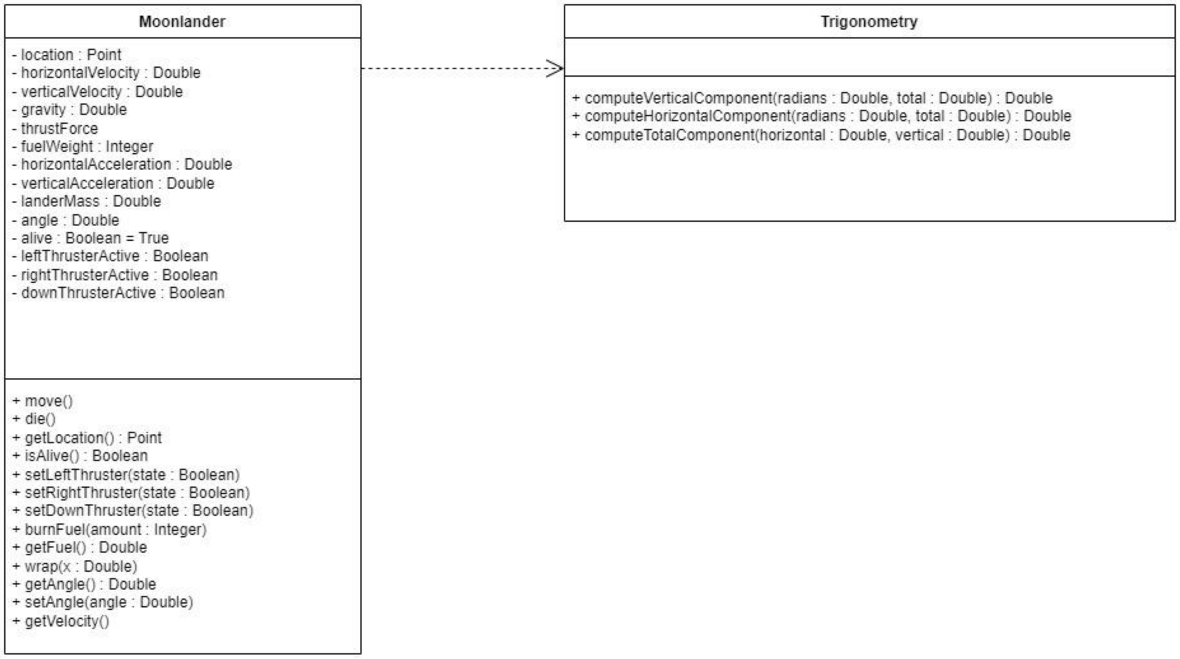
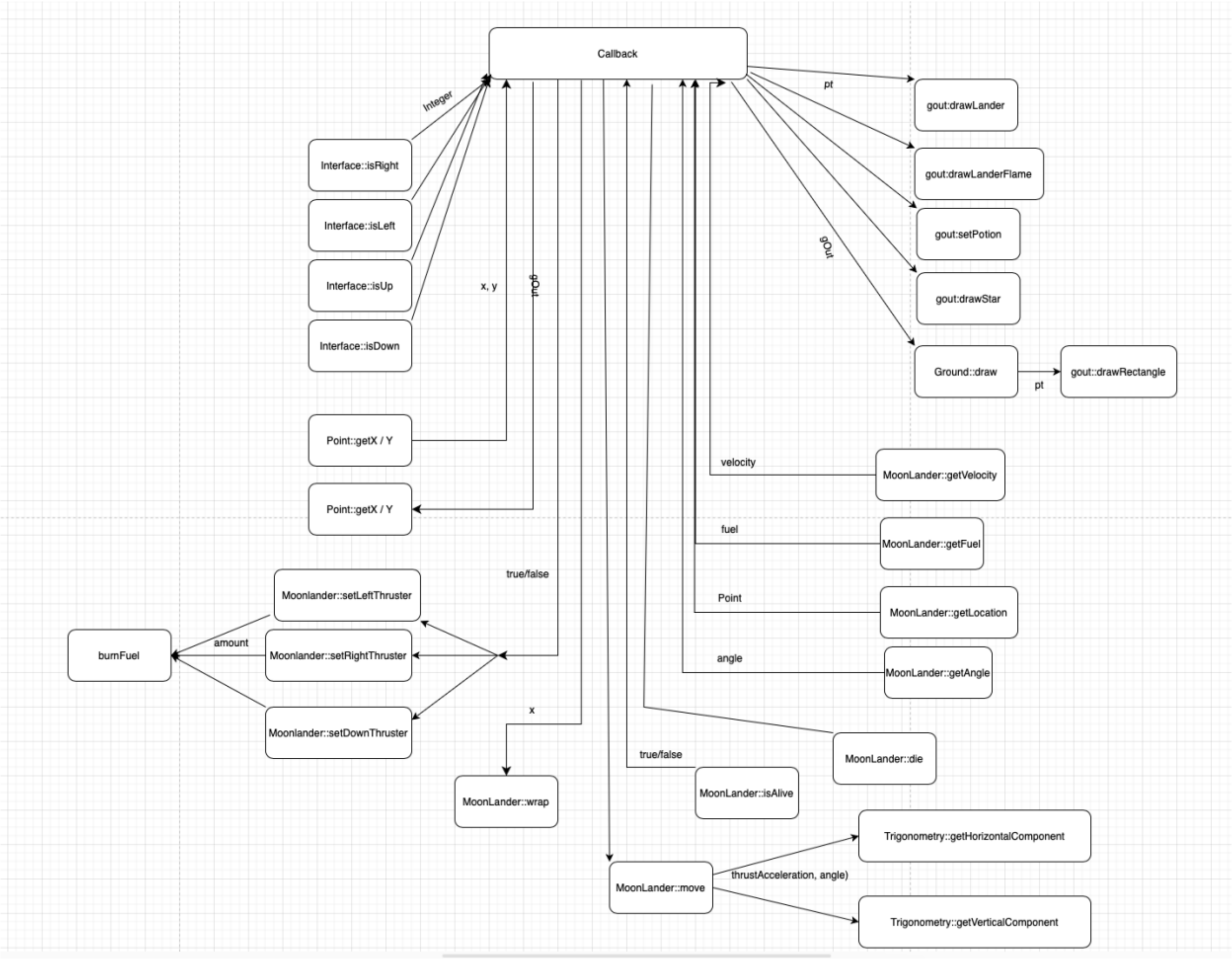
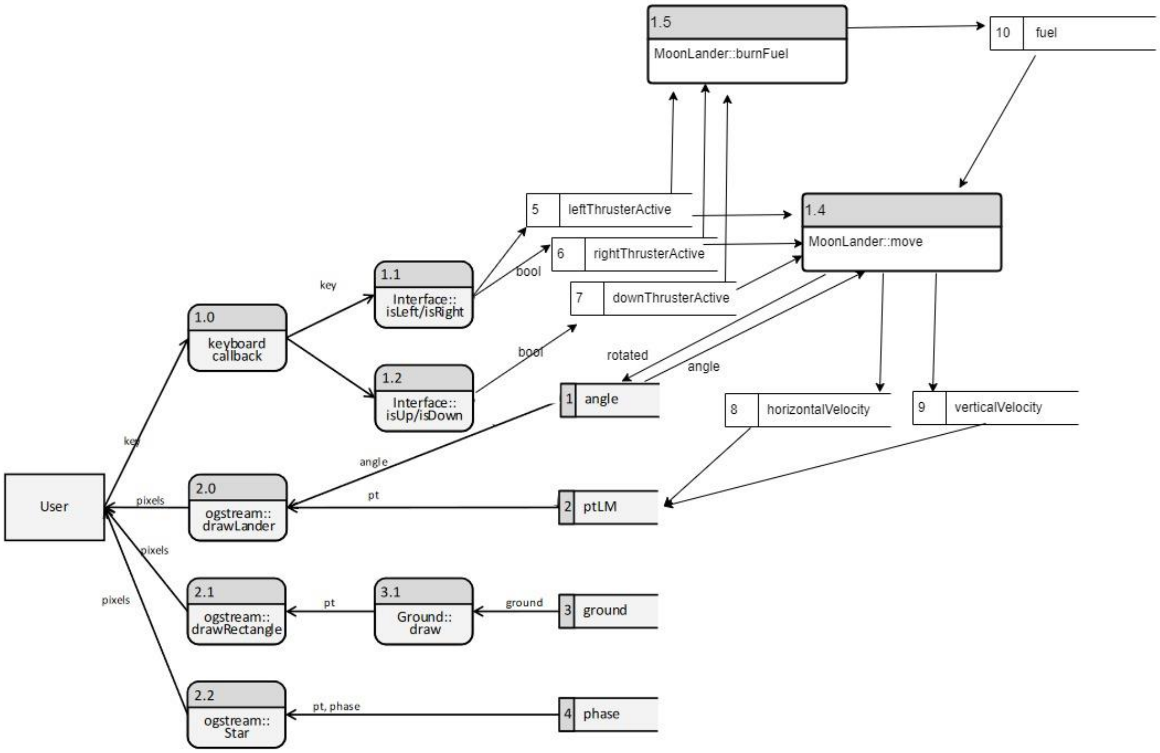
Class Diagram:


Moonlander
- location : Point
- horizontalVelocity : Double
- verticalVelocity : Double
- gravity : Double
- thrustForce
- fuelWeight : Integer
- horizontalAcceleration : Double
- verticalAcceleration : Double
- landerMass : Double
- angle : Double
- alive : Boolean = True
- leftThrusterActive : Boolean
- rightThrusterActive : Boolean
- downThrusterActive : Boolean
+ move()
+ die()
+ getLocation() : Point
+ isAlive() : Boolean
+ setLeftThruster(state : Boolean)
+ setRightThruster(state : Boolean)
+ setDownThruster(state : Boolean)
+ burnFuel(amount : Integer)
+ getFuel() : Double
+ wrap(x : Double)
+ getAngle() : Double
+ setAngle(angle : Double)
+ getVelocity()
Trigonometry
+ computeVerticalComponent(radians : Double, total : Double) : Double
+ computeHorizontalComponent(radians : Double, total : Double) : Double
+ computeTotalComponent(horizontal : Double, vertical : Double) : Double

Structure Chart:


Callback
Integer
Interface::isRight
Interface::isLeft
Interface::isUp
Interface::isDown
x, y
gOut
Point::getX / Y
Point::getX / Y
pt
gout:drawLander
gout:drawLanderFlame
gout:setPotion
gout:drawStar
gOut
Ground::draw
pt
gout::drawRectangle
velocity
MoonLander::getVelocity
fuel
MoonLander::getFuel
Point
MoonLander::getLocation
angle
MoonLander::getAngle
true/false
Moonlander::setLeftThruster
burnFuel
amount
Moonlander::setRightThruster
Moonlander::setDownThruster
x
MoonLander::wrap
true/false
MoonLander::isAlive
MoonLander::die
Trigonometry::getHorizontalComponent
MoonLander::move
thrustAcceleration, angle)
Trigonometry::getVerticalComponent

DFD:



Pseudocode:

```
move()
      IF get_fuel() > 0
      if left thruster
          angle <-- angle + 0.1
          horizontalVelocity <-- horizontalVelocity + 0.1

      if right thruster
          angle <-- angle - 0.1
          horizontalVelocity <-- horizontalVelocity - 0.1

      if down thruster
          thrustAcceleration <-- thrustForce / (landerMass + fuelWeight)
          horizontalVelocity <-- horizontalVelocity +
      Trigonometry::getHorizontalComponent(thrustAcceleration, angle)
```

```
        verticalVelocity <-- verticalVelocity  +
    Trigonometry::getVerticalComponent(thrustAcceleration, angle)

    verticalVelocity <-- verticalVelocity - gravity

    location.addX(horizontalVelocity)
    location.addY(verticalVelocity)
```

```
setLeftThruster(state: Boolean)
    IF get_fuel() > 0
        leftThrusterActive = state
        IF state
        burnFuel(1)
```

Test Cases

Test cases for MoonLander::wrap:

Name	Input	Output
Left	X = -1	X = 200
Right	X = 201	X = 0
Middle	X = 100	X = 100
Far Right	X = -5	X = 200
Far Left	X = 262	X = 0
On Edge	X = 0	X = 0

Test cases for MoonLander::isAlive

Name	Input	Output
Dead #1	Ground: [5,7,11,9,10], Location: [2,10]	False
Dead #2	Ground: [6,7,10,8,14,16] Location: [1, 7]	False
Alive #1	Ground: [4,11,8,10,9] Location: [4,15]	True
Alive #2	Ground: [12,20,13,15,17]	True

	Location: [0,13]	
Alive #3	Ground: [14,20,18,15,13] Location: [3,21]	True