

Playcrafting GGJ 2022: Intro to Game Audio

David Su
@usdivad

<https://github.com/usdivad/GGJ22-GameAudio>

About Me

About Me

- Musician, audio software engineer, sound designer, game developer

About Me

- Musician, audio software engineer, sound designer, game developer
- Audio programmer at Brass Lion Entertainment 

About Me

- Musician, audio software engineer, sound designer, game developer
- Audio programmer at Brass Lion Entertainment 
- Some games I've worked on:



About Me

- Musician, audio software engineer, sound designer, game developer
- Audio programmer at Brass Lion Entertainment 
- Some games I've worked on:



<http://usdivad.com/>

Overview

Overview

- SFX 

Overview

- SFX 
- Music 

Overview

- SFX 
- Music 
- Dialogue 

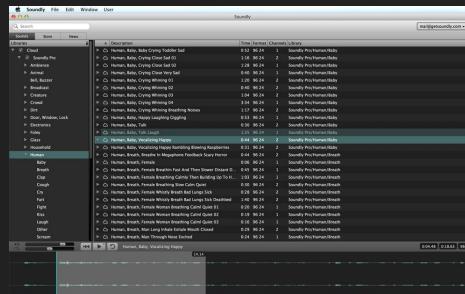
Overview

- SFX 
- Music 
- Dialogue 
- Implementation 

SFX 

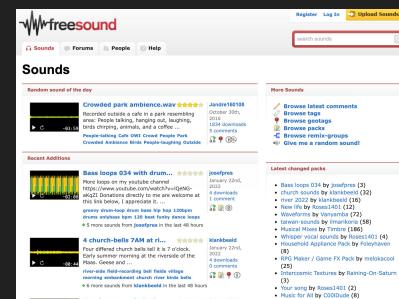
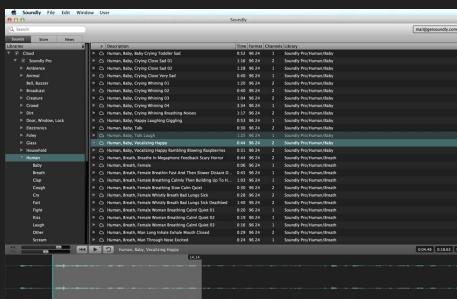
SFX: Sound Sources

- Sound libraries
 - Paid (BOOM, Soundly, A Sound Effect)



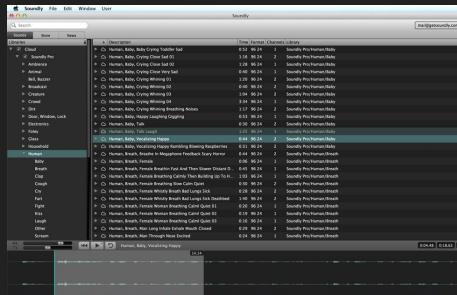
SFX: Sound Sources

- Sound libraries
 - Paid (BOOM, Soundly, A Sound Effect)
 - Free (Freesound, Library of Congress, Sonniss GameAudioGDC)
 - Triple check CC0 / licensing!



SFX: Sound Sources

- Sound libraries
 - Paid (BOOM, Soundly, A Sound Effect)
 - Free (Freesound, Library of Congress, Sonniss GameAudioGDC)
 - Triple check CC0 / licensing!
- Recording sounds yourself
 - Microphone + interface
 - Setup (room, mic position)



The screenshot shows the homepage of the freesound website. At the top, there's a search bar with the placeholder "Search sounds". Below it, a section titled "Recent sounds" displays a list of sounds with their names, file sizes, and download counts. One sound, "Crowed park ambientness.wav" by James101010, is highlighted. Below that, a section titled "Recent additions" shows a list of sounds added recently, with one sound, "Basic loops 024 with drums..." by jessievers, highlighted.



SFX: Sound Design

- DAWs
 - Paid: Pro Tools, REAPER, Logic Pro, Ableton Live
 - Free: Audacity, Tracktion Waveform, Garageband, BandLab
 - Free trials



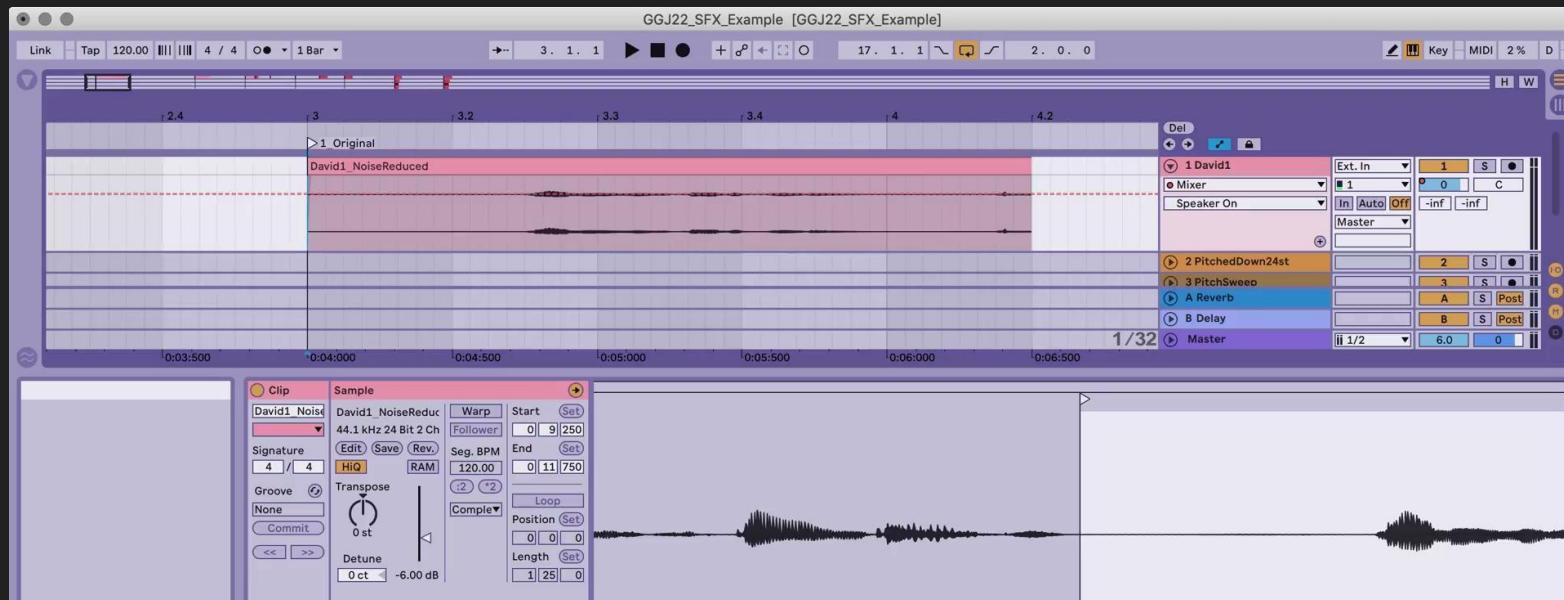
SFX: Sound Design

- DAWs
 - Paid: Pro Tools, REAPER, Logic Pro, Ableton Live
 - Free: Audacity, Tracktion Waveform, Garageband, BandLab
 - Free trials
- Common techniques
 - Trimming
 - Pitch-shifting
 - Reversing
 - Layering
 - Effects



