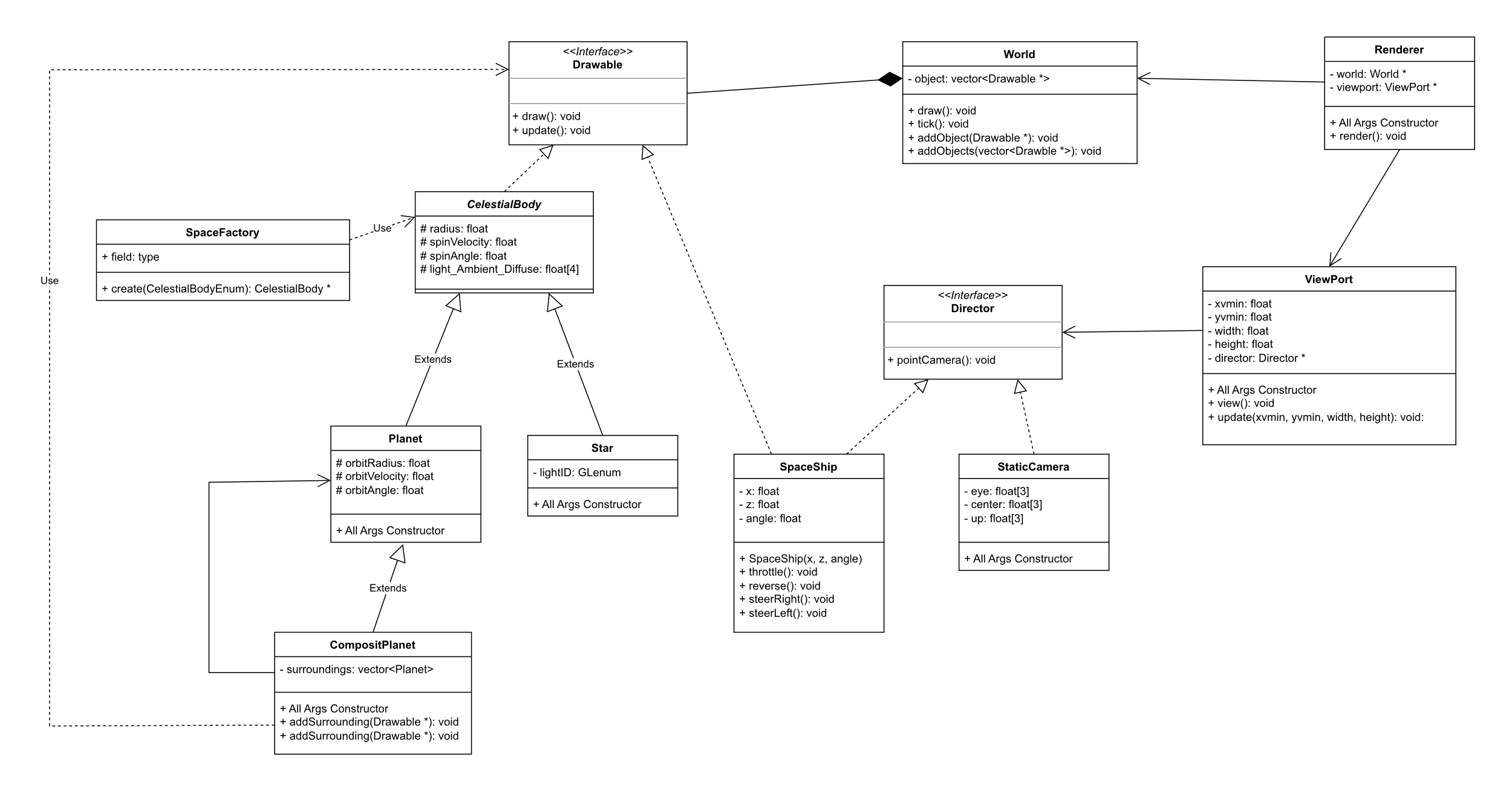
Computer Graphics

Course Project

|  |  |
| --- | --- |
| Karim Alaa MoheyElDin | 19016172 |

# Class Diagram



# Code

## Drawing The Sun

void Star::draw()

