was crucial to properly configuring my computer and Oculus Go HMD for development.

6.4 Why Unity?

Despite Unreal Engine's forthcoming audio engine and my familiarity with its somewhat similar approach to scripting via Blueprints (see section 4.6.3), I chose to develop using Unity for several reasons. Unity is more popular among independent game developers—with whom I hope to collaborate in the future—and I was able to learn the platform much more quickly than Unreal. I will continue to monitor Unreal's development and may consider switching once its audio engine is officially released, especially if it supports real-time synthesis, which Unity currently lacks.

6.5 Why Wwise?

Of the two audio middleware discussed in this document (see section 4.7), Wwise is hardly reminiscent of traditional DAW workflows and presents a foreign graphical user interface. However, Wwise's hierarchical organization and parameter inheritance systems (see section 4.7.2) provide an efficient environment for authoring complex musical processes and non-linear behaviors, both of which are features of VR that I wanted to explore.

FMOD can design interactive music systems that behave similarly to those authored in Wwise, but the composer must do so from a more linear perspective. Through some of my earlier experiments with the two programs, I tended to create more traditional musical forms in FMOD, while Wwise's unfamiliar environment forced me to approach composition in unfamiliar ways. In addition to this shift of perspective, Wwise's ability to handle real-time audio synthesis, MIDI playback, and advanced sound spatialization and acoustic simulations (see section 4.7.2 for more details) solidified my decision.

6.6 User Locomotion

Despite many VR development guides warning against all modes of user locomotion besides teleportation to avoid inducing motion sickness, in my own experimentation, I found no issues with propelling myself around in VR by using the orientation of the HMD and the buttons of the Oculus Go controller. Once inside *Infinite Void*, viewers can move around by looking at the spot and depressing either the index trigger to move toward it or the thumb pad to move away from it. The C# script (see Appendix A) includes the public variable `speed` for adjusting the rate of locomotion.

6.7 Triggerable Events

Events are triggered by the user's actions in each scene of *Infinite Void*. As an example, in one scene the viewer teleports into a complete void. However, if they search around themselves, they will discover that there are spheres orbiting them that become illuminated and begin to produce sounds when looked at directly. Once a sphere is illuminated, the player can tap the thumb pad of the controller to change the sphere's color, indicating that it will remain visible and produce its sound even when the player shifts their gaze. This behavior is accomplished via two C# scripts. The first is attached to the player's game object, and the other is attached to each of the surrounding game objects.

The C# script attached to the player's game object (see Appendix B) uses a ray casting function to detect any game object that the player looks at directly. If the detected game object matches a certain set of criteria, the "Player_Raycast" script will prompt the "Sphere_Response_Script" to perform several functions.

The C# script attached to the game object (see Appendix C) will handle illuminating the object when the player looks directly at it, starting or stopping the associated Wwise events and locking or unlocking the sphere when the controller's thumb pad is pressed.

6.8 Scene Portals

Any game object in Unity, either visible or invisible, can detect collision with another game object. In *Infinite Void*, I used a simple C# script (see Appendix D) that can transform any game object into a portal for the player to pass from one scene to another. The script first triggers an animation defined in Unity that fades the current scene to black before loading the new scene and fading back from black. At this time, I have given myself the freedom to make any object a potential gateway to another experience, to expand the structural possibilities within *Infinite Void* and encourage the viewer to completely explore their surroundings. As my abilities improve and *Infinite Void* expands, it may become useful to identify a consistent object that will always serve as a scene portal or, conversely, make every visible object a portal to another scene.

6.9 Evolutions Orbiting

The initial version of *Infinite Void* includes two scenes that rely on the viewer's movements and actions to generate and control the musical materials and their development over time. The first scene, "Evolutions Orbiting," surrounds the viewer with 64 invisible spheres (see Figure 6-1). No sphere is visible or produces any sound until a Wwise event is triggered by the viewer looking directly at a sphere.

A single Wwise event is tightly spatialized and associated with each sphere. The Wwise event is called within the “Sphere_Response_Script” (see Appendix C) when the sphere is detected by the “Player_Raycast” script (see Appendix B) and plays a single iteration of a complex music playlist container (see Figure 6-2).

The first stage within the music playlist container randomly selects one of twelve possible subgroups. Each subgroup consists of a single pitch class and is contained within a sequence step group that first plays a single strike of a crotale before infinitely sustaining an evolving sample of a full string orchestra playing an octave equivalent of the struck crotale note (see Figure 6-2).

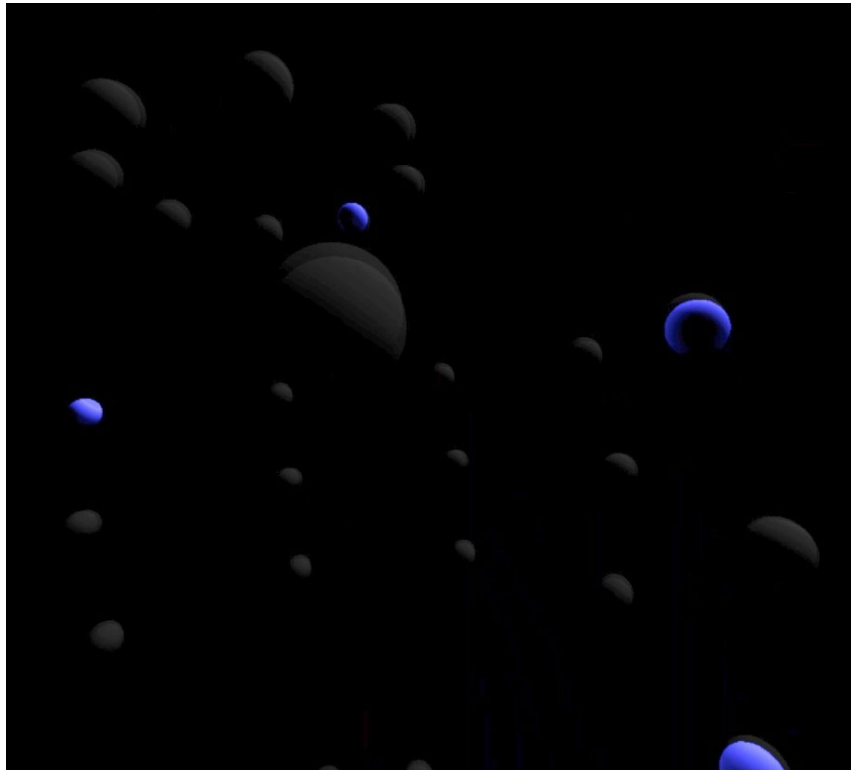


Figure 6-1. Screen capture of the "Evolutions Orbiting" from the initial version of *Infinite Void*.

Group/Segment	Random Type	Avoid Repeat	Weight	Loop Count
Random Step	Standard	12		1
Sequence Step	Standard	1	50	2
Random Step	Standard	1	50	1
CinePerc_Crotales_A5			50	1
CinePerc_Crotales_A6			50	1
Random Step	Standard	1	50	1
String_Evo_04_A0			50	Infinib
String_Evo_04_A1			50	Infinib
String_Evo_04_A2			50	Infinib
String_Evo_04_A3			50	Infinib
String_Evo_04_A4			50	Infinib
String_Evo_04_A5			50	Infinib
Sequence Step	Standard	1	50	2
Random Step	Standard	1	50	1
CinePerc_Crotales_Ab5			50	1
CinePerc_Crotales_Ab6			50	1
Random Step	Standard	1	50	1
String_Evo_04_Ab0			50	Infinib
String_Evo_04_Ab1			50	Infinib
String_Evo_04_Ab2			50	Infinib
String_Evo_04_Ab3			50	Infinib
String_Evo_04_Ab4			50	Infinib
String_Evo_04_Ab5			50	Infinib

Figure 6-2. Screen capture of the first two subgroups of the Wwise music playlist container for the "Evolutions Orbiting" scene of *Infinite Void*.

In addition to flying around the scene (see User Locomotion), the viewer can control the active or inactive state of a sphere, even when it is not within focus, by looking at the desired sphere and lightly pressing the controller's touch pad. These simple modes of interaction, coupled with the narrowly spatialized sounds attached to animated game objects, place the listener in a swirling and evolving drone of their own design.

6.10 Granular Constellations

The second scene included in the initial version of *Infinite Void* focuses on how the locational relationships between the viewer and surrounding objects can enable the spatial exploration of musical materials that are suspended in time through granular synthesis (Roads, 2001).

“Granular Constellations” uses a small script to automatically create and distribute flicking stars within a designated space (see Appendix E). Each star serves as both the emanation and spatial-reference point for a granularized audio recording. A Wwise event is associated with each star and plays a discrete audio recording through Wwise’s SoundSeed Grain plug-in (see Figure 6-3) based on the star’s real-time distance from the player as measured across three planes: X, Y, and Z. The three distances are communicated as real-time parameter controls (RTPC)s (see Game Syncs) to the Wwise sound engine in one C# script (see Appendix F). Each RTPC controls a single parameter of the associated granularized audio file (see Figure 6-4).

The X RTPC (“X_Distance”) controls the duration of each produced grain. As the player nears a star, the grains get shorter. As the player moves farther away, the grains get longer.

The Y RTPC (“Y_Distance”) manipulates each grain’s amplitude envelop. From far away, each grain has a smooth attack and release. As the player moves closer to a star, the amplitude envelope of each grain becomes more abrupt.

The Z RTPC (“Z_Distance”) determines the amount of grains produced per second. Getting closed to a star increases the number of grains per second. Moving away from the star slows down the rate of emission.

Finally, the absolute distance from a star to the player, also communicated as an RTPC (“Grain_Distance”), determines the playback position within the chosen audio file.

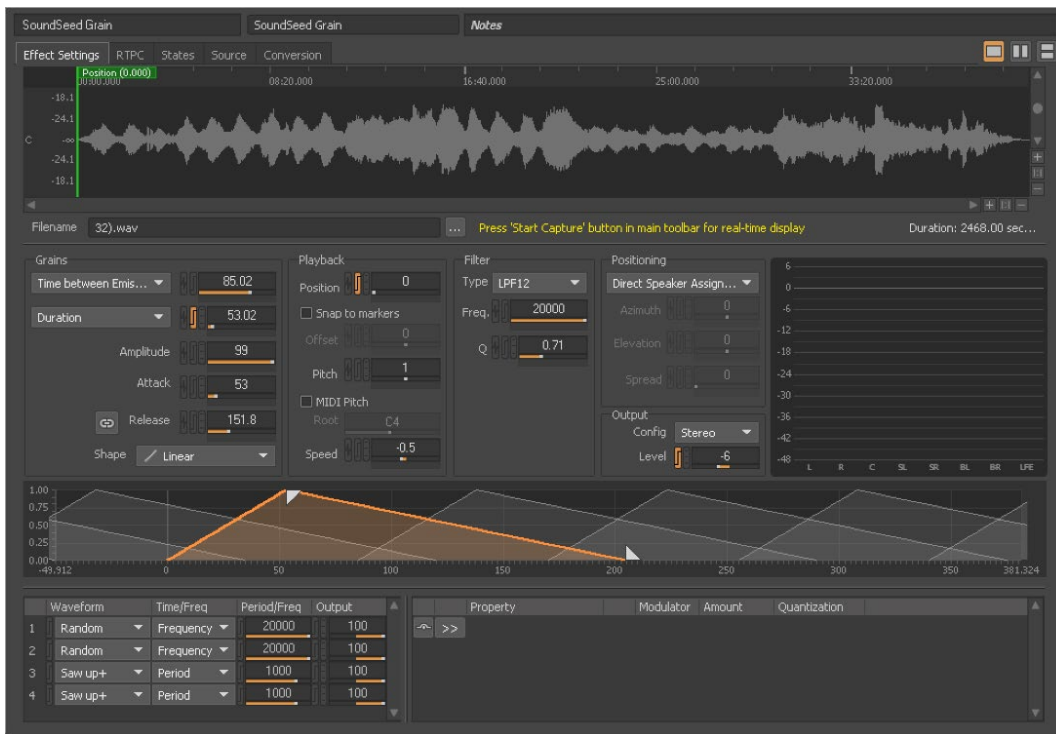


Figure 6-3. Screen capture of Wwise's SoundSeed Grain source plug-in as used in the "Granular Constellations" scene of *Infinite Void*.

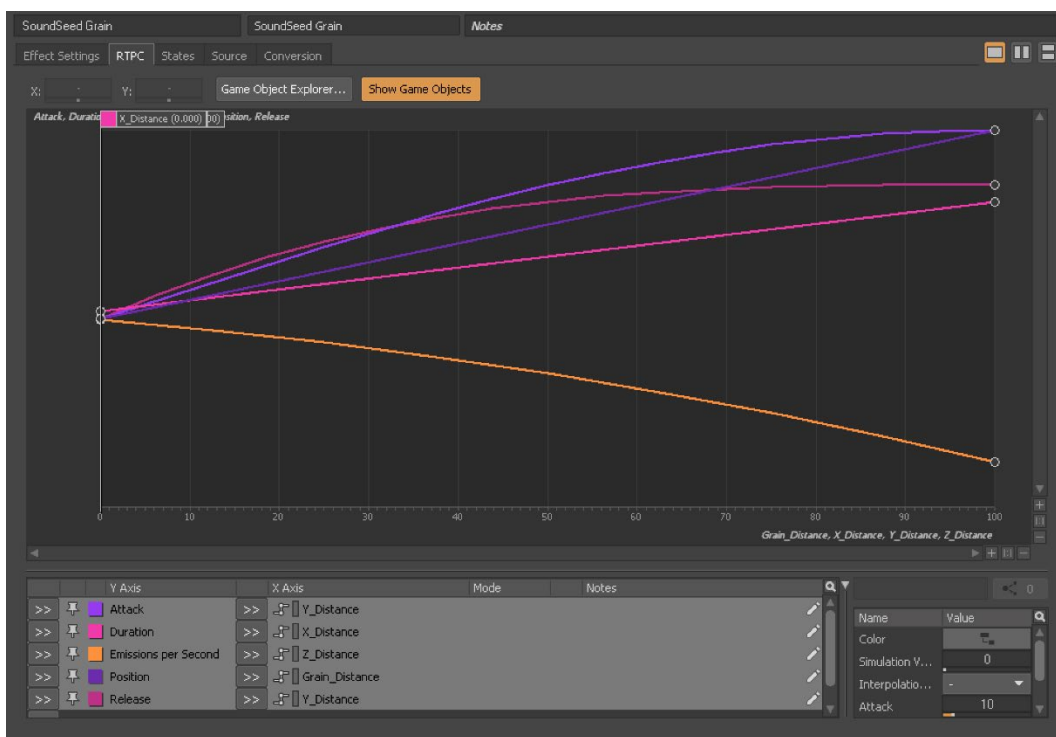


Figure 6-4. Screen capture showing the RTPC mapping for *Infinite Void*'s "Granular Constellations."

To give the listener a stronger sense of motion and presence, the sound produced from each star is spatialized in Wwise according to the listener's position and orientation. In addition to reducing a star's output volume, Wwise also narrows the frequency response range of each star and increases the amount of reverb applied to its output as the listener moves farther away. Rather than completely silencing a sound that is beyond the maximum distance of 100 units, however, the sound is transformed into a ghost-like impression of itself by heavily filtering its output, increasing the amount of reverb, and reducing the volume by approximately 50 dB (see Figure 6-5).

