Assignment - 2

[1]Arjun Verma : IMT2017008
T2-20-CS 606: Computer Graphics
Course Instructors : Prof. Jaya Sreevalsan Nair
Prof. T.K Srikanth

Simple 3D Model Viewer

Abstract. *This technical report contains a brief overview of the methodology involved in solving the given problem statement along with the solutions to the questions posted with the problem statement.*

1. Problem Statement

The given problem statement deals with rendering and manipulation of 3D models. Abstractly, the problem statement requires setting up a scene consisting of 3D X, Y, Z Axes and Blender imported models. The models can then undergo various manipulations such as translations, rotations and scaling. The application also involves model picking and mouse drag camera rotation about one of the primary axes.

2. Methodology

In this section, we discuss the distinct characteristics of the approaches involved in designing the application.

2.1. General Design and Application Instructions

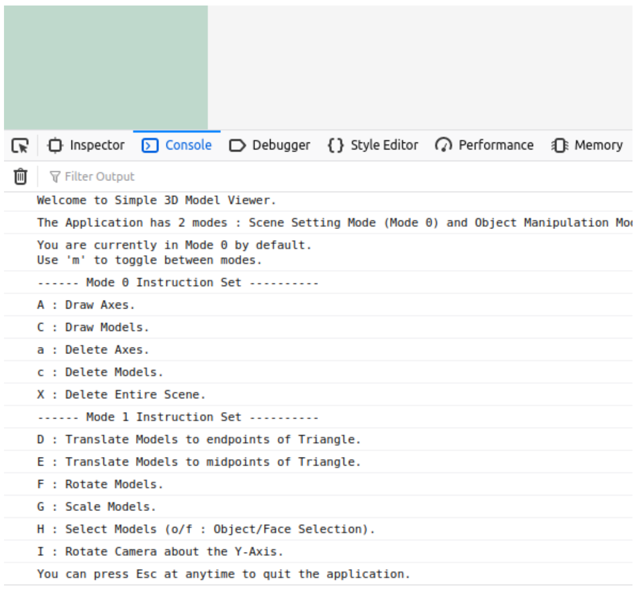
The application has been designed such that it has two modes : Mode 0 and Mode 1. Mode 0 deals with setting up the scene while Mode 1 deals with model manipulation. A user can toggle between the different modes by pressing the 'm' key. A user can try out the different features by enabling that feature action in it's respective mode by pressing the appropriate key. The subsequent action is then click enabled. The application also throws out occasional instructions and messages in the browser console for the ease of use of user. These are simple console.log messages and a sample of it can be seen below with the welcome display message 1.

2.2. Model and Axes Design

The models and axes have been designed and imported from Blender. All models were created manually and then imported as .obj files and read into the application using webgl-obj-loader. The models are essentially designed to represent the different kinds of trees found in three different seasons : Spring, Winter and Autumn.

