Batch: D-2 **Roll No.:** 16010122151

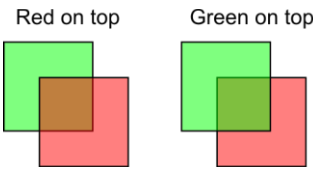
Experiment No. 09

TITLE: Write a program to Implement Transparency

AIM:

Write a program in OpenGL to Implement Transparency

Sample example



Expected OUTCOME of Experiment:

CO3: Implement Clipping, 3D Geometric Transformations and 3D viewing.

Books/ Journals/ Websites referred:

http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-10-transparency/

Algorithm/ Pseudocode for each process:

Blending in OpenGL is commonly known as the technique to implement transparency within objects. Transparency is all about objects (or parts of them) not having a solid colour, but having a combination of colours from the object itself and any other object behind it with varying intensity.

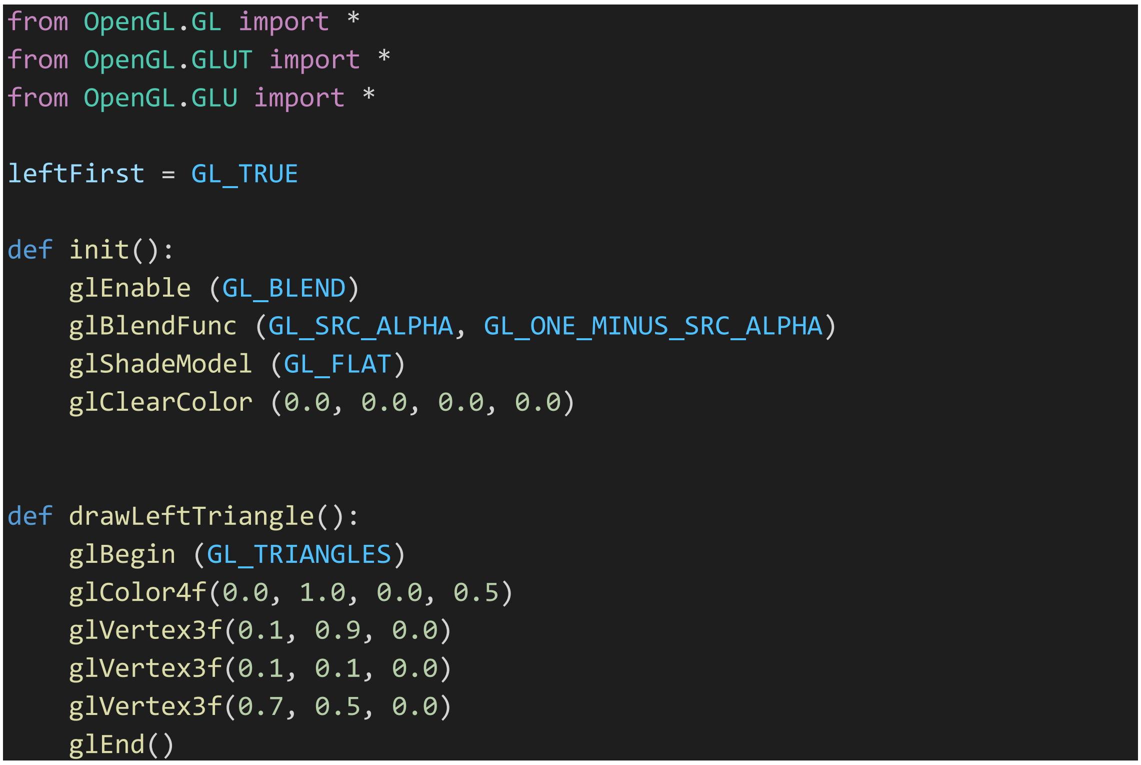
Implementation details:

In order for the previous code to work, you need to setup your blend function.

