140

Sonic College 2018 Jakob Schmid

140

140

Jeppe Carlsen (design, programming) Niels Fyrst, Andreas Peitersen (visual design) Jakob Schmid (audio)

Developed as hobby project over 3 years





140

IGF award 2013 Excellence in Audio - Honorable mention: Technical Excellence

Spilprisen 2014 Sound of the Year

Nordic Game Award 2014 Artistic Achievement



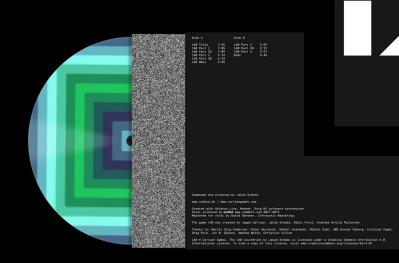
140 Soundtrack

Vinyl

• iam8bit

Digital

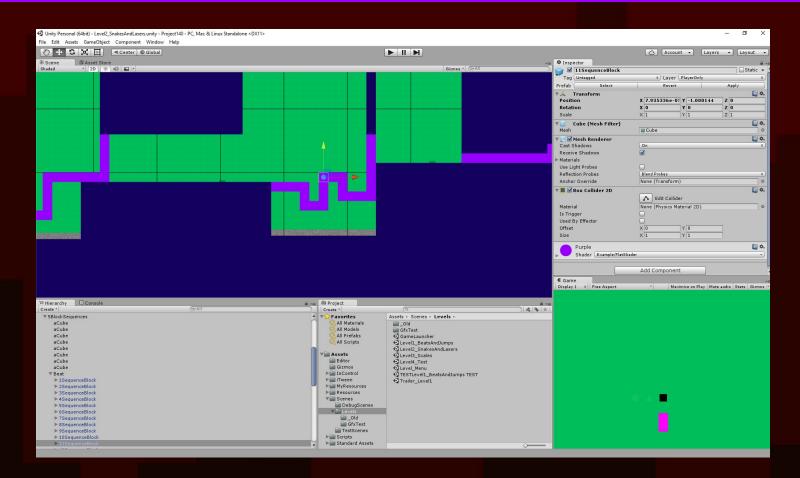
- Steam
- GOG.com
- Spotify
- iTunes
- Amazon



Includes: 140 Vinyl Soundtrack Digital Soundtrack Music by: Limited Edition of 1400 Steam Code for Full Game Jakob Schmid

Carlsen Games iam(3)bit iam8bit.com

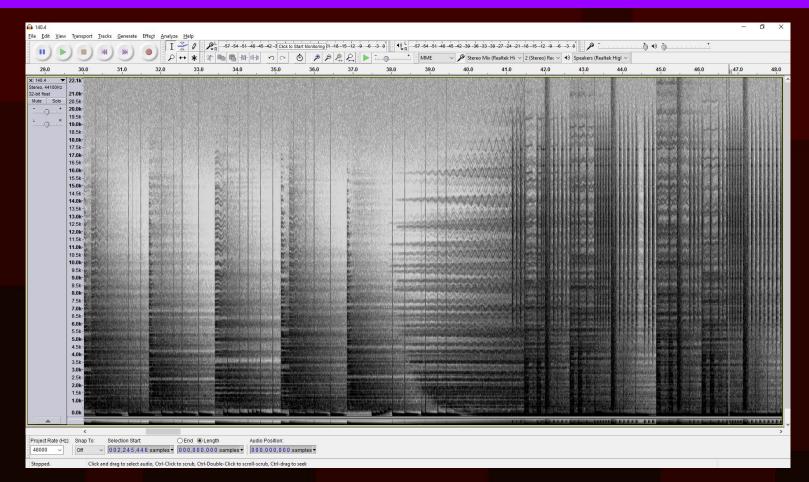
Developed in Unity 3



Ableton Live

Ura level3 [live9-140] -	Ableton Live 9 Suite																				- 0 ×
File Edit Create	View Options Help																				
TAP 140.00 IIII IIII	4 / 4 0 • • 2 Bars •					31.2.1		• •° + 1	O NEW					3.1.	1~0	<u>∽</u> 4.	0.0				B KEY MIDI 3% D
Search (Ctrl + F)				_												_	_				
CATEGORIES	Name	1 Drum 🐨	2 Drum) clap ve 🗑	kywhi snri	laserbea	cute drums	real drums 🖲	sub	9 level0	theme	I2-bass	I2-swell	12-JK-noise	12-JK-jump	12-JK-build	DONTUS	bass	acid	acid	Master
♪ Sounds	P Amp																				▷ 13-layer0 [10.0.0 loop]
BB Drums	Audio Effect Rack																				▷ swapper
√ Instruments	Auto Filter		8	8		8	8			8	8	8	8	8			8	8			I3-layer0-hiding_key0 [2.0.0]
- Audio Effects	Auto Pan	0			0										0			0			I3-layer0-hiding_key1 [2.0.0]
+ MIDI Effects	▶ Beat Repeat																				IO-hideykey-DONTBOUNCE
A Max for Live	E Cabinet		-	-							-	-				-		-		-	IO-hideykey-DONTBOUNCE
<} Plug-ins	D 🔲 Chorus	-								0		8			0						layer1
▶ Clips	D Compressor	-														0					layer1-build layer1-lasmover
++ Samples	🕨 🔲 Corpus	key bea			<u> </u>		0											0			layer1-beat full
	🖻 🔚 Dynamic Tube	key bea							ile i	0				0	0						▷ layer1-beat-key
PLACES	🕨 🔜 EQ Eight	beatB			0					0					0						▷ layer1-beat-rest
Packs	EQ Three				-	-		-				-				0	-	1	-		▷ layer1-variation
L User Library	▶ 📄 Erosion																				layer1-key_wheel [1.0.0 loop]
Current Project	▷ 📄 External Audio Effect			▶ beatB]										8						layer1-key [5.0.0 loop]
Livo9-140	🕨 🥅 Filter Delay																				▷ I3-layer2 [8.0.0 loop]
💾 live8-140	▷ 🚍 Flanger				0																▷ layer2-snare
🛅 samples	▷ 📰 Frequency Shifter																				Iayer2-variation [4.0.0 loop]
C delete	🖻 🔚 Gate														▶ //////	8 ▷ 1]			▷ I3-layer2-jump_killa (8.0.0)
2017-01-02-Fractu	▷ 🗐 Glue Compressor																	•			▷ elevator1
Add Folder	🖻 📰 Grain Delay			1																	▷ layer2-key
	▷ 📰 Limitor																			-	D 13-layer3
	▶ 📰 Looper												•		1				0		▷ I3-layer3-claustrophobic [4.0.0 loop]
	Multiband Dynamics															8 > //////					I3-layer3-jump_killa (2.0.0) I3 scale BB
	▷ 🔲 Overdrive	-		-					-		_	_	-	-				-	_	-	
	▶ 🚍 Phaser	M. From	M. From	M. From	MIDI From		MIDI From	MIDI From		MIDI From	M. From		M. From	MIDI From	0		M. From	MIDI From	MIDI From	MIDI From	
	Ping Pong Delay	All Ins 🔻				All Ins V		All Ins 🔻			All Ins 🔻						All Ins 🔻			All Ins	
	▶ 📰 Redux								I All Channe▼										All Channe		
	Resonators	Monitor Auto 🔻	Monitor Auto	Monitor	Monitor	Monitor Auto		Monitor	Monitor	Monitor	Monitor		Monitor Auto	Monitor			Monitor Auto	Monitor	Monitor	Monitor In Auto	Cue Out
	▷ 📰 Reverb	A. To	A. To	A. To	Audio To	A. To	Audio To	Audio To	Audio To	Audio To	A. To	A. To	A. To	Audio To	Audio To	Audio To	A. To	Audio To	Audio To	Audio To	ii 1/2 ▼
	▷ 📰 Saturator	Master v	Master •	Master V	Master •	Master V	Master	Master •	Master v	Master •	Master V	Master v	Master V	Master •	Master	Master V	Master V	Master •	Master •	Master	Master Out
	Simple Delay																				
	▷ 📰 Spectrum	<u> </u>	C <	1	1 ∩ ∎	C 4	1 1	0		(∩) ∎	C 4	C 4	1. 4	(∩) ∎	I ∩ I≤	1 ₼ ∎	C <	⊲		\bigcirc	Ů 1
	D Tuner		_												-				H		
	▷ 📰 Utility	1	2	3	4	5	6	7	8	9	10	11 S	12	13	14	18	21	22	23	24	Solo
	Vinyl Distortion	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	
	▶ Vocoder ▼	0		0	0	0	0	0	0	0	0	0	0	0				0	0	0	3
																				Þ	E
Onum Rect		Instrument 2	ak			🕕 🕞 Ope	rator		-						(11000		Justo Ban		0000	Caturat	
O Drum Rack		Instrument Rad	C.K			Ope Coarse		Level							Upgrade Rate Ame	ount O	Auto Pan		00		
C2	C#2 D2 D#2 0	10	K	ley Vel C	Chain Hide		0 =								0 0		NNN/	NNNNN			
	<u> </u>	A bd and base	2.3	dB C 🚽	4 5 0	48	550	-7.3 dB					32				1111	11111	VV) ()	+ S
G#1	Instrument A#1 B1	click	-9.3	dB C 🚽	4 5 0	Freq	-	Level	Envelope			allator	64 Fi		Freq Res			1000	6.86	dB	
	M > S					C2.00 KH2	: 0.1	V .57 dB		/ Release Tir			Low		102 Hz 📏 1	.83 Am	ount Rat	0	Shape DC		
weirdbell	dk d d					Coarse		Level	0.00 ms 79.5 m	ns 1.00 ms 0 4		- 1		4			> () ()	() Col	or Analog	Clip
MNS				nstrument or ple Here		$G_{0.5}$	G =	€_20 dB	Initial Peak	Sustain Ve		back Repeat		C-100 %	G* Ĉ	81.	7 % 1/4	0.00" Offset	0.00 % Bas	e Freq	Width Depth Drive Curve Depth Drive Curve Depth Drive Curve Depth Drive Drive Depth Drive Curve Depth
C1	bd thonk D#1		Sam	pie nele		Freq	Multi Fixed	Level	-inf dB 0.0 dE	B −infdB 04 Ke				Time	Tone Volu		Hz		NN) ned	Width 100 % 0.00 % 0.00 % Dr
		V				€57.5 Hz		C.0 dB	Trigger+						-	.0 dB	mat 1			1 1	
						57.5 Hz	<u>\</u> 1 =	- 0.0 dB	_					0%	70 % 6	LU dB		20207			





