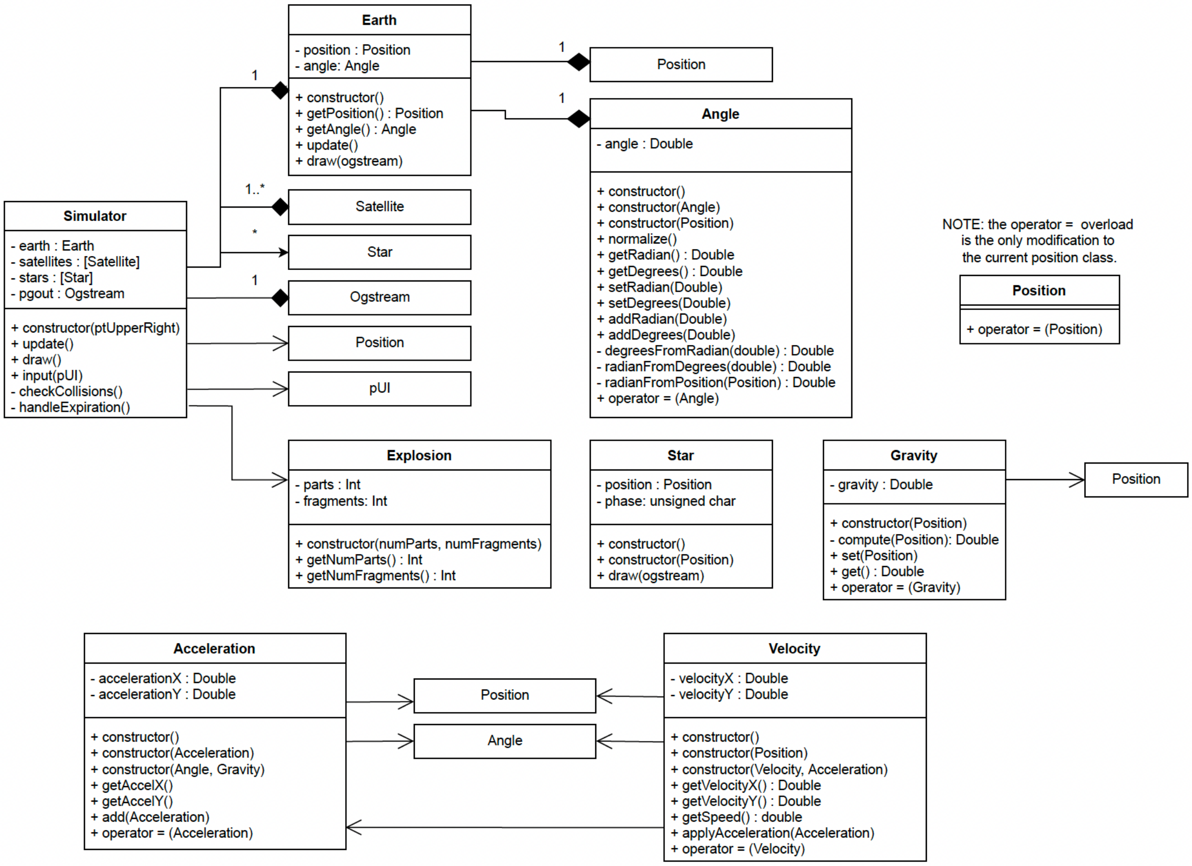
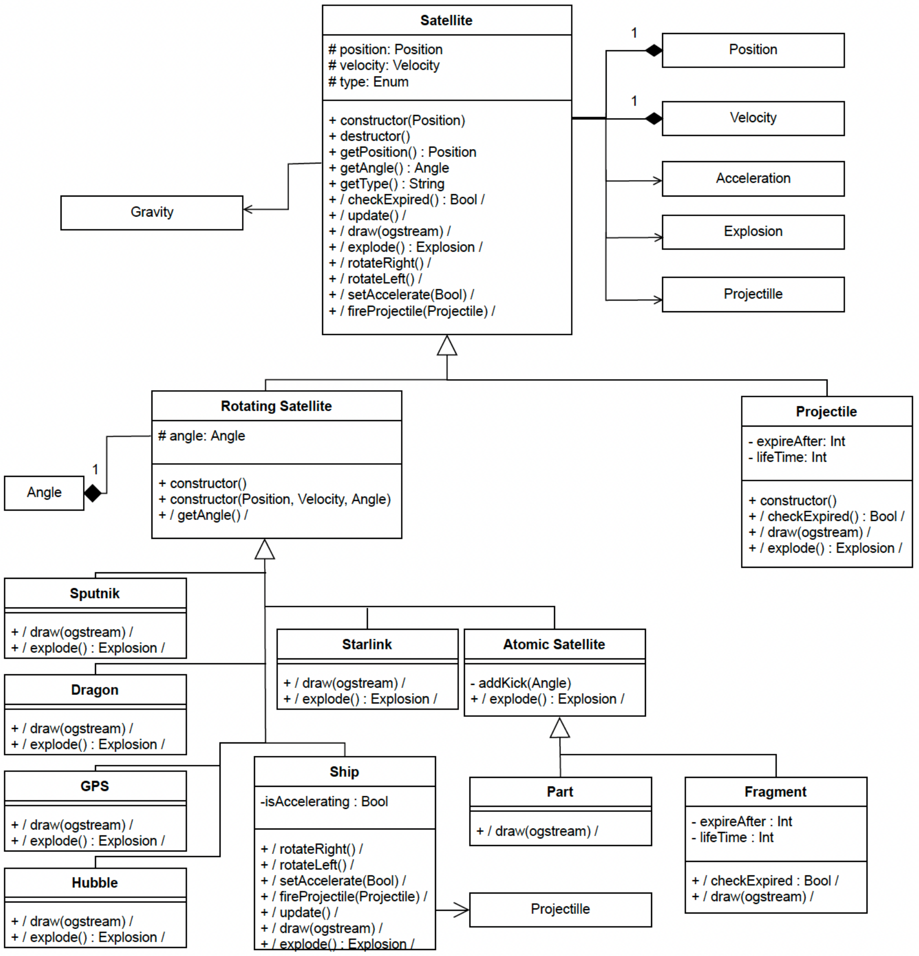
