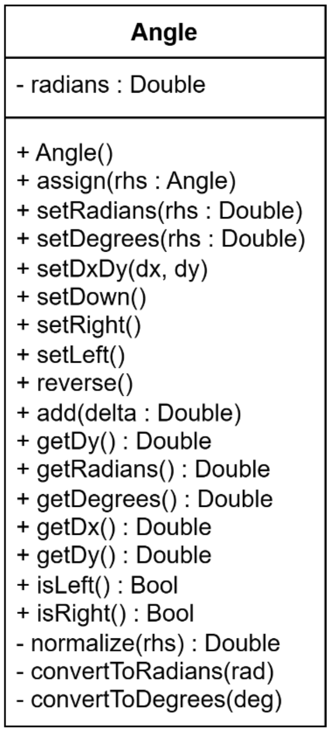
Artillery Design

Class Diagrams

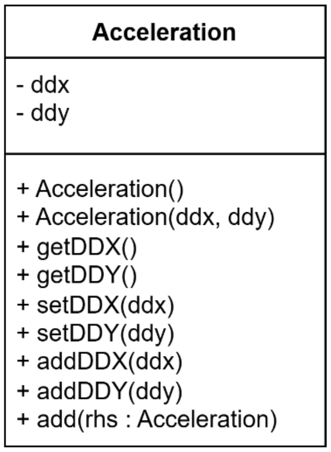
Angle

Everything we need to know about an angle.



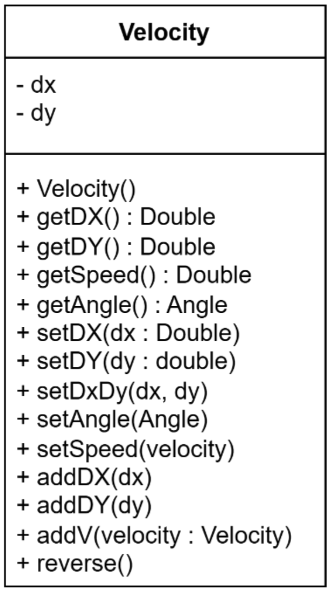
Acceleration

Everything we need to know about acceleration.



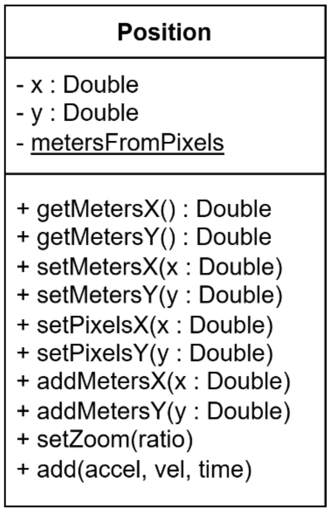
Velocity

Everything we need to know about speed.



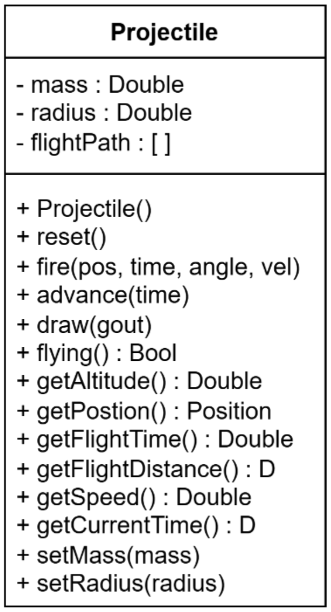
Position

Everything we need to know about the position.



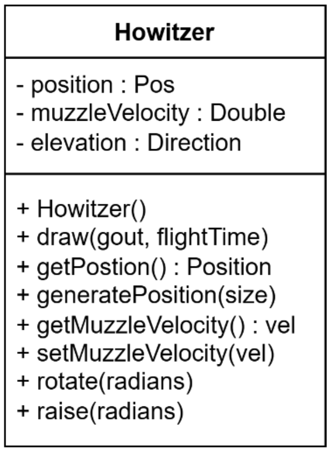
Projectile

A projectile, including how it flies and where it is located.



Howitzer

The gun, including where it is located and where it is pointed.



