

# CMP – 407

## Audio Programming

# Audio Techniques Used

- Audio Middleware
- Dynamic Audio
- Recorded Audio
- Digital Music
- Audio Synthesis
- Spatial Audio
- Real-time Audio Effects
- Compressed Audio Formats

# Audio Middleware

- Wwise 2019.2.7.7402
- Unity 2020.1.4f1
- Wwise is more flexible than the Unity sound engine.
- Can be used in many development environments, not just Unity.

# Dynamic Audio

- Took inspiration from the game “Ape Out”.
- Footsteps are played in time with the beat of the music.
- Make use of “stingers”.
- Cues are set up in the main music track for the stingers to sync to.
- Music switch container for different footstep sounds on different surfaces. Footstep sounds are treated as music segments.
- Separate music track for each footstep sample.
- Tracks are selected at random.
- Events called from Unity set triggers.
- These triggers trigger the stinger to play at the next cue of the music.
- Focus point of each stinger track set to the beginning of their sound waves.
- Creates a feeling of dancing to the music.
- Footstep sounds are sampled from Minecraft.

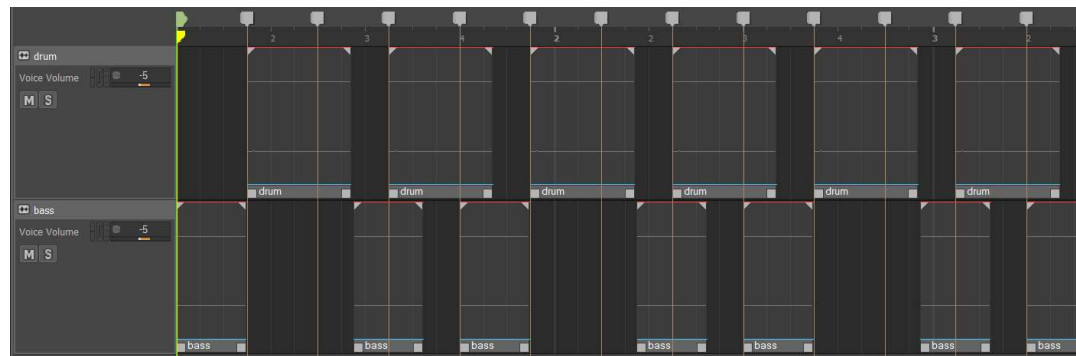
# Recorded Audio

- Use of foley sound effects.
- Linked to events, called from Unity.
- Source editor to split recorded audio into separate sections.
- Only selected sections are stored in sound bank to save memory.
- Use of fade out duration to make sound end less abruptly.
- Sequence container for initial loud crunch first, followed by random container of quieter chewing sounds.
- Use “continuous” setting in sequence containers to play whole sequence.
- Use of “transitions” to reduce the delay between sound effects in sequence container.
- Random container of sound effects for pulling carrots from ground.
- Use “shuffle” setting in random containers to avoid sounds being repeated.
- Pitch and volume of sound effects randomized in parent container, this optimizes work flow and allows for better control.
- Effects are played through different auxiliary busses, depending on in-game auxiliary sends for effects to be applied.



# Digital Music

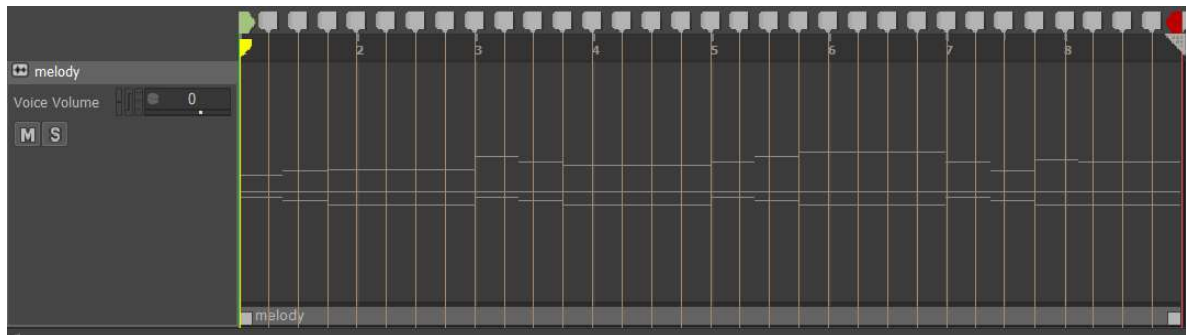
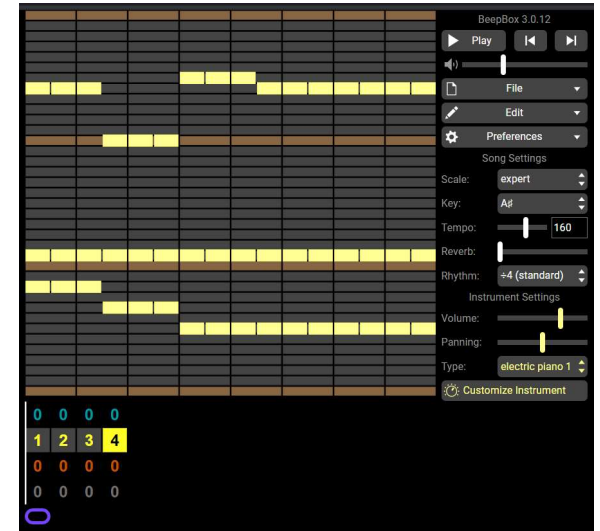
- Main drum beat uses sampled drum sounds that are composed in Wwise to allow for changes in tempo and to map other sounds to it.