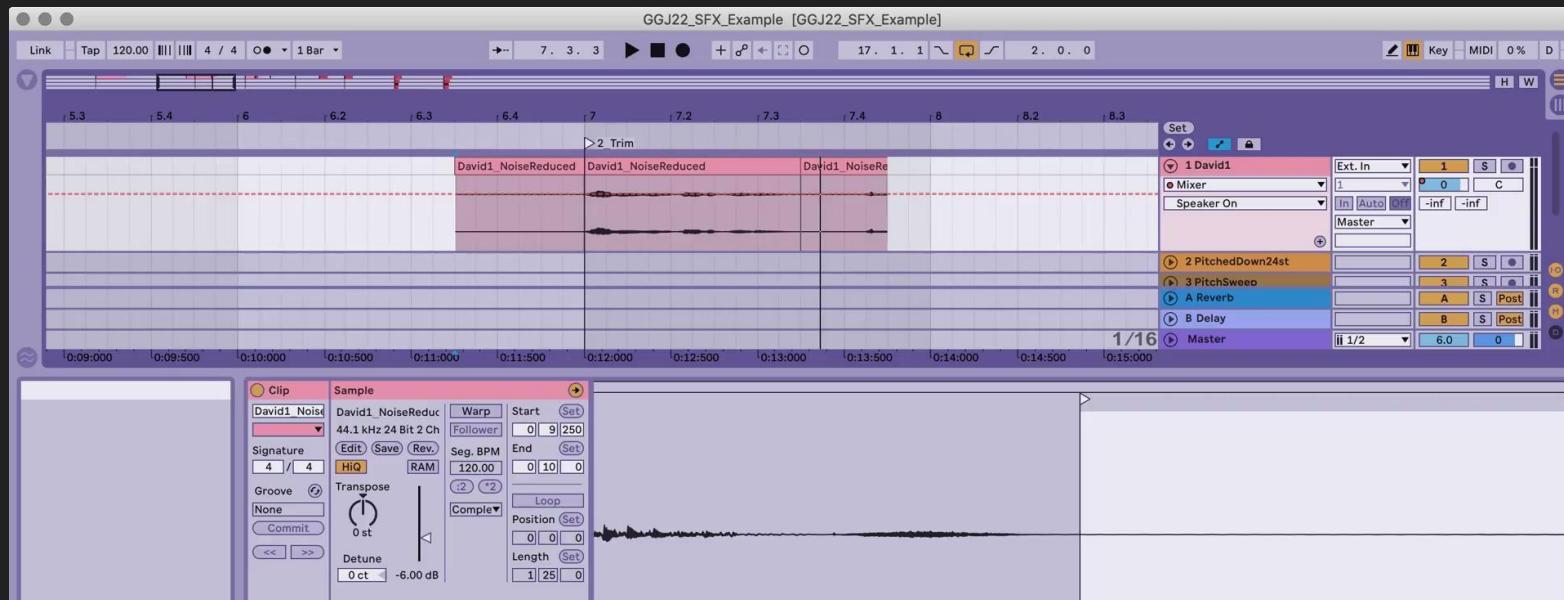
SFX Demo: Designing a Sound

- Original



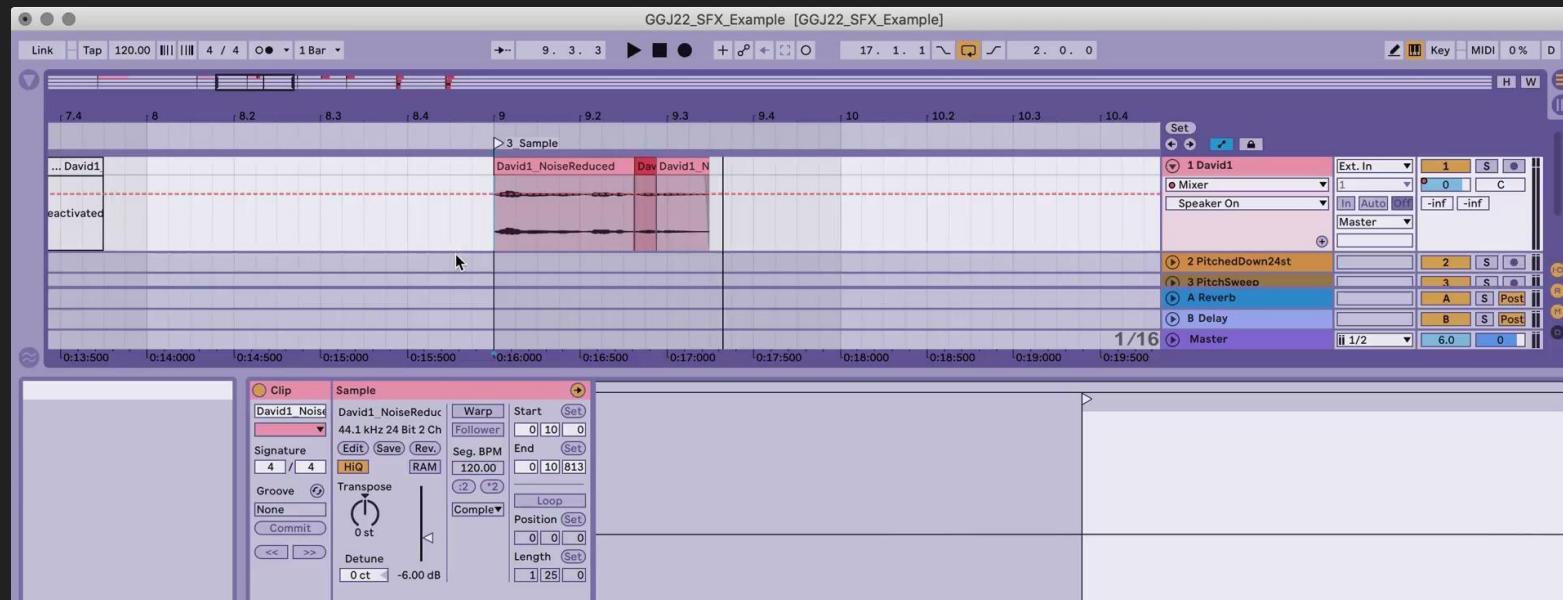
SFX Demo: Designing a Sound

- Trimming



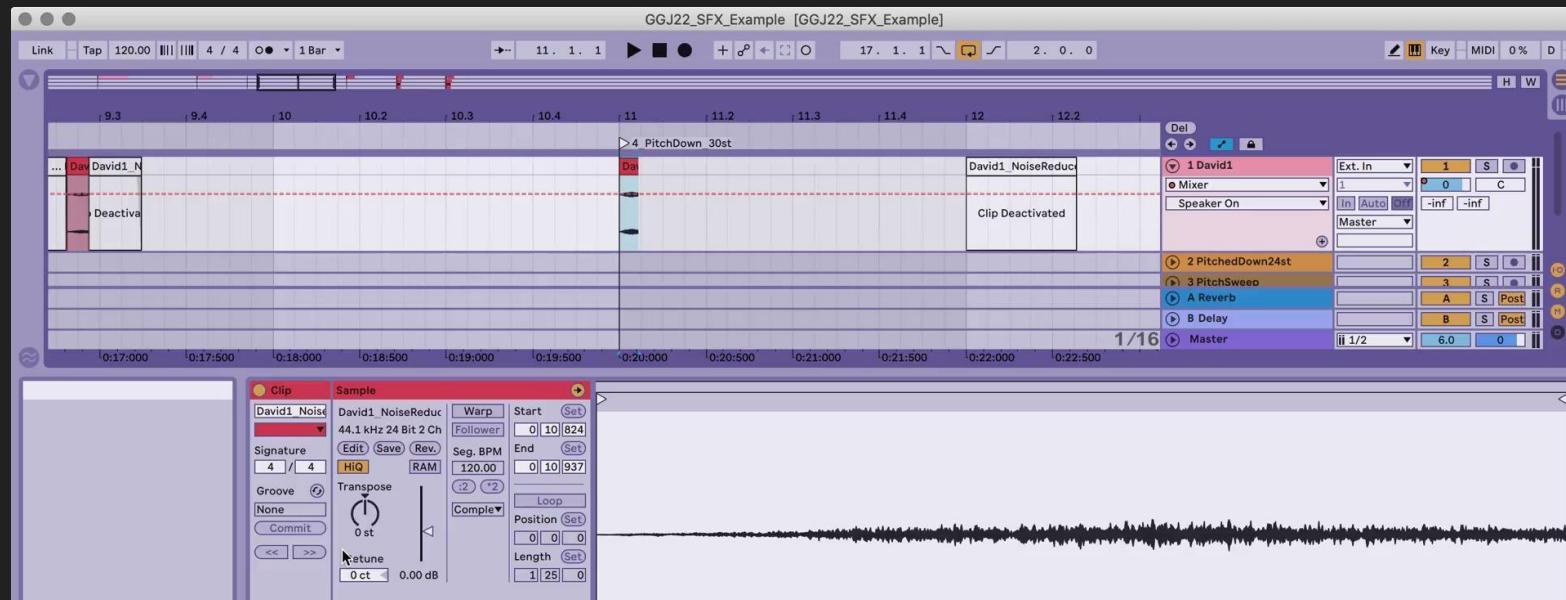
SFX Demo: Designing a Sound

- Trimming/Sampling



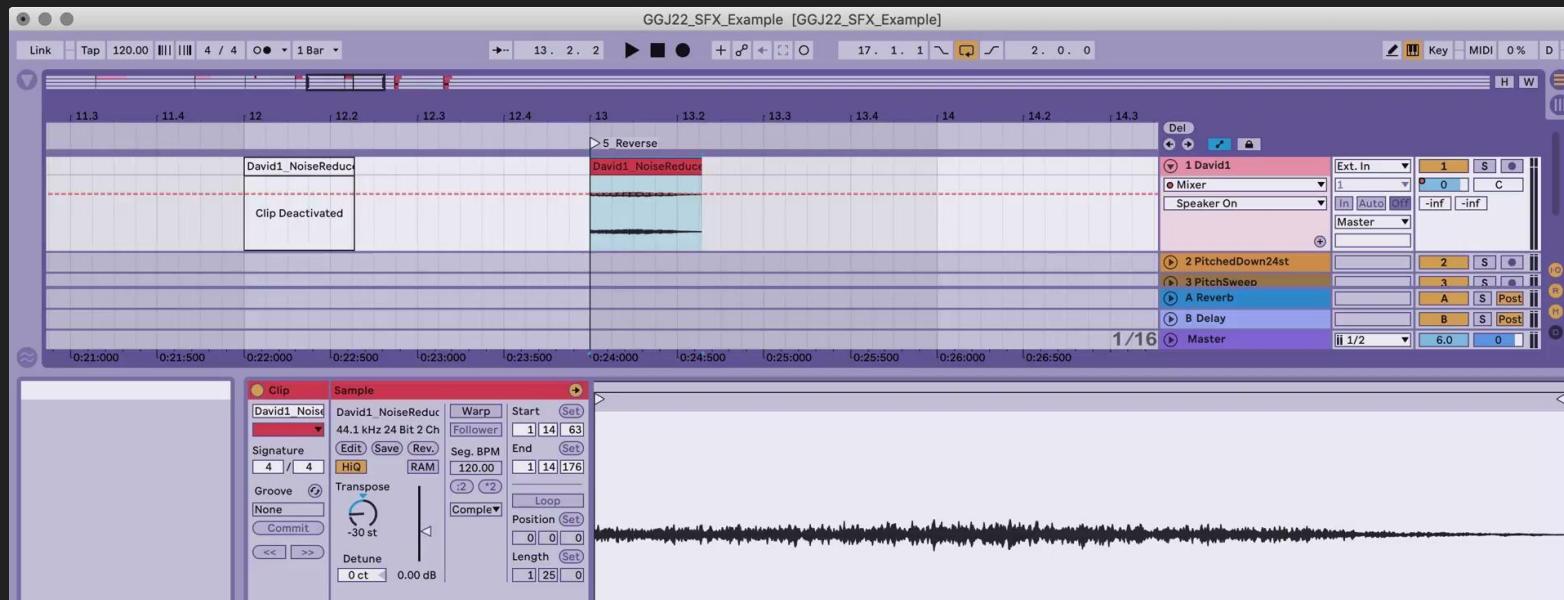
SFX Demo: Designing a Sound

- Pitch-shifting



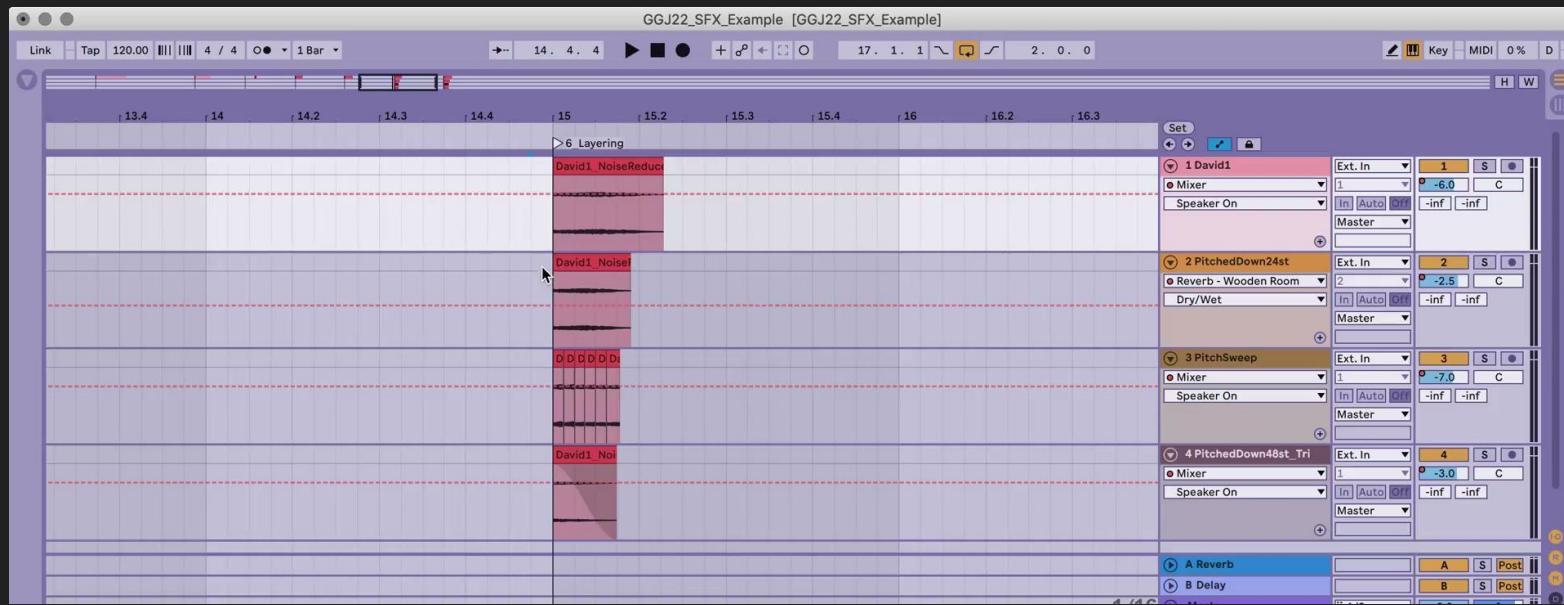
SFX Demo: Designing a Sound

- Reversing



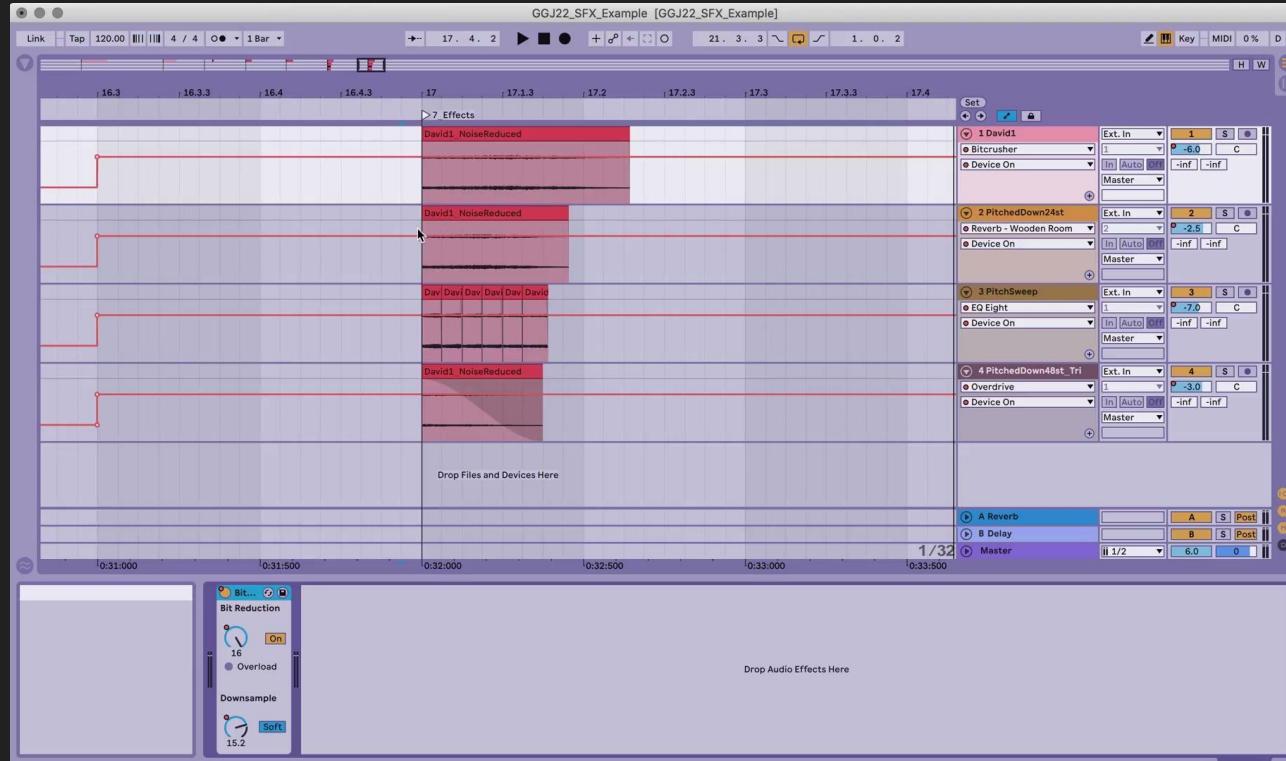
SFX Demo: Designing a Sound

- Layering



SFX Demo: Designing a Sound

- Effects



SFX: Mixing

SFX: Mixing

- Balancing volumes (gain, compressor)

SFX: Mixing

- Balancing volumes (gain, compressor)
- Balancing low, mid, high frequencies (EQ)

SFX: Mixing

- Balancing volumes (gain, compressor)
- Balancing low, mid, high frequencies (EQ)
- Panning and spatialization

SFX: Mixing

- Balancing volumes (gain, compressor)
- Balancing low, mid, high frequencies (EQ)
- Panning and spatialization
- Overall palette

SFX: Mixing

- Balancing volumes (gain, compressor)
- Balancing low, mid, high frequencies (EQ)
- Panning and spatialization
- Overall palette
- Trust your ears!

Music 🎵

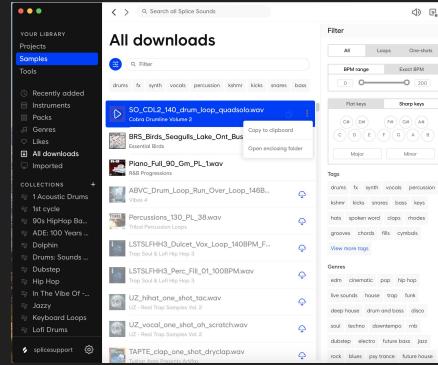
Music: Sources

Music: Sources

- Using existing music

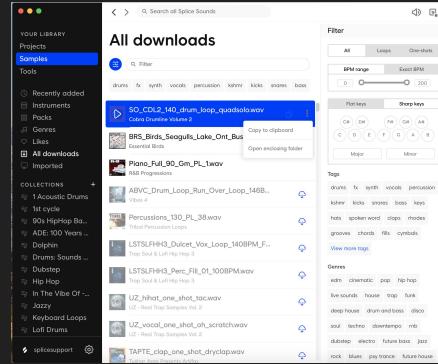
Music: Sources

- Using existing music
 - Loop libraries (Splice, Loopmasters)



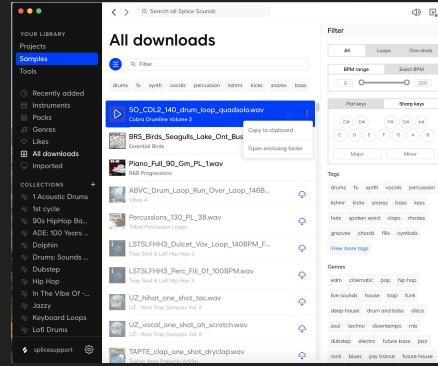
Music: Sources

- Using existing music
 - Loop libraries (Splice, Loopmasters)
