KYAMBOGO UNIVERSITY

FACULTY OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

BACHELOR OF INFORMATION TECHNOLOGY AND COMPUTING

COURSE UNIT: COMPUTER GRAPHICS

COURSE CODE : IT313

PROGRAM : BITC3

TASK: COURSE WORK

LECTURE'S NAME: MR MATOVU MOSES

NAME	REG NO
WALAKIRA JOHN	20/U/ITD/9775/PD
AHAIRWE JORDAN	20/U/ITD/9725/PD
MASABA CALVIN	20/U/ITD/12586/GV
NAGAWA GRACE TRANELLA	20/U/ITD/9764/PD

EXPLANATION

This code is a C++ implementation of an interactive shape drawing program using the OpenGL library. The program allows the user to draw different shapes (circle, rectangle, triangle, and square) on a graphical window and manipulate their colors.

Here is an overview of the code:

Libraries: The necessary OpenGL libraries are included at the beginning of the code. These libraries provide the required functions and constants for OpenGL rendering.

Shape Drawing Functions: Four functions are defined to draw the different shapes: drawCircle(), drawRectangle(), drawTriangle(), and drawSquare(). These functions use OpenGL commands to specify the vertices and draw the shapes on the window.

Display Function: The display() function is responsible for rendering the shapes on the window. It sets the object color based on the objectColor array and calls the appropriate shape drawing function based on the value of the shape variable.

Menu Options Function: The menuOptions() function handles user interaction with the program. It is called when a menu option is selected. Depending on the chosen option, it updates the shape variable or modifies the object and background colors. After making the changes, it triggers a redisplay of the window to reflect the updates.

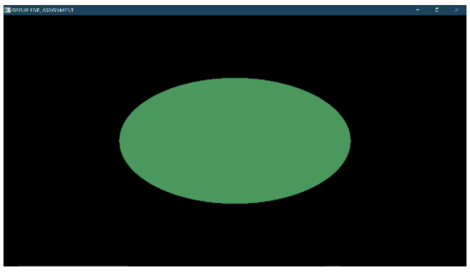
Main Function: The main() function initializes the OpenGL environment, sets the window's properties, and creates menus for selecting different options. The menus are attached to the right mouse button. The display function is registered, and the main event loop is entered using glutMainLoop() to handle user input and continuously update the window.

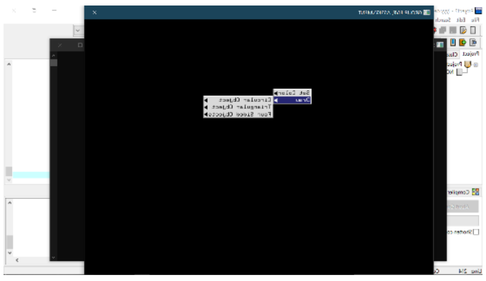
```
#include <GL/glut.h>

#include <cmath>

#include <cstdlib>

#include <ctime>


GLfloat objectColor[] = { 1.0, 1.0, 1.0 };

char shape;


void drawCircle() {

   glBegin(GL_TRIANGLE_FAN);

   glVertex2f(0, 0);

   GLfloat radius = 0.5;

   GLint numSegments = 100;

   for (int i = 0; i <= numSegments; i++) {
       GLfloat theta = 2.0 * 3.1415926 * static_cast<GLfloat>(i) / static_cast<GLfloat>(numSegments);
       GLfloat x = radius * cos(theta);
       GLfloat y = radius * sin(theta);
       glVertex2f(x, y);
   }
   glEnd();
}

void drawRectangle() {
   glBegin(GL_QUADS);
   glVertex2f(-0.8, -0.3);
   glVertex2f(0.8, -0.3);
   glVertex2f(0.8, 0.3);
   glVertex2f(-0.8, 0.3);
   glEnd();
}

void drawTriangle() {
   glBegin(GL_TRIANGLES);
   glVertex2f(-0.5, -0.5);
   glVertex2f(0.5, -0.5);
   glVertex2f(0.0, 0.5);
   glEnd();
}

void drawSquare() {
   glBegin(GL_QUADS);
   glVertex2f(-0.5, -0.5);
   glVertex2f(0.5, -0.5);
   glVertex2f(0.5, 0.5);
   glVertex2f(-0.5, 0.5);
   glEnd();
}

void display() {
   glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
   glLoadIdentity();
   glColor3f(objectColor[0], objectColor[1], objectColor[2]);

   if (shape == 'r') {
       drawRectangle();
   }
   else if (shape == 't') {
       drawTriangle();
   }
   else if (shape == 's') {
       drawSquare();
   }
   else if (shape == 'c') {
       drawCircle();
   }

   glFlush();
   glutSwapBuffers();
}

void menuOptions(int option) {
   switch (option) {
   case 7:
       shape = 's';
       glutPostRedisplay();
       break;
```