{

    glEnable(lightID);

    glLightfv(lightID, GL\_AMBIENT\_AND\_DIFFUSE, light\_Ambient\_Diffuse);

    float w[] = {0.1, 0.1, 0.1, 1};

    glLightfv(lightID, GL\_SPECULAR, w);

    float lightPosition[4] = {0, 0, 0, 1};

    glLightfv(lightID, GL\_POSITION, lightPosition);

    glLightf(lightID, GL\_LINEAR\_ATTENUATION, 0.0001f);

    glLightf(lightID, GL\_QUADRATIC\_ATTENUATION, 0.0001f);

    glMaterialfv(GL\_FRONT, GL\_EMISSION, light\_Ambient\_Diffuse);

    glPushMatrix();

        glRotatef(spinAngle, 0, 1, 0);

        glRotatef(90, 1, 0, 0);

        glutSolidSphere(radius, 40, 40);

    glPopMatrix();

    float reset[4] = {0};

    glMaterialfv(GL\_FRONT, GL\_EMISSION, reset);

}

## Drawing Planets

void Planet::draw(){

    glMaterialfv(GL\_FRONT, GL\_AMBIENT\_AND\_DIFFUSE, light\_Ambient\_Diffuse);

    float w[] = {0.1, 0.1, 0.1, 1};

    glMaterialfv(GL\_FRONT, GL\_SPECULAR, w);

    glPushMatrix();

        glRotatef(orbitAngle, 0, 1, 0);

        glTranslatef(orbitRadius, 0, 0);

        glRotatef(spinAngle, 0, 1, 0);

        glRotatef(90, 1, 0, 0);

        glutSolidSphere(radius, 40, 40);

    glPopMatrix();

    float reset[4] = {0};

    glMaterialfv(GL\_FRONT, GL\_AMBIENT\_AND\_DIFFUSE, reset);

}

## Setup Function

*// Initialization routine.*

void setup(void)

{

    glEnable(GL\_DEPTH\_TEST);

*// Create the world*

    world = new World();

    world->addObject(factory.create(SUN));

    world->addObject(factory.create(MERCURY));

    world->addObject(factory.create(VENUS));

    world->addObject(factory.create(EARTH));

    world->addObject(factory.create(MARS));

    world->addObject(factory.create(JUPYTER));

    world->addObject(factory.create(SATURNE));

    world->addObject(factory.create(URANUS));

    world->addObject(factory.create(NEPTUNE));

*//backgorund*

    float cosmicColor[3] = {0.09, 0.08, 0.43};

    world->addObject(new Planet(400, 0, 0, 0, 0, 0, cosmicColor));

*//add spaceship*

    spaceship = new SpaceShip(-100, 100, 45);

    world->addObject(spaceship);

*// Create the spaceship viewport and renderer*

    spaceshipViewPort = new ViewPort(0, 0, width, height, \*spaceship);

    spaceshipRenderer = new Renderer(world, spaceshipViewPort);

*// Create the fixed viewport and renderer*

    double eye[3] = {0, 100, 0}, center[3]= {0, 0, 0}, up[3] = {0, 0, -1};

    fixedCamera = new StaticCamera (eye, center, up);

    fixedViewPort = new ViewPort(width \* 2 / 3, 0, width / 3, height / 3, \*fixedCamera);

    fixedRenderer = new Renderer(world, fixedViewPort);

    glEnable(GL\_LIGHTING);

    glLightModelfv(GL\_LIGHT\_MODEL\_AMBIENT, globAmb);

    glEnable(GL\_DEPTH\_TEST);

    glClearColor(0.0, 0.0, 0.0, 0.0);

    glutTimerFunc(0, update, 0); *// Initial call of update().*

}

## Draw Function

*// Drawing routine.*

void drawScene(void)

{

    glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

    spaceshipRenderer->render();

    fixedRenderer->render();

    glutSwapBuffers();

}

# A screenshot of a computer Description automatically generatedA screenshot of a computer Description automatically generatedScreen Shots

A screenshot of a computer

Description automatically generatedA screenshot of a computer screen

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

# Compilation

Make sure to include all the .h and .cpp files.

# Additional Resources

<https://www.youtube.com/playlist?list=PLlrATfBNZ98foTJPJ_Ev03o2oq3-GGOS2>

<https://groups.google.com/g/comp.graphics.api.opengl/c/v1HgahK2kyY>

<https://just.edu.jo/~yaser/courses/cs480/Tutorials/OpenGl%20-%20Chapter%208%20%20Light%20&%20Material.htm>