 - Asset stores (itch.io, Unity)



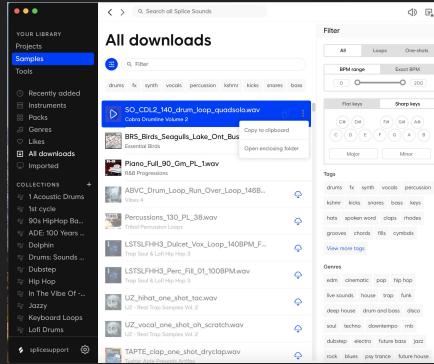
Music: Sources

- Using existing music
 - Loop libraries (Splice, Loopmasters)
 - Asset stores (itch.io, Unity)
 - Ask musicians directly! (Bandcamp, SoundCloud)



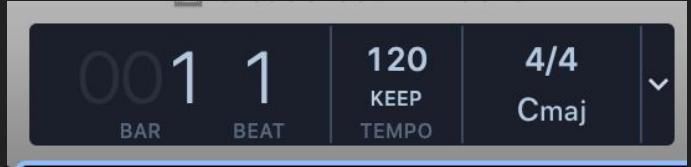
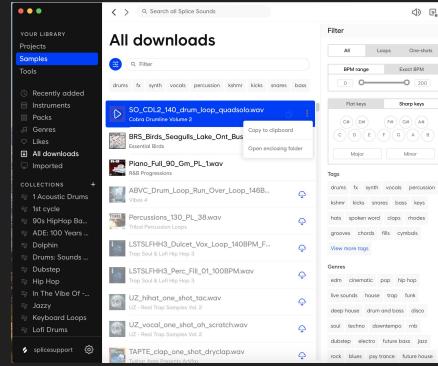
Music: Sources

- Using existing music
 - Loop libraries (Splice, Loopmasters)
 - Asset stores (itch.io, Unity)
 - Ask musicians directly! (Bandcamp, SoundCloud)
- Making your own music
 - DAWs



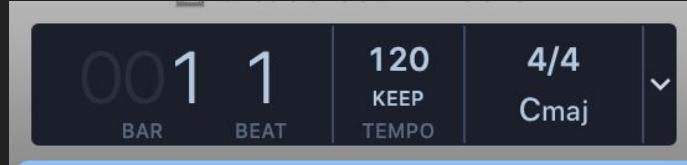
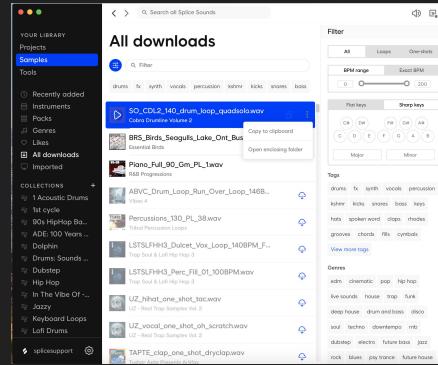
Music: Sources

- Using existing music
 - Loop libraries (Splice, Loopmasters)
 - Asset stores (itch.io, Unity)
 - Ask musicians directly! (Bandcamp, SoundCloud)
- Making your own music
 - DAWs
 - Basic properties
 - Tempo
 - Key



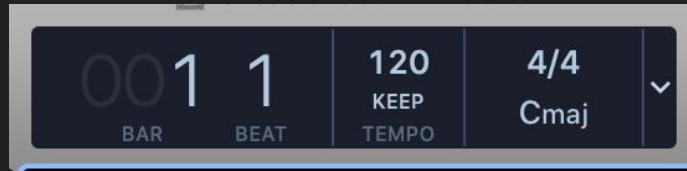
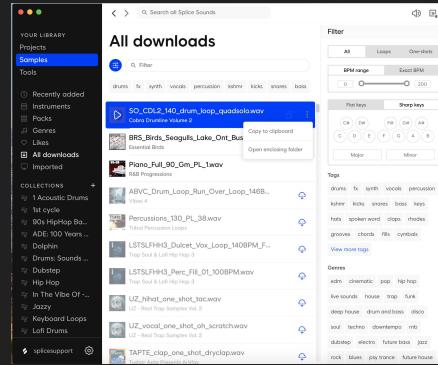
Music: Sources

- Using existing music
 - Loop libraries (Splice, Loopmasters)
 - Asset stores (itch.io, Unity)
 - Ask musicians directly! (Bandcamp, SoundCloud)
- Making your own music
 - DAWs
 - Basic properties
 - Tempo
 - Key
 - Audio vs. MIDI input

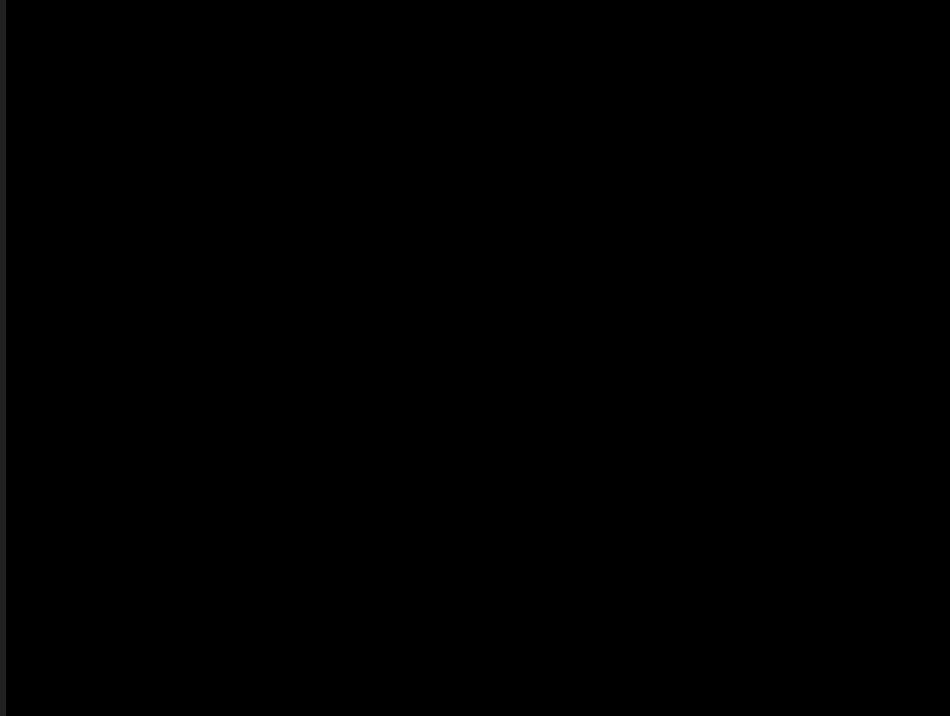


Music: Sources

- Using existing music
 - Loop libraries (Splice, Loopmasters)
 - Asset stores (itch.io, Unity)
 - Ask musicians directly! (Bandcamp, SoundCloud)
- Making your own music
 - DAWs
 - Basic properties
 - Tempo
 - Key
 - Audio vs. MIDI input
 - Exporting



