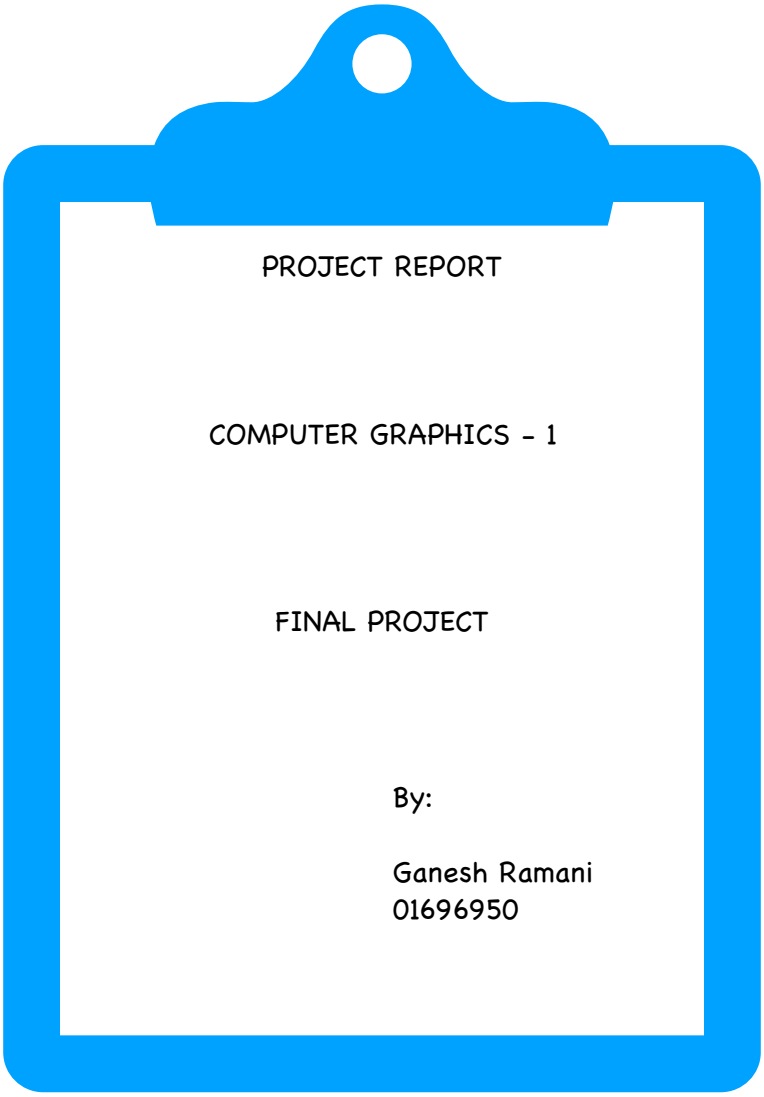


PROJECT REPORT

COMPUTER GRAPHICS - 1

FINAL PROJECT

By:

Ganesh Ramani
01696950

TABLE OF CONTENTS

INTRODUCTION

The Final Project Contains all the Features Listed Below for each Module and the specification for the Modules Highlighting a Feature in Computer Graphics 3D Generation of Models and Completing the Course requirements as mentioned in the Syllabus.In this report I have also Specified Each module and my observations while trying to create a 3D model for this Project.

The Entire Project is Split in 6 Parts each repressing a Weeks worth of submission by me. I have Provided with Screenshots of each module and its output.

I have used Three Js as the main source to make my final project. Three JS is a open source library with a extension to Web GL.

I have used external Java scripts Like Orbit controls , Transform Controls , Water script , DAT Gui and Trackball controls along with Three Js to Complete the project.

I have also Used HTML and CSS to Design the Webpage for the Project.

To start Open Main_page.html

The Link to My Project Archive is As below.

College Website: http://www.cs.uml.edu/~gramani/427546s2018/Final_project/SUBMISSION_FInal/Final_project/

GitHub: https://github.com/GaneshRamani11/CG1FINAL.git

MODULES

Week 1 :

