**Mason Sanders CSCI 438 Low Level Project Documentation**

This project, created using Three.js, is meant to resemble an animation of the planet Saturn. It uses a sphere and two ring geometries. The sphere itself is completely stationary, while the rings rotate and wobble a little bit. The wobble effect is created using a simple sine function based on time.

Both the ring material and the planet material use the MeshToonMaterial, which creates a cel shaded effect with hard shadow borders instead of a smooth gradient. To complete the cel shading effect, I needed to find way to add an outline around both the ring and the planet.

I settled on using OutlineEffect.js provided in the three.js download in the examples/js/effects folder. It uses post processing to get the exact outline type of effect that I wanted to use. The major downside to it though, is that creates the outlines from the normal vectors of mesh, and the ring is 2D geometry which means that the ‘outline’ would be projected down from the ring instead of outward and inward. So there will be a slight outline effect on one side, but if you flipped it over, you would see nothing since the effect is transparent from the other side.

I slightly improved this issue by adding a second ring, exactly the same as the first, except flipped so its normal vectors would be pointing the opposite direction. This allowed for at least the slight outline when viewing both sides of the ring.

Finally, I included the OrbitControls.js from the examples/js/controls folder. This allows for using the left mouse button to orbit, the right mouse to pan, and the scroll wheel to zoom.

External files used:

three.js-master/build/three.js

three.js-master/examples/js/effects/OutlineEffect.js

three.js-master/examples/js/controls/OrbitControls.js

The file LowLevel.html is the html page that should be opened to run the application.

**Function Descriptions in LowLevel.js:**

init():

This function creates the scene, renderer, camera, outline effect, controls, geometries, materials, meshes, lights, and sets some initial positions.

update(now):

this function takes in ‘now’ as the current time and keeps track of delta time, creates motion in the rings, updates the orbit controls, renders through the outline effect, and requests a new animation frame.