Music Demo: Making Your Own



<https://www.bandlab.com/usdivad/humble-pie>

Music: Interactive Music Approaches

Music: Interactive Music Approaches

- Looping

Music: Interactive Music Approaches

- Looping
- Vertical layering

Music: Interactive Music Approaches

- Looping
- Vertical layering
- Horizontal resequencing

Music: Interactive Music Approaches

- Looping
- Vertical layering
- Horizontal resequencing
- Algorithmic composition

Interactive Music: Looping



My Bearer and I
<https://usdivad.itch.io/my-bearer-and-i>

Interactive Music: Looping

- Export tracks as loops



My Bearer and I
<https://usdivad.itch.io/my-bearer-and-i>

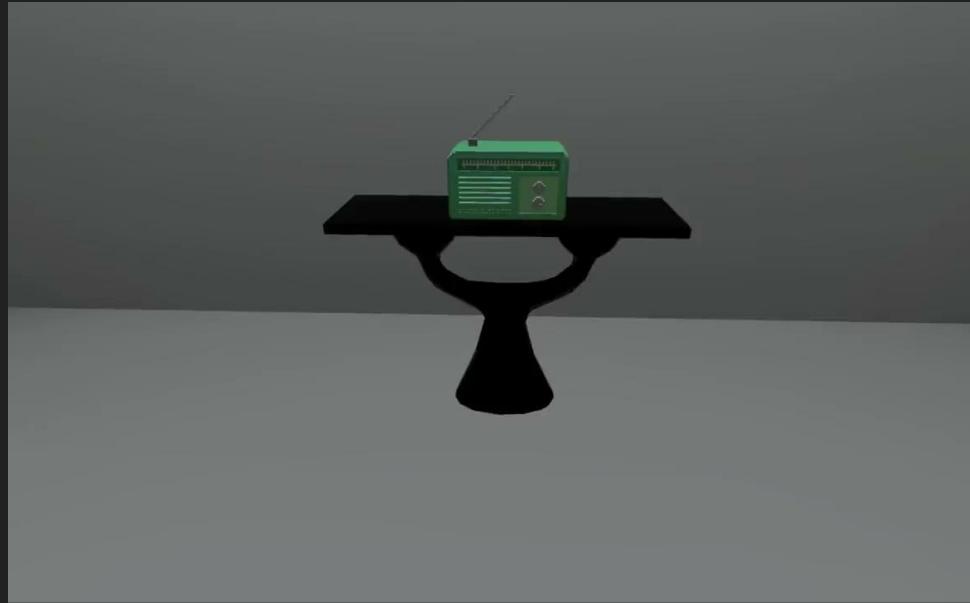
Interactive Music: Looping

- Export tracks as loops
- Head-loop-tail



My Bearer and I
<https://usdivad.itch.io/my-bearer-and-i>

Interactive Music: Vertical Layering

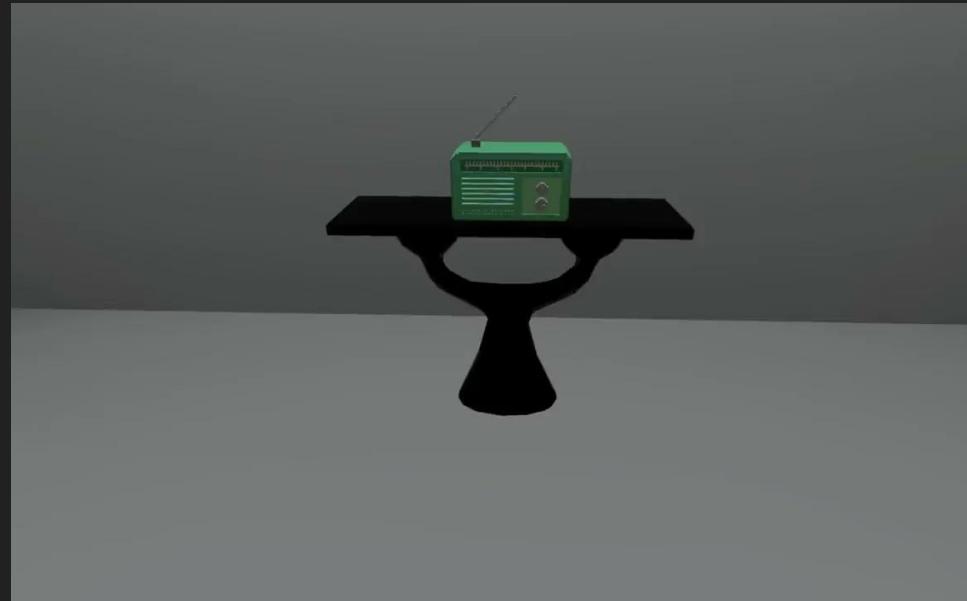


We All Disappear

<https://usdivad.itch.io/we-all-disappear>

Interactive Music: Vertical Layering

- Layer 1: Vocals

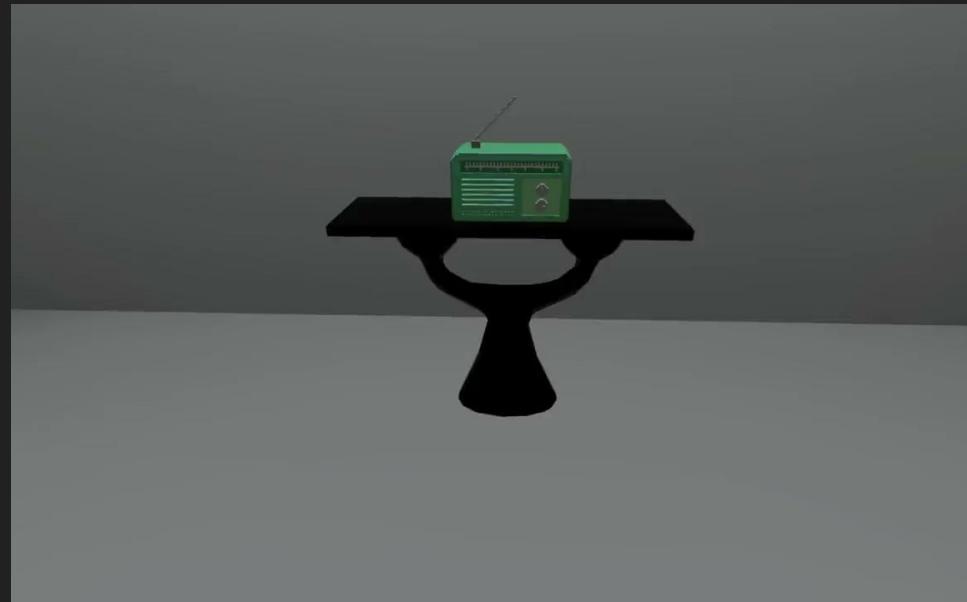


We All Disappear

<https://usdivad.itch.io/we-all-disappear>

Interactive Music: Vertical Layering

- Layer 1: Vocals
- Layer 2: Drums + bass

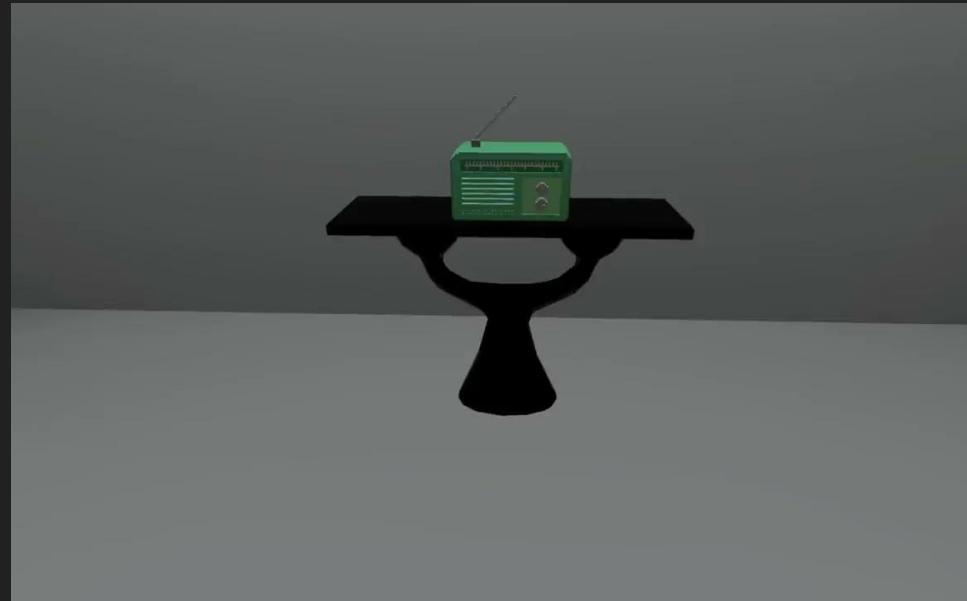


We All Disappear

<https://usdivad.itch.io/we-all-disappear>

Interactive Music: Vertical Layering

- Layer 1: Vocals
- Layer 2: Drums + bass
- Layer 3: Guitars



We All Disappear

<https://usdivad.itch.io/we-all-disappear>

Interactive Music: Horizontal Resequencing



Yi and the Thousand Moons

https://store.steampowered.com/app/701730/Yi_and_the_Thousands_Moons/

Interactive Music: Horizontal Resequencing

- Verse 1
 - Vocals start



Yi and the Thousand Moons

https://store.steampowered.com/app/701730/Yi_and_the_Thousands_Moons/

Interactive Music: Horizontal Resequencing

- Verse 1
 - Vocals start
- Instrumental interlude
 - Vocals stop
 - Violin starts
 - Keyboard starts



Yi and the Thousand Moons

https://store.steampowered.com/app/701730/Yi_and_the_Thousands_Moons/

Interactive Music: Horizontal Resequencing

- Verse 1
 - Vocals start
- Instrumental interlude
 - Vocals stop
 - Violin starts
 - Keyboard starts
- Verse 2
 - Keyboard stops
 - Vocals start



Yi and the Thousand Moons

https://store.steampowered.com/app/701730/Yi_and_the_Thousands_Moons/

Interactive Music: Horizontal Resequencing

- Verse 1
 - Vocals start
- Instrumental interlude
 - Vocals stop
 - Violin starts
 - Keyboard starts
- Verse 2
 - Keyboard stops
 - Vocals start
- Chorus 1
 - Guitar starts
 - Keyboard starts



Yi and the Thousand Moons

https://store.steampowered.com/app/701730/Yi_and_the_Thousands_Moons/

Interactive Music: Horizontal Resequencing

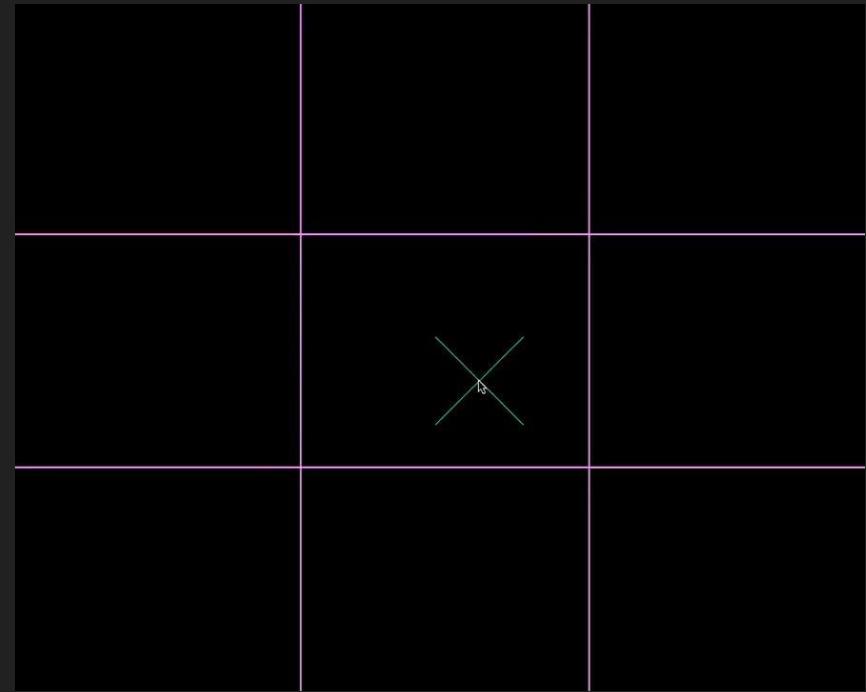
- Verse 1
 - Vocals start
- Instrumental interlude
 - Vocals stop
 - Violin starts
 - Keyboard starts
- Verse 2
 - Keyboard stops
 - Vocals start
- Chorus 1
 - Guitar starts
 - Keyboard starts
- Instrumental interlude 2
 - Vocals stop



