3.– 6. September 2012
in Nürnberg

Herbstcampus

Wissenstransfer
par excellence

Träumen Roboter von elektrischen Schafen?

Spielerisch (besser) programmieren lernen mit Scalatron und Robocode

Joachim Hofer

imbus AG

@johofer, http://jmhofer.johoop.de

Fahrplan

- Motivation
- Robocode
- Scalatron
- Fazit

Kurzvorstellung

- > 10 Jahre Teamleiter Softwareentwicklung
- > 10 Jahre zu viel Enterprise Java
- Praktizierender™ Scrum Master
- Schwerpunkte **Test** und Code Coverage
- **Scala-Enthusiast**
- Autor von eCobertura (inzwischen veraltet...)
- Autor diverser **sbt-Plugins**
  - jacoco4sbt, findbugs4sbt, cpd4sbt, ant4sbt etc
- sporadischer Blogger (und Zwitscherer)

Motivation: Warum lernen?

- Pragmatic Programmer:
  - Wissen und Erfahrung veralten schnell!
  - → *„Learn at least one new language every year.“*
- Lernen...
  - öffnet neue Perspektiven und
  - ermöglicht Kreativität

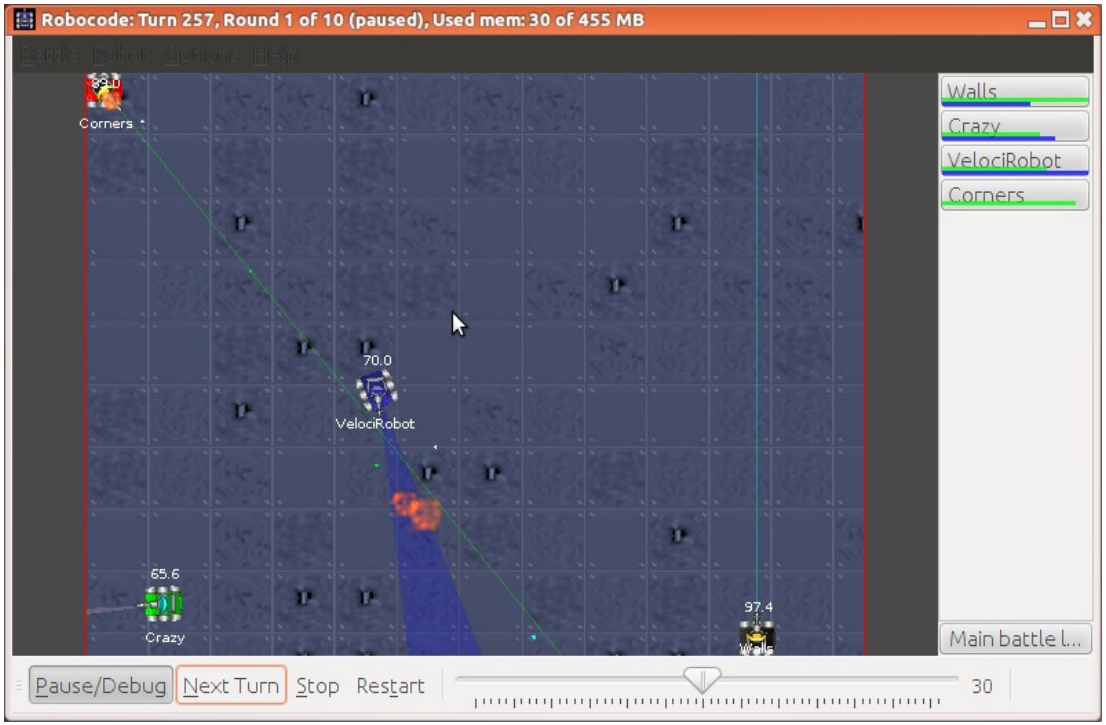
Programmierspiele (Auswahl)

- Core War (Maschinensprache)
- JS Robots (JavaScript)
- Robocode (JVM, .NET)
- Scalatron (Scala, Java)
- Google AI Challenge
  - Ants (beliebige Sprache)
  - Planet Wars (beliebige Sprache)

Motivation: Warum mit Roboterspielen?

