

Robocode

PROP 2012-13 Q2

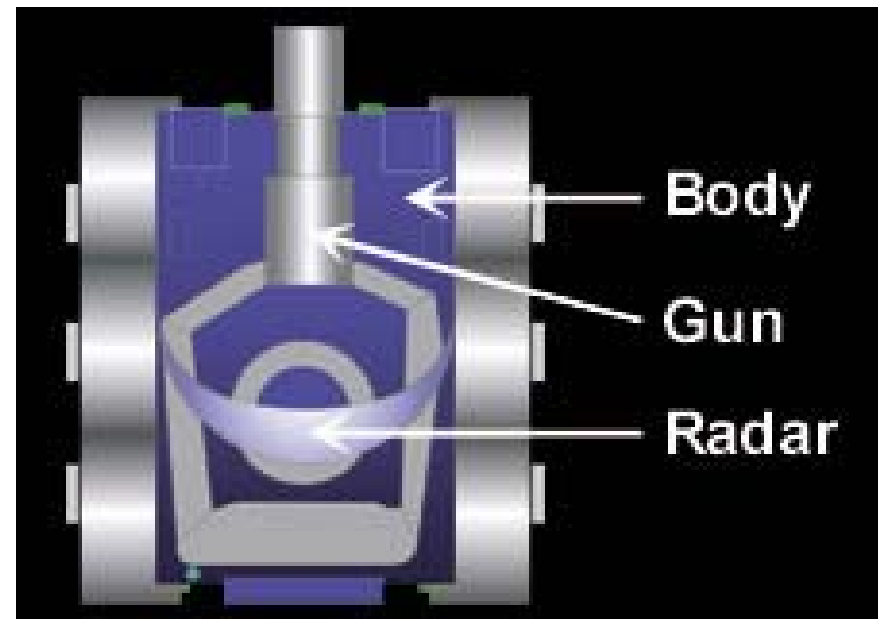
Mario Martin

Aprendre amb Robocode

- Aprenentatge de programació. Alguns exemples:
 - Programació Orientada a Objectes
 - Extendre classes
 - Herència
 - Polimorfisme
 - Fer servir Java docs
 - Cridar al codi d'una API
 - Gestió d'events

Objectiu

- Robot (estil tanc) propi competeix amb d'altres robots.
- Objectiu: sobreviure a tots els altres
- Anatomia del robot:
 - Cos del vehicle
 - Radar
 - Canó (Gun)



Programació bàsica

- Estendre la classe robot amb la definició (millor redefinició o sobrecàrrega dels mètodes) del teu robot, especialment el mètode **run()**

```
import robocode.*;
public class MyFirstRobot extends Robot{
    public void run(){
        ....
    }
}
```

Accions

- Mètodes ja definits a la classe robot, útils i que no cal redefinir:
 - **turnRight(double degree), turnLeft(double degree)**
 - **ahead(double distance), back(double distance)** [Mouen el robot la quantitat de pixels donada. Paren després del moviment o quan es xoca amb alguna cosa]
 - **turnGunRight(double degree), turnGunLeft(double degree).**
 - **turnRadarRight(double degree), turnRadarLeft(double degree)**