Yi and the Thousand Moons

https://store.steampowered.com/app/701730/Yi_and_the_Thousands_Moons/

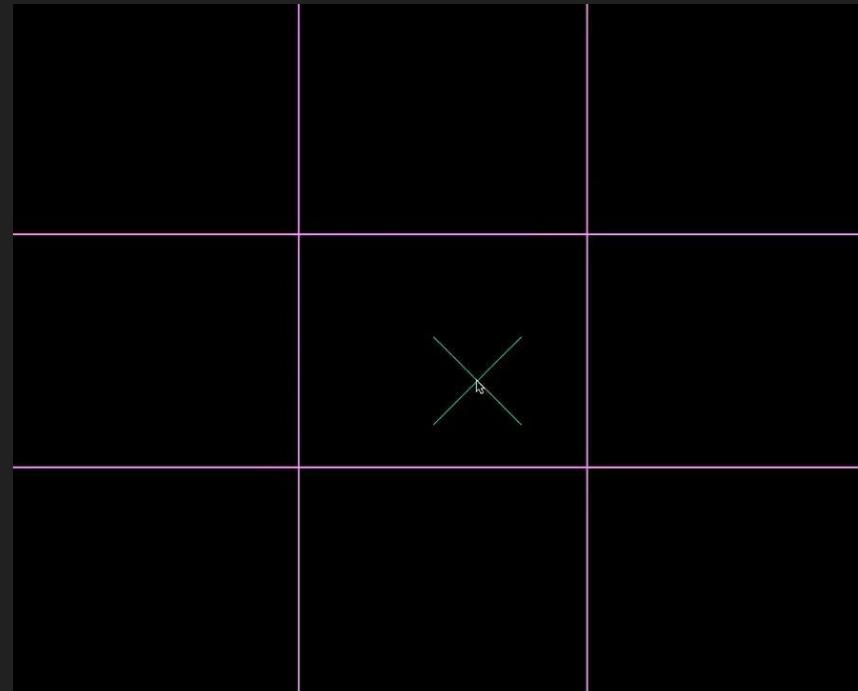
Interactive Music: Algorithmic Composition



<http://usdivad.com/prworkshop2019/>

Interactive Music: Algorithmic Composition

Each time an X/O is placed:

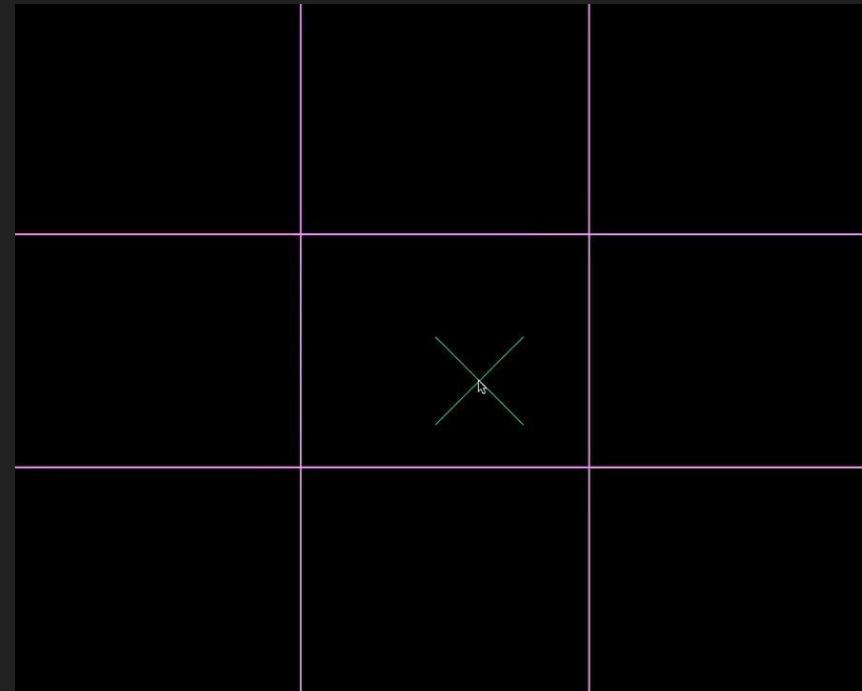


<http://usdivad.com/prworkshop2019/>

Interactive Music: Algorithmic Composition

Each time an X/O is placed:

- Game chooses 4 random notes from the C major scale

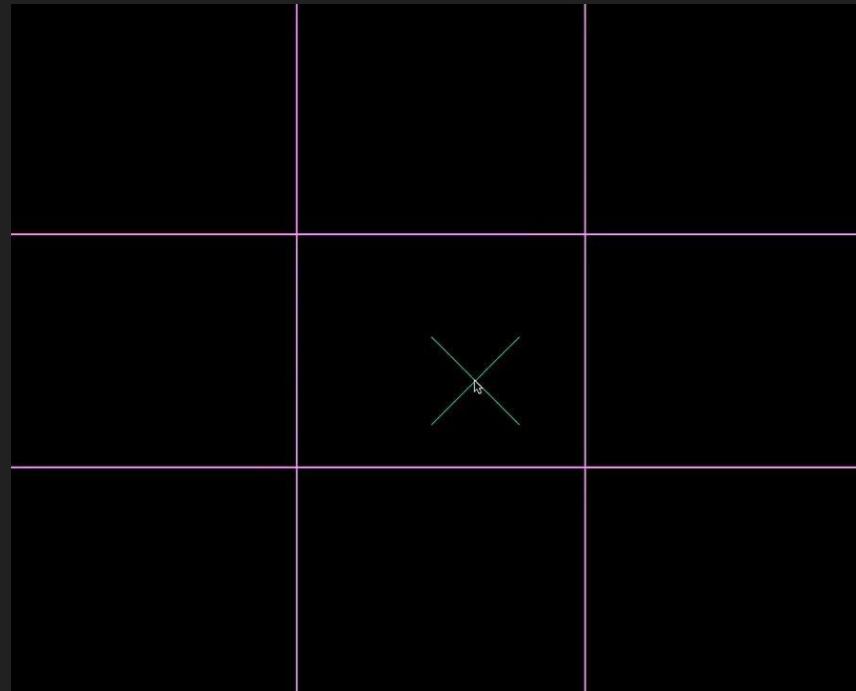


<http://usdivad.com/prworkshop2019/>

Interactive Music: Algorithmic Composition

Each time an X/O is placed:

- Game chooses 4 random notes from the C major scale
- Foreground layer plays those 4 notes as a chord

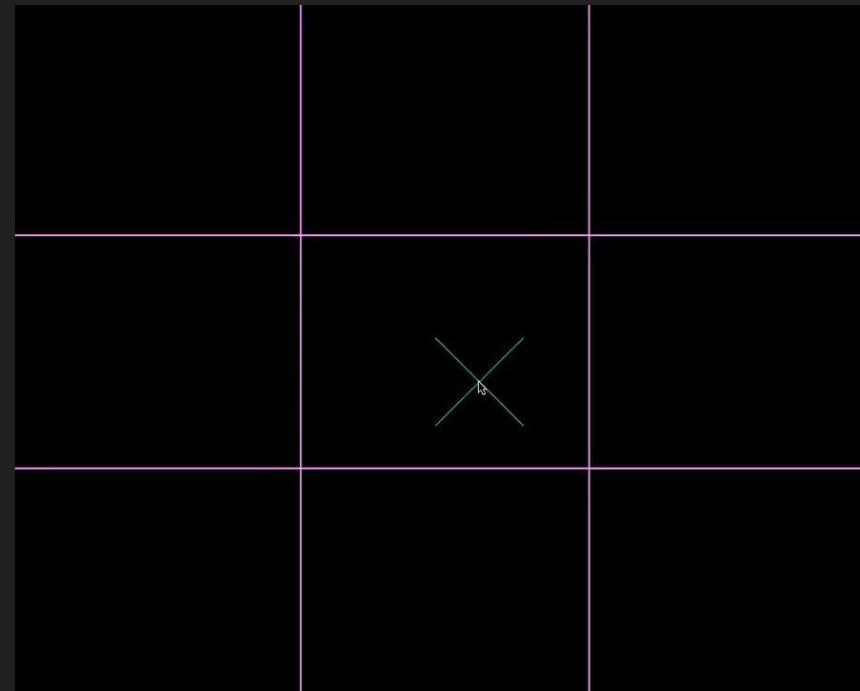


<http://usdivad.com/prworkshop2019/>

Interactive Music: Algorithmic Composition

Each time an X/O is placed:

- Game chooses 4 random notes from the C major scale
- Foreground layer plays those 4 notes as a chord
- Background layer randomly plays 1 of those notes



<http://usdivad.com/prworkshop2019/>

Music: Mixing

- How music and SFX interact sonically
- Spatialization: diegetic vs. non-diegetic
- Dynamic mixing

