Computer Graphics Final Project Report

(COMP.5460)

Aaron Raymond Buthelo

01746342

Date: 04/29/2018

Submitted

Dr. Haim Levkowitz

(Associate Professor at UML)

Submitted

Neha Mishra

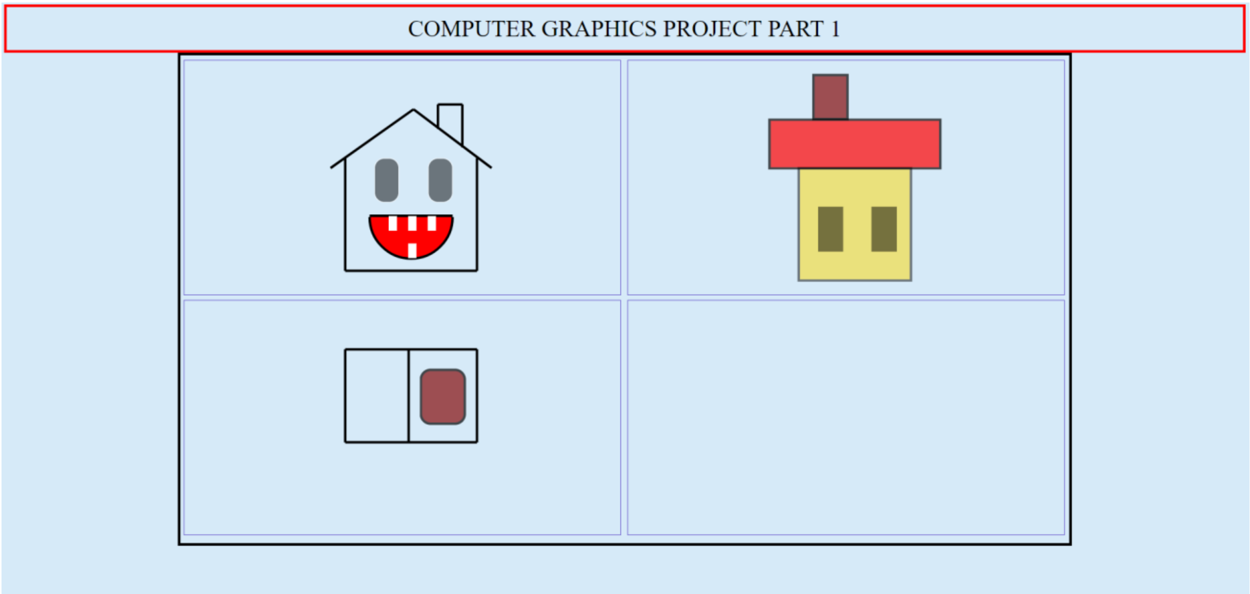
(Teaching Assistant at UML)

Table of contents:

Project Report Submission Week 1:

For the first week, I tried to implement three 2-D elevation views (front, side, top) as explained by the professor (same image which was explained by the professor). The object that I used was a ‘House’. I tried to demonstrate the front view, the side view and the top view of the same house.

I made use of Scalable Vector Graphics (SVG) properties to show the different 2-D views of the object. I have drawn the lines of the object by giving co-ordinate points. I have made use of shapes like arc and rectangle too to show the object for this week’s work. I have used the SVG elements like the path and used commands like moveto M, lineto L and elliptical arc A to draw curves and lines and polygons.



COMPUTER GRAPHICS PROJECT PART 1

As I was relatively new to SVGs, to complete this week’s work I took help from various websites like:

https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute

https://www.w3schools.com/graphics/svg_path.asp

https://www.sitepoint.com/understanding-svg-fill-rule-property/

Project Report Submission Week 2:

For this week's work I used WebGl Renderer, Three.js to render the 3 dimensions to objects. So basically there 3 objects that I am made:

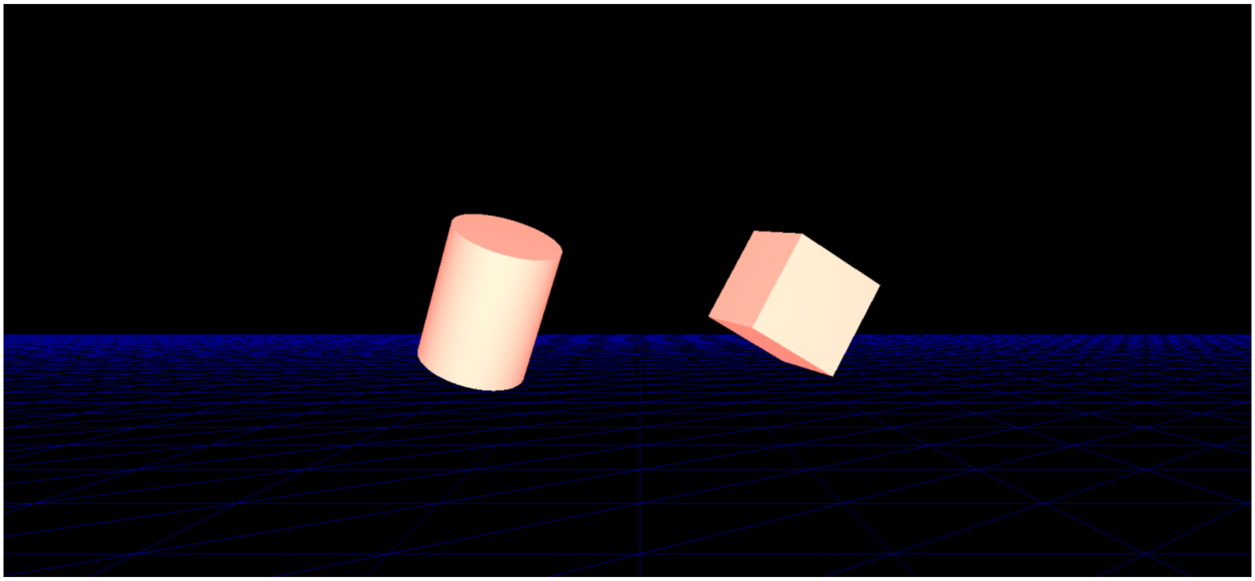
(1) A 3-Dimensional Cube which autorotates around all 3-axis.
(2) A 3-Dimensional Cylinder.
(3) A geometric Plane.

While coding, I created a three.js scene, made use of three.js perspective camera, three.js materials and geometric shapes like cube and cylinder to create objects and animate them.

I used the ambient and point lighting properties to shade the object based on the light sources and distances from the object.

I used the following while doing this week's work:

(1)THREE.WebGLRenderer: Used for rendering 2D and 3D graphics without use of plus-ins.
(2)THREE.PerspectiveCamera: Used to create camera and position view angle.
(3)THREE.AmbientLight: This light globally illuminates all objects in the scene equally.
(4)THREE.PointLight: This light is used to cast shadows.
(5)THREE.MeshLambertMaterial: It is for non-shiny surfaces without specular highlight.
(6)THREE.BoxGeometry: Used to draw 3-D cubes.
(7)THREE.PlaneGeometry: Used to draw a plane.
(8)THREE.CylinderGeometry: Used to draw a 3-D Cylinder.



References: As I was relatively new to THREE.js to complete this week's work, I took help from various websites like:

https://threejs.org

https://www.w3schools.com

http://math.hws.edu/graphicsbook/c5/s1.html

http://jsfiddle.net/83U4t/21/

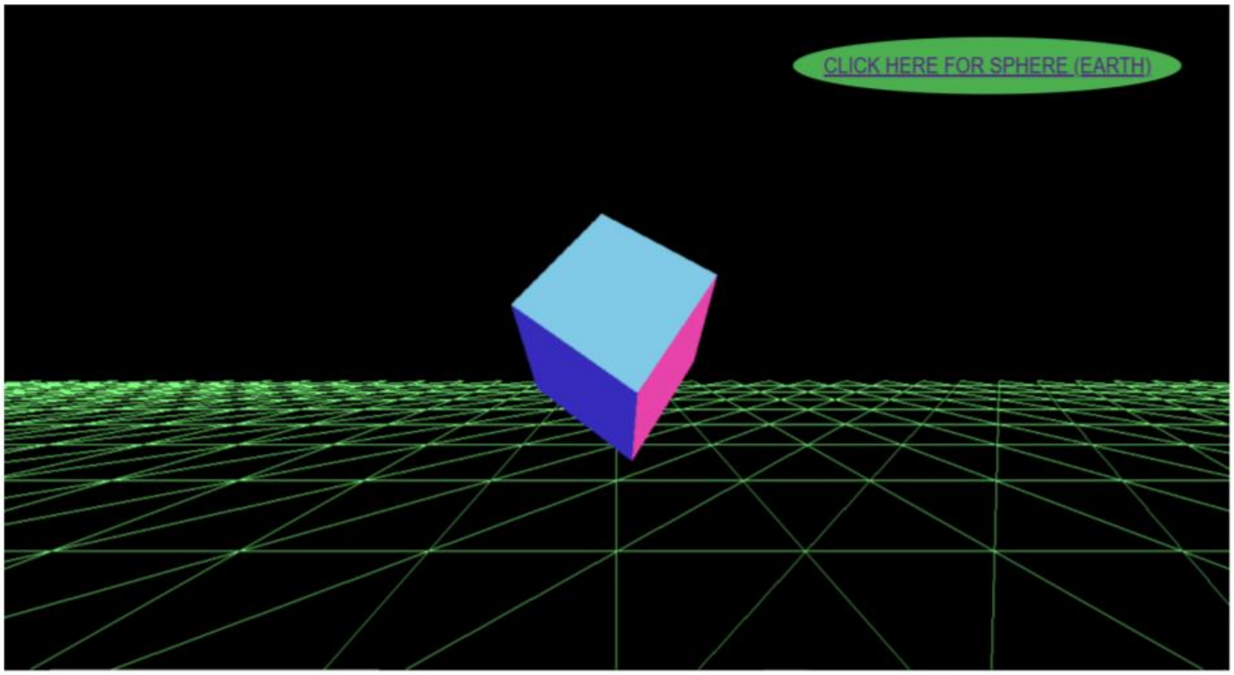
Project Report Submission Week 3:

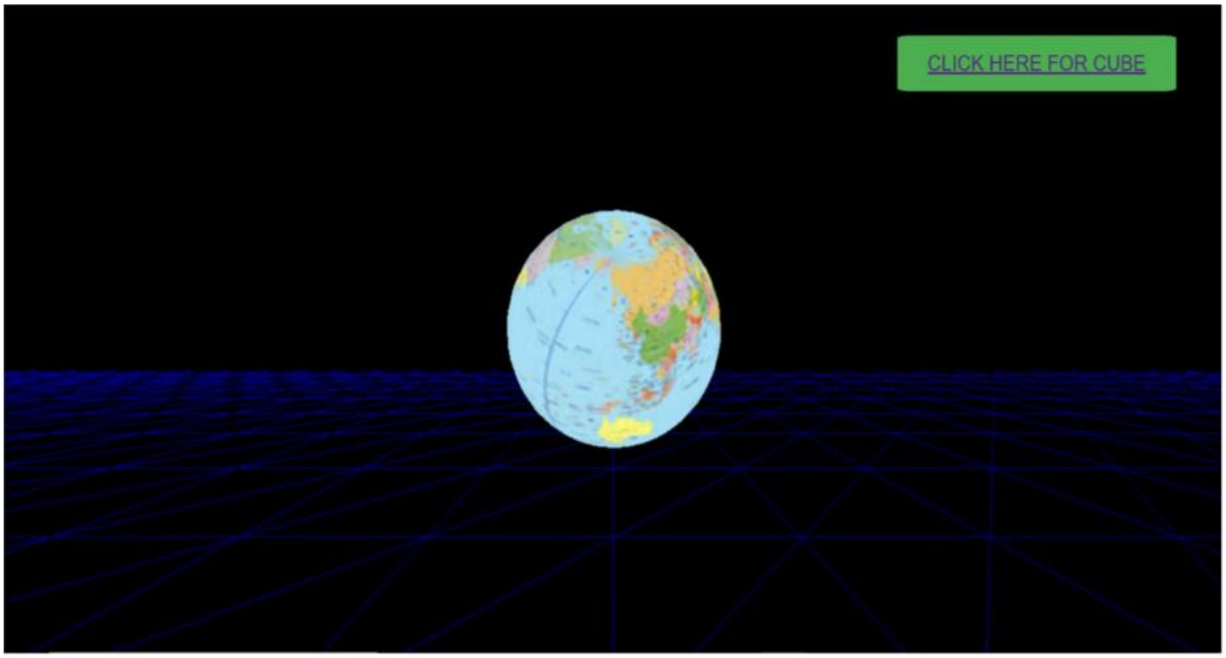
For this week I enabled multiple camera views to view the object. Now we were able to translate the object from one place to another with the use of mouse click or using arrow keys. I also implemented a texture for the object.

