SSN COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UCS1712 – GRAPHICS AND MULTIMEDIA LAB
EX NO: 12-Creating a 3D scene using OpenGL

Name : Mohamed Hashim G
RegNo : 185001094

AIM

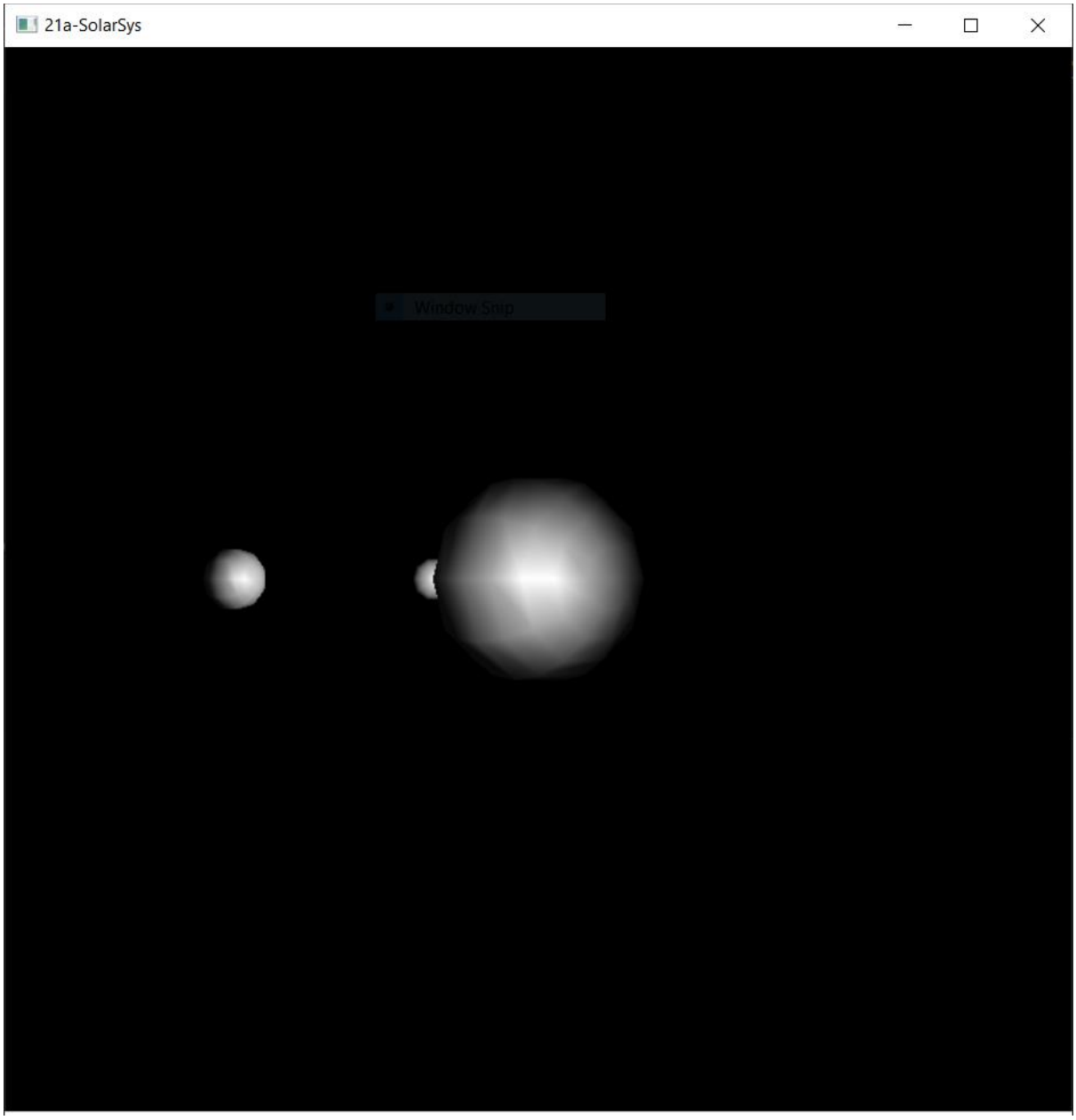
To write a C++ program using OpenGL to draw at least four 3D objects. Apply lighting and texture and render the scene. Apply transformations to create a simple 3D animation

