Brody Davison

(Note: all of the sound sources used are linked in a .txt file in the same directory as this file)

For my game’s character I wanted the sound effects to be silly and non-realistic in tone. Since the character for this game is the offspring of a 5:00 am game jam session and somewhat goofy in appearance I felt it was appropriate to take that direction sonically, especially because the character can stretch to strange proportions. For that stretching effect I created separate events for stretching out and shrinking the character using the same base sample of a tight bowstring being pulled back. I feel that sound effects for related or adjacent in game events should be sonically similar or share similar base traits, so I used the same bowstring sample but heavily modified the processing on the two tracks. Most notably I added a Delay with Reverseness effect to the shrink track. I had originally wanted to just playback the audio of the stretch sound backwards to create the shrink sound effect, but after messing with the plugin for fun I really liked the end result and stuck with it. I also automate the volume of these two tracks in FMOD with regard to the size of the character, such that the bigger size yields higher volume. I felt this was necessary because it added an additional indicator outside of visual changes to how much the player had stretched the character model. For the dash ability for the player I took a slide whistle sample and combined envelope adjustments with EQ’ing out the lower background frequencies to get the sound. The landing sound for the character was created from audio of a rock hitting the dirt but with some of the high end filtered out to try and eliminate some of the background noise in the sample. The FMOD for this also parameterizes the volume of the fall by the character’s falling speed, such that bigger falls yield a higher impact volume.

I also built a set of sounds for when the player character rolls in the environment. One of the issues in playtesting before this project that arose was that players struggled at higher speeds to determine the rotation speed of their character, so I decided that an audio cue would be a great way to mitigate the problem. One of my favorite games, Chibi-Robo, has a feature where every footstep plays a note from an instrument such as an accordion or a xylophone on a pentatonic scale and I thought this project would be a fun way to explore how the developers might have achieved the effect. For my character’s movement I used a bassoon sample in concert C as the ‘root’ of sorts, then created 5 additional variants of the sample for the 2nd, 3rd, 5th, and 6th of the C major pentatonic chord. I choose a major chord to go with the more cheery and goofy aesthetic of the game. I implemented this in FMOD via a multi instrument that randomly selects from each of the bassoon note samples. In code I used the Z rotational value of my character to determine when to play each note, such that certain angle intervals (such as multiples of 90) would trigger the bassoon sample. For the ambient backing track I took a small sample piece from a synthesizer sound (linked in the documentation .txt) and pitched it down to match the concert C for the bassoon. I then reused the samples to create miniature chords using notes from the C major pentatonic scale so that they would fit perfectly with the rolling track. I added some minor EQ to bring out some more of the low end in the synthesizer and to make more room in the mix for the rolling sound’s frequencies. To give both the ambient backing as well as the roll some more variety, I parameterized the pitch of both tracks with the player’s rotation speed. Rather than a linear curve, I stair-stepped (for lack of a better term) different set pitch intervals for different states of the rotation parameter. I chose this approach because in testing the tracks seemed mostly out-of-tune and I found that being more explicit in the pitch values mitigated this somewhat.

With one exception I wanted my interface sounds to be more realistic compared to the player sounds. I felt that this was appropriate to create a distinction between the characteristics of the game world and the user interface elements. My menu opening and closing sounds were built from a sample of a bubble popping next to a microphone. For these two sounds the recording quality was crisp enough that I felt that EQ was almost a detriment to the sound quality, so I mostly stuck to pitch adjustments for each step of the sounds. The menu confirmation sound was built from a sample of someone shutting a latch on a fuse box. The sample had a lot of echo on the back end of the track so I used a different envelope shape than I’m normally comfortable with to bring the volume down before the echo is noticeable. The only interface SFX exception I made was with the item collection jingle. I decided to keep it lighthearted and within the context of the game world because that is something I have seen other games keep continuous such as Super Metroid with its item collection SFX. To achieve this I used the same bassoon sample as with the rolling sound but with more notes and pitches to create more of a short melody.

Lastly, I recently implemented a pulse / forcefield obstacle into the game so I thought it would be reasonable to create an ambient ‘hum’ to correspond to the effect of the forcefield push on the player. The sample I used for the hum is recorded from the static of a CRT television screen. This track has some small interference and clicks in it but I decided to leave those in because it gives the pulse obstacle the characteristic of being broken or rattling in some way. In addition to adding EQ to bring out the low end I also parameterize the hum by the distance of the player to the object. As the player gets closer, the volume of the effect increases to signal the stronger effect of the pulse’s push. Additionally, getting very close to the object mixes in another layer of the exact same sound sample but pitched up a half-step. I did this to give the hum a bit more depth and to stagger some of the click effects in the track so they don’t seem as ‘regular’ or predictable.