This week there are 2 different objects that I have made:
(1) A Cube.
(2) A Sphere (Earth).

I added translation to the object. The objects made could be zoomed in and out with touch or mouse. It is now possible to translate the object and to view the object from multiple views. For the sphere I implemented texture mapping and tried to make it look like the earth (globe).

I have used the following while doing this week's work:
(1)THREE.WebGLRenderer: Used for rendering 2D and 3D graphics without use of plus-ins.
(2)THREE.PerspectiveCamera: Used to create camera and position view angle.
(3)THREE.AmbientLight: This light globally illuminates all objects in the scene equally.
(4)THREE.PointLight: This light is used to cast shadows.
(5)THREE.MeshLambertMaterial: It is for non-shiny surfaces without specular highlight.
(6)THREE.BoxGeometry: Used to draw 3-D cubes.
(7)THREE.PlaneGeometry: Used to draw a plane.
(8)THREE.SphereGeometry: Used to draw a sphere (Earth).
(9)THREE.MeshNormalMaterial: Used to map normal vectors to RGB colors.
(10)THREE.BasicMaterial: Used for drawing geometries in a simple shaded way.
(11)THREE.TextureLoader: Used to load a texture on a object.





References: As I am relatively new to THREE.js to complete this week's work, I took help from various websites like:

https://threejs.org

https://www.w3schools.com

http://learningthreejs.com

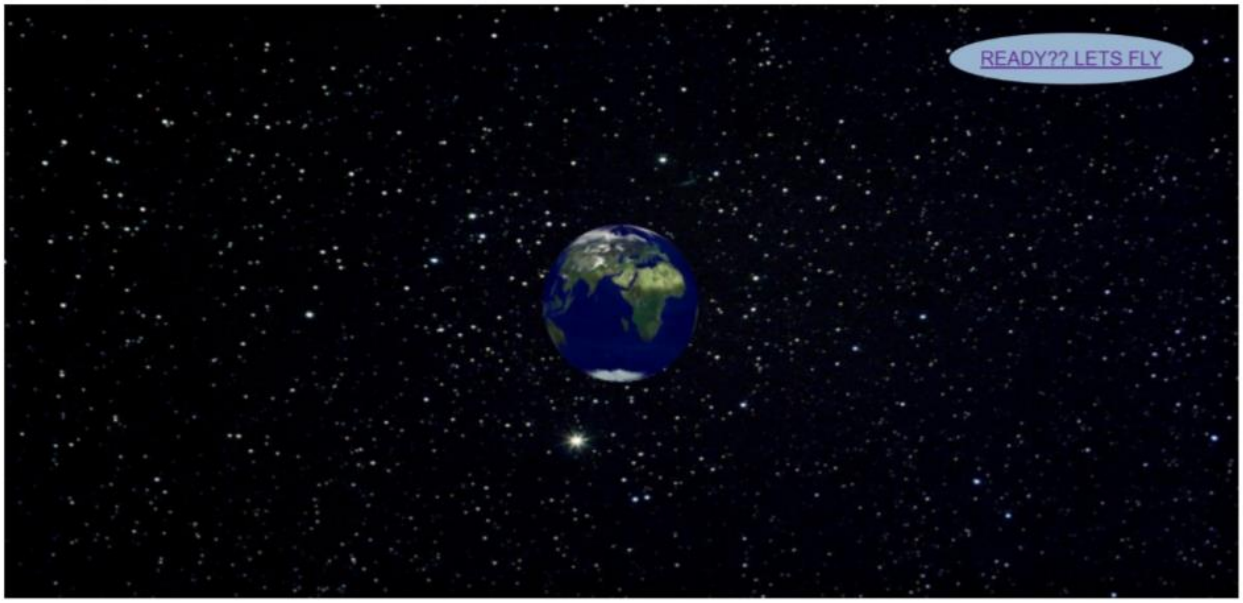
Project Report Submission Week 4:

This week there are 2 different webpages that I have made:

(1) A Rotating Earth.
(2) A Rocket revolving around the Earth.

I have used perspective cameras, Hemisphere Light, Point Light, Ambient Light, to enhance the auto-rotating and revolving 3-D objects created. I have added translation to the object. The objects made can be zoomed in and out with touch or mouse. I have implemented multiple views too. I have implemented texture and bump mapping for the Earth. I have implement hemisphere lighting to show a part of earth (the opposite of part facing sun or source of light) and rocket in darkness to make it feel real.

Some of the new objects I used for this week's work (which haven't been used before):
(1) TweenMax: It is a Three.js animating property.
(2) THREE.IcosahedronGeometry: A class for generating icosahedron geometry.
(3) THREE.FlatShading: A property to apply shading for objects.
(4) THREE.HemisphereLight: A light positioned directly above the scene, with color fading.



References: As I am relatively new to THREE.js to complete this week's work, I took help from various websites like:

https://threejs.org

https://greensock.com

https://codepen.io/

https://www.w3schools.com

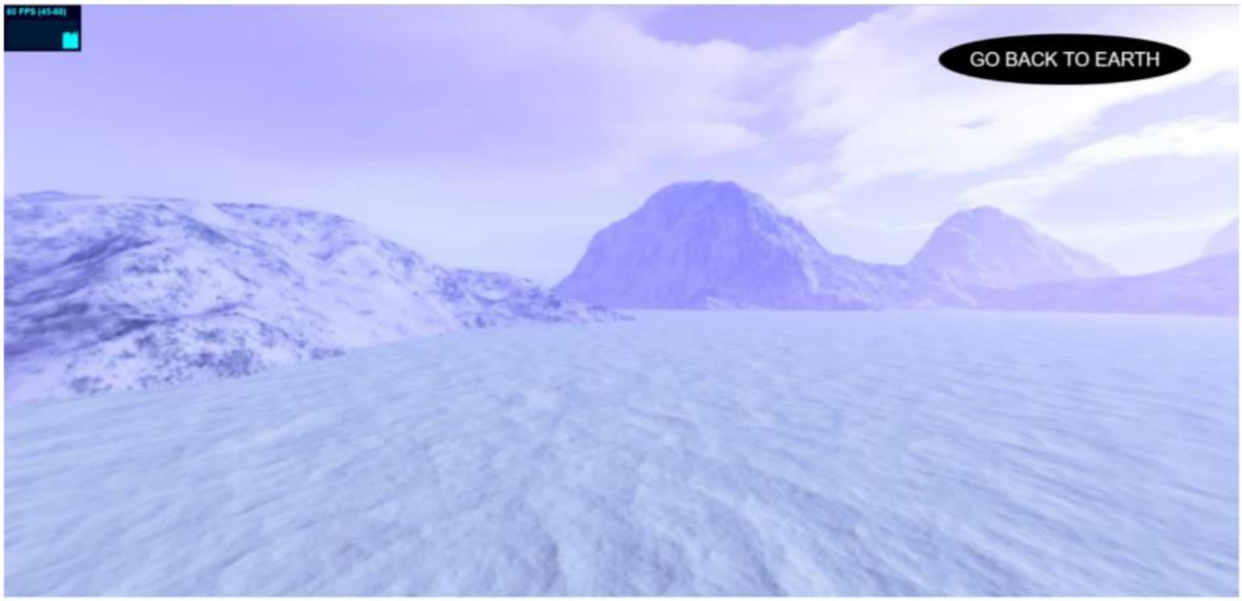
http://learningthreejs.com

The Rocket and the smoke function and the path of the rocket was directly taken from https://codepen.io/ to give a more realistic animation feel.

Project Report Submission Week 5:

For this week I have mapped an environment using texture and image loading three.js properties. I have also made use of the FPS (frames per second/framerate) property that comes along with three.js which gives the frames per second used.

Some of the new objects I used for this week's work (which haven't been used before):
(1) THREE.TextureLoader: Create a texture to apply to a surface or as a reflection or refraction map
(2) THREE.PointLight: A light that gets emitted from a single point in all directions.
(3) THREE.DirectionalLight: A light that gets emitted in a specific direction.
(4) THREE.SpotLight: This light gets emitted from a single point in one direction, along a cone that increases in size the further from the light it gets.