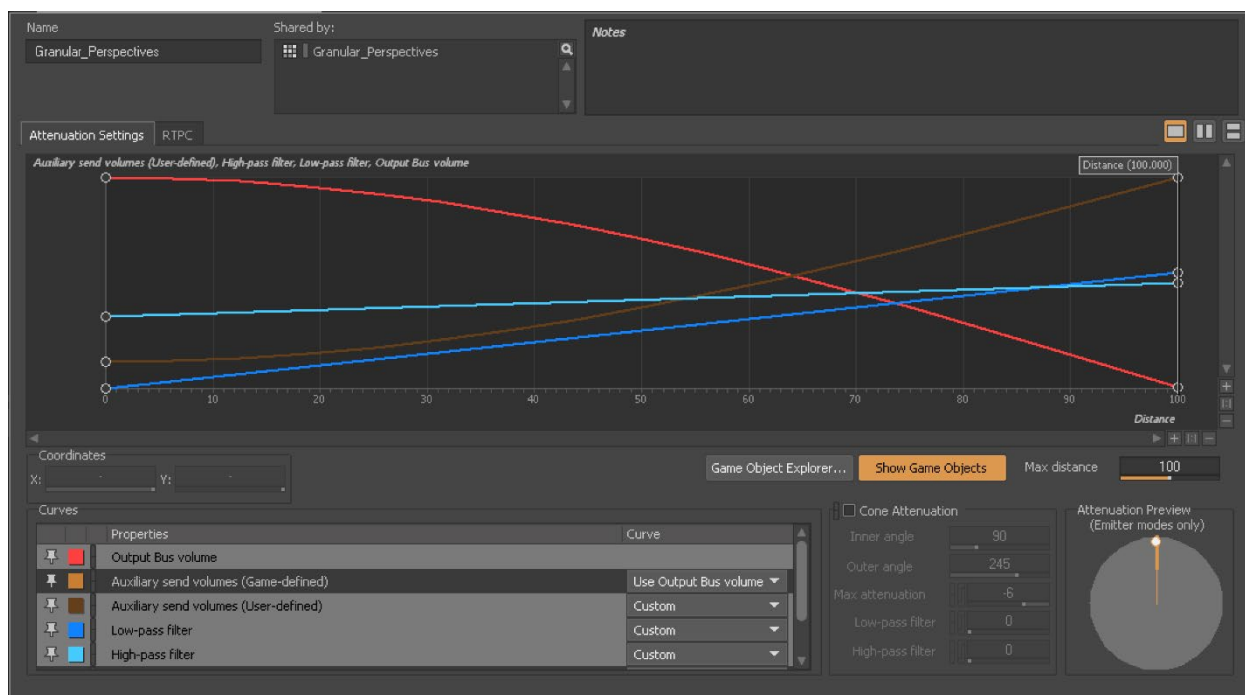


Figure 6-5. Screen capture showing the attenuation curves used in *Infinite Void's* "Granular Constellations" scene.

6.11 Plans for Future Development

In addition to adding scenes that explore some of the other techniques discussed in Chapter 5, both “Evolutions Orbiting” and “Granular Constellations” soon will be further developed to add time triggers (see Time Triggers) and proximity-based deconstructions (see Proximity-Based Deconstructions).

Currently “Evolutions Orbiting” begins with, and maintains, a relatively stable sonic environment. In future versions, I would like the use time triggers to begin the scene with more chaotic textures that become gradually more refined as time passes. Conversely, in “Granular Constellations,” time triggers will be used to reorient a star’s spatial relationships, rather than its sonic material, so that the listener can explore combinations of sound and texture that were inaccessible in the previous star arrangement.

In both scenes, proximity-based deconstructions will be added so that as a listener nears a sonic object a greater sense of depth and detail will emerge from within it. This will allow the listener to gain a better understanding of the sonic particulates that fill each scene within *Infinite Void*.

Chapter 7: Conclusion

For many, the new interactive and immersive digital formats discussed in this document are ushering in the utopian future technology has always promised. Several studies have suggested that unbelievable results may be achievable through treating medical patients in VR (Bobkova, 2017) and social activists see VR as an “empathy machine” capable of forging previously impossible human connections (Archer & Finger, 2018; Rubin, 2018).

In his article “Keep virtual reality weird: Anything is possible, which is why Resident Evil is so boring,” journalist Ben Kuchera shared an email he received from Isaac Cohen, the developer of the LUNE VR experience, that illustrates the philosophical and artistic potential of VR as a medium:

As I get older, I feel like I have been gaining more and more perspectives of 'Reality' and what that means. It could be meeting somebody from another culture, experiencing a new art piece that said something that I had not realized before, or just staring a fern in a different way. These new perspectives make me feel like I understand more of reality (although as I understand more, I realize how much more MASSIVE it was than I ever thought) and [it] makes me feel like I can see it with more clarity, more fidelity. It's so, So, SO beautiful this thing that I'm examining, but as I learn more, as I age, I feel myself moving away from it. It is so hard, to have this feeling of 'Oh wait! now I understand! PLEASE, I need to be closer to that thing. I need to be one with that thing!!!'

but still feel yourself floating away from it, drifting in this space. It's this universal longing that only universal separation can explain. [...]

This realization hurt me for a while. It felt like I couldn't be with that thing. But as I've considered it more and more, I've come to the understanding that there is this small part of me, this iridescent gem at the center of my being, that is a part of it. That is the thing itself. It reminds me of the fable of Indra's Net, and the multitude of gems, all reflecting the entirety of the universe in each node. Making up the entirety of the universe at each node. [...]

LUNE to me is this Cycle of Loneliness -> Enveloped by the Beloved -> Seeing once what was slowly mov[ing] away. looking at that thing with both love and sorrow, content that you had time with it, but still longing for it again -> Loneliness [...]

So how close do I think I came? I have no idea. I think at least [one] person got it. But I think that we will never be fully there. I just hope I can make a tool that helps people practice that movement, and recognize that they themselves are one with the divine. (Kuchera, 2016)

Cohen's email poetically captures the sublime potential that so many creators see in the digital formats that have emerged in the last decade. But not everyone sees the potentials of these new platforms in the same light. Instead, some warn against their power to support destructive cultural appropriation (Yang, 2017), misogyny (Belamire, 2016), and more.

In his 2016 article "How VR Gaming will Wake Us Up to our Fake Worlds," multidisciplinary artist and author Elliott Edge issues a warning call to his readers:

[...] imagine virtual realities that are more meaningful, beautiful, ecumenical, compassionate, sustainable, efficient, and indeed more loving to play. In a word: wiser. The call upon the present generations is to radically shift gears and awaken to what we are really doing. We will accept our responsibility for this crisis or we will go down in the history of the universe as another menagerie of pseudo-intelligent space bumpkins at the far end of the galactic arm who collapsed into dust under the weight of their own illusions. This is the warning shot echoing through modernity fired by none other than Mary Shelley in *Frankenstein*; or, *The Modern Prometheus*. We will end at the hands of none other than both our own psychic and physical creations—our virtual reality worlds—unless we evolve them. [...] It's really that simple.

Our most powerful and influential cultures are far from benign. Culture eats nature; our worst VRs eat the world. [...] The challenge and the gift that we have given ourselves is a whole planet on the brink—the ultimate narrative, the life or death game par excellence. The creator's creation now threatens the continuity of life itself, when its hope was always to uplift. All in all, it's a very exciting time to be alive. You won't get a better opening act to the primate's next stage of awakening than the one where it has fallen into its own otherwise brilliant, multi-millennia in the making trap. The question bubbling up as the Information Age gives birth to the Virtual one is; "What will end up being born out of our totally humanized enclosure when we push hard enough against the thresholds of our collective fantasy worlds?" You will know where you truly are by the writing on the wall, coupled with the acidity index of the ocean.

Before any of these apocalyptic or utopian futures can come to fruition, however, there are many roadblocks that these new formats must first overcome.

Creators currently face a perhaps insurmountable mountain of technical knowledge that must be internalized before they can even begin dreaming in these new formats. As previously mentioned, many manufacturers are aware of this problem, and while they have done a great deal to assist artists in overcoming these hurdles, more help is needed. In addition, increasing the likelihood that their creations would survive inevitable software and hardware changes might encourage more artists to invest the required time and effort. This could be accomplished through further standardization, so long as the new standards do not limit the creative potential of each format.

The formats themselves also need to evolve. So much attention is given to replicating the capabilities and possibilities of reality or other established formats that the unique qualities of these digitally augmented or removed worlds remain undiscovered. This is not to say that the pursuit of reality duplication must be abandoned; rather, the end goal must be moved far beyond, into the yet unknown. And realizing that goal will require the involvement of everyone using these new interactive and immersive digital formats—not only the artists, developers and manufacturers creating them but also the audiences who ultimately bring them to life.

[It's] up to us as players to bring something, too: we must bring vision and a willingness to experience. [VR], after all, isn't a passive trip into a story, like an afternoon spent with a good book. [VR] requires active engagement, movement and consideration. We must be the ones to take things apart. Games can only give us the pieces. What we do with them is always up to us (Batti & Karabinus, 2017).

Works Cited

- 6D.AI. (2018, November 27). *6D.AI Developer Documentation*. Retrieved from 6D.AI: <https://dashboard.6d.ai/user/dashboard/?view=home>
- Anderson, E. (2017, May 6). *Narrative Isn't the End, but a Means to Reach It*. (N. C. Gaming, Producer) Retrieved from nocoastgaming.com/2017/05/06/narrative-isnt-the-end-but-a-means-to-reach-it/
- Apple. (2017, October 27). *About Face ID advanced technology*. Retrieved from Apple Support: <https://support.apple.com/en-us/HT208108>
- Apple. (2017, July 21). *Introducing ARKit: Augmented Reality for iOS*. Retrieved from Apple Developer: <https://developer.apple.com/videos/play/wwdc2017/602/>
- Apple. (2018, June 7). *Apple Unveils ARKit 2*. Retrieved from Apple: www.apple.com/newsroom/2018/06/apple-unveils-arkit-2/
- Apple. (2018, March 3). *Augmented Reality for iOS*. Retrieved from Apple: <https://www.apple.com/ios/augmented-reality/>
- Apple. (2018, July 20). *Scanning and Detecting 3D Object*. Retrieved from Apple Developer Documentation: https://developer.apple.com/documentation/arkit/scanning_and_detecting_3d_objects
- Archer, D., & Finger, K. (2018, March 15). *Walking in Another's Virtual Shoes: Do 360-Degree Video News Stories Generate Empathy in Viewers?* (C. J. Review, Producer) Retrieved from www.cjr.org/tow_center_reports/virtual-reality-news-empathy.php
- Atlantic Re:Think. (2015, September 9). *How Fast Is Technology Accelerating?* (T. Atlantic, Producer) Retrieved from www.theatlantic.com/sponsored/prudential-great-expectations/how-fast-is-technology-accelerating/360/
- Audiokenetic. (2017, July 27). *Wwise*. Retrieved from Audiokenetic - Wwise: <https://www.audiokinetic.com/products/wwise/>
- Audiokenetic. (2019). *Wwise Spatial Audio: Create realistic or out of this world dynamic immersive experiences!* Retrieved from Audiokenetic: <https://www.audiokinetic.com/products/wwise-spatial-audio/>
- Audiokinetic. (2017, April 18). *Wwise Certification*. Retrieved from Audiokinetic: <https://www.audiokinetic.com/learn/certifications/>
- Avatar Orchestra Metaverse. (2003, March 23). *Homepage*. Retrieved from <http://avatarorchestra.blogspot.com/>

- Baalman, M. A. (2010). Spatial Composition Techniques and Sound Spatialisation Technologies. *Organised Sound*, 15(03), 209–218.
- Bailenson, J. (2018). *Experience on Demand: What Virtual Reality Is, How It Works, and What It Can Do*. New York, NY: W. W. Norton & Company.
- Bartle, R. A. (2016). *MMOs from the Insdie Out: The History, Design, Fun, and Art of Massively-Multiplayer Online Role-Playing Games*. New York, NY: Apress, Springer Science+Business Media.
- Batti, B., & Karabinus, A. (2017, May 1). *A Dream of Embodied Experience: On Ian Bogost, Epistemological Gatekeeping, and the Holodeck*. Retrieved from NY Gamer: www.nymgamer.com/?p=16363
- Belamire, J. (2016, October 21). *My First Virtual Reality Groping*. (Mic, Producer) Retrieved from mic.com/articles/157415/my-first-virtual-reality-groping-sexual-assault-in-vr-harassment-in-tech-jordan-belamire#.p6gg5NrLq
- Bizri, H. (2000). Story Telling in Virtual Reality. *Leonardo*, 33(1), 17–19.
- Björk. (2011, July 19). *Björk Biophilia App intro Narrated by David Attenborough*. Retrieved from <https://youtu.be/o8AELvVUFLw>
- Björk, & Huang, A. T. (2015, June 6). *Björk: Stonemilker (360 Dree Virtual Reality)*. Retrieved from YouTube: www.youtube.com/watch?time_continue=72&v=gQEyezu7G20
- Blue Ripple Sound. (2011, May 26). *HOA Technical Notes - Introduction to Higher Order Ambisonics*. Retrieved from www.blueripplesound.com/hoa-introduction
- Bluebrain. (2018, June 27). *Site: Nine Elms*. Retrieved from iTunes App Store: <https://itunes.apple.com/us/app/site-nine-elms/id1088941378>
- Blum, A. (2018, March 19). *ARCore vs. ARKit: Which Is Better for Building Augmented Reality Apps?* Retrieved from Iflexion: <https://www.iflexion.com/blog/arcore-vs-arkit-better-building-augmented-reality-apps/>
- Bobkova, J. (2017, June 21). *The Potential of Virtual Reality in Healthcare*. Retrieved from Medium: [The Potential of Virtual Reality in Healthcare](https://medium.com/@bobkova/the-potential-of-virtual-reality-in-healthcare)
- Bogost, I. (2017, April 25). *Video Games Are Better Without Stories*. (T. Atlantic, Producer) Retrieved from www.theatlantic.com/technology/archive/2017/04/video-games-stories/524148/?utm_source=atltw
- Bohn, D. A. (2016, December 20). *Pro Audio Reference (PAR)*. (A. E. Society, Producer) Retrieved from www.aes.org/par/

