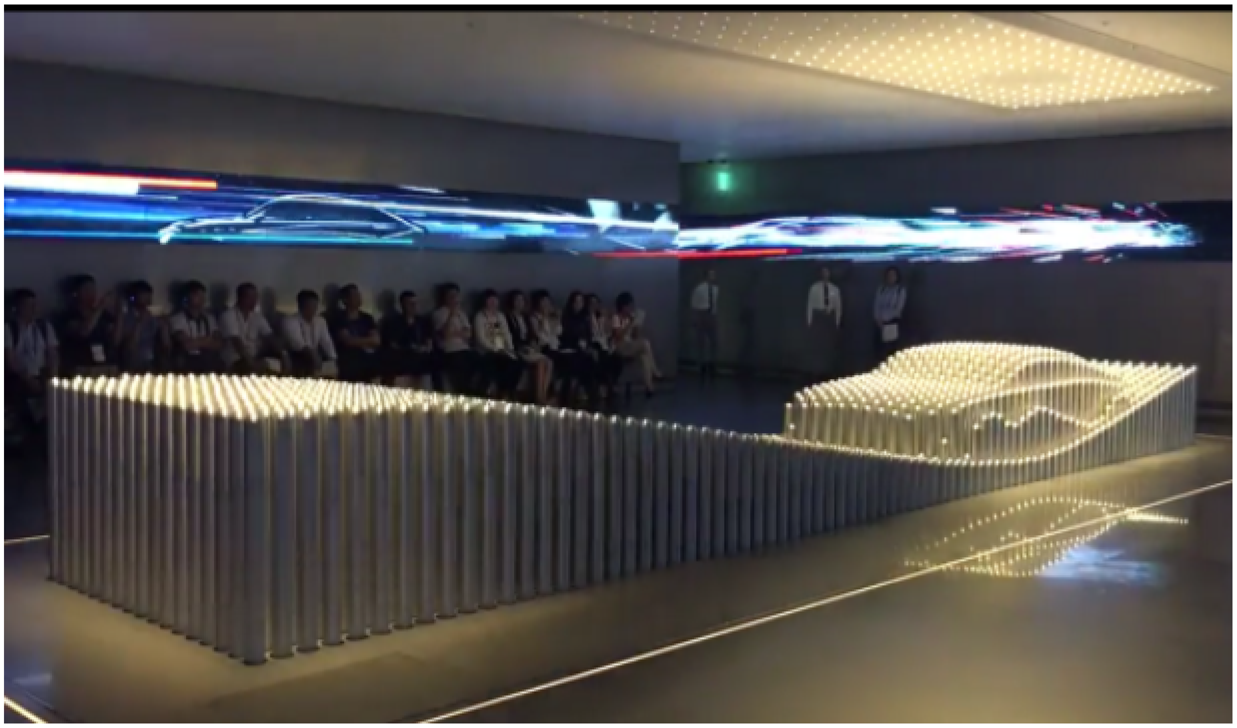
Project - 3 : Platform Simulation - Final

Tristin Greenstein
12/17/2021

Introduction:

This whole project first started with project one being due October 6th. In the seventy plus days since the beginning of this project I have learned several skills and improved my understanding of the various functions of Unity, the application of its various tools and capabilities, as well as develop a deeper understanding of C#. Using Visual Studios development platform in collaboration with Unity, I learned how the runtime and various methods and attributes such as events and delegates affect the overall platform of the application.

This project was to emulate a mechanical hydraulic system that is fed in a design and runs it on a loop through its pistons. The image of such a real life function is shown below:



The Platform Simulation Project was broken down into four parts. Project 1, Project 2A and 2B, and lastly Project 3 due on December 17th. While we will cover Project 3 separately in this report, we will cover all aspects of this Project.

Project Overall Objective:

The Project's overall objective is to create a MxN dimensional matrix that is generated either through user input with the User Interface created, or through reading input from the generated text file created in Scene Programming. From here you will be able to select one of the nodes from the generated matrix and change its height with the User Interface. In the following scene, any nodes selected and height changed will be circulated on a loop across the matrix in a simulated manner.

