Coursework Report Task 1 (Web GL)

UP902282

January 2021

Contents

1 Design/Implementation Decisions

The aim of this task was to show an application that could have a model of earth rotating along it's Y axis, textured with a map of the earth, while also having a model of a satellite orbiting said earth (1). There we're some decisions that had to be taken while developing the application, one of them was between choosing to implement 2 different shaders (one for the earth and another one for the satellite) so that we could have both coloring capabilities and texturing capabilities or using some texturing techniques to apply a texture represented by only one color and applying it to the satellite.

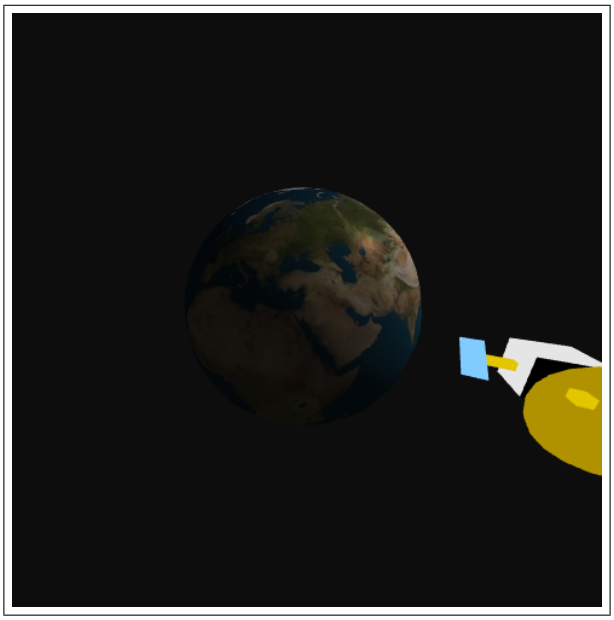

Figure 1: Image from the finished product

Since the choice was to use multiple shaders, there was a recurring problem, which for this application wasn't noticeable, but on larger applications might have bigger impact. For this to work we needed to change between shaders during run-time so that we could draw both the satellite and the earth at the same time using both methods, but changing shaders at run-time is very inefficient and might cause performance to degrade, so in this case it meant that the application for every frame that it processed it had to change between shaders around 6 times.

Another small change made to the design was to implement a dark gray background, instead of it being plain black, this way we could have more contrast between the satellite's dark face and the overall earth. Also the color of the "antenna" on the satellite was changed to a darker gold color, from the rods connecting it to the satellite body.

2 Difficulties/Problems

During the development of this task a lot of problems came along, of all these problems most of them we're fixed by simply adjusting the code, but some of them remained and couldn't be fixed. One of these problems was the implementation of *normals* in the satellite. Normals are what makes the application know which way the models are facing according to the light, which allows the program to calculate the shadow accordingly. As it currently stands the earth has normals with the light correctly applied (3) but the satellite does not (2), which makes it lack depth, comparing to the earth.

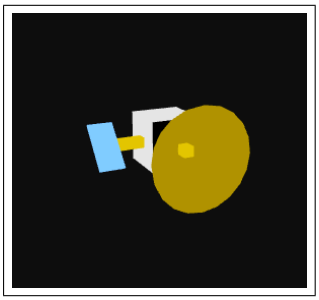

Figure 2: Satellite Model (no normals)

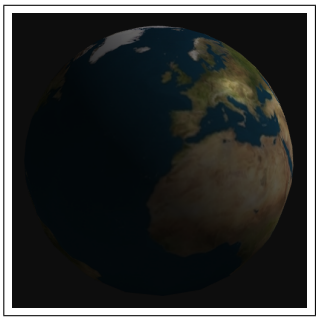

Figure 3: Earth Model (with normals and lighting)

Another major problems that arrived was regarding the camera control system, the way it was supposed to be implemented was to allow the user to control the orbit speed, as well as the orbit radius of the satellite, this was not hard to do, but there are some problems related to the speed of the orbit, which sometimes goes backwards or even stops and stutters for a bit if the speed reaches the minimum threshold, this is a bug that might be related to the FPS of the application.

Figure 4: Default Orbit size

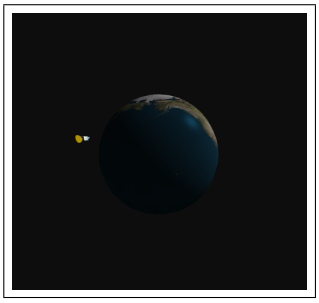

Figure 5: Expanded orbit size

3 How to run the application / Files

To start the application the file *"index.html"* should be opened in *Firefox*, this is because Firefox has better compatibility with loading textures, which might not show in other browsers.

index.html - Contains the main program and main events such as camera controls, draw the models to the screen, manage FPS and also contains the definition and the code of the shaders.

common.js, textures.js - This file contains some common functions and variables that might be used multiple times and are not directly associated with any model in specific.

modelAntenna.js, modelEarth.js, modelSatellite.js, modelSolarPanels.js - These are the files that are responsible for generating the models and assigning all the textures, colors, normals and vertices of the models.

glMatrix.js, webgl-debug.js, wbgl-utils.js - All these files are webGL libraries that contain useful functions to aid in the development of the application.

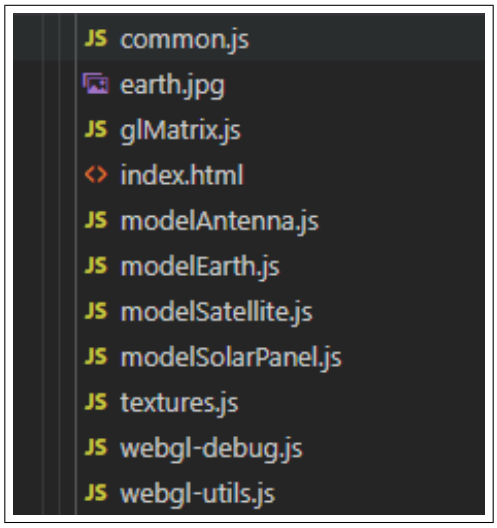

Figure 6: Files hierarchy

4 Conclusion

Overall the aim of the project was completed, although the normals of the satellite are missing and the camera control is not perfect, the rest of the requirements we're completed successfully, this includes the models, the lighting scene, the animations and the whole scenario of the task.