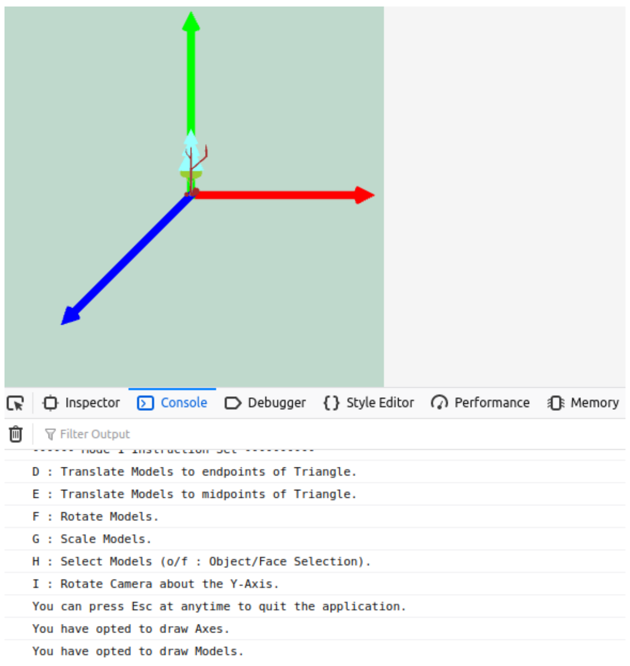
2.3. Mode 0

This is the Scene Setting Mode of the application. In this mode a user can draw Axes or/and Models on the screen by pressing 'A' and 'C' respectively. A user can also choose to delete Axes or/and Models from the Scene by pressing 'a' and 'c' respectively. A user also has at his/her expense to delete the entire scene by pressing 'X'. The sample Model and Axes are shown below 2.



```
Welcome to Simple 3D Model Viewer.
The Application has 2 modes : Scene Setting Mode (Mode 0) and Object Manipulation Mo
You are currently in Mode 0 by default.
Use 'm' to toggle between modes.
------ Mode 0 Instruction Set ----------
A : Draw Axes.
C : Draw Models.
a : Delete Axes.
c : Delete Models.
X : Delete Entire Scene.
------ Mode 1 Instruction Set ----------
D : Translate Models to endpoints of Triangle.
E : Translate Models to midpoints of Triangle.
F : Rotate Models.
G : Scale Models.
H : Select Models (o/f : Object/Face Selection).
I : Rotate Camera about the Y-Axis.
You can press Esc at anytime to quit the application.
```

Figure 1. Welcome Instructions



```
D : Translate Models to endpoints of Triangle.
E : Translate Models to midpoints of Triangle.
F : Rotate Models.
G : Scale Models.
H : Select Models (o/f : Object/Face Selection).
I : Rotate Camera about the Y-Axis.
You can press Esc at anytime to quit the application.
You have opted to draw Axes.
You have opted to draw Models.
```

Figure 2. Axes and Models

2.4. Mode 1

This is the Model Manipulation Mode. The three main components in these mode involve : Transformations, Picking and Camera Rotation. Let us look at each component individually.

2.4.1. Transforming Models

Transforming models involves either translating, rotating or scaling them. These are implemented by maintaining a WorldTransformation Matrix per model. This matrix stores the computations related to the positioning of the model in the world space. For translating the models to the endpoints of a triangle such that it has it's centre is around the origin, the coordinates chosen are (-1.5, -1.5, 0), (1.5, 0, 0) and (0, 1.5, 0). The midpoint translation takes place accordingly. The user can press 'D' and 'E' to perform the respective translations.

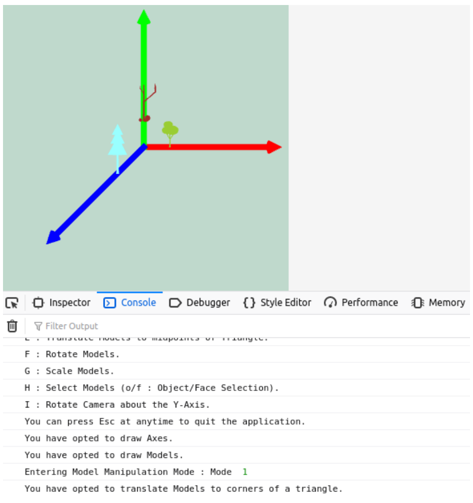
The rotation of models has been implemented in a slightly different but more complete manner than what has been mentioned in the instruction pdf. Rather than just rotating them by a certain angle once and then leave them, on pressing 'F', a user can see the models perform complete rotations such that each model completes a 360 degree rotation about their respective axis every 6 seconds. You can switch to any other state or mode to stop the rotation and the models would retain their respective rotated positions for further manipulations.

The scaling has also been implemented in a slightly deviate manner. The models constructed and imported from Blender were inherently large. As a result of this, it was not feasible to make their original sizes 2x and 3x as mentioned in the pdf. To account for the inherently large size, the models are first scaled down to a factor of 0.2x when they have been initially rendered. The scalings then have been applied to the models in the same ratio as the pdf. So, according to this, the scaling factors 0.5x, 2x and 3x have been mapped to 0.1x, 0.4x and 0.6x of the original imported size, thus preserving the ratio as mentioned in the pdf. The user can press 'G' to initiate this action.

