

Daniel Willborn

Creative Computing

12/9/2019

Capstone Report

Introduction:

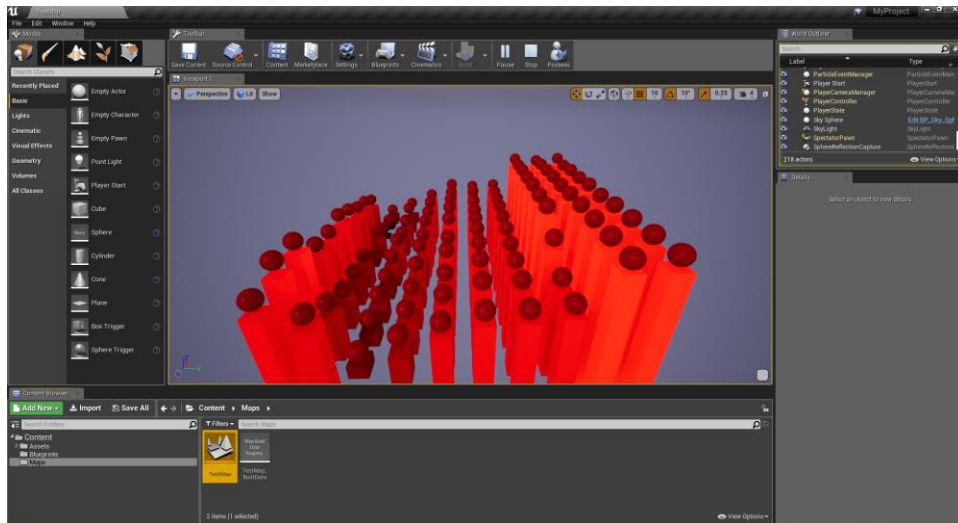
The initial idea from this project was to create a three-dimensional environment using Unreal Engine 4 in which object-based structures would be procedurally generated over time based on a user audio input as well as a user mouse/cursor input. Steps toward this plan began with trying to understand how audio visualization works in Unreal Engine 4, which was the basis for the first iteration of the project.

Process:

The first program I created was meant to be a learning experience for understanding the Unreal Engine 4 framework and how audio interacted with objects in a created level/scene. I used a YouTube video tutorial to help kick things off and I worked my way from there. What resulted for this first portion was a program that visualizes an input song based off its FFT channel values.

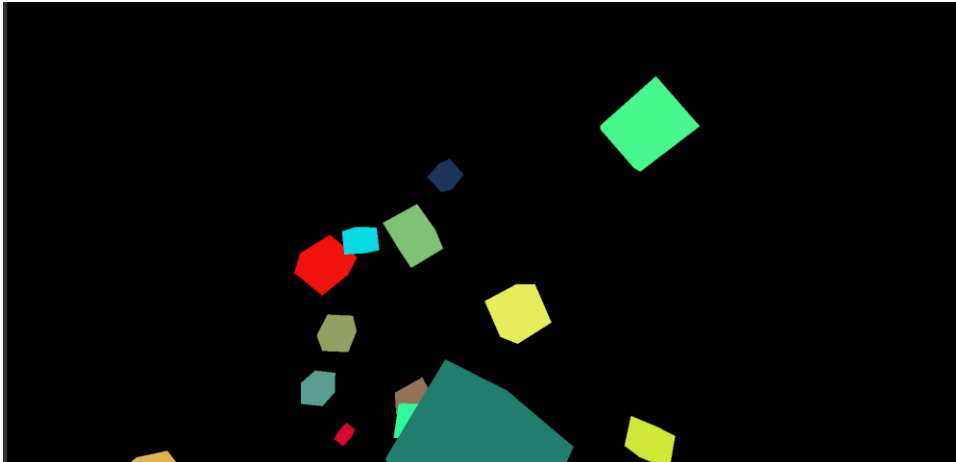
The next step was originally to learn how procedural generation works in UE4, however most references I looked at were describing how to make procedurally generated levels as opposed to structures, and I was not having a very strong grasp of the concept or how I could transfer level generation to structure generation, so I instead looked at how physics worked in UE4, and changed my idea slightly to incorporate physics in some way. I learned how to create physics objects and physics materials and applied that on top of the sound visualization project I previously created. What resulted was a project where balls are being launched off of the

visualizer and bounce based off of the visualizer's and balls' physics properties, and it looked something like this:

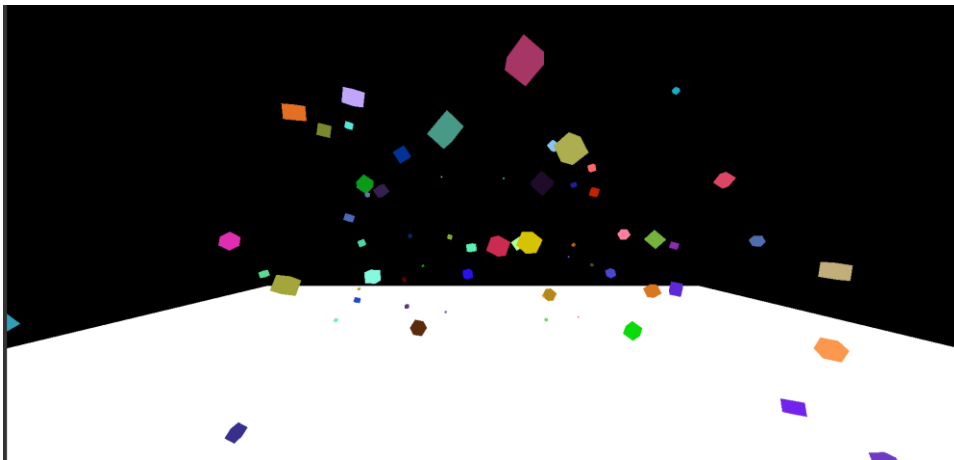


I figured the next step in this project would be to learn how to make dynamic audio inputs work in UE4, however it turned out that the process of making that work was very complicated and involved delving into code that was too complex and unintuitive. Seeing as how dynamic audio inputs would be necessary for a project like this, and that the process would likely not be feasible, I decided to switch frameworks entirely and work my way from the basics back up using THREE.js and TONE.js. This seemed like a better environment to work in, as I was more familiar with JavaScript and web-based application development.

The first step towards using the THREE.js tools was to learn what the framework had to offer, which I started by walking through the tutorial on their website. After that, I set out on a first assignment. These assignments were made to set a short-term end-goal in mind and work towards it while building up to the final project. The first assignment I set was to create a program that spawned cubes on the screen and had them fall towards the bottom with some type of physics. This was the result:

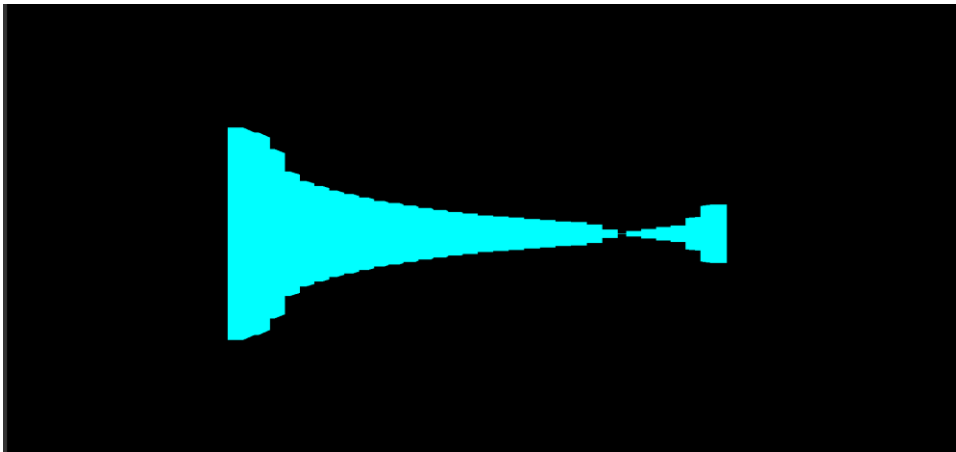


The next steps for this project would be to just keep moving forward, creating assignments that added new features to the programs and helped me learn new aspects of THREE and TONE. The second assignment was an iteration off the first assignment, where cubes would spawn at the mouse point on click, spew them in any XYZ direction with a random velocity, and have them fall down and bounce off of the floor (a plane at the bottom of the screen). On impact, these cubes would make a noise depending on where on the floor they landed, with the floor being divided into 4 quadrants. This was the result:

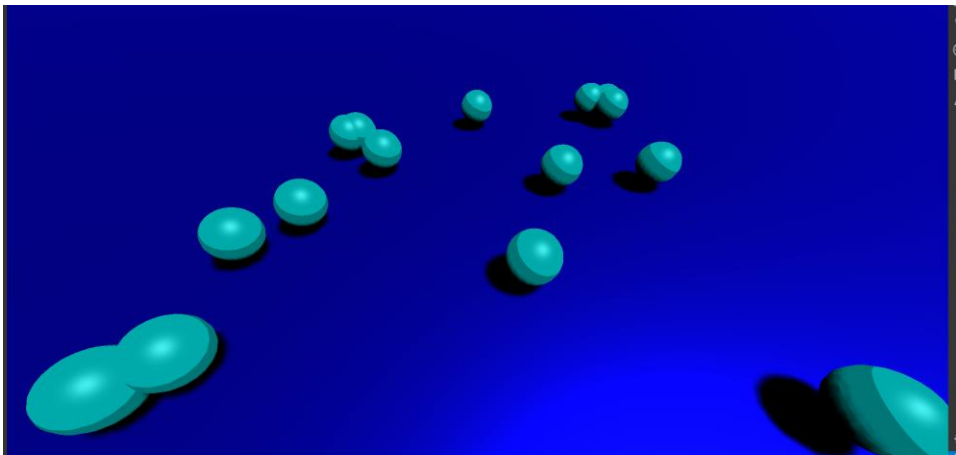


The third assignment was a simple project used for learning the “phaser”, “sequence”, and “FFT” objects used in the TONE.js framework. The goal was to simply create an audio visualizer like the very first program I made with UE4. Notes would play in a sequence from an

array of notes, and a group of cubes would have each height change depending on the FFT value from each channel. This was the result:



The fourth assignment was really the first iteration of what the final project was going to be. I wanted to have a program where the player has to hover over objects on a scene. The object would then sink into the ground and make a noise. Once all the objects were sunk, the scene would reset. This was the basic idea for the project, and I would want to add a gamification aspect to it in a later iteration. This was the result:



The final project iterated off the last assignment. Elements I added were:

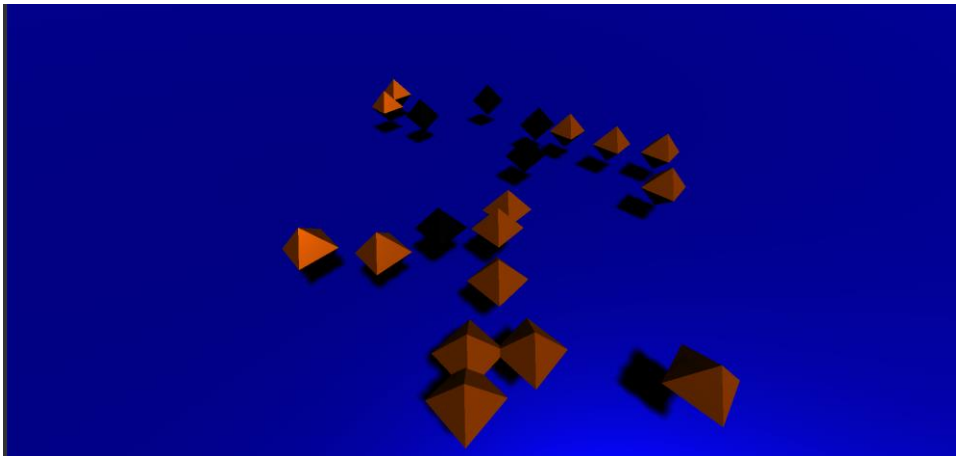
- the objects would be tetrahedrons
- a looping set of chords would play in the background using a sequence

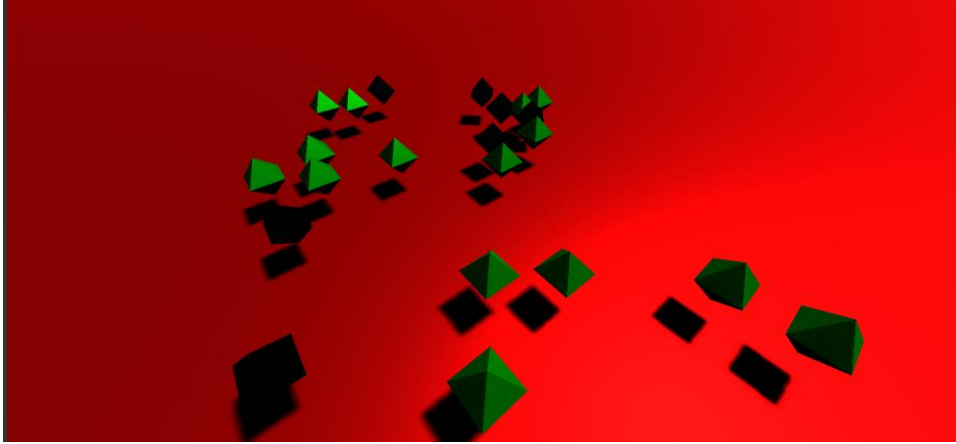
- When an object is hovered over, instead of making a random sound, it would play a note in harmony with the current chord being played
- the colors of the objects and the background would always be complementary
- when the objects were all gone, a smooth scene transition would take place, where the new scene falls from the top and gradually sets in place.

Additionally, I decided to add minor game elements to the projects. These would be:

- A score system is implemented. Each note passed over adds 100 points to the score
- Sour notes are added. When a sour note is hovered over, a bad sound plays and the score is reduced by 300 points.
- The better the player does, the more sour notes are added to/remain on the screen.

The end result looked like this:





Reflection:

The end resulting project was not at all what I had originally set out to do, and I can attribute this to the framework shift. However, the final project I created I believe to have a strong concept and could be greatly improved upon in future iterations. Things I would add would likely be: to make the background more visually interesting, such as adding a water shader that changes color, giving the appearance of being in the middle of the ocean; making the music generation more dynamic as opposed to having a set chord progression on a loop; and improving the gamification aspects of the project, such as including a timer, high score, leaderboard, etc.

The most obvious difficulty in this project was the fact that I completely changed frameworks halfway through. If I had used the THREE.js and TONE.js frameworks from the very beginning, I would have learned more about the frameworks and been able to implement something with more features. That said, I am glad I switched frameworks. There were many difficulties figuring out how to implement certain aspects of Unreal Engine 4, and since my project wanted to have some sort of dynamic audio-visual interaction, Unreal was not the optimal choice. Unreal Engine 4 has tools for audio visualization, but not much else. There was not much more room to expand on the concept I wanted to execute using UE4, as doing so would have a pretty insane learning curve, and I would not have finished by the deadline if I had stuck

with it. There were many pros to switching to THREE and TONE. I was moving into a framework I would be much more comfortable with, as I am much more familiar with JavaScript, and these frameworks had tools specifically for the type of project I was attempting to achieve and allow for much easier development down the line. However, by changing to these frameworks, I would inherently be unable to make a project that was in line with my original idea, and this is what led to the shift in the overall project design. However, with more knowledge of these tools, it could be possible to execute my original idea using these frameworks. In conclusion, although my project was nothing like what I had originally set out to do, it still resulted in something interesting that can be further developed upon, and it helped me hone my creative skills in a way I hadn't done up to this point.