**COMPUTER GRAPHICS PROJECT**

***SUBMITTED BY:***

SHRUTI JAIN

01656195

***SUBMITTED TO :***

DR. HAIM LEVKOWITZ

CHAIR, COMPUTER SCIENCE DEPARTMENT

**IMPLEMENTATION**

The project implements the following features:

1. Application of 3D transformation
2. View of the object from multiple views.
3. Transform camera/viewer/light sources(s).
4. Perspective Vanishing Points.
5. Mapping
6. Window Resize
7. Stats Provider

**IMPLEMENTATION SUMMARY**

1. 3D Transformation
   1. The object, car, can be rotated on either axis. The object has been implemented with a translation into the skybox .
   2. The object can be scaled on either axis . The control bar has the feature but the code has been disabled to not distort the image.
2. Multiple Views
   1. The car and the 3D world (represented by the skybox) can be viewed from either of the axis. To view the 3D world completely , zoom out until the cube appears. Rotate the zoom in either direction on either axis. The same concept applies to the car.
   2. Left click to rotate the world.
   3. Right click to see the panning effect.
3. Transform Camera & Light Sources.
   1. The camera controls on the left controller adjusts the position of the 3D world as per the given coordinate system.
   2. The cameras field of view is user friendly .
   3. Ambient Light and Point light values can be given real time. Also, a constant change of light occurs during the execution. The best way to look at the effects is to disable the translation on all the axis of the group (stated as geo.translate.X()) .
4. Perspective Vanishing Points.
   1. The entire world vanishes after a certain point . Even if an edge reaches the vanishing point, it disappears.
   2. Zoom out completely . After a point the objects will vanish.
5. Mapping
   1. Both the skybox and the car (the tires) maps the texture.
6. Window Resize
   1. As you resize the window panel, The projection of the code inclusive of the camera, field of view and the objects resize accordingly.
7. Stats Provider :JavaScript Performance Monitor
   1. This class provides a simple info box that will help you monitor your code performance.

* **FPS** Frames rendered in the last second. The higher the number the better.
* **MS** Milliseconds needed to render a frame. The lower the number the better.
* **MB** MBytes of allocated memory. (Run Chrome with --enable-precise-memory-info)
* **CUSTOM** User-defined panel support.

**NOTES**

* To execute the program , local server like Servez is preferable for optimal projection.

**BIBLOGRAPGY**

* Threejs.org
* Three Js Documentation
  + <https://threejs.org/docs>
* Ww3schools.org for basic html understanding.
* Three Js tutorials
  + <https://codepen.io/rachsmith/post/beginning-with-3d-webgl-pt-1-the-scene>
  + <https://www.august.com.au/blog/animating-scenes-with-webgl-three-js/>