Nash Domenichelli

T00562771

Comp 3820 Computer Graphics

April 13, 2021

Description

My project consists of two parts. The first part is a two-dimensional wire that you can rotate in 3d by dragging on the screen. The second part is the actual final object with texture mapping that I created using the first application. As far as what the project looks like, it is a small tree with texture mapping inspired by Super Mario Sunshine (2001), and it has that retro feel. On the final version of the project, you can zoom in and out on the object by using the scroll wheel. The tree object is one set of Bezier curves to form an island, a tree trunk, and then the leaves. It does not look perfect with some of the transparent textures but, in terms of how it feels it has the same idea of what I was going for.

Some extra features I would have liked to add is water around the island and the tree being able to bend like it is in the wind. The problem with the wind feature is how I have the tree drawn; I cannot make it bend one way because if I were to pull a point on it would make the whole tree thinner and not actually translate the object. Adding water, I would ideally make a horizontal plane with a water texture, but I could not add this due to time constraints.

Changes from Proposal

My proposal had a lot more ambition and a lot more details but no hard concrete plan of what I wanted to create. The main thing that stayed the same is that I wanted to make something artistic rather than an interactive game. By choosing an artistic approach, there should have been more details and more objects to fill the scene with. Originally, I wanted more camera movement as well where the user could move the camera anywhere but when it is only looking at one object, its not necessary.

There were also ambitious features I wanted to add like mainly water effects, reflections in the water, and even non-Euclidean elements. These features seem out of the scope of the project but were interesting to research. One project I downloaded and looked at was a demo using non-Euclidean geometry in OpenGL and it seemed somewhat like what we were doing but, not similar enough to easily convert to WebGL.

Requirements

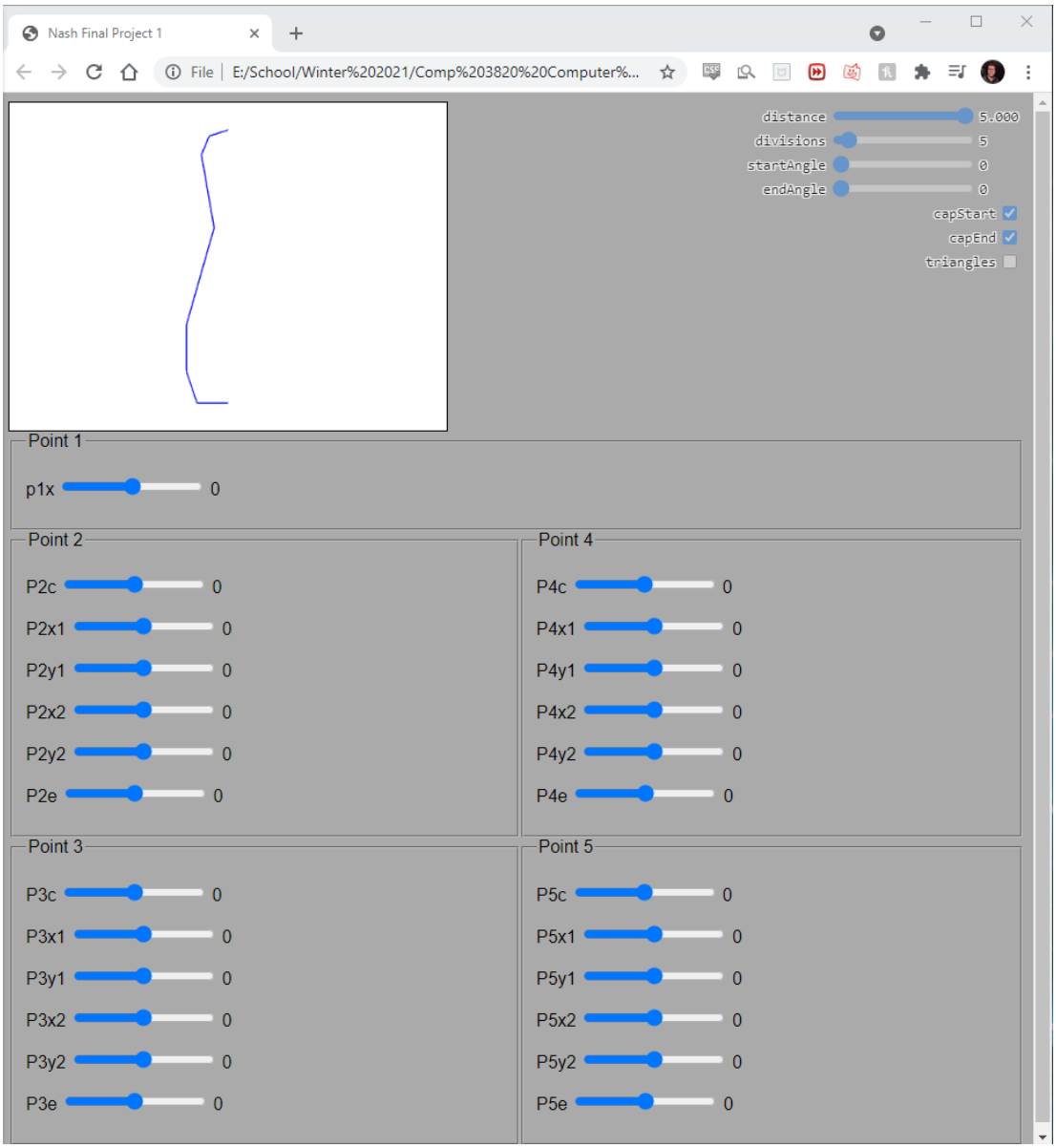
The main requirement was to use WebGL, and I based my project off the Lathe tutorial on webglfundementals. I created my own 3d object with this tutorial and created my own function to zoom the camera in and out with the scroll wheel. Unfortunately, I could not add lighting, but I do have user interactions like dragging the camera around and zooming in and out. I also created the texture I used by finding the texture from the original game, and then editing it to work with WebGL. For the first part of making a 3d object editor, there is a lot of user interaction there by being able to edit the points, as well it being creative and unique.