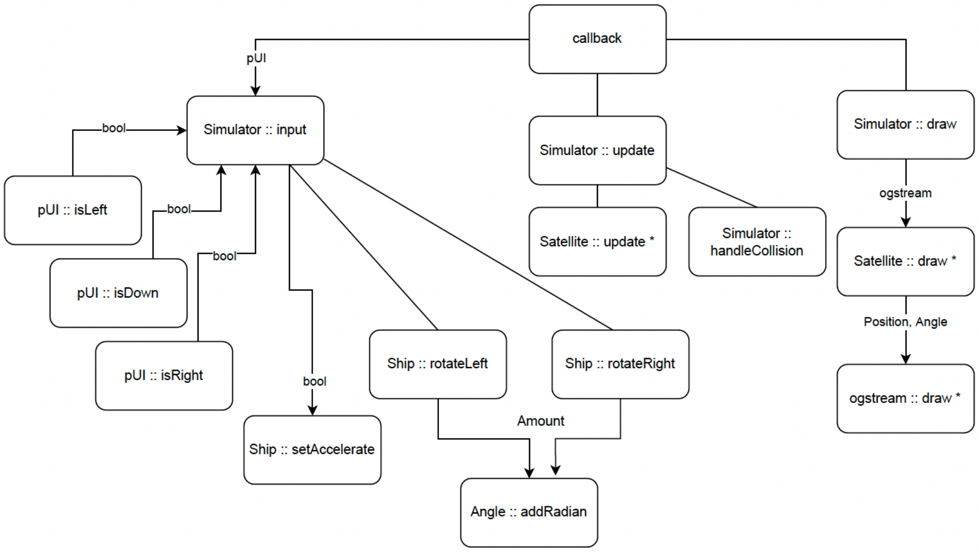
Lab 08: Orbital Design

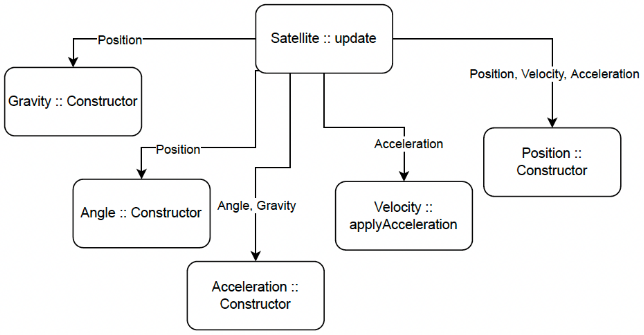
Class Diagrams

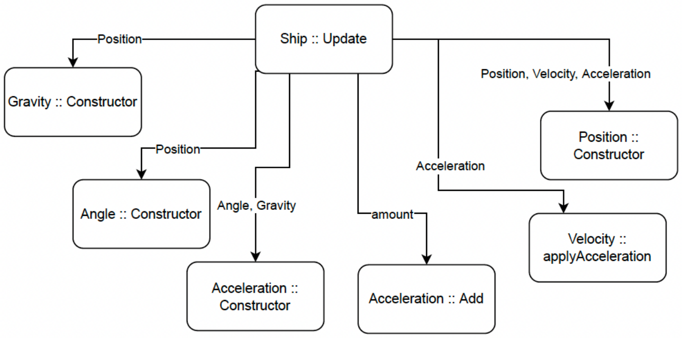


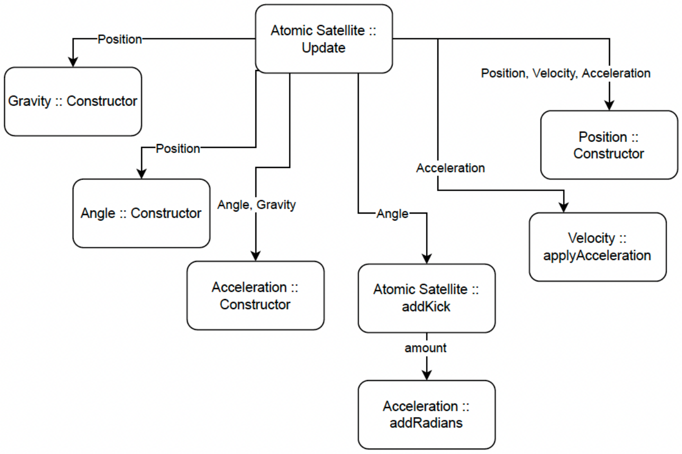

Satellite
# position: Position
# velocity: Velocity
# type: Enum
+ constructor(Position)
+ destructor()
+ getPosition() : Position
+ getAngle() : Angle
+ getType() : String
+ / checkExpired() : Bool /
+ / update() /
+ / draw(ogstream) /
+ / explode() : Explosion /
+ / rotateRight() /
+ / rotateLeft() /
+ / setAccelerate(Bool) /
+ / fireProjectile(Projectile) /
Gravity
1
Position
1
Velocity
Acceleration
Explosion
Projectille
Rotating Satellite
# angle: Angle
+ constructor()
+ constructor(Position, Velocity, Angle)
+ / getAngle() /
1
Angle
Projectile
- expireAfter: Int
- lifeTime: Int
+ constructor()
+ / checkExpired() : Bool /
+ / draw(ogstream) /
+ / explode() : Explosion /
Sputnik
+ / draw(ogstream) /
+ / explode() : Explosion /
Dragon
+ / draw(ogstream) /
+ / explode() : Explosion /
GPS
+ / draw(ogstream) /
+ / explode() : Explosion /
Hubble
+ / draw(ogstream) /
+ / explode() : Explosion /
Starlink
+ / draw(ogstream) /
+ / explode() : Explosion /
Atomic Satellite
- addKick(Angle)
+ / explode() : Explosion /
Ship
-isAccelerating : Bool
+ / rotateRight() /
+ / rotateLeft() /
+ / setAccelerate(Bool) /
+ / fireProjectile(Projectile) /
+ / update() /
+ / draw(ogstream) /
+ / explode() : Explosion /
Projectille
Part
+ / draw(ogstream) /
Fragment
- expireAfter : Int
- lifeTime : Int
+ / checkExpired : Bool /
+ / draw(ogstream) /

Structure Charts

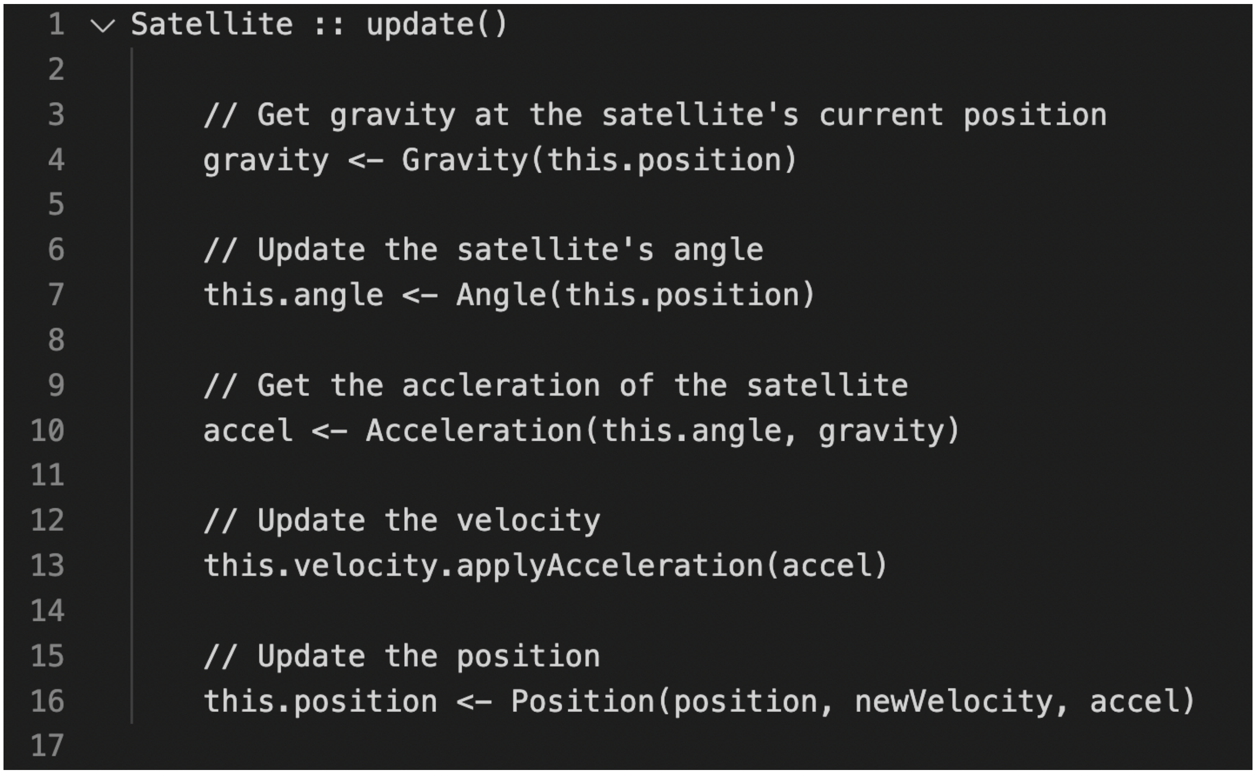





Ship :: Update
Position
Gravity :: Constructor
Position
Angle :: Constructor
Angle, Gravity
Acceleration :: Constructor
amount
Acceleration :: Add
Acceleration
Velocity :: applyAcceleration
Position, Velocity, Acceleration
Position :: Constructor