Concisely, the final position of any model at a given time is computed in the vertex shader as,

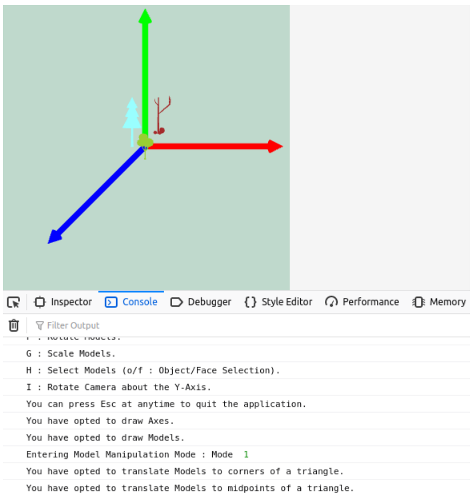
$$ProjectionMat \cdot ViewMat \cdot WorldMat \cdot Vec4Position \tag{1}$$

A few sample transformation displays are shown below in 3, 4, 5 and 6.



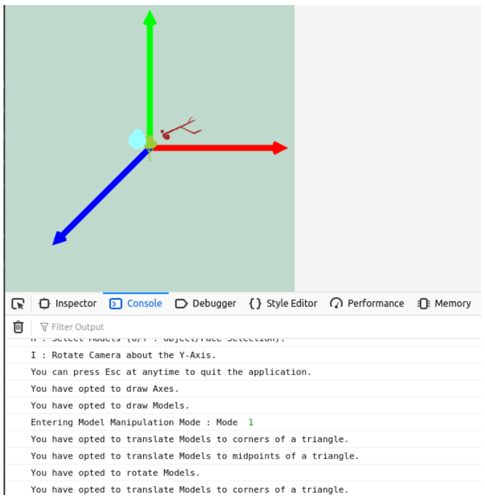
```
E : Translate Models to midpoints of Triangle.
F : Rotate Models.
G : Scale Models.
H : Select Models (o/f : Object/Face Selection).
I : Rotate Camera about the Y-Axis.
You can press Esc at anytime to quit the application.
You have opted to draw Axes.
You have opted to draw Models.
Entering Model Manipulation Mode : Mode  1
You have opted to translate Models to corners of a triangle.
```

Figure 3. Translation on pressing D



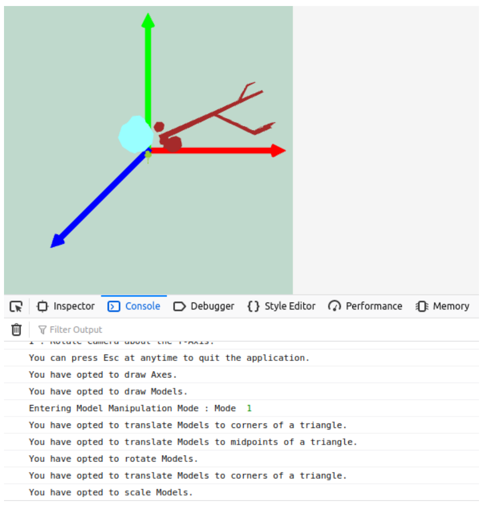
```
F : Rotate Models.
G : Scale Models.
H : Select Models (o/f : Object/Face Selection).
I : Rotate Camera about the Y-Axis.
You can press Esc at anytime to quit the application.
You have opted to draw Axes.
You have opted to draw Models.
Entering Model Manipulation Mode : Mode  1
You have opted to translate Models to corners of a triangle.
You have opted to translate Models to midpoints of a triangle.
```

