Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

Lab 5 Report

Computer Graphics (COMP 342)

Submitted to:

Mr. Dhiraj Shrestha

Department of Computer Science and Engineering

Submitted By:

Mani Dumaru

Roll no.: 15

3rd Year/ 2nd Semester

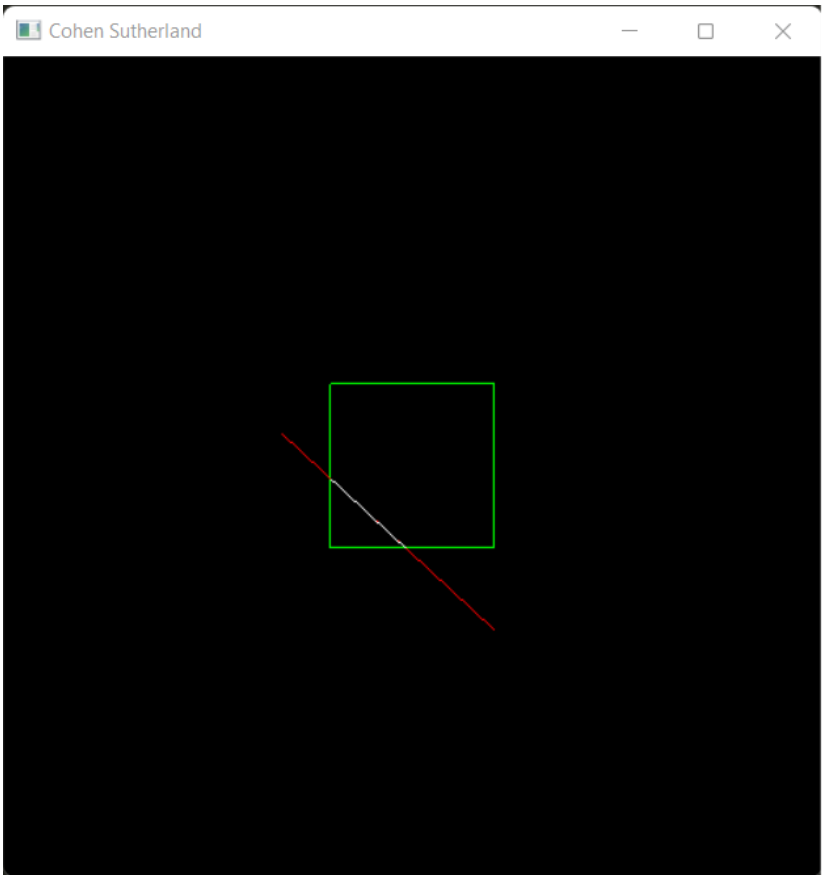
Submission Date: 28th May, 2023

Cohen Sutherland Line clipping:

Algorithm:

1. Assign a region code for each endpoint.
2. If both endpoints have a region code 0000 trivially accept this line.
3. Else, perform the logical AND operation for both region codes.
   3.1 If the result is NOT 0000, trivially reject the line.
   3.2 Else (i.e., result = 0000, needs clipping)
      3.2.1 Choose an endpoint of the line that is outside the window.
      3.2.2 Find the intersection point at the window boundary
      3.2.3 Replace endpoint with the intersection point and update the region code.
      3.2.4 Repeat step 2 until we find a clipped line either trivially accepted or rejected.

Output:

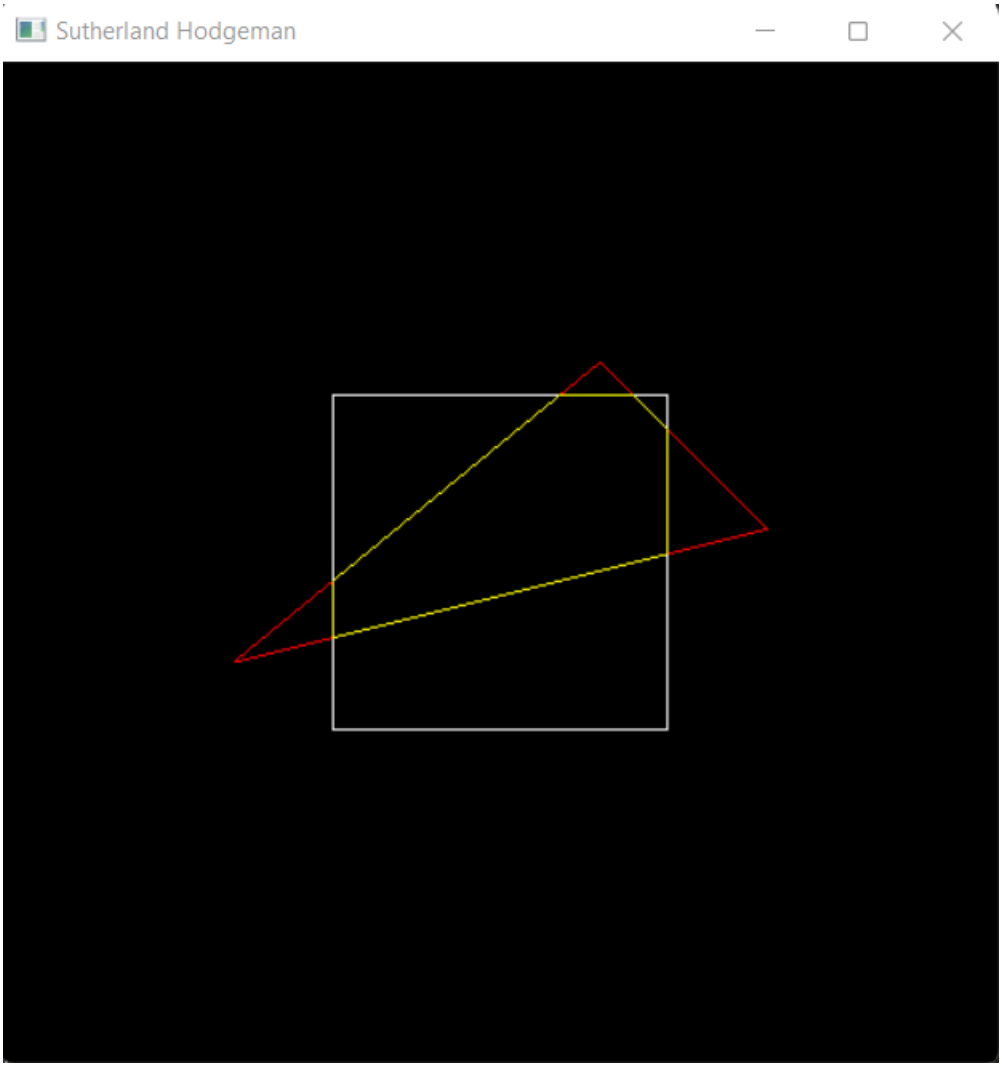


Sutherland Hodgeman Polygon Clipping:

Traverse the edges for four borders (Top, Right, Bottom, Left)

Find the intersection points between clipping window and the edges of the polygon.

Output:



3D Transformations:

Transformation matrices for the following 3D transformations are:

3D translation:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & tx \\ 0 & 1 & 0 & ty \\ 0 & 0 & 1 & tz \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Fixed point 3D Scaling:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} Sx & 0 & 0 & (1-sx)xf \\ 0 & Sy & 0 & (1-sy)yf \\ 0 & 0 & Sz & (1-sz)zf \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

