



Υλοποίηση Πληροφοριακών Συστημάτων - Προγραμματισμός σε Java (robocode) - “know your enemy” -

Βασίλειος Καρακίδας
bkarak@aub.gr



Περιεχόμενα

- Εργασία 2 - Robocode
- Οι αντίπαλοι πράκτορες
- Ανάλυση πρακτόρων
 - ▶ Tracker



Εργασία II

- Δημιουργία ενός πράκτορα για την πλατφόρμα Robocode
- Χαρακτηριστικά:
 - ▶ Αυτόνομος
 - ▶ Να επιτίθεται σε αντίπαλους πράκτορες
 - ▶ Να εντοπίζει αντίπαλους πράκτορες



Εργασία II (2)

- 2 παραδοτέα
 - ▶ Μεταγλωττισμένη έκδοση του πράκτορα (agent-<κωδ>.jar)
 - ▶ Πηγαίος κώδικας (source-<κωδ>.zip)



Εργασία II (3)

- Παράδοση μέσω του ηλεκτρονικού συστήματος παράδοσης εργασιών

<http://www.dmst.aueb.gr/hlektronikh.htm>

- 11/12/2007 - 31/1/2008 (7 μέρες ...)



Αρένα Μάχης

Robocode: Round 1 of 10 (paused)

Battle Robot Options Help

Tracker
DummyRobot

72.0
DummyRobot
98.0
Tracker

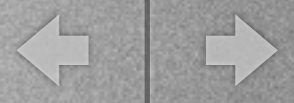
Pause/Debug Next Turn Stop Restart 30

The image shows a Robocode arena window titled "Robocode: Round 1 of 10 (paused)". The arena is a 10x10 grid of blue squares. Two robots are visible: "DummyRobot" (blue) with a health of 72.0 and "Tracker" (yellow) with a health of 98.0. Other robots are represented by small 'P' icons. The interface includes a menu bar with "Battle", "Robot", "Options", and "Help". On the right, there are buttons for "Tracker" and "DummyRobot". At the bottom, there are buttons for "Pause/Debug", "Next Turn", "Stop", and "Restart", along with a progress bar showing 30.



Αντίπαλοι πράκτορες

- **Sample.SittingDuck**
- **Sample.Fire**
- **Sample.Tracker**
- **Sample.SpinBot**



Αντίπαλοι πράκτορες (2)

▼ Robocode	Jan 11, 2008, 9:42 AM	--	Folder
versions.txt	Oct 9, 2007, 11:46 PM	76 KB	Plain text
▶ templates	Dec 19, 2007, 12:15 AM	--	Folder
teamrubble.sh	Sep 9, 2007, 11:32 AM	4 KB	Shell script
teamrubble.bat	Mar 17, 2007, 1:08 PM	4 KB	Document
▼ robots	Today, 1:14 PM	--	Folder
▶ sampleteam	Dec 19, 2007, 12:15 AM	--	Folder
▼ sample	Dec 19, 2007, 12:16 AM	--	Folder
Walls.properties	Jun 28, 2007, 12:03 AM	4 KB	Java p...ies f
Walls.java	Aug 20, 2006, 11:21 PM	4 KB	Java S...ce Fi
Walls.class	Oct 9, 2007, 11:52 PM	4 KB	Class files
TrackFire.properties	Jun 28, 2007, 12:02 AM	4 KB	Java p...ies f
TrackFire.java	Aug 20, 2006, 11:20 PM	4 KB	Java S...ce Fi
TrackFire.class	Oct 9, 2007, 11:52 PM	4 KB	Class files
Tracker.properties	Jun 28, 2007, 12:01 AM	4 KB	Java p...ies f
Tracker.java	Aug 20, 2006, 11:20 PM	8 KB	Java S...ce Fi
Tracker.class	Oct 9, 2007, 11:52 PM	4 KB	Class files
Target\$1.class	Oct 9, 2007, 11:52 PM	4 KB	Class files
Target.properties	Jun 28, 2007, 12:00 AM	4 KB	Java p...ies f
Target.java	Aug 20, 2006, 11:19 PM	4 KB	Java S...ce Fi
Target.class	Oct 9, 2007, 11:52 PM	4 KB	Class files
SpinBot.properties	Jun 27, 2007, 11:59 PM	4 KB	Java p...ies f
SpinBot.java	Aug 20, 2006, 11:18 PM	4 KB	Java S...ce Fi
SpinBot.class	Oct 9, 2007, 11:52 PM	4 KB	Class files
SittingDuck.properties	Jun 27, 2007, 11:58 PM	4 KB	Java p...ies f
SittingDuck.java	Aug 20, 2006, 11:17 PM	4 KB	Java S...ce Fi



Αντίπαλοι πράκτορες (3)

- `Sample.SittingDuck`
 - ▶ Ανενεργός πράκτορας
 - ▶ Κάθεται ακίνητος και περιμένει να ηττηθεί



Αντίπαλοι πράκτορες (4)

