

PROP: Activitat 1

Normativa i Enunciat

Bernardino Casas,
Mario Martin

Primavera 2013

Aquest document és llarg però és imprescindible que el llegiu íntegrament i amb deteniment ja que es donen les instruccions i normes que heu de seguir per a que la vostra activitat sigui avaluada positivament. El professorat de l'assignatura donarà per fet que tots els alumnes coneixen el contingut íntegre d'aquest document.

1 Normativa

1. Tal i com s'explica a la Guia Docent de l'assignatura una de les proves d'avaluació que formen l'assignatura és aquesta activitat relacionada amb la programació que val un 10%.
2. Aquesta activitat és de caràcter individual o per parella i no obligatòria. Es pot aprovar l'assignatura sense haver realitzat aquesta activitat.
3. Per tal de valorar el vostre treball s'han de complir aquests requeriments:
 - El robot ha de ser un treball original vostre.
 - El codi del robot ha de compilar i executar-se.
 - El codi ha d'estar documentat i heu d'explicar amb detall l'estratègia que heu seguit.
 - El vostre robot ha de guanyar al robot exemple Dummy.
 - En altre cas la nota d'aquesta activitat serà un 0 suspès.
4. Els robots que compleixin els criteris esmentats es valoraran segons:
 - la seva execució en l'entorn del STIC amb una sèrie de jocs de prova i
 - la correcció del disseny, implementació i documentació: les decisions de disseny i la seva justificació, l'eficiència dels algorismes i estructures de dades, la llegibilitat, robustesa i estil de programació, etc. Tota la documentació ha d'acompanyar el codi; no heu de lliurar cap documentació en paper.
5. La data límit del lliurament final és el dijous 4 d'Abril de 2013 a les 12 del mig-dia. Si algú no lliura l'activitat abans del límit indicat llavors la nota serà 0. Al Campus Digital (<http://atenea.upc.edu>) hi haurà disponible una bústia amb instruccions pel lliurament.

2 Objectiu de l'activitat

Robocode és un entorn de simulació de guerres de robots, desenvolupat per Alphaworks de IBM®. Robocode permet programar tancs de combat en Java per combatre en el camp de batalla contra altres tancs programats per altres jugadors. Existeixen dos modes de jocs:

- *batalla individual*, on cada robot lluita contra tots els altres
- *batalla en equip*, on un exèrcit de robots lluita per la victòria de manera col·laborativa.

En el nostre cas ens centrarem en la batalla individual.

L'objectiu d'aquesta activitat és crear un robot competitiu que sigui capaç de derrotar d'altres robots en un torneig de Robocode.

3 Descripció del sistema

L'objectiu del joc és generar la I.A. que permeti a un robot derrotar a altres robots. A Robocode hi ha dos elements principals, els robots i les batalles. Les batalles es realitzen en el terreny de combat entre els robots, que juguen autonomament seguint una programació concreta que defineix el seu comportament.

A continuació s'explica aquests dos elements.

3.1 Els robots

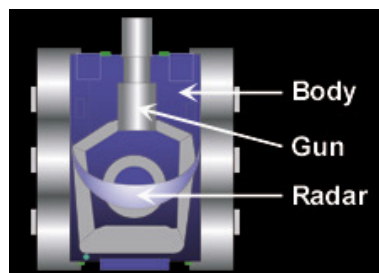


Figura 1: Parts d'un robot

Com es pot veure a la figura 1 els robots estan formats per tres parts:

- Cos (body) - porta el canó amb el radar a sobre. El cos s'usa per moure el robot endavant i enrera, i també per girar a l'esquerra o a la dreta.
- Canó (gun) - muntat a sobre del cos s'usa per disparar bales d'energia. El canó pot girar a l'esquerra o a la dreta.

- Radar - muntat a sobre del canó s'usa per detectar els altres robots. El radar pot girar a l'esquerra o a la dreta. El radar genera `ScannedRobot` events quan es detecten els robots.