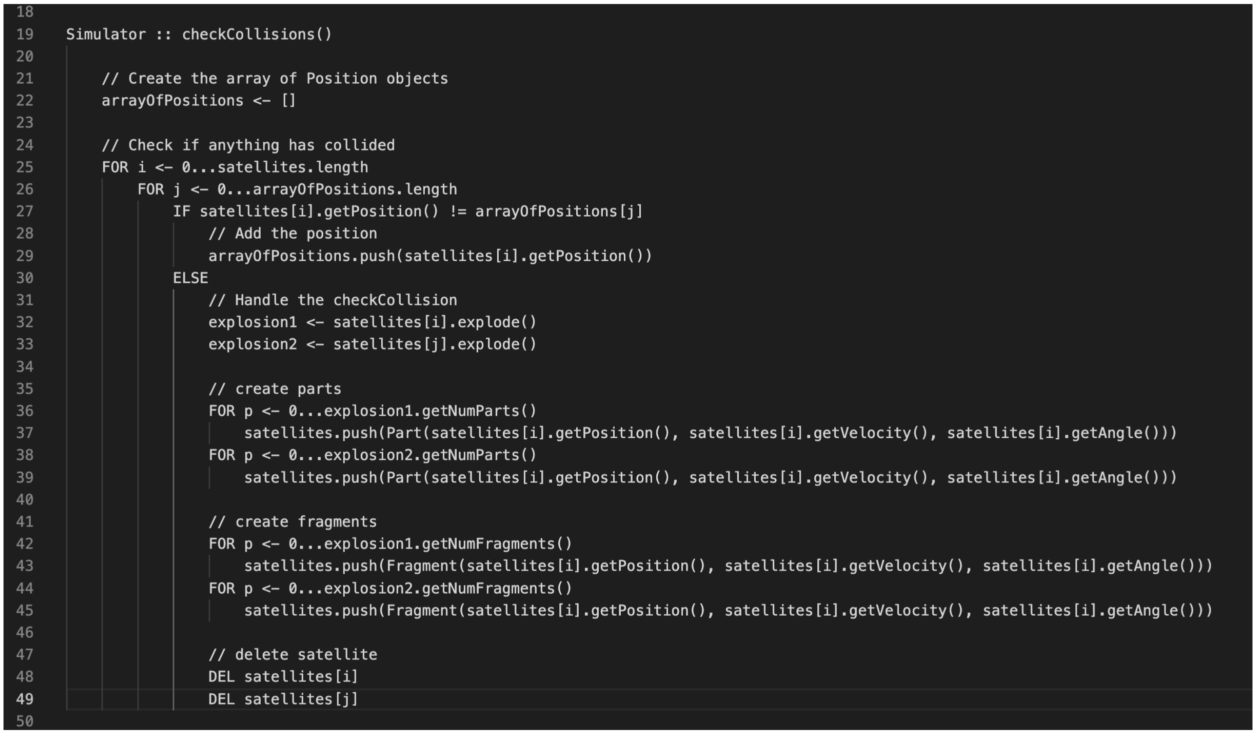

Atomic Satellite :: Update
Position
Gravity :: Constructor
Position
Angle :: Constructor
Angle, Gravity
Acceleration :: Constructor
Angle
Atomic Satellite :: addKick
amount
Acceleration :: addRadians
Acceleration
Velocity :: applyAcceleration
Position, Velocity, Acceleration
Position :: Constructor

Pseudocode:



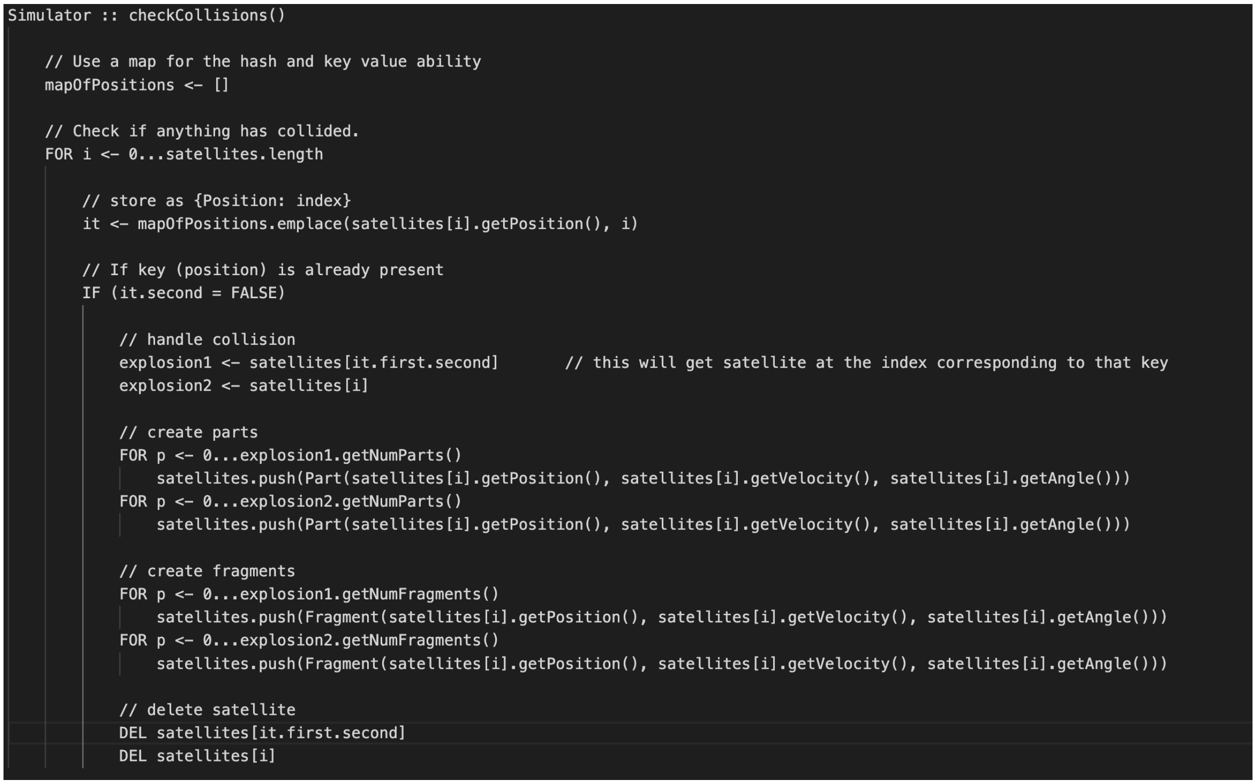
```
Satellite :: update()

    // Get gravity at the satellite's current position
    gravity <- Gravity(this.position)

    // Update the satellite's angle
    this.angle <- Angle(this.position)

    // Get the accleration of the satellite
    accel <- Acceleration(this.angle, gravity)

    // Update the velocity
    this.velocity.applyAcceleration(accel)

    // Update the position
    this.position <- Position(position, newVelocity, accel)