Figure 4. Translation on pressing E



```
H : Select Models (o/f : Object/Face Selection).
I : Rotate Camera about the Y-Axis.
You can press Esc at anytime to quit the application.
You have opted to draw Axes.
You have opted to draw Models.
Entering Model Manipulation Mode : Mode  1
You have opted to translate Models to corners of a triangle.
You have opted to translate Models to midpoints of a triangle.
You have opted to rotate Models.
You have opted to translate Models to corners of a triangle.
```

Figure 5. Rotated Models

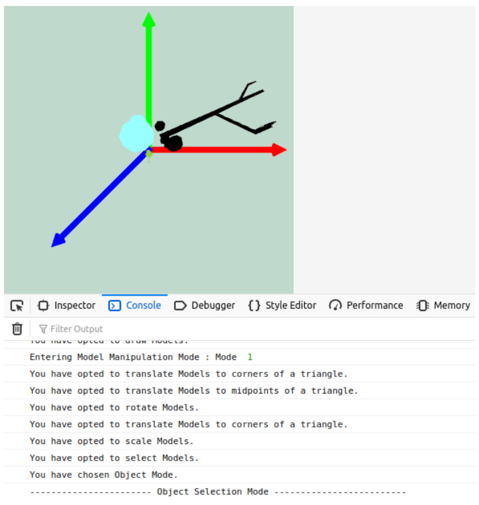


```
You can press Esc at anytime to quit the application.
You have opted to draw Axes.
You have opted to draw Models.
Entering Model Manipulation Mode : Mode  1
You have opted to translate Models to corners of a triangle.
You have opted to translate Models to midpoints of a triangle.
You have opted to rotate Models.
You have opted to translate Models to corners of a triangle.
You have opted to scale Models.
```

Figure 6. Scaled Models

2.4.2. Model Picking

Model Picking can be initiated by pressing 'H'. User must then press 'o' to enter the object picking mode. Picking has been implemented by assigning a unique ID to models based on their color. The pixel color at the mouse position is then captured on clicking and compared to this ID. If it is a match, the model is picked and this action has been visualized by changing the color of the model to black when picked. Only object picking mode has been implemented in this application. Reason being that the entire model has been stored as a single file. Note, although a single model can consist of multiple objects (see Autumn model : the branched tree and boulders), they have all been unioned together in the end to be compiled into a single model. This was done for the better usage of the "decimate" feature of Blender to bring down the computational power that would be required to render the models on screen. As a result of this, the multiple "faces" of a model all belong to a single file of the model with uniform color and hence there's no distinguishing between them. A sample of the picked model is shown below 7.

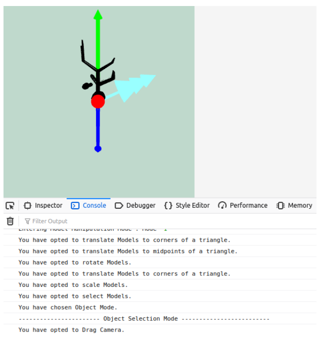


```
Entering Model Manipulation Mode : Mode  1
You have opted to translate Models to corners of a triangle.
You have opted to translate Models to midpoints of a triangle.
You have opted to rotate Models.
You have opted to translate Models to corners of a triangle.
You have opted to scale Models.
You have opted to select Models.
You have chosen Object Mode.
---------------------- Object Selection Mode ------------------------
```

Figure 7. Autumn Model is picked and highlighted.