- Bonnington, C. (2015, March 13). *YOU CAN NOW WATCH AND UPLOAD 360-DEGREE VIDEOS ON YOUTUBE*. Retrieved from Wired:
<https://www.wired.com/2015/03/youtube-360-degree-video/>
- Brackeys. (2018, May 20). *How to Fade Between Scenes in Unity*. Retrieved from YouTube:
<https://youtu.be/Oadq-IrOazg>
- Brogan, J. (2017, September 1). *Our Virtual Future: What Philosophers Say About the Inevitability of VR*. (Slate, Producer) Retrieved from slate.com/technology/2017/09/what-philosophers-say-about-the-inevitability-of-virtual-reality.html
- Bucher, J. K. (2018). *Storytelling for Virtual Reality: Methods and Principles for Crafting Immersive Narratives*. New York, NY: Routledge.
- Buckley, S. (2015, August 19). *This Is How Valve's Amazing Lighthouse Tracking Technology Works*. Retrieved from Gizmodo: <https://gizmodo.com/this-is-how-valve-s-amazing-lighthouse-tracking-technol-1705356768>
- Buzzi, M. (2018, December 7). *The Best Laptops for VR in 2019*. Retrieved from PC Magazine:
<https://www.pcmag.com/roundup/360989/the-best-laptops-for-vr>
- Cabrera, A. (2014, February 27). *Ambisonics*. (Ambisonics, Producer) Retrieved from w2.mat.ucsb.edu/240/D/notes/Ambisonics.html
- CDM Blogs. (2008, October 8). *Hands-on with Bloom, New Generative iPhone App by Eno and Chilvers*. Retrieved from Create Digital Music: Hands-on with Bloom, New Generative iPhone App by Eno and Chilvers
- Cheng, W. (2014). *Sound Play: Video Games and the Musical Imagination*. New York, NY: Oxford University Press.
- Christiansen, P. (2018, November 15). *Presence, Preservation, and Virtual Reality*. (Playthepast, Producer) Retrieved from www.playthepast.org/?p=6181
- Cockos. (2009, June 28). *JSFX Programming*. Retrieved from Reaper:
<http://reaper.fm/sdk/js/js.php>
- Code, B. (2016, November 7). *Video Games are Boring*. (G. Industry, Producer) Retrieved from www.gamesindustry.biz/articles/2016-11-07-video-games-are-boring
- Colucci, R., & al., e. (2015). Nonlinear Music Analysis and Composition. *European Society for Mathematics and the Arts*, 87–94.
- Conservation International. (2008, January 13). *Homepage*. Retrieved from <https://www.conservation.org>

- Constine, J. (2014, July 22). *Facebook's \$2 Billion Acquisition of Oculus Closes, Now Official*. (T. Crunch, Producer) Retrieved from techcrunch.com/2014/07/21/facebooks-acquisition-of-oculus-closes-now-official/
- Constine, J. (2018, October 27). *Oculus VR Content Will Be Forward-Compatible with ITs Future Headsets*. (T. Crunch, Producer) Retrieved from techcrunch.com/2018/09/26/oculus-compatibility/
- Cox, C., & Warner, D. (2017). *Audio Culture: Readings in Modern Music*. New York, NY: Bloomsbury Academic.
- D'Anastasio, C. (2016, July 6). *VR Has Its Roots In Ancient Rome*. (Kotaku, Producer) Retrieved from kotaku.com/vr-has-its-roots-in-ancient-rome-1783199702
- Deahl, D. (2018, March 27). *You Can Now Control GarageBand with Your Face on iPhone X*. (T. Verge, Producer) Retrieved from www.theverge.com/2018/3/27/17169984/control-garageband-face-arkit-iphone-x
- Dean, R. T., & McLean, A. (2018). *The Oxford Handbook of Algorithmic Music*. New York, NY: Oxford University Press.
- Defunct Games. (2015, September 10). *Defunct Games*. Retrieved from YouTube: <https://youtu.be/Ta2L3Z7OMng>
- Deia, Z. (2018, May 23). *Understanding Surround Sound Formats*. (Crutchfield, Producer) Retrieved from www.crutchfield.com/S-e38JxrWlcR1/learn/learningcenter/home/hometheater_surround.html
- DeLahunta, S. (2022). Virtual Reality and Performance. *PAJ: A Journal of Performance and Art*, 24(1), 105–114.
- Dolby. (2016, March 1). *Dolby Atmos for Virtual Reality*. Retrieved from Dolby: <https://www.dolby.com/us/en/professional/content-creation/vr.html>
- Dombal, R. (2011, June 29). *Björk*. Retrieved from Pitchfork: pitchfork.com/features/interview/7996-bjork/.
- Downey, S. (2014). History of the (Virtual) Worlds. *The Journal of Technology Studies*, 40(1/2), 54–66.
- Dredge, S. (2012, September 26). *Brian Eno and Peter Chilvers talk Scape, iPad apps and generative music*. Retrieved from The Guardian: [https://www.theguardian.com/music/appsblog/2012/sep/26/brian-eno-scape-ipad-apps](http://www.theguardian.com/music/appsblog/2012/sep/26/brian-eno-scape-ipad-apps)
- Durbin, J. (2017, January 27). *Oculus Sensors Are Technically Hackable Webcams*. Retrieved from UploadVR: <https://uploadvr.com/hackable-webcam-oculus-sensor-be-aware/>
- EccoVR. (2016, April 17). *Homepage*. Retrieved from <http://www.eccovr.com/>

- Edge, E. (2016, June 28). *How VR Gaming Will Wake US Up to Our Fake Worlds*. (I. f. Technologies, Producer) Retrieved from ieet.org/index.php/IEET2/more/Edge20160628
- Ellis, C. (2018, February 12). *VLC Media Player now supports 360-degree video and resolutions up to 8K*. Retrieved from Tech Radar: <https://www.techradar.com/news/vlc-media-player-now-supports-360-degree-video-and-resolutions-up-to-8k>
- EMPAC. (2018, March 15). *Wave Field Synthesis System*. (E. M. (EMPAC), Producer) Retrieved from empac.rpi.edu/research/wave-field-synthesis
- Eno, A. (2010, November 23). *Cross-platform MIDI processing library for mono and .NET (ALSA, CoreMIDI, Android, WinMM and UWP)*. Retrieved from Github: <https://github.com/atsushieno/managed-midi>
- Eno, B. (2018, November 28). *Brian Eno sets out to change music (again) with Bloom: 10 World*. Retrieved from Digital Trends: <https://www.digitaltrends.com/music/brian-eno-peter-chilvers-bloom-10-worlds/>
- Eno, B., & Chilvers, P. (2018). *Bloom: 10 Worlds*. Retrieved from Generative Music.com: <http://generativemusic.com/bloom10worlds.html>
- Facebook. (2016, March 13). *Facebook 360*. Retrieved from Facebook 360: <https://facebook360.fb.com/>
- Facebook. (2018, May 1). *F8 2018: Oculus Go Available Now for \$199*. Retrieved from Facebook New Room: <https://newsroom.fb.com/news/2018/05/oculus-go/>
- Farmer, C. (2017, November 13). *Apaix World*. Retrieved from <https://apaix.world/>
- Farmer, C. (2018, November 20). *Making your face the effects automator with an iPhoneX, Unity, Max and Ableton*. Retrieved from Chase Farmer: <http://chasefarmer.com/articles/2018-11-20-making-your-face-the-fx-automator/>
- Fellget, P. (1975). Ambisonics. Part One: General System Description. *Studio Sound*, 17, 20–22.
- Feltham, J. (2018, July 12). *Why Virtual-Virtual Reality Remain A VR Experience You Must Play*. (U. VR, Producer) Retrieved from uploadvr.com/why-virtual-virtual-reality-remains-a-vr-experience-you-must-play/
- Firelight Technologies. (2017, March 28). *FMOD Studio*. Retrieved from FMOD: <https://www.fmod.com/studio>
- Firelight Technologies. (2017, March 18). *Licensing*. Retrieved from FMOD: <https://www.fmod.com/licensing>
- Firelight Technologies. (2018, August 22). *Firelight Technologies FMOD Studio API*. Retrieved from FMOD:

https://www.fmod.com/docs/api/content/generated/common/lowlevel_introduction.html#file-formats-1

- Firelight Technologies. (2018, July 26). *FMOD Studio User Manual*. Retrieved from FMOD: <https://www.fmod.com/resources/documentation-studio?page=welcome-to-fmod-studio.html>
- Flood, K. (2012, February 27). *The Language of Biophilia: A Q&A with Nikki Dibben*. Retrieved from Vice: www.vice.com/en_us/article/4x48ng/the-language-of-ibiophilii-a-qa-with-nikki-dibben
- FMOD. (2018, August 15). *FMOD Studio Plug-in SDK*. Retrieved from FMOD: https://www.fmod.com/resources/documentation-api?page=content/generated/overview/plugin_api_dsp.html#/
- Frankenfield, J. (2018, December 13). *Second Life Economy*. (Investopedia, Producer) Retrieved from www.investopedia.com/terms/s/second-life-economy.asp
- Fredrick, T. (2018, July 11). *Open Sound Control (OSC) for Unity*. Retrieved from Github: <https://github.com/thomasfredericks/UnityOSC>
- Funkhouser, T. A., Tsingos, N., Carlbom, I., Elko, G., Sondhi, M., & West, J. (2005, June 27). *Interactive Acoustic Modeling of Complex Environments*. (P. University, Producer) Retrieved from www.cs.princeton.edu/~funk/acoustics.html
- Future Music. (2011, January 13). *A brief history of Ableton Live*. Retrieved from Music Radar: <https://www.musicradar.com/tuition/tech/a-brief-history-of-ableton-live-357837>
- Gatheral, G. (2017, June 15). *Procedural Audio in the new Unreal Audio Engine - PANow - 18th May 2017*. Retrieved from YouTube: <https://youtu.be/auh342KUUoM>
- Gerzon, M. (1971). Whither Four Channels? *Audio Annual*, 36–41.
- Gerzon, M. (1974). Surround-Sound Psychoacoustics: Criteria for the Design of Matrix and Discrete Surround-Sound Systems. *Wireless World*.
- Gerzon, M. (1974). What's Wrong with Quadraphonies. *Studio Sound*.
- Gerzon, M. (1975). Ambisonics. Part Two: Studio Techniques. *Studio Sound*, 24–26, 28, 40.
- Gillies, M. (2018, June 24). *Setting up Unity for Oculus GO development*. Retrieved from Medium: <https://medium.com/virtual-reality-virtual-people/setting-up-unity-for-oculus-go-development-43d3e8c12f59>
- Goetz, C. (2015, November 30). *Stepping out of the Virtual World Paradigm*. (S. Media, Producer) Retrieved from Memory Insufficient: meminsf.silverstringmedia.com/art/stepping-out-of-the-virtual-world-paradigm/

- Goltz, F. (2018). Ableton Link – A technology to synchronize music software. In H. v. Coler, F. Neumann, & D. Runge (Ed.), *Proceedings of the Linux Audio Conference*. Berlin: TU Berlin.
- Google. (2016, August 24). *Cardboard*. Retrieved from Google: <https://vr.google.com/cardboard/manufacturers/>
- Google. (2016, August 5). *Use spatial audio in 360-degree and VR videos*. Retrieved from YouTube Help: <https://support.google.com/youtube/answer/6395969>
- Google. (2017, November 28). *Discover Resonance Audio | Resonance Audio | Google Developers*. (Google, Producer) Retrieved from developers.google.com/resonance-audio/discover/overview
- Google. (2018, November 6). *ARCore - Supported Devices*. Retrieved from Google ARCore Documentation: <https://developers.google.com/ar/discover/supported-devices>
- Google. (2018, June 18). *ARCore: Fundamental Concepts*. Retrieved from Google ARCore: https://developers.google.com/ar/discover/concepts#light_estimation
- Google. (2019, March 1). *Developing with Resonance Audio*. Retrieved from Google Resonance Audio: <https://resonance-audio.github.io/resonance-audio/develop/overview.html>
- Gosalia, A. (2018, February 23). *Announcing ARCore 1.0 and new updates to Google Lens*. Retrieved from Google: <https://www.blog.google/products/arcore/announcing-arcore-10-and-new-updates-google-lens/>
- Gould, G. (1966, April). The Prospects of Recording. *High Fidelity Magazine*, pp. 46-63.
- Grayson, J. (1970). New Materials and Methods for the Musical Instrument Designer, the Audio-Kenetic Sculpture, Musician and Composer. *Leonardo*, Vol. 3, 295-304.
- Grayson, N. (2014, December 19). *A Game That Creates Weird Alien Art Galleries*. (Kotaku, Producer) Retrieved from kotaku.com/a-game-that-creates-weird-alien-art-galleries-1673041917
- Grunthaner, J. (2015, November 11). *Tripping Through Digital Landscapes in Virtual Reality and on Canvas*. (Hyperallergic, Producer) Retrieved from hyperallergic.com/252786/tripping-through-digital-landscapes-in-virtual-reality-and-on-canvas/.
- Ha, A. (2016, September 1). *OmniVirt brings 360-degree advertising to big publishers*. Retrieved from Tech Crunch: <https://techcrunch.com/2016/08/01/omnivirt-nyt/>
- Harley, J. (1995). Generative Processes in Algorithmic Composition: Chaos and Music. *Leonardo*, 28(3), 221–224.

- Harvey, A., & Samyn, M. (2007, July 11). *Reatime Art Manifest*. (T. o. Tales, Producer)
Retrieved from tale-of-tales.com/tales/RAM.html
- Haus, G. (Autumn, 1983). EMPS: A System for Graphic Transcription of Electronic Music Scores. *Computer Music Journal*, 7(3), 31–36.
- Herrmutt Lobby. (2010, January 7). *Homepage*. Retrieved from www.herrmuttlobby.com
- Herrmutt Lobby. (2015, August 31). *Playground Homepage*. Retrieved from <http://www.getplayground.com/>
- Herrmutt Lobby. (2016, January 20). *Playground • Music at your fingertips*. Retrieved from YouTube: <https://www.youtube.com/watch?v=9MeuaehxIwA&feature=youtu.be>
- Hirway, H. (2015, December 15). *Episode 60: Björk 'Stonemilker'*. (R. P. Song Exploder, Producer) Retrieved from <http://songexploder.net/bjork>
- Hodges, P. (2011, December 12). *Ambisonic Info | Channel Formats*. (A. Info, Producer)
Retrieved from ambisonic.info/index.html
- Hodges, P. (2018, May 20). *Ambisonic Software*. Retrieved from Ambisonic.info:
<https://ambisonic.info/practical/software.html>
- Hoguet, B. (2014, October 28). *A Short History of Interactivity: The Revolution Was Coming for Time*. (Medium, Producer) Retrieved from medium.com/@benhoguet/a-short-history-of-interactivity-6fe72f7defea.
- Hollister, S. (2015, September 8). *YouTube's Ready To Blow Your Mind With 360-Degree Videos*. Retrieved from Gizmodo: gizmodo.com/youtubes-ready-to-blow-your-mind-with-360-degree-videos-1690989402
- Hollister, S. (2015, March 13). *YouTube's Ready To Blow Your Mind With 360-Degree Videos*. Retrieved from Gizmodo: <https://gizmodo.com/youtubes-ready-to-blow-your-mind-with-360-degree-videos-1690989402>
- Horwitz, J. (2018, May 2). *Oculus and Facebook detail Red volumetric VR camera with Surround 360*. Retrieved from Venture Beat:
<https://venturebeat.com/2018/05/02/oculus-and-facebook-detail-red-volumetric-vr-camera-with-surround360/>
- HTC. (2016). *HTC Vive PRE User Manual*. Retrieved from https://www.htc.com/managed-assets/shared/desktop/vive/vive_pre_user_guide.pdf
- HTC. (2017, February 26). *VIVE TRACKER GO BEYOND VR CONTROLLERS*. Retrieved from Vive: <https://www.vive.com/us/vive-tracker/>
- HTC. (2019, February 15). *Vive Cosmos*. Retrieved from HTC:
<https://www.vive.com/eu/cosmos/>