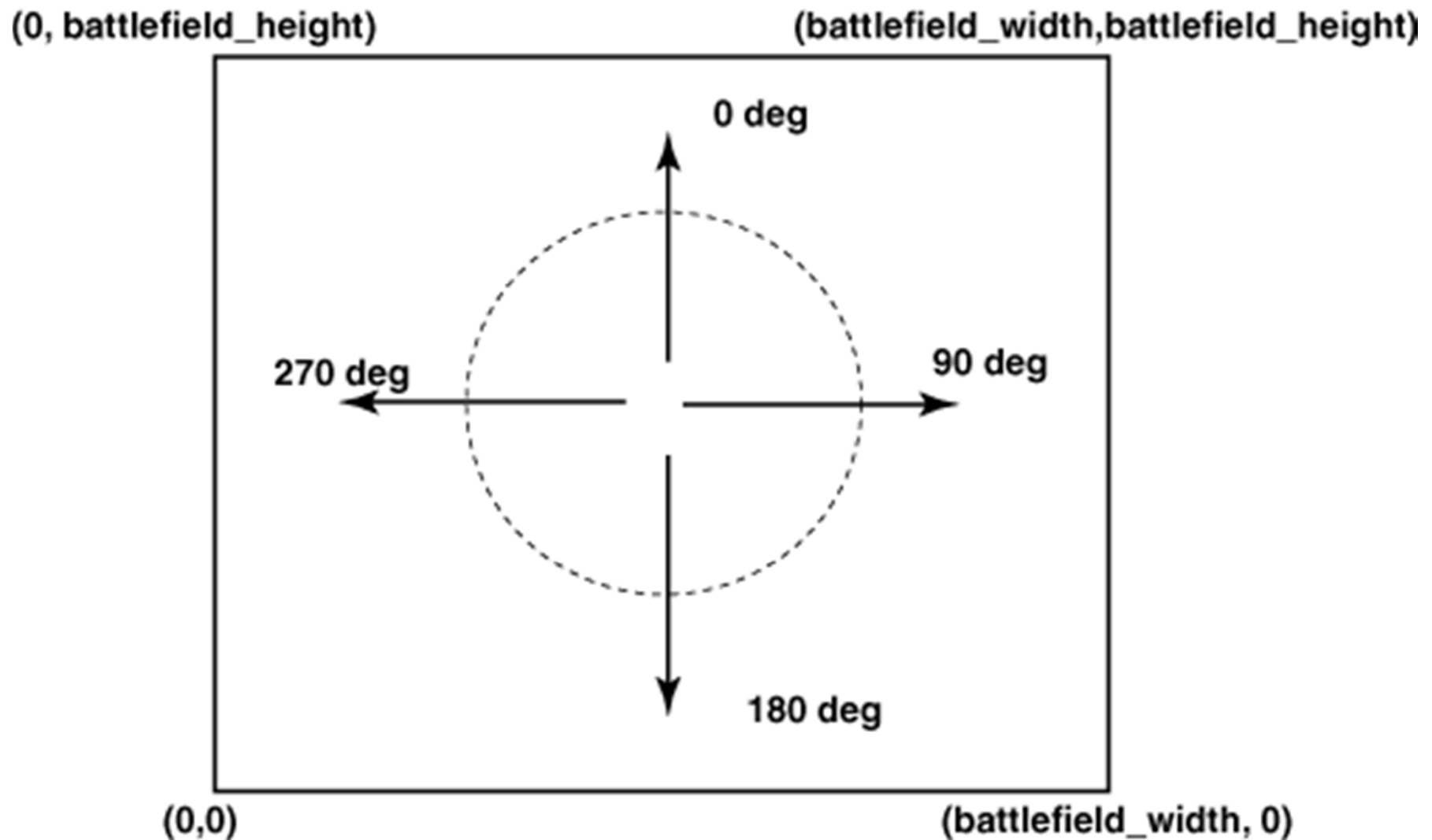
Notes sobre els girs

- Quan el robot gira, ho fa de forma solidaria excepte que posem els flags següents:
 - **setAdjustGunForRobotTurn(boolean flag):** If the flag is set to true, the gun will remain in the same direction while the vehicle turns.
 - **setAdjustRadarForRobotTurn(boolean flag):** If the flag is set to true, the radar will remain in the same direction while the vehicle (and the gun) turns.
 - **setAdjustRadarForGunTurn(boolean flag):** If the flag is set to true, the radar will remain in the same direction while the gun turns. It will also act as if `setAdjustRadarForRobotTurn(true)` has been called.