Module Number	Model	Features
1	Cube	• Movement of Object using trackball Controls. • Viewing Multiple Faces of Cube. • Generation of Sphere Using Cube Geometry. • Static Cube Dimensions.
2	Sphere	• Generation of Sphere Using Sphere Geometry. • Wireframe Material Enabled. • Static Sphere Dimensions.
3	Cone	• Generation of Cone Using Cone Geometry. • Wireframe Material Enabled. • Static Cone Dimensions.
4	Cylinder	• Generation of Cylinder Using Cylinder Geometry. • Wireframe Material Enabled. • Static Cylinder Dimensions. • Enabling Rotation of Cylinder Along Y axis.
5	Tree	• Generation of Cylinder and Cone. • Basic Material Used. • Static Cylinder & Cone Dimensions. • Enabling Rotation of Tree Along X axis.
6	Multiple Primitives	• Generation of Cylinder ,Cone,Sphere ,Cube Dynamically. • Basic Material Used. • Enabling Rotation of Model Generated Along X axis. • User Feedback on the Number of Object to be generated. • Positioning of Multiple Objects using Loops.

Highlights :

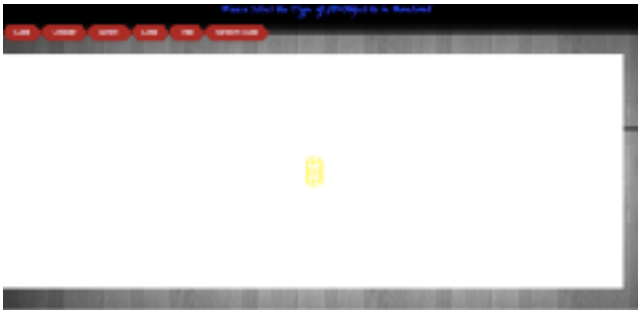
- Rotation of Objects in 3D.
- Positioning of Object in 3D.
- Modelling in 3D.
- Materials and Color Addition.
- Different Types of Material and their Properties.
- Rendering of Primitive 3D Objects

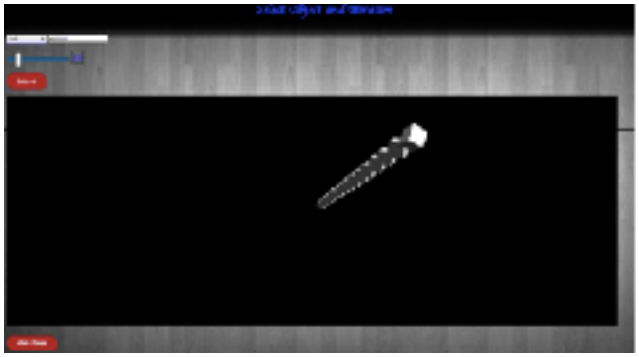
A. Cube.
B. Sphere.
C. Cylinder.
D. Cone.

- Rendering a Primitive Tree with Cylinder as the Stem and Cone as the Leaves.
- Multiples object Renderer:
  A. Cube
  B. Sphere
  C. Cone
  D. Cylinder
  E. User Input on the Number of Iterations

Details of Implementation :

I. This submission I have tried too implement the basic rendering of primitives using Three JS.
II. This Implementation served as a Introduction to Three JS and its API's.
III. I have used to basic API's such as Sphere , Cube, Cone and cylinder to construct the Assignment.
IV. I also tried to implement a basic structure of a tree which can be modeled using primitives and apply rotation and elevations in the submission.
V. I tried to give user Feedback and Input in the submission my rendering and position multiple renders as the per user Input.





Week - 2:

Module Number	Model	Features
1	Cube with User Specified Dimension	• Movement of Object using Orbit Controls. • Viewing Multiple Faces of Cube. • Generation of Cube Using Cube Geometry. • Dynamic Cube Dimensions. • HTML DOM Injection of Cube Width , Height and Length to be inputted any User. • Vanishing points on Zooming out from the cube Rendered.
2	Solar System	• Generation of a 9 Spheres representing a Planet. • Particle Geometry Function used to represent stars. • Mesh Basic Material with Color Coding for each planet is used.. • Planets position is placed by determination and not dynamically for Better View. • Used Line Dashed Material for Orbits in a Circle Geometry.

Module Number	Model	Features
3	Solar System with Lights	• Generation of a 9 Spheres representing a Planet. • Particle Geometry Function used to represent stars. • Mesh Phong Material with Color Coding for each planet is used. • Used Line Dashed Material for Orbits in a Circle Geometry. • Planets position is placed by determination and not dynamically for Better View. • Ambient Lighting Used to Illuminate the Materials. • Rotation of the planets and Orbits are Enabled.
4	Basic Animation	• Generation of Cylinder,Cube,2 Spheres. • The Animation makes the 2 Spheres Move around the Cube and Cylinder. • Static Dimensions of objects. • Enabling Rotation of Cylinder Along Y axis. • The Position is Automatically incremented using Loops to create a Animation.

