**Three.js – Project 3**

**Lily Forry**

**CMSC 405 6380 Computer Graphics (2225)**

**Professor: Lauren King**

**Due: June 29, 2022**

**User’s Guide**

The goal of Project 3 was the use three.js to develop a unique 3D animated scene. The scene needed to be a minimum of 640 x 480 pixels, contain 6 unique shapes, use multiple lighting effects, and let the user change different aspects of the animation through toggles. For further information, see Project Requirements. Diskworld-1 was what I chose as the starting point for this project, since it already included most of what was required in the rubric and only needed slight tweaking.

**Getting Started**

The main file, Project3HTML.html, needs to be within the threejs folder, as it pulls from three.js.

Recommended browser: I utilized WaterFox and Edge when testing everything.

When Project3HTML.html is opened, a bunch of items should appear.  
A picture containing shape

Description automatically generated   
These items consist of a plates, torus, cylinders, cones, spheres, and rectangles.

The viewpoint can be changed on the animation by utilizing the arrow keys to change the X and Y rotation, and the pageup and pagedown keys to change the Z axis, with the home button resetting any rotations.

The Animate check-box can be checked in order to start the animation.  
A picture containing diagram

Description automatically generated

The Viewpoint Lighting check-box can be unchecked to remove viewpoint lighting.  
A picture containing graphical user interface

Description automatically generated

The Sun Lighting slider can be changed so different levels of sunlight will be provided to the scene.  
A picture containing diagram

Description automatically generated

Lastly, Show Axle, Show Car, or Show world radio buttons can be clicked on to change which aspect of the scene is being shown.   
Show Axle will only show the car’s axle  
Graphical user interface, application, icon

Description automatically generated  
  
Show car will only show the car  
Graphical user interface, application, Teams

Description automatically generated  
Show World will show the whole scene

**Project Requirements**

Include 6 different shapes

Torus, Cylinder, Rectangle, Cone, Sphere, Plate

Use multiple lighting effects

Sun Light, Viewpoint Light

Include radio buttons, slider bars, or other widgets to turn on or off certain components

Animate, Show Axle, Show Car, Show World, Sun Lighting, Viewpoint Lighting

Use Three.js

**Test Plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case | Input | Expected Output | Actual Output | Pass/Fail |
| 1 | Create Scene Canvas | A 640 x 640 canvas colored Grey should appear on the page below any text or widgets. | A 640 x 640 canvas colored Grey appears on the page below any text or widgets. | Pass |
| 2 | Create Widgets | Two Checkboxes labeled “Animate” and “Viewpoint Lighting” should display, as well as three radiobuttons labeled “Show Axle, Show Car, and Show World” should be displayed. Lastly, a “Sun Lighting” Slider should be displayed. | Two Checkboxes labeled “Animate” and “Viewpoint Lighting” are displayed, as well as three radiobuttons labeled “Show Axle, Show Car, and Show World”. Lastly, a “Sun Lighting” Slider is displayed. | Pass |
| 3 | Create Text | A header labeled “Forry’s Project 3 Implementation” should appear at the top of the screen. Corresponding Text for all of the widgets should also be displayed, as well as the controls for scene manipulation. | A header labeled “Forry’s Project 3 Implementation” appears at the top of the screen. Corresponding Text for all of the widgets is also displayed, as well as the controls for scene manipulation. | Pass |
| 4 | Create Platform | A disk should appear with a grass texture and road on it. | A disk appears with a grass texture and road on it. | Pass |
| 5 | Create Car | A red car object made out of red rectangles, blue torus wheels, and yellow axle cylinders should appear on top of the road. | A red car object made out of red rectangles, blue torus wheels, and yellow axle cylinders appears on top of the road. | Pass |
| 6 | Create Trees | Basic trees made from brown cylinders and green cones should appear on the platform. | Basic trees made from brown cylinders and green cones appear on the platform. | Pass |
| 7 | Create View Light | A white light should appear from the camera. | A white light appears from the camera. | Pass |
| 8 | Create Sun Light | An ambient yellow light should appear from the sky. | An ambient yellow light appears from the sky. | Pass |
| 9 | Check Slider Bar | As the slider bar is moved to the right, the sun lighting should get more intense. As the slider is moved to the left, the sun lighting should get less intense. | As the slider bar is moved to the right, the sun lighting gets more intense. As the slider is moved to the left, the sun lighting gets less intense. | Pass |
| 10 | Perform Viewpoint Transformations | When the user uses the arrow and pageup/down keys, the scene should rotation in relation to the key pressed. | When the user uses the arrow and pageup/down keys, the scene rotates in relation to the key pressed. | Pass |
| 11 | Select “animate” | From the diskworld scene, the car should rotate around the paved world and the largest central tree should shrink. | From the diskworld scene, the car rotates around the paved world and the largest central tree shrinks. | Pass |
| 12 | Select “Axle” | Just a view of the car’s Axle should be shown. | Just a view of the car’s Axle is shown. | Pass |
| 13 | Select “Car” | Just a view of the car should be shown. | Just a view of the car is shown. | Pass |
| 14 | Select “World” | The initial full scene with the disk, car, and threes should be displayed. | The initial full scene with the disk, car, and threes is displayed. | Pass |
| 15 | Select “Viewpoint” | If checked, viewpoint lighting should be shown. If not checked, it should not be shown. | When checked, viewpoint lighting is shown. When not checked, it is not shown. | Pass |
| 16 | Close the Page | The browser page should close. | The browser page closes. | Pass |

**Lessons Learned**

I really don’t like working with computer graphics and GUIs, in general. I find solution driven programming where you’re coming up with a solution to a problem way more fulfilling that messing with rudimentary graphics.  
JavaScript really isn’t that bad, although moving from Java one week to JavaScript the next week did catch me by surprise and throw me for a loop.  
Somehow, messing with threejs made less sense than with JOGL, which was really surprising. I really wanted to have a sun/moon cycle with adding different lighting effects on two different spheres, but I couldn’t figure out a way to do this. The way threejs handles lighting is extremely different to how JOGL handles it.  
Lastly, many of threejs’ lighting sources don’t cast shadows, I’d like to try something actually messing with casting shadows in the future.