Logan Barclay

Connor Nagel

CS464

12/15/2017

**Final Project:**

**Mario 64**

**Proposal:**

**Abstract**

We would like to re-create the first level from the Nintendo 64 game Mario 64. The implementation would be similar to the 3rd assignment we did. We could create a camera that either follows Mario around, or flies through the level giving a bird’s eye view of the level. We would like to update the original version by using more polygons and some of the advanced shading and lighting techniques we have learned. If time permits, we could add actual game features such as moving enemies and a Mario that can jump.

**Implementation**

We would like to use Blender (or a similar program) to model the original 3D terrain from the game. After some research, it appears that we can draw the model in Blender, export it to a OBJ format and use Three.js to render the model in WebGL. We’re not sure to what extent we can update the graphics with the modern tools available to us, but better textures combined with advanced lighting and shading models, we should be able to make it look much more modern.

I did not get my camera working properly in the 3rd assignment, so I will need to fix that. Once fixed, we would use a refined version of that for the camera that follows Mario around the map. We can create a Mario avatar that is centered in the camera’s view and implement the jumping motion to get over obstacles.

Another area we would need to research would be how to model things such as bridges and trees. So far, we have only done one consistent plane, nothing with a potential underside or multiple layers. Incorporating something like a sun in the sky could really help to bring the level to life.

**Note:**

We are open to other project ideas. If we are not allowed to use Blender or Three.js, We have a backup project in mind to create a music visualizer. We think we can implement the music visualizer with just WebGL, but haven’t looked into it too far. We would like some feedback on whether or not you think this project is too ambitious.

**Screenshots:**



**Status Update:**

So far, we have a model of the terrain in Blender and a few rough character models. We are currently researching ways to import the .obj file into WebGL. Three.js has a way of doing it, but we haven’t implemented it yet. We are also looking into how to apply custom textures to the character models. We plan to import the file as the geometry in the A3 code to see if we can get a camera on the ground. We haven’t started any of the texture or lighting work, but those will be the next steps once we have a working geometry.  
  
   


**Final Report:**

**Final Implementation:**

We weren’t able to implement all of the features we wanted with this project. I think the modeling in Blender kind of caught us both off guard with the fairly steep learning curve. We ended up using most of our time modeling the level and characters. For the camera, we weren’t able to get Three.js’ first person camera working properly so we stayed with the “flying perspective” camera. There is not any collision detection in the level, so you are able to fly through the objects and walls. As far as updating the graphics compared to the Nintendo 64 version, we were able to use Blender’s smooth shading on the level (MarioSmooth.html). This is a step in the positive direction, but Phong shading and more advanced lighting would really make the level look more modern. We would have liked to increase the triangle count to 5,000 or so to really smooth things out, and use more photo-realistic textures as well. This project was definitely larger than we expected it to be, and we now have a newfound respect for game developers.

**Experience with Blender:** Logan Barclay

This project was fairly ambitious for us since neither of us had used Blender before. I already had it installed on my computer from a previous failed attempt to learn it. A good friend of mine is well versed in Blender and makes his own games using Unity 3D. He does all of his modeling with Blender, and he was an invaluable resource for me as I stumbled my way through learning it. I started by following along tutorials from the Blender wiki pages. The basic landscape tutorials helped me get started with a larger grid and creating some terrain features. I also used these tutorials to figure out how to add and place textures on my model. The mario level is a fairly simple one, with around 1000 faces and nearly 600 vertices. I took it slow with the model and started with the outside border walls. I just created vertices where I needed them and clicked and drug them upwards to create the wall. I then made each object in the terrain individually by adding vertices and creating faces. Some of the objects were easier to create well above the map and then “lower” them down onto the terrain. One of the hardest parts of using Blender for me was navigating the camera and view around the map. I would create or place something on the map that looked great, but when I moved the camera I realized it was completely wrong. This was because I was trying to model in perspective view. Once I switched the camera to the orthographic view things became much easier.

**Experience with blender:** Connor Nagel

Most of my time with blender was spent watching online tutorials and working with a cheat sheet of shortcut keys while I created the models. Getting used to working in 3d was difficult, I essentially had to learn to sculpt virtually. For the most part I used a cube to start, then I applied a subdivision modifier to the shape to give it more vertices to work with. To get the shape of the object I extruded the faces of the subdivided cube. Once I had the basic shape I switched to the sculpt tool to make finer changes to the model. I ended up having to recreate my models about 5 or 6 times before I was happy with the result. After getting models roughly correct, I started using the texture paint tool to give the model custom coloring. After all the of the work I put in I consider myself proficient working with the modeling tools in blender.

**Problems:**

I had problems getting the materials for my models to be imported into the webgl scene. I was able to get the model vertices and edges to show, therefore I think it has something to do with the way I selected and added the materials to the object. I used the texture paint feature to paint the model directly instead of creating a texture separately. When I saw that the materials weren’t working I had to completely scrap the models I had to try to repaint them. I found a few ways to make this process easier including using UV maps to unwrap the faces of the model into a 2d coordinate plane, which makes coloring easier. However, I couldn’t completely figure out how to create materials in blender. The nodes used to create layered effects didn’t make complete sense to me and I couldn’t figure it out in time for the presentation.

**Mario 64 User Guide:**

To use the flat shading version: Open MarioFlat.html in Firefox web browser

To use the smooth shading version: Open MarioSmooth.html in Firefox web browser

Navigation: Using the mouse to aim the camera, you can use the left mouse button to move forward and the right mouse button to move backwards. Alternatively, you can use the A and D keys to strafe left and right, and the W and S keys to move forward and backwards.

**Final Screenshots:**