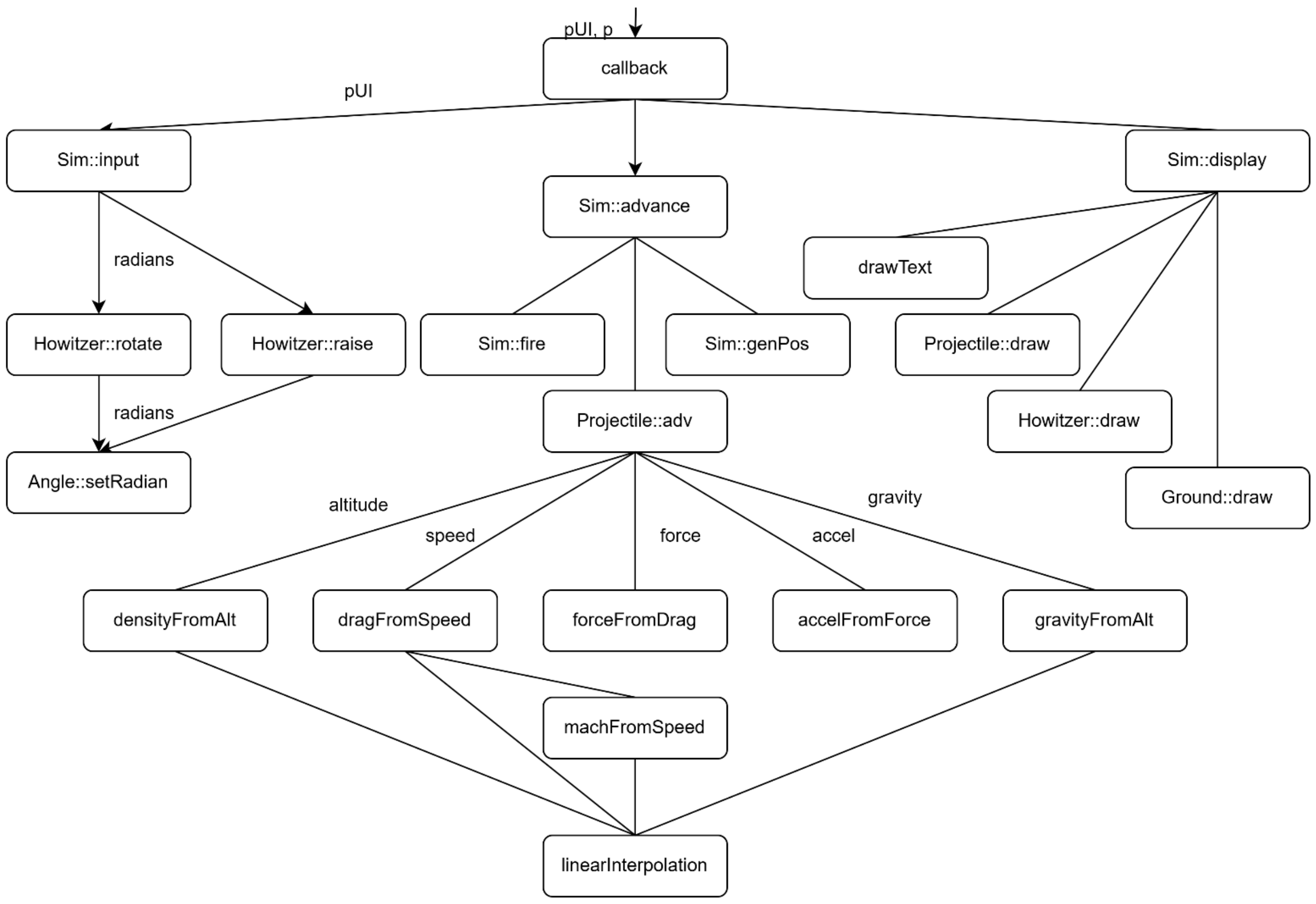
Simulator

This connects all the elements together. It does not know how the projectile flies or anything about the gun, but it does control the relationship between those elements.

Simulator
- interval : Double - howitzer : Howitzer - projectile : Projectile - status : status - simTime : Double - ground : Ground

+ Simulation(...)
+ reset()
+ fire()
+ display()
+ advance()
+ input(Interface)
+ setInterval(interval)
+ setMuzzleVelocity(vel)
+ setDiameter(diameter)
- hitTarget()
- getHeightMeters()

Structure Chart



Pseudocode

The main function to handle all the physics is called in `Projectile::advance()`. This moves the projectile along the path taking into account all the forces which act on it.

```
Projectile.advance()
   Pvt ← flightpath.back()
   Speed ← pvt.v.getSpeed()
   Altitude ← pvt.pt.getMetersY()

   MODIFY VELOCITY TO HANDLE WIND RESISTANCE
   Density ← densityFromAltitude(altitude)
   dragCoefficient ← dragFromSpeed(speed, altitude)
   windResitance ← forceFromDrag(density, dragCoefficient, radius, speed)
   accelerationDrag ← accelerationFromForce(windResistance, mass)
   velocityWind ← velocityFromAcceleration(accelerationDrag, interval),
                    pvt.v.getDirection()
   velocityWind.reverse()
   pvt.v += velocityWind

   MODIFY VELOCITY TO HANDLE GRAVITY
   accelerationGravity ← gravityFromAltitude(altitude)
   velocityGravity ← velocityFromAcceleration(accelerationGravity, interval)
                       Angle.setDown()
   Pvt.v += velocityGravity

   INTERTIA
   Pvt.pt.addMetersX(velocityFromAcceleration(pvt.v.getDX(), interval)
   Pvt.pt.addMetersY(velocityFromAcceleration(pvt.v.getDY(), interval)

   ADD IT TO THE BACK OF THE FLIGHT PATH
   Flightpath.push_back(pvt);
```

The callback function does three things: handle input, performs processing, and handles output.

```
Callback(UI, sim)
   Sim.input(UI)
   Sim.advance()
   Sim.display()
```