
# Vorstellung

Wolfgang Scholz
Raymund Fülöp

05.12.2016

# Historie des Programmierwettbewerbs

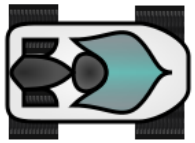

- Java-Programmierwettbewerb
- Echtzeitsimulation
- Grafisches Framework

- Erster Wettbewerb 2002: inspiriert von cRobots
- Passau, München, Regensburg, Nürnberg
- Jedes Jahr ein neues Framework (Internet ist erlaubt)
- www.jrobots.de

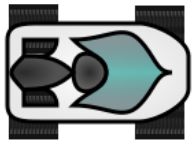
# Ablauf

- Technische Vorstellung des Wettbewerbs
- Pizza-Pause
- Bereitstellung des Frameworks
- Programmierphase
- Abgabe-Deadline
- Turnier
- Preisverleihung

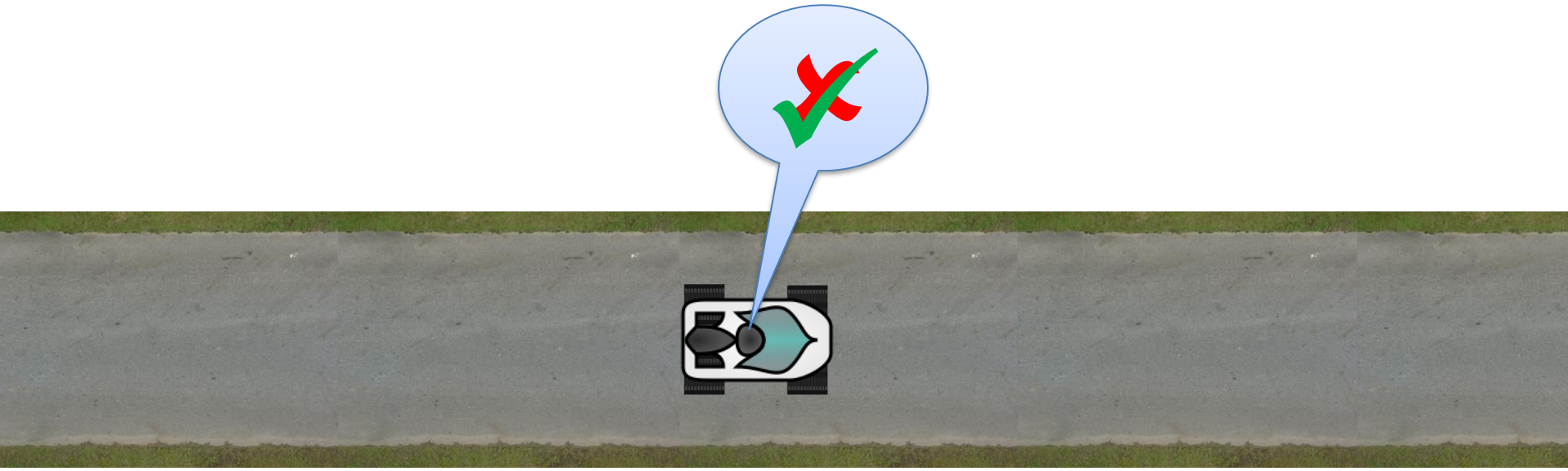
# Jedes Team programmiert einen virtuellen JRobot.



# Seine Bewegung wird physikalisch berechnet.



Wie ein modernes, selbstfahrendes Fahrzeug ist er mit allerhand Sensorik ausgestattet.



Insbesondere fehlt aber eine Kamera.

# Auf der Rennbahn werden die Begegnungen simuliert.

# Wettbewerbsregeln

Turnier am Schluss

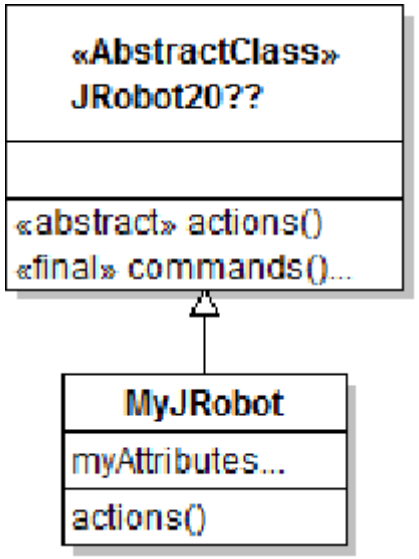
Modus: „Reise nach Jerusalem“

Gewonnen hat, wer als Letzter übrig bleibt

Rennen werden nach der Walltime abgebrochen

# Teilnahmevoraussetzungen

- Systemvoraussetzungen:
  - Java 6
  - Windows / Linux / MacOS
  - 32/64 Bit, kein ARM
- Code in ***einer*** Java-Datei
  - Innere Klassen sind erlaubt.
- Oberklasse **`JRobot2016A`**
  - enthält Methoden und Regeln für diesen Wettbewerb
  - Zur Teilnahme muss von dieser Klasse geerbt werden.



