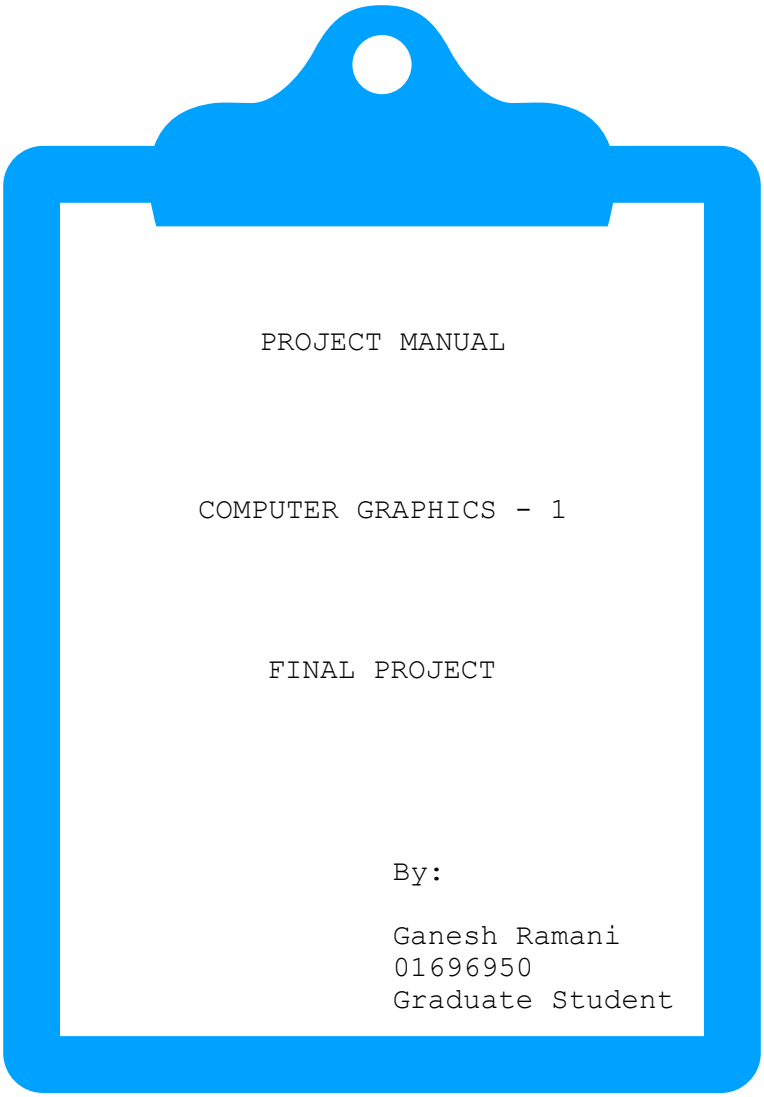

PROJECT MANUAL

COMPUTER GRAPHICS - 1

FINAL PROJECT

By:

Ganesh Ramani
01696950
Graduate Student

TABLE OF CONTENTS

The Modules From Week 1 to Week 6

Week-1:

Module 1 :Cube:

- A 3D cube with set dimensions.
- Wireframe Enabled.
- External module called Trackball JS used to move cube.
- No Rotation or other effects in the render.

Module 2 : Sphere:

- A 3D Sphere with set dimensions.
- Wireframe Enabled.
- External module called Trackball JS used to move cube.
- No Rotation or other effects in the render.

Module 3 : Cone:

- A 3D Cone with set dimensions.
- Wireframe Enabled.
- Rotation Effect added in render.

Module 4 : Cylinder:

- A 3D Cylinder with set dimensions.
- Wireframe Enabled.
- Rotation Effect added in render

Module 5 : Tree:

- A 3D Tree with a cylinder as stem and Cone as the leaves with set dimensions.
- Wireframe Enabled.
- External module called Trackball JS used to move Tree.
- Rotation Effect added in render

Module 6 : Random Objects Generation:

- A 3D Random Object (cube, sphere ,cylinder, cone) with set dimensions.
- Wireframe Enabled.
- External module called Trackball JS used to move cube.
- No Rotation or other effects in the render.
- User can select the object and number of objects through a slider input.
- The objects are generated at a pre-determined Distance from the initial position.