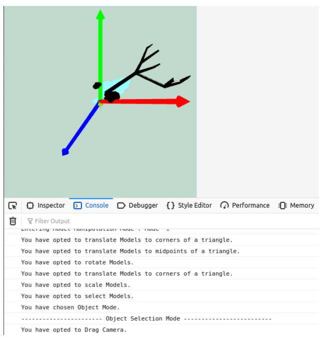
2.4.3. Camera Rotation

Camera Rotation can be initiated by pressing 'I' on the keyboard. On click the camera is shifted to an initial position of rotation, where the user is now viewing the scene from the X-axis. The user can then drag the mouse towards the right for a counter-clockwise rotation and towards the left for a clockwise rotation. The rotation has been implemented about the Y-axis. Y-axis was chosen because of the way the Z-axis has been represented in the application. For better visualization purposes of the previous modes, the Z-axis has been drawn in the X-Y plane. It is not the accurate mathematical representation of the Z-axis (this has been commented out in the code if one wishes to try). As a result of this, along the other directions, the rotation of camera gives a very distorted meaning to the scene and thus the Y-axis was chosen. Another thing to point out, Depth Testing has only been enabled in this mode. In all the previous modes, it has been intentionally disabled. This is because one of the coordinates of the triangle is of the form (0, y, 0). As a result of this, the Y-axis and the model are in the same plane and same depth. This again gives a very distorted view if one were to run the application the very first time without enabling rotations. So, to account for this, only in the Camera Rotation mode has the depth testing been enabled. The camera can be rotated to any point in the scene and all subsequent transformations, additions and deletions respect this position. The initial and rotated camera positions are shown in 8, 9.



```
You have opted to translate Models to corners of a triangle.
You have opted to translate Models to midpoints of a triangle.
You have opted to rotate Models.
You have opted to translate Models to corners of a triangle.
You have opted to scale Models.
You have opted to select Models.
You have chosen Object Mode.
---------------------- Object Selection Mode ------------------------
You have opted to Drag Camera.
```

Figure 8. Initial Camera position

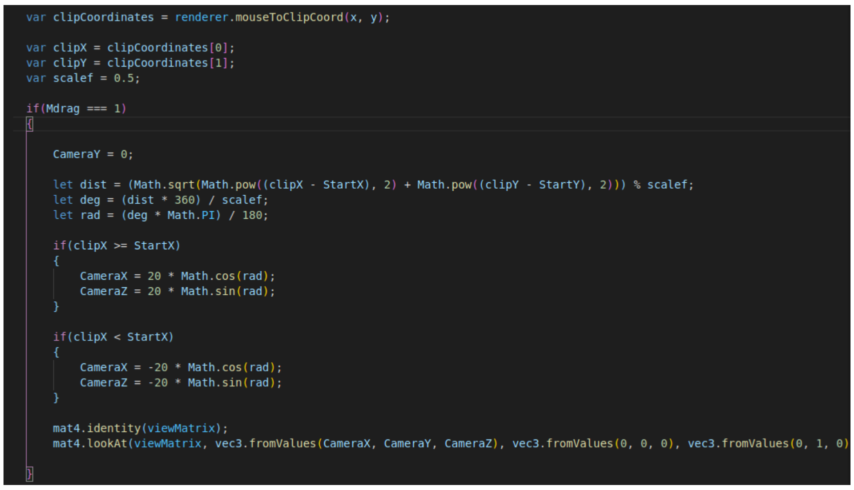


```
You have opted to translate Models to corners of a triangle.
You have opted to translate Models to midpoints of a triangle.
You have opted to rotate Models.
You have opted to translate Models to corners of a triangle.
You have opted to scale Models.
You have opted to select Models.
You have chosen Object Mode.
---------------------- Object Selection Mode ------------------------
You have opted to Drag Camera.
```

