

# Robocode - cos'è

Un'arena dove robot da combattimento controllati da codice si danno battaglia.

Obiettivo: ottenere più punti degli avversari (reward) al termine dello scontro.

Nato per insegnare la programmazione in Java in modo divertente, ha attirato presto l'attenzione di diversi ambiti dell'informatica per la complessità del problema che modella.

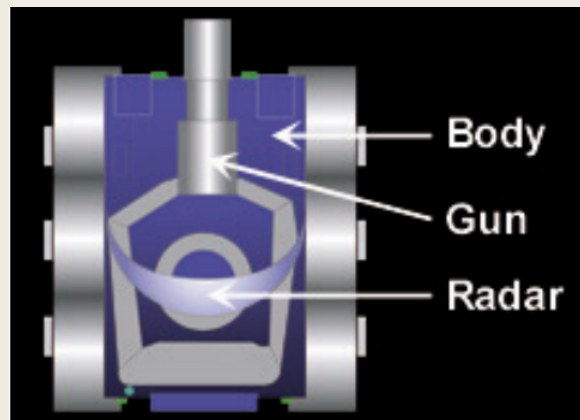
Progetto open-source, nato nel 2005. Tuttora mantenuto ed aggiornato.

# L'arena

- Arena rettangolare
- Dimensione variabile
- Nessun ostacolo

# I robot

- Possono muoversi avanti e indietro
- Possono ruotare
- Possiedono un radar rotante che scannerizza un arco molto ridotto per individuare nemici, proiettili e muri.
- Possiede un cannone con cui sparare.
- Cannone e radar ruotano indipendentemente
- Ogni robot ha un ammontare di Punti Salute, finiti i quali, esplode.



# Punteggio e danni

I punteggi sono dati in base ai danni inflitti.

Essere l'ultimo sopravvissuto non è quasi mai sinonimo di vittoria.

Bisogna puntare a sopravvivere ma senza infliggere danni, non si riuscirà a vincere.

I danni vengono inflitti colpendo l'avversario con i proiettili. Ogni proiettile ha un'intensità che va da 0.1 a 3. Ogni proiettile infligge danno a chi spara ma se colpisce, fa recuperare Punti Salute. Più è forte il colpo, più saranno gli HP consumati e recuperati. Sparare a vuoto è equivalente a farsi colpire.

La velocità del proiettile è inversamente proporzionale alla potenza.

Si subiscono danni colpendo i bordi dell'arena.

Si subiscono e infliggono danni anche scontrandosi con gli avversari.

# Turni e movimento

Il gioco è strutturato a turni. Ad ogni turno viene computato il comportamento di tutti i robot ed eseguito contemporaneamente.

Lo spostamento e la rotazione delle varie parti del robot ha un vincolo massimo all'interno del turno: non ci può muovere per più di 8 pixel per turno.

Un robot non può muoversi e ruotare nello stesso turno.

# Esempio di combattimento



# Aspetti tecnici

Linguaggio da utilizzare: Java.

Compatibile con AspectJ, Scala, Clojure e integrabile in Eclipse.

Il codice che dovremo scrivere verrà valutato ad ogni turno.

Ambiente rappresentato come insieme di eventi  
(onScannedRobot, onHitByBullet, onHitRobot...)

Azioni rappresentate come chiamate della libreria robocode  
(setAhead, setTurnRadarRight, setFire...)

# Riferimenti

Sito ufficiale:

<http://robocode.sourceforge.net/>

Download:

<http://sourceforge.net/projects/robocode/files/>

Wiki:

<http://robowiki.net/>

Applying Machine Learning to Robocode:

<http://www.dinbedstemedarbejder.dk/Dat3.pdf>