Music: Musical Game Mechanics

Music: Musical Game Mechanics

- Rhythm

Music: Musical Game Mechanics

- Rhythm
- Melody

Music: Musical Game Mechanics

- Rhythm
- Melody
- Lyrics

Music: Musical Game Mechanics

- Rhythm
- Melody
- Lyrics
- ++

Musical Mechanics: Rhythm



Trolls Music Stars

<https://thinker-tinker.com/pages/trolls-music-stars>

Musical Mechanics: Melody/Harmony



One Hand Clapping

<https://handy-games.com/en/games/one-hand-clapping/>

Musical Mechanics: Lyrics



Evergreen Blues

https://store.steampowered.com/app/1131870/Evergreen_Blues/

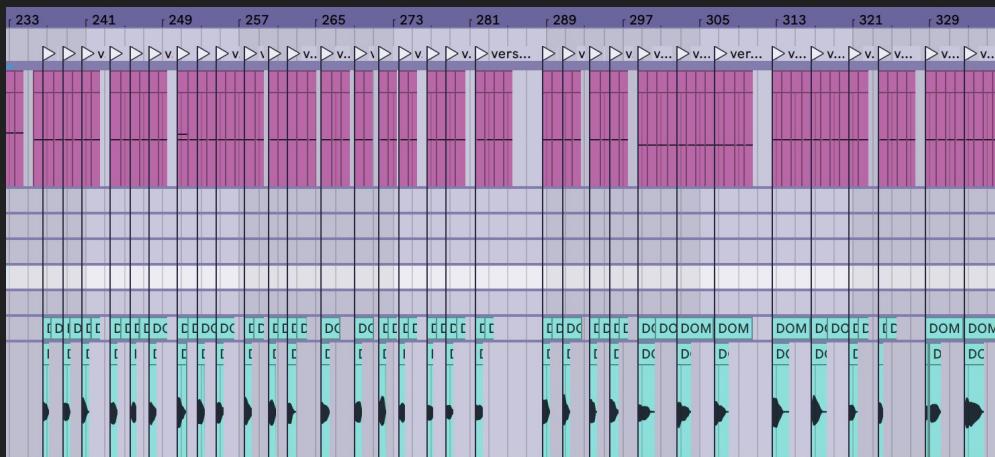
Dialogue



Dialogue

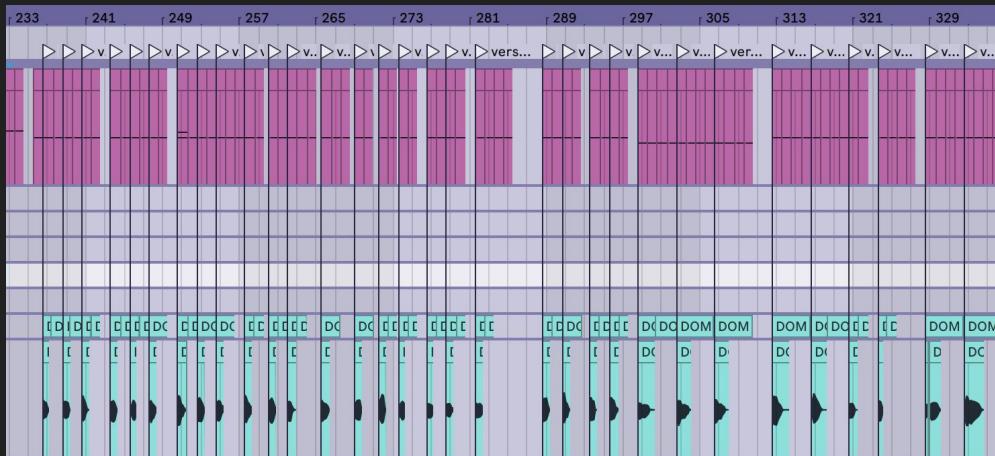
Dialogue

- Directing, editing



Dialogue

- Directing, editing
- Timing and game interactions



```

  * But Id
106 # musicPlayEvent: verse3_build, 1.0
107 # object: instantiateObject, TreeBuild1, TreeBuildEffect, -5, -5.79, 25.9, 0.8
108 * Burn
109 # musicPlayEvent: verse3_burn, 1.0
110
111 # object: instantiateObject, Fire1, FireEffect, -6.6, -11.68, 36.57, -90.0
112 # object: instantiateObject, Fire2, FireEffect, -9.52, -3.7, 36.21, -90.0
113 # object: instantiateObject, Fire3, FireEffect, -5.4, -4, 36.75, -90.0
114 # object: instantiateObject, Fire4, FireEffect, 9.4, -18.49, 35.95, -90.0
115
116 # dialogue: hideUIForDuration, 1.0
117
118 * the
119 # musicPlayEvent: verse3_the, 1.0
120 * a
121 # musicPlayEvent: verse3_a, 1.0
122
123 # dialogue: hideUIForDuration, 1.0
124
125 # trigger: moveCharacterToTarget, PlayerComm3Stage1, PlayerComm3Stage1Verse3GoTarget
126 .5, false, true, true
127
128 # object: hideObject, mouse
129 # object: scaledObject, HandheldIcon, 0.005000002, 0.005000002, 0.005000002
130
131 * silver screen
132 # musicPlayEvent: verse3_silverscreen, 1.0
133 # trigger: moveCharacterToTarget, BannerSilverL, ChengyuBannerVerse3Target, 0.0
134 .false, false, false
135 # trigger: moveCharacterToTarget, BannerSilverR, ChengyuBannerVerse3Target, 0.0
136 .false, false, false
137
138 * smokened screen
139 # musicPlayEvent: verse3_smokedscreen, 1.0
140 # trigger: moveCharacterToTarget, BannerSmokeL, ChengyuBannerVerse3Target, 0.0
141 .false, false, false
142 # trigger: moveCharacterToTarget, BannerSmokeR, ChengyuBannerVerse3Target, 0.0
143 .false, false, false
144
145 * sacred screen
146 # musicPlayEvent: verse3_sacredscreen, 1.0
147 # trigger: moveCharacterToTarget, BannerSacredL, ChengyuBannerVerse3Target, 0.0
148 .false, false, false
149 # trigger: moveCharacterToTarget, BannerSacredR, ChengyuBannerVerse3Target, 0.0
150 .false, false, false

```

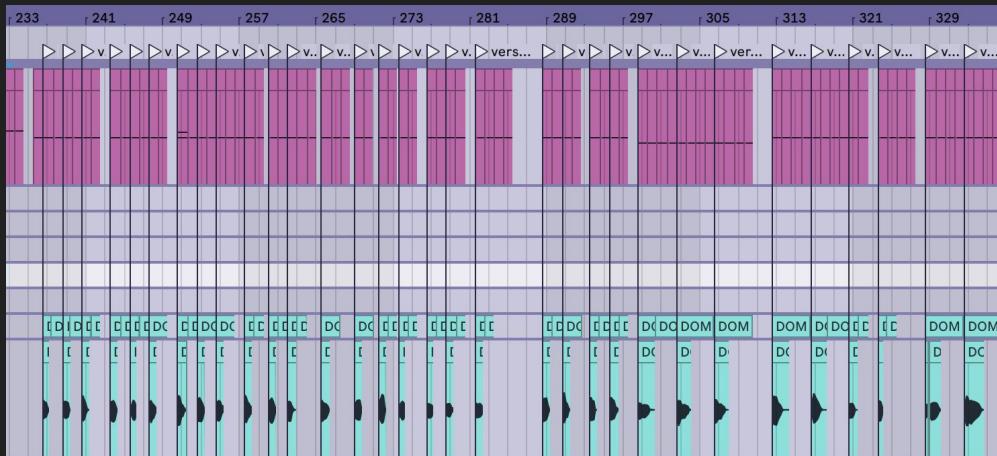
```

170 * Sights
171 # musicPlayEvent: verse3_sights, 1.0
172 # object: scaleObject, EyeMural, 0.6, 0.5, 1.0
173
174 -evergreen_verse3_sightsounds = "sights"
175
176 // ** go
177 // *** unseen
178 // ** stop
179 // *** serene
180
181 * Sounds
182 # musicPlayEvent: verse3_sounds, 1.0
183 # object: scaleObject, EarMural, 0.6, 0.5, 1.0
184
185 -evergreen_verse3_sightsounds = "sounds"
186 // ** pause
187 // *** ringing
188 // ** play
189 // *** singing
190
191
192
193 # dialogue: hideUIForDuration, 1.0
194
195 # object: hideObject, prforest_126x100
196
197 * {evergreen_verse3_sightsounds == "sights"} go
198 # musicPlayEvent: verse3_go, 1.0
199 # trigger: moveCharacterToTarget, PlayerComm3Stage2,
200 PlayerComm3Stage2Verse3Target, 5.0, false, true, true
201 * {evergreen_verse3_sightsounds == "sights"} stay
202 # musicPlayEvent: verse3_stay, 1.0
203 * {evergreen_verse3_sightsounds == "sounds"} play
204 # musicPlayEvent: verse3_play, 1.0
205 # trigger: moveCharacterToTarget, PlayerComm3Stage2,
206 PlayerComm3Stage2Verse3Target, 5.0, false, true, true
207 * {evergreen_verse3_sightsounds == "sounds"} pause
208 # musicPlayEvent: verse3_pause, 1.0

```

Dialogue

- Directing, editing
- Timing and game interactions
- Interrupts



```

106 * ButId
107 # musicPlayEvent: verse3_build, 1.0
108 # object: instantiateObject, TreeBuild1, TreeBuildEffect, -5, -5.79, 25.9, 0.8
109 * Burn
110 # musicPlayEvent: verse3_burn, 1.0
111
112 # object: instantiateObject, Fire1, FireEffect, -6.6, -11.68, 36.57, -90.0
113 # object: instantiateObject, Fire2, FireEffect, -9.52, -3.7, 36.21, -90.0
114 # object: instantiateObject, Fire3, FireEffect, -5.4, -4, 36.75, -90.0
115 # object: instantiateObject, Fire4, FireEffect, 9.4, -18.49, 35.95, -90.0
116
117 # dialogue: hideUIForDuration, 1.0
118
119 * the
120 # musicPlayEvent: verse3_the, 1.0
121 * a
122 # musicPlayEvent: verse3_a, 1.0
123
124 # dialogue: hideUIForDuration, 1.0
125
126 # trigger: moveCharacterToTarget, PlayerComm3Stage1, PlayerComm3Stage1Verse3GoTarget
127 .5, false, true, true
128 # object: hideObject, moon
129 # object: scaledObject, HandheldMoon, 0.005000002, 0.005000002, 0.005000002
130
131 * silver screen
132 # musicPlayEvent: verse3_silverscreen, 1.0
133 # trigger: moveCharacterToTarget, BannerSilverL, ChengyuBannerVerse3Target, 0.0
134 .false, false, false
135 # trigger: moveCharacterToTarget, BannerSilverR, ChengyuBannerVerse3Target, 0.0
136 .false, false, false
137
138 * smoked screen
139 # musicPlayEvent: verse3_smokedscreen, 1.0
140 # trigger: moveCharacterToTarget, BannerSmokeL, ChengyuBannerVerse3Target, 0.0
141 .false, false, false
142 # trigger: moveCharacterToTarget, BannerSmokeR, ChengyuBannerVerse3Target, 0.0
143 .false, false, false
144
145 * sacred screen
146 # musicPlayEvent: verse3_sacredscreen, 1.0
147 # trigger: moveCharacterToTarget, BannerSacredL, ChengyuBannerVerse3Target, 0.0
148 .false, false, false
149 # trigger: moveCharacterToTarget, BannerSacredR, ChengyuBannerVerse3Target, 0.0
150 .false, false, false

```

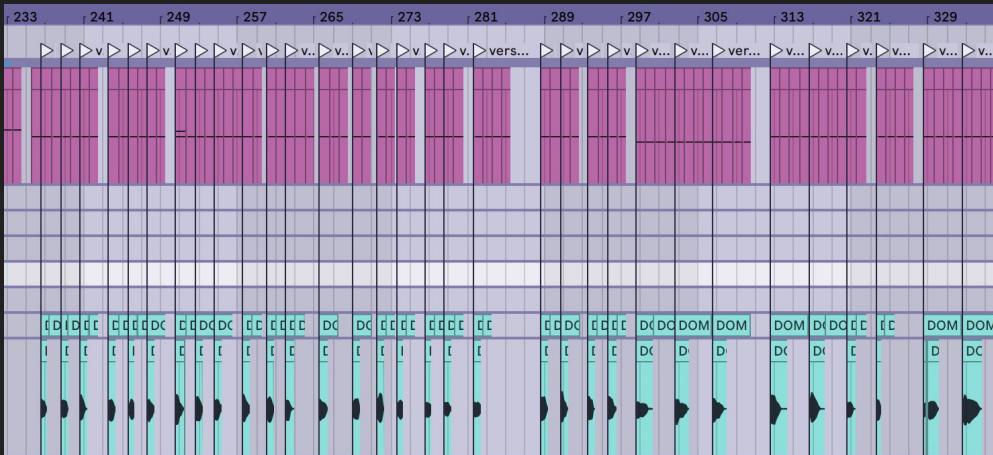
```

170 * Sights
171 # musicPlayEvent: verse3_sights, 1.0
172 # object: scaleObject, EyeMural, 0.6, 0.5, 1.0
173
174 -evergreen_verse3_sightsorsounds = "sights"
175
176 // ** go
177 // *** unseen
178 // ** stop
179 // *** serene
180
181 * Sounds
182 # musicPlayEvent: verse3_sounds, 1.0
183 # object: scaleObject, EarMural, 0.6, 0.5, 1.0
184
185 -evergreen_verse3_sightsorsounds = "sounds"
186 // ** pause
187 // *** ringing
188 // ** play
189 // *** singing
190
191
192
193 # dialogue: hideUIForDuration, 1.0
194
195 # object: hideObject, prforest_126x100
196
197 * {evergreen_verse3_sightsorsounds == "sights"} go
198 # musicPlayEvent: verse3_go, 1.0
199 # trigger: moveCharacterToTarget, PlayerComm3Stage2,
200 PlayerComm3Stage2Verse3Target, 5.0, false, true, true
201 * {evergreen_verse3_sightsorsounds == "sights"} stay
202 # musicPlayEvent: verse3_stay, 1.0
203 * {evergreen_verse3_sightsorsounds == "sounds"} play
204 # musicPlayEvent: verse3_play, 1.0
205 # trigger: moveCharacterToTarget, PlayerComm3Stage2,
206 PlayerComm3Stage2Verse3Target, 5.0, false, true, true
207 * {evergreen_verse3_sightsorsounds == "sounds"} pause
208 # musicPlayEvent: verse3_pause, 1.0

