

```

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

float sunPosition = 1.0f; // Sun position, starts from above the horizon (1.0)
float sunSpeed = -0.01f; // Speed of the sun's movement (negative for sunset, positive for sunrise)
float skyColor[] = {0.0f, 0.0f, 0.0f}; // Sky color (RGB), initially black (before sunrise)
float groundLevel = -0.8f; // The position of the ground surface (Y coordinate)
bool isSunset = false; // Flag to track if it's currently sunset
bool isDelay = false; // Flag to track if the 3-second delay is active

// Declare delaySunrise function before using it
void delaySunrise(int value);

void updateSunPosition(int value) {
    // Update the sun's position
    sunPosition += sunSpeed; // Move the sun according to the current speed

    // Handle sunset (sun is falling)
    if (sunPosition < groundLevel && !isDelay) {
        // If the sun is below the ground and no delay is active, change the sky to blue (daytime)
        skyColor[0] = 0.0f;
        skyColor[1] = 0.5f;
        skyColor[2] = 1.0f;

        // After sunset, reverse the direction (start sunrise)
        sunSpeed = 0.01f; // Change the sun's speed to positive for sunrise
        isSunset = true; // Mark that we have reached sunset
        glutTimerFunc(3000, delaySunrise, 0); // Start the delay of 3 seconds before sunrise
    }

    // Handle sunrise (sun is rising)
    if (sunPosition > 1.0f) {
        // If the sun is above the horizon, change the sky to black (night)
        skyColor[0] = 0.0f;
        skyColor[1] = 0.0f;
        skyColor[2] = 0.0f;

        // After sunrise, reverse the direction (start sunset)
        sunSpeed = -0.01f; // Change the sun's speed to negative for sunset
        isSunset = false; // Mark that the sun is rising
    }

    // Redraw the scene
    glutPostRedisplay();
    glutTimerFunc(20, updateSunPosition, 0); // Continuously update the sun position
}

void delaySunrise(int value) {

```

```

// Set the delay to allow the sky to remain black for 3 seconds after sunset
isDelay = false; // Disable delay flag after 3 seconds
glutPostRedisplay(); // Redraw the scene after the delay
}

void drawSun() {
    // Draw the sun as a yellow circle (a simple disk)
    if (sunPosition > groundLevel) { // Only draw the sun if it's above the ground
        glPushMatrix();
        glTranslatef(0.0f, sunPosition, 0.0f); // Move the sun based on its position

        glColor3f(1.0f, 1.0f, 0.0f); // Sun color (yellow)
        glutSolidSphere(0.1, 20, 20); // Draw the sun
        glPopMatrix();
    }
}

void drawGround() {
    // Draw the ground as the top layer
    glEnable(GL_DEPTH_TEST); // Enable depth testing so the ground hides the sun

    glBegin(GL_QUADS);
    glColor3f(0.3f, 0.3f, 0.3f); // Ground color (dark gray)
    glVertex3f(-1.0f, groundLevel, 0.0f);
    glVertex3f(1.0f, groundLevel, 0.0f);
    glVertex3f(1.0f, -1.0f, 0.0f);
    glVertex3f(-1.0f, -1.0f, 0.0f);
    glEnd();

    glDisable(GL_DEPTH_TEST); // Disable depth test after drawing ground (ground must be top layer)
}

void display() {
    // Set background color (sky)
    glClearColor(skyColor[0], skyColor[1], skyColor[2], 1.0f);
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    // Draw the sun and the ground
    drawSun();
    drawGround();

    glutSwapBuffers();
}

void init() {
    glEnable(GL_DEPTH_TEST); // Enable depth testing for 3D effects
    glClearColor(skyColor[0], skyColor[1], skyColor[2], 1.0f); // Initial sky color
}

```

```
int main(int argc, char** argv) {  
    glutInit(&argc, argv);  
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);  
    glutInitWindowSize(800, 600);  
    glutCreateWindow("Sunrise and Sunset Animation");  
  
    init();  
  
    glutDisplayFunc(display);  
    glutTimerFunc(25, updateSunPosition, 0); // Start the sun's movement  
  
    glutMainLoop();  
    return 0;  
}
```