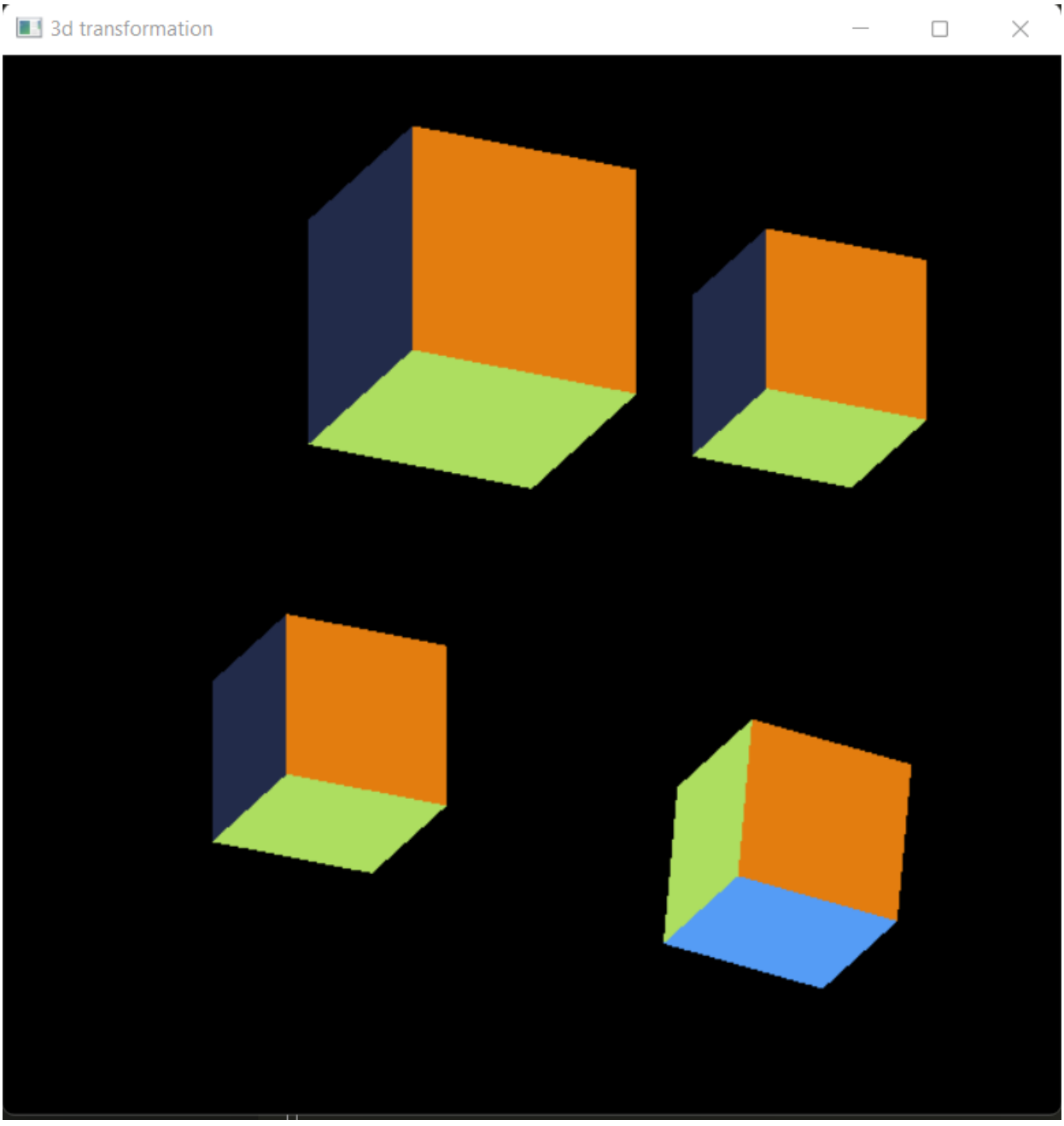
3D Rotation:

x-axis: $\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\Theta & -\sin\Theta & 0 \\ 0 & \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$

y-axis: $\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\Theta & 0 & \sin\Theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\Theta & 0 & \cos\Theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$

z-axis: $\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 & 0 \\ \sin\Theta & \cos\Theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$

Output:



Conclusion:

In this way, Cohen Sutherland Line clipping algorithm, Sutherland Hodgeman Polygon Clipping algorithm and various 3D transformations were implemented in PyOpenGL.