Batch: C2 Roll No.: 110

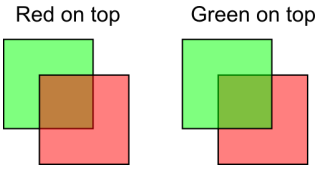
Experiment No. 08

TITLE: Write a program to Implement Transparency

AIM:

Write a program in OpenGL to Implement Transparency

Sample example



Expected OUTCOME of Experiment:

CO3

understanding of opengl and transparency

Books/ Journals/ Websites referred:

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

Algorithm/ Pseudocode for each process:

1) Set image values rgb
2) Set transparency alpha value
3) Calculate sum of all aphs values to find the normalized value of alpha
4) render the objects.

Implementation details:

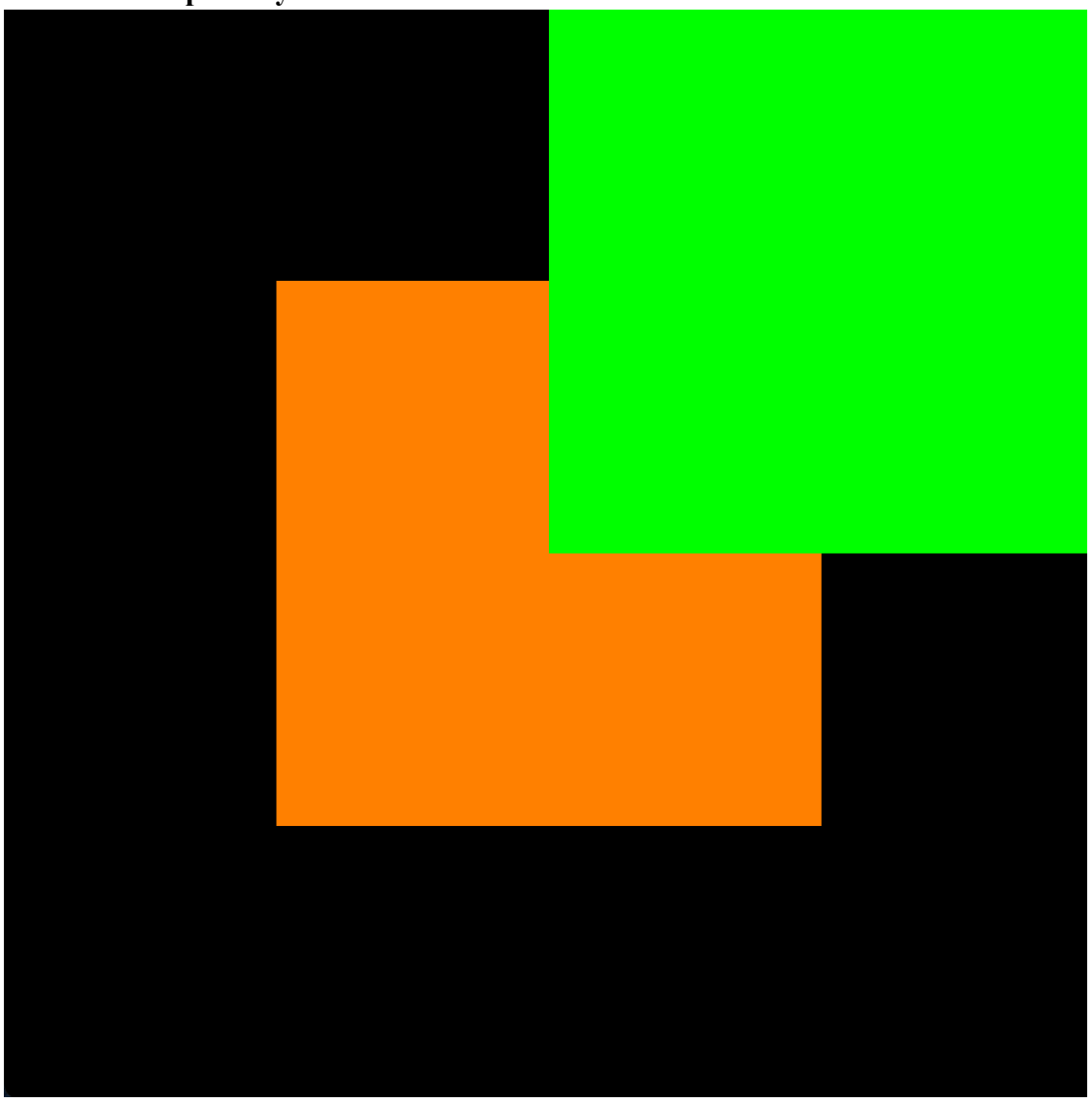
```
from OpenGL.GL import *
from OpenGL.GLUT import *
from OpenGL.GLU import *
from OpenGL.GL import *
from OpenGL.GLUT import *
```

```
# Define vertices and colors for the squares
square1_vertices = [
(-0.5, -0.5),
(-0.5, 0.5),
(0.5, 0.5),
(0.5, -0.5)
]

square2_vertices = [
(0, 0),
(0, 1),
(1, 1),
(1, 0)
]

colors = [
(1, 0.5, 0,0.5), # Orange
(0, 1, 0,0.2) # Green
]

def draw_square(vertices, color):
glColor4fv(color)
for vertex in vertices:
glVertex2fv(vertex)

def display():
glClear(GL_COLOR_BUFFER_BIT)
glColor3f(1.0, 1.0, 1.0) # Set color to white (RGB
values: 1.0, 1.0, 1.0)
glEnable(GL_BLEND)
```

