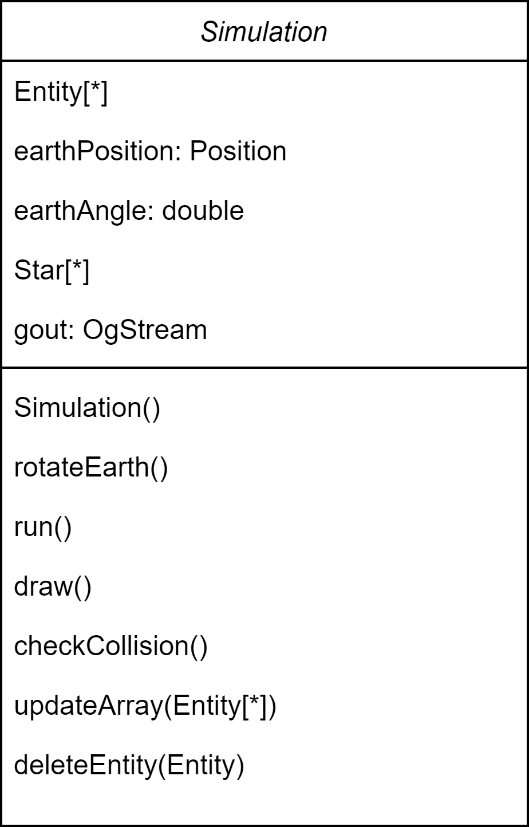
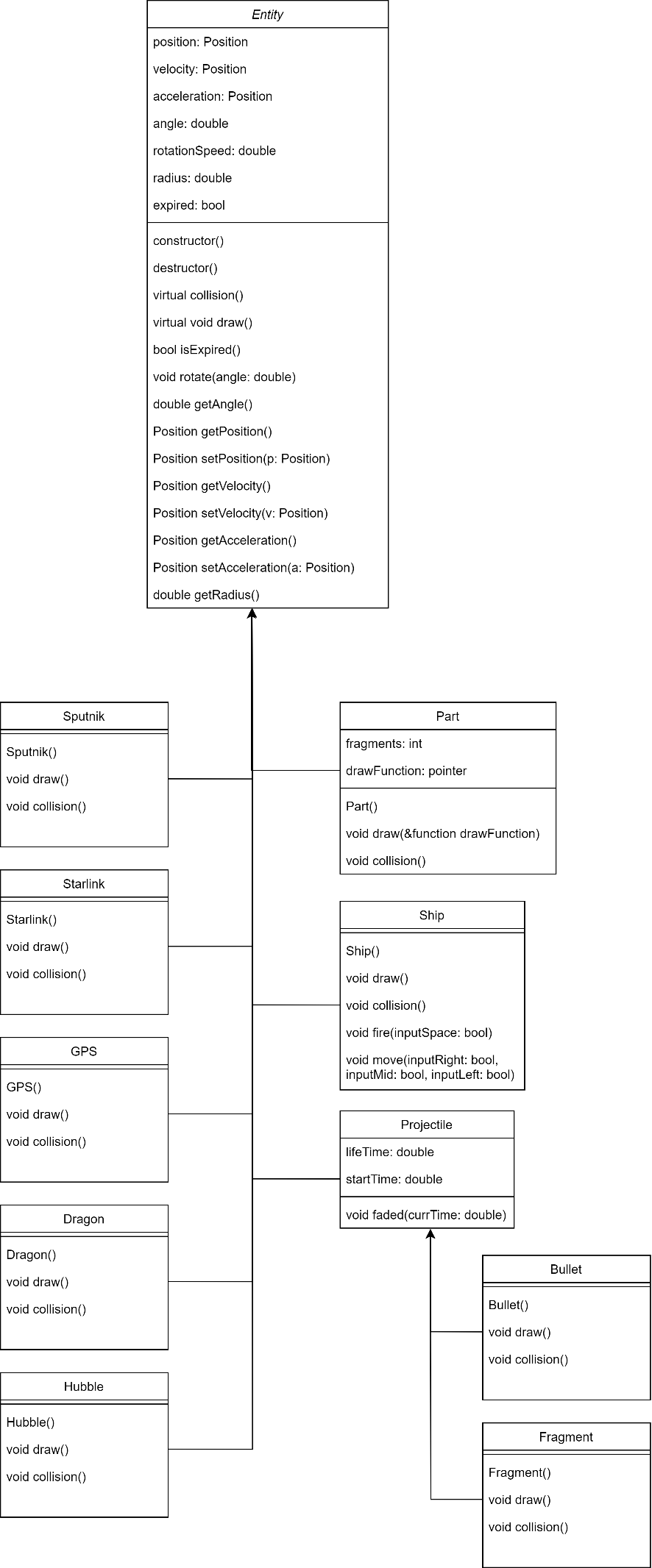
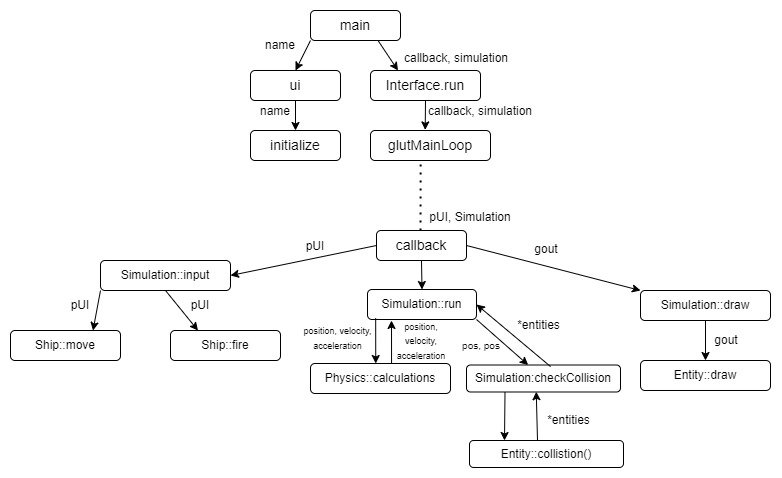
Lab 08: Orbit Design