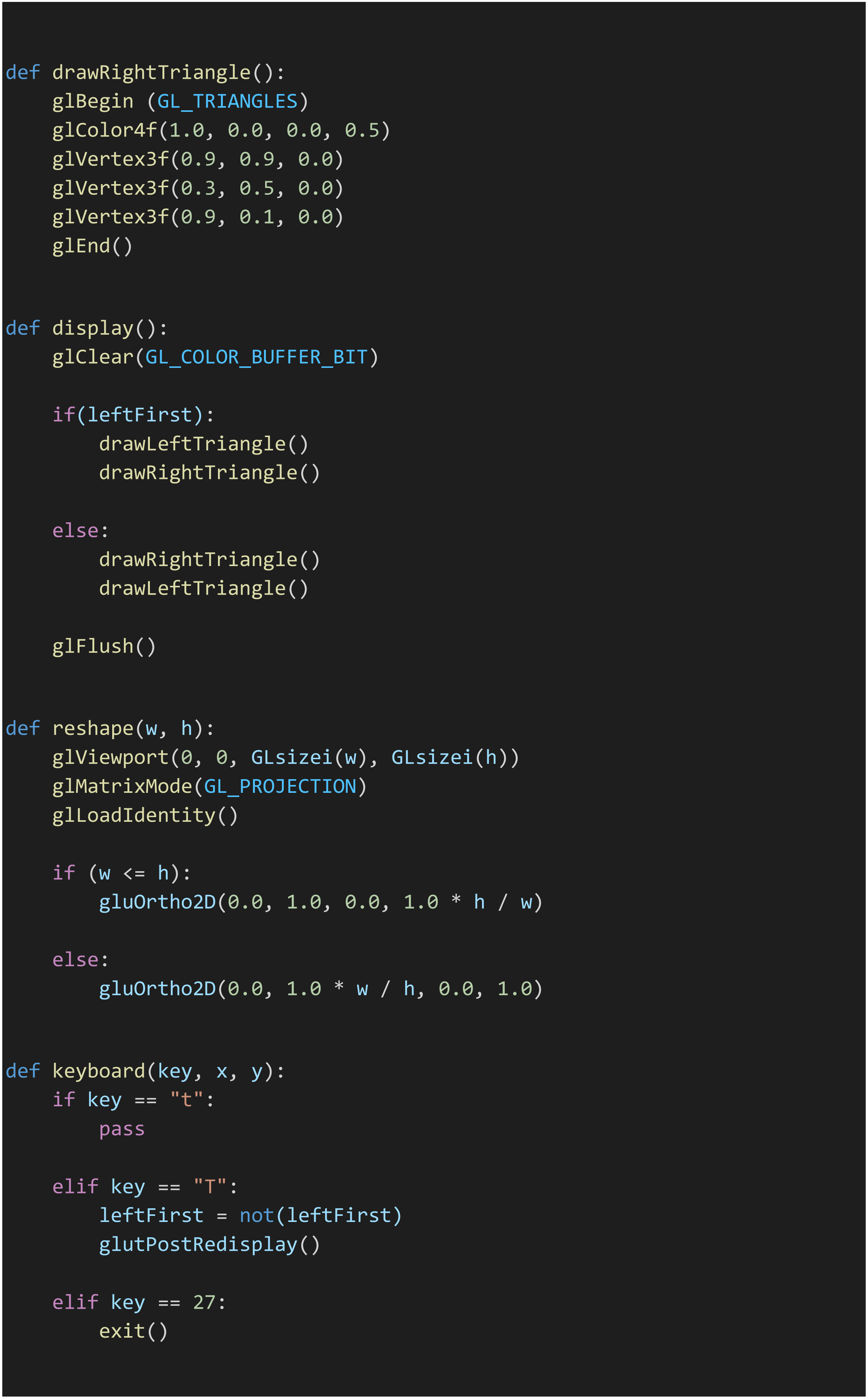
```
// Enable blending
glEnable(GL_BLEND);
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
```

Which means,
New color in framebuffer =
current alpha in framebuffer * current color in framebuffer + (1 - current alpha in framebuffer) * shader's output color

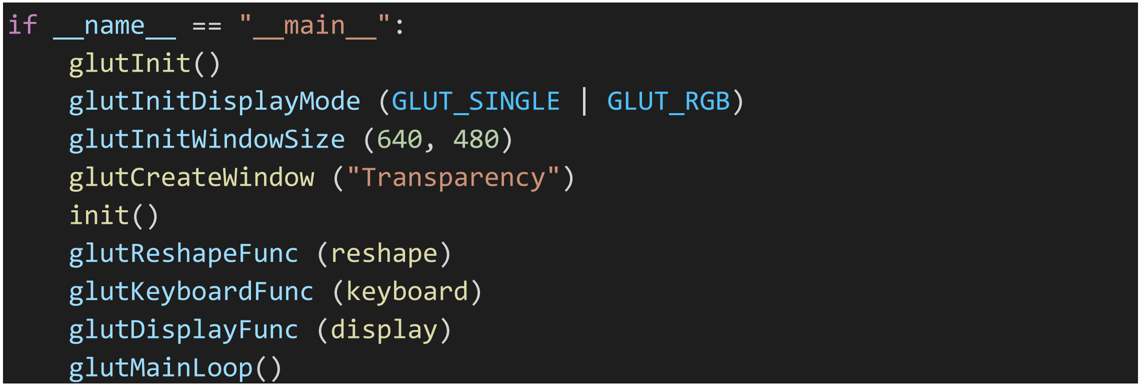
Example from the image above, with red on top
new color = 0.5*(0,1,0) + (1-0.5)*(1,0.5,0.5); // (the red was already blended with the white background)
new color = (1, 0.75, 0.25) = the same orange



```
from OpenGL.GL import *
from OpenGL.GLUT import *
from OpenGL.GLU import *

leftFirst = GL_TRUE

def init():
    glEnable (GL_BLEND)
    glBlendFunc (GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)
    glShadeModel (GL_FLAT)
    glClearColor (0.0, 0.0, 0.0, 0.0)


def drawLeftTriangle():
    glBegin (GL_TRIANGLES)
    glColor4f(0.0, 1.0, 0.0, 0.5)
    glVertex3f(0.1, 0.9, 0.0)
    glVertex3f(0.1, 0.1, 0.0)
    glVertex3f(0.7, 0.5, 0.0)
    glEnd()
```