Audio in 140

Audio in 140

Overview

- Control game from music
- Interactive music
- Music timing in 140
- Fun audio tricks

140 demo

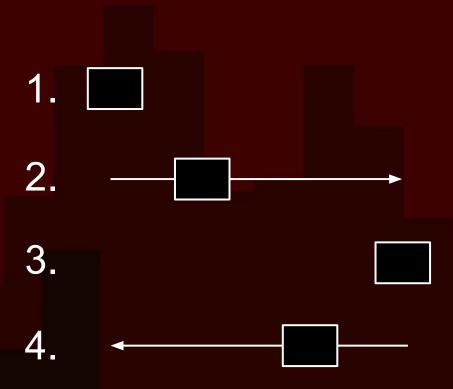


Control Game from Music

Moving a Platform



Moving a Platform



wait for 16th note #1

start moving

wait for 16th note #8

start moving

Basic Approach

- Play music loop
- Use audio time from loop to control game elements (instead of game time)

Unity built-in audio:

AudioSource.time

AudioSource.timeSamples

(seconds)
(samples)

FMOD Unity Integration:

EventInstance.getTimelinePosition

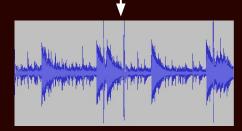
(milliseconds)

Music Events

'Waiting for 16th note #8' means:

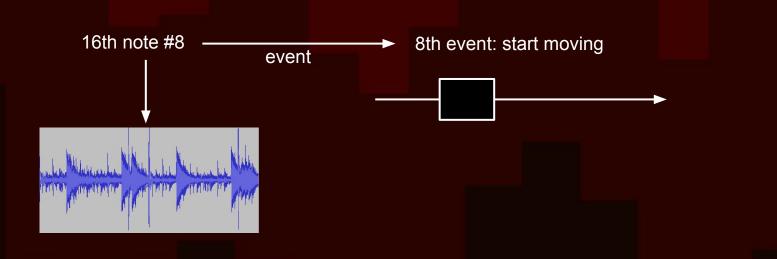
- Get audio time from playing loop
- When next musical beat reached, raise event
- On the 8th event, do something

16th note #8



Music Events

Game elements listen for events and trigger animation on beats



Music Events

- When next musical beat reached, raise event
 - And when is that, exactly?

Useful Calculations

For a given tempo, how long is a note in seconds?





140 beat/m





140 beat/m * 4 note/beat = 560 note/m





140 beat/m * 4 note/beat = 560 note/m = 560/60 note/s

Tempo

How long is a 140 BPM 16th note in seconds?

```
140 beat/m * 4 note/beat = 560 note/m
= 560/60 note/s
```

This means that we have:

60/560 s/note

Tempo

How long is a 140 BPM 16th note in seconds?

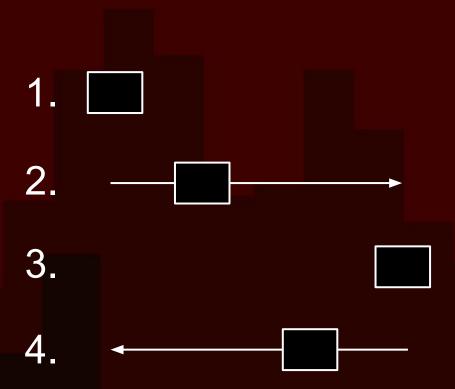
```
140 beat/m * 4 note/beat = 560 note/m
= 560/60 note/s
```

This means that we have:

60/560 s/note ~= 0.10714 s/note



Moving a Platform



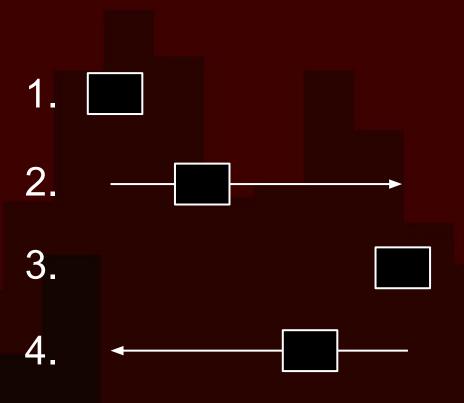
wait for 16th note #0

start moving

wait for 16th note #8

start moving

Moving a Platform



wait for 16th note #0, time = 0 s

wait for 16th note #8, time = 8 * 0.10714 s = 0.857 s

Summary

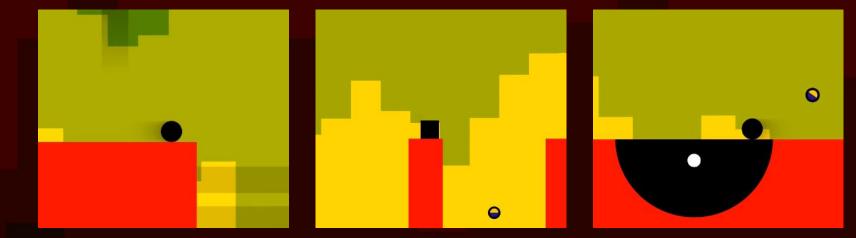
- Game elements wait for music events to control animation
- Music system observes AudioSource.time (Unity built-in)
- ... or EventInstance.getTimelinePosition (FMOD Unity)
- Tempo can be converted to seconds
- Music events are triggered when a given time has been reached



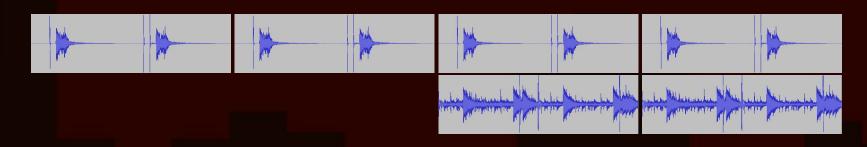
Interactive Music

Interactive Music Mixing

We wanted to mix music interactively in Unity.



time



The Music Timing Problem

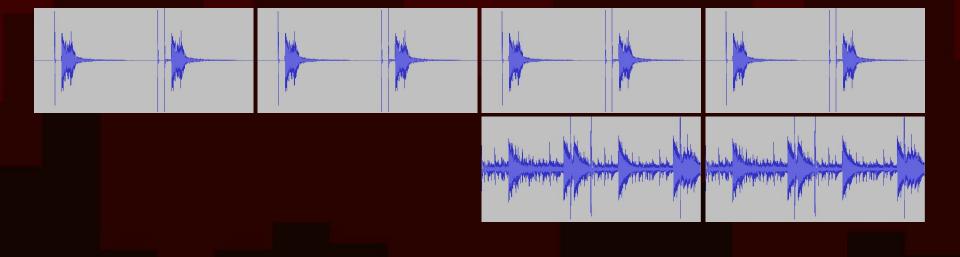
For beat-oriented music, loops should be synchronized with sample accuracy.