HighLights:

- A 3D Cube Rendered with User Co-ordinates with Dynamic HTML DOM Manipulation using Javascript and Three JS to Accomplish the Task.
- A primitive Rendering of a Solar System with Spheres Placed in a orbit and with the Orbits Drawn Around the sphere representing a planet in our Solar System.
- A 3D rendering of a solar system using MeshLambert Material in three Js and with the orbits rotating to form a constructive basic Solar system.
- A Basic rendering of a cube , Cylinder and 2 Spheres with Animation.

Details of Implementation :

I. 1. Used the Cube Geometry function readily Available in the Three JS API to Render a Cube.
II. I have created Node Elements which are number Boxes to get user inputs for the width, length and height of the cube to rendered.
III. I had to learn how to inject HTML elements dynamically through W3 Schools.
IV. The infected Elements are modeled and Styled using CSS to be placed at particular Positions.
V. The 3D rendering is done once the user Enters all the three fields.
VI. There are No checks placed to force the user to enter all Inputs.
VII. Used Primitive renderings of a sphere to pose as the planets in our solar system,

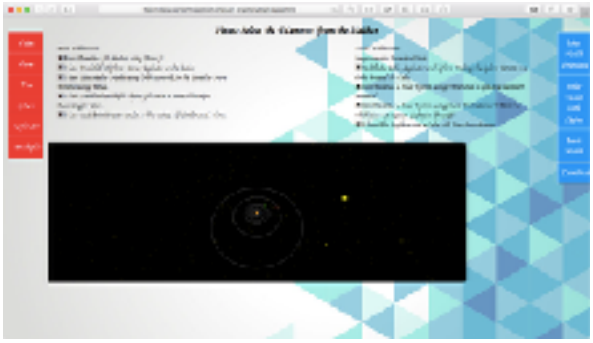
VIII. I have user particle geometry and Randomly spread the particles which are cubes to form Stars surrounding my Solar System.

IX. I have manually set the Position of the orbits and Planets for a Better View of the Rendering .I have created a Three Group to add All the Planets .I used Mesh Basic material in this rendering which does not have any effects when lights are introduced using the API.

X. I have user particle geometry and Randomly spread the particles which are cubes to form Stars surrounding my Solar System.I have manually set the Position of the orbits and Planets for a Better View of the Rendering . I have created a Three Group to add All the Planets .

XI. I used Mesh Basic material in the above rendering which does not have any effects when lights are introduced using the API.So, I used Mesh Lambert Material which has lighting features on them in the second rendering.

XII.I have Made the 2 Spheres move around the cube and Cylinder. This was the basic render which helped me create the solar System.



Week - 3 :

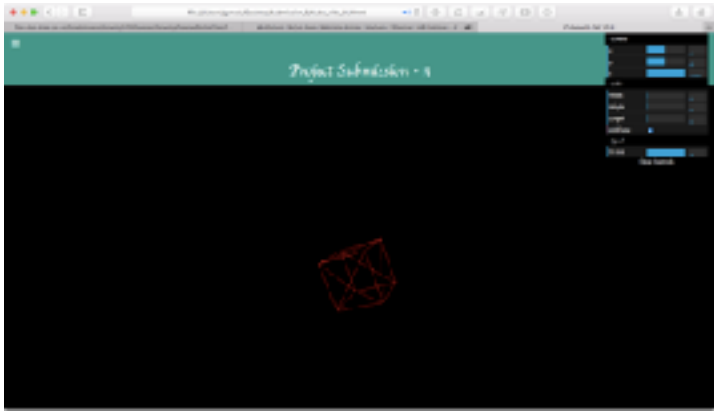
Module Number	Model	Features
1	Cube	• Movement of Object using trackball Controls. • Viewing Multiple Faces of Cube. • Generation of Sphere Using Cube Geometry. • Dynamic Cube Dimensions. • Dat GUI used for User Interface. • Scaling of the Object was Achieved. • Camera Positioning of Object was Achieved.
2	Sphere	• Generation of Sphere Using Sphere Geometry. • Wireframe Material Enabled. • Dynamic Sphere Dimensions. • Dat GUI used for User Interface. • Scaling of the Object was Achieved. • Camera Positioning of Object was Achieved.
3	Cone	• Generation of Cone Using Cone Geometry. • Wireframe Material Enabled. • Dynamic Cone Dimensions. • Dat GUI used for User Interface. • Scaling of the Object was Achieved. • Camera Positioning of Object was Achieved.
4	Cylinder	• Generation of Cylinder Using Cylinder Geometry. • Wireframe Material Enabled. • Dynamic Cylinder Dimensions. • Dat GUI used for User Interface. • Scaling of the Object was Achieved. • Camera Positioning of Object was Achieved.

