**Batch: D - 1 Roll No.: 16010122096**

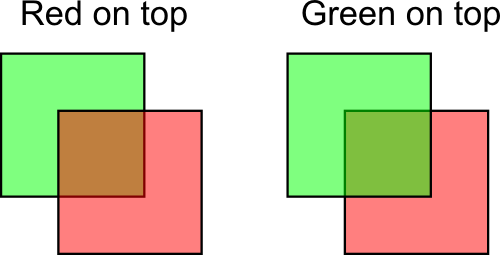
**Experiment No. 09**

|  |
| --- |
| **TITLE**: Write a program to Implement Transparency |

**AIM:**

Write a program in OpenGL to Implement Transparency

Sample example



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected OUTCOME of Experiment:**

CO4: Understand the computer Input & interaction, Curves and Computer Animation **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

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

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Algorithm/ Pseudocode for each process:**

1. Initialize:

a. Set the background color to white.

b. Enable blending to allow transparency.

c. Set the blending function to use source alpha and one minus source alpha.

2. Define Display Function:

a. Clear the color buffer.

b. Draw the green square:

i. Set color to green with alpha 0.5.

ii. Define vertices to form a square.

c. Draw the red square (on top of the green):

i. Set color to red with alpha 0.5.

ii. Define vertices to form a square.

d. Flush the output to display the results.

3. Main Function:

a. Initialize OpenGL.

b. Set display mode to single buffer with RGB.

c. Create a window with a given title.

d. Register the display function.

e. Enter the OpenGL main loop.

**Implementation details:**

#include <GL/glut.h>

void init() {

// Set clear color to white

glClearColor(1.0, 1.0, 1.0, 1.0);

// Enable blending

glEnable(GL\_BLEND);

glBlendFunc(GL\_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA);

}

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT);

// First set: Green on bottom, Red on top

// Draw green square

glColor4f(0.0f, 1.0f, 0.0f, 0.95f); // RGBA with 0.5 alpha for transparency

glBegin(GL\_POLYGON);

glVertex2f(-0.8f, -0.5f);

glVertex2f(-0.3f, -0.5f);

glVertex2f(-0.3f, 0.0f);

glVertex2f(-0.8f, 0.0f);

glEnd();

// Draw red square on top

glColor4f(1.0f, 0.0f, 0.0f, 0.4f); // RGBA with 0.5 alpha for transparency

glBegin(GL\_POLYGON);

glVertex2f(-0.55f, -0.25f);

glVertex2f(-0.05f, -0.25f);

glVertex2f(-0.05f, 0.25f);

glVertex2f(-0.55f, 0.25f);

glEnd();

// Second set: Red on bottom, Green on top

// Draw red square

glColor4f(0.0f, 1.0f, 0.0f, 0.25f); // RGBA with 0.5 alpha for transparency

glBegin(GL\_POLYGON);

glVertex2f(0.3f, -0.5f);

glVertex2f(0.8f, -0.5f);

glVertex2f(0.8f, 0.0f);

glVertex2f(0.3f, 0.0f);

glEnd();

// Draw green square on top

glColor4f(1.0f, 0.0f, 0.0f, 0.5f); // RGBA with 0.5 alpha for transparency

glBegin(GL\_POLYGON);

glVertex2f(0.55f, -0.25f);

glVertex2f(1.05f, -0.25f);

glVertex2f(1.05f, 0.25f);

glVertex2f(0.55f, 0.25f);

glEnd();

glFlush();

}

int main(int argc, char\*\* argv) {

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(500, 500);

glutInitWindowPosition(100, 100);

glutCreateWindow("Transparency Example - Two Sets of Squares");

init();

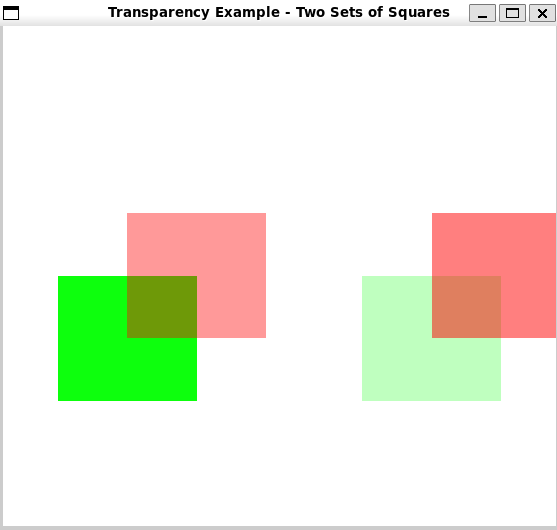
glutDisplayFunc(display);

glutMainLoop();

return 0;

}

**Output(s) (Screen Shot):**

****

**Conclusion and discussion:**

In this experiment, OpenGL transparency was implemented by blending colors, allowing objects to appear partially transparent.

**Date:**

**Signature of faculty in-charge**

**Post lab Question**

**Improvise the code and take user input for transparency percentage**

**(Make it interactive with user)**

*#include* <GL/glut.h>

*#include* <iostream>

float alphaGreenBottom = 0.5f;

float alphaRedTop = 0.5f;

float alphaRedBottom = 0.5f;

float alphaGreenTop = 0.5f;

void init() {

*// Set clear color to white*

    glClearColor(1.0, 1.0, 1.0, 1.0);

*// Enable blending*

    glEnable(GL\_BLEND);

    glBlendFunc(GL\_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA);

}

void display() {

    glClear(GL\_COLOR\_BUFFER\_BIT);

*// First set: Green on bottom, Red on top*

*// Draw green square*

    glColor4f(0.0f, 1.0f, 0.0f, alphaGreenBottom); *// Set transparency from user input*

    glBegin(GL\_POLYGON);

    glVertex2f(-0.8f, -0.5f);

    glVertex2f(-0.3f, -0.5f);

    glVertex2f(-0.3f, 0.0f);

    glVertex2f(-0.8f, 0.0f);

    glEnd();

*// Draw red square on top*

    glColor4f(1.0f, 0.0f, 0.0f, alphaRedTop); *// Set transparency from user input*

    glBegin(GL\_POLYGON);

    glVertex2f(-0.55f, -0.25f);

    glVertex2f(-0.05f, -0.25f);

    glVertex2f(-0.05f, 0.25f);

    glVertex2f(-0.55f, 0.25f);

    glEnd();

*// Second set: Red on bottom, Green on top*

*// Draw red square*

    glColor4f(1.0f, 0.0f, 0.0f, alphaRedBottom); *// Set transparency from user input*

    glBegin(GL\_POLYGON);

    glVertex2f(0.3f, -0.5f);

    glVertex2f(0.8f, -0.5f);

    glVertex2f(0.8f, 0.0f);

    glVertex2f(0.3f, 0.0f);

    glEnd();

*// Draw green square on top*

    glColor4f(0.0f, 1.0f, 0.0f, alphaGreenTop); *// Set transparency from user input*

    glBegin(GL\_POLYGON);

    glVertex2f(0.55f, -0.25f);

    glVertex2f(1.05f, -0.25f);

    glVertex2f(1.05f, 0.25f);

    glVertex2f(0.55f, 0.25f);

    glEnd();

    glFlush();

}

int main(int *argc*, char\*\* *argv*) {

*// Take user input for transparency values*

    std::cout << "Enter transparency value for Green Square (Bottom) (0.0 - 1.0): ";

    std::cin >> alphaGreenBottom;

    std::cout << "Enter transparency value for Red Square (Top of Green) (0.0 - 1.0): ";

    std::cin >> alphaRedTop;

    std::cout << "Enter transparency value for Red Square (Bottom) (0.0 - 1.0): ";

    std::cin >> alphaRedBottom;

    std::cout << "Enter transparency value for Green Square (Top of Red) (0.0 - 1.0): ";

    std::cin >> alphaGreenTop;

*// Initialize GLUT and create window*

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

    glutInitWindowSize(500, 500);

    glutInitWindowPosition(100, 100);

    glutCreateWindow("Transparency Example - User Input for Transparency");

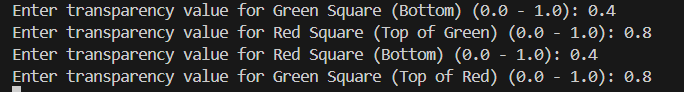
    init();

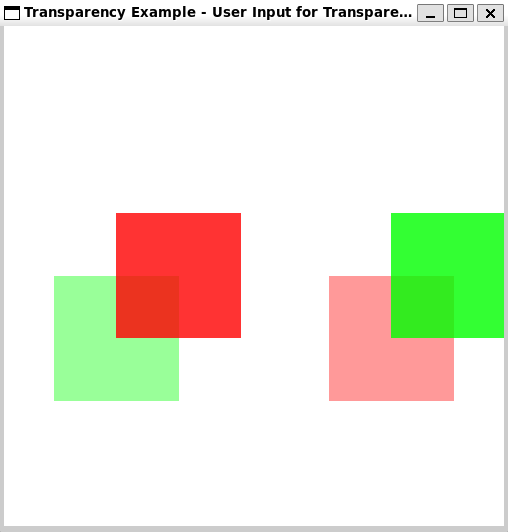
    glutDisplayFunc(display);

    glutMainLoop();

*return* 0;

}

****

****