References: As I am relatively new to THREE.js to complete this week's work, I took help from various websites like:

https://threejs.org

https://greensock.com

https://codepen.io/

https://www.w3schools.com

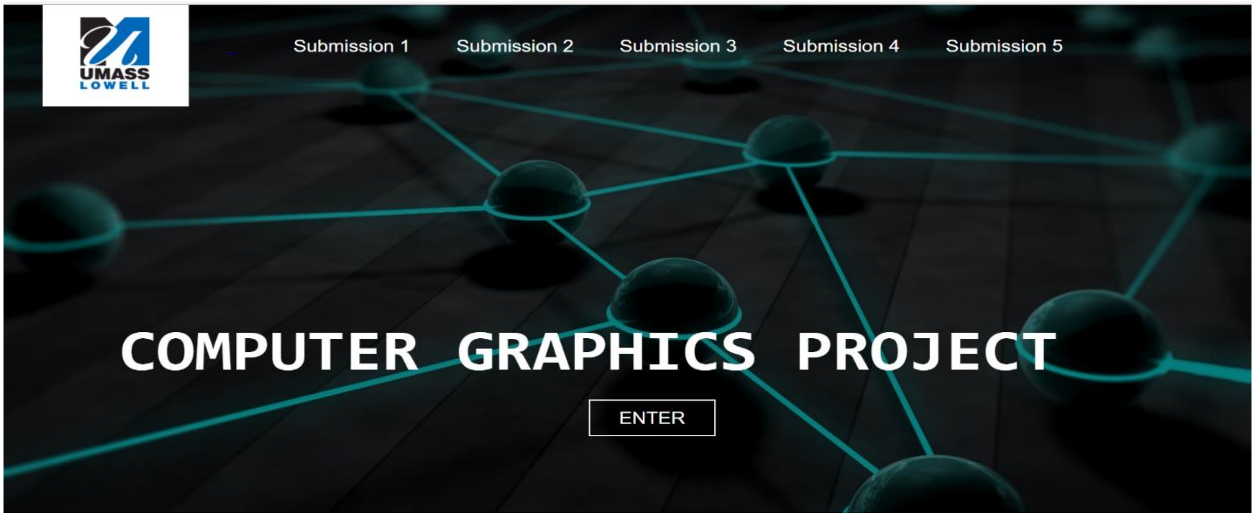
http://learningthreejs.com

And a ton of YouTube videos the helped me understand how to implement THREE.js

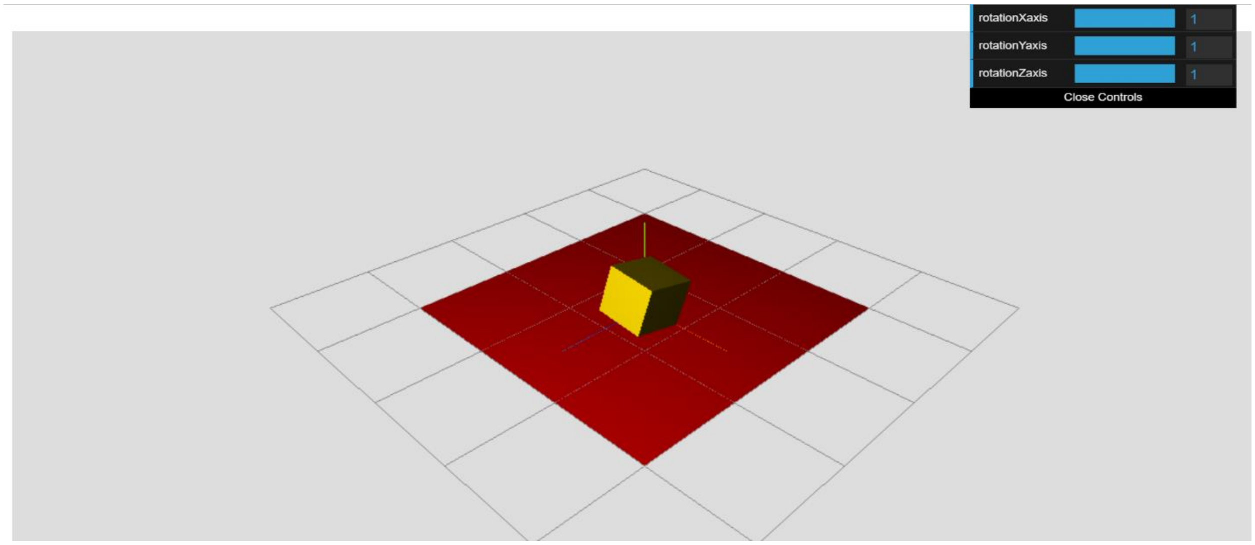
Final Report:

In this 5 weeks of doing the Computer Graphics project, I learnt how to create objects (Geometric) and play with different textures and map them on the object using THREE.js.
I also learnt how to use different lighting properties to create shadows and various other effects. I also learnt how to execute Texture mapping and bump mapping on an object and how to map an environment using images. This was my first step towards creating an animation or a game with the help of THREE.js and Webgl. I look forward to using what I have learned to make a more realistic project in future.
For the final week I have concatenated all the submissions and I done till date.

