Computer Graphics Assignment 2

Naveen Kumar V R
IMT2017029
International Institute of Information Technology
Email: naveenkumar.vr@iiitb.org

1. Goal

1) Create 3D models (using a modelling tool)
2) Importing 3D mesh models
3) Transformations of 3D objects
4) View Transformations
5) Picking model objects and their constituent parts in 3D

2. Part 1

Creating a set of 3D solid models

2.1. Models made using Blender

- Created a alien
- Created a cube cone
- Created a Shuriken

3. Part 2

3.1. Step A

- The axis model is imported and it is rendered and duplicated to two more. These are made to world coordinates and transformed these axis accordingly.
- x, y, z axis are colored with Red, Green and Blue respectively.

3.2. Step B

- By using the available parser we convert .obj to an object containing model information.
- We import these models after parsing.
- Models are imported before step as the axis model should be imported to create the scene.

3.3. Step C

The imported models are rendered at the origin.

3.4. Step D

These three models are now translated to vertices (0,0.5,0),(0.5,-0.5,0),(-0.5,-0.5,0) which form a triangle centered at origin.

3.5. Step E

The three models are now translated to midpoint of the sides of the triangle created in step D.

3.6. Step F

The three models are now rotated about x,y and z-axis respectively by 90 degrees.

3.7. Step G

- The three models are now scaled by 0.5x, 2x, 3x respectively.
- This scaling can be undone by hitting 'r' button.

3.8. Step H

There are two modes in this step

3.8.1. Object Mode.

- In this mode we can click on any object.

3.8.2. Face Mode.

- In this mode we can click on any face of an object.

3.9. Step I

- This is Camera Mode.
- We can select which axis we want to rotate about by pressing 'x','y' or 'z'.
- In this mode we can drag our mouse and the scene is rotated about the selected axis.

4. Key Concepts

4.1. Picking Object

Picking is implemented by detecting the color at the location of mouse click then using this color to compare with the colors of all faces and objects. If the color matches then that object or face is selected.

4.2. Camera

Camera is the point from where the objects are perceived. So moving the camera will move the scene. While implementing the camera actually doesn't move but the scene is transformed accordingly. First the transformation matrix is calculated for camera and then its inverse is applied on the scene.

5. Questions

5.1. To what extent were you able to reuse code from Assignment 1?

The transformation code used to compute transformations is the same in both assignments so, I could have done it that way. But this time I used a different file for transformations as it had more helper functions but the code logic is the same. For rotating and object created of different objects(humanoid) I used the scene trans- formation code with a little tweak. The keybindings use the same web API used in assignment 1.

5.2. What were the primary changes in the use of WebGL in moving from 2D to 3D?

Firstly z coordinates are non zero now. To make the view feel like 3d there was a need to add perspective view.

5.3. How were the translate, scale and rotate matrices arranged such that rotations and scaling are independent of the position or orienta- tion of the object?

Similar to 2D all the models are created at origin. Then the transformation matrix is calculated. Only the transformation matrix changes when rotated or scaled or translated so original object is always origin. This way rotations and scaling are independent of the position or orientation of the object.