NATIONAL UNIVERSITY OF MODERN LANGUAGES

Human Computer Interaction Lab Report - 8

NAME: M Abdul Rafay

ROLL NO: FL23791

PROGRAM: BSSE (5th Semester)

COURSE: Human Computer Interaction

SUBMITTED TO: Sir Khateeb

DATE: Sept, 27/2025 **DAY: SATURDAY**

1. **Task No 1: Make the cube rotate automatically (without key presses) using the idle function.**

```
#include <stdio.h>
#include <math.h>
#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>

double rotate_y = 0;
double rotate_x = 0;

// Cube color (starts white)
GLfloat cubeColor[3] = { 1.0f, 1.0f, 1.0f };

void display();
void specialKeys(int key, int x, int y);
void keyboard(unsigned char key, int x, int y);
void idle();

void display() {
  glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
  glLoadIdentity();

  // Rotate automatically
  glRotatef(rotate_x, 1.0, 0.0, 0.0);
  glRotatef(rotate_y, 0.0, 1.0, 0.0);

  // Draw cube with selected color
  glBegin(GL_QUADS);
```

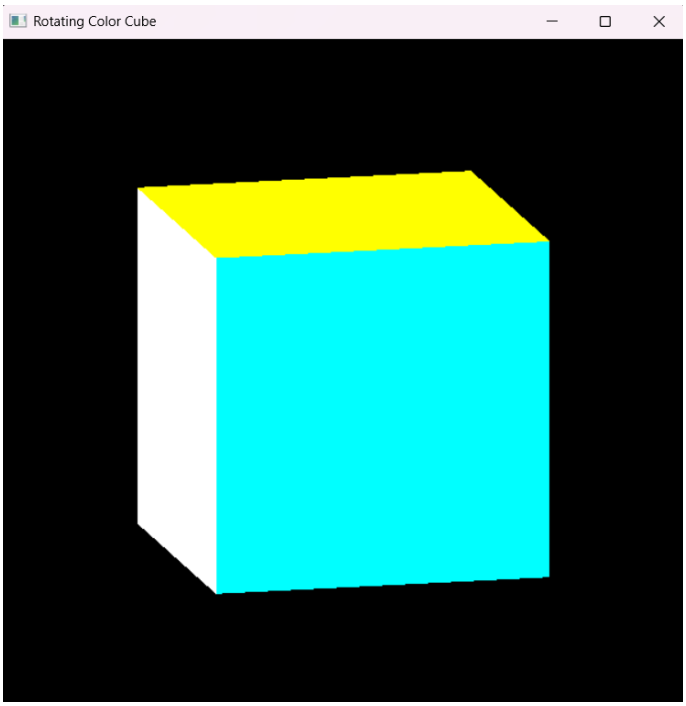
```
// FRONT face
glColor3f(cubeColor[0], cubeColor[1], cubeColor[2]);
glVertex3f(-0.5, -0.5, 0.5);
glVertex3f(0.5, -0.5, 0.5);
glVertex3f(0.5, 0.5, 0.5);
glVertex3f(-0.5, 0.5, 0.5);

// BACK face
glColor3f(cubeColor[0] * 0.8, cubeColor[1] * 0.8, cubeColor[2] * 0.8);
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);

// LEFT face
glColor3f(0.0, cubeColor[1], cubeColor[2]);
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);

// RIGHT face
glColor3f(cubeColor[0], 0.0, cubeColor[2]);
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);

// TOP face
glColor3f(cubeColor[0], cubeColor[1], 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);

    // BOTTOM face
    glColor3f(0.0, cubeColor[1], cubeColor[2]);
    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();
}
void specialKeys(int key, int x, int y) {
    if (key == GLUT_KEY_RIGHT)
        rotate_y += 5;
    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;

    glutPostRedisplay();
}
void keyboard(unsigned char key, int x, int y) {
    switch (key) {
    case 'r': case 'R': // Red
```

```
    cubeColor[0] = 1.0; cubeColor[1] = 0.0; cubeColor[2] = 0.0;
    break;
  case 'g': case 'G': // Green
    cubeColor[0] = 0.0; cubeColor[1] = 1.0; cubeColor[2] = 0.0;
    break;
  case 'b': case 'B': // Blue
    cubeColor[0] = 0.0; cubeColor[1] = 0.0; cubeColor[2] = 1.0;
    break;
  case 'y': case 'Y': // Yellow
    cubeColor[0] = 1.0; cubeColor[1] = 1.0; cubeColor[2] = 0.0;
    break;
  case 27: // ESC to exit
    exit(0);
    break;
  }
  glutPostRedisplay();
}
void idle() {
  rotate_x += 0.2;  // speed for X-axis
  rotate_y += 0.3;  // speed for Y-axis
  glutPostRedisplay();
}
int main(int argc, char* argv[]) {
  glutInit(&argc, argv);
  glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
  glutInitWindowSize(600, 600);
  glutCreateWindow("Rotating Color Cube");

  glEnable(GL_DEPTH_TEST);

  glutDisplayFunc(display);
```

```
    glutSpecialFunc(specialKeys);
    glutKeyboardFunc(keyboard);
    glutIdleFunc(idle); // <-- automatic rotation

    glClearColor(0, 0, 0, 1); // background black

    glutMainLoop();
    return 0;
}
```

Output :



Task No 2 : Change Cube Color with Keyboard Key Press

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

GLfloat angle = 0.0f;
```

```
void Draw() {
  glClear(GL_COLOR_BUFFER_BIT);
  glColor3f(1.0, 1.0, 1.0);

  glPushMatrix();
  glTranslatef(0.5f, 0.5f, 0.0f);  // move pivot to center
  glRotatef(angle, 0.0f, 0.0f, 1.0f); // rotate around Z-axis
  glBegin(GL_LINES);
  glVertex2f(-0.3f, 0.0f);
  glVertex2f(0.3f, 0.0f);
  glEnd();
  glPopMatrix();

  glFlush();
  angle += 0.1f;   // rotation speed
  if (angle > 360.0f) angle -= 360.0f;

  glutPostRedisplay();
}

void Initialize() {
  glClearColor(0.0, 0.0, 0.0, 1.0);
  glMatrixMode(GL_PROJECTION);
  glLoadIdentity();
  glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0);
}

int main(int argc, char** argv) {
  glutInit(&argc, argv);
  glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
  glutInitWindowSize(300, 300);
```

```
    glutCreateWindow("Rotating Line Animation");
    Initialize();
    glutDisplayFunc(Draw);
    glutMainLoop();
    return 0;
}
```

Output: