**Lab Report. 10**

**Subject: Computer Graphics Lab**



Submitted to: Dr. Junaid

Submitted by: Nayab Ashraf (RC-299)

**Date of Submission: 11 December, 2024**

National University of Modern Languages, Department of Computer Science, Rawalpindi Campus.

**Task 01:**

#include <stdio.h>

#include <stdarg.h>

#include <math.h>

//#define GL\_GLEXT\_PROTOTYPES

#ifdef \_\_APPLE\_\_

#include <GLUT/glut.h>

#else

#include <GL/glut.h>

#endif

// ----------------------------------------------------------

// Function Prototypes

// ----------------------------------------------------------

void display();

void specialKeys();

// ----------------------------------------------------------

// Global Variables

// ----------------------------------------------------------

double rotate\_y = 0;

double rotate\_x = 0;

// ----------------------------------------------------------

// display() Callback function

// ----------------------------------------------------------

void display() {

// Clear screen and Z-buffer

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

// Reset transformations

glLoadIdentity();

// Other Transformations

// glTranslatef( 0.1, 0.0, 0.0 ); // Not included

// glRotatef( 180, 0.0, 1.0, 0.0 ); // Not included

// Rotate when user changes rotate\_x and rotate\_y

glRotatef(rotate\_x, 1.0, 0.0, 0.0);

glRotatef(rotate\_y, 0.0, 1.0, 0.0);

// Other Transformations

// glScalef( 2.0, 2.0, 0.0 ); // Not included

//Multi-colored side - FRONT

// Yellow side - Front

glBegin(GL\_POLYGON);

glColor3f(1.0, 1.0, 0.0);

glVertex3f(0.5, -0.5, -0.5);

//glVertex3f(0.5, 0.5, -0.5);

//glVertex3f(-0.5, 0.5, -0.5);

glVertex3f(-0.5, -0.5, -0.5);

glVertex3f(0, 0.5, 0);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(1.0, 1.0, 1.0);//White back

glVertex3f(0.5, -0.5, 0.5);

//glVertex3f(0.5, 0.5, 0.5);

//glVertex3f(-0.5, 0.5, 0.5);

glVertex3f(-0.5, -0.5, 0.5);

glVertex3f(0, 0.5, 0);

glEnd();

// Purple side - RIGHT

glBegin(GL\_POLYGON);

glColor3f(1.0, 0.0, 1.0);

glVertex3f(0.5, -0.5, -0.5);

//glVertex3f(0.5, 0.5, -0.5);

//glVertex3f(0.5, 0.5, 0.5);

glVertex3f(0.5, -0.5, 0.5);

glVertex3f(0, 0.5, 0);

glEnd();

// Green side - LEFT

glBegin(GL\_POLYGON);

glColor3f(0.0, 1.0, 0.0);

glVertex3f(-0.5, -0.5, 0.5);

//glVertex3f(-0.5, 0.5, 0.5);

//glVertex3f(-0.5, 0.5, -0.5);

glVertex3f(-0.5, -0.5, -0.5);

glVertex3f(0, 0.5, 0);

glEnd();

// Red side - BOTTOM

glBegin(GL\_POLYGON);

glColor3f(1.0, 0.0, 0.0);

glVertex3f(0.5, -0.5, -0.5);

glVertex3f(0.5, -0.5, 0.5);

glVertex3f(-0.5, -0.5, 0.5);

glVertex3f(-0.5, -0.5, -0.5);

glEnd();

glFlush();

glutSwapBuffers();

}

// ----------------------------------------------------------

// specialKeys() Callback Function

// ----------------------------------------------------------

void specialKeys(int key, int x, int y) {

// Right arrow - increase rotation by 5 degree

if (key == GLUT\_KEY\_RIGHT)

rotate\_y -= 5;

// Left arrow - decrease rotation by 5 degree

else if (key == GLUT\_KEY\_LEFT)

rotate\_y += 5;

else if (key == GLUT\_KEY\_UP)

rotate\_x += 5;

else if (key == GLUT\_KEY\_DOWN)

rotate\_x -= 5;

// Request display update

glutPostRedisplay();

}

// ----------------------------------------------------------

// main() function

// ----------------------------------------------------------

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

// Initialize GLUT and process user parameters

glutInit(&argc, argv);

// Request double buffered true color window with Z-buffer

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

// Create window

glutCreateWindow("Awesome Cube");

// Enable Z-buffer depth test

glEnable(GL\_DEPTH\_TEST);

// Callback functions

glutDisplayFunc(display);

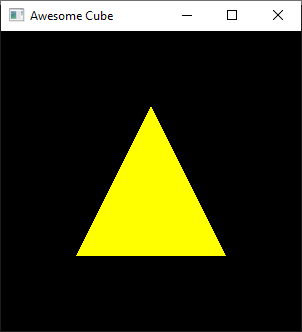
glutSpecialFunc(specialKeys);

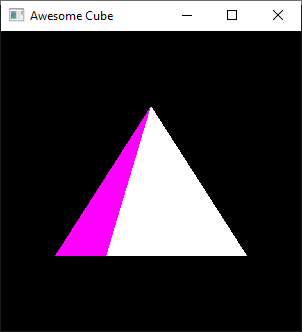
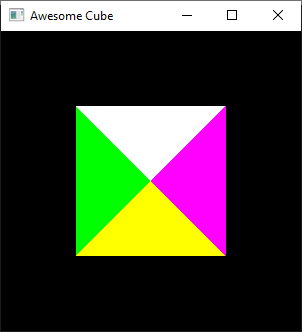
// Pass control to GLUT for events

glutMainLoop();

// Return to OS

return 0;

****}

****

**Task 02:**

#include <stdio.h>

#include <stdarg.h>

#include <math.h>

//#define GL\_GLEXT\_PROTOTYPES

#ifdef \_\_APPLE\_\_

#include <GLUT/glut.h>

#else

#include <GL/glut.h>

#endif

// ----------------------------------------------------------

// Function Prototypes

// ----------------------------------------------------------

void display();

void specialKeys();

// ----------------------------------------------------------

// Global Variables

// ----------------------------------------------------------

double rotate\_y = 0;

double rotate\_x = 0;

// ----------------------------------------------------------

// display() Callback function

// ----------------------------------------------------------

void display() {

// Clear screen and Z-buffer

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

// Reset transformations

glLoadIdentity();

glOrtho(-2.0f, 2.0f, -2.0f, 2.0f, -2.0f, 2.0f);

// Other Transformations

// glTranslatef( 0.1, 0.0, 0.0 ); // Not included

// glRotatef( 180, 0.0, 1.0, 0.0 ); // Not included

// Rotate when user changes rotate\_x and rotate\_y

glRotatef(rotate\_x, 1.0, 0.0, 0.0);

glRotatef(rotate\_y, 0.0, 1.0, 0.0);

// Other Transformations

// glScalef( 2.0, 2.0, 0.0 ); // Not included

//Multi-colored side - FRONT

// Yellow side - Front

glBegin(GL\_TRIANGLES);

glColor3f(1.0, 1.0, 0.0);

glVertex3f(0.5, -0.5, -0.5);

//glVertex3f(0.5, 0.5, -0.5);

//glVertex3f(-0.5, 0.5, -0.5);

glVertex3f(-0.5, -0.5, -0.5);

glVertex3f(0, 0.5, 0);

glVertex3f(0.5, -0.5, -0.5);

glVertex3f(-0.5, -0.5, -0.5);

glVertex3f(0, -1.5, 0);

glEnd();

glBegin(GL\_TRIANGLES);

glColor3f(1.0, 1.0, 1.0);//White back

glVertex3f(0.5, -0.5, 0.5);

//glVertex3f(0.5, 0.5, 0.5);

//glVertex3f(-0.5, 0.5, 0.5);

glVertex3f(-0.5, -0.5, 0.5);

glVertex3f(0, 0.5, 0);

glVertex3f(0.5, -0.5, 0.5);

glVertex3f(-0.5, -0.5, 0.5);

glVertex3f(0, -1.5, 0);

glEnd();

// Purple side - RIGHT

glBegin(GL\_TRIANGLES);

glColor3f(1.0, 0.0, 1.0);

glVertex3f(0.5, -0.5, -0.5);

//glVertex3f(0.5, 0.5, -0.5);

//glVertex3f(0.5, 0.5, 0.5);

glVertex3f(0.5, -0.5, 0.5);

glVertex3f(0, 0.5, 0);

glVertex3f(0.5, -0.5, -0.5);

glVertex3f(0.5, -0.5, 0.5);

glVertex3f(0, -1.5, 0);

glEnd();

// Green side - LEFT

glBegin(GL\_TRIANGLES);

glColor3f(0.0, 1.0, 0.0);

glVertex3f(-0.5, -0.5, 0.5);

//glVertex3f(-0.5, 0.5, 0.5);

//glVertex3f(-0.5, 0.5, -0.5);

glVertex3f(-0.5, -0.5, -0.5);

glVertex3f(0, 0.5, 0);

glVertex3f(-0.5, -0.5, 0.5);

glVertex3f(-0.5, -0.5, -0.5);

glVertex3f(0, -1.5, 0);

glEnd();

glFlush();

glutSwapBuffers();

}

// ----------------------------------------------------------

// specialKeys() Callback Function

// ----------------------------------------------------------

void specialKeys(int key, int x, int y) {

// Right arrow - increase rotation by 5 degree

if (key == GLUT\_KEY\_RIGHT)

rotate\_y -= 5;

// Left arrow - decrease rotation by 5 degree

else if (key == GLUT\_KEY\_LEFT)

rotate\_y += 5;

else if (key == GLUT\_KEY\_UP)

rotate\_x += 5;

else if (key == GLUT\_KEY\_DOWN)

rotate\_x -= 5;

// Request display update

glutPostRedisplay();

}

// ----------------------------------------------------------

// main() function

// ----------------------------------------------------------

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

// Initialize GLUT and process user parameters

glutInit(&argc, argv);

// Request double buffered true color window with Z-buffer

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

// Create window

glutCreateWindow("Awesome Cube");

// Enable Z-buffer depth test

glEnable(GL\_DEPTH\_TEST);

// Callback functions

glutDisplayFunc(display);

glutSpecialFunc(specialKeys);

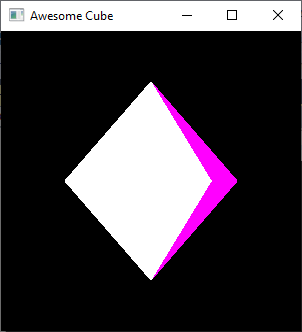
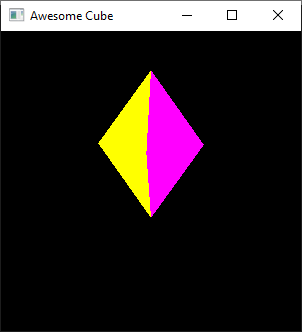
// Pass control to GLUT for events

glutMainLoop();

// Return to OS

return 0;

}

****