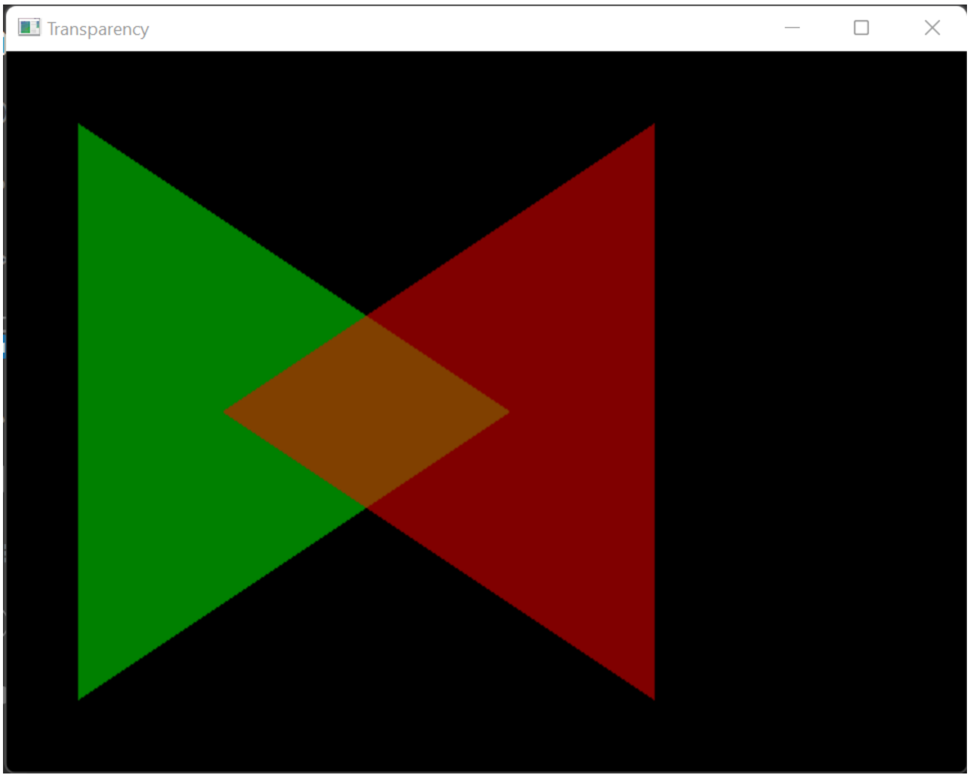


```
def drawRightTriangle():
    glBegin (GL_TRIANGLES)
    glColor4f(1.0, 0.0, 0.0, 0.5)
    glVertex3f(0.9, 0.9, 0.0)
    glVertex3f(0.3, 0.5, 0.0)
    glVertex3f(0.9, 0.1, 0.0)
    glEnd()


def display():
    glClear(GL_COLOR_BUFFER_BIT)

    if(leftFirst):
        drawLeftTriangle()
        drawRightTriangle()

    else:
        drawRightTriangle()
        drawLeftTriangle()

    glFlush()


def reshape(w, h):
    glViewport(0, 0, GLsizei(w), GLsizei(h))
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()

    if (w <= h):
        gluOrtho2D(0.0, 1.0, 0.0, 1.0 * h / w)

    else:
        gluOrtho2D(0.0, 1.0 * w / h, 0.0, 1.0)


def keyboard(key, x, y):
    if key == "t":
        pass

    elif key == "T":
        leftFirst = not(leftFirst)
        glutPostRedisplay()

    elif key == 27:
        exit()
```



```
if __name__ == "__main__":
    glutInit()
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB)
    glutInitWindowSize (640, 480)
    glutCreateWindow ("Transparency")
    init()
    glutReshapeFunc (reshape)
    glutKeyboardFunc (keyboard)
    glutDisplayFunc (display)
    glutMainLoop()
```

Output(s) (Screen Shot):



Conclusion and discussion:

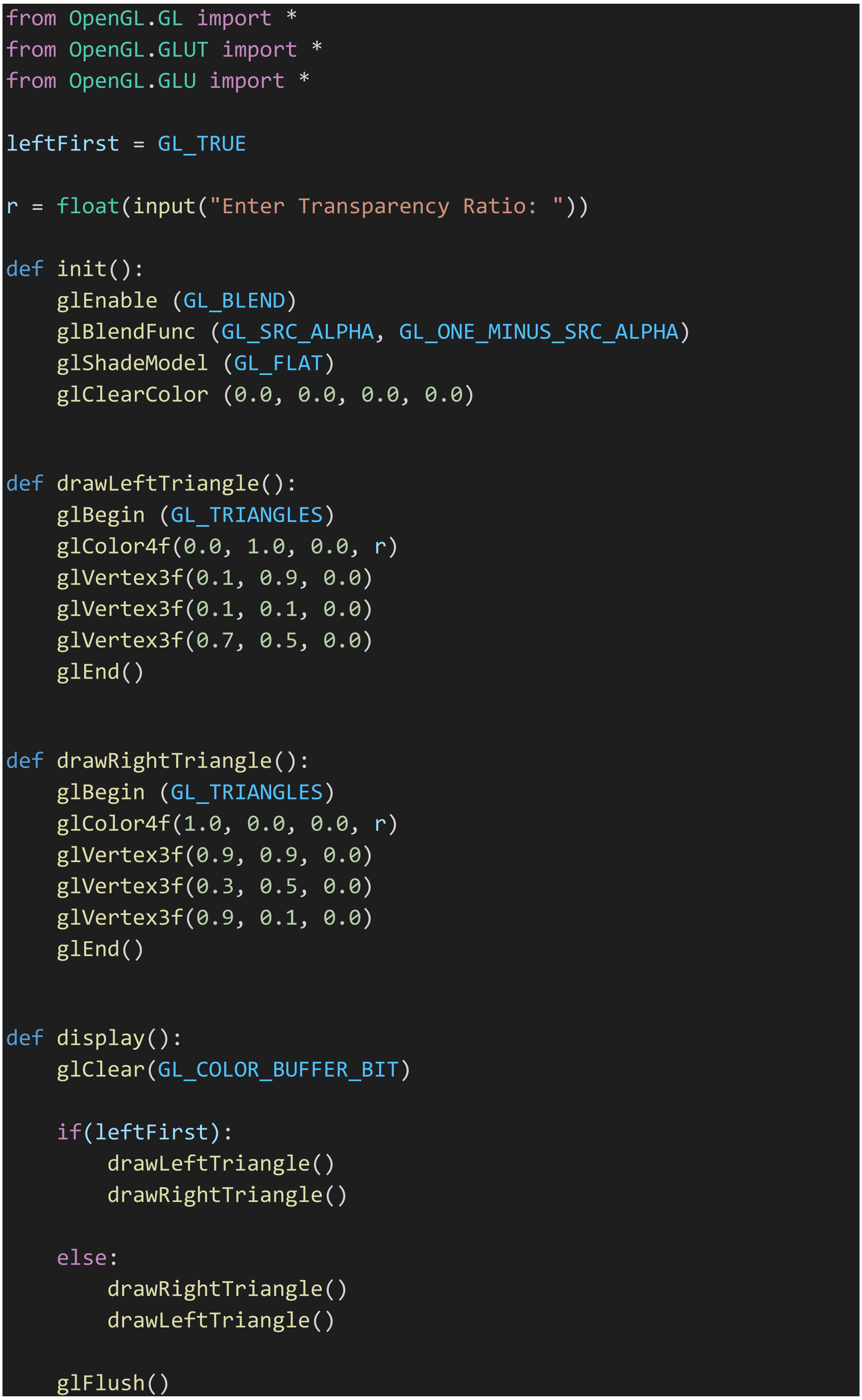
Through this experiment, we successfully learned about and implemented transparency using OpenGL in Python.
We have created two overlapping triangles and kept the transparency ratio as 0.5.

Date:

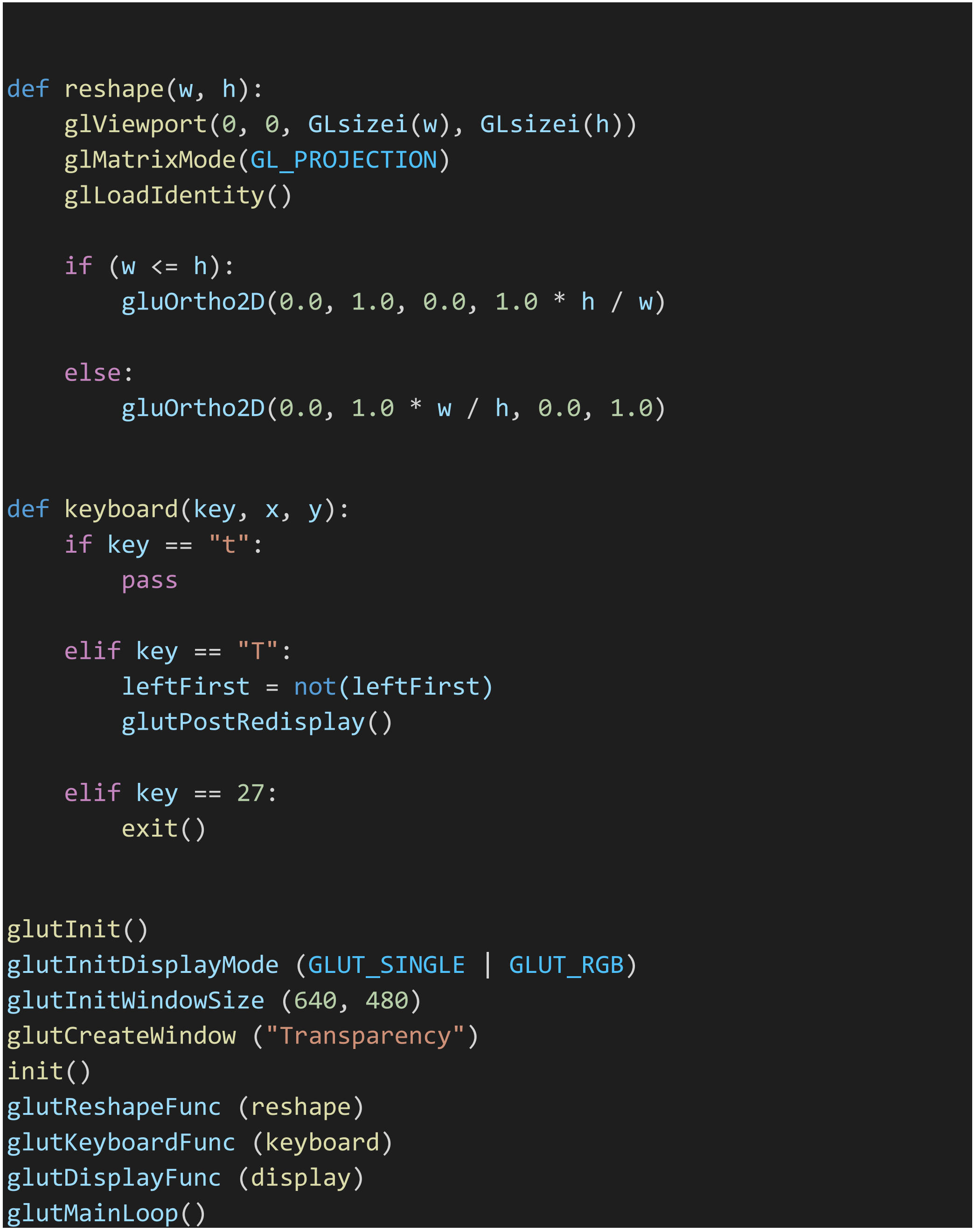
Signature of faculty in-charge

Post lab Question

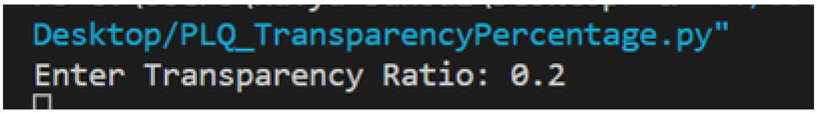
Improvise the code and take user input for transparency percentage (Make it interactive with user)



