Project 1 - Platform Simulation:

Tristin Greenstein
10/6/2021

Problem:

Design and develop a software system that will simulate Kinetic Art for a mechanical hydraulic system. Using user assigned M and N we shall create a platform made of cubes that will change its y axis randomly within the bounds set by the user as well as change its colors on trigger.

Objective:

To increase understanding of Game Objects in their relation to script functions, the control of multiple game objects originated from a script, as well as cycling through color schemes.

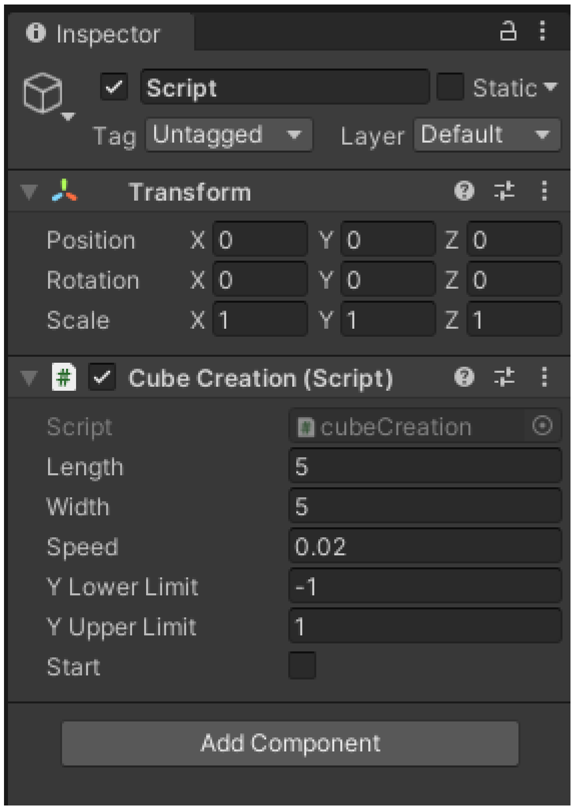
Over the course of this assignment I ran into several roadblocks. Some of which required help from the professor as well as my peers to overcome. The nature of these issues mostly coincided with unfamiliarity with C# language or the Unity Software. I will now go more in detail about my experience developing this application.

When I first started this assignment I generated the cubes based on our previous homework Building Blocks. Using that as the template, I quickly was able to generate a M by N matrix of cubes. This is where I reached my first issue. I was unfamiliar with creating arrays of type GameObject as well as initializing it as empty with no set size. I reached out to the professor and was enlightened on how to move forward. I was soon able to make the M by N cubes be generated, transformed, positioned, and stored in the array.

Next after this I transitioned to trying to make the cubes randomly move on there Y axis. I first started with trying Lerp but was unsuccessful due to my unfamiliarity and unclear documentation. I then decided to recreate Lerp but using my own methods. I broke this down into several parts. First the code looked at my boolean array called cubeMoving, if this was set to false it then called the method nextPosition that randomly filled the array with the next y axis position of that individual cube. It then set cubeMoving to true. Back to the update method, the next if statement checked if cubeMoving was true. If true we then moved on to the movement method that would transform the cubes position but the public speed variable. It would keep calling the movement method until the current cube position matched the next position. cubeMoving would be set to false and the process would restart. The development of this process was filled with minor errors, such as all the cubes moving at the same time or the arrays not updating in sync, but overall this process did not cause major issues.

Next we moved onto the color requirement of this assignment. Funny enough this caused me more issues than the movement code. I tried the same method I used for the movement of the cubes but it caused immense lag for 5 minutes(depending on how many cubes initially generated) and then the color would instantly change. This highlighted two issues. One that the color instantly changed and not gradually, and two the immense lag. I next reached out to the professor for help and he tried to get me to understand Color.lerp. While I saw his thought process and saw how he implemented it in his code, I still had difficulty implementing in mine. With the help of a fellow classmate I was able to rework my code to integrating lerp into my existing movement method. By having the color change at the rate of movement that currently had no lag, I was able to negate any possible issue. But then I ran into a minor problem. I currently had only hard coded this to work on the red spectrum, how do I allow the other colors required for the assignment? I had to step back and create a new global color value to tell the color changing method which color scale to randomly generate. Once this was implemented I was finally complete.

Overall I rate this assignment medium difficulty. Less hard than the first opengl assignment but it still had some hard points.



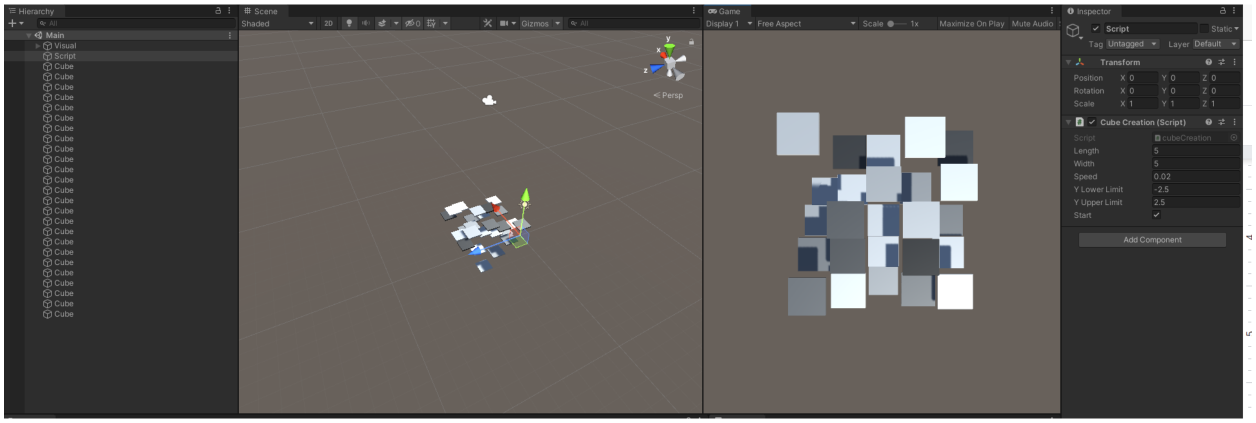
✓ Script ☐ Static
Tag Untagged Layer Default

Transform

	X	Y	Z
Position	0	0	0
Rotation	0	0	0
Scale	1	1	1

✓ Cube Creation (Script)

Script	cubeCreation
Length	5
Width	5
Speed	0.02
Y Lower Limit	-1
Y Upper Limit	1
Start	☐



```
Main
  Visual
  Script
  Cube
  Cube
  Cube
  Cube
  Cube
  Cube
  Cube
  Cube
  Cube
  Cube
  Cube
  Cube
  Cube
  Cube
  Cube
  Cube
  Cube
  Cube
  Cube
  Cube
  Cube
  Cube
  Cube
  Cube
  Cube
```

```
Script                                  Static
Tag Untagged        Layer Default

Transform
Position     X 0     Y 0     Z 0
Rotation     X 0     Y 0     Z 0
Scale        X 1     Y 1     Z 1

Cube Creation (Script)
Script           cubeCreation
Length           5
Width            5
Speed            0.02
Y Lower Limit    -2.5
Y Upper Limit    2.5
Start            ✓
```