Robocode disposa de multitud de mètodes per controlar els robots, per exemple per gestionar els disparats i els atacs als nostres enemics.

Al principi de cada combat tot robot comença amb un nivell d'energia per defecte, i un robot mor quan la seva energia disminueix a 0. L'energia d'un robot pot disminuir per diferents causes:

- Rebi el tret d'un enemic
- Colisioni amb un robot enemic
- Colisioni amb parets o obstacles del camp de batalla.

A més a més, l'energia alliberada en el tret també resta energia al robot.

Quanta més energia s'utilitzi en un dispar, major mal s'inflingirà al robot enemic, però també més energia restarà del nostre tanc.

Però no tot són pèrdues d'energia. També es recupera energia quan s'encerta amb un dispar a algun enemic, o bé, de forma constant per l'efecte de "refredament dels canons".

3.2 Terreny de combat

Robocode usa el sistema de coordenades cartesianes. Això significa que la coordenada (0, 0), tal i com es pot veure a la figura 2, està localitzada abaix a l'esquerra del terreny de combat.

Robocode usa com a direcció el sentit de les agulles del rellotge. Així doncs 0 o 360 graus és el "Nord", 90 graus "Est", 180 graus "Sud" i 270 graus és l'Oest.

3.3 Accions d'un robot

Les accions que pot emprendre un robot són les següents:

```
/**
 * Mou el robot cap endavant la distància indicada pel paràmetre.
 * S'executa automàticament i no retorna res fins que no ha
 * acabat el moviment. En el cas de que el robot col·lisió amb
 * algun obstacle s'atura.
 */
public void ahead(double distancia)
```

