Divyansh Agrawal

Concepts used

1. When importing the “object.txt” file using pandas and converting it into a numpy array, split method was used since the array had strings in it. So the string was split depending on the ‘,’ position and then the coordinates were extracted.

2. Transformation matrices:
   For rotating the object upon mouse movements, I used the concept of transformation matrices.
   New vertices will be the product of the initial coordinates matrix and the rotation matrix. Since the rotation can be along any axis, the rotation matrix depends on the axis. The image below shows the rotation matrices along all the 3 axes. ‘a’ is the angle of rotation.

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(a) & -sin(a) \\ 0 & sin(a) & cos(a) \end{bmatrix} R_y = \begin{bmatrix} cos(a) & 0 & sin(a) \\ 0 & 1 & 0 \\ -sin(a) & 0 & cos(a) \end{bmatrix} R_z = \begin{bmatrix} cos(a) & -sin(a) & 0 \\ sin(a) & cos(a) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3. Translation of points:
   The input coordinates are with respect to the shape. The canvas in tkinter has its origin at the top left, so the coordinates were translated to draw the shape in the center of canvas.

4. Changing the color according to the viewing angle:
   When a face is parallel to the z axis (coming out of the screen), it should have the color #00005F (say dark blue) and when a face is parallel to the z axis, it should have a color #0000FF (say light blue). For doing this, following logic was used:
   - A face will have 3 edges, so take a cross product of any 2 edges. This cross product (say vector **C**) will be normal to the face. If **C** is parallel to the Z axis then color the face dark blue. If **C** is perpendicular to the Z axis then color that face light blue. So the face color will change according to the value of the cross product.