- Cues are assigned on off beats of drum and bass rhythm for interactive sound effects to be played via “stingers”.

# Digital Music

- Melody composed with online tool, BeepBox.
- Saved as a MIDI file and imported into Wwise.
- This can be played using a synthesiser.



# Audio Synthesis

- For the melody, the frequency mode is set to MIDI Note to be used with MIDI file.
- Use 2 oscillators, one with a sine wave and one with sawtooth wave for electric but still full effect.
- Offset the pulse width of the sawtooth and sine wave to 25 and 75 respectively to make the sound more jagged.
- Output level of envelope set to have very fast attack, mid level decay and sustain a long release. Attempting to simulate keyboard.



# Audio Synthesis

- For bee buzzing sound effect used a single wave with a low base frequency of 35Hz to hear the individual beats of the bee's wings.
- Single oscillator with square wave for a harsh, piercing sound.
- Changed the timbre of the oscillator by changing its pulse width so that the wave became asymmetrical, creating a drone like effect.
- Used a Wwise pitch shifter plugin effect to increase the pitch of the sound whilst keeping it softened.

# Spatial Audio

- Using Wwise “positioning” functionality to make use of stereo output and play the sound from the correct direction depending on position.
- Using attenuation to set the volume of the bee's buzzing sound, dependent on distance from the player.
- Attenuation fades logarithmically so it is most noticeable when close and not annoying at a distance.
- Not using cone attenuation as it is low frequency. Low frequency sounds travel through objects easily, like the player's head. Didn't sound right.



# Spatial Audio

- For the speaker playing the melody, I got most control of the spatial audio by using RTPC to manipulate volume depending on distance, emitter angle and listener angle.
- You can stand behind the speaker and look different directions to see the effects of these parameters.



# Real-time Audio Effects

- Low-pass filter is used to muffle the sound of the speaker when occluded by the wall.
- This drowns out the higher frequencies of the sound.
- A timer is used to transition smoothly between states for more realistic effect.
- Wwise states are used to trigger this effect and are switched by events sent from Unity when the player is in a trigger box.

# Real-time Audio Effects

- Delay effect applied to an auxiliary bus that multiple audio effects are played through when in the cave.
- Gives the effect of an echo which would be apparent in a large empty space like this.
- Footsteps are effected.
- Can carry a carrot in the cave and chomp on it!

# Real-time Audio Effects

- The pitch of the bee's buzzing sound is adjusted dynamically using Real Time Parameter Controls, specifically the distance from the player.
- Use a steep, logarithmic curve to map pitch shifting to distance when close.
- Creates a kind of doppler effect as the bee flies past.

# Real-time Audio Effects

- RTPC is used to fade out all sounds when the player dies.

# Compressed Audio Formats

- Using Vorbis Data Compression to compress wav files.
- Significantly reduces file size while maintaining good audio quality.
- Can also reduce sample rate.
- Don't have any files large enough to warrant streaming.
- Using separate sound banks for separating sound files associated with events. Not all sound effects will be needed all the time. Different levels for example. Can save on memory by only loading needed sound banks.
- De-selected audio files that are not in use will not be added to audio banks, but will still be available for later use.

Audio Source	Lang...	Orig...	Con...	Original SR	Converted SR	Original Size	Converted Size	Size Ratio	Duration	Bandwidth
grass1	SFX	2.0	2.0	48000	48000	49.5 KB	3.9 KB	12.8:1	0.263	14.7 KB/s