- Hu, C. (2018, November 29). *What's The Actual Business Case For Virtual Reality In Music?* Retrieved from Forbes: <https://www.forbes.com/sites/cheriehu/2018/11/29/whats-the-actual-business-case-for-virtual-reality-in-music/#48fece3b1f98>
- iExist. (2018, March 20). *Consciousness - The VR Concept Album*. Retrieved from Kickstarter: <https://www.kickstarter.com/projects/844845263/consciousness-the-vr-concept-album>
- Ingraham, N. (2017, March 8). *Vimeo now supports 360-degree video*. Retrieved from Engadget: <https://www.engadget.com/2017/03/08/vimeo-now-supports-360-degree-video/>
- Joyce, K. (2017, December 8). *PlayStation VR's Impressive Sales Boasts A Secure Future For VR*. Retrieved from VR Focus: www.vrfocus.com/2017/12/playstation-vrs-impressive-sales-boasts-a-secure-future-for-vr/
- Kastrenakes, J. (2017, September 25). *GPS will be accurate within one foot in some phones next year*. Retrieved from The Verge: <https://www.theverge.com/circuitbreaker/2017/9/25/16362296/gps-accuracy-improving-one-foot-broadcom>
- Katz, M. (2010). *Capturing Sound: How technology has changed music*. Berkeley: University of California Press.
- Kickstarter. (2016, January 5). *A Brief History of Oculus, from Day Zero to Day One*. Retrieved from Kickstarter: medium.com/kickstarter/a-brief-history-of-oculus-from-day-zero-to-day-one-8878aae002f8
- Kirn, P. (2017, June 6). *A conversation with David Zicarelli and Gerhard Behles*. Retrieved from Create Digital Music: <http://cdm.link/2017/06/conversation-david-zicarelli-gerhard-behles/>
- Kirn, P. (2018, April 25). *Free tools for Live unlock 3D spatial audio, VR, AR*. Retrieved from Create Digital Music: <http://cdm.link/2018/04/meet-the-free-tools-for-live-10-that-unlock-spatial-audio-vr-sound/>
- Kuchera, B. (2016, July 6). *Keep Virtual Reality Weird*. (Polygon, Producer) Retrieved from www.polygon.com/2016/7/6/11991296/keep-virtual-reality-weird
- Langer, M. (2018, April 14). *Computational Perception, Comp 546: Lecture Notes Week 16–23*. (M. University, Producer) Retrieved from <http://www.cim.mcgill.ca/~langer/546.html>
- Langley, H. (2017, May 3). *Inside-out v Outside-in: How VR tracking works, and how it's going to change*. Retrieved from Wearable: <https://www.wearable.com/vr/inside-out-vs-outside-in-vr-tracking-343>
- Leap Motion. (2015, August 29). *VR Design Best Practices—Where the physical and digital worlds collide, there be dragons*. Retrieved from Medium: <https://medium.com/@LeapMotion/vr-design-best-practices-bb889c2dc70>

- Lenhart, A. (2015, August 6). *Video Games Are Key Elements in Friendships for Many Boys*. Retrieved from Pew Research Center: <https://www.pewinternet.org/2015/08/06/chapter-3-video-games-are-key-elements-in-friendships-for-many-boys/>
- Lieberman, Z. (2015, January 22). *Homepage*. Retrieved from www.zach.li/
- Lieberman, Z. (2017, September 6). *Instagram*. Retrieved from www.instagram.com/p/BYs0iI3g8kZ/?hl=en
- Linden Lab. (2001, January 3). *Official Site | Second Life*. (L. Labs, Producer) Retrieved from secondlife.com/
- Linden Lab. (2007, February 6). *Linden Scripting Language Wiki Documentation*. Retrieved from http://wiki.secondlife.com/wiki/LSL_Portal
- Ling, R. (2015). The playful use of mobile phones and its link to social cohesion. In R. Ling, *Playful Identities: The Ludification of Digital Media Cultures* (pp. 181-197). Amsterdam: Amsterdam University Press.
- Lowood, H. (2014). Game Engines and Game History. *History of Games International Conference Proceedings* (pp. 179 - 198). Stanford Libraries, Palo Alto: Kinephanos. Retrieved from https://www.kinephanos.ca/Revue_files/2014-Lowood.pdf
- Magic Leap. (2018, September 8). *Magic Leap One*. Retrieved from <https://www.magicleap.com/magic-leap-one>
- Magnusson, T. (2010). Designing Constraints: Composing and Performing with Digital Musical. *Computer Music Journal*, 34(No. 4 (Winter 2010)), 62-73.
- Maher, S. (2015, November 23). *Introducing 360 Video on Facebook*. Retrieved from Facebook Newsroom: newsroom.fb.com/news/2015/09/introducing-360-video-on-facebook/.
- markNine. (2008, October 9). *Bloom - iPhone App - By Brian Eno*. Retrieved from YouTube: <https://www.youtube.com/watch?v=-swFqAT8yaA>
- Marshall, B. (2008, October 14). *Brian Eno's Bloom: new album or ambient joke?* Retrieved from <https://www.theguardian.com/music/musicblog/2008/oct/14/brian-eno-bloom-ipod-iphone>
- Martin, G. D. (2016, February 1). *My Experience as a Virtual War Photographer in Battlefield 1*. (K. Screen, Producer) Retrieved from killscreen.com/articles/corpography-battlefield-1-beta/
- Matney, L. (2016, June 24). *Facebook just bought VR audio company Two Big Ears and is making their tech free to developers*. Retrieved from Tech Crunch: <https://techcrunch.com/2016/05/23/facebook-just-bought-vr-audio-company-two-big-ears-and-is-making-their-tech-free-to-developers/>

- Matney, L. (2017, April 8). *Facebook debuts its first dedicated virtual reality app, Facebook 360*. Retrieved from Tech Crunch: <https://techcrunch.com/2017/03/08/facebook-debuts-its-first-dedicated-virtual-reality-app-facebook-360/>
- Matney, L. (2017, November 28). *Virtual Reality Headset Unit Sales Are Slowly Improving*. Retrieved from Tech Crunch: techcrunch.com/2017/11/28/virtual-reality-headset-unit-sales-are-slowly-improving/
- Maurer IV, J. A. (1999, June 4). *Interactive Electronic in Computer Music*. (C. f. Stanford, Producer) Retrieved from cirma.stanford.edu/~blackrse/interactive_electronics.html.
- McLeran, A. (2017, March 15). *The Future of Audio in Unreal Engine | GDC 2017 | Unreal Engine*. Retrieved from YouTube: <https://youtu.be/ErejaBCicds>
- McMahan, A. (2003). Immersion, Engagement, and Presence: A Method for Analyzing 3-D Video Games. *The Video Game Theory Reader*, 67–86.
- McNeill, E. (2014, September 22). *The Gender Gap in VR*. (Gamasutra, Producer) Retrieved from www.gamasutra.com/blogs/EMcNeill/20140922/226135/The_Gender_Gap_in_VR.php
- MelodyVR. (2015, June 18). *Homepage*. Retrieved from www.melodyvr.com
- Microsoft. (2017, May 12). *Hololens*. Retrieved from www.microsoft.com/en-us/hololens
- Milani, M. (2012, December 26). *Peter Chilvers: Visual and Tactical Music*. Retrieved from DigiCult: <http://digicult.it/digimag/issue-048/peter-chilvers-visual-and-tactical-music/>
- Minion, M. (2017, September 1). *Pygmalion's Spectacles: Using Berkeley's Immaterialism to Understand the Potential for Telepresence in Virtual Reality*. (M. o. Minion, Producer) Retrieved from musingsofamariominion.com/2017/09/01/pygmalions-spectacles-using-berkeleys-immaterialism-to-understand-the-potential-for-telepresence-in-virtual-reality
- Morales-Mazanares, R., & al., e. (2001). SICIB: An Interactive Music Composition System Using Body Movments. *Computer Music Journal*, 25(2), 25–36.
- Murphy, J., & al, e. (2012). Musical Robotic in a Loudspeaker World: Developments in Alternative Approaches to Localization and Spatialization. *Leonardo Music Journal*, 22, 41–48.
- Nechvatal, J. (2014, April 23). *The Lush Life of Virtual Architecture*. (Hyperallergic, Producer) Retrieved from hyperallergic.com/122097/the-lush-life-of-virtual-architecture/
- Nolo. (2018, December 19). *CVI*. Retrieved from Nolo: <https://www.nolovr.com>
- Nonesuch. (2012, Decemember 21). *Björk Unveils New Holiday Version of 'Solstice' App from 'Biophilia'*. Retrieved from Nonesuch: www.nonesuch.com/journal/bjork-unveils-new-holiday-version-solstice-app-biophilia-2011-12-21

- Oculus. (2016, January 26). *Oculus Audio SDK Guide*. Retrieved from Oculus Developers Documentation:
<https://developer.oculus.com/documentation/audiosdk/latest/concepts/book-audiosdk/>
- Oculus. (2014, December 21). *Developer Documentation*. Retrieved from Oculus:
<https://developer.oculus.com/documentation/>
- Oculus. (2015, January 21). *Oculus Rift: Step Into the Game*. Retrieved from
<https://www.kickstarter.com/projects/1523379957/oculus-rift-step-into-the-game/description>
- Oculus. (2016, February 10). *Introduction to Virtual Reality Audio*. (O. D. Center, Producer) Retrieved from developer.oculus.com/documentation/audiosdk/latest/concepts/book-audio-intro/
- Oculus. (2017, September 2017). *Beyond Surround Sound: Audio Advance in VR*. (Oculus, Producer) Retrieved from www.oculus.com/blog/beyond-surround-sound-audio-advances-in-vr/.
- Oculus. (2017, November 21). *Oculus Store: Dispatch*. Retrieved from
<https://www.oculus.com/experiences/go/1279074825522400/>
- Oculus. (2018, September 26). *Introducing Oculus Quest, Our First 6DOF All-in-One VR System, Launching Spring 2019*. Retrieved from Oculus:
<https://www.oculus.com/blog/introducing-oculus-quest-our-first-6dof-all-in-one-vr-system-launching-spring-2019/>
- Oculus. (2018, October 28). *Oculus Quest*. Retrieved from Oculus:
<https://www.oculus.com/quest/>
- Oculus. (2018, August 15). *Oculus Store: Netflix*. Retrieved from
<https://www.oculus.com/experiences/go/890562797701371/>
- Oliveros, P. (1995). Acoustic and Virtual Space as a Dynamic Element of Music. *Leonardo Music Journal*, 5, 19–22.
- Olszacki, P. (2017, November 28). *Getting a room plan with ARKit (The case where size matters)*. Retrieved from Medium: <https://medium.com/inborn-experience/getting-a-room-plan-with-arkit-the-case-where-size-matters-1e104ee42421>
- Omnivirt. (2017, February 19). *Interactive 360° Video*. Retrieved from Omnivirt:
<https://www.omnivirt.com/360-video-hotspots>
- Omnivirt. (2013, August 13). *360° VIRTUAL REALITY AD PLATFORM FOR ADVERTISERS AND PUBLISHERS*. Retrieved from Omnivirt: <https://www.omnivirt.com/>

- Owsinki, B. (2013, July 15). *5 Reasons Why Album Apps Will Fail*. Retrieved from Forbes: www.forbes.com/sites/bobbyowsinski/2013/07/15/5-reasons-why-album-apps-will-fail/#26929edb20c6
- Pagliery, J. (2015, September 23). *Facebook launches 360-degree videos - and they're amazing*. Retrieved from CNN Business: <https://money.cnn.com/2015/09/23/technology/facebook-360-video/index.html>
- Pareles, J. (2013, July 4). *Jay-Z Is Watching, and He Knows Your Friends*. Retrieved from www.nytimes.com/2013/07/05/arts/music/jay-z-is-watching-and-he-knows-your-friends.html
- Patel, N. (2018, July 19). *The US Patent Office has issued 10 million patents*. Retrieved from The Verge: <https://www.theverge.com/2018/6/19/17478898/uspto-utility-patents-10-million-issued>
- PICINIC. (2011, September 19). *Scott Snibbe on Biophilia, An App Album / PICINIC Festival 2011*. Retrieved from Vimeo: vimeo.com/29256409
- Pope, S. T. (1994). Editor's Notes: A Taxonomy of Computer Music. *Computer Music Journal*, 18(1), 5–7.
- Pratt, C. J. (2017, May 15). *In the Shadow of the Holodeck*. (Medium, Producer) Retrieved from medium.com/@charlesjpratt/in-the-shadow-of-the-holodeck-ae6bbb4ac195
- Reeps One. (2016, June 2019). *YouTube*. Retrieved from www.youtube.com/watch?v=OMLgliKYqal
- Rieger, M. (2018, November). *Spatial Audio Formats*. Retrieved from VRTONUNG: <https://www.vrtonung.de/en/spatial-audio-support-for-360-video-platform/>
- Roads, C. (2001). *Microsounds*. Boston: MIT Press.
- Roettgers, J. (2017, May 22). *Study Predicts Fewer Than 10 Million Monthly U.S. VR Headset Users This Year, 17 Million by 2019*. Retrieved from Variety: variety.com/2017/digital/news/vr-headset-data-mau-2017-2019-1202440211/
- Roginska, A., & Geluso, P. (2018). *Immersive Sound: The Art and Science of Binaural and Multi-Channel Audio*. New York, NY: Routledge.
- Roli. (2017, December 23). *Seaboard RISE 25*. Retrieved from Roli: <https://roli.com/products/seaboard/rise-25>
- Rubin, P. (2018). *Future Presence: How Virtual Reality is Changing Human Connection, Intimacy, and the Limits of Ordinary Life*. New York, NY: HarperCollins Publishers.