- Sample.Fire
 - ▶ Κάθεται ακίνητος
 - ▶ Περιστρέφει το πυροβόλο 360 μοίρες
 - ▶ Αν χτυπηθεί μετακινείται



Αντίπαλοι πράκτορες (6)

- Sample. SpinBot
 - ▶ Κάνει κύκλους (360 μοίρες)
 - ▶ Αν εντοπίσει αντίπαλο πθροβολεί, αλλά συνεχίζει την πορεία του



Αντίπαλοι πράκτορες (5)

- Sample.Tracker
 - ▶ Περιστρέφει το κανόνι μέχρι να εντοπίσει αντίπαλο πράκτορα
 - ▶ Όταν τον εντοπίσει μετακινείται κοντά και πυροβολεί



Ανάλυση πράκτορα (I)

- **Sample.Tracker**
 - ▶ Ο πηγαίος κώδικας αποτελείται από μία κλάση (Tracker)
 - ▶ Χρησιμοποιεί δύο βοηθητικές συναρτήσεις και τρία γεγονότα



Ανάλυση πράκτορα (2)

- Η βασική συνάρτηση είναι η `run()`
- Τα γεγονότα που χρησιμοποιεί:
 - ▶ `onScannedRobot()`
 - ▶ `onHitRobot()`
 - ▶ `onWin()`



Ανάλυση πράκτορα (3)

```
int count = 0; // Keeps track of how long we've
// been searching for our target
double gunTurnAmt; // How much to turn our gun when searching
String trackName; // Name of the robot we're currently tracking

public void run() {
    // Set colors
    setBodyColor(new Color(128, 128, 50));
    setGunColor(new Color(50, 50, 20));
    setRadarColor(new Color(200, 200, 70));
    setScanColor(Color.white);
    setBulletColor(Color.blue);

    // Prepare gun
    trackName = null; // Initialize to not tracking anyone
    setAdjustGunForRobotTurn(true); // Keep the gun still when we turn
    gunTurnAmt = 10; // Initialize gunTurn to 10
    [.....]
}
```



Ανάλυση πράκτορα (4)

```
while (true) {  
    // turn the Gun (looks for enemy)  
    turnGunRight(gunTurnAmt);  
    // Keep track of how long we've been looking  
    count++;  
    // If we've haven't seen our target for 2 turns, look left  
    if (count > 2) {  
        gunTurnAmt = -10;  
    }  
    // If we still haven't seen our target for 5 turns, look right  
    if (count > 5) {  
        gunTurnAmt = 10;  
    }  
    // If we *still* haven't seen our target after 10 turns, find another  
    target  
    if (count > 11) {  
        trackName = null;  
    }  
}
```




Ανάλυση πράκτορα (5)

```
public void onHitRobot(HitRobotEvent e) {
    // Only print if he's not already our target.
    if (trackName != null && !trackName.equals(e.getName())) {
        out.println("Tracking " + e.getName() + " due to collision");
    }
    // Set the target
    trackName = e.getName();
    // Back up a bit.
    // Note: We won't get scan events while we're doing this!
    // An AdvancedRobot might use setBack(); execute();
    gunTurnAmt = normalRelativeAngle(e.getBearing() + (getHeading() -
getRadarHeading()));
    turnGunRight(gunTurnAmt);
    fire(3);
    back(50);
}
```



Ανάλυση πράκτορα (6.1)

```
public void onScannedRobot(ScannedRobotEvent e) {  
    // If we have a target, and this isn't it, return immediately  
    // so we can get more ScannedRobotEvents.  
    if (trackName != null && !e.getName().equals(trackName)) {  
        return;  
    }  
  
    // If we don't have a target, well, now we do!  
    if (trackName == null) {  
        trackName = e.getName();  
        out.println("Tracking " + trackName);  
    }  
    // This is our target.  Reset count (see the run method)  
    count = 0;  
    [...]
```



Ανάλυση πράκτορα (6.2)

```
// If our target is too far away, turn and move toward it.
if (e.getDistance() > 150) {
    gunTurnAmt = normalRelativeAngle(e.getBearing() + (getHeading() - getRadarHeading()));

    turnGunRight(gunTurnAmt); // Try changing these to setTurnGunRight,
    turnRight(e.getBearing()); // and see how much Tracker improves...
    // (you'll have to make Tracker an AdvancedRobot)
    ahead(e.getDistance() - 140);
    return;
}

// Our target is close.
gunTurnAmt = normalRelativeAngle(e.getBearing() + (getHeading() - getRadarHeading()));
turnGunRight(gunTurnAmt);
fire(3);

// Our target is too close! Back up.
if (e.getDistance() < 100) {
    if (e.getBearing() > -90 && e.getBearing() <= 90) {
        back(40);
    } else {
        ahead(40);
    }
}
scan();
}
```



Ανάλυση πράκτορα (4)

```
/**
 * onWin: Do a victory dance
 */
public void onWin(WinEvent e) {
    for (int i = 0; i < 50; i++) {
        turnRight(30);
        turnLeft(30);
    }
}
```