**Project 8 Part 2 Rotating a platonic solid with OpenCV (perspective)**

Name: Satvik Matta Period: 2 Date: 5/23/2023

Did you name your file l082.cpp (Lower case L, then 082)? Yes

Does your file compile & run on terminals? Yes

Did you use a rotation matrix? Yes

Did you start from the coordinates I provided for the cube?? Yes

Does your code generate the coordinates.txt file? Yes

Does your code generate the coordinates2d.txt file? Yes

Does your code generate the rotation.avi file? Yes

Describe here in words all the transformations you applied to vertices, for each describe how you implemented it in your code (by multiplying with a matrix, what was the matrix, or by adding a matrix, what was that matrix… be specific):

Rotations: Multiplied using Matrix

Rotation Matrix 1:

double md[] = {1,0,0,0,0, cos(radians), sin(radians), 0, 0, -1\*sin(radians), cos(radians), 0, 0, 0, 0, 1};

Rotation Matrix 2:

double md2[] = {cos(radians), 0, -1\*sin(radians), 0, 0, 1, 0, 0, sin(radians), 0, cos(radians), 0, 0, 0, 0, 1};

Translation: Added to the point at time of rendering

Scaling: Done to the point at time of creation

Describe in words the rotation you did (be very specific and complete!!):

**Rotated the points around the x axis and the y axis by the same radians per frame.**

Did you use homogenous coordinates? Yes

(that allows you to combine all transformations into one matrix)

Did you combine all those transformations into one single matrix? No

If you used only one transformation matrix, what was it?

What functions/methods from OpenCV did you use?

**Line()**

**Circle()**

**.converto()**

**.write()**

**Point()**

What functions/methods from OpenCV did you experiment with but ended not using?

**cvtColor(image, image, COLOR\_BGR2RGB);**

Did you do a perspective rendering? Yes

What is the position of the eye you used? **(800,200,50)**

What is the plane of the screen you projected on? **(500,0,0)**

Did you name your video rotation.avi? Yes

What functions/methods from OpenCV did you use?

**Point()**

What functions/methods from OpenCV did you experiment with but ended not using?

**N/A**

Did you implement the following?

a) A rotation that goes out of the viewing screen: No

b) A rotation such that some of the vertices cross the viewing plane: No

Obs.: feel free to rotate any platonic solid, around any line, and you may put the position of the screen/viewing window in any place as long as the rotating platonic solid can be seen reasonably.