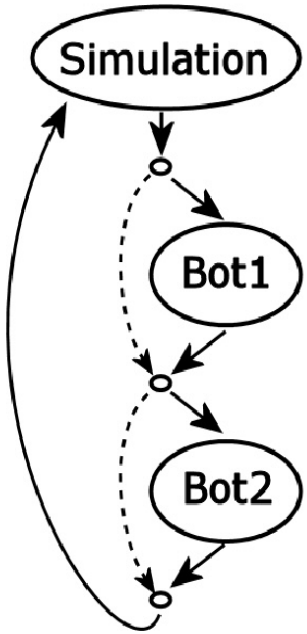
# Programmierung

- Framework wird als zip-Archiv geliefert
- Programmieren in Java IDE (z.B. Eclipse)
- GUI steht immer zur Verfügung
- Einfaches Debugging

Nochmal im Detail:

```
while (#Panzer > 1) {
   für alle Bots: actions()
   Simulationszyklus:
      Projektile spawnen
      Energie lädt sich auf
      Zeit vergeht:
         Panzer fahren
         Projektile fliegen
         Kollisionen finden statt
      Scanner blitzen
}
```

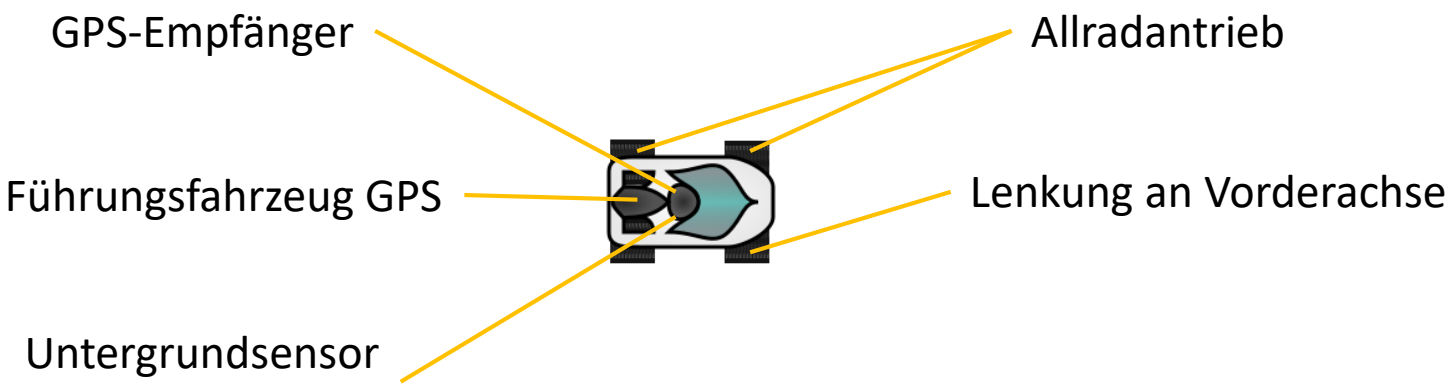
Simulationszyklus



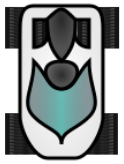
# Der JRobot ist für den Wettbewerb individuell ausgestattet.

**Sensoren**

**Aktoren**



# Fahren



Kinetisches Modell

Trägheit und Beschleunigung
(auch beim Drehen)