```

Dialogue

- Directing, editing
- Timing and game interactions
- Interrupts
- Mixing (ducking, spatialization)



```

  * But Id
106 # musicPlayEvent: verse3_build, 1.0
107 # object: instantiateObject, TreeBuild1, TreeBuildEffect, -5, -5.79, 25.9, 0.8
108 * Burn
109 # musicPlayEvent: verse3_burn, 1.0
110 -
111 # object: instantiateObject, Fire1, FireEffect, -6.6, -11.68, 36.57, -90.0
112 # object: instantiateObject, Fire2, FireEffect, -9.52, -3.7, 36.21, -90.0
113 # object: instantiateObject, Fire3, FireEffect, -5.4, -4, 36.75, -90.0
114 # object: instantiateObject, Fire4, FireEffect, 9.4, -18.49, 35.95, -90.0
115 -
116 # dialogue: hideUIForDuration, 1.0
117 *
118 * the
119 # musicPlayEvent: verse3_the, 1.0
120 * a
121 # musicPlayEvent: verse3_a, 1.0
122 -
123 # dialogue: hideUIForDuration, 1.0
124 *
125 # trigger: moveCharacterToTarget, PlayerCommStage1, PlayerCommStage1Verse3GoTarget
126 # object: scaledObject, HandheldIcon, 0.005000002, 0.005000002, 0.005000002
127 -
128 * silver screen
129 # musicPlayEvent: verse3_silverscreen, 1.0
130 # trigger: moveCharacterToTarget, BannerSilverL, ChengyuBannerVerse3Target, 0.0
131 . false, false, false
132 # trigger: moveCharacterToTarget, BannerSilverR, ChengyuBannerVerse3Target, 0.0
133 . false, false, false
134 * smoked screen
135 # musicPlayEvent: verse3_smokedscreen, 1.0
136 # trigger: moveCharacterToTarget, BannerSmokeL, ChengyuBannerVerse3Target, 0.0
137 . false, false, false
138 # trigger: moveCharacterToTarget, BannerSmokeR, ChengyuBannerVerse3Target, 0.0
139 . false, false, false
140 -
141 * sacred screen
142 # musicPlayEvent: verse3_sacredscreen, 1.0
143 # trigger: moveCharacterToTarget, BannerSacredL, ChengyuBannerVerse3Target, 0.0
144 . false, false, false
145 # trigger: moveCharacterToTarget, BannerSacredR, ChengyuBannerVerse3Target, 0.0
146 . false, false, false

```

```

  * Sights
170 # musicPlayEvent: verse3_sights, 1.0
171 # object: scaledObject, EyeMural, 0.6, 0.5, 1.0
172 -
173 -evergreen_verse3_sightsorsounds = "sights"
174 -
175 // ** go
176 // *** unseen
177 // ** stop
178 // *** serene
179 -
180 * Sounds
181 # musicPlayEvent: verse3_sounds, 1.0
182 # object: scaledObject, EarMural, 0.6, 0.5, 1.0
183 -
184 -evergreen_verse3_sightsorsounds = "sounds"
185 // ** pause
186 // *** ringing
187 // ** play
188 // *** singing
189 -
190 -
191 -
192 -
193 # dialogue: hideUIForDuration, 1.0
194 -
195 # object: hideObject, prforest_126x100
196 -
197 * {evergreen_verse3_sightsorsounds == "sights"} go
198 # musicPlayEvent: verse3_go, 1.0
199 # trigger: moveCharacterToTarget, PlayerCommStage2,
200 PlayerCommStage2Verse3Target, 5.0, false, true, true
201 * {evergreen_verse3_sightsorsounds == "sights"} stay
202 # musicPlayEvent: verse3_stay, 1.0
203 * {evergreen_verse3_sightsorsounds == "sounds"} play
204 # musicPlayEvent: verse3_play, 1.0
205 # trigger: moveCharacterToTarget, PlayerCommStage2,
206 PlayerCommStage2Verse3Target, 5.0, false, true, true
207 * {evergreen_verse3_sightsorsounds == "sounds"} pause
208 # musicPlayEvent: verse3_pause, 1.0

```

Dialogue

- Procedurally generated audio for dialogue



<https://github.com/usdivad/Animalese>

Implementation



Implementation: Philosophy

Implementation: Philosophy

- Making the game make sound!

Implementation: Philosophy

- Making the game make sound!
- Telling the game what audio to play and when, where, how to play it

Implementation: Philosophy

- Making the game make sound!
- Telling the game what audio to play and when, where, how to play it
 - **What:** the sound, event, layer/section of music, audio effect...

Implementation: Philosophy

- Making the game make sound!
- Telling the game what audio to play and when, where, how to play it
 - **What:** the sound, event, layer/section of music, audio effect...
 - **When:** pre-scheduling, specific beat/bar, synced to animation/gameplay events...

Implementation: Philosophy

- Making the game make sound!
- Telling the game what audio to play and when, where, how to play it
 - **What:** the sound, event, layer/section of music, audio effect...
 - **When:** pre-scheduling, specific beat/bar, synced to animation/gameplay events...
 - **Where:** in-game position, position within speaker mix...

Implementation: Philosophy

- Making the game make sound!
- Telling the game what audio to play and when, where, how to play it
 - **What:** the sound, event, layer/section of music, audio effect...
 - **When:** pre-scheduling, specific beat/bar, synced to animation/gameplay events...
 - **Where:** in-game position, position within speaker mix...
 - **How:** one-shot, loop, voice limiting/prioritization, in-memory vs. streaming, sample rate...

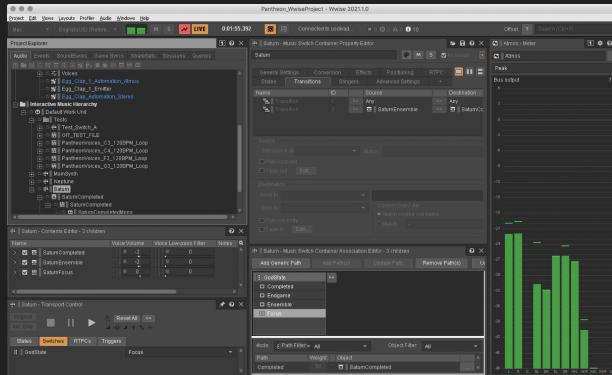
Implementation: Tools and Middleware

Implementation: Tools and Middleware

- Game engine built-in audio features (Unity, Unreal Engine, Godot)

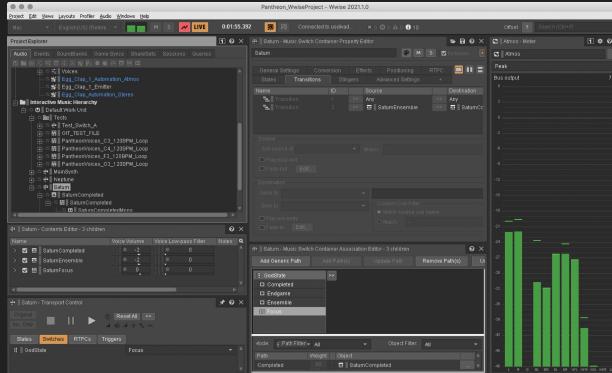
Implementation: Tools and Middleware

- Game engine built-in audio features (Unity, Unreal Engine, Godot)
- Middleware (Wwise, FMOD)



Implementation: Tools and Middleware

- Game engine built-in audio features (Unity, Unreal Engine, Godot)
- Middleware (Wwise, FMOD)
- Real-time audio tools (Pure Data, Max/MSP, RTcmix, ChucK)



Implementation: Tasks and Techniques

Implementation: Tasks and Techniques

- Spatialization
 - 2D, 3D
 - Position, orientation

Implementation: Tasks and Techniques

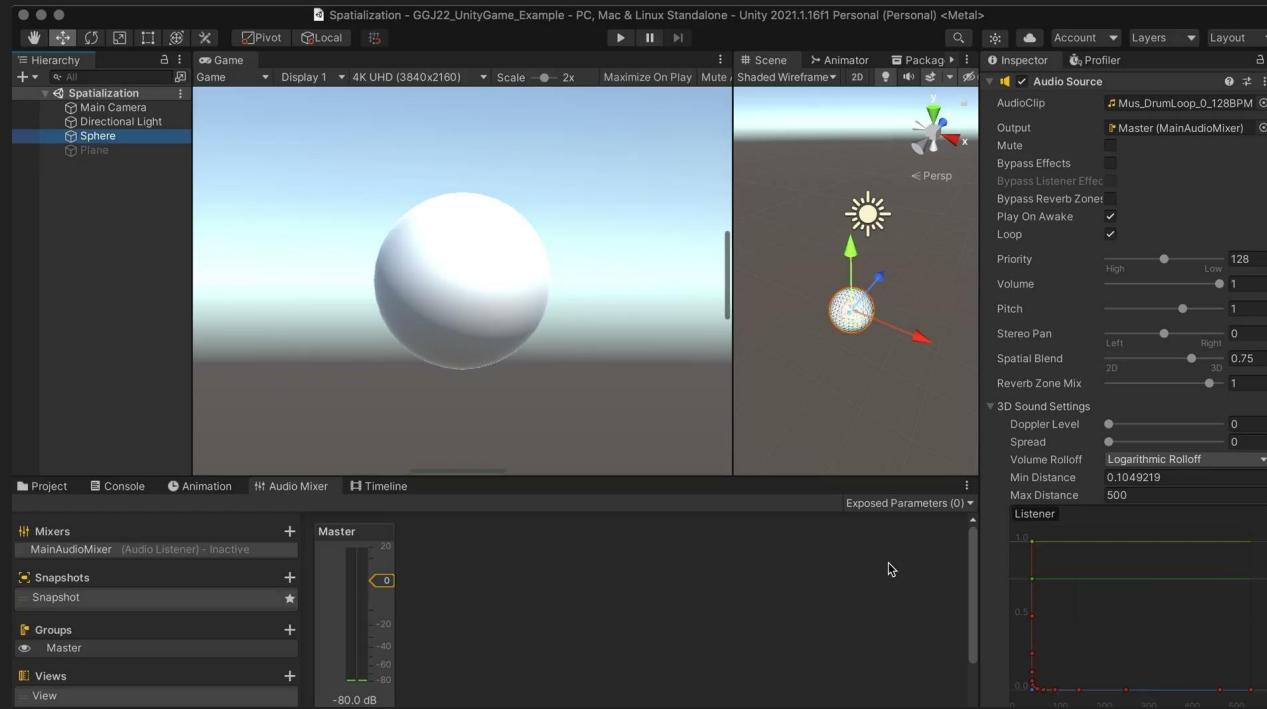
- Spatialization
 - 2D, 3D
 - Position, orientation
- Randomization
 - Pitch
 - Triggered samples

Implementation: Tasks and Techniques

- Spatialization
 - 2D, 3D
 - Position, orientation
- Randomization
 - Pitch
 - Triggered samples
- Real-time effects
 - Reverb
 - EQ/filtering