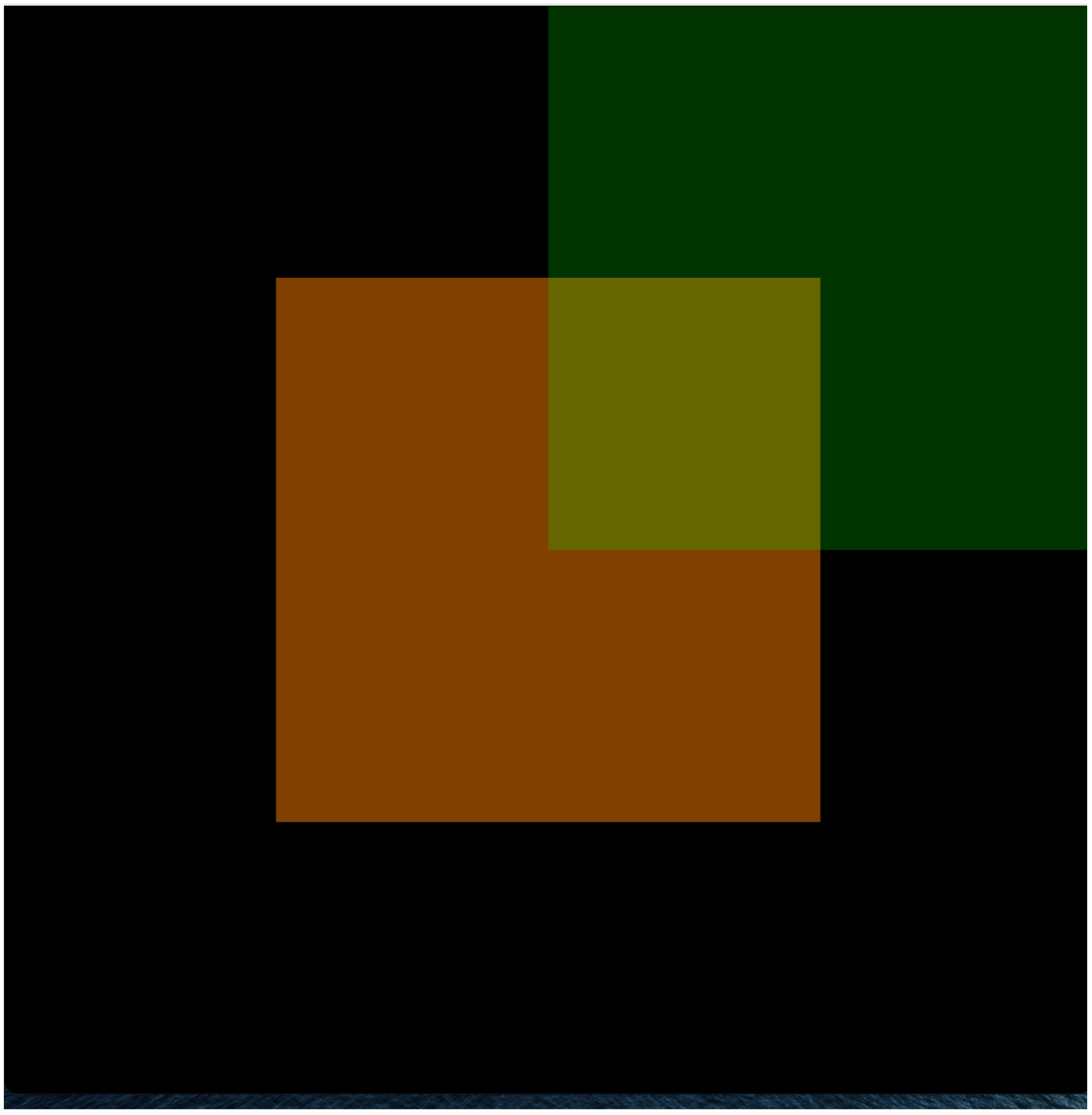
```
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)
glBegin(GL_QUADS)
# Enable blending for transparency
draw_square(square1_vertices,colors[0])
draw_square(square2_vertices,colors[1])


glEnd()
glFlush()



glutInit()
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB)
glutInitWindowSize(800, 800)
glutCreateWindow(b"PyOpenGL transparency Example")
glutDisplayFunc(display)
glutMainLoop()
```