Week-2:

Module 1 : Primitive Solar System:

- A primitive Solar System with orbit.
- A THREE.Group is created For all the Planets represented by Spheres.
- The Stars on the background is a random Spread of circles which are placed around the Group Model.
- To Make the solar system I have used sphere each with its own size to differentiate the planets.
- MESH basic material is used in the module and no Additional Lights are Used.
- The Positions and Rotation speed of the planets are all pre-determined and hard coded.
- The planets placed in the group is rotated and position increment takes place.

Module 2 : Solar System with Lights:

- A primitive Solar System with orbit.
- A THREE.Group is created For all the Planets represented by Spheres.
- The Stars on the background is a random Spread of circles which are placed around the Group Model.
- To Make the solar system I have used sphere each with its own size to differentiate the planets.
- MESH PHONG material is used in the module and no Additional Lights are Used.
- The Positions and Rotation speed of the planets are all pre-determined and hard coded.
- The planets placed in the group is rotated and position increment takes place.
- Ambient Lights are added to illuminate the Material.

Module 3 :Basic Animation:

- A basic Animated Model of 2 Sphere Orbiting a cube and Cylinder.
- The Mesh Basic Material is used in rendering the Object.
- The Position of the sphere is constantly changed at a incremental rate.

Module 4 :Basic Animation:

- A Cube is rendered based on the User input.
- I have dynamically injected the Input elements with javascript to render the cube.
- And the Length breadth and height if the cue is determined by the User .
- It is a mash basic material with wireframe enabled as true.
- The Cube rendered has orbit controls to view the cube in different angles.

- The user input is limited only to the dimensions and not the type of material or color.

Week-3 :

Module 1 :Cube with DAT GUI

```
var cam = gui.addFolder('Camera');
cam.add(camera.position, 'x', -1000, 1000).listen();
cam.add(camera.position, 'y', -1000, 1000).listen();
cam.add(camera.position, 'z', -1000, 1000).listen();
cam.open();

var box = gui.addFolder('Cube');
box.add(cube.scale, 'x', 0, 100).name('Width').listen();
box.add(cube.scale, 'y', 0, 100).name('Height').listen();
box.add(cube.scale, 'z', 0, 100).name('Length').listen();
box.add(cube.material, 'wireframe').listen();
box.open();

var speed= gui.addFolder('Speed');
model= speed.add(cube.rotation,'x',0,0.2).name("X-Axis").listen
    speed.open();

    model.onChange(animate);
```

Camera Position	X	Y	Z
Minimum	-1000	-1000	-1000
Maximum	1000	1000	1000

Scaling	X	Y	Z
Minimum	0	0	0
Maximum	100	100	100

Material	Wireframe
Enabled	Default
Solid	Option

Rotation	X	Y	Z
	Enabled	Not ENabled	Not Enabled

The User Does not have the Option to input the Length , Breadth and Height of the Cube as it is Hard-coded.

This Module has been Implemented to show User Control over :
- Rotation along X axis.
- Scaling.
- Camera Position Movement.
- Material Change through Checkbox.

The User does not have the Access to Change :
- Color.
- Material Type.
- Specify the dimensions of the cube.

Module 2 :Cube with DAT GUI

Scaling	X	Y	Z
Minimum	0	0	0
Maximum	100	100	100

Material	Wireframe
Enabled	Default
Solid	Option

Rotation	X	Y	Z
	Enabled	Not ENabled	Not Enabled

The User Does not have the Option to input the Height and Radius of the Cone as it is Hard-coded.

This Module has been Implemented to show User Control over :
- Rotation along X axis.
- Scaling.
- Camera Position Movement.
- Material Change through Checkbox.

The User does not have the Access to Change :
- Color.
- Material Type.
- Specify the dimensions of the Cone.

Module 3 :Sphere with DAT GUI

Scaling	X	Y	Z
Minimum	0	0	0
Maximum	100	100	100

Material	Wireframe
Enabled	Default
Solid	Option

Rotation	X	Y	Z
	Enabled	Not ENabled	Not Enabled

The User Does not have the Option to input the Radius of the Sphere as it is Hard-coded.

This Module has been Implemented to show User Control over :
- Rotation along X axis.
- Scaling.
- Camera Position Movement.
- Material Change through Checkbox.

