Lab 08 Artillery Design

# Class Diagrams:

Simulator:

The class has a fidelity of complete because it completely matches and represents the needs of the design concern.

Robustness level is fragile because it is currently not tested only reviewed by two engineers.

Convenience is seamless because the client needs only call one function Simulator :: Run to use the class.

Abstraction is complete because no implementation details are leaked to the client, everything is wrapped in simulator :: run.

Simulator

+Simulator(ptUpperRight: Position)

+Run(pUI: interface, gout: ogstream)

-doInput(pUI: interface)

-update()

-draw(gout: ogstream)

-getKeys() : [bool]

-applyKeys(keys : [bool])

-getDrag() : double

-getGravity() : double

-checkCollisions()

-drawText(gout: ogstream)

-ptUpperRight: Position

-howitzer: Howitzer

-projectile: Projectile

-ground: Ground

-physics: Physics

- time: double

-keysIndex: enum

Ground:

The class has a fidelity of complete because it completely matches and represents the needs of the design concern.

Robustness level is fragile because it is currently not tested only reviewed by two engineers.

Convenience is seamless because all the needs of the ground class are handled and data is returned in a valid and easy to use state to the client.

Abstraction is complete because there are no implementation details revealed to the client as we provide all necessary methods the client needs.

Ground

-ground: [double]

-posUpperRight: Position

-posTarget: Position

-iHowitzerLocation: Int

+Ground(posUpperRight: Position)

+reset(posHowitzer: Position)

+draw(gout: ogstream)

+hitGround(posProjectile: position) : Bool

+hitTarget(posProjectile: position) : Bool

+getElevationMeters(pos: position) : Double

-getTarget() : Position

## Look Ups:

The class has a fidelity of complete because it completely matches and represents the needs of the design concern and nothing extra is done.

Robustness level is fragile because it is currently not tested only reviewed by two engineers.

Convenience is seamless because all the extra work to interpolate and is done by the class and returned conveniently to the client.

Abstraction is complete because the client does not know anything about how the values are stored or found. They only know that the interface returns a tuple of long doubles.

Look Up

+searchTable() : (long double)

-interpolate(x0: long double, y0: long double, x1: long double, y1: long double, current\_x: long double) : long double

-*table : [(double, double)]*

Gravity Look Up

-*table : [(double, double)]*

Air Density Look Up

-*table : [(double, double)]*

Mach Look Up

-*table : [(double, double)]*

Drag Look Up

-*table : [(double, double)]*

# Structure Chart:

The whole structure chart

A picture containing text, whiteboard

Description automatically generated

Simulator :: doInput

Diagram

Description automatically generated

Simulator :: update

Diagram

Description automatically generated

Simulator :: draw

Diagram

Description automatically generated

# Pseudocode:

SearchTable(notContainedKey, lookUpTable)

iterator 🡨 lookUpTable [0]

high 🡨 lookUpTable [0][0]

low 🡨 lookUpTable [0][0]

searching 🡨 true

WHILE iterator != lookUpTable[-1] and searching = true

IF iterator[0] < notContainedKey

IF iterator[0] > low

low 🡨 iterator

ELSE

high 🡨 iterator[0]

searching 🡨 false

iterator++

keys 🡨 (low, high)

return keys