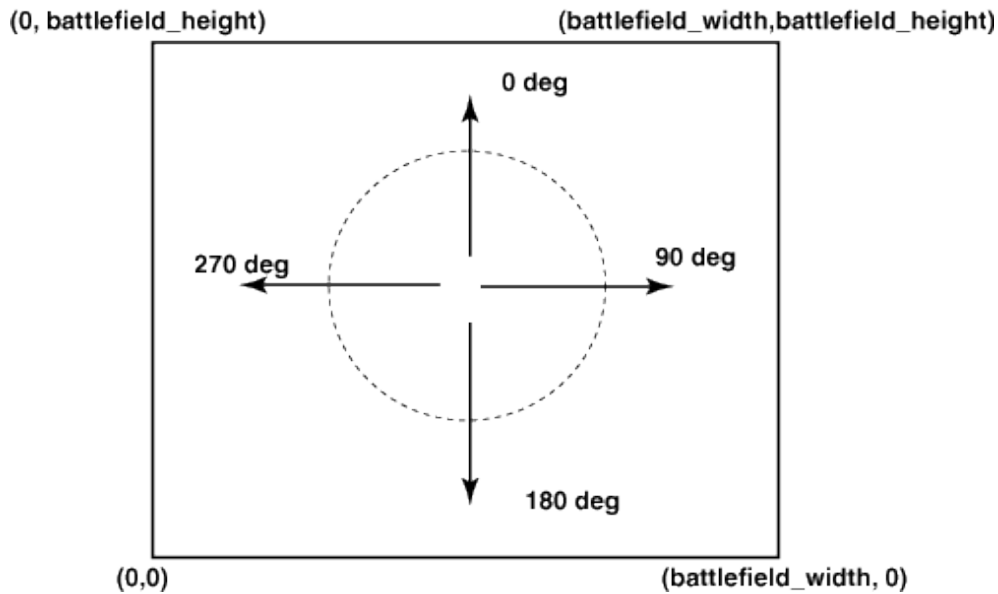


Figura 2: Terreny de combat

```

/**
 * Mou el robot cap enrera la distància indicada pel paràmetre.
 * S'executa automàticament i no retorna res fins que no ha
 * acabat el moviment. En el cas de que el robot col·lisió amb
 * algun obstacle s'atura.
 */
public void back(double distancia)

/**
 * Dispara una bala d'energia. El rang vàlid de potència va des
 * de 0.1 fins a 3. La bala segueix el trajecte cap a on apunti
 * el canó. Si la bala arriba a l'enemic li causarà el dany
 * corresponent a (4*potencia). S'executa immediatament. A més,
 * l'energia amb que es carrega la bala es resta de l'energia
 * actual del robot.
 */
public void fire(double potencia)

/**
 * Rota el robot els graus indicats a l'esquerra. S'executa
 * immediatament i no retorna res fins que no finalitza.
 */
public void turnLeft(double graus)

/**

```

```

    * Rota el robot els graus indicats a la dreta. S'executa
    * immediatament i no retorna res fins que no finalitza.
    */
public void turnRight(double distancia)

/**
 * Rota el canó del robot i el radar els graus indicats a l'esquerra.
 * S'executa immediatament i no retorna res fins que no finalitza.
 */
public void turnGunLeft(double distancia)

/**
 * Rota el canó del robot i el radar els graus indicats a la dreta.
 * S'executa immediatament i no retorna res fins que no finalitza.
 */
public void turnGunRight(double distancia)

/**
 * Retorna la direcció a la que apunta el front del robot en graus
 * (0 a 360).
 */
public double getHeading()

```

3.4 Els events

Durant el combat es generen “events”. Aquests events són situacions que es produeixen a partir de les quals el robot decideix quines accions emprendre.

Tot tanc hereda de la classe `Robot`, i aquesta classe ja té per defecte mètodes per gestionar els events. Però aquests mètodes es poden sobreescrivre i així programar les accions que el robot ha de realitzar en cas que es generi un tipus d'event.

Els diferents tipus d'events són els següents:

- `ScannedRobotEvent`
- `BulletHitEvent`
- `HitRobotEvent`
- `HitWallEvent`
- `HitByBulletEvent`

3.4.1 ScannedRobotEvent

ScannedRobotEvent es genera quan el radar detecta un robot al seu pas per una zona angular. Aquest event el gestiona el mètode onScannedRobotEvent. Alguns dels mètodes principals són:

```
/**
 * Retorna l'angle (en graus) de l'enemic en relació a l'angle
 * que apunta el nostre robot. Retorna un valor entre -180
 * graus i 180 graus. Per exemple, si el nostre tanc mira cap
 * al nord (0 graus) i l'enemic està a l'esquerra, aquest mètode
 * tornaria -90.
 */
public double getBearing()

/**
 * Retorna la distància a l'enemic.
 */
public double getDistance()
```

3.4.2 BulletHitEvent

BulletHitEvent es genera quan un dispar del nostre robot encerta a un dels enemics. Aquest event el gestiona el mètode onBulletHitEvent. Alguns dels mètodes principals són:

```
/**
 * Retorna que li resta a l'adversari descomptant el dispar.
 */
public double getEnergy()
```

3.4.3 HitRobotEvent

HitRobotEvent es genera quan el nostre robot xoca amb un altre robot. Aquest event el gestiona el mètode onHitRobotEvent. Alguns dels mètodes principals són:

```
/**
 * Retorna l'angle (en graus) de l'enemic en relació a l'angle
 * que apunta el nostre robot. Retorna un valor entre -180
 * graus i 180 graus. Per exemple, si el nostre tanc mira cap
 * al nord (0 graus) i l'enemic està a l'esquerra, aquest mètode
 * tornaria -90.
 */
public double getBearing()
```



```
/**
 * Retorna que li resta a l'adversari descomptant el dispar.
 */
public double getEnergy()
```

