- **Source Code**

- ***cgl6.h***

```
/*
CG Practical - 6
Write OpenGL program to draw sunrise and sunset
g++ -o cgl6 cgl6.cpp -lGL -lGLU -lglut
./cgl6
*/

#include<math.h>
#include<GL/glut.h>
using namespace std;

// Sun Position
float sunX = -0.7;
float sunY = -0.3;
float sunZ = -1.4;

// Color of the Sun
float sunR = 2.0;
float sunG = 1.0;
float sunB = 0.0;

// Background Color
float bgR = 0;
float bgG = 0;
float bgB = 0;

bool flag = true;

void drawSun() {
    glColor3f(sunR, sunG, sunB);
    glTranslatef(sunX, sunY, sunZ);
    glutSolidSphere(0.1, 30, 30);
}

// Chnage the size according to viewport
void reshape(int w, int h) {
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
```

```
    glLoadIdentity();
    double aspect = double(w) / double(h);
    gluPerspective(45, aspect, 0.1, 200);
}

void update(int value) {
    if (flag) {
        sunX += 0.001;
        sunY += 0.0007;
        sunR -= 0.001;
        sunG += 0.002;
        bgB += 0.001;

        if (sunX > 0.3)
            flag = false;
    }
    else {
        sunX += 0.001;
        sunY -= 0.0007;
        sunR += 0.001;
        sunG -= 0.002;
        bgB -= 0.001;

        if (sunX < -0.3)
            flag = true;
    }
    glutPostRedisplay();
    glutTimerFunc(20, update, 0);
}

void display() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glClearColor(bgR, bgG, bgB, 0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    glPushMatrix();
    drawSun();
    glPopMatrix();
    glutSwapBuffers();
}
```

```
int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(500, 500);
    glutCreateWindow("Sunsrise and Sunset");
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutPostRedisplay();
    glutTimerFunc(20, update, 0);
    glutMainLoop();
    return 0;
}
```

- **Output**