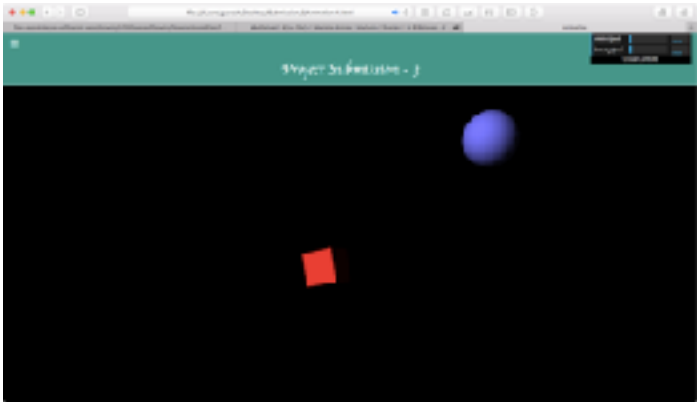
Module Number	Model	Features
5	Texture Mapped Sphere	• Texture Mapping a Sphere With a .jpg file. • Directional Lighting was used to Maintain Light on the sphere. • Rotation was Enabled. • Vanishing Points Control using Orbit Control . JS
6	Texture Mapped Cube	• Texture Mapping a Cube With a .jpg file. • Directional Lighting was used to Maintain Light on the Cube. • Rotation was Enabled. • Vanishing Points Control using Orbit Control . JS
7	Texture Mapped Sphere with Bump Map	• Texture Mapping a Sphere With a .jpg file. • Directional Lighting was used to Maintain Light on the sphere. • Rotation was Enabled. • Vanishing Points Control using Orbit Control . JS • Bump Map was Added with a Bump Scale of 1 onto the Sphere.
8	Animation 2	• Animation of a Bouncing Ball and Rotating Cube controlled Using a GUI. • The GUI controls Have rotation speed and Bounce speed. • The bouncing effect is due to position change.

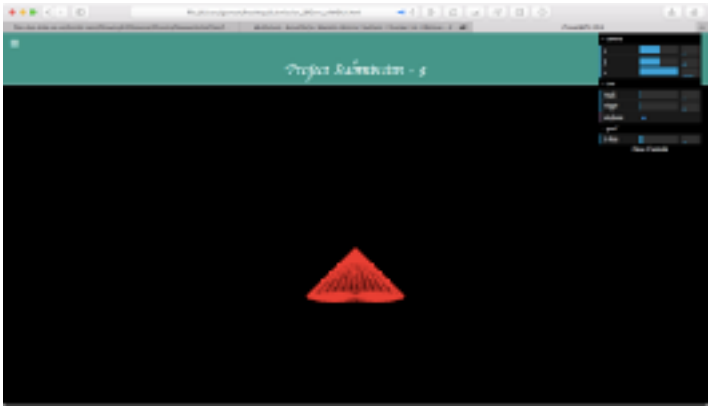
Highlights:

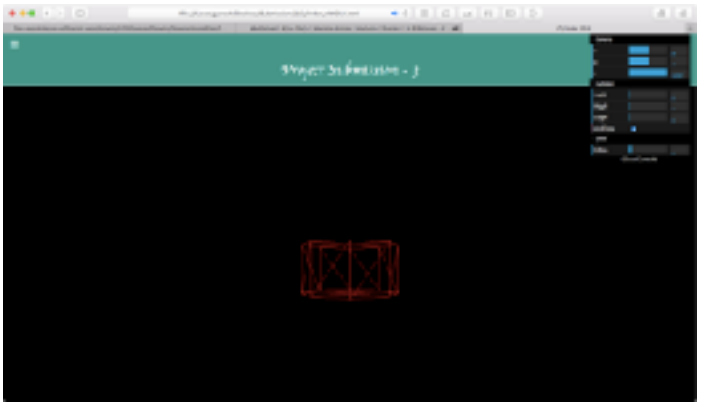
- I have used DAT GUI with Three JS to Simulate 3D objects with change of scale and Transformation.
- DAT GUI is a graphical user interface which is a light weight controller in javascript library to provide a User interface which can help scale and transform or perform operations on the Three JS rendered simulation.
- Cube Rendered in three JS with GUI. It has options of changing width, height and length of the cube along with scaling the cube in X,Y,Z directions.
- Cone Rendered in three JS with GUI. It has options of changing width, height and length of the cube along with scaling the cube in X,Y,Z directions.
- Sphere Rendered in three JS with GUI. It has options of changing width, height and length of the cube along with scaling the cube in X,Y,Z directions.
- Cylinder Rendered in three JS with GUI. It has options of changing width, height and length of the cube along with scaling the cube in X,Y,Z directions.
- Simple Animation of a cube and Sphere with Bouncing and Rotation Speeds controlled through the GUI.
- A 3D Mapped Sphere resembling a Earth. I have Blended the texture of Earth onto a Sphere with Orbit controls to turn and view the Sphere in elevated Angles.