- That means a precision of 0.00002 s

Loop Transition

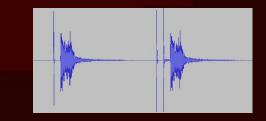
Goal:

- Start loop A and let it run for a while.
- Then start loop B.
- B should be sample-accurately synchronized with loop A.



Start loop A:

audioSourceA.Play()

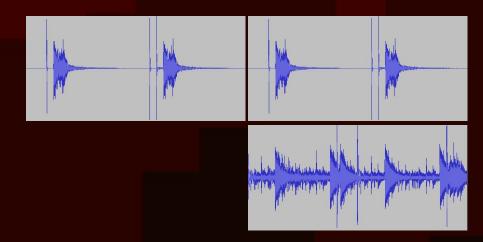


Start loop A:

audioSourceA.Play()

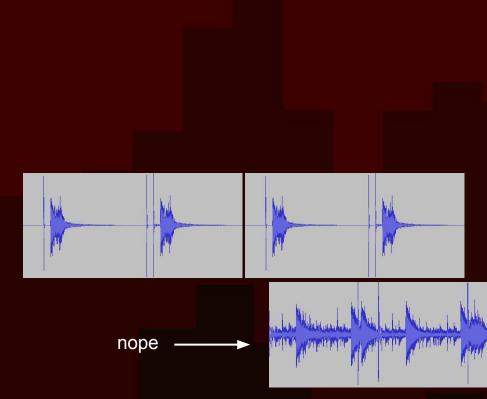
Exactly when A loops, start loop B:

Wait for A to loop, then: audioSourceB.Play()



It doesn't work!

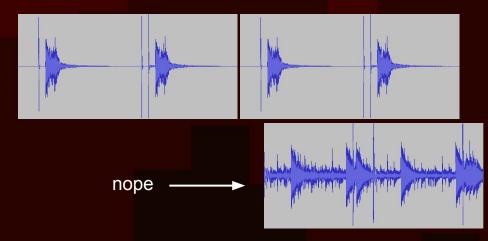
New sound is out of sync.

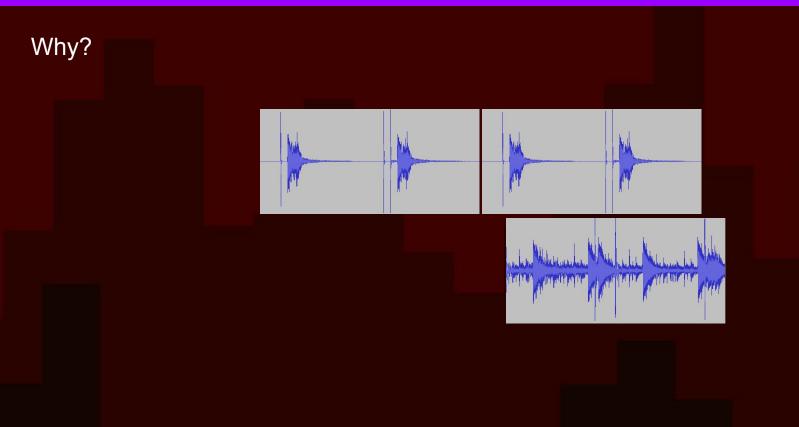


It doesn't work!

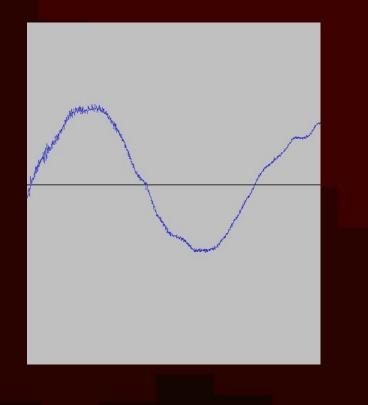
New sound is out of sync.

The problem exists in every sound engine.



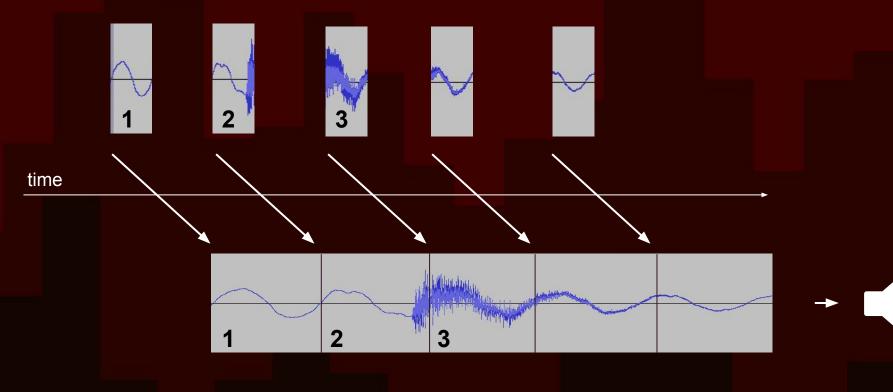


Audio is rendered a fixed number of samples at a time:

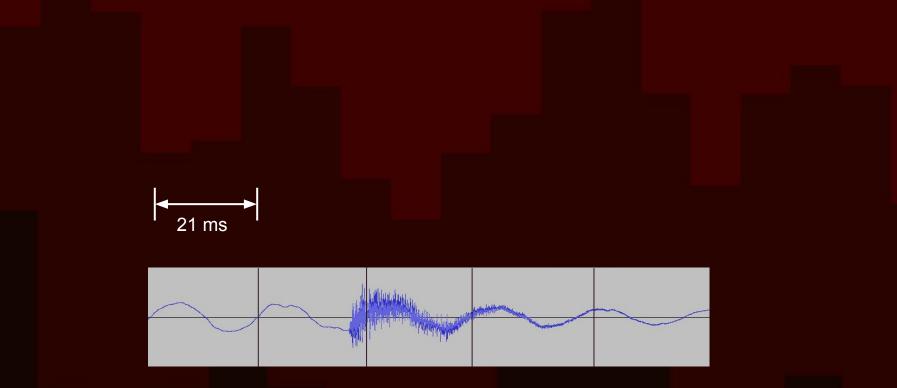


1024 sample buffer, 21 ms

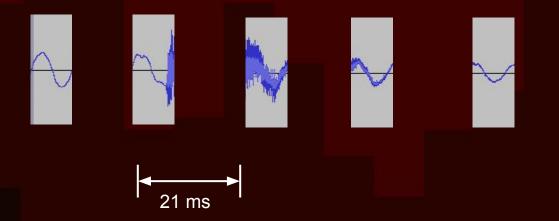
The sound card plays a buffer while the next one is being rendered:



If buffers are e.g. 1024 samples long, we need a new one every 21 ms.



In this case, a new buffer is rendered every 21 ms:



- New sounds won't start immediately, but earliest in the next audio buffer
- Their start time will also be quantized to buffer start times, e.g. 21 ms

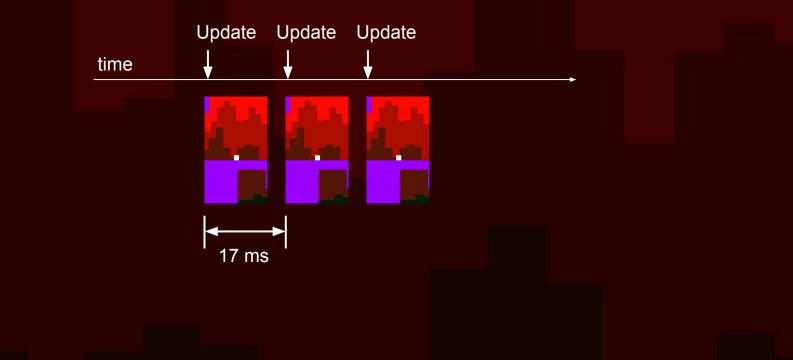


In Unity, our audio code will probably be in an Update method

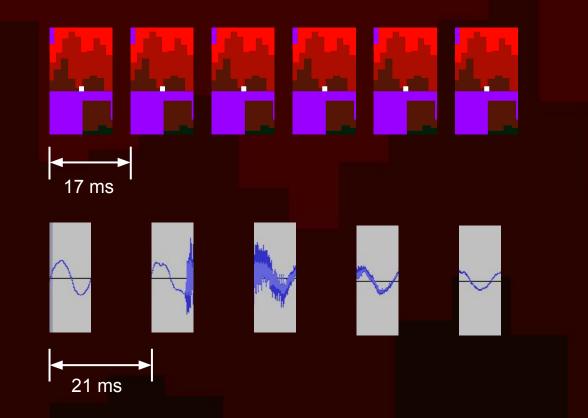
```
using UnityEngine;
public class MyAwesomeScript : MonoBehaviour
    public AudioSource mySound;
    // Use this for initialization
    void Start()
    // Update is called once per frame
    void Update()
        mySound.Play();
```

•••

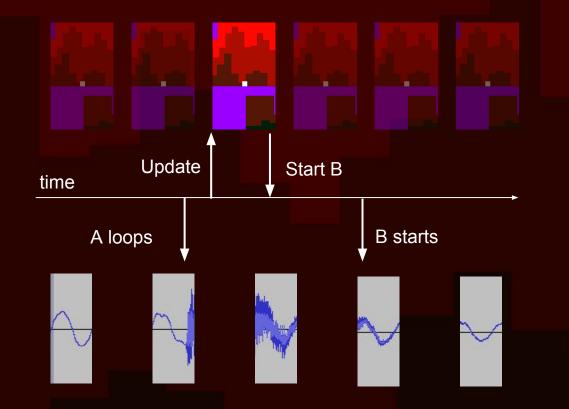
Unity Update methods are called for every video frame (e.g. 17 ms at 60 FPS)



Audio buffers and video frames are not synchronized:

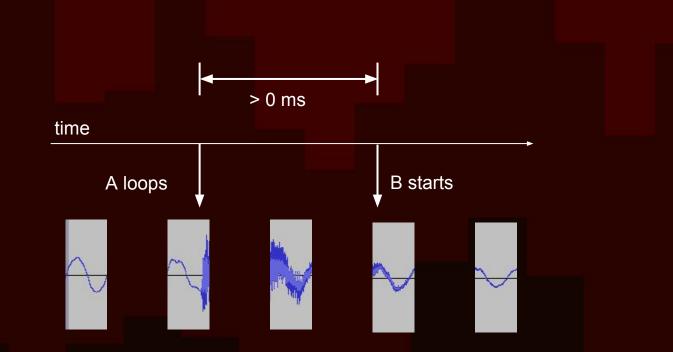


So, if we want to start a sound B exactly when another sound A loops...



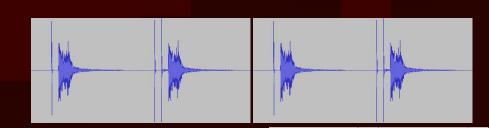
Immediately is Too Late

So, if we want to start a sound B exactly when another sound A loops... Detecting it in Update and playing B immediately is too late!



Immediately is Too Late

So, if we want to start a sound B exactly when another sound A loops... Detecting it in Update and playing B immediately is too late!



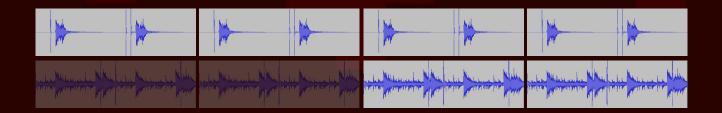


Interactive Music Solutions

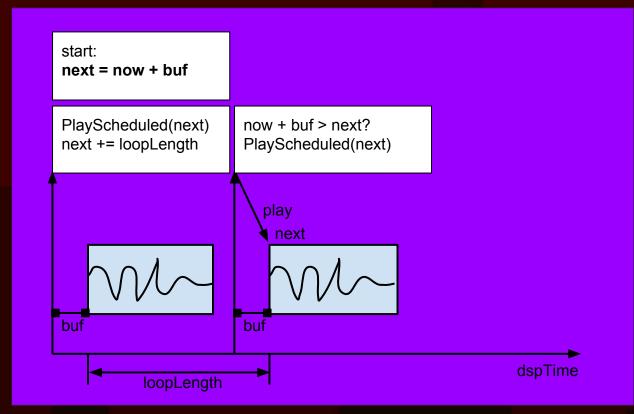
- Synchronized Loops
- PlayScheduled

Solution A: Synchronized Loops

- All loops should be exactly same length, or integer multiples
- All loops should be started in the same frame, possibly muted
- New loops cannot be started
- Never change pitch



Solution B: PlayScheduled



Solution B: PlayScheduled

<u>Start</u>:

```
buf = 0.1 // as low as possible
next = AudioSettings.dspTime + buf
<u>Update</u>:
```

```
now = AudioSettings.dspTime
if(now + buf > next)
    audio.PlayScheduled( next )
    next += loopLength
```

- see http://www.schmid.dk/gallery/play_scheduled/ for C# code

Solution Comparison

Solution A: Synchronized Loops

- Very simple to implement
- Requires a loop for every single independent musical element
- Loops must be same length or integer multiple
- Pitch cannot be changed

Solution Comparison

Solution A: Synchronized Loops

- Very simple to implement
- Requires a loop for every single independent musical element
- Loops must be same length or integer multiple
- Pitch cannot be changed

Solution B: PlayScheduled

- Non-trivial implementation
- Flexible: Individual notes can be sequenced
- No length requirement for music sounds
- Pitch can be changed

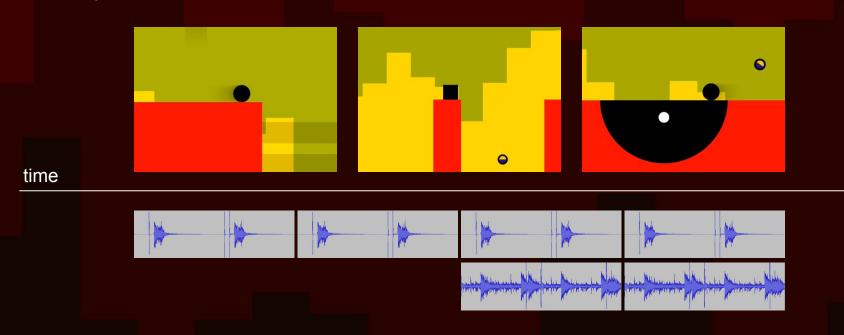
Summary

- Synchronizing loops with sample accuracy is tricky
- Audio is rendered in buffers, delaying and quantizing sounds
- Unity Update calls correspond to video frames, not audio buffers
- Immediately is too late: detecting loop and reacting in Update results in a delay
- Solution A: Synchronized Loops
- Solution B: PlayScheduled

Music Timing in 140

Music Timing in 140

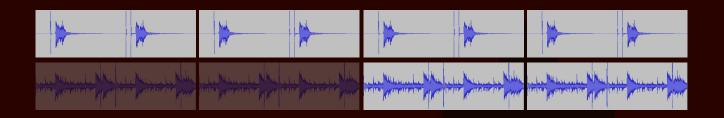
- We wanted the music to be mixed interactively with the gameplay.
- Loops should be sample-accurate.
- We were using Unity 3 at the time, which limited our options (no PlayScheduled)



Music Requirements

Simple solution with sample-accurate timing:

- All music must be loops of a fixed length, or multiples of that length.
- Start all loops in same frame, possibly muted.



Music Requirements

Simple solution with sample-accurate timing:

- All music must be loops of a fixed length, or multiples of that length.
- Start all loops in same frame, possibly muted.

