**UNITY UNIVERSITY ADAMA CAMPUS**

**DEPARTMENT OF COMPUTER SCIENCE**

**GAPHICS GROUP ASSIGNMENT**



**Group members** **Id-number**

1. Abdulselam Jeylan 06117/14

2. Abel Mesfin 04575/14

3. Eyuel Abiyot 04570/14

4. Nooreldaim Zeinu 04637/14

5. Yohannes Asirat 04579/14

**1. Displays a static picture of a tetrahedron**

#include <GL/glut.h>

void display(){

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_LINES);

for (GLfloat i = -2.5; i <= 2.5; i+= 0.25){

glVertex3f(i,0,2.5);

glVertex3f(i,0,-2.5);

glVertex3f(2.5,0,i);

glVertex3f(-2.5,0,i);

}

glEnd();

glBegin(GL\_TRIANGLE\_STRIP);

glColor3f(1,1,1); glVertex3f(0,2,0);

glColor3f(1,0,0);glVertex3f(-1,0,1);

glColor3f(0,1,0);glVertex3f(1,0,1);

glColor3f(0,0,1);glVertex3f(0,0,-1.4);

glColor3f(1,1,1);glVertex3f(0,2,0);

glColor3f(1,0,0);glVertex3f(-1,0,1);

glEnd();

glFlush();

}

void init(){

glClearColor(0.1, 0.39, 0.88, 1.0);

glColor3f(1.0, 1.0, 1.0);

glEnable(GL\_CULL\_FACE);

glCullFace(GL\_BACK);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glFrustum(-2, 2, -1.5, 1.5,1,40);

glMatrixMode(GL\_MODELVIEW);

glTranslatef(0,0,-3);

glRotatef(50,1,0,0);

glRotatef(70,0,1,0);

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowPosition(80,80);

glutInitWindowSize(800, 600);

glutCreateWindow("Tetrahedron");

glutDisplayFunc(display);

init();

glutMainLoop();

}

**2. Display Ethiopian flag**

#include <windows.h>

#include <GL/glut.h>

#define GLUT\_DISABLE\_ATECTIT\_HACK;

void init()

{

glClearColor(1,1,1,0);

glMatrixMode(GL\_PROJECTION);

glOrtho(0, 100, 0, 100, 0, 10);

}

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_QUADS);

glColor3f(0.0, 1.0, 0.0);

glVertex2i(10,80);

glVertex2i(40,80);

glVertex2i(40,70);

glVertex2i(10,70);

glColor3f(1.0, 1.0,0.0);

glVertex2i(10,70);

glVertex2i(40,70);

glVertex2i(40,60);

glVertex2i(10,60);

glColor3f(1.0, 0.0, 0.0);

glVertex2i(10,60);

glVertex2i(40,60);

glVertex2i(40,50);

glVertex2i(10,50);

glEnd();

glFlush();

}

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

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowPosition(80,80);

glutInitWindowSize(800, 600);

glutCreateWindow("Ethiopian Flag");

init();

glutDisplayFunc(display);

glutMainLoop();

}

**3. Display the following Shapes**

#include <GL/glut.h>

#include <math.h>

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

// Red square

glColor3f(1.0, 0.0, 0.0);

glBegin(GL\_QUADS);

glVertex2f(-0.9, 0.5);

glVertex2f(-0.5, 0.5);

glVertex2f(-0.5, 0.9);

glVertex2f(-0.9, 0.9);

glEnd();

// Green square

glBegin(GL\_QUADS);

glColor3f(0.0, 1.0, 0.0);

glVertex2f(-0.9, -0.1);

glVertex2f(-0.5, -0.1);

glVertex2f(-0.5, -0.5);

glVertex2f(-0.9, -0.5);

glEnd();

//white grayish gradient square

glBegin(GL\_QUADS);

glColor3f(0.5, 0.5, 0.5);

glColor3f(0.75, 0.75, 0.75);

glVertex2f(-0.9, -0.6);

glVertex2f(-0.5, -0.6);

glColor3f(0.75, 0.75, 0.75);

glColor3f(0.5, 0.5, 0.5);

glVertex2f(-0.5, -0.9);

glVertex2f(-0.9, -0.9);

glEnd();

// Yellow hexagon

glColor3f(1.0, 1.0, 0.0);

glBegin(GL\_POLYGON);

for (int i = 0; i < 6; ++i)