- Weil es Spaß macht!
  - höhere Motivation → besserer Lerneffekt
- *"Tell me, and I'll forget. Show me, and I may remember. Involve me, and I'll understand."*
- Studien zu Game-based Learning

Robocode



- Links:
  - http://robocode.sourceforge.net/
  - http://robowiki.net/

Robocode – Spielregeln (1)

- Ziel: feindliche(n) Roboter ausschalten
- Varianten:
  - 1 vs 1, „Melee“, Team (!), TwinDuel
  - Nano, Micro, Mini
- Wettbewerbe:
  - RoboRumble@Home

Robocode – Spielregeln (2)

- Roboter besteht aus:
  - Radar: findet und verfolgt Gegner
  - Kanone: schießt auf Gegner
  - Fahrwerk: bewegt den Roboter

- Roboter können:
  - Radar/Kanone/Fahrwerk drehen
  - Vorwärts/rückwärts fahren
  - Schießen

- Sicht-/Spielfeld ist nicht diskret

Robocode – APIs

- Java-API
  - Scala, Java, Clojure, JRuby...
- .NET-API
  - C++, C#, F#, Visual Basic...
  - leider nur mit Visual XXX, Mono nicht unterstützt
- Objektorientiert:
  - Ableiten von Robot/AdvancedRobot
  - Event Handling

Robocode – Robot API (1)

- Eigenschaften
  - Name, Energy, X, Y, Velocity
  - Heading, GunHeading, RadarHeading
- „Aktionen“
  - DoNothing
  - Ahead, Back
  - TurnLeft, TurnGunLeft, TurnRadarLeft
  - TurnRight, TurnGunRight, TurnRadarRight

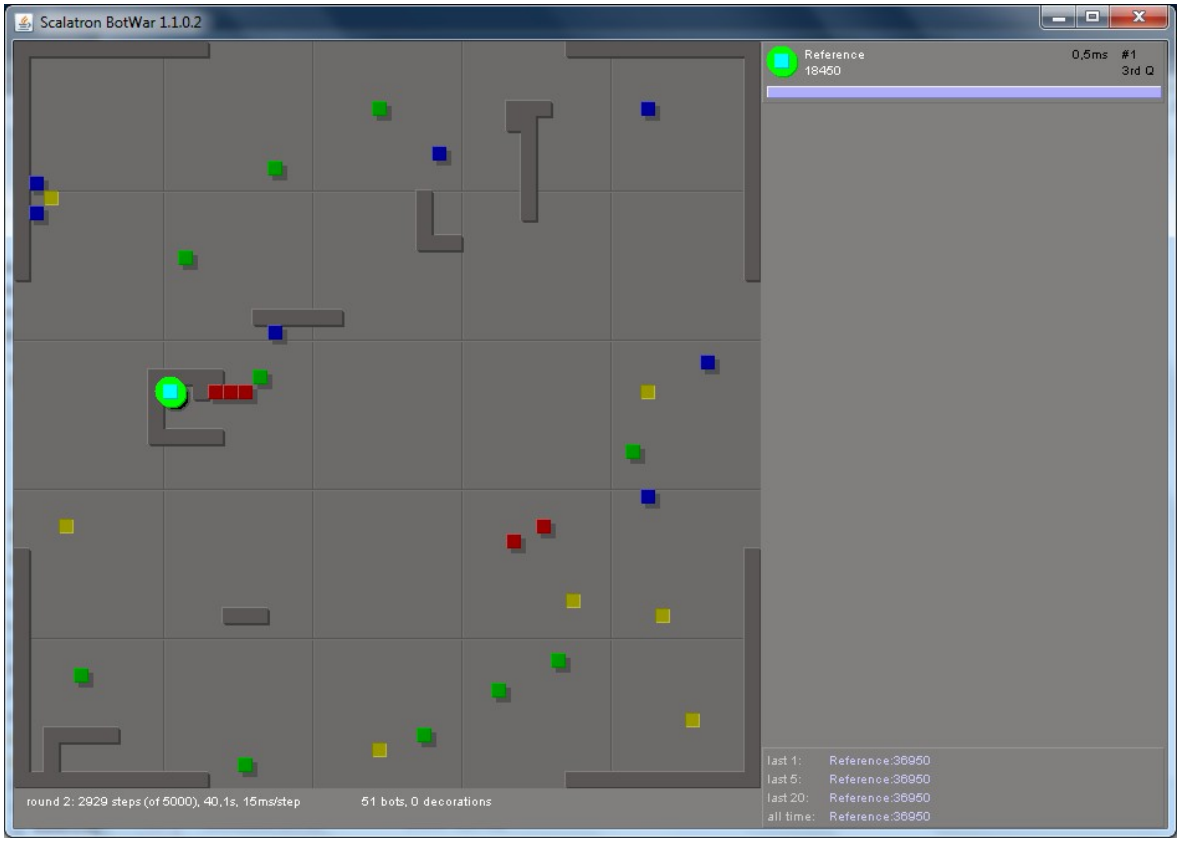
Robocode – Robot API (2)