During game progression:

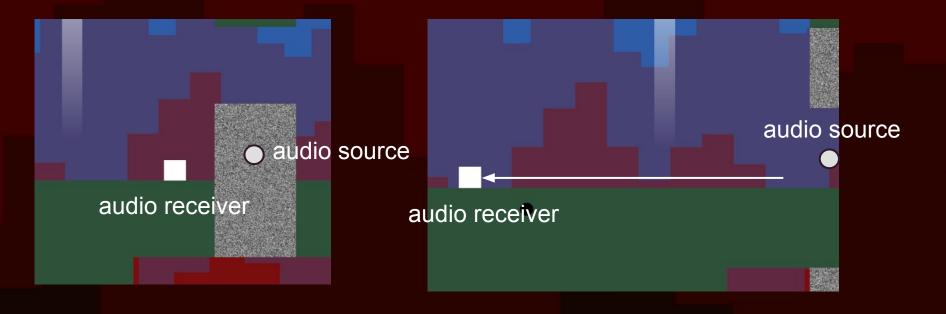
- Control volume/muting and pan.
- Never change pitch unless just before stopping a loop.

Localized Music Loops

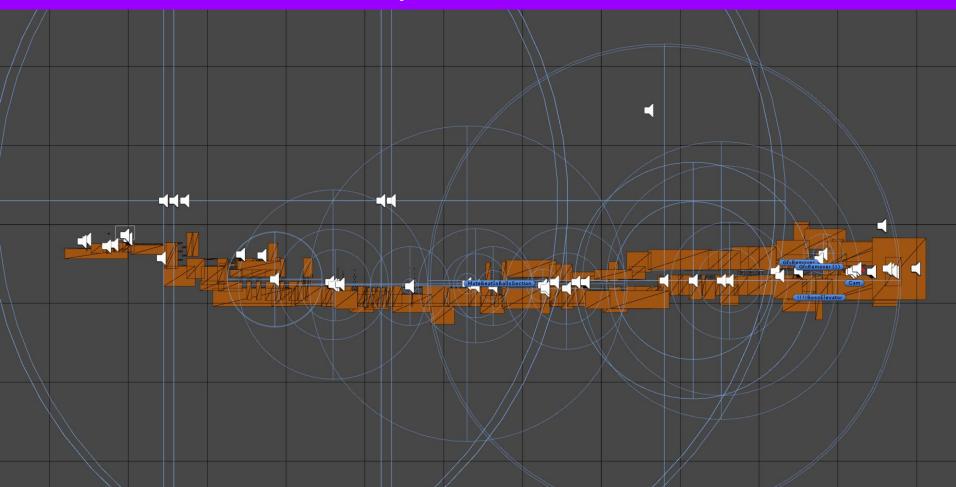
- Music loops are "physically" placed in level geometry
- Dynamic mixing occurs as player moves around
- Certain areas can gain unique atmosphere based on music

Localized Music Loops

Simple attenuation and panning for music loops using the built-in audio system



Localized Music Loops - Level 4



140 demo



Example Audio Loop

🔻 🚅 🗹 Audio Source

E = mE

AudioClip
Output
Mute
Bypass Effects
Bypass Listener Effect
Bypass Reverb Zones
Play On Awake
Loop
Priority

v.	0	111	m	ie	

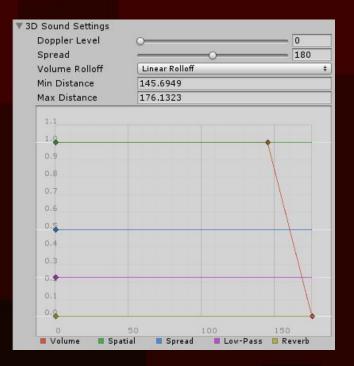
Pitch

Stereo Pan

Spatial Blend

Reverb Zone Mix

ce		🔯 🌣,
	🚔l4-layer4-modnet_up	0
	None (Audio Mixer Group)	0
·		
fects		
nes		
	High Low	0
	O	1
	o	1
	Left Right	0
	0	1
	2D 3D	0
	(F)	



AudioSync component: handles all loops

- Handles fade in/out
- Controls filters
- Adds loop to 'layer' (group)

🛙 💽 🗹 Audio Sync (Script)		💽 🌣,
Script	@ AudioSync	0
Fade In Time	2	
Fade Out Time	1	
Fade On Mutes		
Override Start Pitch	-1	
Add To Audio Layer	-1	
Fade In On Play		
Dont Reset Volume On Umute		
Dont Stop On Mirror Death		
Fade With Filter		23
Low Pass Filter	#airborne (Audio Low Pass Filter)	0
High Pass Filter	None (Audio High Pass Filter)	0
Freq Filter Start	10	
Freq Filter End	5000	
Filter Curve Power	2	
🕼 🗹 Audio Death Down San	nple (Script)	a 🔅,
Script	@AudioDeathDownSample	0
Beat Layer	2	
Death Volume Scale	1	
Unmute On Death		
Down Sample Scale	1	
Downsample	1	
7 🚧 🗹 Audio Low Pass Filter		a \$,
Cutoff Frequency	O	- 5000
Lowpass Resonance Q	1	
🛚 📈 🗹 Audio Echo Filter		💽 \$,
Delay	643	
Decay Ratio	0.4	
Wet Mix	0.4	
Dry Mix	0.6	

AudioSync component: handles all loops

- Handles fade in/out
- Controls filters
- Adds loop to 'layer' (group)

AudioDeathDownSample: downsample filter (more on this later)

	🔯 🌣,
@ AudioSync	0
2	
1	
-1	
-1	
	22
# airborne (Audio Low Pass Filter)	0
None (Audio High Pass Filter)	0
10	
5000	
2	
mple (Script)	a *,
💽 AudioDeathDownSample	0
2	
1	
1	
1	
	🔯 🌣,
	- 5000
1	
	💽 \$,
643	
0.4	
0.4	
0.6	
	2 1 ✓ -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 5000 2 mple (Script) © AudioDeathDownSample 2 1 1 1 643 0.4 0.4

AudioSync component: handles all loops

- Handles fade in/out
- Controls filters
- Adds loop to 'layer' (group)

AudioDeathDownSample: downsample filter

AudioLowPassFilter: built-in Unity LPF

🔻 健 🗹 Audio Sync (Script)		💽 🌣,
Script	@ AudioSync	0
Fade In Time	2	
Fade Out Time	1	
Fade On Mutes		
Override Start Pitch	-1	
Add To Audio Layer	-1	
Fade In On Play		
Dont Reset Volume On Umute		
Dont Stop On Mirror Death		
Fade With Filter		222
Low Pass Filter	🚧 airborne (Audio Low Pass Filter)	0
High Pass Filter	None (Audio High Pass Filter)	0
Freq Filter Start	10	
Freq Filter End	5000	
Filter Curve Power	2	
🔻 🕼 🗹 Audio Death Down Sa	mple (Script)	💽 ‡,
Script	@ AudioDeathDownSample	0
Beat Layer	2	
Death Volume Scale	1	
Unmute On Death		
Down Sample Scale	1	
Downsample	1	
🔻 🚧 Audio Low Pass Filter		💽 🌣,
Cutoff Frequency	<u>_</u>	- 5000
Lowpass Resonance Q	1	
🔻 📈 🗹 Audio Echo Filter		💽 🔅,
Delay	643	
Decay Ratio	0.4	
Wet Mix	0.4	
Dry Mix	0.6	

AudioSync component: handles all loops

- Handles fade in/out
- Controls filters
- Adds loop to 'layer' (group)