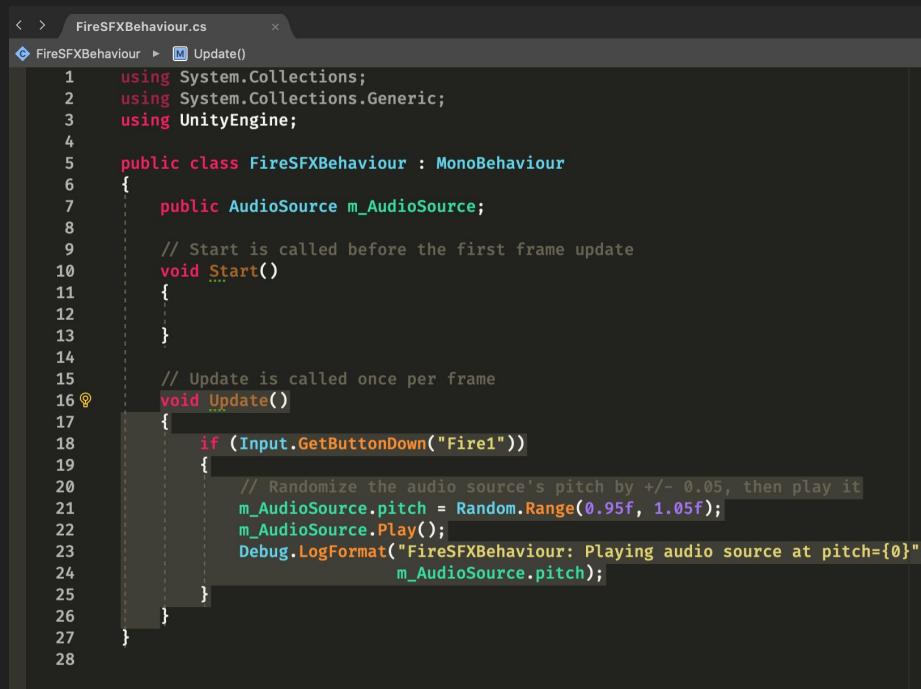
Implementation Demo: Spatialization

- Spatializing a looping sound in 3D



Implementation Demo: Randomization

- Randomizing pitch of a one-shot to create variety

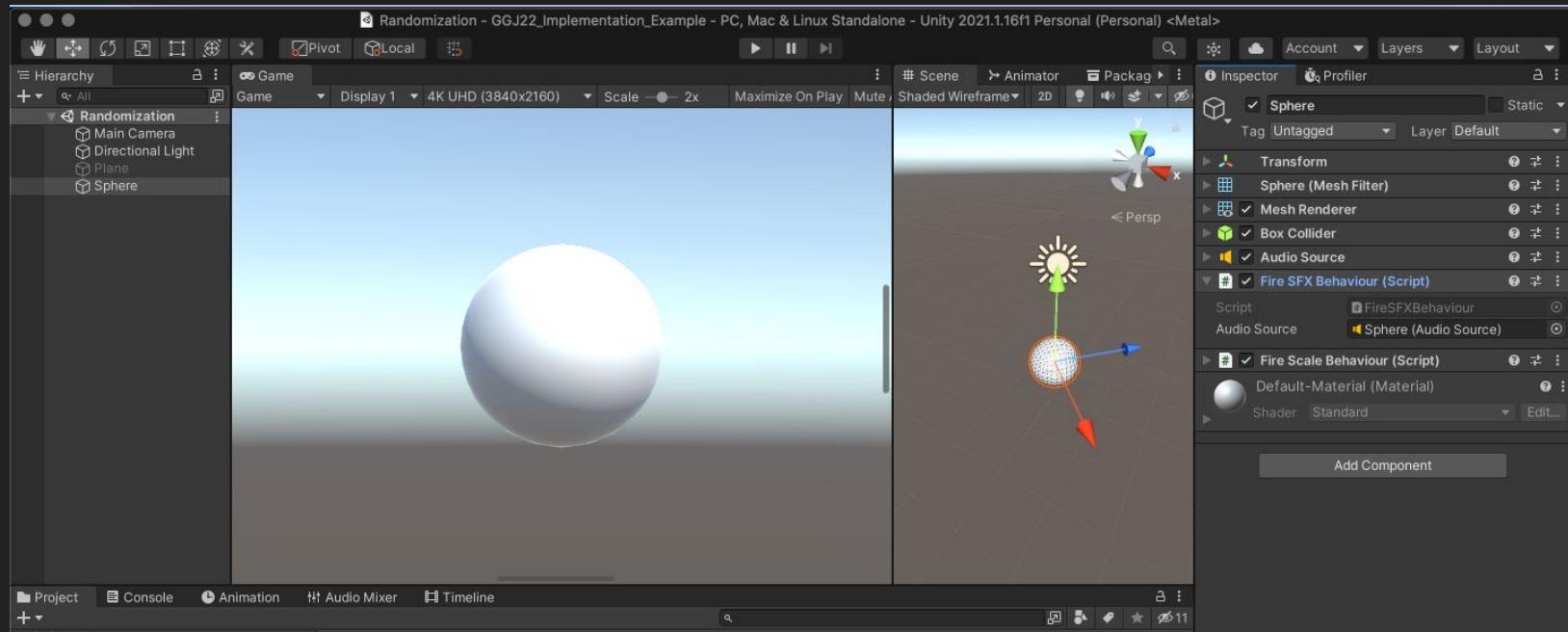


```
FireSFXBehaviour.cs
FireSFXBehaviour > [M] Update()

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class FireSFXBehaviour : MonoBehaviour
6  {
7      public AudioSource m_AudioSource;
8
9      // Start is called before the first frame update
10     void Start()
11     {
12     }
13
14
15     // Update is called once per frame
16     void Update()
17     {
18         if (Input.GetButtonDown("Fire1"))
19         {
20             // Randomize the audio source's pitch by +/- 0.05, then play it
21             m_AudioSource.pitch = Random.Range(0.95f, 1.05f);
22             m_AudioSource.Play();
23             Debug.LogFormat("FireSFXBehaviour: Playing audio source at pitch={0}",
24                             m_AudioSource.pitch);
25         }
26     }
27 }
28
```

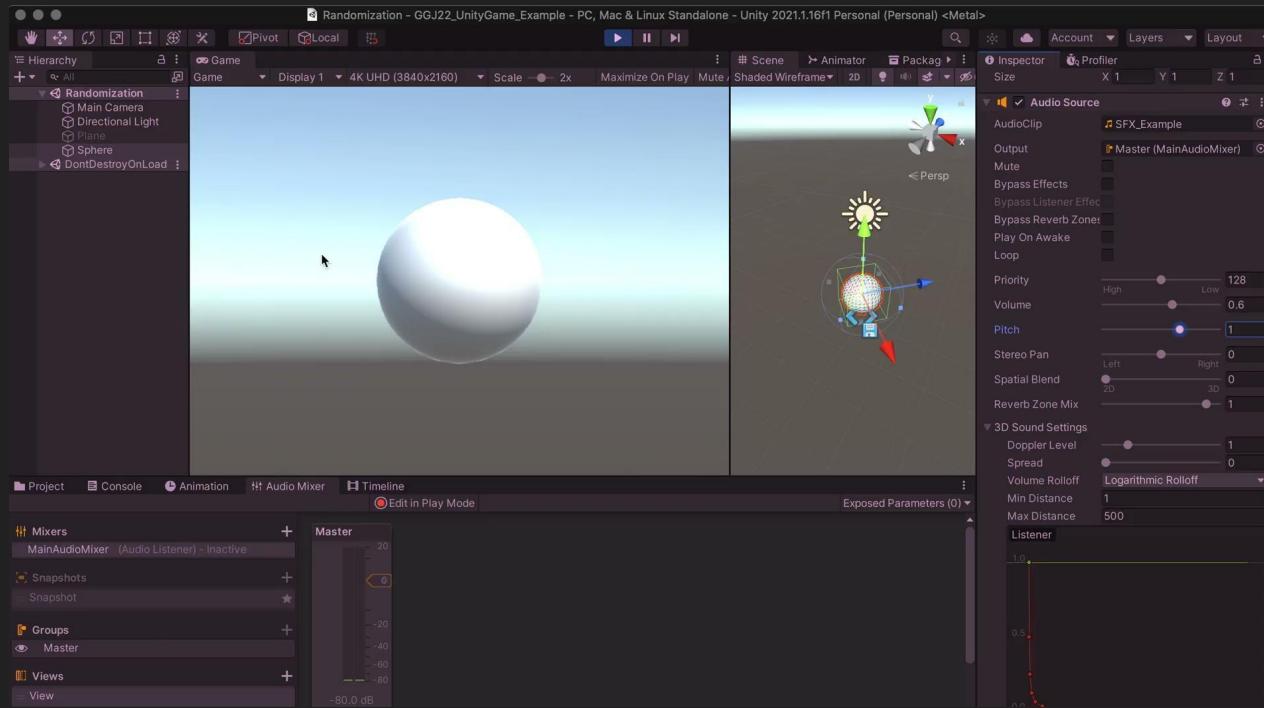
Implementation Demo: Randomization

- Randomizing pitch of a one-shot to create variety



Implementation Demo: Randomization

- Randomizing pitch of a one-shot to create variety



Resources



Resources: SFX

- Free
 - Freesound <https://freesound.org/>
 - Library of Congress <https://www.loc.gov/audio/>
 - Sonniss GameAudioGDC <https://sonniss.com/gameaudioqdc>
- Paid
 - BOOM <https://www.boomlibrary.com/>
 - Tonsturm <https://tonsturm.com/>
 - Soundly <https://getsoundly.com/>

Resources: DAWs

- Free
 - Audacity <https://www.audacityteam.org/>
 - Tracktion Waveform <https://www.tracktion.com/products/waveform-free>
 - BandLab <https://www.bandlab.com/>
 - GarageBand <https://www.apple.com/mac/garageband/>
- Paid
 - Pro Tools <https://www.avid.com/pro-tools>
 - REAPER <https://www.reaper.fm/>
 - Logic Pro <https://www.apple.com/logic-pro/>
 - Ableton Live <https://www.ableton.com/en/live/>

Resources: Music

- Libraries
 - Splice <https://splice.com/>
 - Loopmasters <https://www.loopmasters.com/>
 - Library of Congress <https://www.loc.gov/audio/>
- Asset stores
 - itch.io <https://itch.io/game-assets/tag-music>
 - Unity <https://assetstore.unity.com/audio/music>
- Ask musicians directly
 - Bandcamp <https://bandcamp.com/>
 - SoundCloud <https://soundcloud.com/>

Resources: Implementation

- Game engines
 - Unity <https://docs.unity3d.com/Manual/AudioOverview.html>
 - Unreal <https://docs.unrealengine.com/4.27/en-US/WorkingWithAudio/>
 - Godot <https://docs.godotengine.org/en/stable/tutorials/audio/index.html>
- Middleware
 - Wwise <https://wwwaudiokinetic.com/products/wwise/>
 - FMOD <https://www.fmod.com/>
 - Elias <https://eliassoftware.com/>
 - OpenAL <https://www.openal.org/>
- Real-time audio environments
 - Pure Data <https://puredata.info/> + libpd <https://github.com/libpd/libpd>
 - RTcmix <http://rtcmix.org/> + uRTcmix <http://rtcmix.org/urtcmix/>
 - Chuck <https://chuck.stanford.edu/> + Chunity <https://chuck.stanford.edu/chunity/>
 - Tone.js <https://tonejs.github.io/>

Resources: Learning

- Books
 - *Game Sound* by Karen Collins <https://mitpress.mit.edu/books/game-sound>
 - *Beep to Boom* by Simon N Goodwin <https://www.taylorfrancis.com/books/mono/10.4324/9781351005548/beep-boom-simon-goodwin>
- Conferences (incl. past talks)
 - GDC <https://qdconf.com/conference/game-audio>
 - GDC Vault
https://www.qdcvault.com/search.php#&conference_id=&category=free&firstfocus=&keyword=game+audio
 - GameSoundCon <https://www.gamesoundcon.com/>
- Twitter
 - #gameaudio <https://twitter.com/search?q=%23gameaudio>

Thank you!

David Su
@usdivad

<https://github.com/usdivad/GGJ22-GameAudio>