EE 267 Final Project:
Interactive Augmented Reality
Control Panel Music Visualizer

Kevin Johnson, Justin Cheung

June 9, 2018

1 Introduction

The digital music listening experience has undergone vast transformations in the past decades. In our lifetimes, we have seen the cassette, record and CD give way to digital formats. Digital music players such as windows Media Player, ITunes or Spotify usually consist of a list of text for each song, and do little to offer an immersive visual experience. At most, the Windows Media Player visualiser presents a 2D graphic (that is not always music sensitive). We aim to extend these digital music listening experiences using the tools we learnt in EE 267.

Given the inherent visual medium offered by VR and the possibility for "physical" interaction offered by AR and hand tracking, Our project combines mediums to create a visual, physical and auditory sensory music experience. This came to augmented reality in the form of a virtual control panel that allows the user to choose and play songs, visualise them in the form of generative 3D graphs, and manipulate the position and shading of these visualisations.

We chose to develop on the Meta 2 to explore augemented reality with accurate gestural hand tracking, in and out tracking, and a wide field of view. While the headset seems to still be in a beta stage, with difficulties in SDK compatability and hardware initialization, it proved to be an exciting platform to work with.

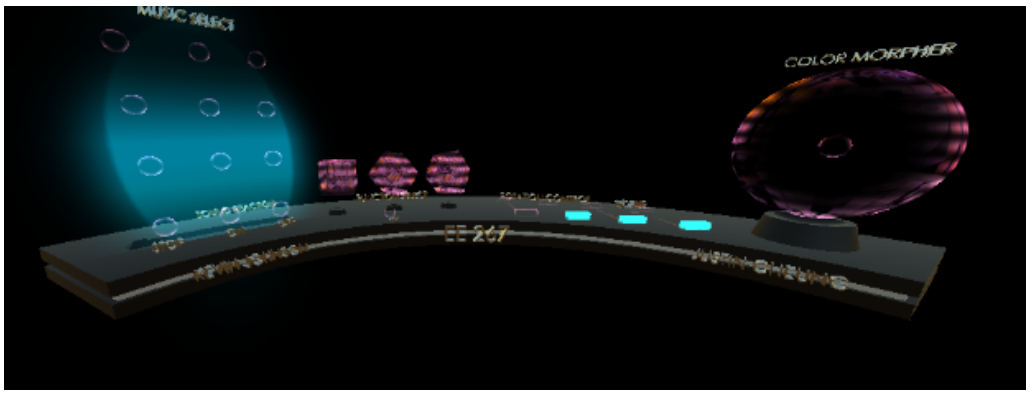

Figure 1 – The virtual control panel.

2 Control Panel Components

2.1 Music Select

To make our project as interesting as possible, we decided to add an augmented reality juke-box/music selector. Pillbox mesh renders were used for each of the interactive buttons. The user could activate each song by grabbing each pillbox shaped button with their hands. The music player had the following songs and functions:

Song selection

The user can select from 9 songs:

1. Instant Crush, Daft Punk
2. Happy Violence, Dada life
3. Dancer in the Dark, Marc Philippe
4. Tron SoundTrack
5. Foreign Language, Flight Facilities
6. Der Komissar, Falco
7. Bad Boy, Red Velvet (instrumental)
8. Africa, Toto
9. HomeTree Avatar

Sound Off

This function allows the user to turn the current audio source off.

Sound Reaction

The user is presented with two buttons that turn on/off the sound reaction of the visualizer. Inside the main graph visualizer script, a function was constructed to pull the amplitude of the audio signal from the audio source playing in the environment. Then, boolean logic was written to add this amplitude to the current function being displayed.

2.2 Function Select and 3D Graph

The main feature to our project was a 3D visualizer consisting of the following three graphs:

1. Ripple Graph
2. Cylindrical Time
3. Spherical Time

These were algorithmically generated functions formed from multiple unity clones. The graph was generated by first instantiating cloned cubes upon starting the program. Next, two for loops were used to determine the cubes position based upon what function the user selected.

2.3 Sliders

Four sliders were implemented in the control panel. These were positionally (1D) clamped unity objects whose position scaled a corresponding variables. The four sliders controlled the following:

1. **Rotate X Scale** - Rotate the main graph about the x axis
2. **Rotate Y Scale** - Rotate the main graph about the y axis
3. **Rotate Z Scale** - Rotate the main graph about the z axis
4. **Scale** - Scale the entire graph

FFT Graph (Circular Spectrum Analyser)

Upon initialisation, a circle of prefab spheres is created above the sliders. A smaller, rotating circle of spheres resides within it. This is a circular spectrum analyser. Each sphere corresponds to a frequency bin of the sampled audio. Each sphere's radial distance from the center is scaled to the amplitude of its corresponding audio frequency at each render update. To create a spectrum analyser-type visual, the radial distance is clamped at a maximum value. This is rendered instantly, that is, the sphere can move from min distance to max distance in very few frames. This creates the 'maxed out' effect of a certain channel. After a channel hits this maximum, the radial distance decays linearly to the original radius to create a smooth return transition. The frequency binning is implemented using an AudioListener component. Each render update listens for the current audio output, performs an FFT on it, and creates an array of the amplitude of each frequency bin.

Color Sphere (3D Slider) and Custom Shader

The colour sphere is implemented as a floating sphere clamped in all 3 directions, allowing it to be moved within a clamping sphere. This manifests itself as a 3D slider.The X,Y and Z distances from the origin were each used to scale a global shader values. We wrote a custom shader that scales RGB Albedo values by the vertex's distance from the object's origin (this required multiple transforms from world-model space, as the shader resided in world space). The magnitude of this scaling was set by the global shader values set by the 3D slider. A fourth (dependent) variable was extracted from the 3D slider. The magnitude of the sphere's distance to centre was used to scale the reflectivity value (in Unity it is named "metallicness") used by the shader. This shader was best visualised by the graph visualisations, as each of its units had large positional variance. The shader created dynamic colour spectrums across the graph.

3 CAD

The Control Panel and its module were created in Rhinoceros 3D (architectural modelling software). Rhino has the advantage of being able to export OBJ files that can be imported into Unity and maintain the grouping structure of the original Rhino file. Materials and textures applied in Rhino can also be carried over. Note that one must be careful that the normals of each surface point outwards upon export, otherwise Unity's shaders will not appear to render anything (shaders will be shading the interior surface of the mesh).

The 3D text was created in Solidworks using the Extrude Text feature. Each word was imported into Unity as a mesh and had a metallic material applied to it.

4 Conclusion

Many difficulties arose throughout the project which were quite frustrating at first. One of the major roadblocks was finding a computer that could support the Meta 2 Headset. Next, the most challenging part of the project in our eyes was writing the process to enable the interactive control slider. It was extremely difficult to debug because it would work sometimes and not the other, but once we finally figured out the issue we were relieved.

All in all, this project allowed us to learn a great deal of useful skills on constructing an augmented reality project. First, we had to go through many tutorials to learn how Unity works. Next, figured out how to pass variables between files in C# and javascript such that they were easily accessible in Unity's graphical user interface. Finally, we had to figure out how use Meta's libraries to add interaction to all of our objects. This project was a great opportunity to solidify the material learned throughout the quarter such as rotation, translation and scale matrices, clipping planes, multiple coordinate system hygiene, shaders and much more.