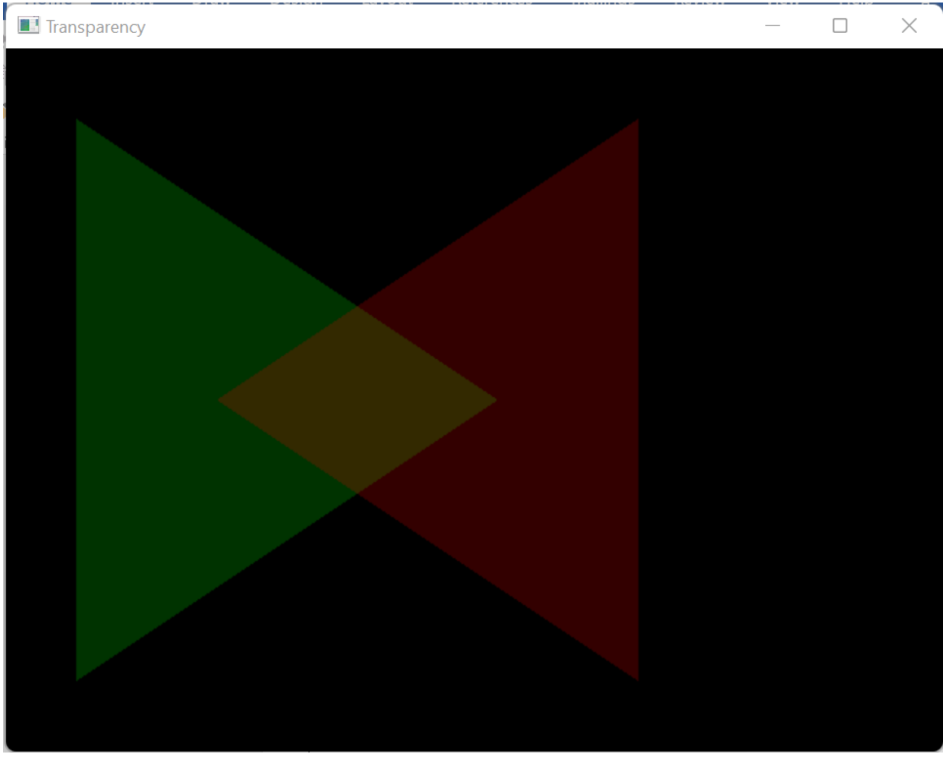
```
from OpenGL.GL import *
from OpenGL.GLUT import *
from OpenGL.GLU import *

leftFirst = GL_TRUE

r = float(input("Enter Transparency Ratio: "))

def init():
    glEnable (GL_BLEND)
    glBlendFunc (GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)
    glShadeModel (GL_FLAT)
    glClearColor (0.0, 0.0, 0.0, 0.0)


def drawLeftTriangle():
    glBegin (GL_TRIANGLES)
    glColor4f(0.0, 1.0, 0.0, r)
    glVertex3f(0.1, 0.9, 0.0)
    glVertex3f(0.1, 0.1, 0.0)
    glVertex3f(0.7, 0.5, 0.0)
    glEnd()


def drawRightTriangle():
    glBegin (GL_TRIANGLES)
    glColor4f(1.0, 0.0, 0.0, r)
    glVertex3f(0.9, 0.9, 0.0)
    glVertex3f(0.3, 0.5, 0.0)
    glVertex3f(0.9, 0.1, 0.0)
    glEnd()


def display():
    glClear(GL_COLOR_BUFFER_BIT)

    if(leftFirst):
        drawLeftTriangle()
        drawRightTriangle()

    else:
        drawRightTriangle()
        drawLeftTriangle()

    glFlush()
```



```
def reshape(w, h):
    glViewport(0, 0, GLsizei(w), GLsizei(h))
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()

    if (w <= h):
        gluOrtho2D(0.0, 1.0, 0.0, 1.0 * h / w)

    else:
        gluOrtho2D(0.0, 1.0 * w / h, 0.0, 1.0)


def keyboard(key, x, y):
    if key == "t":
        pass

    elif key == "T":
        leftFirst = not(leftFirst)
        glutPostRedisplay()

    elif key == 27:
        exit()


glutInit()
glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB)
glutInitWindowSize (640, 480)
glutCreateWindow ("Transparency")
init()
glutReshapeFunc (reshape)
glutKeyboardFunc (keyboard)
glutDisplayFunc (display)
glutMainLoop()
```



```
Desktop/PLQ_TransparencyPercentage.py"
Enter Transparency Ratio: 0.2
```


Transparency