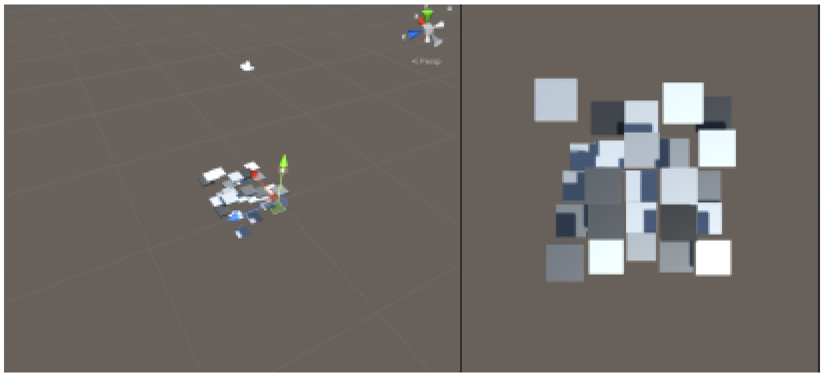
Project 3 Objective:

Project 3's specific objective is to read the generated text file from project 2b, generate the MxN matrix from the data in that file, and to create a looping simulation based on the heights selected in the previous scene and stored in the text file.

Requirements:

Project 1:

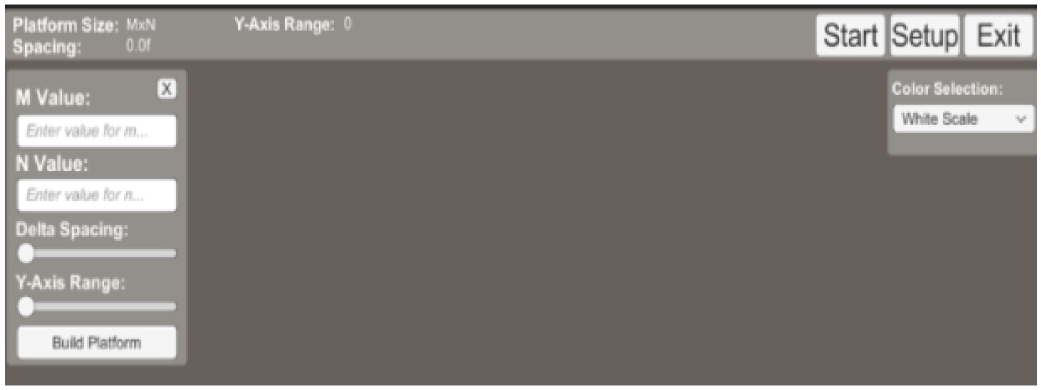
- Configuration of the MxN platform
- Spacing between each base unit for the platform
- Input Keys for Testing Features



Project 2:

2A:

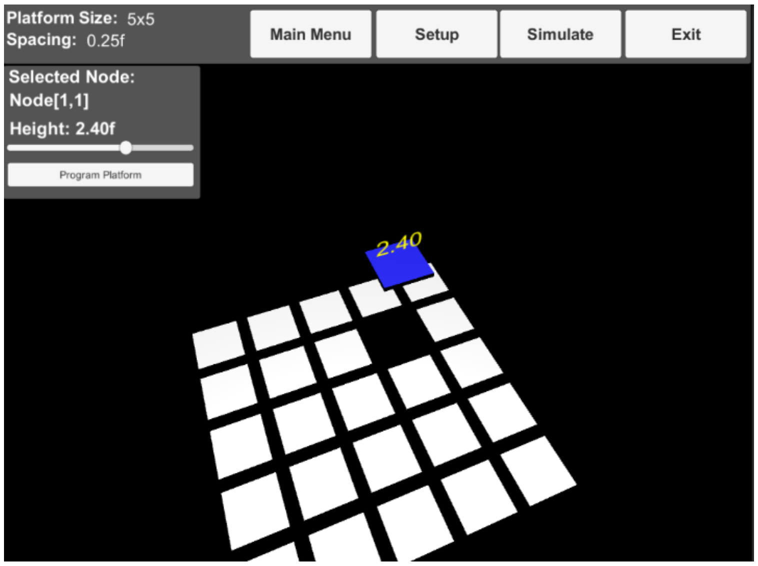
- User Interface Panels for the creation and manipulation of the matrix and nodes
    - M & N Values
    - Delta Spacing
    - Y Axis Range
- Change the original code to allow the manipulations of the User Interface in real time.