Output(s) (Screen Shot):

Without transparency



With transparency



Conclusion and discussion:

Thus we have understood how transparency works and we have implemented transparency in openGL.

Date: 28 sept 23

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 *
from OpenGL.GL import *
from OpenGL.GLUT import *

# Define vertices and colors for the squares
square1_vertices = [
(-0.5, -0.5),
(-0.5, 0.5),
(0.5, 0.5),
(0.5, -0.5)
]

square2_vertices = [
(0, 0),
(0, 1),
(1, 1),
(1, 0)
]

colors = [
(1, 0.5, 0,int(input("please enter orange transparency
1-10 "))/10), # Orange
(0, 1, 0,int(input("please enter green transparency
1-10 "))/10) # Green
```

```
]

def draw_square(vertices, color):
glColor4fv(color)
for vertex in vertices:
glVertex2fv(vertex)

def display():
glClear(GL_COLOR_BUFFER_BIT)
glColor3f(1.0, 1.0, 1.0) # Set color to white (RGB
values: 1.0, 1.0, 1.0)
glEnable(GL_BLEND)
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)
glBegin(GL_QUADS)
# Enable blending for transparency
draw_square(square1_vertices,colors[0])
draw_square(square2_vertices,colors[1])

glEnd()
glFlush()



glutInit()
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB)
glutInitWindowSize(800, 800)
glutCreateWindow(b"PyOpenGL transparency Example")
glutDisplayFunc(display)
glutMainLoop()
```

```
PROBLEMS 32  OUTPUT  PORTS  TERMINAL

/Users/aatmaj/.venv/bin/python "/Users/aatmaj/ComputerGraphics/Exp 7/transparency.py"
(.venv) aatmaj@Aatmajs-MacBook-Pro ComputerGraphics % /Users/aatmaj/.venv/bin/python "/Users/aatmaj/ComputerGraphics/Exp 7/transparency.py"
please enter orange transparency 1-10 7
please enter green transparency 1-10 4
```