Instructions

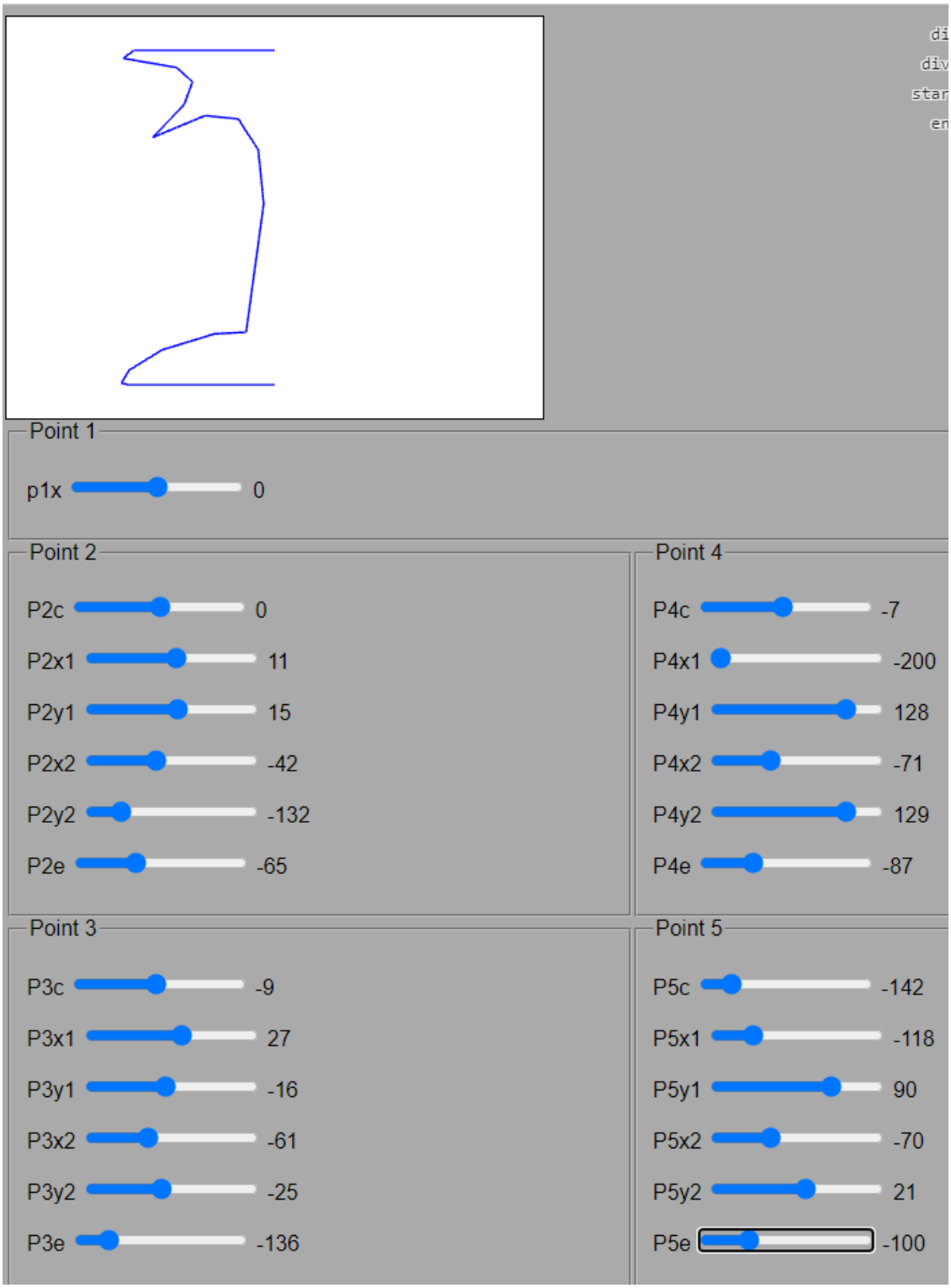
Unzip the NashProject folder and run it with servez. There are two main files called editor.html for the editor I made and created the tree object, and then with that I copied the slider values into the project file. The project.html file is the main project, but I thought submitting the editor would be useful too because the tree is not that impressive on its own. On the project file you can click and drag the object to rotate the camera around it and use the scroll wheel to zoom in and out. There are also UI components to change triangles and other math involved with the curves but those are left over components from the tutorial I was using and there are issues if I try to delete them.