Figure 9. Rotated Camera

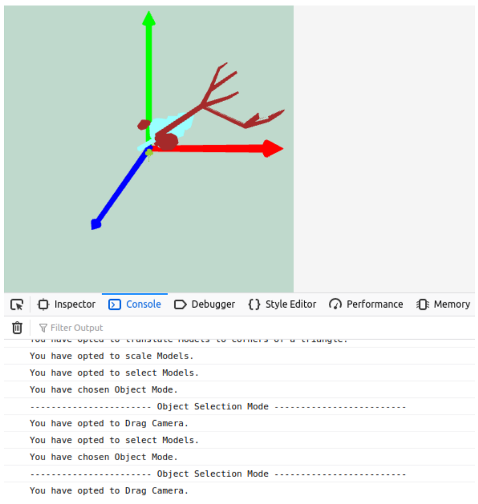
Mathematically, the rotation has been implemented by recomputing the View Matrix using the lookAt() function again and again with respect to the new position of the camera during the rotation. Let us first see, how the angle by which to be rotated is computed. We first choose a scale, which is 0.5 for the submitted implementation. The scale represents the distance by which you would have to drag the mouse to complete one 360 degree rotation. All intermediate degrees are uniformly divided within the scale. Thus, we first compute the distance between the current coordinates in the drag position and the start of the drag motion. This distance is then converted to the degrees (θ) to be rotated. This in turn is converted into radians and the new X, Z coordinates of the camera are then computed as $Rcos(\theta)$ and $Rsin(\theta)$, where *R* is the initial distance from the origin. Refer to the code snippet 10 for further clarity.

A few samples of manipulations after rotation have been shown in 11, 12, 13.



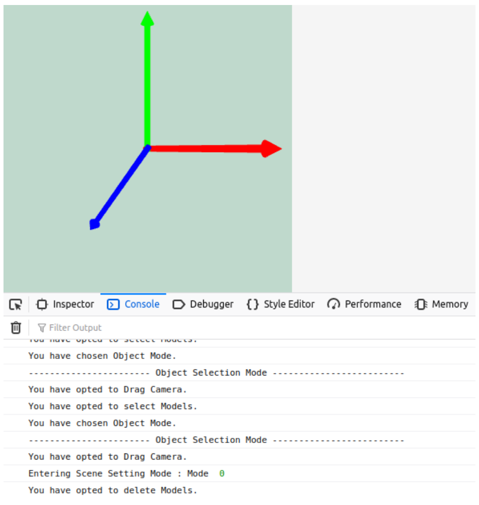
```
var clipCoordinates = renderer.mouseToClipCoord(x, y);

var clipX = clipCoordinates[0];
var clipY = clipCoordinates[1];
var scalef = 0.5;

if(Mdrag === 1)
{

    CameraY = 0;

    let dist = (Math.sqrt(Math.pow((clipX - StartX), 2) + Math.pow((clipY - StartY), 2))) % scalef;
    let deg = (dist * 360) / scalef;
    let rad = (deg * Math.PI) / 180;

    if(clipX >= StartX)
    {
        CameraX = 20 * Math.cos(rad);
        CameraZ = 20 * Math.sin(rad);
    }

    if(clipX < StartX)
    {
        CameraX = -20 * Math.cos(rad);
        CameraZ = -20 * Math.sin(rad);
    }

    mat4.identity(viewMatrix);
    mat4.lookAt(viewMatrix, vec3.fromValues(CameraX, CameraY, CameraZ), vec3.fromValues(0, 0, 0), vec3.fromValues(0, 1, 0)

}
```

Figure 10. Code Snippet for Rotation about Y-Axis



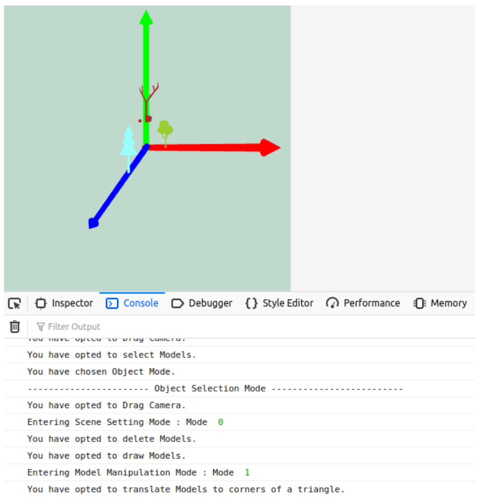
```
You have opted to translate models to corners of a triangle.
You have opted to scale Models.
You have opted to select Models.
You have chosen Object Mode.
---------------------- Object Selection Mode ------------------------
You have opted to Drag Camera.
You have opted to select Models.
You have chosen Object Mode.
---------------------- Object Selection Mode ------------------------
You have opted to Drag Camera.
```