Vorwärts schneller als rückwärts

Autopilot steuert Ketten: nur Richtung und Geschwindigkeit anzugeben

Fahren kostet Energie

# Die Rennstrecke

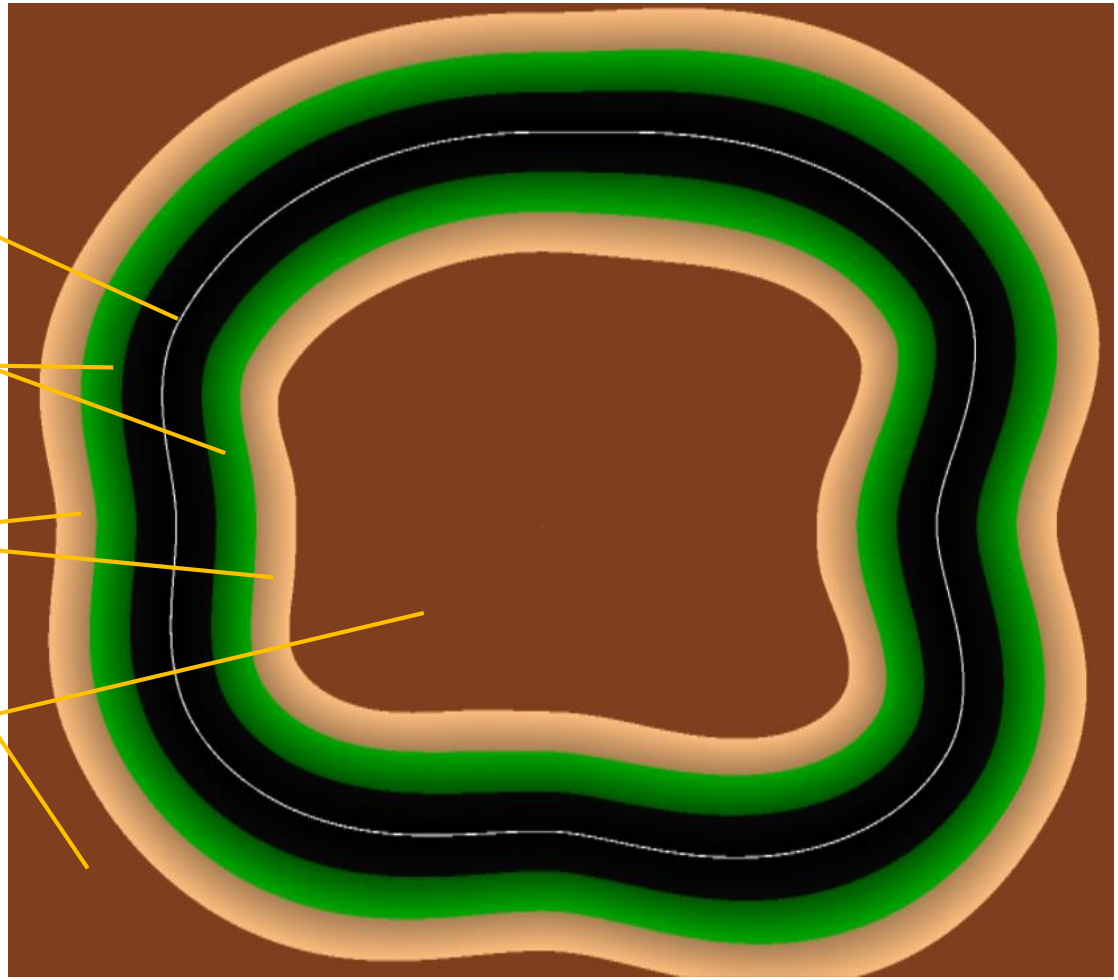
**Bodentypen**

Straße:
griffig, schnell

Gras:
rutschig, matschig

Sand:
locker, steil

Fels:
hart, unpassierbar



**Eigenschaften**

- Geschlossen (Rundtour)
- 2 Runden

**Zufallselemente**

- Kurven
- Ort (GPS)
- Start/Ziel
- Fahrtrichtung

# Speicherverbrauch

In jedem Simulationsschritt wird der Systemzustand neu abgespeichert.

~~Der Speicherverbrauch der Bots wird in der GUI angezeigt.~~

Bitte vermeidet unverhältnismäßig viele Zustandsdaten.

# Säumigkeit

Bots dürfen ihr Zeitkontingent nicht überschreiten.

Bei Zeitüberschreitung oder Exception
wird der Bot „ersetzt“ (neu instanziert).

Der Zustand des Bots vor der Ersetzung geht verloren.

# Tipps

***Konstanten***

Beim Framework-Start werden alle Konstanten ausgegeben.

***get und set***

Alle sensorischen Funktionen beginnen mit get.

Alle Befehle beginnen mit set.

***addDebugLine***

Mit diesem Befehl erscheinen grafische Linien.

***Debuggen***

Mit dem Zeitstrahl kann man zu jedem Frame springen.

Weitersimuliert ab dem Cursor mit *Crop & Simulate.*

Unter Breakpointeigenschaften immer *Suspend VM wählen.*

# Framework Bereitstellung

Das Framework steht zum Download bereit:

www.jrobots.de

Events

2016-12-05

AUCH PER USB-STICK

Wichtig: Die richtige Version heißt

JRobots-Competition-2016-12-05

# Tipps

***Konstanten***

Beim Framework-Start werden alle Konstanten ausgegeben.

***get und set***

Alle sensorischen Funktionen beginnen mit get.

Alle Befehle beginnen mit set.

***addDebugLine***

Mit diesem Befehl erscheinen grafische Linien.

***Debuggen***

Mit dem Zeitstrahl kann man zu jedem Frame springen.

Weitersimuliert ab dem Cursor mit *Crop & Simulate.*

Unter Breakpointeigenschaften immer *Suspend VM wählen.*