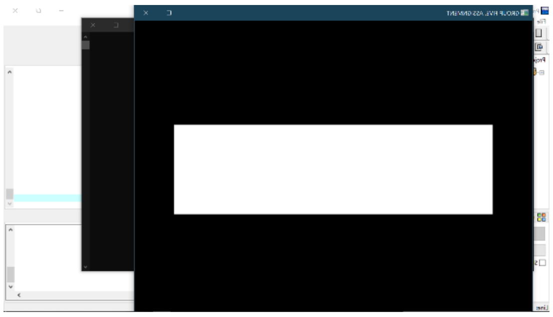
```
case 6:
   shape = 'r';
   glutPostRedisplay();
   break;
case 5:
   shape = 't';
   glutPostRedisplay();
   break;
case 4:
   shape = 'c';
   glutPostRedisplay();
   break;
case 3: {
   glClear(GL_COLOR_BUFFER_BIT);
   GLfloat red = static_cast<GLfloat>(rand()) / static_cast<GLfloat>(RAND_MAX);
   GLfloat green = static_cast<GLfloat>(rand()) / static_cast<GLfloat>(RAND_MAX);
   GLfloat blue = static_cast<GLfloat>(rand()) / static_cast<GLfloat>(RAND_MAX);
   objectColor[0] = red;
   objectColor[1] = green;
   objectColor[2] = blue;
   glutPostRedisplay();
   break;
}
case 2: {
   GLfloat red = static_cast<GLfloat>(rand()) / static_cast<GLfloat>(RAND_MAX);
   GLfloat green = static_cast<GLfloat>(rand()) / static_cast<GLfloat>(RAND_MAX);
   GLfloat blue = static_cast<GLfloat>(rand()) / static_cast<GLfloat>(RAND_MAX);
   glClearColor(red, green, blue, 1.0);
   glutPostRedisplay();
   break;
}
```

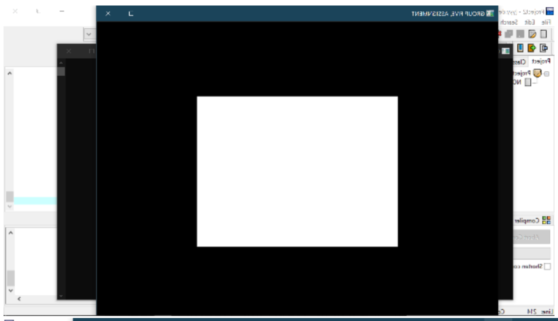
```
    case 1:
        glClearColor(0.0, 0.0, 0.0, 1.0);
        glutPostRedisplay();
        break;
    }
}

int main(int argc, char** argv) {
    shape = '\0';

    glutInit(&argc, argv);
    gluOrtho2D(-10, 10, -10, 10);
    glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE);
    glutInitWindowSize(800, 600);
    glutCreateWindow("GROUP FIVE, ASSIGNMENT");

    int menuFourSides = glutCreateMenu(menuOptions);
    glutAddMenuEntry("Object 3", 6);
    glutAddMenuEntry("Object 4", 7);
    int menuTriangular = glutCreateMenu(menuOptions);
    glutAddMenuEntry("Object 2", 5);
    int menuCircular = glutCreateMenu(menuOptions);
    glutAddMenuEntry("Object 1", 4);
    int menuSetBgColors = glutCreateMenu(menuOptions);
    glutAddMenuEntry("Clear Background Color", 1);
    glutAddMenuEntry("Set Background Color", 2);
    int menuSetObjColors = glutCreateMenu(menuOptions);
    glutAddMenuEntry("Set Object Color", 3);

    int menuDraw = glutCreateMenu(menuOptions);
    glutAddSubMenu("Circular Object", menuCircular);
    glutAddSubMenu("Triangular Object", menuTriangular);
    glutAddSubMenu("Four Sided Objects", menuFourSides);

    int menuSetColors = glutCreateMenu(menuOptions);
    glutAddSubMenu("Background Color", menuSetBgColors);
    glutAddSubMenu("Object Color", menuSetObjColors);

    int menuRight = glutCreateMenu(menuOptions);
    glutAddSubMenu("Set Color", menuSetColors);
    glutAddSubMenu("Draw", menuDraw);
    glutAttachMenu(GLUT_RIGHT_BUTTON);

    glutDisplayFunc(display);

    srand(static_cast<unsigned int>(time(0)));

    glutMainLoop();
    return 0;
}
```

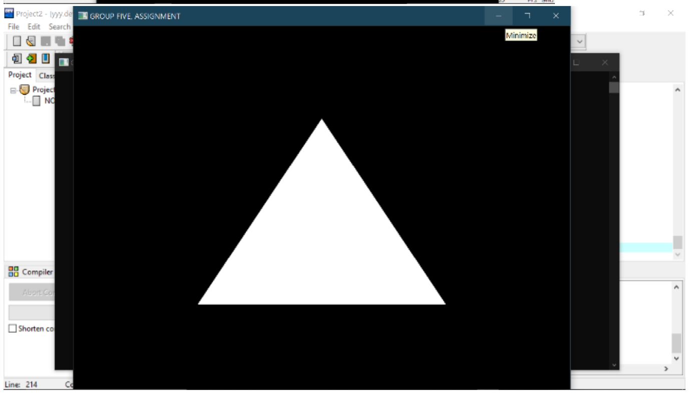
Output of the code

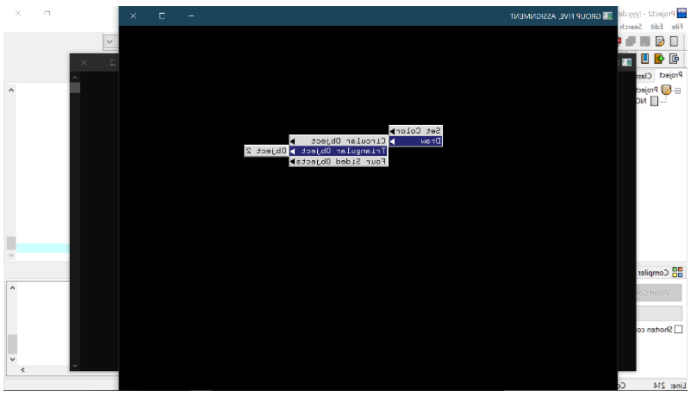





GROUP FIVE ASSIGNMENT


GROUP FIVE ASSIGNMENT


GROUP FIVE ASSIGNMENT
Minimize
Project2 - lyyy.de
File Edit Search
Project Class
Compiler
Shorten co
Line: 214


GROUP FIVE ASSIGNMENT
Set Color
Draw
Circular Object
Triangular Object
Four Sided Objects
Object 2

Conclusion

The code allows the user to select different shapes, change the object color, and modify the background color. Each shape is associated with a menu option, and the color options are organized into separate menus. When a menu option is chosen, the corresponding function is executed to update the program's state, and the window is redrawn to reflect the changes.