{

glVertex2f(0.5 \* cos(i \* 2 \* 3.1415926 / 6), 0.5 + 0.5 \* sin(i \* 2 \* 3.1415926 / 6));

}

glEnd();

// Blue triangle

glColor3f(0.0, 0.0, 1.0);

glBegin(GL\_TRIANGLES);

glVertex2f(0.0, -0.5);

glVertex2f(0.5, -1.0);

glVertex2f(-0.5, -1.0);

glEnd();

// Triangle with gradient

glBegin(GL\_TRIANGLES);

glColor3f(1.0, 0.0, 0.0);

glVertex2f(0.0, -0.5);

glColor3f(0.0, 1.0, 0.0);

glVertex2f(0.5, -1.0);

glColor3f(0.0, 0.0, 1.0);

glVertex2f(0.5, 0.0);

glEnd();

glFlush();

}

void init()

{

glClearColor(0.0, 0.0, 0.0, 1.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(-1.0, 1.0, -1.0, 1.0);

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(500, 500);

glutInitWindowPosition(100, 100);

glutCreateWindow("Display Shapes");

init();

glutDisplayFunc(display);

glutMainLoop();

return 0;

}

**4. Display the Germany flag**

#include <windows.h>

#include <GL/glut.h>

#define GLUT\_DISABLE\_ATECTIT\_HACK;

void init()

{

glClearColor(1,1,1,0);

glMatrixMode(GL\_PROJECTION);

glOrtho(0, 100, 0, 100, 0, 10);

}

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_QUADS);

glColor3f(0.0, 0.0, 0.0);

glVertex2i(10,80);

glVertex2i(40,80);

glVertex2i(40,70);

glVertex2i(10,70);

glColor3f(1.0, 0.0,0.0);

glVertex2i(10,70);

glVertex2i(40,70);

glVertex2i(40,60);

glVertex2i(10,60);

glColor3f(1.0, 1.0, 0.0);

glVertex2i(10,60);

glVertex2i(40,60);

glVertex2i(40,50);

glVertex2i(10,50);

glEnd();

glFlush();

}

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

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowPosition(80,80);

glutInitWindowSize(800, 600);

glutCreateWindow("Deutsch Flag");

init();

glutDisplayFunc(display);

glutMainLoop();

}

**5. Display the following shape**

#include <GL/glut.h>

#include <cmath>

void drawCircle(float centerX, float centerY, float radius, int segments, float r, float g, float b) {

glColor3f(r, g, b);

glBegin(GL\_LINE\_LOOP);

for(int i = 0; i < segments; ++i) {

float theta = 2.0f \* 3.1415926f \* float(i) / float(segments);

float x = radius \* cosf(theta);

float y = radius \* sinf(theta);

glVertex2f(x + centerX, y + centerY);

}

glEnd();

}

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT);

// Background color

glClearColor(0.2f, 0.4f, 0.6f, 1.0f);

glClear(GL\_COLOR\_BUFFER\_BIT);

// Circles

float centerX = 0.0f, centerY = 0.4f;

drawCircle(centerX, centerY, 0.50f, 100, 1.0f, 0.0f, 0.0f); // Red outer circle

drawCircle(centerX, centerY, 0.45f, 100, 1.0f, 1.0f, 0.0f); // Yellow circle

drawCircle(centerX, centerY, 0.35f, 100, 0.0f, 0.0f, 0.0f); // Black circle

// Green filled circle

glColor3f(0.0f, 1.0f, 0.0f);

glBegin(GL\_POLYGON);

for(int i = 0; i < 100; ++i) {

float theta = 2.0f \* 3.1415926f \* float(i) / float(100);

float x = 0.1f \* cosf(theta);

float y = 0.1f \* sinf(theta);

glVertex2f(x + centerX, y + centerY);

}

glEnd();

// Orange rectangle

glColor3f(1.0f, 0.5f, 0.0f);

glBegin(GL\_QUADS);

glVertex2f(-0.4f, -0.5f);

glVertex2f(0.4f, -0.5f);

glVertex2f(0.4f, -0.3f);

glVertex2f(-0.4f, -0.3f);

glEnd();

glFlush();

}

void init() {

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(-1.0, 1.0, -1.0, 1.0);

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(500, 500);

glutInitWindowPosition(100, 100);

glutCreateWindow("Display Shapes");

init();

glutDisplayFunc(display);

glutMainLoop();

return 0;

}