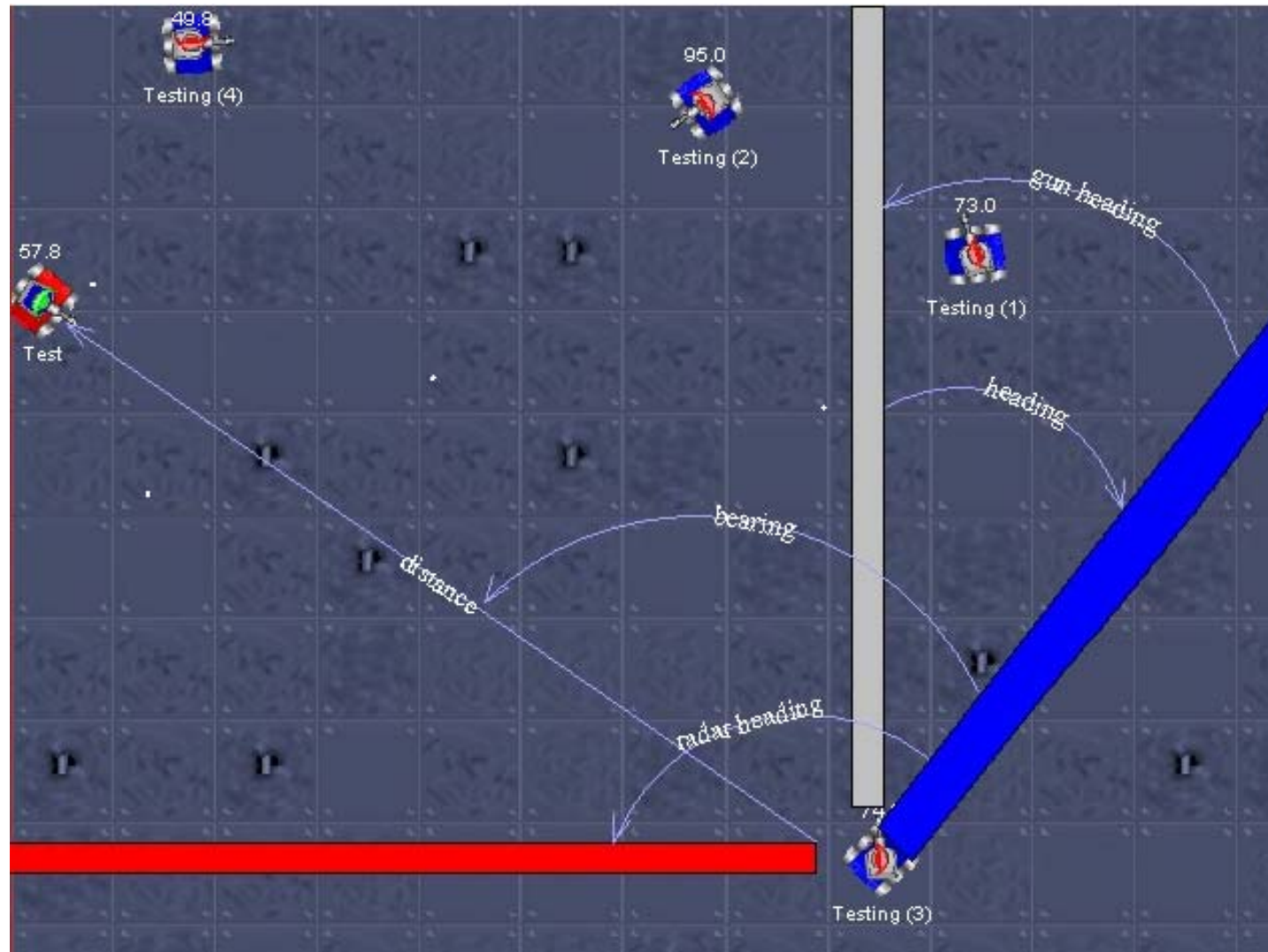
Accions (2)