2B(Rewrite of entire project):

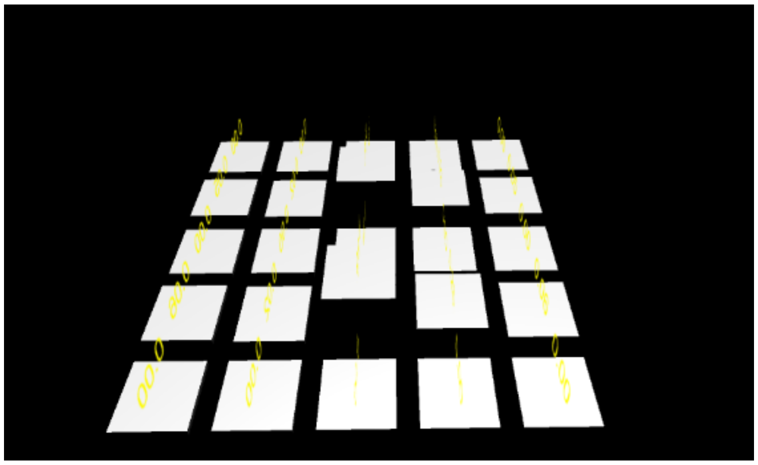
- Rewrite Project to work off the event delegate system
- Break down functions into four separate scenes

        - Main Menu, Setup, Programming, Simulation
            - Simulation will be covered in project 3
    - Establish Programming Scene Logic
        - Main Menu and Setup logic created in project 2a
        - Selection of a node and changing its height.
            - Store info to be written to file
        - Write all programmed nodes to file to be loaded in project 3.



Project 3:

- Load the program file
    - File Structure:
        - <u>Header</u>: Will contain the configuration for the platform, dimensions, and spacing.
        - <u>Body</u>: Will contain the data that will be simulated.
- Create platform based on the configuration data
- Start simulating the programmed data



<u>Process Of Project 3:</u>

Project 3 started off with using the code handout given on the assignment page. With some null checks as well as changing naming convention for scenes and buttons, I was able to make it functional. Functional in this case means four separate scenes, User Interface that allows switching between scenes, two more separate User Interfaces in which the first one handles input to generate the matrix and the second one handles node selection to change height. With help from the code the professor shared in the lecture & lab on 11/24, I was able to quickly set up a basic read from file and the structure for some of my classes. As this project was only focused on loading text files and simulation, that made the current scripts PlatformCameraControl, PlatformGenericSinglton, and PlatformConfigurationData unimportant and not necessary to be worked on. In fact I also made only minor adjustments in UIManager as well. The majority of this project only focused on PlatformManager and PlatformDataNode. In the UIManager what I added was another event to trigger on the simulation scene to trigger the simulate variable and turn off the programming process. What also is triggered is removing all nodes from the scene, a method created in the previous project. This is when the matrix is generated from the file created in the programming scene. The method reads the file, sorts it into variables, then feeds it into the existing BuildPlatform method. After this, the update method cycles through the array of nodes and tells it to simulate, and move to the height now stored in the NextPosition variable. Finally it shifts to the next node after returning to zero height. Unfortunately I keep running into a weird issue where it seems that the runtime of the nodes are affected by computer speed. That the nodes wont run the same with the exact same file. It will do the first 3 shifts correctly and then backspace and then jump. I have a educated guess that the program is waiting for the other node to finish going to zero before simulating it again but it also should not be going to the next node if it isnt finished simulating the first. While the simulation is functional, it does present the previous issues. It also presents an issue when choosing two nodes that do not have a space between them. I am unable to correct these issues at this time.

The best success is when using this writefile for some reason:

```
5,5,0.297561,2.185366
0,0,0
0,1,0
0,2,0
0,3,0
0,4,0
1,0,0
1,1,0
1,2,0
1,3,0
1,4,1.22463
2,0,0
2,1,0
2,2,0
2,3,0
2,4,0
3,0,0
3,1,0
3,2,0
3,3,0
3,4,1.10093
4,0,0
4,1,0
4,2,0
4,3,0
4,4,0
```