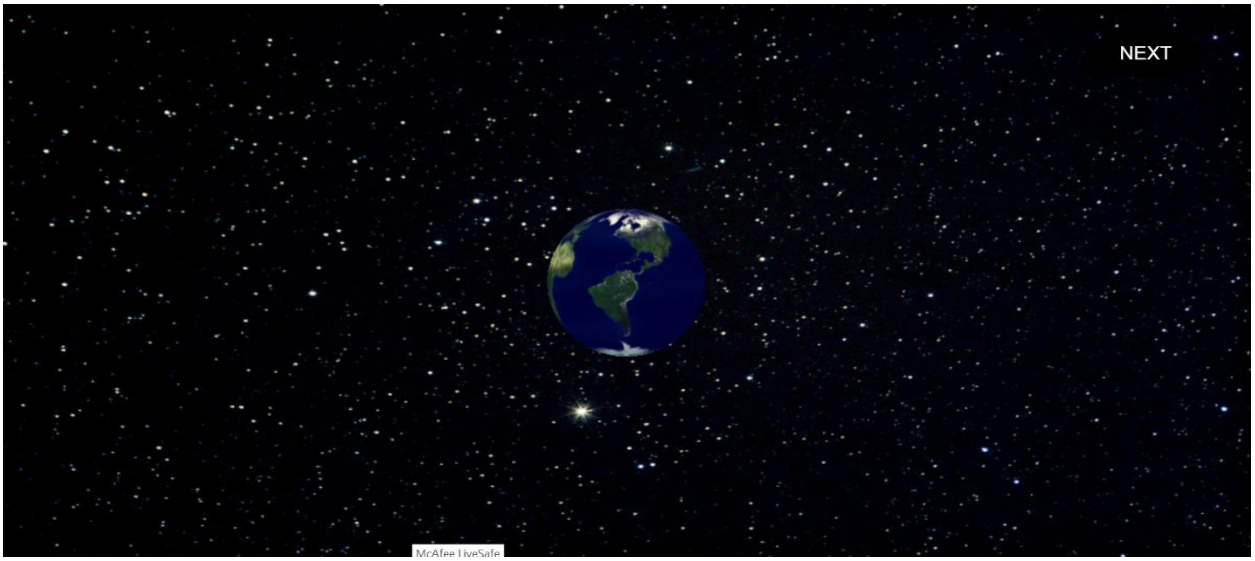
For this week I have included the following webpages:



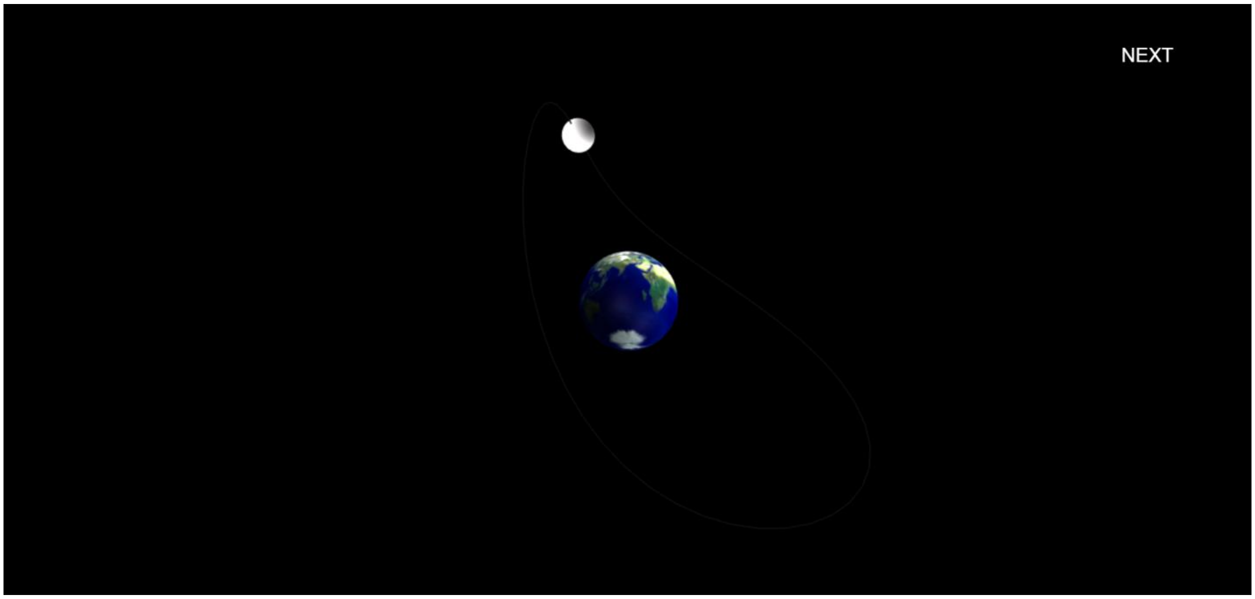
This page has links to all the submissions made till date. You can go to Each week’s submissions from the home page, also click on enter to view the final Project.