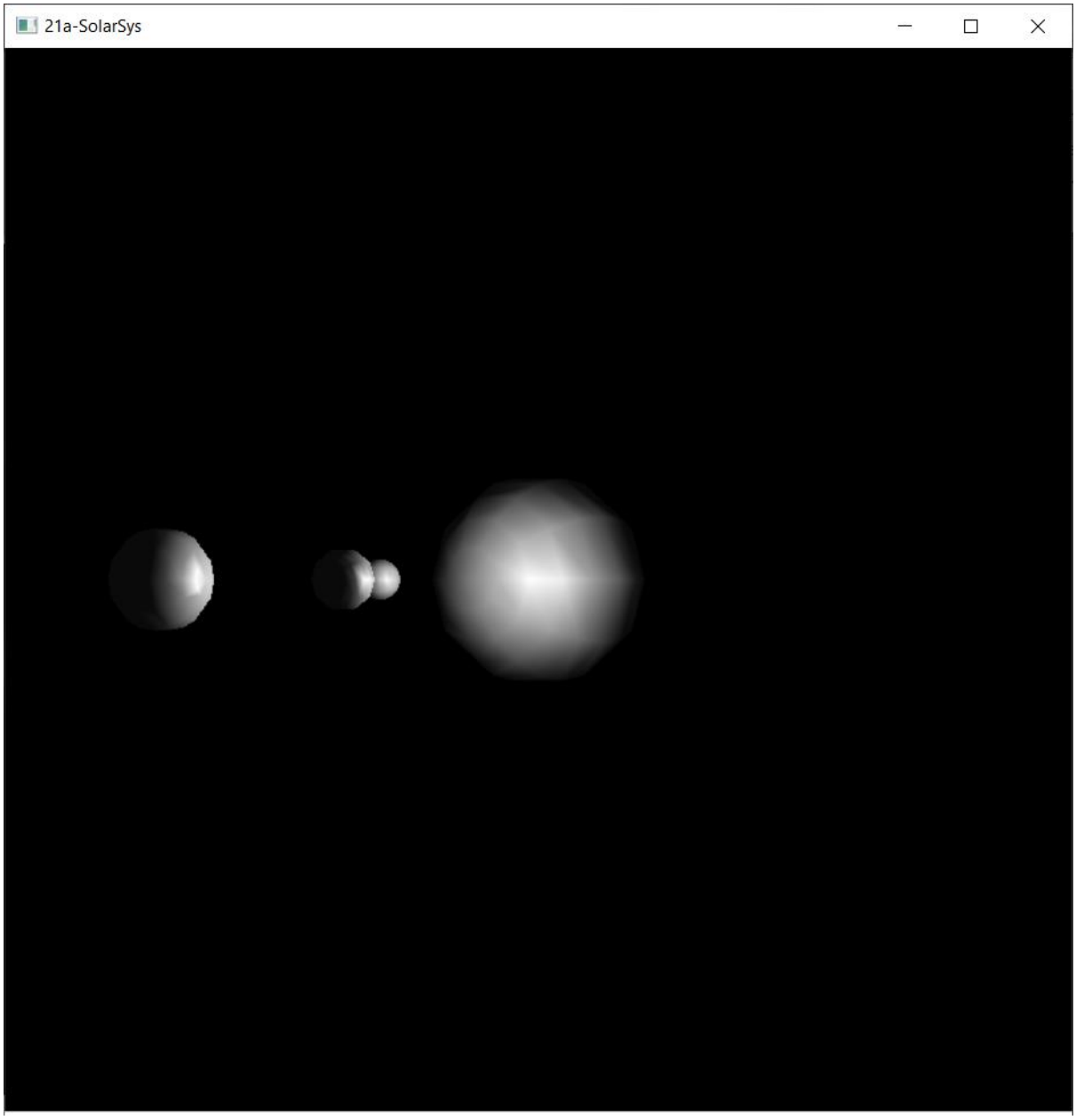
CODE:

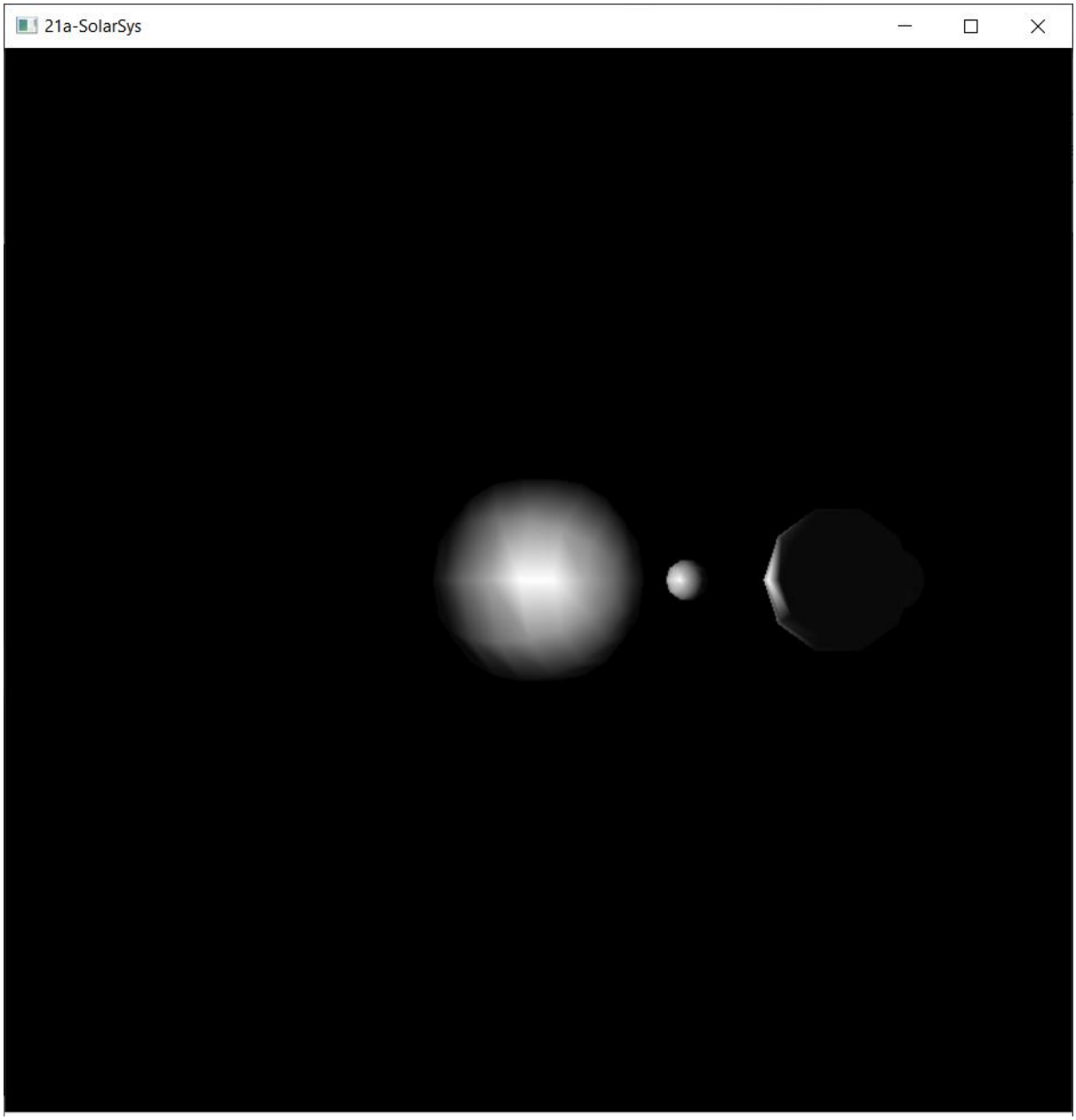
```
#include<gl/glut.h>
#include<iostream>
using namespace std;

int alpha=0;

void init()
{
    glClearColor(0,0,0,1);
    GLfloat mat_specular[] = {1.0, 1.0, 1.0, 1.0};
    GLfloat mat_shininess[] = {50.0};
    GLfloat lpos[]={1.0,1.0,1.0,1.0};

    glShadeModel(GL_SMOOTH);
    glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
    glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);
    glLightfv(GL_LIGHT0, GL_POSITION, lpos);

    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    glEnable(GL_DEPTH_TEST);
}

void disp()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-500, 500, -500, 500, -500, 500);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
```

```
    gluLookAt(0, 0, 300, 0, 0, 0, 0, 1, 0);
    glRotatef(alpha, 0, 1, 0);
    // SUN
    glColor3f(0.5f,0.5f,0.2f);
    glutSolidSphere(100, 10, 10);

    // Mars
    gluLookAt(0, 0, 350, 50, 0, 50, 0, 1, 0);
    glutSolidSphere(30, 10, 10);


    // Earth
    gluLookAt(0, 0, 200, -100, 0, 200, 0, 1, 0);
    glutSolidSphere(50, 10, 10);

    // Jupiter
    gluLookAt(0, 0, -300, -180, 0, 170, 0, 1, 0);
    glutSolidSphere(70, 10, 10);

    //Pluto
    gluLookAt(0, 0, 500, 200, 0, -300, 0, 1, 0);
    glutSolidSphere(20, 10, 10);

    gluLookAt(0, 0, 300, 0, 0, 0, 0, 1, 0);
    glFlush();
}

void timer(int)
{
    alpha++;
    glutPostRedisplay();
    glutTimerFunc(100, timer, 0);
}

int main(int argc, char * argv[])
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(750, 750);
    glutInitWindowPosition(700, 0);
    glutCreateWindow("21a-SolarSys");
    init();
    glutDisplayFunc(disp);
    glutTimerFunc(100, timer, 0);
    glutMainLoop();
    return 0;
}
```

OUTPUT:




21a-SolarSys



RESULT:

Thus an animation using 3D objects has been simulated.