The User does not have the Access to Change :
- Color.
- Material Type.
- Specify the dimensions of the Sphere.

Module 4 :Cylinder with DAT GUI

Scaling	X	Y	Z
Minimum	0	0	0
Maximum	100	100	100

Material	Wireframe
Enabled	Default
Solid	Option

Rotation	X	Y	Z
	Enabled	Not ENabled	Not Enabled

The User Does not have the Option to input the Height and Radius of the Cylinder as it is Hard-coded.

This Module has been Implemented to show User Control over :
- Rotation along X axis.
- Scaling.
- Camera Position Movement.
- Material Change through Checkbox.

The User does not have the Access to Change :
- Color.
- Material Type.
- Specify the dimensions of the Cylinder.

Module 5 :Animation

- I have used a Sphere and Cube of Mesh Lambert Material.
- The Size and Dimensions of the figures are Hardcoded.

The User has the Ability to Control the Bouncing Speed of the Ball and the rotation speed of the Cube.

The Rotation has a Maximum and Minimum values 0.0 to 0.5 Respectively.
The Bouncing has a Maximum and Minimum values 0.0 to 0.5 Respectively.

- The values can be changed using the Slider on the Dat GUI panel.
- The values on change will be iteratively used in the render Animation function to increase and decrease the Speeds as per User Input.

A spotlight is used to Illuminate the Entire Plane.
It is set at a position of (-40, 60, -10) to illuminate the Animation.

Module 6 & 7 & 8 :Texture and Bump Mapping on Cube and Sphere

- This Module is entirely to Highlight the Usage of Mapping of textures on A 3D object.
- Mesh Phong materials are Used as it has property of doing Shine on to the object .
- The Image used is that of Earth which is Mapped onto Both a cube and Sphere.
- Bump map of the Earth obtained through the internet is mapped on Sphere to create the Effect of Mountains and Plateaus of the Earth

Observations are :

A. For A cube the Image is Mapped on each face separately and not entirely Mapped as a single image but as 8 images.
B. For a Sphere the Wrapping is a Single Image for the Entire Body.
C. We need Lights to illuminate the Image Mapped else it is rendered as a Black object on to the Screen.
D. Bump Map is mapped Along with the texture to add depth and make depth visible in the sphere.

Week-4:

Module 1 : Shearing of Primitive Objects.

Camera Position	X	Y	Z
Set Position	-1000	-1000	-1000

Material	Wireframe
Enabled	Wireframe

Available Options
CUBE
SPHERE
CONE
CYLINDER

Rotation	X	Y	Z
	Not Enabled	Not Enabled	Not Enabled

Shear Paramters	X	Y	Z
Minimum	0	0	0
Maximum	32767	32767	32767

External Controls used
OrbitControls.js

The Shear Matrix is Represented in the Below Format in the Program :

The Input Parameters are Displayed on the Page Correspond to :

Shear Paramters	X	Y	Z
S	Szx	Syx	Syz

Shear Paramters	Sxy	Szy	Sxz
Value Assigned	0	0	0

```
//          | 1    Syx  Szx  0 |
//          |                   |
//          | Sxy  1    Szy  0 |
//      M = |                   |
//          | Sxz  Syz  1    0 |
//          |                   |
//          | 0    0    0    1 |
//          |                   |
```

Module 2 : Transformation of Primitive Objects.

Camera Position	X	Y	Z
Set Position	1000	500	1000

Material	Wireframe
Enabled	Wireframe

Available Options
CUBE
SPHERE
CONE
CYLINDER

External Controls used
TransformControls.js

Module 3 : Car Axle:

Camera Position	X	Y	Z
Set Position	1000	0	0

External Controls used
OrbitControls.js

Object Positions	X	Y	Z
Wheel 1	0	0	0
Wheel 2	0	0	500
Wheel 3	700	0	0
Wheel 4	700	0	500
RIM1	0	0	0
RIM2	0	0	500
RIM3	700	0	0
RIM4	700	0	250

Object Positions	X	Y	Z
Axle1	0	0	250
Axle 2	700	0	250
Car Base	350	0	250

Position of Objects Set	Wireframe	Geometry	Radius
Wheel 1	Yes	Torus	Hardcoded
Wheel 1	Yes	Torus	Hardcoded
Wheel 1	Yes	Torus	Hardcoded
Wheel 1	Yes	Torus	Hardcoded
RIM1	Yes	Cylinder	Hardcoded
RIM1	Yes	Cylinder	Hardcoded
RIM1	Yes	Cylinder	Hardcoded
RIM1	Yes	Cylinder	Hardcoded
Axle1	No	Cylinder	Hardcoded
Axle 2	No	Cylinder	Hardcoded
Car Base	No	Box	Hardcoded

Module 4 : Solar System with Texture and Bump Maps:

External Controls used
OrbitControls.js
Minimum Distance - 50
Maximum Distance - 300

Object Positions	Sphere Radius	Texture	Bump Map	Light Colour	Light Type
Mercury	30	YES	YES	White	Directional
Venus	30	YES	YES	White	Directional
Earth	30	YES	YES	White	Directional
Mars	30	YES	NO	White	Directional
Jupiter	30	YES	NO	White	Directional
Neptune	30	YES	NO	White	Directional
Saturn	30	YES	NO	White	Hempishere
Uranus	30	YES	NO	White	Directional
Sun	30	YES	NO	Orange	Hemisphere

Observations:

- Enabling Both Texture and Bump Maps and rendering Multiple Objects has a lot of Load effect on the Laptop GPU.
- On Multiple Renders the rotation slows down on the planets and affects the whole system performance.
- The Shear Matrix skews the Image based on the user Input.
- The transform Controls have a useful grid helper to transform the Objects.
- On transforming once the object reach the Far end of the Clipping Plane Vanish.
- The Car Axle Model has Multiple Objects which need to rotate Simultaneously to Animate the Render.
- Mesh Lambert and Mesh Phong Materials Need Light to be illuminate in the object to see the color on the surface.

Week-5:

Module 1 : Simple Car.

- The Simple Car is a Basic 3D render of a car .
- The Car Model has been constructed using the following Parameters:

  1. The positions of the Car objects are all hard Coded and have been placed by using Calculations .
  2. The Color of each object in the car is Unique to Identify Each individual Component in the Car.
  3. The Lights used in the construction is a Directional Light and is placed inside the Sphere to Produce a Dim Headlight Effect in the Model.
  4. The Geometries Used in designing the Car are :
     - Sphere (Headlights)
     - Torus (Wheels)
     - Cylinder (Axle and Rims)
     - Cube (Body)
- The Car has the Following Specifications of Each Geometry and Camera Movements .

	Color	Specifications	Rotation	Material	Geometry	Positioning (X,Y,Z)
Headlights	White	Radius : 50	No	Mesh Phong	Sphere	-350,120,350
Rims	Blue	Radius : 200	No	Mesh Basic	Cylinder	0,0,250
Wheels	Red	Radius : 200	No	Mesh Basic	Torus	700,0,500 700,0,0
Body	Pink,Green, Orange,White	(L,B,H): 1400,250,450	No	Mesh Basic	Cube	350,120,250 550,420,250 50,420,250
Axle	Blue	Radius : 50	No	Mesh Basic	Cylinder	700,0,250 700,0,0

- Orbit Controls is Used as an external JS to move and transform around and Inside the 3 D object.
- Camera is Positioned at (5000,1000,2000)(X,Y,Z) Positions.
- Orbit controls has Zoom Enabled.

Module 2 : Car Animation.

- The Simple Car Animation is a Basic 3D render of a car moving on a road.
- The Car Model has been constructed using the following Parameters:

  1. The positions of the Car objects are all hard Coded and have been placed by using Calculations .
  2. The Color of each object in the car is Unique to Identify Each individual Component in the Car.
  3. The Lights used in the construction is a Directional Light and is placed inside the Sphere to Produce a Dim Headlight Effect in the Model.
  4. The Geometries Used in designing the Car are :
     - Sphere (Headlights)
     - Torus (Wheels)
     - Cylinder (Axle and Rims)
     - Cube (Body)
     - Plane(road)
- The Car has the Following Specifications of Each Geometry and Camera Movements .