AudioDeathDownSample: downsample filter

AudioLowPassFilter: built-in Unity LPF

AudioEchoFilter: built-in Unity delay

Script AudioSync Fade In Time 2 Fade Out Time 1 Fade On Mutes Image: Script Script Script Override Start Pitch -1 Add To Audio Layer -1 Fade In On Play Image: Script	🔻 📴 🗹 Audio Sync (Script)		₽ \$,
Fade Out Time 1 Fade On Mutes Image: Constraint of the set of	Script	@ AudioSync	0
Fade On Mutes Override Start Pitch 1 Add To Audio Layer -1 Fade In On Play Dont Reset Volume On Umute Dont Stop On Mirror Death Fade With Filter Icow Pass Filter High Pass Filter None (Audio Low Pass Filter) O Freq Filter Start 10 Freq Filter Start Freq Filter End 5000 Filter Curve Power 2 Ce M Audio Death Down Sample (Script) Beat Layer 2 Death Volume Scale 1 Unmute On Death Down Sample Scale 1 Down Sample Scale 1 Down Sample Scale 1 Down Sample Scale 1 Downsample 1 Pelay 643 Decay Ratio Wet Mix	Fade In Time	2	
Override Start Pitch -1 Add To Audio Layer -1 Fade In On Play	Fade Out Time	1	
Add To Audio Layer -1 Fade In On Play	Fade On Mutes		
Fade In On Play Dont Reset Volume On Umute Dont Stop On Mirror Death Fade With Filter Low Pass Filter Iow Pass Filter High Pass Filter None (Audio Low Pass Filter) O High Pass Filter None (Audio High Pass Filter) O Freq Filter Start 10 Freq Filter End 5000 Filter Curve Power 2 Oeath Down Sample (Script) Script Beat Layer 2 Death Volume Scale 1 Unmute On Death Down Sample Scale 1 Downsample 1 V M Audio Low Pass Filter Cutoff Frequency Lowpass Resonance Q 1 V M Audio Echo Filter Delay 643 Decay Ratio Wet Mix	Override Start Pitch	-1	
Dont Reset Volume On Umute Dont Stop On Mirror Death Fade With Filter Low Pass Filter High Pass Filter None (Audio Low Pass Filter) O High Pass Filter None (Audio High Pass Filter) O Freq Filter Start 10 Freq Filter End 5000 Filter Curve Power 2 Ce M Audio Death Down Sample (Script) Beat Layer 2 Death Volume Scale 1 Unmute On Death Down Sample Scale 1 Down Sample Scale 1 V M Audio Low Pass Filter Cutoff Frequency Lowpass Resonance Q 1 V M Audio Echo Filter Delay 643 Decay Ratio Wet Mix	Add To Audio Layer	-1	
Dont Stop On Mirror Death Fade With Filter Low Pass Filter Low Pass Filter Migh Pass Filter None (Audio Low Pass Filter) o High Pass Filter None (Audio High Pass Filter) o Freq Filter Start 10 Freq Filter End 5000 Filter Curve Power 2 Image: Current Power 3 Image: Current Power 3 Image: Current Power 4 Image: Current Power 3 Image: Current Power 4 Image: Current Power 3 Image: Current Power 4 1 1<	Fade In On Play		
Fade With Filter Low Pass Filter Low Pass Filter High Pass Filter None (Audio Low Pass Filter) Freq Filter Start 10 Freq Filter End 5000 Filter Curve Power 2 C Madio Death Down Sample (Script) Script Beat Layer 2 Death Volume Scale 1 Unmute On Death Down Sample Scale 1 Downsample 1 V M Audio Low Pass Filter Cutoff Frequency Lowpass Resonance Q 1 V M Audio Echo Filter Delay 643 Decay Ratio Wet Mix	Dont Reset Volume On Umute		
Low Pass Filter # airborne (Audio Low Pass Filter) ○ High Pass Filter None (Audio High Pass Filter) ○ Freq Filter Start 10 ○ Freq Filter End 5000 ○ Filter Curve Power 2 ○ Image: Current Power 2 ○ Image: Current Power ○ ○ Image: Power ○ ○ ○ Image: Power ○ ○ ○ Image: Power ○ ○ ○<	Dont Stop On Mirror Death		
High Pass Filter None (Audio High Pass Filter) ○ Freq Filter Start 10 ● Freq Filter End 5000 ● Filter Curve Power 2 ● Audio Death Down Sample (Script) ● ◆ Script ● AudioDeathDownSample ● Beat Layer 2 ● Death Volume Scale 1 ● Unmute On Death ● ● Down Sample Scale 1 ● Downsample 1 ● V ✓ Audio Low Pass Filter ● ● Cutoff Frequency ● ● Lowpass Resonance Q 1 ● V ✓ Audio Echo Filter ● ● Delay 643 ● Decay Ratio 0.4 ●	Fade With Filter		
Freq Filter Start 10 Freq Filter End 5000 Filter Curve Power 2 Image: Script Image: Audio Death Down Sample (Script) Image: Script Down Sample Scale Image: Script (Script) Image: Script Down Sample Scale Image: Script (Script) Image: Scale Down Sample Scale Image: Script (Script) Image: Scale Down Sample Scale I Image: Script (Script) Image: Scale Down Sample Scale I Image: Script (Script) Image: Scale Down Sample Scale I Image: Scale	Low Pass Filter	≓ airborne (Audio Low Pass Filter)	0
Freq Filter End 5000 Filter Curve Power 2 C Audio Death Down Sample (Script) * Script AudioDeathDownSample Beat Layer 2 Death Volume Scale 1 Unmute On Death 0 Down Sample Scale 1 Downsample 1 V # Audio Low Pass Filter * Cutoff Frequency 5000 Lowpass Resonance Q 1 V Audio Echo Filter * Delay 643 Decay Ratio 0.4 Wet Mix 0.4	High Pass Filter	None (Audio High Pass Filter)	0
Filter Curve Power 2 Image: Script AudioDeath Down Sample (Script) Script AudioDeathDownSample Beat Layer 2 Death Volume Scale 1 Unmute On Death 0 Down Sample Scale 1 Downsample 1 V# Maudio Low Pass Filter Image: Scipt Source Cutoff Frequency 5000 Lowpass Resonance Q 1 V# Maudio Echo Filter Image: Scipt Source Delay 643 Decay Ratio 0.4 Wet Mix 0.4	Freq Filter Start	10	
Image: Constraint of the second s	Freq Filter End	5000	
Script	Filter Curve Power	2	
Beat Layer 2 Death Volume Scale 1 Unmute On Death □ Down Sample Scale 1 Downsample 1 ✓ ✓ ✓ Audio Low Pass Filter Cutoff Frequency □ Lowpass Resonance Q 1 ✓ ✓ ✓ Audio Echo Filter □ ◆, Delay 643 Decay Ratio 0.4 Wet Mix 0.4	🖲 🖬 Audio Death Down San	nple (Script)	2 \$,
Death Volume Scale 1 Unmute On Death	Script	@ AudioDeathDownSample	0
Unmute On Death Down Sample Scale 1 Downsample 1	Beat Layer	2	
Down Sample Scale 1 Downsample 1 V & Audio Low Pass Filter Cutoff Frequency Image: Cutoff Frequency Lowpass Resonance Q 1 V & Audio Echo Filter Image: Cutoff Frequency Delay 643 Decay Ratio 0.4 Wet Mix 0.4	Death Volume Scale	1	
Downsample 1 Image: Cutoff Frequency 5000 Lowpass Resonance Q 1 Image: Cutoff Frequency 5000 Delay 643 Decay Ratio 0.4 Wet Mix 0.4	Unmute On Death		
Image: Constraint of the second se	Down Sample Scale	1	
Cutoff Frequency 5000 Lowpass Resonance Q 1 Addio Echo Filter 2 Delay 643 Decay Ratio 0.4 Wet Mix 0.4	Downsample	1	
Lowpass Resonance Q 1 Audio Echo Filter Delay Decay Ratio Wet Mix 0.4 0.4	🔻 🚧 Audio Low Pass Filter		₽ \$,
Audio Echo Filter Image: Constraint of the second	Cutoff Frequency		00
Delay 643 Decay Ratio 0.4 Wet Mix 0.4	Lowpass Resonance Q	1	
Decay Ratio 0.4 Wet Mix 0.4	🔻 🚧 🗹 Audio Echo Filter	(2 \$,
Wet Mix 0.4	Delay	643	
	Decay Ratio	0.4	
Dry Mix 0.6	Wet Mix	0.4	
	Dry Mix	0.6	

Masking Loops

Music loops can be masked:

- Fade using filters
- Delay
- Unmuting at specific beats





• Over 20 loops running simultaneously

🕼 🗹 Gravity Boss Audio (S	Script) 🔲	\$,
Script	🕞 GravityBossAudio	0
► Filters		
Weapon Laser	🕞 Weapon (WeaponLaser)	0
Audio_weapon Hit	🕞 WeaponHit (AudioSync)	0
Audio_weapon Miss	🕞 WeaponMiss (AudioSync)	0
Audio_1_hit Down	🕞 1_hit (AudioSync)	0
Audio_1_miss Down	© 1_miss (AudioSync)	0
Audio 1_chords Down	©1_chords (AudioSync)	0
Audio_2_first	© 2_first (AudioSync)	0
Audio_2_hit Down	© 2_hit (AudioSync)	0
Audio_2_miss Down	© 2_miss (AudioSync)	0
Audio 2_chords Down	© 2_chords (AudioSync)	0
Audio_3_first	© 3_first (AudioSync)	0
Audio_3_hit Down	🕞 3_hit (AudioSync)	0
Audio_3_miss Down	💽 3_miss (AudioSync)	0
Audio 3_chords Down	© 3_chords (AudioSync)	0
Audio_3_gate Chord	<pre>gateChord (Audio Source)</pre>	0
Audio_end Sequence	4_endSequence (Audio Source)	0
Audio_weapon Collect Sound	WeaponCollect (Audio Source)	0
Lead Up Ambient	intropad (AudioSync)	0
Allmusic Layer	@ AMusicLayer5 (AudioSyncLayer)	0
Respawn Sound	🕞 0_respawn (AudioSync)	0
Intro Pad	🕞 intropad (AudioSync)	0
Gate Sound	🕞 gate_opened (AudioSync)	0
Boss Intro	BossRumble (AudioSync)	0
Gate Fade Speed	0.04	-
Gate Max Volume	0.5	