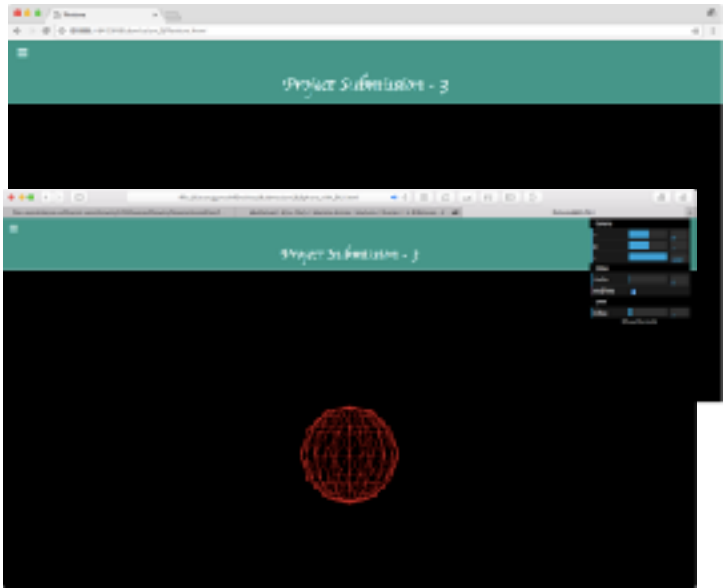
- A 3D Mapped Cube. I have Blended the texture of Earth onto a Sphere with Orbit controls to turn and view the Cube in elevated Angles. I have used the same texture that has been used on the sphere.
- A 3D Mapped Sphere resembling a Earth. I have Blended the texture and bump Maps of Earth onto a Sphere with Orbit controls to turn and view the Sphere in elevated Angles. I have added bump map of the earth to have a better rendering Earth.

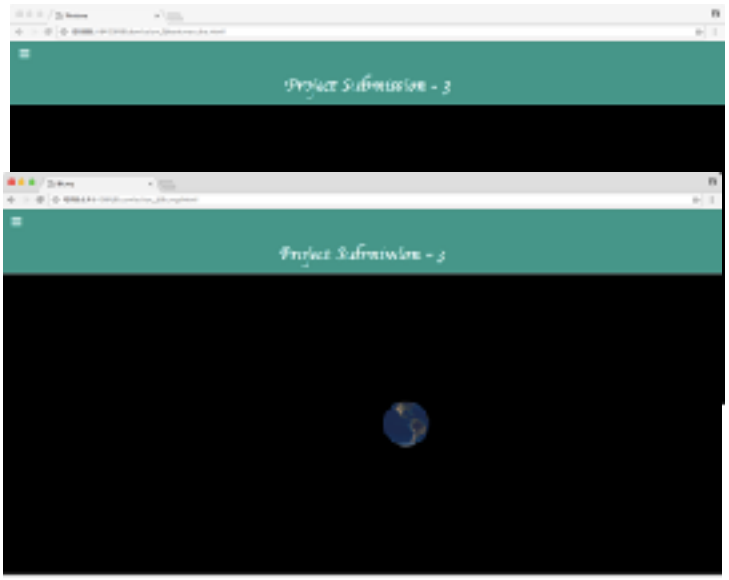












Week - 4 :

Module Number	Model	Features
1	Shearing	• Movement of Object using Orbit Controls. • Viewing Multiple Faces of Objects. • HTML DOM Injection type of Object and Shear Values. • Used a Shear Matrix to Shear the Objects rendered as per user Values. • Users have Options to Input the Shearing needed in the Object. • Vanishing points on Zooming out from the cube Rendered. • The Objects with shearing enabled is cube, sphere, cylinder, Cone
2	Transformation	• Movement of Object using Transform Controls. • Viewing Multiple Faces of Objects. • Used Transform Helper to Translate the generated object in X,Y,Z directions using the helper arrows. • Vanishing points on Zooming out from the cube Rendered. • The Objects with translation enabled is cube, sphere, cylinder, Cone
3	A interactive Solar System	• Generation of a 9 Spheres representing a Planet. • Mesh Phong Material Used. • Tabs to View Each Planet. • Directional Lights in All Planets. • Hemisphere Light with Orange Color Used on SUn. • Every Planet Is Bump and Texture Mapped. • The Size of the Planets / Sphere is Hardcoded. • Zoom in and Zoom out Controls Enabled using port Controls. • Rotation of the planets Enabled.
4	A Car Axle	• Generation of 4 car tires with rims. • A car Under Body . • Car Tyre is made through torus Geometry. • Rims are made through Cylinder Geometry. • UnderBody is Made using a Cube. • All the Geometry are placed in a Group. • The positions of the Geometry is hard coded. • The Animation is through rotating the Cylinder and Torus Geometry along Z axis,

Highlights:

- Shearing : I have created a Module to Shear a Primitive Object by using Shear Matrix available in the Three JS build.
- Transformation : I have used Transform Controls to transform the object in the canvas which is projected.I have used transform controls to transform all primitive objects in a set plane.
- A 3D Mapped(Texture and Bump) planetary System.Have used Hemisphere Lights to highlight the texture and Bump Maps

NOTE : Please refresh After Each render as it consumes a lot of Memory and GPU.

- A 3D rendering of a car Axle with the base of a car using Cylinder and Torus Geometry.

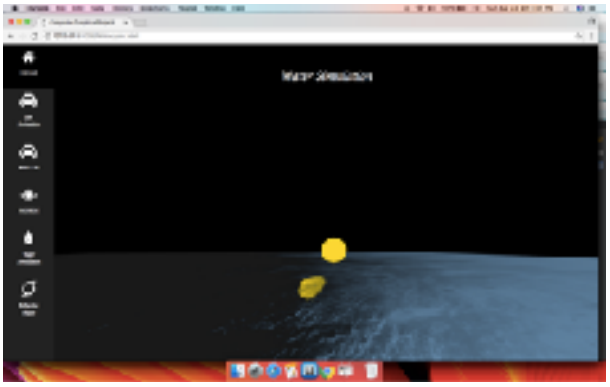
Week - 5 :

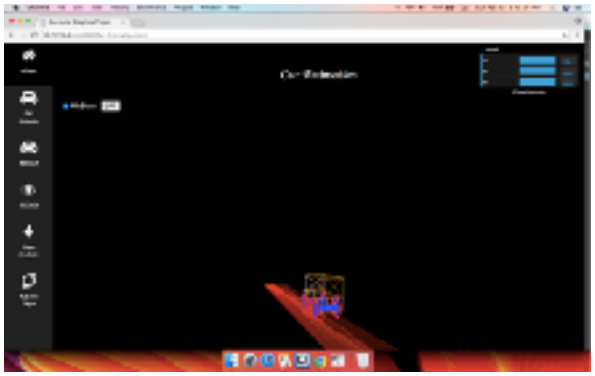
Module Number	Model	Features
1	Basic Car.	• Generation of 4 car tires with rims. • A car Under Body . • A car Body with Cubes. • Car Tyre is made through torus Geometry. • Rims are made through Cylinder Geometry. • UnderBody is Made using a Cube. • All the Geometry are placed in a Group. • The positions of the Geometry is hard coded. • The Animation is through rotating the Cylinder and Torus Geometry along Z axis, • Wireframe Enabled. • Mesh Basic Material Used. • Directional Lights Used as Headlights Place in the Sphere
2	Car Animated.	• Generation of 4 car tires with rims. • A car Under Body . • A car Body with Cubes. • Car Tyre is made through torus Geometry. • Rims are made through Cylinder Geometry. • UnderBody is Made using a Cube. • All the Geometry are placed in a Group. • The positions of the Geometry is hard coded. • The Animation is through rotating the Cylinder and Torus Geometry along Z axis, • Wireframe Enabled. • Mesh Basic Material Used. • A plane is Positioned As a Road and is Texture Mapped. • Animation wit tires turning and position of The Car is Changed with Each render. • Directional Lights Used as Headlights Place in the Sphere
3	Skybox	• Generation of Cube Using Cone Geometry. • Cube Is Texture Mapped. • The Mapping is On Both Sides to create an Effect of a Skybox. • Individual Images are Mapped on Each face of the cube to Provide a 360 degree View of the Place. • Can Zoom in and Zoom Out and Move around the Skybox with the Mouse.

