

**Project 4: WebGL Graphics**

**Zachary Finnegan**

**CMSC 405 Computer Graphics**

**5/10/20**

**Explanation**

I spent quite a few hours creating the above graphic that I will describe in detail. I started with the spike ball and wanted to do something with that. My original plan was to have a little stick figure person like what is in diskplay but with more joints who would pull or be chased by the spike ball.

The process of creating the spike ball ended up disabusing me of that notion. The spike ball took forever, and I didn’t think I had enough time to create a person and animate all the movements that would be needed.

As I was falling asleep I though about Pac-Man and thus the spike ball turned into a defensive tool for the Pac-Man Ghosts when they are in their vulnerable form. The ghost has two modes, normal mode and beefy mode.

Normal mode looks close to the original Pac-Man Ghosts. The skin is blue, with the alpha value turned way down and a medium emissive value to give it an ethereal vibe. The eyes, mouth and moustache (this was an accident while making the eyes and I loved it so much I kept it.) are bright white and have an emissive value.

Beefy mode makes the ghost look like it is bursting out of the belt and gives the skin a red tinge with an emissive value with a red tinge as well. I did this to make it look like the beefy ghost is a little roided up or on a lot of pre-workout and kind of jacked aka redlining. The other features of beefy ghost are eyebrows that make it look mean and the spikes on the bottom are longer and angled backwards to give it a feel of speeding along. The last aspect of beefy ghost is that the animation is twice as fast during beefy ghost mode.

I wanted to make a wimpy ghost mode, but I ran out of time. Maybe next time.

The ghost pulls the spike ball behind it with a bronze colored chain that is anchored on the ball and the belt around the ghost. The chain moves in sync with the ghosts up and down movements as it moves forward. The chain anchored to the spike ball doesn’t move while every subsequent chain link moves slightly more until it reaches the ghosts belt. That link moves at the same rate as the ghost. This makes the chain look pretty lifelike and I am very happy with how it turned out!

There is also a floor that the ghost drags the spike ball across towards but off center of the viewer. Once the ghost and ball leave the scene, they return to the screen in the upper left-hand corner of the screen in a loop.

I used diskplay as my base and added in the spotlight features from diskworld-2 so I could have those features. I looked at most of the other sample code to get some ideas and figure out how some features work but I don’t believe I used any of that code directly.

There are four light options:

1. A light shining directly into the ghost’s face.
2. A light shining perpendicular to the scene.
3. An ambient ish light. It has a point of origin, but it looks fairly ambient.
4. There are two spotlights. One from above down and one from the side on the same level as the scene, facing the scene.

Light 1-3 can have their colors changed by the user with the 3 sliders.

I couldn’t figure out how to get textures into this scene. Everything I tried just broke everything.

So, I went to the ask the professor discussion board to see if anyone had asked you about that yet. Lo and Behold, someone had asked if we could use Three.js and you said yes. Which was unfortunate as I had already created my scene using regular WebGL and GLMatrix.

Because I couldn’t get textures to work in my Pac-Man WebGL basic scene I decided to add it too my project 3 Three.js based project and it worked! I am submitting both scenes as part of my project 4. I am hoping I won’t get too many points knocked off for not having all the features in my Pac-Man scene. I believe the Pac-Man scene meets all the other requirements.

In addition, I believe my project 3 scene with the textures added pretty much meets the requirements of project 4 as well. I wanted to provide an explanation as to why I had two scenes, one of which was the one submitted for project 3, in the submission folder.

**Test Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Expected Output | Actual Output | Pass? |
| Project 3 Snowman Scene with Textures | The Snowman should have textures added.   1. Bark for the arms and eyebrows. 2. Carrot for the nose. 3. Black silk that is shiny for the hat. 4. Red silk for the belt on the hat. Also, shiny. 5. Snow for the body. 6. Brass for the buckle. 7. Coal for the coals. |  |  |
| Pac-Man Scenes Below | | | |
| Animation & Head on Light & Ambient (ish) light  & Normal Ghost | The ghost, chain and spike ball should move towards and to the right of the viewer. The chain will move from a fixed point on the ball to the ghost progressively.  The ghost moves out of frame and then returns to the frame on the opposite side of the scene it disappeared on. Looping.  There should be a light shining in the ghost’s face and a general lightness throughout the scene.  Normal looking Pac-Man Ghost |  | Pass |
| Beefy Ghost with the same Spotlights only | The beefy ghost with angry eyebrows and red tinged skin should show up and during the movement the parts illuminated should changed based on if the object is in the scope of one of the spotlights. |  | Pass |
| Regular ghost & different camera angle & red side light | The scene should be viewed from the upper right side and there should be a red light that reflects off everything but doesn’t affect the ghost much except for the eyes and specular highlights. |  | Pass |
| Beefy ghost & normal angle & white head on light & purple side and ambient light. | The ghost should be fairly unaffected by the purple light. The spike ball, chain, spikes and floor should have a purple hue.  The ghost isn’t affected due to the emissive color, I think. |  |  |
| Reset Button | Everything should return to normal |  | Pass |

**Lessons**

I learned that base WebGL is a huge pain and offers a lot more flexibility than I really care about. Given the option I would not use it. If the shaders weren’t already created in diskworld-2/diskplay I would have been thoroughly up a creek. I understood the ones that were in diskplay enough to modify it to add in the spotlight functionality that had been removed but that is about it. I am sure it isn’t as complicated as I imagine but maybe not by a lot. I am really glad we didn’t have to build them from scratch.

I learned more about the possible interactions between CSS and Javascript. The new functionality helped reduce the amount of code I had to write. That was fantastic.

Although this project using WebGL base was a huge pain I believe I learned a lot more about the fundamental of computer graphics than I did with Three.js. There were a lot of things that I was able to accomplish without much knowledge of what was going on underneath. WebGL does not hold you hand as far as I can tell and makes you really think about what you want to accomplish and figure out how to accomplish it.

Specifically, I learned the most about how lighting and materials work using WebGL. Emission, spotlights, and other material properties. The metal should be shiny, the ghost should be emissive, and the belt should be duller than the metal but not so dull as to not appear worn out.

I also learned that there is a lot I don’t know how to do. Specifically, regarding the spikes on the ball. I figured out how to place them easily by creating a slightly smaller sphere with very few slices and stacks, placing the vertices in an array I could import to my program and then use those vertices as placements for the spikes. I thought that was pretty ingenious. Probably obvious to someone who knows what they are doing but I was excited about it.

The thing I couldn’t figure out is how to orient all the spikes so that the base of each spike was facing towards the sphere’s origin and perpendicular. There should be a way to orient the spikes in relation to something else, but I couldn’t figure it out. The documentation for WebGL and GLMatrix I found on the web was unhelpful and uncited as it was unhelpful, and I am holding a grudge.

So, instead of a nice couple lines of code I went and put in the necessary rotation values for every spike. That took forever. I believe there are 82 spikes on that ball. So painful. It looks awesome though.

I wish I could have found some better documentation. I only used the book and the existing code samples.

Fun project.

**References**

Eck, D. J. (2018). *Introduction to computer graphics.* Hobart and William Smith Colleges.

Greater, E., Lundin, M., Manini, S., Antigay, L. (n.d.) *TrackballControls.js*

King, L. (2020). *diskplay.html*

Three.js (n.d.) *Creating a scene.* Three.js.org. <https://threejs.org/docs/index.html#manual/en/introduction/Creating-a-scene>

Unknown Authors (n.d.) *diskworld-2.html*

Unknown Authors (n.d.) *basic-object-models-IFS.html*