**UML Class Diagrams**





**Structure Chart**



**Pseudocode**

Physics::calculations(position, velocity, acceleration)

h <- sqrt(pow(position.getMetersX(), 2.0) +

pow(position.getMetersY(), 2.0 - R)

g <- G \* pow((R / (R + h)), 2.0)

d <- atan2(position.getMetersY(), position.getMetersX())

acceleration.setX(-(g \* sin(d))

acceleration.setY(g \* cos(d))

velocity.setX(velocity.getMetersX() + acceleration.getMetersX()

+ 48.0)

velocity.setY(velocity.getMetersY() + acceleration.getMetersY()

+ 48.0)

x <- position.getMetersX() + velocity.getMetersX() \* 48.0 + (0.5

\* acceleration.getMetersX()) \* pow(48.0,2))

y <- position.getMetersY() + velocity.getMetersY() \* 48.0 + (0.5

\* acceleration.getMetersY()) \* pow(48.0,2))

RETURN x,y

END

Simulation::run

FOR i <- 0 ... entities.Count

position <- entities[i].getPosition()

velocity <- entities[i].getVelocity()

acceleration <- entities[i].getAcceleration()

x,y <-physicsEngine.calculations(position, velocity,

acceleration)

entities[i].setPosition(x,y)

FOR i <- 0 ... entities.Count - 1

crashed <- checkCollision(entities[i].getPosition(),

entities[i+1].getPosition())

IF crashed

Entities.append(entities[i+1].collision())

END

**Test Cases**

Ship Test Cases:

1 – contructor

Setup:

Ship()

Exercise:

Verify:

Ship->getAngle() ==

Ship->getAcceleration() == Position(0.0, 0.0)

Ship->getPosition() == Position(-450px, 450px)

Ship->getVelocity() == Position(0.0, -2000.0)

Ship->getRadius() == 10.0

2 - Left

Setup:

Ship()

Exercise:

Ship->move(true, false, false)

Verify:

Ship->getAngle() ==

3 - Right

Setup:

Ship()

Exercise:

Ship->move(false, true, false)

Verify:

Ship->getAngle() ==

4 - Down

Setup:

Ship()

Exercise:

Ship->move(false, false, true)

Verify:

Ship->getAcceleration() == Position(2.0,0.0)

5 - Fire

Setup:

Ship()

Exercise:

Bullets = Ship->fire(true)

Verify:

Bullets != []

6 – double turn

Setup:

Ship()

Exercise:

Ship()->move(true, true, false)

Verify:

Ship->getAngle() ==

Physics Test Cases

1 – GPS First Callback Loop

Setup:

double x;

double y;

Position pos = Position(0.0,42164000.0);

Position vel = Position(-3100,0.0);

Position acceleration = Position(0,0);

Physics engine;

Exercise:

x,y = engine.calculations(pos, vel, acceleration);

Verify:

assert(x == -148800);

assert(y == 42163200);

2 – GPS Second Callback Loop

Setup:

double x;

double y;

Position pos = Position(-148800.0,42163200);

Position vel = Position(-3100,-10.7708);

Position acceleration = Position(0,0);

Physics engine;

Exercise:

x,y = engine.calculations(pos, vel, acceleration);

Verify:

assert(x == -297597);

assert(y == 42161900);

3 – GPS Third Callback Loop

Setup:

double x;

double y;

Position pos = Position(-297597, 42161900);

Position vel = Position(-3099.96,-21.5415);

Position acceleration = Position(0,0);

Physics engine;

Exercise:

x,y = engine.calculations(pos, vel, acceleration);

Verify:

assert(x == -446390);

assert(y == 42160100);

4 – GPS Fourth Callback Loop

Setup:

double x;

double y;

Position pos = Position(-446390, 42160100);

Position vel = Position(-3099.89,-32.312);

Position acceleration = Position(0,0);

Physics engine;

Exercise:

x,y = engine.calculations(pos, vel, acceleration);

Verify:

assert(x == -595176);

assert(y == 42157800);

5 – GPS Fifth Callback Loop

Setup:

double x;

double y;

Position pos = Position(-595176, 42157800);

Position vel = Position(--3099.77,-43.0822);

Position acceleration = Position(0,0);

Physics engine;

Exercise:

x,y = engine.calculations(pos, vel, acceleration);

Verify:

assert(x == -743954);

assert(y == 42155000);

6 – GPS Sixth Callback Loop

Setup:

double x;

double y;

Position pos = Position(-743954, 42155000);

Position vel = Position(--3099.62, -53.8519);

Position acceleration = Position(0,0);

Physics engine;

Exercise:

x,y = engine.calculations(pos, vel, acceleration);

Verify:

assert(x == -892722);

assert(y == 42151600);