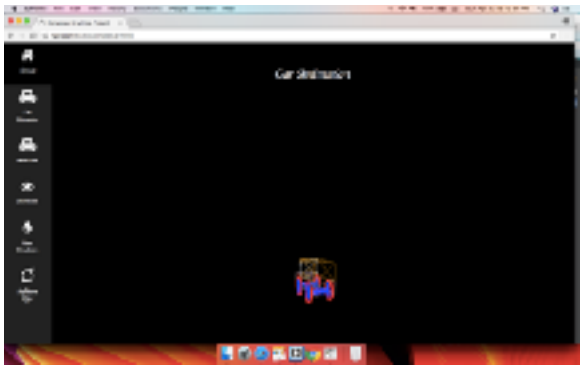
Module Number	Model	Features
4	Water Simulation	• Water JS is used to simulate a Flowing water. • A plane buffer Geometry Used . • Texture of water is Mapped on the Plane Buffer Geometry. • A function called water is Used and the color of the water is set. • A uniform Shader is Used to create a ripple effect.
5	Reflective Material	• Generation of Cube Using Cone Geometry. • Cube Is Texture Mapped. • The Mapping is On Both Sides to create an Effect of a Skybox. • Individual Images are Mapped on Each face of the cube to Provide a 360 degree View of the Place. • Can Zoom in and Zoom Out and Move around the Skybox with the Mouse. • A sphere Place in Between the Skybox. • The Zoom in and Out is Restricted . • A cube Camera is Used to render the Image on the Sphere to Create a Reflective Material. • The reflectivity is set to 1. • A reflection the Texture Mapped Cube is seen on the Sphere as it rotates. • Environment Mapping done using cube camera.

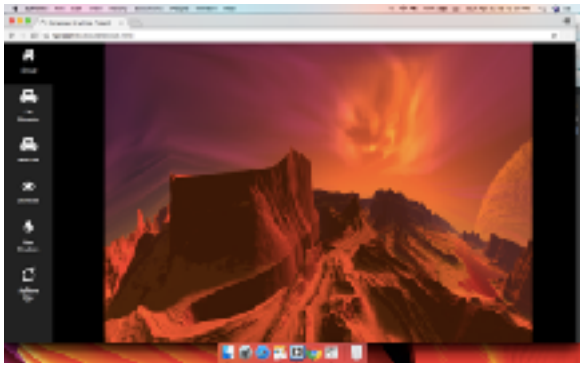
Highlights:

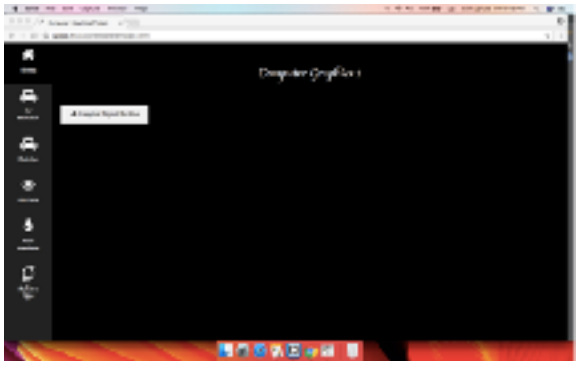
- Car : This is a improvement from the previous car Axle Submission. I have Built a Basic Car.
- Car Animation : The car built is Animated and I have used a Pane Geometry with texture to simulate. A road.
- Skybox : It is a Basic Skybox Wrapped around a Cube with a 360 degree Moveable view.
- Reflective Object : It basically is environment Mapping on a sphere to display a reflective Object .
- Water Simulation : Using the Water.js build in three JS I have simulated a Flowing Ocean.