```

Check Collisions Solution 1:



```

Simulator :: checkCollisions()

    // Create the array of Position objects
    arrayOfPositions <- []

    // Check if anything has collided
    FOR i <- 0...satellites.length
        FOR j <- 0...arrayOfPositions.length
            IF satellites[i].getPosition() != arrayOfPositions[j]
                // Add the position
                arrayOfPositions.push(satellites[i].getPosition())
            ELSE
                // Handle the checkCollision
                explosion1 <- satellites[i].explode()
                explosion2 <- satellites[j].explode()

                // create parts
                FOR p <- 0...explosion1.getNumParts()
                    satellites.push(Part(satellites[i].getPosition(), satellites[i].getVelocity(), satellites[i].getAngle()))
                FOR p <- 0...explosion2.getNumParts()
                    satellites.push(Part(satellites[i].getPosition(), satellites[i].getVelocity(), satellites[i].getAngle()))

                // create fragments
                FOR p <- 0...explosion1.getNumFragments()
                    satellites.push(Fragment(satellites[i].getPosition(), satellites[i].getVelocity(), satellites[i].getAngle()))
                FOR p <- 0...explosion2.getNumFragments()
                    satellites.push(Fragment(satellites[i].getPosition(), satellites[i].getVelocity(), satellites[i].getAngle()))

                // delete satellite
                DEL satellites[i]
                DEL satellites[j]

```

Check Collisions Solution 2:



```
Simulator :: checkCollisions()

    // Use a map for the hash and key value ability
    mapOfPositions <- []

    // Check if anything has collided.
    FOR i <- 0...satellites.length

        // store as {Position: index}
        it <- mapOfPositions.emplace(satellites[i].getPosition(), i)

        // If key (position) is already present
        IF (it.second = FALSE)

            // handle collision
            explosion1 <- satellites[it.first.second]       // this will get satellite at the index corresponding to that key
            explosion2 <- satellites[i]

            // create parts
            FOR p <- 0...explosion1.getNumParts()
                satellites.push(Part(satellites[i].getPosition(), satellites[i].getVelocity(), satellites[i].getAngle()))
            FOR p <- 0...explosion2.getNumParts()
                satellites.push(Part(satellites[i].getPosition(), satellites[i].getVelocity(), satellites[i].getAngle()))

            // create fragments
            FOR p <- 0...explosion1.getNumFragments()
                satellites.push(Fragment(satellites[i].getPosition(), satellites[i].getVelocity(), satellites[i].getAngle()))
            FOR p <- 0...explosion2.getNumFragments()
                satellites.push(Fragment(satellites[i].getPosition(), satellites[i].getVelocity(), satellites[i].getAngle()))

            // delete satellite
            DEL satellites[it.first.second]
            DEL satellites[i]
```

Test Cases:

Class :: Angle		45 degrees		
Name	**Pre-Condition**	**Input**	**Output**	**Post-condition**
convertToDegrees	Angle = 0	degreesFromRadians(0.785398)	45	Angle = 0.785398
convertToRadians	Angle = 0	radianFromDegrees(45)	0.785398	Angle = 0.785398
normalize	Angle = -5.49779	normalize()		Angle = 0.785398
addRadian Positive	Angle = 0.785398	addRadian(0.349066)		Angle = 1.134464
addRadian Negative	Angle = 0.785398	addRadian(-0.261799)		Angle = 0.523599
addDegrees Positive	Angle = 0.785398	addDegree(45)		Angle = 1.5708
addDegrees Negative	Angle = 0.785398	addDegree(30)		Angle = 0.261799
operatorEquals	Angle = 0.785398	Angle = Angle(Position{21082000, 36515095})	angle	Angle = 0.523599

Class :: Satellite				
Name		**Input**	**Output**	

checkExpired FragmentFalse	expiredAfter = 5
	lifeTime = 4
		checkExpired()	FALSE	expiredAfter = 5
				lifeTime = 4

checkExpired FragmentTrue	expiredAfter = 5
	lifeTime = 5
		checkExpired()	TRUE	expiredAfter = 5
				lifeTime = 5

checkExpired ProjectileFalse	expiredAfter = 5
	lifeTime = 4
		checkExpired()	FALSE	expiredAfter = 5
				lifeTime = 4

checkExpired ProjectileTrue	expiredAfter = 5
	lifeTime = 5
		checkExpired()	TRUE	expiredAfter = 5
				lifeTime = 5

checkExpired SatelliteFalse		checkExpired()	FALSE	

satelliteUpdate	Position(0, 42164000)
	Velocity = (-3100, 0)
		Update()		Position = (-148800, 42163224.503522)
				Velocity = (-3100, -10.77078442)

explodeSputnik		Explode()	Explosition	explosition.getFragments = 4
				explosition.getParts = 0

explodeDragon		Explode()	Explosition	explosition.getFragments = 2
				explosition.getParts = 3

explodeGPS		Explode()	Explosition	explosition.getFragments = 2
				explosition.getParts = 3

explodeHubble		Explode()	Explosition	explosition.getFragments = 0
				explosition.getParts = 4

explodeShip		Explode()	Explosition	explosition.getFragments = 4
				explosition.getParts = 0

explodeAtomic		Explode()	Explosition	explosition.getFragments = 0
				explosition.getParts = 0

explodeProjectile		Explode()	Explosition	explosition.getFragments = 0
				explosition.getParts = 0

explodeStarLink		Explode()	Explosition	explosition.getFragments = 2
				explosition.getParts = 2

shipRotateRigth	Angle = 0.785398	rotateRigth()		Angle = 0.885398
shipRotateLeft	Angle = 0.785398	rotateRigth()		Angle = 0.685398
satelliteRotateRigth	Angle = 0.785398	rotateRigth()		Angle = 0.785398
satelliteRotateLeft	Angle = 0.785398	rotateRigth()		Angle = 0.785398

shipAccelerationTrue	Position(0, 42164000)
	Velocity = (-3100, 0)
		setAcelerated(TRUE)		Position = (-148800, 42162707.50587)
				Velocity = (-3100, -21.5415688)

shipAccelerationFalse	Position(0, 42164000)
	Velocity = (-3100, 0)
		setAcelerated(FALSE)		Position = (-148800, 42163224.503522)
				Velocity = (-3100, -10.77078442)

satelliteAccelerationTrue	Position(0, 42164000)
	Velocity = (-3100, 0)
		setAcelerated(TRUE)		Position = (-148800, 42163224.503522)
				Velocity = (-3100, -10.77078442)

satelliteAccelerationFalse	Position(0, 42164000)
	Velocity = (-3100, 0)
		setAcelerated(FALSE)		Position = (-148800, 42163224.503522)
				Velocity = (-3100, -10.77078442)

fileProjectileTrue	Position(0, 42164000)
	Velocity = (-3100, 0)
		fireProjectile(Projectile)		Position = (-148800+NewPositionX, 42163224.503522+NewPositionY)
				Velocity = (-3100+NewVelocityX, -10.77078442+NewVelocityY)