Marcelo Fan Club

Naman Bhushan, Satvik Srinivasan, Nathan Wang

CMPM 151

17 March 2021

# Sax and Violins Audio Critique

Sax and Violins is a 2D twin stick shooter, where the player controls a saxophone wielding musician against an army of orchestral instruments. The audio is closely tied to the game, as the sound effects are mostly diegetic, directly correlating to the actual instruments. To attack, the player clicks on the beat and creates music note bullets from their saxophone. There are 3 different types of notes (quarter, half, and whole) which each have their own properties and sound lengths. The frequency of the notes is decided using 2 sequencers that output midi notes. These notes follow a mostly precomposed melody based off of the root notes of Pachelbel's Canon in D (in C).

Enemies share the same shooting system while using different instruments. There are 5 different enemy types: 4 violins and 1 cello. The violins output quarter notes or half notes that are an octave above the saxophone's on specific beats if they are alive in the game scene. The cello is a boss enemy that ties together the melody by playing root notes from the Canon. If there are multiple enemies on screen, the shooting sound will play at differing octaves in order to maintain the harmony.

The composition is generative in two ways. First, the melody is partially determined by the player. There are precomposed melodies for each of the 3 types of notes - quarter, half, and whole - that run at the same tempo and are played when the appropriate note is shot. Given there are 3 choices for each beat, the number of permutations of possible melodies the player can compose is very high. The harmony is determined by the enemies. There are 4 types of violins, which shoot either quarter or half notes on specific beats, depending on the type. The cello spawns mid-game and only shoots whole notes at the beginning of each measure it is alive. Additionally, Since the enemy type spawning is randomized, the harmony is also semi-randomized. With the generative properties of both the melody and harmony, there are endless possibilities for the composition of one playthrough.

The melodies are translated and expressed through instruments within the patch. The instruments, as mentioned before, are 4 violins and a cello. Violins one and two use a version of Mike Moreno's cello instrument. We made minor adjustments to the cello for personal preference. From what we noticed, a part of its resonance comes from a noise wave with a bandpass on it. The bandpass controls the tremolo of the instrument, so compared to the violin, the cello has a lower one. The violins in addition to having a higher tremolo, have a reverb effect which breathes air into the higher pitches played.

Parallel to the violins, we made a quick saxophone using a tutorial on how to play a C note. To make the patch itself, Q Visible hooked us up with a tutorial on loading in and using patches. Our limitation here was time, and while it's a good skill to learn, soon we plan to either make our own saxophone instrument or utilise an equation which should help read the frequencies of different sample speeds and save them in a sample speed to midi function of some sort. Our saxophone uses sample speeds as an "midi input" in our sequencer, which is why the violins and saxophones use different sequencers although different by only an octave.

Our team used this system in order to create a sense of realism and immersion as almost every sound is diegetic to the game's world. The sax itself is also audio reactive, as it expands when the sax noise is playing. The tempo of the shooting also increases if the player loses half their health, which better connects the player to the game. However, that isn't to say that the audio is perfect. Excluding the shooting system, many more sound effects could have been added to increase the immersion, like death sounds, footsteps, etc..

The note system is also flawed in its own way, since the bullet is created at the start of the beat. This means that the only difference between different notes is what the length of the audio, and how long they can't shoot. If the game was fully connected with the audio, the player would hold down the mouse to shoot and the bullets would appear at the end. Our team thought of this system, but chose not to in order to simplify the system and create more responsive gameplay.

Roles

Naman: Game concept and name, Pure Data note structure, Unity programming

Satvik: Instruments and ½ of note sequencing process in Pure Data, Unity programming

Nathan: Art assets, Unity programming, game design, critique writing.