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

Name: Randy Fu Period: 5 Date: 4/28

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):

I applied rotations is all 3 axis. The x and y axis have a rotation of 0.05 degrees every frame while the z-axis has a rotation of .0000005 degrees every frame. I also did a scaling matrix with a value of 100 so the cube is scaled by 100.

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

I applied rotations is all 3 axis. The x and y axis have a rotation of 0.05 degrees every frame while the z-axis has a rotation of .00005 degrees every 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? Yes

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

A matrix of all three axis rotations

What functions/methods from OpenCV did you use?

Mat

VideoWriter

VideoWriter.write

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

Did you do a perspective rendering? Yes

What is the position of the eye you used? (500,0,0)

What is the plane of the screen you projected on? (400,0,0)

Did you name your video rotation.avi? Yes

What functions/methods from OpenCV did you use?

Circle

Point

Scalar

FILLED

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

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.