

1

COMPUTER GRAPHICS

GROUP ASSIGNMENT

**Names**   **ID**

1. Tesfalem Berhe 04535/14
2. Yesuneh Moges 04558/14
3. Zelalem Muluneh 04559/14
4. Mulisa Tekle 04303/14
5. Matiyos Markos 04548/14
6. Michias Eshetu 04560/14

**SUBMITTED TO:**

Mr.KETEMA.

**1, Static picture of tetrahedron**

#include <GL/glut.h>

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glLoadIdentity();

glTranslatef(0.0f, 0.0f, -5.0f);

glBegin(GL\_TRIANGLES);

// Front

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

glVertex3f(0.0f, 1.0f, 0.0f);

glVertex3f(-1.0f, -1.0f, 1.0f);

glVertex3f(1.0f, -1.0f, 1.0f);

// Right

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

glVertex3f(0.0f, 1.0f, 0.0f);

glVertex3f(1.0f, -1.0f, 1.0f);

glVertex3f(1.0f, -1.0f, -1.0f);

// Back

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

glVertex3f(0.0f, 1.0f, 0.0f);

glVertex3f(1.0f, -1.0f, -1.0f);

glVertex3f(-1.0f, -1.0f, -1.0f);

// Left

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

glVertex3f(0.0f, 1.0f, 0.0f);

glVertex3f(-1.0f, -1.0f, -1.0f);

glVertex3f(-1.0f, -1.0f, 1.0f);

glEnd();

glutSwapBuffers();

}

void init() {

glEnable(GL\_DEPTH\_TEST);

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB | GLUT\_DEPTH);

glutInitWindowSize(800, 600);

glutCreateWindow("Tetrahedron");

init();

glutDisplayFunc(display);

glutMainLoop();

return 0;

}

1. **Ethiopian flag**

#include <GL/glut.h>

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_QUADS);

// Green

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

glVertex2f(-1.0f, 0.333f);

glVertex2f(1.0f, 0.333f);

glVertex2f(1.0f, 1.0f);

glVertex2f(-1.0f, 1.0f);

// Yellow

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

glVertex2f(-1.0f, -0.333f);

glVertex2f(1.0f, -0.333f);

glVertex2f(1.0f, 0.333f);

glVertex2f(-1.0f, 0.333f);

// Red

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

glVertex2f(-1.0f, -1.0f);

glVertex2f(1.0f, -1.0f);

glVertex2f(1.0f, -0.333f);

glVertex2f(-1.0f, -0.333f);

glEnd();

glutSwapBuffers();

}

void init() {

glClearColor(0.0f, 0.0f, 0.0f, 1.0f);

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);

glutInitWindowSize(800, 600);

glutCreateWindow("Ethiopian Flag");

init();

glutDisplayFunc(display);

glutMainLoop();

return 0;

}

1. **Display shapes**

#include <GL/glut.h>

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_QUADS);

// Red Square

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

glVertex2f(-0.8f, 0.2f);

glVertex2f(-0.4f, 0.2f);

glVertex2f(-0.4f, 0.6f);

glVertex2f(-0.8f, 0.6f);

// Green Square

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

glVertex2f(0.4f, -0.2f);

glVertex2f(0.8f, -0.2f);

glVertex2f(0.8f, 0.2f);

glVertex2f(0.4f, 0.2f);

glEnd();

glBegin(GL\_POLYGON);

// Yellow Hexagon

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

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

glVertex2f(0.3f \* cos(i \* 2.0f \* 3.14159f / 6), 0.3f \* sin(i \* 2.0f \* 3.14159f / 6));

}

glEnd();

glBegin(GL\_TRIANGLES);

// Multicolored Triangle

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

glVertex2f(-0.2f, -0.8f);

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

glVertex2f(0.2f, -0.8f);

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

glVertex2f(0.0f, -0.4f);

glEnd();

glutSwapBuffers();