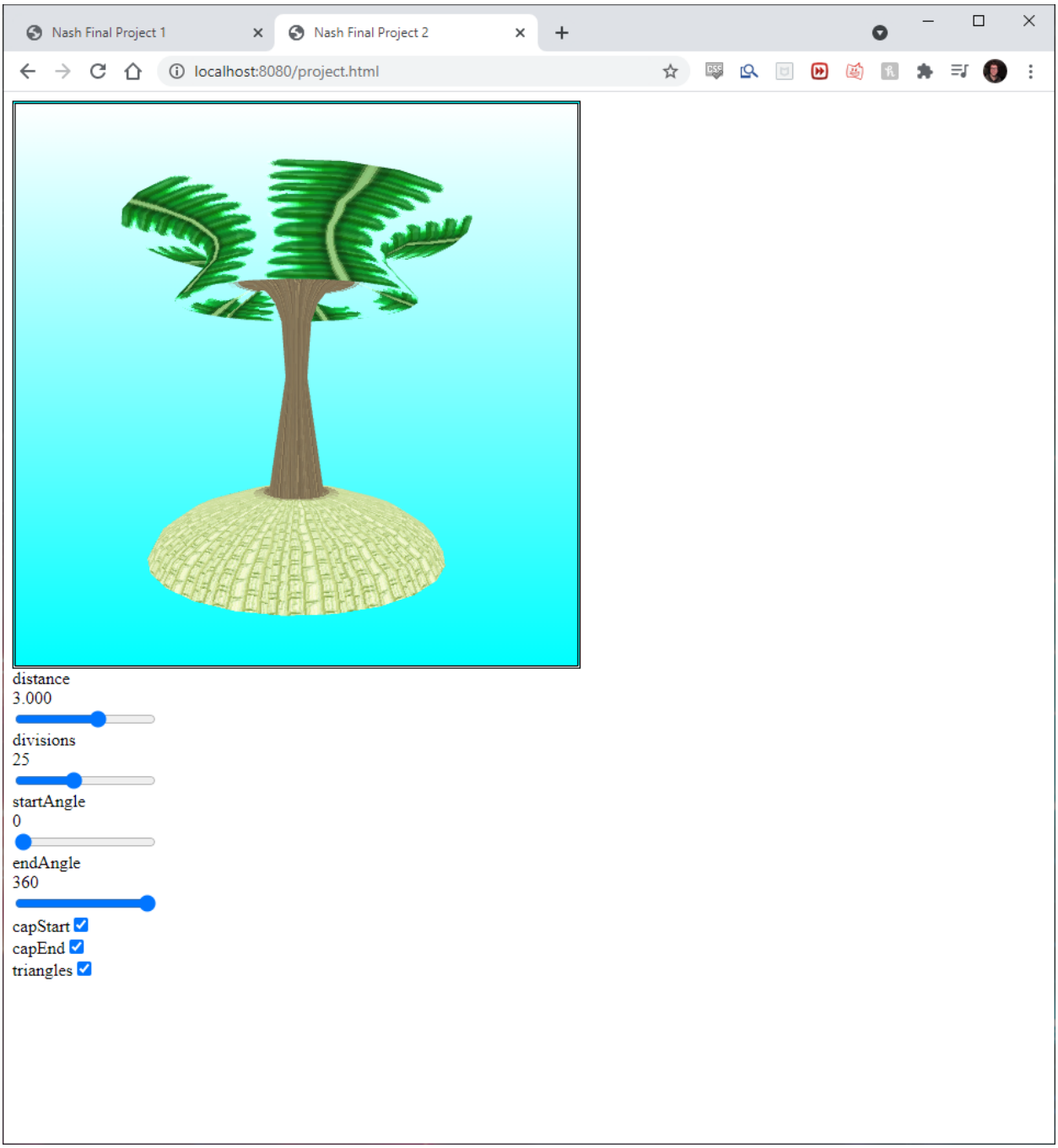
Screenshots

Editor.html

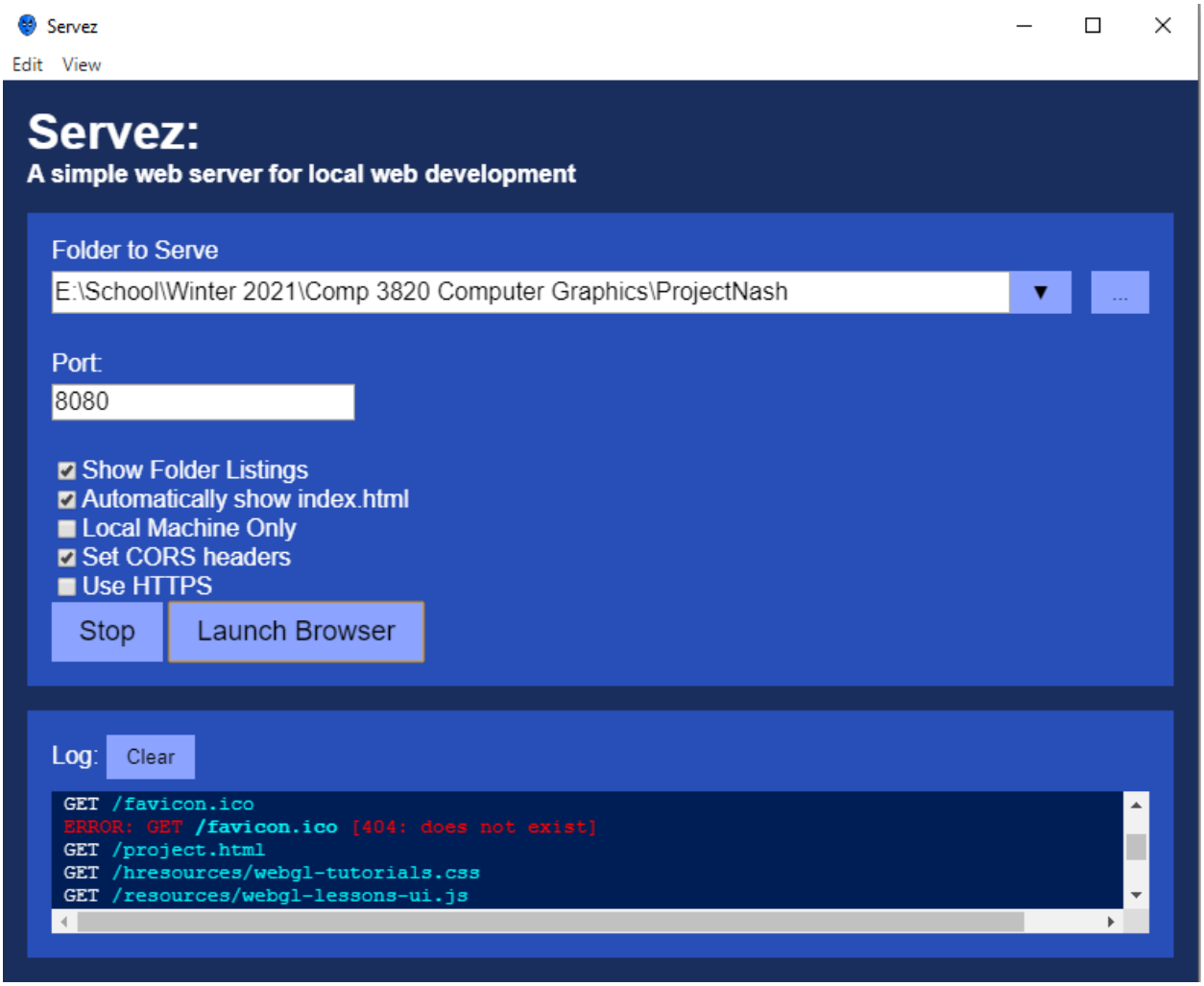


Editor.html with tree numbers to create curves



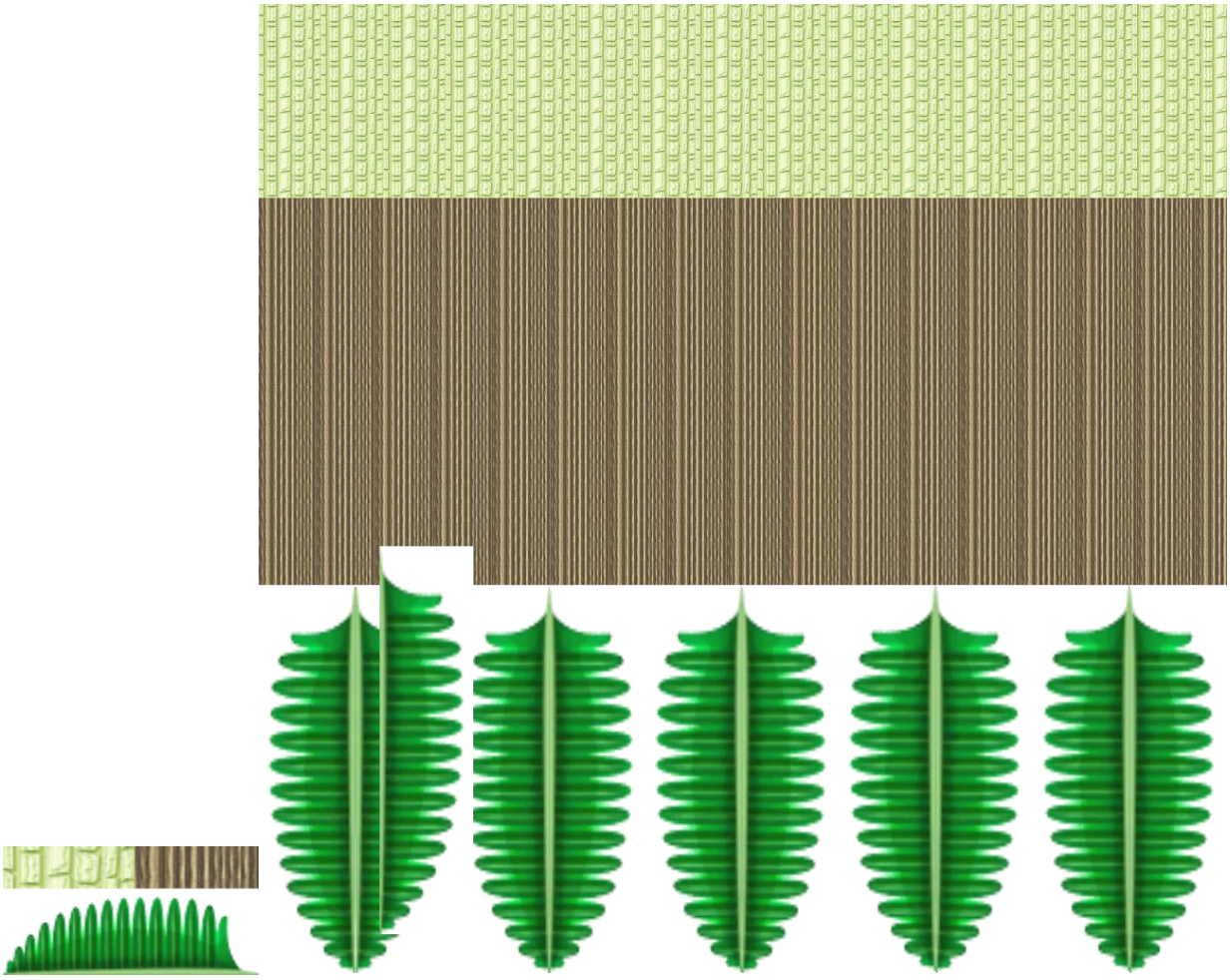

Nash Final Project 1
Nash Final Project 2
localhost:8080/project.html
distance
3.000
divisions
25
startAngle
0
endAngle
360
capStart
capEnd
triangles

My Servez settings



In game screenshot of tree texture I used

Original texture from the game (left) and edited texture I used (right)



What I learned

WebGL is interesting in the sense that it is low level, and you must program things exactly and there are not a lot of helper functions. WebGL is also surprisingly fast to use and very powerful because it is so low level. It allows you to add a lot of objects with no performance issues, as seen in some demos and examples.

There also seems like there are more intuitive and user-friendly ways to create what we are doing in WebGL such as three.js or using something that is more common like OpenGL. It is good to know the basics, but it can be very challenging to program low level functions.

When I signed up for the course, it was originally supposed to be Virtual Reality but because it was online, this was also a fun experience even though I think virtual reality would have been more fun.

References

https://webglfundamentals.org/webgl/lessons/webgl-3d-geometry-lathe.html

https://webglfundamentals.org/webgl/lessons/webgl-3d-perspective.html

https://www.models-resource.com/gamecube/supermariosunshine/model/33340/