rotationXaxis 1
rotationYaxis 1
rotationZaxis 1

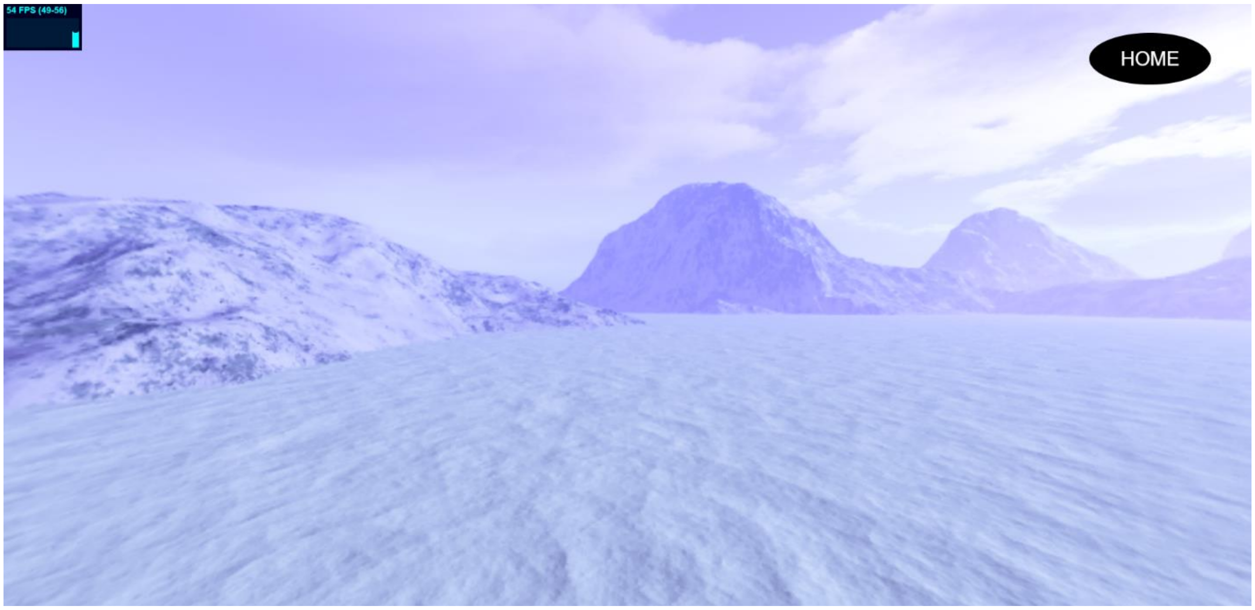
Here I have made use of the dat.gui library to implement the close control box to control the rotation manually.



Here I have made use of sphere geometry and Image Loader to map the sphere with an image to make it look like Earth, I have used bump mapping too.



Here I have used sphere geometry to create the Earth and the moon, I have given the path of the moon by using THREE.Curves.CinquefoilKnot. This part is hugely inspired by https://codepen.io/



Finally, here I have made use to 6 different images to create a skybox and to map surrounding to give a 360-degree rotation camera view.

I have tried to implement every feature which was mentioned by the professor in the mail, which I look forward to explaining during my in-class presentation.

For completion of this projects, I have made use of tons of online resources and referred to many YouTube videos, this project is not entirely done by me, but done with the help of online resources, understanding them and implementing them to accomplish my desired output.

I took a lot of efforts from my side to learn something new every time I tried to implement and build up on previous weeks work, but I am glad I was able to learn lot of things while completing this project which I will be using in the future!