**Evaluative Report**

**Object Models**

The specification requested that two object models were created, one in the shape of a sphere to represent the earth, the other in the shape of a cube to represent a satellite. Both objects have been generated successfully, the spherical object has a radius of 5 units and the cube object has a height, width and depth of 2 units.

**Textures**

The requirement brief for the project stated that the spherical object should be textured with the image provided, while the cube should have two different colours for its sides. I was able to successfully map the textured image of the earth onto the sphere, giving the impression that the image itself is spherical.

Initially I wanted to use colours for the sides of the cube, however I found this to be difficult to implement in webgl. As an alternative I created a 2px x 1px image where 1 pixel is dark grey and the other is a golden yellow to use as a texture. Unfortunately the image editor I was using blurred the two colours together but I was able to map the individual colours so that one side is the dark grey colour and the others are the golden yellow colour.

**Lighting**

The minimum requirement of this program in terms of lighting was to implement a directional light from above the scene. This has been achieved successfully using per-fragment lighting, commonly referred to as the phong lighting model.

In addition to the specification I have also implemented a small amount of ambient light to ensure that the underside of the sphere is still visible even though it is not directly lit. I have also attempted to implement specular lighting, whereby the point of the object model the light reaches first is illuminated however it appears to me that the specular point is appearing inside the sphere rather than on the surface.

**Animation**

All of the animation requirements outlined in the initial brief have been included in my program. The earth spins slowly on its vertical axis, while the satellite orbits the earth around the horizontal plane, the starting radius for the orbit is 7.5 units which allows it to comfortably pass the outside of the earth without colliding. For the final part of the animation, the satellite rotates during its orbit so that the dark grey side faces the earth at all times.

**Interactivity**

The specification also requested that the program should be interactive, whereby the user should be able to use the keyboard to control the speed and radius of the orbit of the satellite.

I have implemented this interactivity so that the user is able to perform the following operations to interact with the scene. Pressing the “A” key increases the speed of the orbit, while pressing the “Z” key decreases the speed by a corresponding interval. Pressing “PgUp” increases the radius of orbit, while “PgDn” decreases the speed of the orbit. In addition to the requested functionality I have also enforced a lower limit on the radius of the orbit. This is to prevent a collision with the sphere.

**Scene Control**

The specification stated that the user should have full control of the scene so that they could use the mouse and/or keyboard to rotate, zoom and move the viewing point. Through clicking and dragging the mouse the user is able to rotate the entire scene so that it can be viewed from a different perspective. Using the arrow keys the user is also able to zoom in and out, as well as move horizontally along the x-axis. In addition to the specification I have also implemented limits on the amount the user can zoom in and out of the screen, this prevents them from being able to move inside the cube or sphere which would give them a distorted view of the scene.

I feel the scene control could be enhanced further by allowing the user to additionally rotate the camera as well as the entire scene. This would provide the user with a greater amount of control over their viewing position and could allow them to perhaps follow the cubes orbit around the sphere.

**User Interface**

There was no requirement outlined in the specification for a user interface, however I felt it would be useful to assign an area of the HTML page for detailing the current state of each of the interactive variables, as well as to explain how the user is able to interact with the scene including keyboard controls. I feel this area could be further enhanced by adding feedback on the rotation of the scene,

**Browser Compatibility**

The program I have produced works as expected in the latest versions of both Google Chrome (v31) and Mozilla Firefox (v25). However in order for it to work in Chrome, the code needs to be accessed from a web server, for the purposes of debugging I was able to run it successfully from a locally hosted Apache2 server on my own machine.

**Conclusion**

Overall I believe that my program has met all of the requirements defined in the specification. In some areas I have also introduced my own enhancements to improve the realism of the animation and to provide the user with feedback on the way they are interacting with each component of the scene.

If I were to have more time available for this project, I would look at enhancing the lighting and the way the user interacts with the program. I feel some of my keyboard controls are not completely natural to a user and researching the keyboard controls for computer games would enable me to fine tune this.