* **World**: **DEFENSIEF – ENKEL FORMEEL**
  + Objecten (Klassen):
    - Ship
    - Asteroid
    - Bullet
  + Attributen:
    - width
    - height
    - maxWidthHeight = Double.*MAX\_VALUE*
  + Methodes:
    - **addShip(**Coordinate initLocation, Velocity initVelocity, Double angle**)**
    - **removeShip(**ShipremShip**)**
    - **addAsteroid(**Coordinate initLocation, Velocity initVelocity, Double angle)
    - **removeAsteroid(**AsteroidremAsteroid**)**
    - **addBullet(**Coordinate initLocation, Velocity initVelocity, Double angle**)**
    - **removeBullet(**BulletremBullet**)**
    - **evolve(**doubletime**)** (zie p7 voor uitwerking)
  + Opmerkingen:
    - Objecten overlappen niet
    - Objecten zijn allemaal rond
    - Objecten bevinden zich altijd volledig binnen in de wereld
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
* **Ship:**
  + Objecten:
    - Mass: (eventueel gewoon attribuut i.p.v. nieuwe klasse)
    - Thruster:
  + Attributen:
  + Opmerkingen:
    - Position aspecten: **DEFENSIEF**
    - Velocity aspecten: **TOTAAL**
    - Radius aspecten: **DEFENSIEF**
    - Direction/Angle: **NOMINAAL**
* **Mass:**
  + Attributen:
    - **double** accelResistance
    - **double** collisionEffect
  + Methodes:
    - computeMass(**double** massDensity**, double** radius)
    - setMass(**double** newMass)
    - …
  + Constructors:
    - **Mass(double** massDensity**, double** radius**)**
    - **Mass(double** mass**)**
  + Opmerkingen:
    - Bepaalt de weerstand bij het versnellen
    - Bepaalt het effect
    - (= (4/3) \* *Math.PI* \* *Math.Pow*(Object.getRadius(),3) \* Object.getMassDensity())
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
* **Thruster:**
  + Attributen:
    - **static final double** force = 1,1E18
    - **public boolean** enabled
  + Methodes:
    - **enable**()
    - **disable**()
    - …
  + Opmerkingen:
    - Niet zeker of dit een aparte klasse moet zijn
    - Kan misschien methode zijn die “thrust” aanroept
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
* **Asteroid**:
  + Objecten:
    - Coordinate
    - Velocity
    - Mass
  + Attributen:
    - **public final** radius
    - **public static** double massDensity = 2,65E12 kg/km³
  + Methodes:
    - **getPosition**()
    - **getVelocity**()
    - **getRadius**()
    - **getMass**()
    - **getCollisionPosition**()
    - **getCollisionTime**()
    - …
  + Opmerkingen:
    - Position aspecten: **DEFENSIEF**
    - Velocity aspecten: **TOTAAL**
    - Radius aspecten: **DEFENSIEF**
    - radius veranderd nooit na initialisatie (minimumRadius = 0)
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
* **Bullet:**
  + Objecten:
    - Coordinate
    - Velocity
  + Attributen
    - **public final** radius = 3
    - **private final** ShipsourceShip (mag nooit null zijn)
    - **public final double** initVelocity = 250
    - **public static final** double massDensity = 7.8E12
    - **public double** angle
  + Methodes**:**
    - **getVelocity**()
    - **getRadius**()
    - **getSource**()
    - **getCollisionPosition()**
    - **getCollisionTime**()
  + Opmerkingen:
    - Position aspecten: **DEFENSIEF**
    - Velocity aspecten: **TOTAAL**
    - Bij colliding Bullet verdwijnt
    - Als Bullet uit het veld vliegt is er geen effect (Bullet verdwijnt)
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    - . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Collision gedrag

* Als twee **Ships** botsen: snelheid wordt aangepast en in de tegengestelde zin
* idem **Asteroids**
* Als een **Bullet** iets raakt (**Ship**, **Asteroid** of andere **Bulet**) verdwijnen bijden entiteiten.
  + **Asteroid** wordt gesplitst in 2 kleinere als eerste **asteroid** radius > 30Km
  + 2 kleinere asteroids : **radius** = sourceAsteroid.getRadius() / 2
  + **angle** van 1ste **asteroid**: random, 2de **asteroid**: -random (tegengesteld)
  + **velocity** = 1,5 \* sourceAsteroid.getVelocity()
  + **Position:** sourceAsteroid.getRadius / 2 weg van het midden van sourceAsteroid
* Als een **Ship** een **Asteroid** raakt verdwijnt het **Ship**, de **Asteroid** blijft hetzelfde
* Als de verticale rand van de wereld geraakt wordt: **velocityX** = **-velocityX**
* Analoog horizontale rand: **velocityY** = **-velocityY**.
* Als een **Bullet** de rand voor een tweede keer raakt verdwijnt de **Bullet**

Aanpassing snelheid:

**BIJ OVERFLOW: RETURN <> VERWACHTE WAARDE**

**Lees p8 Efficient Collision Predicition**

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .