Brian Livingston
5/17/21
Culmination Reflection
Prof. Terao

Project Reflection.

I knew from the beginning of the project that it was very ambitious. I wanted to challenge myself by making a product that I would expect to make a living from. During this time I tried to make sure that I broke it down into tiers so I would know exactly what I could get done.

I begin by using Udemy to take a class in building vsts. It made sense to me and I thought I could get more done using Udemy. Unfortunately, it didn't really transfer over to utilizing the tutorials and JUCE. JUCE is a framework that was made with C++, and the framework basically has many things that you would use to make real-time audio software already in a neat package. You can use it without having to build certain parts from scratch.

An example was memory allocation. It was already taken care of within JUCE and we didn't have to utilize it manually through writing new code for it. You can go into some projects having a skeleton and change different parts about it so that you can utilize it the way you want. Also if you are keen on making it from scratch you could use an override as well and make your own templates.

I was trying to use JUCE but I didn't know that the curve would be so steep from what I was taught. The people that I was learning about it from were the people from Output who actually use JUCE to make their vsts. They were the ones teaching a class so I thought it would be easily transferable but it just wasn't. I think most of the problem was the fact that I didn't know C++.

I know a little about C sharp from coding in Unity and I heard of C before but I hadn't used any of the higher languages very much. What was extremely helpful was going to learn C++.com. It started off teaching how to navigate writing with C++ as a beginner. It even shows beginner level things that you would learn in any programming language such as Hello world, terminology, chapter tests, loops, arrays, classes, object oriented programming, file management and IDE setup for Windows, Mac and Linux machines. It's very helpful even though I haven't gotten all the way to the end of their lessons. I'm on chapter 4 of the book but that was very helpful week after week. I just got better and better at C++.

The other thing that was hard for me was actually understanding what it was I wanted to do. I had never built a VST before and I had no clue what had to go into it digitally. I needed to understand how a synthesizer was made and how sound was processed in JUCE. I was taking a music synthesis class which taught about synths so that really helped.

I wasn't sure if I was going to make anything worthwhile at this point. I'm glad that I had an advisor who had experience with this type of stuff already and he was able to point me away from certain things. He actually was the person who told me to go to learn C++. com, and then he took the project that I originally wanted to make and made a smaller more on my level and doable. We decided on making an additive synth so it wasn't going to be easy for me to do but it was going to be helpful to make it a smaller project.

I created it and it's amazing that I made it that far to having an actual product that I could give to someone and they can play it. This is only the beginning. The other thing is I wanted to pitch this idea to the studio that I work for now. I wanted to pitch it to him because they were getting samples from drummers, pianists and I wanted to package them in a VST.

It would come with samples and then people can give you sound packs and this could be a sound pack product that could be built upon. I've been pitching it trying to work for them and I was happy to know like I can actually make something that someone can have. It's worth it and

i'm thinking later on as I begin to build more and more things that I'll have templates and I'll be able to make VSTs for other people. That's what I want to do. I want to make sure that I can make these things for people.

What I learned that is extremely important is if you're going to learn or do anything, schedule it! Make sure you put a schedule for when you're going to do it because if you don't then something else that seems pressing or even something that feels like it won't affect your time tables will subvert all you planned to do that day. I lost so many days because I didn't have a known pattern of coding and reading days.

I had three jobs, school, and lead Deacon responsibilities at my church. I have to check the mail for the church, count it's offerings, deposit it and field requests for help during COVID which includes going to people's homes. Mind you I don't do this alone but it's a small church and we don't have many people so much is done by few hands. We also began in person services outdoors which required for me to make a schematic to put our sound equipment together for a live outdoor service and come practice setting it up to make sure it works. I am the only person who knows how to put the sound together at the moment.

It was difficult but necessary. I know now if I was put in a situation to learn something all I would have to do is make sure along with milestones for the project on the calendar I put down what days and times I am going to study. Also, having a place to get answers relatively quickly. I found that sometimes I had the right idea but, I was implementing it incorrectly and being told how it is done helps get you to the next place. I want to continue to work on this project because my internship offered me a salary which I took and I want to plan ways in which I can be invaluable to this company. This is only the beginning. I am working towards a small digital empire.