

Programming Assignment-2 Report

Abhinav Kamath(IMT2019001)

• Summary

In this programming assignment, I have implemented an application which allows the users to move around and animate 3D objects.

- In mode 0, no change can be made
- The shapes appear in 3D mode along with the axes.
- User can toggle to top-view whenever necessary
- Each object can be selected by clicking on it, this is indicated by the object color changing to black
- Selected object can be scaled up/down in both modes
- Selected object can be rotated around x or y or z axis in both modes
- Selected object can be made to move on a quadratic curve during the animation phase
- In the animation phase coordinates for curve can be picked by clicking on the screen on 2 points in topView mode.
- The speed of animation can be increased or decreased, animation can be viewed in both modes.
- The camera can be rotated around any one of the axes at a given time in 3D mode

• Key bindings

- i -> init()
- t -> toggle between 3D view and topView
- a -> animate
- x -> rotate camera around X-axis
- y -> rotate camera around Y-axis
- z -> rotate camera around Z-axis
- X -> rotate object around X-axis
 - Use arrowleft and right to do the rotation
- Y -> rotate object around Y-axis
 - Use arrowleft and right to do the rotation
- Z -> rotate object around Z-axis
 - Use arrowleft and right to do the rotation
- + -> scale up
- - -> scale down
- ArrowUp -> speed up animation
- ArrowDown -> slow down animation

• Questions posed in the assignment

- **To what extent were you able to reuse code from Assignment 1**

Answer – All the transform functions like translate, rotate, scale remained same. Fragment shader remained the same. So did the renderer.js file.

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

Answer - The model view matrix goes from 3 x 3 to a 4 x 4 matrix . The z axis is no longer 0 but is given a value. Each vertex has 3 coordinates in place of 2 coordinates. The camera is an added functionality

- **How were the translate, scale and rotate matrices arranged? Can your implementation allow rotations and scaling during the animation?**

Answer – The matrices were arranged in the order translate, rotate and then scale. The implementation can allow rotations and scaling during animation with a few tweaks.

- **How did you choose a value for t_1 in computing the coefficients of the quadratic curve? How would you extend this to interpolating through n points ($n > 3$) and still obtaining a smooth curve?**

Answer - t_1 can be randomly chosen, I chose the value of t_1 to be 0.5.

For n points one way of doing is split the interval from $[0,1]$ into n equal parts and use each t to solve a n degree polynomial.

• Sources

- <https://javascript.info/keyboard-events> - For the keypress events.
- <https://github.com/Amit-Tomar/T2-21-CS-606/tree/main/src/example5> - Examples from the tutorial class
- <https://github.com/davidwparker/programmingtil-webgl>
- https://www.youtube.com/watch?v=oDiSqQT_szo&list=PLPqKsyEGhUnaOdIFL KvdkXAQWD4DoXnFI&ab_channel=DavidParker