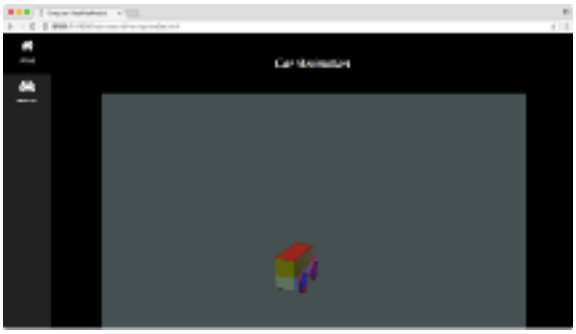


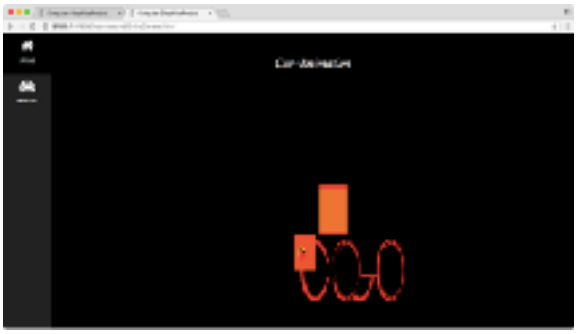


Week -6 :

Module Number	Model	Features
1	Car with Spotlight	• Generation of 4 car tires with rims. • A car Under Body . • A car Body with Cubes. • Car Tyre is made through torus Geometry. • Rims are made through Cylinder Geometry. • UnderBody is Made using a Cube. • All the Geometry are placed in a Group. • The positions of the Geometry is hard coded. • MeshPhong Material Used. • A plane is Positioned As a Road and is Texture Mapped. • Animation wit tires turning and position of The Car is Changed with Each render. • Directional Lights Used as Headlights Place in the Sphere • Spot Lights Used to Illuminate the Car
2	Car With Point Lights	• Generation of 4 car tires with rims. • A car Under Body . • A car Body with Cubes. • Car Tyre is made through torus Geometry. • Rims are made through Cylinder Geometry. • UnderBody is Made using a Cube. • All the Geometry are placed in a Group. • The positions of the Geometry is hard coded. • MeshPhong Material Used. • A plane is Positioned As a Road and is Texture Mapped. • Animation wit tires turning and position of The Car is Changed with Each render. • Directional Lights Used as Headlights Place in the Sphere • Point Lights Used to Illuminate the Car

Module Number	Model	Features
3	Car with Orthographic Camera	• Generation of 4 car tires with rims. • A car Under Body . • A car Body with Cubes. • Car Tyre is made through torus Geometry. • Rims are made through Cylinder Geometry. • UnderBody is Made using a Cube. • All the Geometry are placed in a Group. • The positions of the Geometry is hard coded. • The Animation is through rotating the Cylinder and Torus Geometry along Z axis, • Wireframe Enabled. • MeshPhong Material Used. • Animation wit tires turning and position of The Car is Changed with Each render. • Directional Lights Used as Headlights Place in the Sphere • Orthographic Camera Used.





Accomplished :

I have Used the below :

Camera Types Used	Light Types Used	Material	Geometries	External Controls
Perspective	SpotLight	Mesh Basic	Cube	Transform
Orthographic	Directional Light	Mesh Standard	Sphere	Trackball
Cube	Ambient Light	Mesh Phong	Cone	Orbit
	Point Light	Mesh Dashed Line	Cylinder	Water
			Torus	
			Circle	
			Plane	
			Plane Buffer	
			Circle Buffer	

I have successfully completed the Following parts :

- Shearing
- Transformation
- Camera Controls to view all Faces
- Vanishing Point.
- Environment Mapping
- Bump Mapping
- Texture Mapping
- Light Sources .
- Camera Sources.
- Materials and Geometry.
- Modeling of 3 D Objects.
-

Software Required :

- Brackets (Text Editor for HTML,CSS,JS)
- THREE JS Build.
- DAT GUI Build.
- Chrome Web Browser.

INSTRUCTIONS TO EXECUTE:

Please Have Internet connection as the Javascript for Three JS is a http Link .
Please view the Website In Full screen.

REFERENCES :

- https://threejs.org
- https://www.youtube.com/watch?v=YKzyhcyAijo
- https://www.youtube.com/watch?v=lshPMbN5ws8
- I have used various web resources.
- Three Js Beginner tutorials to construct a Cube on Youtube.
- I have used the Official Three JS module from the Three JS website to download the Build and orbit control Files.
- I have visited three JS website to find out about the materials and different types of Geometry's available to construct the Website.
- I have Used CSS tricks.com to style my Web Page.
- I have used Stack over Flow to figure out the geometry of a orbit.
- I also used Youtube videos to learn on how to light a canvas using Three JS.
- CSS web Templates.
- Youtube videos on how to Implement Water JS in Three JS.