

Lab Taks-9

Md. Abdul Muneem Adnan

ID: 20-44213-3

- 1. Question-** Develop a code that will have four different objects (keep it simple). The objects will move to the left, right, up and down in a loop.

Code:

```
#include<cstdio>

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

GLfloat position = 0.0f;
GLfloat position1 = 0.0f;
GLfloat speed = 0.1f;
void dis();
void display();

void update(int value) {

    if(position <-1.5)
        position = 1.0f;

    position -= speed;

    glutPostRedisplay();

    glutTimerFunc(100,update,0);
}

void update1(int value) {

    if(position1 >1.5)
        position1 = -1.0f;

    position1 += speed;
```

```
        glutPostRedisplay();

        glutTimerFunc(100,update1,0);
    }

void init() {
    glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
}

void disback(int val)
{
    glutDisplayFunc(display);
}

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

    glPushMatrix();
    glTranslatef(position,0.0f, 0.0f);
    glBegin(GL_POLYGON);
        glColor3f(0.0f, 1.0f, 0.0f);
        glVertex2f(-0.2f, -0.2f);
        glVertex2f( 0.2f, -0.2f);
        glVertex2f( 0.2f,  0.2f);
        glVertex2f(-0.2f,  0.2f);

    glEnd();
    glPopMatrix();

    glPushMatrix();
    glTranslatef(position1,0.0f, 0.0f);
    glBegin(GL_POLYGON);
        glColor3f(1.0f, 0.0f, 0.0f);
        glVertex2f(-0.2f, -0.2f);
        glVertex2f( 0.2f, -0.2f);
```

```

    glVertex2f(0.2f, 0.2f);
    glVertex2f(-0.2f, 0.2f);
glEnd();
glPopMatrix();

glPushMatrix();
glTranslatef(0.0f,position, 0.0f);
glBegin(GL_POLYGON);
    glColor3f(0.0f, 0.0f, 1.0f);
    glVertex2f(-0.2f, -0.2f);
    glVertex2f( 0.2f, -0.2f);
    glVertex2f( 0.2f,  0.2f);
    glVertex2f(-0.2f,  0.2f);

glEnd();
glPopMatrix();

glPushMatrix();
glTranslatef(0.0f,position1, 0.0f);
glBegin(GL_POLYGON);
    glColor3f(1.0f, 1.0f, 0.0f);
    glVertex2f(-0.2f, -0.2f);
    glVertex2f( 0.2f, -0.2f);
    glVertex2f( 0.2f,  0.2f);
    glVertex2f(-0.2f,  0.2f);
glEnd();
glPopMatrix();

glutTimerFunc(2500,disback,0);
glFlush();
}

void dis()
{
    glutDisplayFunc(display);
}

```

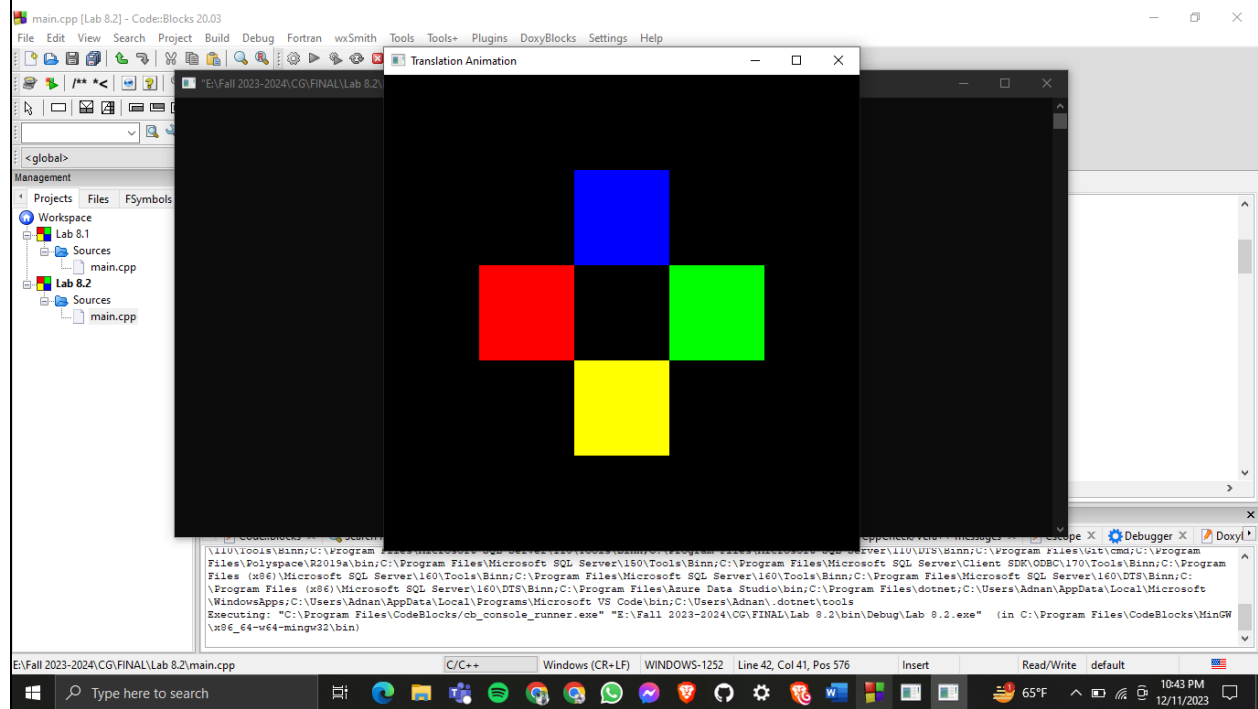
```

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitWindowSize(520, 520);
    glutInitWindowPosition(50, 50);
    glutCreateWindow("Translation Animation");
    glutDisplayFunc(dis);
    init();

    glutTimerFunc(100, update, 0);
    glutTimerFunc(100, update1, 0);
    glutMainLoop();
    return 0;
}

```

Output Screenshot (Full Screen)-



- 2. Question-** Create a simple day and night scenario using keyboard interaction. The key 'D' or 'd' will initiate the day mode, and the key 'N' or 'n' will initiate the night mode.