}

void init() {

glClearColor(0.0f, 0.0f, 0.0f, 1.0f);

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);

glutInitWindowSize(800, 600);

glutCreateWindow("Shapes");

init();

glutDisplayFunc(display);

glutMainLoop();

return 0;

}

1. **German flag**

#include <GL/glut.h>

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_QUADS);

// Red Square

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

glVertex2f(-0.8f, 0.2f);

glVertex2f(-0.4f, 0.2f);

glVertex2f(-0.4f, 0.6f);

glVertex2f(-0.8f, 0.6f);

// Green Square

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

glVertex2f(0.4f, -0.2f);

glVertex2f(0.8f, -0.2f);

glVertex2f(0.8f, 0.2f);

glVertex2f(0.4f, 0.2f);

glEnd();

glBegin(GL\_POLYGON);

// Yellow Hexagon

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

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

glVertex2f(0.3f \* cos(i \* 2.0f \* 3.14159f / 6), 0.3f \* sin(i \* 2.0f \* 3.14159f / 6));

}

glEnd();

glBegin(GL\_TRIANGLES);

// Multicolored Triangle

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

glVertex2f(-0.2f, -0.8f);

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

glVertex2f(0.2f, -0.8f);

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

glVertex2f(0.0f, -0.4f);

glEnd();

glutSwapBuffers();

}

void init() {

glClearColor(0.0f, 0.0f, 0.0f, 1.0f);

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);

glutInitWindowSize(800, 600);

glutCreateWindow("Shapes");

init();

glutDisplayFunc(display);

glutMainLoop();

return 0;

}

**5, Display the following shapes**

#include <GL/glut.h>

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT);

// Set the background color to blue

glClearColor(0.0f, 0.0f, 1.0f, 1.0f);

// Draw the concentric circles

const float radius1 = 0.1f;

const float radius2 = 0.2f;

const float radius3 = 0.3f;

const float radius4 = 0.4f;

const float radius5 = 0.5f;

const int numSegments = 100;

glBegin(GL\_TRIANGLE\_FAN);

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

glVertex2f(0.0f, 0.0f);

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

float angle = 2.0f \* 3.14159f \* i / numSegments;

glVertex2f(radius1 \* cos(angle), radius1 \* sin(angle));

}

glEnd();

glBegin(GL\_LINE\_LOOP);

glColor3f(0.0f, 0.0f, 0.0f); // Black

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

float angle = 2.0f \* 3.14159f \* i / numSegments;

glVertex2f(radius2 \* cos(angle), radius2 \* sin(angle));

}

glEnd();

glBegin(GL\_LINE\_LOOP);

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

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

float angle = 2.0f \* 3.14159f \* i / numSegments;

glVertex2f(radius3 \* cos(angle), radius3 \* sin(angle));

}

glEnd();

glBegin(GL\_LINE\_LOOP);

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

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

float angle = 2.0f \* 3.14159f \* i / numSegments;

glVertex2f(radius4 \* cos(angle), radius4 \* sin(angle));

}

glEnd();

glBegin(GL\_LINE\_LOOP);

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

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

float angle = 2.0f \* 3.14159f \* i / numSegments;

glVertex2f(radius5 \* cos(angle), radius5 \* sin(angle));

}

glEnd();

// Draw the rectangle

glBegin(GL\_QUADS);

glColor3f(1.0f, 0.6f, 0.2f); // Orange

glVertex2f(-0.3f, -0.8f);

glVertex2f(0.3f, -0.8f);

glVertex2f(0.3f, -0.6f);

glVertex2f(-0.3f, -0.6f);

glEnd();

glutSwapBuffers();

}

void init() {

glClearColor(0.0f, 0.0f, 1.0f, 1.0f); // Set the initial background color to blue

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);

glutInitWindowSize(800, 600);

glutCreateWindow("Complex Shape");

init();

glutDisplayFunc(display);

glutMainLoop();

return 0;

}