- Sag, A. (2019, January 4). *Location-Based VR: The Next Phase Of Immersive Entertainment*. Retrieved from Forbes: <https://www.forbes.com/sites/moorinsights/2019/01/04/location-based-vr-the-next-phase-of-immersive-entertainment/#273cdc923f57>
- Samsung. (2016, September 4). *Gear VR*. Retrieved from Samsung: <https://www.samsung.com/global/galaxy/gear-vr/>
- Schifrin, L. (2011). *Music COmposition for Film and Television*. New York, NY: Leonard Corporation.
- Schutze, S. (2018). *New Realities in Audio: A Practical Guide for VR, AR, MR, and 360 Video*. New York, NY: Taylor & Francis.
- Seegert, A. (2009). 'Doing There' vs. 'Being There'" Performing Presence in Interactive Fiction. *Journal of Gaming and Virtual worlds*, 1(1), 23–37.
- Sherbourne, S. (2017, October 18). *Ambisonics and VR/360 Audio in Pro Tools | HD*. Retrieved from Avid Blogs: <http://www.avidblogs.com/ambisonics-vr360-audio-pro-tools-hd/>
- Sherburne, P. (2017, February 16). *A Conversation With Brian Eno About Ambient Music*. (Pitchfork, Producer) Retrieved from <https://pitchfork.com/features/interview/10023-a-conversation-with-brian-eno-about-ambient-music/>
- Sicat, M. (2017, July 31). *Queering the COntroller*. (A. G. Studies, Producer) Retrieved from analoggamestudies.org/2017/07/queering-the-controller/
- Singleton, M. (2017, June 29). *Jay Z Has USed His Latest Albums as Corporate Bargaining Chips*. Retrieved from The Verge: www.theverge.com/2017/6/29/15892648/jay-z-albums-corporate-bargaining-chips-444-sprint-tidal
- Smule. (2001, April 2). *Homepage*. Retrieved from <https://www.smule.com>
- Smule. (2010, April 23). *Smule iTunes Page*. Retrieved from <https://itunes.apple.com/us/app/ocarina/id293053479?mt=8&app=itunes>
- Soper, T. (2017, September 21). *How Oculus Audio Engineers Are Using New Sound Technology to Enhance Virtual Reality Experiences*. (Geekwire, Producer) Retrieved from www.geekwire.com/2017/oculus-audio-engineers-using-new-sound-technology-enhance-vr-experiences/.
- Starovoitov, V., & Samal, D. (1999). A geometric approach to face recognition. *Proceedings of the IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP'99)*. Antalya: IEEE.
- Steam Audio. (2017, March 2). *Steam Audio :: Introducing Steam Audio*. (S. Community, Producer) Retrieved from steamcommunity.com/games/596420/announcements/detail/521693426582988261

- Sterne, J. (2012). *MP3: The Meaning of a Format*. Durham, NC: Duke University Press.
- Sweet, M. (2015). *Writing Interactive Music for Video Games: A Composer's Guide*. New York, NY: Addison-Wesley Professional.
- Taylor, D. G. (2016, April 10). *Virtual reality technologies that failed hard*. Retrieved from Grunge: <https://www.grunge.com/11190/virtual-reality-technologies-that-failed-hard/>
- The Atlantic Media Company. (2015, September 9). *How Fast Is Technology Accelerating?* Retrieved from The Atlantic: www.theatlantic.com/sponsored/prudential-great-expectations/how-fast-is-technology-accelerating/360/
- The Game of Life. (2011, July 3). *About Wave Field Synthesis*. (T. G. Life, Producer) Retrieved from gameoflife.nl/en/about/about-wave-field-synthesis/.
- The Holladay Brothers. (2018, January 12). *Homepage*. Retrieved from <http://www.theholladaybrothers.com/>
- The MIDI Association. (2019, April 22). *MIDI Polyphonic Expression (MPE) Specification Adopted!* Retrieved from MIDI Association: <https://www.midi.org/articles-old/midi-polyphonic-expression-mpe>
- The MIDI Association. (2019, February 18). *The MIDI Manufacturers Association (MMA) and the Association of Music Electronics Industry (AMEI) announce MIDI 2.0™ Prototyping*. Retrieved from MIDI Association: <https://www.midi.org/articles-old/the-midi-manufacturers-association-mma-and-the-association-of-music-electronics-industry-amei-announce-midi-2-0tm-prototyping>
- Thinkwik. (2018, April 20). *CryEngine vs Unreal vs Unity: Select the Best Game Engine*. Retrieved from Medium: <https://medium.com/@thinkwik/cryengine-vs-unreal-vs-unity-select-the-best-game-engine-eaca64c60e3e>
- TNW. (2016, March 24). *This engine is dominating the gaming industry right now*. Retrieved from The Next Web: <https://thenextweb.com/gaming/2016/03/24/engine-dominating-gaming-industry-right-now/>
- Turi, J. (2014, February 16). *The Sights and Scents of the Sensorama Simulator*. (Engaget, Producer) Retrieved from www.engadget.com/2014/02/16/morton-heiligs-sensorama-simulator/
- Unity. (2014, July 1). *Microphone*. Retrieved from Unity Developer Documentation: <https://docs.unity3d.com/Manual/class-Microphone.html>
- Unity. (2014, July 1). *Reverb Zones*. Retrieved from Unity Developer Documentation: <https://docs.unity3d.com/Manual/class-AudioReverbZone.html>
- Unity. (2014, July 1). *Scripting Overview*. Retrieved from Unity Developer Documentation: <https://docs.unity3d.com/Manual/ScriptingConcepts.html>

- Unity. (2014, July 28). *Unity Growth Facts*. Retrieved from Unity Company Facts: <https://unity3d.com/public-relations>
- Unity. (2018, September 17). *Real-Time Filmmaking, Explained*. Retrieved from Unity Solutions: <https://unity.com/solutions/film/real-time-filmmaking-explained>
- Unity. (2018, December 1). *Unity's evolving best practices*. Retrieved from Unity - Creator's Corner: <https://unity3d.com/how-to/unity-best-practices-for-engine-performance>
- Unity. (2019). *Unity Online Tutorials*. Retrieved from Unity: <https://unity.com/learn>
- Unreal. (2018, November 10). *Audio System Overview*. Retrieved from Unreal Engine Developer Documentation: <https://docs.unrealengine.com/en-us/Engine/Audio/Overview>
- Unreal. (2018, December 1). *Audio Volume Overview*. Retrieved from Unreal Engine Developer Documentation: <https://docs.unrealengine.com/en-US/Engine/Actors/Volumes/AudioVolumeDetails>
- Unreal Engine. (2018, October 28). *Blueprints Overview*. Retrieved from Unreal Engine Documentation: <https://docs.unrealengine.com/en-US/Engine/Blueprints/Overview>
- Valve. (2017, March 23). *A BENCHMARK IN IMMERSIVE AUDIO SOLUTIONS FOR GAMES AND VR*. Retrieved from Steam Audio: <https://valvesoftware.github.io/steam-audio/>
- Vasquez, J. C., Tahiroglu, K., & Kildal, J. (2017). Idiomatic Composition Practices for New Musical Instruments: Context, Background and Current Applications. *NIME* (pp. 174-179). Denmark: Aalborg University Copenhagen.
- Vieyra Software. (2018, September 24). *Vieyra Software*. Retrieved from Medium: <https://medium.com/@vieyrasoftware/comparing-google-arcore-and-apple-arkit-81b4727132ad>
- Virtual Reality Society. (2015, October 16). *History of Virtual Reality*. Retrieved from Virtual Reality Society: www.vrs.org.uk/virtual-reality/history.html
- Wanderlvst. (2017, July 9). *[Block Rocking Beats] EP1: Visual Music Production Tool in Virtual Reality*. Retrieved from youtu.be/8aXi-yV5-oU
- Waves Audio Ltd. (2013, February 21). *Michael Gerzon - A true Audio Visionary*. (Acoustics.net, Producer) Retrieved from archive.today/2013.02.21-134930/http://www.acoustics.net/content.asp?id=46
- Waves Audio Ltd. (2017, October 10). *Ambisonics Explained: A Guide for Sound Engineers*. (W. A. Ltd., Producer) Retrieved from www.oculus.com/blog/beyond-surround-sound-audio-advances-in-vr/

- White, B. (2018, September 4). *iOS vs Android Audio Apps*. Retrieved from Future Sonic:
<https://futuresonic.io/blog/ios-vs-android-in-music-apps/>
- Wingfield, N., & Isaac, M. (2016, July 11). *Pokémon Go Brings Augmented Reality to a Mass Audience*. Retrieved from The New York Times:
<https://www.nytimes.com/2016/07/12/technology/pokemon-go-brings-augmented-reality-to-a-mass-audience.html>
- Yang, R. (2017, April 5). *If You Walk in Someone Else's Shoes, Then You've Taken Their Shoes: Empathy Machines as Appropriation Machines*. (R. -D. Blog, Producer) Retrieved from
www.blog.radiator.debacl.us/2017/04/if-you-walk-in-someone-elses-shoes-then.html
- Zhu, A. Z., Atanasov, N., & Daniilidis, K. (2017). Event-based Visual Inertial Odometry. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu: IEEE.

Bibliography

- 6D.AI. (2018, November 27). *6D.AI Developer Documentation*. Retrieved from 6D.AI:
<https://dashboard.6d.ai/user/dashboard/?view=home>
- Anderson, E. (2017, May 6). *Narrative Isn't the End, but a Means to Reach It*. (N. C. Gaming, Producer) Retrieved from nocoastgaming.com/2017/05/06/narrative-isnt-the-end-but-a-means-to-reach-it/
- Apple. (2017, October 27). *About Face ID advanced technology*. Retrieved from Apple Support: <https://support.apple.com/en-us/HT208108>
- Apple. (2017, July 21). *Introducing ARKit: Augmented Reality for iOS*. Retrieved from Apple Developer: <https://developer.apple.com/videos/play/wwdc2017/602/>
- Apple. (2018, June 7). *Apple Unveils ARKit 2*. Retrieved from Apple: www.apple.com/newsroom/2018/06/apple-unveils-arkit-2/
- Apple. (2018, March 3). *Augmented Reality for iOS*. Retrieved from Apple: <https://www.apple.com/ios/augmented-reality/>
- Apple. (2018, July 20). *Scanning and Detecting 3D Object*. Retrieved from Apple Developer Documentation: https://developer.apple.com/documentation/arkit/scanning_and_detecting_3d_objects
- Archer, D., & Finger, K. (2018, March 15). *Walking in Another's Virtual Shoes: Do 360-Degree Video News Stories Generate Empathy in Viewers?* (C. J. Review, Producer) Retrieved from www.cjr.org/tow_center_reports/virtual-reality-news-empathy.php/
- Atlantic Re:Think. (2015, September 9). *How Fast Is Technology Accelerating?* (T. Atlantic, Producer) Retrieved from www.theatlantic.com/sponsored/prudential-great-expectations/how-fast-is-technology-accelerating/360/
- Audiokenetic. (2017, July 27). *Wwise*. Retrieved from Audiokenetic - Wwise: <https://www.audiokinetic.com/products/wwise/>
- Audiokenetic. (2019). *Wwise Spatial Audio: Create realistic or out of this world dynamic immersive experiences!* Retrieved from Audiokenetic: <https://www.audiokinetic.com/products/wwise-spatial-audio/>
- Audiokinetic. (2017, April 18). *Wwise Certification*. Retrieved from Audiokinetic: <https://www.audiokinetic.com/learn/certifications/>

- Avatar Orchestra Metaverse. (2003, March 23). *Homepage*. Retrieved from <http://avatarorchestra.blogspot.com/>
- Baalman, M. A. (2010). Spatial Composition Techniques and Sound Spatialisation Technologies. *Organised Sound*, 15(03), 209–218.
- Bailenson, J. (2018). *Experience on Demand: What Virtual Reality Is, How It Works, and What It Can Do*. New York, NY: W. W. Norton & Company.
- Bartle, R. A. (2016). *MMOs from the Insdie Out: The History, Design, Fun, and Art of Massively-Multiplayer Online Role-Playing Games*. New York, NY: Apress, Springer Science+Buiness Media.
- Batti, B., & Karabinus, A. (2017, May 1). *A Dream of Embodied Experience: On Ian Bogost, Epistemological Gatekeeping, and the Holodeck*. Retrieved from NY Gamer: www.nymgamer.com/?p=16363
- Belamire, J. (2016, October 21). *My First Virtual Reality Groping*. (Mic, Producer) Retrieved from mic.com/articles/157415/my-first-virtual-reality-groping-sexual-assault-in-vr-harassment-in-tech-jordan-belamire#.p6gg5NrLq
- Bizri, H. (2000). Story Telling in Virutal Reality. *Leonardo*, 33(1), 17–19.
- Björk. (2011, July 19). *Björk Biophilia App intro Narrated by David Attenborough*. Retrieved from <https://youtu.be/o8AELvVUFLw>
- Björk, & Huang, A. T. (2015, June 6). *Björk: Stonemilker (360 Dree Virtual Reality)*. Retrieved from YouTube: www.youtube.com/watch?time_continue=72&v=gQEyezu7G20
- Blue Ripple Sound. (2011, May 26). *HOA Technical Notes - Introduction to Higher Order Ambisonics*. Retrieved from www.blueripplesound.com/hoa-introduction
- Bluebrain. (2018, June 27). *Site: Nine Elms*. Retrieved from iTunes App Store: <https://itunes.apple.com/us/app/site-nine-elms/id1088941378>
- Blum, A. (2018, March 19). *ARCore vs. ARKit: Which Is Better for Building Augmented Reality Apps?* Retrieved from Iflexion: <https://www.iflexion.com/blog/arcore-vs-arkit-better-building-augmented-reality-apps/>
- Bobkova, J. (2017, June 21). *The Potential of Virtual Reality in Healthcare*. Retrieved from Medium: [The Potential of Virtual Reality in Healthcare](https://medium.com/@jbobkova/the-potential-of-virtual-reality-in-healthcare)
- Bogost, I. (2017, April 25). *Video Games Are Better Without Stories*. (T. Atlantic, Producer) Retrieved from www.theatlantic.com/technology/archive/2017/04/video-games-stories/524148/?utm_source=atltw
- Bohn, D. A. (2016, December 20). *Pro Audio Reference (PAR)*. (A. E. Society, Producer) Retrieved from www.aes.org/par/

- Bonnington, C. (2015, March 13). *YOU CAN NOW WATCH AND UPLOAD 360-DEGREE VIDEOS ON YOUTUBE*. Retrieved from Wired:
<https://www.wired.com/2015/03/youtube-360-degree-video/>
- Brackeys. (2018, May 20). *How to Fade Between Scenes in Unity*. Retrieved from YouTube:
<https://youtu.be/Oadq-IrOazg>
- Brogan, J. (2017, September 1). *Our Virtual Future: What Philosophers Say About the Inevitability of VR*. (Slate, Producer) Retrieved from slate.com/technology/2017/09/what-philosophers-say-about-the-inevitability-of-virtual-reality.html
- Bucher, J. K. (2018). *Storytelling for Virtual Reality: Methods and Principles for Crafting Immersive Narratives*. New York, NY: Routledge.
- Buckley, S. (2015, August 19). *This Is How Valve's Amazing Lighthouse Tracking Technology Works*. Retrieved from Gizmodo: <https://gizmodo.com/this-is-how-valve-s-amazing-lighthouse-tracking-technol-1705356768>
- Buzzi, M. (2018, December 7). *The Best Laptops for VR in 2019*. Retrieved from PC Magazine:
<https://www.pcmag.com/roundup/360989/the-best-laptops-for-vr>
- Cabrera, A. (2014, February 27). *Ambisonics*. (Ambisonics, Producer) Retrieved from w2.mat.ucsb.edu/240/D/notes/Ambisonics.html
- CDM Blogs. (2008, October 8). *Hands-on with Bloom, New Generative iPhone App by Eno and Chilvers*. Retrieved from Create Digital Music: Hands-on with Bloom, New Generative iPhone App by Eno and Chilvers
- Cheng, W. (2014). *Sound Play: Video Games and the Musical Imagination*. New York, NY: Oxford University Press.
- Christiansen, P. (2018, November 15). *Presence, Preservation, and Virtual Reality*. (Playthepast, Producer) Retrieved from www.playthepast.org/?p=6181
- Cockos. (2009, June 28). *JSFX Programming*. Retrieved from Reaper:
<http://reaper.fm/sdk/js/js.php>
- Code, B. (2016, November 7). *Video Games are Boring*. (G. Industry, Producer) Retrieved from www.gamesindustry.biz/articles/2016-11-07-video-games-are-boring
- Colucci, R., & al., e. (2015). Nonlinear Music Analysis and Composition. *European Society for Mathematics and the Arts*, 87–94.
- Conservation International. (2008, January 13). *Homepage*. Retrieved from <https://www.conservation.org>

- Constine, J. (2014, July 22). *Facebook's \$2 Billion Acquisition of Oculus Closes, Now Official*. (T. Crunch, Producer) Retrieved from techcrunch.com/2014/07/21/facebooks-acquisition-of-oculus-closes-now-official/
- Constine, J. (2018, October 27). *Oculus VR Content Will Be Forward-Compatible with ITs Future Headsets*. (T. Crunch, Producer) Retrieved from techcrunch.com/2018/09/26/oculus-compatibility/
- Cox, C., & Warner, D. (2017). *Audio Culture: Readings in Modern Music*. New York, NY: Bloomsbury Academic.
- D'Anastasio, C. (2016, July 6). *VR Has Its Roots In Ancient Rome*. (Kotaku, Producer) Retrieved from kotaku.com/vr-has-its-roots-in-ancient-rome-1783199702
- Deahl, D. (2018, March 27). *You Can Now Control GarageBand with Your Face on iPhone X*. (T. Verge, Producer) Retrieved from www.theverge.com/2018/3/27/17169984/control-garageband-face-arkit-iphone-x
- Dean, R. T., & McLean, A. (2018). *The Oxford Handbook of Algorithmic Music*. New York, NY: Oxford University Press.
- Defunct Games. (2015, September 10). *Defunct Games*. Retrieved from YouTube: <https://youtu.be/Ta2L3Z7OMng>
- Deia, Z. (2018, May 23). *Understanding Surround Sound Formats*. (Crutchfield, Producer) Retrieved from www.crutchfield.com/S-e38JxrWlcR1/learn/learningcenter/home/hometheater_surround.html
- DeLahunta, S. (2022). Virtual Reality and Performance. *PAJ: A Journal of Performance and Art*, 24(1), 105–114.
- Dolby. (2016, March 1). *Dolby Atmos for Virtual Reality*. Retrieved from Dolby: <https://www.dolby.com/us/en/professional/content-creation/vr.html>
- Dombal, R. (2011, June 29). *Björk*. Retrieved from Pitchfork: pitchfork.com/features/interview/7996-bjork/.
- Downey, S. (2014). History of the (Virtual) Worlds. *The Journal of Technology Studies*, 40(1/2), 54–66.
- Dredge, S. (2012, September 26). *Brian Eno and Peter Chilvers talk Scape, iPad apps and generative music*. Retrieved from The Guardian: [https://www.theguardian.com/music/appsblog/2012/sep/26/brian-eno-scape-ipad-apps](http://www.theguardian.com/music/appsblog/2012/sep/26/brian-eno-scape-ipad-apps)
- Durbin, J. (2017, January 27). *Oculus Sensors Are Technically Hackable Webcams*. Retrieved from UploadVR: <https://uploadvr.com/hackable-webcam-oculus-sensor-be-aware/>
- EccoVR. (2016, April 17). *Homepage*. Retrieved from <http://www.eccovr.com/>

- Edge, E. (2016, June 28). *How VR Gaming Will Wake US Up to Our Fake Worlds*. (I. f. Technologies, Producer) Retrieved from ieet.org/index.php/IEET2/more/Edge20160628
- Ellis, C. (2018, February 12). *VLC Media Player now supports 360-degree video and resolutions up to 8K*. Retrieved from Tech Radar: <https://www.techradar.com/news/vlc-media-player-now-supports-360-degree-video-and-resolutions-up-to-8k>
- EMPAC. (2018, March 15). *Wave Field Synthesis System*. (E. M. (EMPAC), Producer) Retrieved from empac.rpi.edu/research/wave-field-synthesis
- Eno, A. (2010, November 23). *Cross-platform MIDI processing library for mono and .NET (ALSA, CoreMIDI, Android, WinMM and UWP)*. Retrieved from Github: <https://github.com/atsushieno/managed-midi>
- Eno, B. (2018, November 28). *Brian Eno sets out to change music (again) with Bloom: 10 World*. Retrieved from Digital Trends: <https://www.digitaltrends.com/music/brian-eno-peter-chilvers-bloom-10-worlds/>
- Eno, B., & Chilvers, P. (2018). *Bloom: 10 Worlds*. Retrieved from Generative Music.com: <http://generativemusic.com/bloom10worlds.html>
- Facebook. (2016, March 13). *Facebook 360*. Retrieved from Facebook 360: <https://facebook360.fb.com/>
- Facebook. (2018, May 1). *F8 2018: Oculus Go Available Now for \$199*. Retrieved from Facebook New Room: <https://newsroom.fb.com/news/2018/05/oculus-go/>
- Farmer, C. (2017, November 13). *Apaix World*. Retrieved from <https://apaix.world/>
- Farmer, C. (2018, November 20). *Making your face the effects automator with an iPhoneX, Unity, Max and Ableton*. Retrieved from Chase Farmer: <http://chasefarmer.com/articles/2018-11-20-making-your-face-the-fx-automator/>
- Fellget, P. (1975). Ambisonics. Part One: General System Description. *Studio Sound*, 17, 20–22.
- Feltham, J. (2018, July 12). *Why Virtual-Virtual Reality Remain A VR Experience You Must Play*. (U. VR, Producer) Retrieved from uploadvr.com/why-virtual-virtual-reality-remains-a-vr-experience-you-must-play/
- Firelight Technologies. (2017, March 28). *FMOD Studio*. Retrieved from FMOD: <https://www.fmod.com/studio>
- Firelight Technologies. (2017, March 18). *Licensing*. Retrieved from FMOD: <https://www.fmod.com/licensing>
- Firelight Technologies. (2018, August 22). *Firelight Technologies FMOD Studio API*. Retrieved from FMOD:

https://www.fmod.com/docs/api/content/generated/common/lowlevel_introduction.html#file-formats-1

- Firelight Technologies. (2018, July 26). *FMOD Studio User Manual*. Retrieved from FMOD: <https://www.fmod.com/resources/documentation-studio?page=welcome-to-fmod-studio.html>
- Flood, K. (2012, February 27). *The Language of Biophilia: A Q&A with Nikki Dibben*. Retrieved from Vice: www.vice.com/en_us/article/4x48ng/the-language-of-ibiophilii-a-qa-with-nikki-dibben
- FMOD. (2018, August 15). *FMOD Studio Plug-in SDK*. Retrieved from FMOD: https://www.fmod.com/resources/documentation-api?page=content/generated/overview/plugin_api_dsp.html#/
- Frankenfield, J. (2018, December 13). *Second Life Economy*. (Investopedia, Producer) Retrieved from www.investopedia.com/terms/s/second-life-economy.asp
- Fredrick, T. (2018, July 11). *Open Sound Control (OSC) for Unity*. Retrieved from Github: <https://github.com/thomasfredericks/UnityOSC>
- Funkhouser, T. A., Tsingos, N., Carlbom, I., Elko, G., Sondhi, M., & West, J. (2005, June 27). *Interactive Acoustic Modeling of Complex Environments*. (P. University, Producer) Retrieved from www.cs.princeton.edu/~funk/acoustics.html
- Future Music. (2011, January 13). *A brief history of Ableton Live*. Retrieved from Music Radar: <https://www.musicradar.com/tuition/tech/a-brief-history-of-ableton-live-357837>
- Gatheral, G. (2017, June 15). *Procedural Audio in the new Unreal Audio Engine - PANow - 18th May 2017*. Retrieved from YouTube: <https://youtu.be/auh342KUUoM>
- Gerzon, M. (1971). Whither Four Channels? *Audio Annual*, 36–41.
- Gerzon, M. (1974). Surround-Sound Psychoacoustics: Criteria for the Design of Matrix and Discrete Surround-Sound Systems. *Wireless World*.
- Gerzon, M. (1974). What's Wrong with Quadraphonies. *Studio Sound*.
- Gerzon, M. (1975). Ambisonics. Part Two: Studio Techniques. *Studio Sound*, 24–26, 28, 40.
- Gillies, M. (2018, June 24). *Setting up Unity for Oculus GO development*. Retrieved from Medium: <https://medium.com/virtual-reality-virtual-people/setting-up-unity-for-oculus-go-development-43d3e8c12f59>
- Goetz, C. (2015, November 30). *Stepping out of the Virtual World Paradigm*. (S. Media, Producer) Retrieved from Memory Insufficient: meminsf.silverstringmedia.com/art/stepping-out-of-the-virtual-world-paradigm/

- Goltz, F. (2018). Ableton Link – A technology to synchronize music software. In H. v. Coler, F. Neumann, & D. Runge (Ed.), *Proceedings of the Linux Audio Conference*. Berlin: TU Berlin.
- Google. (2016, August 24). *Cardboard*. Retrieved from Google: <https://vr.google.com/cardboard/manufacturers/>
- Google. (2016, August 5). *Use spatial audio in 360-degree and VR videos*. Retrieved from YouTube Help: <https://support.google.com/youtube/answer/6395969>
- Google. (2017, November 28). *Discover Resonance Audio | Resonance Audio | Google Developers*. (Google, Producer) Retrieved from developers.google.com/resonance-audio/discover/overview
- Google. (2018, November 6). *ARCore - Supported Devices*. Retrieved from Google ARCore Documentation: <https://developers.google.com/ar/discover/supported-devices>
- Google. (2018, June 18). *ARCore: Fundamental Concepts*. Retrieved from Google ARCore: https://developers.google.com/ar/discover/concepts#light_estimation
- Google. (2019, March 1). *Developing with Resonance Audio*. Retrieved from Google Resonance Audio: <https://resonance-audio.github.io/resonance-audio/develop/overview.html>
- Gosalia, A. (2018, February 23). *Announcing ARCore 1.0 and new updates to Google Lens*. Retrieved from Google: <https://www.blog.google/products/arcore/announcing-arcore-10-and-new-updates-google-lens/>
- Gould, G. (1966, April). The Prospects of Recording. *High Fidelity Magazine*, pp. 46-63.
- Grayson, J. (1970). New Materials and Methods for the Musical Instrument Designer, the Audio-Kenetic Sculpture, Musician and Composer. *Leonardo*, Vol. 3, 295-304.
- Grayson, N. (2014, December 19). *A Game That Creates Weird Alien Art Galleries*. (Kotaku, Producer) Retrieved from kotaku.com/a-game-that-creates-weird-alien-art-galleries-1673041917
- Grunthaner, J. (2015, November 11). *Tripping Through Digital Landscapes in Virtual Reality and on Canvas*. (Hyperallergic, Producer) Retrieved from hyperallergic.com/252786/tripping-through-digital-landscapes-in-virtual-reality-and-on-canvas/.
- Ha, A. (2016, September 1). *OmniVirt brings 360-degree advertising to big publishers*. Retrieved from Tech Crunch: <https://techcrunch.com/2016/08/01/omnivirt-nyt/>
- Harley, J. (1995). Generative Processes in Algorithmic Composition: Chaos and Music. *Leonardo*, 28(3), 221–224.

- Harvey, A., & Samyn, M. (2007, July 11). *Reatime Art Manifest*. (T. o. Tales, Producer)
Retrieved from tale-of-tales.com/tales/RAM.html
- Haus, G. (Autumn, 1983). EMPS: A System for Graphic Transcription of Electronic Music Scores. *Computer Music Journal*, 7(3), 31–36.
- Herrmutt Lobby. (2010, January 7). *Homepage*. Retrieved from www.herrmuttlobby.com
- Herrmutt Lobby. (2015, August 31). *Playground Homepage*. Retrieved from <http://www.getplayground.com/>
- Herrmutt Lobby. (2016, January 20). *Playground • Music at your fingertips*. Retrieved from YouTube: <https://www.youtube.com/watch?v=9MeuaehxIwA&feature=youtu.be>
- Hirway, H. (2015, December 15). *Episode 60: Björk 'Stonemilker'*. (R. P. Song Exploder, Producer) Retrieved from <http://songexploder.net/bjork>
- Hodges, P. (2011, December 12). *Ambisonic Info | Channel Formats*. (A. Info, Producer)
Retrieved from ambisonic.info/index.html
- Hodges, P. (2018, May 20). *Ambisonic Software*. Retrieved from Ambisonic.info:
<https://ambisonic.info/practical/software.html>
- Hoguet, B. (2014, October 28). *A Short History of Interactivity: The Revolution Was Coming for Time*. (Medium, Producer) Retrieved from medium.com/@benhoguet/a-short-history-of-interactivity-6fe72f7defea.
- Hollister, S. (2015, September 8). *YouTube's Ready To Blow Your Mind With 360-Degree Videos*. Retrieved from Gizmodo: gizmodo.com/youtubes-ready-to-blow-your-mind-with-360-degree-videos-1690989402
- Hollister, S. (2015, March 13). *YouTube's Ready To Blow Your Mind With 360-Degree Videos*. Retrieved from Gizmodo: <https://gizmodo.com/youtubes-ready-to-blow-your-mind-with-360-degree-videos-1690989402>
- Horwitz, J. (2018, May 2). *Oculus and Facebook detail Red volumetric VR camera with Surround 360*. Retrieved from Venture Beat:
<https://venturebeat.com/2018/05/02/oculus-and-facebook-detail-red-volumetric-vr-camera-with-surround360/>
- HTC. (2016). *HTC Vive PRE User Manual*. Retrieved from https://www.htc.com/managed-assets/shared/desktop/vive/vive_pre_user_guide.pdf
- HTC. (2017, February 26). *VIVE TRACKER GO BEYOND VR CONTROLLERS*. Retrieved from Vive: <https://www.vive.com/us/vive-tracker/>
- HTC. (2019, February 15). *Vive Cosmos*. Retrieved from HTC:
<https://www.vive.com/eu/cosmos/>

- Hu, C. (2018, November 29). *What's The Actual Business Case For Virtual Reality In Music?* Retrieved from Forbes: <https://www.forbes.com/sites/cheriehu/2018/11/29/whats-the-actual-business-case-for-virtual-reality-in-music/#48fece3b1f98>
- iExist. (2018, March 20). *Consciousness - The VR Concept Album*. Retrieved from Kickstarter: <https://www.kickstarter.com/projects/844845263/consciousness-the-vr-concept-album>
- Ingraham, N. (2017, March 8). *Vimeo now supports 360-degree video*. Retrieved from Engadget: <https://www.engadget.com/2017/03/08/vimeo-now-supports-360-degree-video/>
- Joyce, K. (2017, December 8). *PlayStation VR's Impressive Sales Boasts A Secure Future For VR*. Retrieved from VR Focus: www.vrfocus.com/2017/12/playstation-vrs-impressive-sales-boasts-a-secure-future-for-vr/
- Kastrenakes, J. (2017, September 25). *GPS will be accurate within one foot in some phones next year*. Retrieved from The Verge: <https://www.theverge.com/circuitbreaker/2017/9/25/16362296/gps-accuracy-improving-one-foot-broadcom>
- Katz, M. (2010). *Capturing Sound: How technology has changed music*. Berkeley: University of California Press.
- Kickstarter. (2016, January 5). *A Brief History of Oculus, from Day Zero to Day One*. Retrieved from Kickstarter: medium.com/kickstarter/a-brief-history-of-oculus-from-day-zero-to-day-one-8878aae002f8
- Kirn, P. (2017, June 6). *A conversation with David Zicarelli and Gerhard Behles*. Retrieved from Create Digital Music: <http://cdm.link/2017/06/conversation-david-zicarelli-gerhard-behles/>
- Kirn, P. (2018, April 25). *Free tools for Live unlock 3D spatial audio, VR, AR*. Retrieved from Create Digital Music: <http://cdm.link/2018/04/meet-the-free-tools-for-live-10-that-unlock-spatial-audio-vr-sound/>
- Kuchera, B. (2016, July 6). *Keep Virtual Reality Weird*. (Polygon, Producer) Retrieved from www.polygon.com/2016/7/6/11991296/keep-virtual-reality-weird
- Langer, M. (2018, April 14). *Computational Perception, Comp 546: Lecture Notes Week 16–23*. (M. University, Producer) Retrieved from <http://www.cim.mcgill.ca/~langer/546.html>
- Langley, H. (2017, May 3). *Inside-out v Outside-in: How VR tracking works, and how it's going to change*. Retrieved from Wearable: <https://www.wearable.com/vr/inside-out-vs-outside-in-vr-tracking-343>
- Leap Motion. (2015, August 29). *VR Design Best Practices—Where the physical and digital worlds collide, there be dragons*. Retrieved from Medium: <https://medium.com/@LeapMotion/vr-design-best-practices-bb889c2dc70>

- Lenhart, A. (2015, August 6). *Video Games Are Key Elements in Friendships for Many Boys*. Retrieved from Pew Research Center: <https://www.pewinternet.org/2015/08/06/chapter-3-video-games-are-key-elements-in-friendships-for-many-boys/>
- Lieberman, Z. (2015, January 22). *Homepage*. Retrieved from www.zach.li/
- Lieberman, Z. (2017, September 6). *Instagram*. Retrieved from www.instagram.com/p/BYs0iI3g8kZ/?hl=en
- Linden Lab. (2001, January 3). *Official Site | Second Life*. (L. Labs, Producer) Retrieved from secondlife.com/
- Linden Lab. (2007, February 6). *Linden Scripting Language Wiki Documentation*. Retrieved from http://wiki.secondlife.com/wiki/LSL_Portal
- Linden Labs. (2001, January 3). *Official Site | Second Life*. (L. Labs, Producer) Retrieved from secondlife.com/
- Ling, R. (2015). The playful use of mobile phones and its link to social cohesion. In R. Ling, *Playful Identities: The Ludification of Digital Media Cultures* (pp. 181-197). Amsterdam: Amsterdam University Press.
- Lowood, H. (2014). Game Engines and Game History. *History of Games International Conference Proceedings* (pp. 179 - 198). Stanford Libraries, Palo Alto: Kinephanos. Retrieved from https://www.kinephanos.ca/Revue_files/2014-Lowood.pdf
- Magic Leap. (2018, September 8). *Magic Leap One*. Retrieved from <https://www.magicleap.com/magic-leap-one>
- Magnusson, T. (2010). Designing Constraints: Composing and Performing with Digital Musical. *Computer Music Journal*, 34(No. 4 (Winter 2010)), 62-73.
- Maher, S. (2015, November 23). *Introducing 360 Video on Facebook*. Retrieved from Facebook Newsroom: newsroom.fb.com/news/2015/09/introducing-360-video-on-facebook/.
- markNine. (2008, October 9). *Bloom - iPhone App - By Brian Eno*. Retrieved from YouTube: <https://www.youtube.com/watch?v=-swFqAT8yaA>
- Marshall, B. (2008, October 14). *Brian Eno's Bloom: new album or ambient joke?* Retrieved from <https://www.theguardian.com/music/musicblog/2008/oct/14/brian-eno-bloom-ipod-iphone>
- Martin, G. D. (2016, February 1). *My Experience as a Virtual War Photographer in Battlefield 1*. (K. Screen, Producer) Retrieved from killscreen.com/articles/corprography-battlefield-1-beta/

- Matney, L. (2016, June 24). *Facebook just bought VR audio company Two Big Ears and is making their tech free to developers*. Retrieved from Tech Crunch: <https://techcrunch.com/2016/05/23/facebook-just-bought-vr-audio-company-two-big-ears-and-is-making-their-tech-free-to-developers/>
- Matney, L. (2017, April 8). *Facebook debuts its first dedicated virtual reality app, Facebook 360*. Retrieved from Tech Crunch: <https://techcrunch.com/2017/03/08/facebook-debuts-its-first-dedicated-virtual-reality-app-facebook-360/>
- Matney, L. (2017, November 28). *Virtual Reality Headset Unit Sales Are Slowly Improving*. Retrieved from Tech Crunch: techcrunch.com/2017/11/28/virtual-reality-headset-unit-sales-are-slowly-improving/
- Maurer IV, J. A. (1999, June 4). *Interactive Electronic in Computer Music*. (C. f. Stanford, Producer) Retrieved from ccrma.stanford.edu/~blackrse/interactive_electronics.html.
- McLeran, A. (2017, March 15). *The Future of Audio in Unreal Engine | GDC 2017 | Unreal Engine*. Retrieved from YouTube: <https://youtu.be/ErejaBCids>
- McMahan, A. (2003). Immersion, Engagement, and Presence: A Method for Analyzing 3-D Video Games. *The Video Game Theory Reader*, 67–86.
- McNeill, E. (2014, September 22). *The Gender Gap in VR*. (Gamasutra, Producer) Retrieved from www.gamasutra.com/blogs/EMcNeill/20140922/226135/The_Gender_Gap_in_VR.php
- MelodyVR. (2015, June 18). *Homepage*. Retrieved from www.melodyvr.com
- Microsoft. (2017, May 12). *Hololens*. Retrieved from www.microsoft.com/en-us/hololens
- Milani, M. (2012, December 26). *Peter Chilvers: Visual and Tactical Music*. Retrieved from DigiCult: <http://digicult.it/digimag/issue-048/peter-chilvers-visual-and-tactical-music/>
- Minion, M. (2017, September 1). *Pygmalion's Spectacles: Using Berkeley's Immaterialism to Understand the Potential for Telepresence in Virtual Reality*. (M. o. Minion, Producer) Retrieved from musingsofamariominion.com/2017/09/01/pygmalions-spectacles-using-berkeleys-immaterialism-to-understand-the-potential-for-telepresence-in-virtual-reality
- Morales-Mazanares, R., & al., e. (2001). SICIB: An Interactive Music Composition System Using Body Movments. *Computer Music Journal*, 25(2), 25–36.
- Murphy, J., & al, e. (2012). Musical Robotic in a Loudspeaker World: Developments in Alternative Approaches to Localization and Spatialization. *Leonardo Music Journal*, 22, 41–48.
- Nechvatal, J. (2014, April 23). *The Lush Life of Virtual Architecture*. (Hyperallergic, Producer) Retrieved from hyperallergic.com/122097/the-lush-life-of-virtual-architecture/

- Nolo. (2018, December 19). *CVI*. Retrieved from Nolo: <https://www.nolovr.com>
- Nonesuch. (2012, Decemember 21). *Björk Unveils New Holiday Version of 'Solstice' App from 'Biophilia'*. Retrieved from Nonesuch: www.nonesuch.com/journal/bjork-unveils-new-holiday-version-solstice-app-biophilia-2011-12-21
- Oculus. (2016, January 26). *Oculus Audio SDK Guide*. Retrieved from Oculus Developers Documentation: <https://developer.oculus.com/documentation/audiosdk/latest/concepts/book-audiosdk/>
- Oculus. (2014, December 21). *Developer Documentation*. Retrieved from Oculus: <https://developer.oculus.com/documentation/>
- Oculus. (2015, January 21). *Oculus Rift: Step Into the Game*. Retrieved from <https://www.kickstarter.com/projects/1523379957/oculus-rift-step-into-the-game/description>
- Oculus. (2016, February 10). *Introduction to Virtual Reality Audio*. (O. D. Center, Producer) Retrieved from developer.oculus.com/documentation/audiosdk/latest/concepts/book-audio-intro/
- Oculus. (2017, September 2017). *Beyond Surround Sound: Audio Advance in VR*. (Oculus, Producer) Retrieved from www.oculus.com/blog/beyond-surround-sound-audio-advances-in-vr/.
- Oculus. (2017, November 21). *Oculus Store: Dispatch*. Retrieved from <https://www.oculus.com/experiences/go/1279074825522400/>
- Oculus. (2018, September 26). *Introducing Oculus Quest, Our First 6DOF All-in-One VR System, Launching Spring 2019*. Retrieved from Oculus: <https://www.oculus.com/blog/introducing-oculus-quest-our-first-6dof-all-in-one-vr-system-launching-spring-2019/>
- Oculus. (2018, October 28). *Oculus Quest*. Retrieved from Oculus: <https://www.oculus.com/quest/>
- Oculus. (2018, August 15). *Oculus Store: Netflix*. Retrieved from <https://www.oculus.com/experiences/go/890562797701371/>
- Oliveros, P. (1995). Acoustic and Virtual Space as a Dynamic Element of Music. *Leonardo Music Journal*, 5, 19–22.
- Olszacki, P. (2017, November 28). *Getting a room plan with ARKit (The case where size matters)*. Retrieved from Medium: <https://medium.com/inborn-experience/getting-a-room-plan-with-arkit-the-case-where-size-matters-1e104ee42421>
- Omnivert. (2017, February 19). *Interactive 360° Video*. Retrieved from Omnivert: <https://www.omnivirt.com/360-video-hotspots>

- Omnivirt. (2013, August 13). *360° VIRTUAL REALITY AD PLATFORM FOR ADVERTISERS AND PUBLISHERS*. Retrieved from Omnivirt: <https://www.omnivirt.com/>
- Owsinki, B. (2013, July 15). *5 Reasons Why Album Apps Will Fail*. Retrieved from Forbes: www.forbes.com/sites/bobbyowsinski/2013/07/15/5-reasons-why-album-apps-will-fail/#26929edb20c6
- Pagliery, J. (2015, September 23). *Facebook launches 360-degree videos - and they're amazing*. Retrieved from CNN Business: <https://money.cnn.com/2015/09/23/technology/facebook-360-video/index.html>
- Pareles, J. (2013, July 4). *Jay-Z Is Watching, and He Knows Your Friends*. Retrieved from www.nytimes.com/2013/07/05/arts/music/jay-z-is-watching-and-he-knows-your-friends.html
- Patel, N. (2018, July 19). *The US Patent Office has issued 10 million patents*. Retrieved from The Verge: <https://www.theverge.com/2018/6/19/17478898/uspto-utility-patents-10-million-issued>
- PICINIC. (2011, September 19). *Scott Snibbe on Biophilia, An App Album / PICINIC Festival 2011*. Retrieved from Vimeo: vimeo.com/29256409
- Pope, S. T. (1994). Editor's Notes: A Taxonomy of Computer Music. *Computer Music Journal*, 18(1), 5–7.
- Pratt, C. J. (2017, May 15). *In the Shadow of the Holodeck*. (Medium, Producer) Retrieved from medium.com/@charlesjpratt/in-the-shadow-of-the-holodeck-ae6bbb4ac195
- Reeps One. (2016, June 2019). *YouTube*. Retrieved from www.youtube.com/watch?v=OMLgliKYqal
- Rieger, M. (2018, November). *Spatial Audio Formats*. Retrieved from VRTONUNG: <https://www.vrtonung.de/en/spatial-audio-support-for-360-video-platform/>
- Roads, C. (2001). *Microsounds*. Boston: MIT Press.
- Roettgers, J. (2017, May 22). *Study Predicts Fewer Than 10 Million Monthly U.S. VR Headset Users This Year, 17 Million by 2019*. Retrieved from Variety: variety.com/2017/digital/news/vr-headset-data-mau-2017-2019-1202440211/
- Roginska, A., & Geluso, P. (2018). *Immersive Sound: The Art and Science of Binaural and Multi-Channel Audio*. New York, NY: Routledge.
- Roli. (2017, December 23). *Seaboard RISE 25*. Retrieved from Roli: <https://roli.com/products/seaboard/rise-25>
- Rubin, P. (2018). *Future Presence: How Virtual Reality is Changing Human Connection, Intimacy, and the Limits of Ordinary Life*. New York, NY: HarperCollins Publishers.

- Sag, A. (2019, January 4). *Location-Based VR: The Next Phase Of Immersive Entertainment*. Retrieved from Forbes: <https://www.forbes.com/sites/moorinsights/2019/01/04/location-based-vr-the-next-phase-of-immersive-entertainment/#273cdc923f57>
- Samsung. (2016, September 4). *Gear VR*. Retrieved from Samsung: <https://www.samsung.com/global/galaxy/gear-vr/>
- Schifrin, L. (2011). *Music COmposition for Film and Television*. New York, NY: Leonard Corporation.
- Schutze, S. (2018). *New Realities in Audio: A Practical Guide for VR, AR, MR, and 360 Video*. New York, NY: Taylor & Francis.
- Seegert, A. (2009). 'Doing There' vs. 'Being There'" Performing Presence in Interactive Fiction. *Journal of Gaming and Virtual worlds*, 1(1), 23–37.
- Sherbourne, S. (2017, October 18). *Ambisonics and VR/360 Audio in Pro Tools | HD*. Retrieved from Avid Blogs: <http://www.avidblogs.com/ambisonics-vr360-audio-pro-tools-hd/>
- Sherburne, P. (2017, February 16). *A Conversation With Brian Eno About Ambient Music*. (Pitchfork, Producer) Retrieved from <https://pitchfork.com/features/interview/10023-a-conversation-with-brian-eno-about-ambient-music/>
- Sicat, M. (2017, July 31). *Queering the COntroller*. (A. G. Studies, Producer) Retrieved from analoggamestudies.org/2017/07/queering-the-controller/
- Singleton, M. (2017, June 29). *Jay Z Has USed His Latest Albums as Corporate Bargaining Chips*. Retrieved from The Verge: www.theverge.com/2017/6/29/15892648/jay-z-albums-corporate-bargaining-chips-444-sprint-tidal
- Smule. (2001, April 2). *Homepage*. Retrieved from <https://www.smule.com>
- Smule. (2010, April 23). *Smule iTunes Page*. Retrieved from <https://itunes.apple.com/us/app/ocarina/id293053479?mt=8&app=itunes>
- Soper, T. (2017, September 21). *How Oculus Audio Engineers Are Using New Sound Technology to Enhance Virtual Reality Experiences*. (Geekwire, Producer) Retrieved from www.geekwire.com/2017/oculus-audio-engineers-using-new-sound-technology-enhance-vr-experiences/.
- Starovoitov, V., & Samal, D. (1999). A geometric approach to face recognition. *Proceedings of the IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP'99)*. Antalya: IEEE.
- Steam Audio. (2017, March 2). *Steam Audio :: Introducing Steam Audio*. (S. Community, Producer) Retrieved from steamcommunity.com/games/596420/announcements/detail/521693426582988261