Code:

```

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

```

```
bool isDay = true;

void init() {
    glClearColor(0.0, 0.0, 0.0, 1.0);
    gluOrtho2D(0, 800, 0, 600);
}

void drawSunOrMoon() {
    if (isDay) {
        glColor3f(1.0, 1.0, 0.0);
    } else {
        glColor3f(1.0, 1.0, 1.0);
    }

    glBegin(GL_POLYGON);
    for (int i = 0; i < 360; i++) {
        float theta = i * 3.14159265 / 180;
        glVertex2i(725 + 25 * cos(theta), 550 + 25 * sin(theta));
    }
    glEnd();
}

void drawTree() {
    glColor3f(0.5, 0.2, 0.0);

    glBegin(GL_POLYGON);
    glVertex2i(600, 200);
    glVertex2i(650, 200);
    glVertex2i(650, 300);
    glVertex2i(600, 300);
    glEnd();

    glColor3f(0.0, 0.8, 0.0);

    glBegin(GL_TRIANGLES);
    glVertex2i(550, 300);
    glVertex2i(625, 400);
    glVertex2i(700, 300);
    glEnd();
}

void drawScene() {
```

```

glClear(GL_COLOR_BUFFER_BIT);

if (isDay) {

    glColor3f(0.0, 1.0, 1.0);
    glBegin(GL_POLYGON);
    glVertex2i(0, 0);
    glVertex2i(800, 0);
    glVertex2i(800, 600);
    glVertex2i(0, 600);
    glEnd();

    drawSunOrMoon();
} else {

    glColor3f(0.0, 0.0, 0.1);
    glBegin(GL_POLYGON);
    glVertex2i(0, 0);
    glVertex2i(800, 0);
    glVertex2i(800, 600);
    glVertex2i(0, 600);
    glEnd();

    drawSunOrMoon();
}

drawTree();

glFlush();
}

void keyboardFunc(unsigned char key, int x, int y) {
    switch (key) {
        case 'D':
        case 'd':
            isDay = true;
            break;
        case 'N':
        case 'n':
            isDay = false;
            break;
    }
    glutPostRedisplay();
}

```

```

}

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(800, 600);
    glutCreateWindow("Day and Night with Tree");

    init();
    glutDisplayFunc(drawScene);
    glutKeyboardFunc(keyboardFunc);

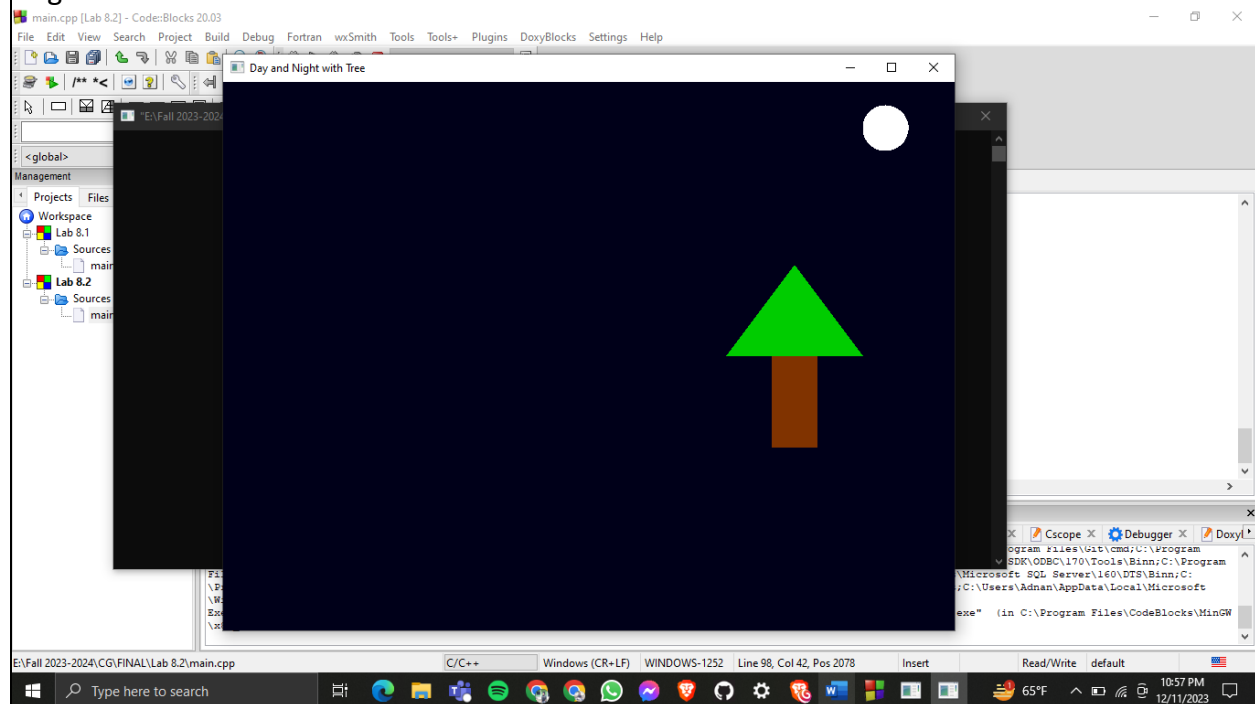
    glutMainLoop();

    return 0;
}

```

Output Screenshot (Full Screen)-

Night:



Day:

