**Part 1 - Report**

**Sphere (Earth)**

I started off by creating the sphere. I had to create a buffer for the sphere and inside there i put in the vertex coordinates and the vertex indices. At first I didn’t know how to create a sphere because, in the tutorial i mostly created cube, triangles and cuboids. But, I found a formula for creating a sphere from lecture 5. Even though i had the formula i still manage to run into errors. I had an error where the item\_size and num\_size was wrong because it was not big enough. I also had an error because I used gl.Triangle\_Strip instead of gl.Triangles so my sphere looked nothing like a sphere. After that, I started adding texture to the sphere. I implemented the formula that I had used for the vertex coordinate into the texture coordinate. I used the image “Earth.jpg” provided in the moodle for my sphere.

**Satellite**

I implemented a buffers for satellite (setUpCubeBuffers). I added the vertex coordinates and vertex indices. This time I didn't have to use any formula to create a cube, I just had to input vectors for the top, bottom, left, right, front and back cube. I used gl.array\_buffer for this. I created a cube of size 2x2x2 and this will be the main body of the satellite. I created cuboids on both sides with the size of 0.2x0.2x0.5 this will be my rod and then translated it by 4.0 on the x-axis and -4.0 for the opposite rod. Next, I created the cuboids for my panels with the size of 1x0x2 and then translated it by 8.0 on the x-axis and -8.0 axis for the opposite panel.

I created a small cuboid for the antenna with the size of 0.2x 0.2x0.4 and then translated it by 2.0 in z-axis.

**Animation**

I created a loop for the satellite to rotate around the earth. It updates the position of the box

pwgl.animationStartTime

pwgl.x = Math.cos(pwgl.angle) \* pwgl.circleRadius;

pwgl.z = Math.sin(pwgl.angle) \* pwgl.circleRadius;

With these as their axis it will allow the satellite to rotate as long as the y-axis is above 1.

In order to make the earth rotate slowly in its y-axis. I had to create a variable Radians = 50 \* Math.PI / 180

**User Inputs (Interaction)**

With the help from Tutorial 8 and Tutorial 9 i implemented the code from it to make the viewport rotate in y-axis and z-axis. It increases the radius it is rotating by pressing down the right arrow and decrease the radius (pwgl.circleRadius) by pressing down the left arrow. It increases the speed of the satellite rotation but you have to keep pressing it to speed up. This is done by increasing the pwgl.animationStartTime everytime the down arrow key is pressed. Alt-mouse drag and mouse wheel is also working correctly all i did was use the function that I learned from the tutorials.

**Lighting**

I have added lighting to the models, it will reflect lights whenever the light hits the object. With the help of tutorial 9 i implemented the vectors L, N, V and R which will be used to Phong reflective model. A single point light will emit and illuminate the models. I created uniform variables such as ambient, diffuse, specular for the colours of the light and light position to position the light. I inputted the calculation for diffuse and specular and the overall reflection calculation that uses the weighted sum of the diffuse, ambient and specular. After doing the calculation it stores it in vLightWeighting variables.

**Problems**

There are problems that I could not find a solution for. There seems to be a white lines around the earth and i could not remove it. I tried removing it by configuring around the texture coordinates formula that I used but i was unable to fix it.

I could not able to make the satellite's black side face the earth while it is rotating around.

I was not able to decrease the speed of the satellite’s rotation and I could only manage to pause the satellite by pressing the down arrow key.

The shift drag function works but because, the earth keeps rotating it changes the direction of x and z therefore, it moves in the x direction during the start of the animation but gradually changes

**Testing against the specification**

|  |  |
| --- | --- |
| The scene consists of the planet earth in the middle with a satellite orbiting around it along a circular orbit within the horizontal plane. |  |
| The scene is illuminated from top-right by a *directional* light that is at a 60-degree angle with the horizontal plane if viewed in front view. |  |
| The earth model is a sphere of radius 10 and mapped with an earth image |  |
| The earth rotates slowly around its own vertical axis |  |
| The satellite consists of a main body of a cube of size of 2x2x2 and two “solar panels” that are attached to the two opposite sides of the main body through two connection “rods”. The rods are cuboids of size 0.2x0.2x0.5 and *golden* in colour. The solar panels are *blueish* thin rectangular objects of 1x2 in size. The panels always face upwards. One side of the cube representing the main body of the satellite has a *black* colour |  |
| You should be able to control the radius of the circular orbit (with the left and right arrow keys) |  |
| You should be able to control the speed (up and down arrow keys) of the satellite at runtime |  |
| You should have full viewport/scene-navigation control: translations along x- (shift plus mouse drag) |  |
| You should have full viewport/scene-navigation control: translations along y- (alt plus mouse drag) |  |
| You should have full viewport/scene-navigation control: translations along z-direction (mouse wheel) |  |
| You should have full viewport/scene-navigation control: rotations around x- and y-axes (mouse drags) |  |