- Events
  - OnScannedRobot, OnHitWall, OnRobotDeath...
  - OnBulletHit, OnBulletHitBullet, OnHitByBullet...
- Hauptschleife
  - Run

Robocode – Demo

- Wir hacken den **Uber-Bot**... – in **F#**!

Scalatron



- Links:
  - http://scalatron.github.com

Scalatron – Spielregeln (1)

- Ziel:
  - möglichst viel Energie sammeln
- Varianten:
  - „Freestyle“ und Bot vs Bot
  - Spielfeld-Größe/-Begrenzung
- Wettbewerbe:
  - „Freestyle“ Benchmark

Scalatron – Spielregeln (2)

- Es gibt:
  - Bots und Mini-Bots (Spieler)
  - Fluppets und Snorgs (Tiere)
  - Zugars und Toxiferas (Pflanzen)
- Bots/Mini-Bots können:
  - Mini-Bots spawnen und einsammeln
  - Energie sammeln (Fluppets/Zugars)
  - Explodieren (nur Mini-Bots)
- Sicht-/Spielfeld ist diskret

Scalatron – API

- Plain-Text-Protokoll:
  - `Opcode(key=value,key=value,...)|Opcode(...)|...`
  - Beispiel:

    ```
    Move(direction=1:0)|Spawn(direction=0:-1,name="mini1")
    ```

  - auch Rückgabewerte (Karte etc) in diesem Format
- Einstiegspunkt (Scala):
  - `ControlFunctionFactory.create: String => String`
- neuerdings auch Java möglich
  - Aber wer hackt schon freiwillig Java?

Scalatron – Benutzung

- per Web-Interface
  - Verwaltung (für Lehrer/Organisator)
  - Tutorial, Editor und Simulation (für Schüler/Spieler)
  - einfach JAR aufrufen!
- zu Fuß
  - zB per SBT

Scalatron – Demo

- Wir hacken den nächsten **Uber-Bot**... – in **Scala**!

Vielen Dank!

Joachim Hofer

imbus AG

@johofer, http://jmhofer.johoop.de

imbus AG

Spezialisierter Lösungsanbieter für Software-Qualitätssicherung und Software-Test

Innovativ seit 1992

Erfahrung und Know-how aus über 4.000 erfolgreichen Projekten

180 Mitarbeiter an vier Standorten in Deutschland

Beratung, Test-Services, Training, Tools, Datenqualität

Für den gesamten Software-Lebenszyklus

www.imbus.de