• Separate loops for hit and miss

 Muted / unmuted when hit or miss is determined

🔻 健 🗹 Gravity Boss Audio (S	Script)	\$,
Script	GravityBossAudio	0
▶ Filters		
Weapon Laser	🕞 Weapon (WeaponLaser)	0
Audio_weapon Hit	🕞 WeaponHit (AudioSync)	0
Audio_weapon Miss	🕞 WeaponMiss (AudioSync)	0
Audio_1_hit Down	🕞 1_hit (AudioSync)	0
Audio_1_miss Down	€ 1_miss (AudioSync)	0
Audio 1_chords Down	@1_chords (AudioSync)	0
Audio_2_first	© 2_first (AudioSync)	0
Audio_2_hit Down	© 2_hit (AudioSync)	0
Audio_2_miss Down	© 2_miss (AudioSync)	0
Audio 2_chords Down	@ 2_chords (AudioSync)	0
Audio_3_first	© 3_first (AudioSync)	0
Audio_3_hit Down	🕞 3_hit (AudioSync)	0
Audio_3_miss Down	© 3_miss (AudioSync)	0
Audio 3_chords Down	© 3_chords (AudioSync)	0
Audio_3_gate Chord	<pre></pre>	0
Audio_end Sequence	4_endSequence (Audio Source)	0
Audio_weapon Collect Sound	WeaponCollect (Audio Source)	0
Lead Up Ambient	intropad (AudioSync)	0
Allmusic Layer	@ AMusicLayer5 (AudioSyncLayer)	0
Respawn Sound	© 0_respawn (AudioSync)	0
Intro Pad	e intropad (AudioSync)	 0
Gate Sound	@gate_opened (AudioSync)	 0
Boss Intro	BossRumble (AudioSync)	0
Gate Fade Speed	0.04	
Gate Max Volume	0.5	

• Each stage has its own set of loops

		(crint)	ø.
1.1.1.1	健 🗹 Gravity Boss Audio (S		
	Script	💽 GravityBossAudio	0
18	Filters		1020
	Weapon Laser	@ Weapon (WeaponLaser)	0
	Audio_weapon Hit	💽 WeaponHit (AudioSync)	0
1.00	Audio_weapon Miss	💽 WeaponMiss (AudioSync)	0
	Audio_1_hit Down	💽 1_hit (AudioSync)	0
1	Audio_1_miss Down	©1_miss (AudioSync)	0
	Audio 1_chords Down	©1_chords (AudioSync)	0
14	Audio_2_first	© 2_first (AudioSync)	0
	Audio_2_hit Down	© 2_hit (AudioSync)	0
2	Audio_2_miss Down	© 2_miss (AudioSync)	0
	Audio 2_chords Down	© 2_chords (AudioSync)	0
14	Audio_3_first	© 3_first (AudioSync)	0
	Audio_3_hit Down	© 3_hit (AudioSync)	0
2	Audio_3_miss Down	© 3_miss (AudioSync)	0
13	Audio 3_chords Down	© 3_chords (AudioSync)	0
14	Audio_3_gate Chord	0_gateChord (Audio Source)	0
	Audio_end Sequence	4_endSequence (Audio Source)	0
- 23	Audio_weapon Collect Sound	WeaponCollect (Audio Source)	0
	Lead Up Ambient	intropad (AudioSync)	0
14	Allmusic Layer	@ AMusicLayer5 (AudioSyncLayer)	0
	Respawn Sound	© 0_respawn (AudioSync)	0
	Intro Pad	intropad (AudioSync)	0
- 33)	Gate Sound	@gate_opened (AudioSync)	0
13	Boss Intro	BossRumble (AudioSync)	0
	Gate Fade Speed	0.04	
1	Gate Max Volume	0.5	-

Level 4 Boss

• Each stage is in a different key

- Loops have long Ableton Reverb tails
- Reverb tails and key change requires special transitional first loop

🖲 🗹 Gravity Boss Audio (S	Script)	\$,
Script	🕞 GravityBossAudio	0
▶ Filters		
Weapon Laser	💽 Weapon (WeaponLaser)	0
Audio_weapon Hit	💽 WeaponHit (AudioSync)	0
Audio_weapon Miss	🕞 WeaponMiss (AudioSync)	0
Audio_1_hit Down	© 1_hit (AudioSync)	0
Audio_1_miss Down	©1_miss (AudioSync)	0
Audio 1_chords Down	©1_chords (AudioSync)	0
Audio_2_first	© 2_first (AudioSync)	0
Audio_2_hit Down	© 2_hit (AudioSync)	0
Audio_2_miss Down	© 2_miss (AudioSync)	0
Audio 2_chords Down	© 2_chords (AudioSync)	0
Audio_3_first	© 3_first (AudioSync)	0
Audio_3_hit Down	© 3_hit (AudioSync)	0
Audio_3_miss Down	© 3_miss (AudioSync)	0
Audio 3_chords Down	© 3_chords (AudioSync)	0
Audio_3_gate Chord	<pre>gateChord (Audio Source)</pre>	0
Audio_end Sequence	4_endSequence (Audio Source)	0
Audio_weapon Collect Sound	📢 WeaponCollect (Audio Source)	0
Lead Up Ambient	intropad (AudioSync)	0
Allmusic Layer	🕞 AMusicLayer5 (AudioSyncLayer)	0
Respawn Sound	🕞 0_respawn (AudioSync)	0
Intro Pad	intropad (AudioSync)	0
Gate Sound	🕞 gate_opened (AudioSync)	0
Boss Intro	BossRumble (AudioSync)	0
Gate Fade Speed	0.04	
Gate Max Volume	0.5	

Level 4 Boss

- House chords are faded in using filter between hits to create tension
- Final chord is faded in, anticipating end sequence

🔻 📴 🗹 Gravity Boss Audio (S	Script)	(
Script	🕞 GravityBossAudio	0
▶ Filters		
Weapon Laser	🕞 Weapon (WeaponLaser)	0
Audio_weapon Hit	🕞 WeaponHit (AudioSync)	0
Audio_weapon Miss	🕞 WeaponMiss (AudioSync)	0
Audio_1_hit Down	🕞 1_hit (AudioSync)	0
Audio_1_miss Down	© 1_miss (AudioSync)	0
Audio 1_chords Down	€ 1_chords (AudioSync)	0
Audio_2_first	© 2_first (AudioSync)	0
Audio_2_hit Down	© 2_hit (AudioSync)	0
Audio_2_miss Down	© 2_miss (AudioSync)	0
Audio 2_chords Down	© 2_chords (AudioSync)	0
Audio_3_first	🕞 3_first (AudioSync)	0
Audio_3_hit Down	🕞 3_hit (AudioSync)	0
Audio_3_miss Down	© 3_miss (AudioSync)	0
Audio 3_chords Down	@ 3_chords (AudioSync)	0
Audio_3_gate Chord	<pre>gateChord (Audio Source)</pre>	0
Audio_end Sequence	4_endSequence (Audio Source)	0
Audio_weapon Collect Sound	dWeaponCollect (Audio Source)	0
Lead Up Ambient	intropad (AudioSync)	0
Allmusic Layer	@ AMusicLayer5 (AudioSyncLayer)	0
Respawn Sound	© 0_respawn (AudioSync)	0
Intro Pad	💽 intropad (AudioSync)	0
Gate Sound	🕞 gate_opened (AudioSync)	0
Boss Intro	BossRumble (AudioSync)	0
Gate Fade Speed	0.04	
Gate Max Volume	0.5	

140 demo



Doppler Effect

Disable Doppler effect to avoid drifting loops!