	Color	Specifications	Rotation	Material	Geometry	Positioning (X,Y,Z)
Headlights	White	Radius : 50	No	Mesh Phong	Sphere	-350,120,350
Rims	Blue	Radius : 200	Yes	Mesh Basic	Cylinder	0,0,250
Wheels	Red	Radius : 200	Yes	Mesh Basic	Torus	700,0,500 700,0,0
Body	Pink,Green, Orange,White	Aprrox Length,Breadth and Height : 1400,250,450	No	Mesh Basic	Cube	350,120,250 550,420,250 50,420,250
Axle	Blue	Radius : 50	yes	Mesh Basic	Cylinder	700,0,250 700,0,0
Road	Texture Mapped		No	Mesh basic	Plane	

- Orbit Controls is Used as an external JS to move and transform around and Inside the 3 D object.
- DAT GUI is used to provide a easy animation movement of the car
- Camera is Positioned at (5000,1000,2000)(X,Y,Z) Positions.
- Orbit controls has Zoom Enabled.
- The Road is Texture Mapped by using a JPG File.
- The Animation is due to change in projections of the Group by a Incrementation of position in the X axis by +0.5 at each render .

- Dat GUI is used to control of scale the camera and increase the Speed.

Module 3 : Skybox (360 Degree View):

- I have tried to Create a Skybox or a 360 Degree View in by Mapping texture on the Cube.
- The Skybox has a Bloody Valley Range texture Mapped on a Cube .
- The Cube is of the Dimension (L:100,B:100,H:100).
- The Cube is Mapped with a texture Named Front , Back, left, Right, Top, Bottom on each side .
- The Side or the Wrapping used in the applying Texture is THREE.DoubleSide which maps the texture on both Sides of the Cube .
- The Camera is Positioned at the Centre of the Cube and is Controlled using Orbit Control file provided with the THREE JS build .
- The Camera's Maximum Zoom out is set at 40 to limit the camera from Moving out of the Cube.
- So on rotation we can see a 360 Degree view of the Blood Valley.
- We can download Skybox files from the internet to apply various Textures on the cube.

Module 4 : Water Simulation:

- The water Simulation is by using a external JS in the three JS build called water JS.
- The simulation uses a Plane Buffer Geometry which in turn calls a Water Function in the water JS file.
- The Material Used is called Water .
- The Texture is Loaded onto the Material and Wrapped Both Sides.
- water.material.uniforms.time.value += 1.0 / 20.0; is Updated at each Render to produced a Ripple Effect.

Module 5 : Reflection:

(Outer Box)

- I have tried to Create a Skybox or a 360 Degree View in by Mapping texture on the Cube.
- The Skybox has a Bloody Valley Range texture Mapped on a Cube .
- The Cube is of the Dimension (L:100,B:100,H:100).
- The Cube is Mapped with a texture Named Front , Back, left, Right, Top, Bottom on each side .
- The Side or the Wrapping used in the applying Texture is THREE.DoubleSide which maps the texture on both Sides of the Cube .
- The Camera is Positioned at the Centre of the Cube and is Controlled using Orbit Control file provided with the THREE JS build .
- The Camera's Maximum Zoom out is set at 40 to limit the camera from Moving out of the Cube.
- So on rotation we can see a 360 Degree view of the Blood Valley.
- We can download Skybox files from the internet to apply various Textures on the cube.

(Reflective Material)

- Created a Sphere Placed in between the Skybox rendered Above.

- The Sphere of radius 10 .
- The Sphere is rendered using a Mesh Basic Material .
- A cube Camera is used and the environment mapping on the sphere is (mirrorCubeCamera.renderTarget) which projects the Mirror Effect in the Sphere.
- The Mirror Cube Camera is Updated at each render to project the reflection on the Sphere.
- Cube Camera Creates 6 cameras that render to a WebGLRenderTargetCube.

Week-6:

Module 1 : Car with Point Lights :

- The Simple Car is a Basic 3D render of a car .
- The Car Model has been constructed using the following Parameters:

  1. The positions of the Car objects are all hard Coded and have been placed by using Calculations .
  2. The Color of each object in the car is Unique to Identify Each individual Component in the Car.
  3. The Lights used in the construction is a Directional Light and is placed inside the Sphere to Produce a Dim Headlight Effect in the Model.
  4. The Geometries Used in designing the Car are :
     - Sphere (Headlights)
     - Torus (Wheels)
     - Cylinder (Axle and Rims)
     - Cube (Body)
- The Car has the Following Specifications of Each Geometry and Camera Movements .

	Color	Specifications	Rotation	Material	Geometry	Positioning (X,Y,Z)
Headlights	White	Radius : 50	No	Mesh Phong	Sphere	-350,120,350
Rims	Blue	Radius : 200	No	Mesh Phong	Cylinder	0,0,250
Wheels	Red	Radius : 200	No	Mesh Phong	Torus	700,0,500 700,0,0
Body	Pink,Green, Orange,White	(L,B,H): 1400,250,450	No	Mesh Phong	Cube	350,120,250 550,420,250 50,420,250
Axle	Blue	Radius : 50	No	Mesh Phong	Cylinder	700,0,250 700,0,0

- Orbit Controls is Used as an external JS to move and transform around and Inside the 3 D object.
- Camera is Positioned at (5000,1000,2000)(X,Y,Z) Positions.
- Orbit controls has Zoom Enabled.
- Point Lights Enabled .

Module 2 : car with spot Lights :

- The Simple Car is a Basic 3D render of a car .
- The Car Model has been constructed using the following Parameters:

  1. The positions of the Car objects are all hard Coded and have been placed by using Calculations .
  2. The Color of each object in the car is Unique to Identify Each individual Component in the Car.
  3. The Lights used in the construction is a Directional Light and is placed inside the Sphere to Produce a Dim Headlight Effect in the Model.
  4. The Geometries Used in designing the Car are :
     - Sphere (Headlights)
     - Torus (Wheels)
     - Cylinder (Axle and Rims)
     - Cube (Body)
- The Car has the Following Specifications of Each Geometry and Camera Movements .

	Color	Specifications	Rotation	Material	Geometry	Positioning (X,Y,Z)
Headlights	White	Radius : 50	No	Mesh Phong	Sphere	-350,120,350
Rims	Blue	Radius : 200	No	Mesh Phong	Cylinder	0,0,250
Wheels	Red	Radius : 200	No	Mesh Phong	Torus	700,0,500 700,0,0
Body	Pink,Green, Orange,White	(L,B,H): 1400,250,450	No	Mesh Phong	Cube	350,120,250 550,420,250 50,420,250
Axle	Blue	Radius : 50	No	Mesh Phong	Cylinder	700,0,250 700,0,0

- Orbit Controls is Used as an external JS to move and transform around and Inside the 3 D object.
- Camera is Positioned at (5000,1000,2000)(X,Y,Z) Positions.
- Orbit controls has Zoom Enabled.
- Spot Lights Enabled .

Module 3 : car with Orthographic Camera:

- The Simple Car is a Basic 3D render of a car .
- The Car Model has been constructed using the following Parameters:

  1. The positions of the Car objects are all hard Coded and have been placed by using Calculations .
  2. The Color of each object in the car is Unique to Identify Each individual Component in the Car.
  3. The Lights used in the construction is a Directional Light and is placed inside the Sphere to Produce a Dim Headlight Effect in the Model.
  4. The Geometries Used in designing the Car are :
     - Sphere (Headlights)
     - Torus (Wheels)
     - Cylinder (Axle and Rims)
     - Cube (Body)
- The Car has the Following Specifications of Each Geometry and Camera Movements .

	Color	Specifications	Rotation	Material	Geometry	Positioning (X,Y,Z)
Headlights	White	Radius : 50	No	Mesh Phong	Sphere	-350,120,350
Rims	Blue	Radius : 200	No	Mesh Phong	Cylinder	0,0,250
Wheels	Red	Radius : 200	No	Mesh Phong	Torus	700,0,500 700,0,0
Body	Pink,Green, Orange,White	(L,B,H): 1400,250,450	No	Mesh Phong	Cube	350,120,250 550,420,250 50,420,250
Axle	Blue	Radius : 50	No	Mesh Phong	Cylinder	700,0,250 700,0,0

- Orbit Controls is Used as an external JS to move and transform around and Inside the 3 D object.
- Camera is Positioned at (5000,1000,2000)(X,Y,Z) Positions.
- Orbit controls has Zoom Enabled.
- Orthographic Camera Enabled