- Sterne, J. (2012). *MP3: The Meaning of a Format*. Durham, NC: Duke University Press.
- Sweet, M. (2015). *Writing Interactive Music for Video Games: A Composer's Guide*. New York, NY: Addison-Wesley Professional.
- Taylor, D. G. (2016, April 10). *Virtual reality technologies that failed hard*. Retrieved from Grunge: <https://www.grunge.com/11190/virtual-reality-technologies-that-failed-hard/>
- The Atlantic Media Company. (2015, September 9). *How Fast Is Technology Accelerating?* Retrieved from The Atlantic: www.theatlantic.com/sponsored/prudential-great-expectations/how-fast-is-technology-accelerating/360/
- The Game of Life. (2011, July 3). *About Wave Field Synthesis*. (T. G. Life, Producer) Retrieved from gameoflife.nl/en/about/about-wave-field-synthesis/.
- The Holladay Brothers. (2018, January 12). *Homepage*. Retrieved from <http://www.theholladaybrothers.com/>
- The MIDI Association. (2019, April 22). *MIDI Polyphonic Expression (MPE) Specification Adopted!* Retrieved from MIDI Association: <https://www.midi.org/articles-old/midi-polyphonic-expression-mpe>
- The MIDI Association. (2019, February 18). *The MIDI Manufacturers Association (MMA) and the Association of Music Electronics Industry (AMEI) announce MIDI 2.0™ Prototyping*. Retrieved from MIDI Association: <https://www.midi.org/articles-old/the-midi-manufacturers-association-mma-and-the-association-of-music-electronics-industry-amei-announce-midi-2-0tm-prototyping>
- Thinkwik. (2018, April 20). *CryEngine vs Unreal vs Unity: Select the Best Game Engine*. Retrieved from Medium: <https://medium.com/@thinkwik/cryengine-vs-unreal-vs-unity-select-the-best-game-engine-eaca64c60e3e>
- TNW. (2016, March 24). *This engine is dominating the gaming industry right now*. Retrieved from The Next Web: <https://thenextweb.com/gaming/2016/03/24/engine-dominating-gaming-industry-right-now/>
- Turi, J. (2014, February 16). *The Sights and Scents of the Sensorama Simulator*. (Engaget, Producer) Retrieved from www.engadget.com/2014/02/16/morton-heiligs-sensorama-simulator/
- Unity. (2014, July 1). *Microphone*. Retrieved from Unity Developer Documentation: <https://docs.unity3d.com/Manual/class-Microphone.html>
- Unity. (2014, July 1). *Reverb Zones*. Retrieved from Unity Developer Documentation: <https://docs.unity3d.com/Manual/class-AudioReverbZone.html>
- Unity. (2014, July 1). *Scripting Overview*. Retrieved from Unity Developer Documentation: <https://docs.unity3d.com/Manual/ScriptingConcepts.html>

- Unity. (2014, July 28). *Unity Growth Facts*. Retrieved from Unity Company Facts: <https://unity3d.com/public-relations>
- Unity. (2018, September 17). *Real-Time Filmmaking, Explained*. Retrieved from Unity Solutions: <https://unity.com/solutions/film/real-time-filmmaking-explained>
- Unity. (2018, December 1). *Unity's evolving best practices*. Retrieved from Unity - Creator's Corner: <https://unity3d.com/how-to/unity-best-practices-for-engine-performance>
- Unity. (2019). *Unity Online Tutorials*. Retrieved from Unity: <https://unity.com/learn>
- Unreal. (2018, November 10). *Audio System Overview*. Retrieved from Unreal Engine Developer Documentation: <https://docs.unrealengine.com/en-us/Engine/Audio/Overview>
- Unreal. (2018, December 1). *Audio Volume Overview*. Retrieved from Unreal Engine Developer Documentation: <https://docs.unrealengine.com/en-US/Engine/Actors/Volumes/AudioVolumeDetails>
- Unreal Engine. (2018, October 28). *Blueprints Overview*. Retrieved from Unreal Engine Documentation: <https://docs.unrealengine.com/en-US/Engine/Blueprints/Overview>
- Valve. (2017, March 23). *A BENCHMARK IN IMMERSIVE AUDIO SOLUTIONS FOR GAMES AND VR*. Retrieved from Steam Audio: <https://valvesoftware.github.io/steam-audio/>
- Vasquez, J. C., Tahiroglu, K., & Kildal, J. (2017). Idiomatic Composition Practices for New Musical Instruments: Context, Background and Current Applications. *NIME* (pp. 174-179). Denmark: Aalborg University Copenhagen.
- Vieyra Software. (2018, September 24). *Vieyra Software*. Retrieved from Medium: <https://medium.com/@vieyrasoftware/comparing-google-arcore-and-apple-arkit-81b4727132ad>
- Virtual Reality Society. (2015, October 16). *History of Virtual Reality*. Retrieved from Virtual Reality Society: www.vrs.org.uk/virtual-reality/history.html
- Wanderlvst. (2017, July 9). *[Block Rocking Beats] EP1: Visual Music Production Tool in Virtual Reality*. Retrieved from youtu.be/8aXi-yV5-oU
- Waves Audio Ltd. (2013, February 21). *Michael Gerzon - A true Audio Visionary*. (Acoustics.net, Producer) Retrieved from archive.today/2013.02.21-134930/http://www.acoustics.net/content.asp?id=46
- Waves Audio Ltd. (2017, October 10). *Ambisonics Explained: A Guide for Sound Engineers*. (W. A. Ltd., Producer) Retrieved from www.oculus.com/blog/beyond-surround-sound-audio-advances-in-vr/

- White, B. (2018, September 4). *iOS vs Android Audio Apps*. Retrieved from Future Sonic:
<https://futuresonic.io/blog/ios-vs-android-in-music-apps/>
- Wingfield, N., & Isaac, M. (2016, July 11). *Pokémon Go Brings Augmented Reality to a Mass Audience*. Retrieved from The New York Times:
<https://www.nytimes.com/2016/07/12/technology/pokemon-go-brings-augmented-reality-to-a-mass-audience.html>
- Yang, R. (2017, April 5). *If You Walk in Someone Else's Shoes, Then You've Taken Their Shoes: Empathy Machines as Appropriation Machines*. (R. -D. Blog, Producer) Retrieved from
www.blog.radiator.debacl.us/2017/04/if-you-walk-in-someone-elses-shoes-then.html
- Zhu, A. Z., Atanasov, N., & Daniilidis, K. (2017). Event-based Visual Inertial Odometry. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu: IEEE.

Appendix A

The “Player_Control” C# script for use in Unity that controls the player’s locomotion in *Infinite Void* based upon the HMD’s orientation and hand-held controller.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player_Control : MonoBehaviour
{
    public float speed; // variable to set speed of user locomotion

    void Update()
    {
        // if the trigger is pressed
        if (OVRInput.Get(OVRInput.Button.PrimaryIndexTrigger) == true)
        {
            // move the player forward
            transform.Translate(Vector3.forward * speed * Time.deltaTime, Camera.main.transform);
        }
        // else if the thumbpad is pressed
        else if (OVRInput.Get(OVRInput.Button.One) == true)
        {
            // move the player backwards
            transform.Translate(Vector3.forward * -speed * Time.deltaTime, Camera.main.transform);
        }
    }
}
```

Appendix B

The “Player_Raycast” C# script for use in Unity that detects when the player looks directly at a game object in *Infinite Void* and prompts the “Sphere_Response_Script” (see Appendix C) to perform several functions.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player_Raycast : MonoBehaviour
{
    public string layerName; // layer of detectable objects
    public string triggerObjectTag; // tag of detectable objects
    public float maxDistance; // max distance to test for objects
    public float coolDownTime; // time before an object will be detected again

    private RaycastHit hit;
    private bool objectHit;
    private float waitTimer;
    private float transparency;
    private GameObject lastObject;

    // initialize the script at runtime
    private void Start()
    {
        objectHit = false;
        waitTimer = 0;
    }

    // called once per frame
    void FixedUpdate()
    {
        Ray raydirection = new Ray(Camera.main.transform.position, Camera.main.transform.forward);
        LayerMask layerMask = LayerMask.GetMask(layerName);

        // if an object is detected
        if (Physics.Raycast(raydirection, out hit, maxDistance, layerMask))
        {
            // if that object matches the specified tag and it has not already been detected
            if (hit.collider.tag == triggerObjectTag && objectHit == false)
            {
                objectHit = true;

                // call the Glow() function of the object's "Sphere_Response_Control" script
                hit.collider.gameObject.GetComponent<Sphere_Response_Conrol>().Glow();
                lastObject = hit.collider.gameObject; // temporarily store the object's ID
            }
        }

        // if an object is not, or no longer detected
        else
        {
            objectHit = false;

            // if an object was previously detected
            if (lastObject != null)
            {
                // call the Dim() function of the object's "Sphere_Response_Control" script
            }
        }
    }
}
```

```
        lastObject.GetComponent<Sphere_Response_Control>().Dim();
    }

    // if an object has been detected start a timer to control
    // when it can be detected for a second time
    if (objectHit == true)
    {
        if (waitTimer < coolDownTime)
        {
            waitTimer += Time.deltaTime;
        }
        else if (waitTimer >= coolDownTime)
        {
            waitTimer = 0;
            objectHit = false;
        }
    }
}
}
```

Appendix C

The “Sphere_Response_Script” C# script for Unity that determines the response of the associated game object in *Infinite Void* when it is detected by the “Player_Raycast” script (see Appendix B).

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Sphere_Response_Control : MonoBehaviour
{
    public float speed; // sets the transition rate between invisible and visible

    private float transparency;
    private float volume;
    private bool locked;
    private float red;
    private float green;
    private float blue;
    private bool eventIsPlaying;

    // initialize several script parameters
    private void Start()
    {
        red = 1.0f;
        green = 1.0f;
        blue = 1.0f;
        eventIsPlaying = false;
        locked = false;
    }

    // every frame update the current appearance of the sphere
    void Update()
    {
        this.GetComponent<MeshRenderer>().material.color = new Color(red, green, blue, transparency);
    }

    // public function that will be called by when the object this script is attached to
    // is detected by the Player_Raycast script
    // -----
    // this function triggers a Wwise event and transition the object from invisible to visible
    public void Glow()
    {
        // if the Wwise event has not already been triggered
        if (eventIsPlaying == false)
        {
            // play the Wwise event
            AkSoundEngine.PostEvent("Play_Evo_1", gameObject);
            eventIsPlaying = true;
        }

        // if the sphere is not completely illuminated
        if (transparency < 2.0f)
        {
            // keep making it brighter
            transparency += Time.deltaTime * speed;
        }
    }
}
```

```

// if the sphere is completely illuminated
else
{
    // and the controller's thumb pad is pressed
    if (OVRInput.Get(OVRInput.Touch.PrimaryTouchpad) == true)
    {
        // make the sphere blue...
        transparency = 2.0f;
        red = 0.0f;
        green = 0.0f;
        blue = 1.0f;

        // ...and set its state to "locked"
        locked = true;
    }
}
}
// public function that will be called by when the object this script is attached to
// is no longer detected by the Player_Raycast script
// -----
// this function triggers a Wwise event and transition the object from visible to invisible
public void Dim()
{
    // if the sphere is not "locked" on...
    if (locked == false)
    {
        // ...and it is not yet invisible
        if (transparency > 0.0f)
        {
            // keep making it more transparent
            transparency -= Time.deltaTime * speed;
        }
        // if the sphere is invisible
        else
        {
            // keep it invisible...
            transparency = 0.0f;

            // ...and if the Wwise event is playing
            if (eventIsPlaying == true)
            {
                // tell the Wwise event to stop playing
                AkSoundEngine.PostEvent("Stop_Evo_1", gameObject);
                eventIsPlaying = false;
            }
        }
    }
}
// if the sphere is "locked" on...
if (locked == true)
{
    // ...and the controller's thumb pad is pressed
    if (OVRInput.Get(OVRInput.Touch.PrimaryTouchpad) == true)
    {
        // make the sphere invisible
        transparency = 0.0f;
        red = 1.0f;
        green = 1.0f;
        blue = 1.0f;

        // ...and set its state to "unlocked"
        locked = false;
    }
}
}
}
}

```

Appendix D

A C# script for Unity that triggers an animation to fade the current scene to black via a Unity-defined animation and then loads a targeted scene (Brackeys, 2018).

```
using UnityEngine;
using UnityEngine.SceneManagement;

public class SceneChanger : MonoBehaviour
{
    public Animator animator;
    public string targetScene;

    // When player enters gameObject trigger transition to a targeted scene
    void OnTriggerEnter(Collider other)
    {
        if (other.tag == "Player")
        {
            FadeToScene();
        }
    }

    // function to begin scene fade out
    public void FadeToScene ()
    {
        animator.SetTrigger("FadeOut");
    }

    // function to be called at end of fade out animation
    // Load target scene
    public void OnFadeComplete()
    {
        SceneManager.LoadScene(targetScene);
    }
}
```

Appendix E

A C# script for spawning any prefab object randomly within a scene in Unity.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Random_Prefab_Spawner : MonoBehaviour
{
    public int numberOfObjects;           // declare the number of prefabs to spawn
    public GameObject prefabObject;      // what prefab
    public float range;                  // the range to generate within
    private Vector3 spawnLocation;       // the spawn location
    private float x;                     // the x location
    private float y;                     // the y location
    private float z;                     // the z location

    void Start()
    {
        // repeat the process for number of requested objects
        for (int i = 0; i < numberOfObjects; i++)
        {
            // generate a random number within the given range for x, y, and z
            spawnLocation.x = Random.Range(range * -1.0f, range);
            spawnLocation.y = Random.Range(range * -1.0f, range);
            spawnLocation.z = Random.Range(range * -1.0f, range);
            // create a clone of the indicated prefab at the location
            Instantiate(prefabObject, spawnLocation, Quaternion.identity);
        }
    }
}
```

Appendix F

A C# script for finding and sending the X, Y, and Z distances between a Unity game object and the player game object as RTPCs to the Wwise sound engine.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class XYZ_To_Player : MonoBehaviour
{
    GameObject player; // a reference to the player's gameObject
    float x_Distance; // a variable for hold the x distance
    float y_Distance; // y distance
    float z_Distance; // z distance

    private void Start()
    {
        // Find the player gameObject and assign it's ID to the 'player' variable
        player = GameObject.Find("Player");
    }

    // Update is called once per frame
    void Update()
    {
        // find the absolute distance on each plane and limit results to be between 0 and 100 units
        x_Distance =
            Mathf.Clamp(Mathf.Abs(player.transform.position.x - transform.position.x), 0.0f, 100.0f);
        y_Distance =
            Mathf.Clamp(Mathf.Abs(player.transform.position.y - transform.position.y), 0.0f, 100.0f);
        z_Distance =
            Mathf.Clamp(Mathf.Abs(player.transform.position.z - transform.position.z), 0.0f, 100.0f);

        // Set each of the corresponding Wwise RTPC to the current distance value
        AkSoundEngine.SetRTPCValue("X_Distance", x_Distance);
        AkSoundEngine.SetRTPCValue("Y_Distance", y_Distance);
        AkSoundEngine.SetRTPCValue("Z_Distance", z_Distance);
    }
}
```