Figure 11. Autumn is Deselected in Object Mode



```
You have chosen Object Mode.
---------------------- Object Selection Mode ------------------------
You have opted to Drag Camera.
You have opted to select Models.
You have chosen Object Mode.
---------------------- Object Selection Mode ------------------------
You have opted to Drag Camera.
Entering Scene Setting Mode : Mode  0
You have opted to delete Models.
```

Figure 12. Models are deleted but Rotated Axes remain

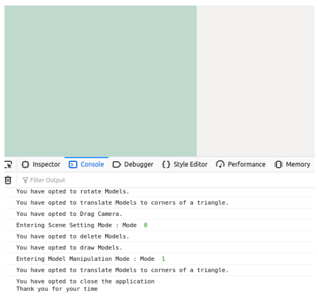


```
You have opted to select Models.
You have chosen Object Mode.
---------------------- Object Selection Mode ------------------------
You have opted to Drag Camera.
Entering Scene Setting Mode : Mode  0
You have opted to delete Models.
You have opted to draw Models.
Entering Model Manipulation Mode : Mode  1
You have opted to translate Models to corners of a triangle.
```

Figure 13. Models are added and translated according to Rotated Axes

2.5. Quitting the Application

A user can choose to quit the application at any time by pressing the ’Esc’ key. On doing this, the screen is rendered clear and an exit message is displayed on the browser console. A sample of this is shown as below 14.



```
You have opted to rotate Models.
You have opted to translate Models to corners of a triangle.
You have opted to Drag Camera.
Entering Scene Setting Mode : Mode  0
You have opted to delete Models.
You have opted to draw Models.
Entering Model Manipulation Mode : Mode  1
You have opted to translate Models to corners of a triangle.
You have opted to close the application
Thank you for your time
```

Figure 14. Quit Application

3. Experiments

A few experiments that were tried but may or may not have been included in the final code are,

- Rather than importing 3D Models, modelling in Blender was preferred .
- Model manipulations with Axes drawn such that the Z-axis now corresponds to a mathematically accurate representation.
- Camera Rotations about the X and Z axis.
- Allowing the user to reconstruct specific scene objects after deleting them.

4. Questions

The following section addresses the questions posted in the problem statement.

(1) To what extent were you able to reuse code from Assignment 1?
Answer. I would say that I was able to utilize about 60% of the previous code. In the first assignment itself I had abstracted out the concept of each model having it's own Model Matrix and a World Matrix in the vertex shader and conceptually the implementations were done accordingly. Thus, adding in a new View Matrix and Projection Matrix were easy to figure out. However, upon further reading, it was discovered that both : the World and Model Matrix are somewhat synonymous terms and the entirety of these transformations were then recomputed into a single representation, the WorldTransform Matrix, which gives out the position of an object in the world space. The new implementations required additional code.

(2) What were the primary changes in the use of WebGL in moving from 2D to 3D?
Answer. As mentioned in the previous question, adding in the View and Projection Matrix were the crux of going from 2D to 3D. Another change required was figuring out the use of webgl model loaders to import in the created 3D models.

(3) How were the translate, scale and rotate matrices arranged such that rotations and scaling are independent of the position or orientation of the object?
Answer. The computation of the WorldTransformation Matrix for each model also required each model to maintain separate matrices for the three different transformations; Scale, Rotate and Translate. With each transformation requirement, updates were made to each of the above three matrices accordingly. Finally, when the object is about to be rendered we compute the final WorldTransformation Matrix as a product of the above three matrices. The order of multiplication followed is : Scale, Rotate and finally Translate.

5. Conclusion

In conclusion, the report has successfully elaborated on all the required deliverables except face picking, the reason for which has been discussed above. The methodology section briefly discusses the approaches behind the implementation of the application and also displays sample images for the same. All the questions entailed in the problem statement have also been addressed. The trials and errors that were encountered have also been briefly discussed in the experiments section.