Theodore Shore
5/16/2021
AME 196

Final Project Report

My final project was partially inspired by a theremin, however, instead of controlling pitch with one hand and volume with the other hand, you control the pitch and trigger sound production with a Wii remote, the volume by moving the mouse up and down, and the pan by moving the mouse left and right. Also, the program uses three different oscillators — a sine wave, a square wave, and a sawtooth wave — and the user can select one, two, or all three of them to play at the same time using the keyboard.

To play the instrument, you select which oscillator(s) you want to use by pressing and holding the A, S, D, Z, X, and/or C keys on the keyboard. The Z, X, and C keys produce pitches a perfect fourth below the A, S, and D keys. The A and Z keys produce sine waves, the S and X keys produce BlitSquares, and the D and C keys produce BlitSaws. You control the gain with the vertical positioning of the mouse cursor on the screen: the bottom edge is zero, the top edge is 1, and with a mouse height of x% of the screen, the gain is set to $(x/100)^2$. The pan is controlled by the horizontal cursor position: the far left edge for a pan value of -1.0 (far left), the edge of the left quarter of the screen for a pan value of 1.0 (far right), and linear scaling between those two points. The pitch is controlled by the angle of the Wii remote: point straight ahead for middle C, angle upwards for higher notes and angle downward for lower notes. To trigger sound, press and hold the A button on the Wii remote, and let go to stop the sound.

My project did not turn out quite as I had intended. I wanted to have more control over the oscillators while playing notes. When I first pictured my project, I intended for the user to be able to add or remove active oscillators while sound was playing by pressing or letting go of keyboard

keys while holding the A button, but in the demo version, once sound had been triggered, the on/off state of each oscillator locked until the A button got released. I believe this was because my play_notes function waits for Wii remote events, so keyboard events never trigger play_notes(). After the performance I was able to work around this issue by creating a variable called wii_position whose value is set by play_notes(), and using that value in set_key_positions() to control a second trigger method for the sound. I also discovered that the panning feature didn't work, even though the pan value was being set as intended, the output sound remained mono. I still do not understand why the panning feature doesn't work, particularly because I used almost identical code to control the gain, and that worked perfectly. I'm starting to suspect that the problem is not with my code, but is instead with the computer software, but I'm reticent to blame something out of my control without evidence, particularly because other programs that use panning work fine. All I know is that the output panning does not match the pan value of the Pan2 objects p[i].pan.

I had very little time to practice or compose because of my heavy finals schedule, but I was surprised by how sensitive the pitch control was on my program. I've played theremins before and this was significantly more difficult to control than I expected. I also found the keyboard and mouse interface to be clunky and hard to use quickly. Maybe that's due to a lack of practice, or maybe it's an inherent flaw in my design.

Once I had more time to fine-tune my project, I was able to figure out a way for my program to respond to changes on the keyboard while notes are playing. If I could spend even more time on this project, I would also like to improve the gain balance between the oscillators, fix the panning problem, and to program better harmonies instead of being locked to unisons and perfect fourths, maybe by figuring out a way to sustain a pitch on one oscillator while playing a different

pitch on another oscillator. Ideally, I could use the B button on the Wii remote to store a pitch, possibly using two instances of each oscillator type to hold the stored and current notes. This project turned out to be more difficult to implement than I had expected, and I still feel like it has a lot of room to grow.