3.4.4 HitWallEvent

HitWallEvent es genera quan el robot xoca amb un mur del que limiten el terreny de combat. Aquest event el gestiona el mètode onHitWallEvent. Alguns dels mètodes principals són:

```
/**
 * Retorna l'angle (en graus) del mur en relació a l'angle
 * que apunta el nostre robot. Retorna un valor entre -180
 * graus i 180 graus. Per exemple, si el nostre tanc mira cap
 * al nord (0 graus) i el mur està a l'esquerra, aquest mètode
 * tornaria -90.
 */
public double getBearing()
```

3.4.5 HitByBulletEvent

HitByBulletEvent es genera quan el nostre robot rep un tret. Aquest event el gestiona el mètode onHitByBulletEvent. Alguns dels mètodes principals són:

```
/**
 * Retorna l'angle (en graus) de l'enemic en relació a l'angle
 * que apunta el nostre robot. Retorna un valor entre -180
 * graus i 180 graus. Per exemple, si el nostre tanc mira cap
 * al nord (0 graus) i l'enemic està a l'esquerra, aquest mètode
 * tornaria -90.
 */
public double getBearing()
```

4 Com fer el teu primer robot

Totes els robots que es creen a Robocode hereden de la classe `robocode.Robot`. Aquesta classe té mètodes que permeten interactuar amb el joc. Un robot bàsicament té tres parts:

- Atributs del robot i mètode propis del nostre robot.

- Mètode `run()`. És el mètode que crida el gestor de combat quan comença el combat. Bàsicament es divideix en dues parts:
 - Àrea on es defineixen les accions que es faran només un cop per cada instància de robot.
 - Àrea dins d'un `while` infinit on es defineixen les accions repetitives que el robot anirà realitzant durant el combat.
- Implementació dels mètodes per gestionar els diferents events.

Exemple:

```

1 package <nom_equip>;
2 import robocode.*
3 import java.awt.Color;
4
5 public class Pepe extends Robot {
6     // Declaració d'atributs i mètodes privats del nostre robots.
7
8     // Mètode run que indica el que farà el robot cada torn.
9     public void run() {
10         // Accions inicials
11         setColors(Color.red, Color.yellow, Color.magenta)
12
13         // Accions per cada torn
14         while (true) {
15             // comportament desitjat del robot
16             ahead(100);
17             turnGunRight(360);
18             back(100);
19             turnGunLeft(360);
20         }
21     }
22
23     // Mètodes de captura dels events
24     public void onScannedRobot(ScannedRobotEvent e) {
25         fire(1);
26     }
27
28     public void onHitByBulletEvent(HitByBulletEvent e) {
29         turnLeft(90 - e.getBearing());
30         ahead(30);
31     }
32 }

```

A Altra informació

A.1 Instal·lació de Robocode

NOTA: El compilador de Java ha d'estar instal·lat a la màquina per tal de poder instal·lar i executar Robocode!

En aquesta secció s'explicarà com instal·lar robocode a Linux. Cal tenir present que Robocode NO està instal·lat als ordinadors de l'escola, per tant caldrà que instal·leu en la vostra compte d'usuari el sistema Robocode per tal de poder-lo fer servir.




1. Cal obrir un terminal i escriure:

```
java -jar robocode-x.y.z-setup.jar
```

(x.y.z ha de ser substituït per número de la versió de Robocode).

2. El millor lloc per instal·lar-lo és en el directori ~/ (que és /home/est/usuari/).
3. Un cop instal·lat el sistema podreu accedir-hi a partir del directori: ~/robocode. Llavors només cal executar l'script `robocode.sh` per iniciar el sistema.

A.2 Més recursos

-  <http://robocode.sourceforge.net/> A la web del projecte robocode podeu trobar molta informació addicional sobre aquest projecte.
-  <http://robocode.sourceforge.net/docs/robocode/> API completa amb l'explicació dels mètodes de cadascuna de les classes que componen robocode.
-  <http://robowiki.net/wiki/Robocode/FAQ> Preguntes freqüents sobre el projecte robocode.