- **getX() , getY()** Obté les coordenades actuals del robot
- **getHeading(), getGunHeading(), getRadarHeading()**
Obtenen direcció en graus del vehicle, canó i radar
- **getBattleFieldWidth(), getBattleFieldHeight()**
Obtenen dimensions del mon

Coordenades



Coordenades



Accions: Disparar

- Dues accions per disparar: **fire(double power)** i **fireBullet(double power)**
 - Cada dispar redueix energia del robot
 - Cada robot comença amb energia per defecte i es destrueix quan arriba a zero.
 - Dispara amb paràmetre d'energia que està en un rang de [1..3]. Com més energia més mal a l'altre (i més energia perduda)
 - **fireBullet()** retorna objecte de la classe `Bullet` que ens dona informació sobre com hem disparat.

Accions

- **getEnergy()**
Obté l'energia que li queda al robot.
- Més informació al JavaDoc de la classe Robot.

Energia

- El robot per energia
 - Disparar
 - Xocar
 - Ser “tocat”
- El robot guanya energia
 - Tocar a un altre
 - Amb el temps

Events

- Interrupció del **run** davant de possibles events
 - ScannedRobotEvent
 - HitByBulletEvent
 - BulletHit
 - HitRobotEvent
 - HitWallEvent
- Programació de la gestió dels events
 - ScannedRobotEvent → onScannedRobot(ScannedRobotEvent e)
 - HitByBulletEvent → onHitByBullet(HitByBulletEvent e)
 - BulletHit → onBulletHit(BulletHitEvent e)
 - HitRobotEvent → onHitRobot(HitRobotEvent e)
 - HitWallEvent → onHitWall(HitWallEvent e)
- Mètodes implementat per defecte "donothing" sobrecarregar els que es creguin necessaris.

Altres events

- Altres events
 - onBulletMissed
 - onBulletHitBullet
- No cal programar-los tots.

Events

- Cada event té informació associada per facilitar la programació.
- Exemple: ScannedRobotEvent
 - **getHeading()** [**getBearing()**] Retorna orientació del robot [respecte vehicle] en graus
 - **getDistance()** Retorna distancia en pixels al robot detectat
 - **getVelocity()**
 - **getEnergy()**
- Veure info per cada event a JavaDocs

Exemple de robot simple

```
package meurobot;
import robocode.*;

public class Meurobot extends Robot
{
    public void run() {
        turnLeft(getHeading());
        while(true) {
            ahead(1000);
            turnRight(90);
        }
    }
    public void onScannedRobot(ScannedRobotEvent e) {
        fire(1);
    }
    public void onHitByBullet(HitByBulletEvent e) {
        turnLeft(180);
    }
}
```

Advanced robot

- En el robot standard, les accions es fan una darrera de l'altre, dona lloc a moviment robòtic.
- En l'advanced robot pots fer moviments complexes.

Advanced robot vs. Robot

Robot class:

- `turnRight ()`
- `turnLeft ()`
- `turnGunRight ()`
- `turnGunLeft ()`
- `turnRadarRight ()`
- `turnRadarLeft ()`
- `ahead ()`
- `back ()`

AdvancedRobot class:

- `setTurnRight ()`
- `setTurnLeft ()`
- `setTurnGunRight ()`
- `setTurnGunLeft ()`
- `setTurnRadarRight ()`
- `setTurnRadarLeft ()`
- `setAhead ()`
- `setback ()`

En advanced robot

```
package meurobot;
import robocode.*;
public class Meurobot extends AdvancedRobot
{
    public void run() {
        turnLeft(getHeading());
        while(true) {
            setTurnRight(10000);
            setTurnGunRight(20);
            ahead(2000);
            execute()
        }
    }
    public void onScannedRobot(ScannedRobotEvent e) {
        fire(1);
    }
    public void onHitByBullet(HitByBulletEvent e) {
        turnLeft(180);
    }
}
```