Project Settings	>	Input
Network Emulation	>	Tags and Layers
Graphics Emulation	>	Audio
-		Time
Snap Settings		Player

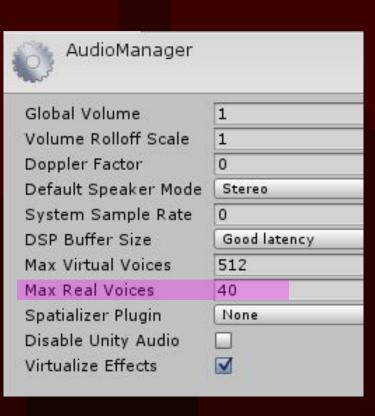
AudioManager	
Global Volume	1
Volume Rolloff Scale	1
Doppler Factor	0
Default Speaker Mode	Stereo
System Sample Rate	0
DSP Buffer Size	Good latenc
Max Virtual Voices	512
Max Real Voices	40
Spatializer Plugin	None
Disable Unity Audio	
Virtualize Effects	

Max Real Voices

In Unity 3, if more than 32 sounds are playing at once, we lose sample accuracy!

Same limitation in Unity 2017, but the number can be increased.

140 currently uses 40 voices.



Summary

- In 140, we start all loops at once and control their volume
- Localized music loops: volume and pan using built-in audio system
- Use filters and effects to mask loops
- Level 4 boss music uses muting, fading, and filtering of 20 loops
- Disable Doppler effect
- Check that 'Max Real Voices' is enough

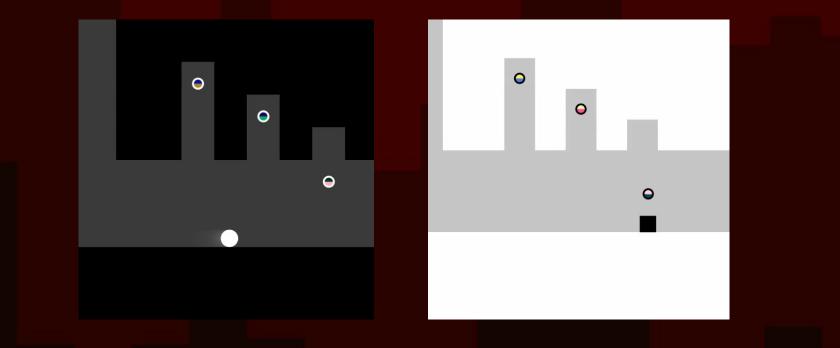
Fun Audio Tricks

Fun Audio Tricks

- Modulation
- Cassette tape jam
- Downsampling
- Fake crash

Menu Modulation

- When picking up a mirror mode key, modulate ambient track from Cm to Dm.
- Track contains no rhythmic elements, so loop synchronization is not an issue.



From Semitones to Frequency

Modulate ambient track from Cm to Dm:

Frequency of D relative to C (+2 semitones, well-tempered):

 $2^{2/12} \sim 1.12246204830937$

From Semitones to Frequency

Modulate ambient track from Cm to Dm:

Frequency of D relative to C (+2 semitones, well-tempered):

 $2^{2/12} \sim 1.12246204830937$

Gradual pitch change code, as f goes from 0 to 1:

relativePitch = pow(2.0, 2.0 / 12.0)
source.pitch = lerp(1.0, relativePitch, f)

Cassette Tape Jam

When a key is delivered, the playing music is stopped with a cassette tape jam-inspired effect.





Image credit: Kristi Bogel

Cassette Tape Jam

The tape jam effect is achieved with:

- applying strong vibrato to all music tracks,
- enveloping pitch towards 0 and volume towards silence.

Cassette Tape Jam Code

The tape jam effect is achieved with:

- applying strong vibrato to all music tracks,
- enveloping pitch towards 0 and volume towards silence.

As f goes from 0 to 1:

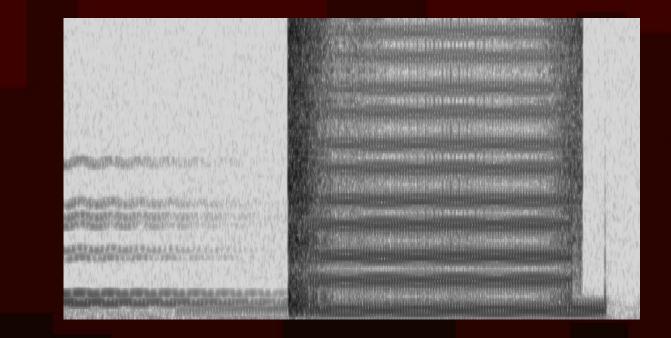
Downsampling

When the player dies, the visuals turn black and white, and the audio is brutally distorted.



Downsampling

- The death audio effect is a simple variable downsampling filter.
- It sounds ugly on purpose.

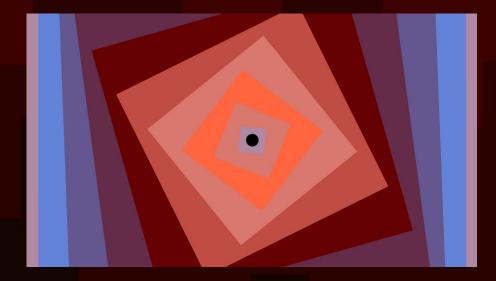


Downsampling Filter Code

The simplest variable downsampling filter: repeat every D'th sample D times.

Fake crash

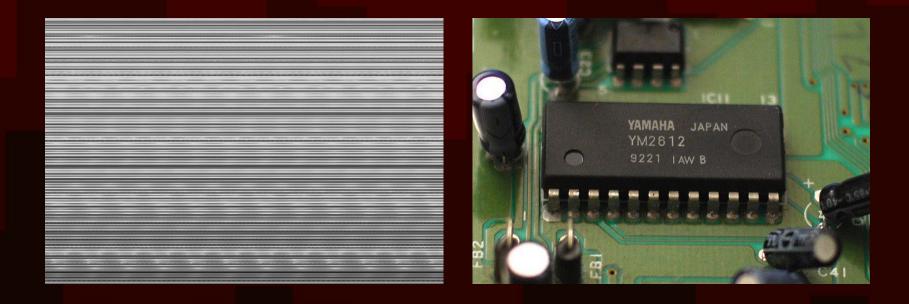
When the final boss is beaten, the game simulates the game crashing. Or rather, how the game *would* crash if it was running on a SEGA Genesis.





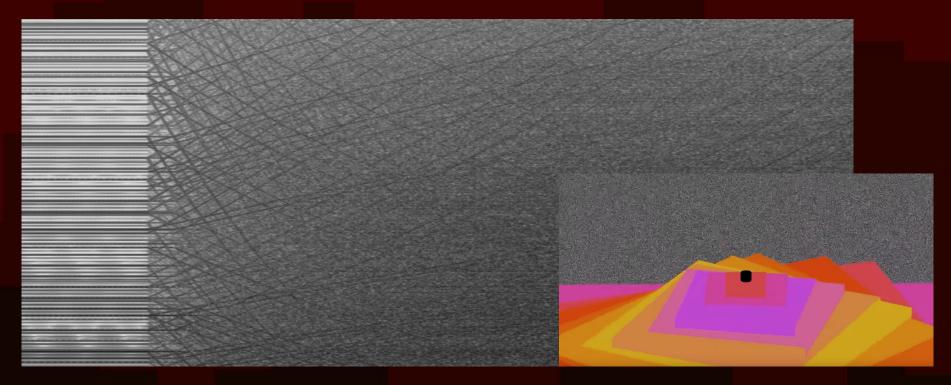
Fake crash

- The final chord of the boss fight and the screen is unchanged for 9 seconds, leaving at least one YouTuber very nervous.
- Crashes on old oscillator-based systems would have similar behaviour.



Glissando and 3D Rotation

- The oscillators slowly starts individually wandering towards a final chord.
- The game rotates the view, for the first time exposing a 3D world.



Summary

- Pitch change on playing track works for ambient music.
- Vibrato and amplitude and pitch envelopes simulate cassette tape jam.
- Downsampling filter is implemented OnAudioFilterRead method.
- Game ends with fake crash sound with hanging oscillators.
- Fake crash is resolved with oscillators gliding towards final chord.

Questions?

