

Software Engineering Projekt (SEP) mit ROBOCODE

Klaus Knopper <robocode@knopper.net>

Stand: 2014

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



Kurzbeschreibung

Es wird mit den Methoden des Software Engineering in Teamarbeit ein virtuelles Roboter-Team von Planung (Spezifikation, Design, Gewinn-Strategie, Ablaufplan, ...) bis zur Implementation aufgebaut, und in Form einer gemeinsamen Abschlusspräsentation der Teams sowie eines technischen Handbuchs dokumentiert.

Lernziele:

Praktische Anwendung des Software Engineering, Fähigkeit zum Arbeiten im Team, Fähigkeit zur Dokumentation und Präsentation der Arbeitsergebnisse.

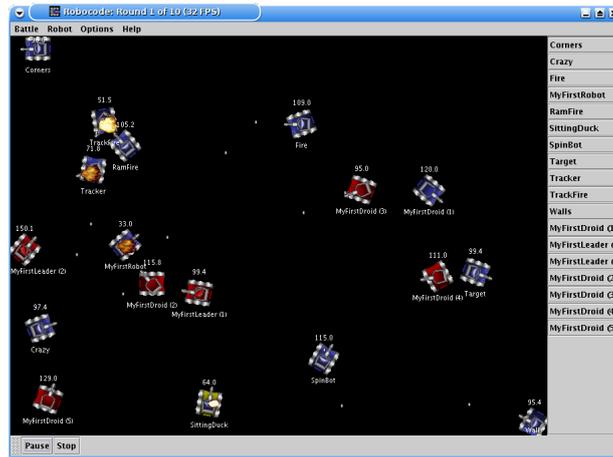
Voraussetzung für die Teilnehmer:

- Grundkenntnisse in JAVA
- Grundlagen Software-Engineering

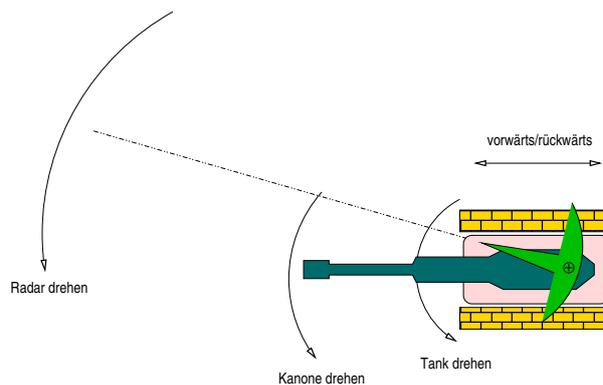
Das Projekt ist Betriebssystem-unabhängig, und kann sowohl auf den Übungs-Notebooks im Multifunktions-Pool, als auch auf mitgebrachten, eigenen Rechnern implementiert werden.

1 Einführung: Robocode

Robocode ist ein OpenSource-Projekt von IBM, das zum Erlernen bzw. Vertiefen der objektorientierten Programmiersprache JAVA eingesetzt wird. Da in Robocode selbst bereits eine umfangreiche graphische Laufzeit- und Entwicklungsumgebung vorhanden ist, mit Verweis auf die ausführliche Beschreibung der API für den Programmierer, sind hiermit auch die im Studium erlernten Methoden der zielgerichteten Softwareentwicklung praktisch einsetzbar.



Robocode ist eine Simulation von Robotern (Tanks), die sich in einer virtuellen Arena einstellbarer Größe bewegen, ihre Umgebung mit Hilfe eines Radars scannen und sich mit Hilfe von Projektilen gegenseitig bekämpfen können. Weiterhin sind Methoden zum Nachrichtenaustausch zwischen „befreundeten“ Robotern, Hilfen zur optischen Kontrolle der Strategieverfolgung (Farbwechsel, Sichtbarmachen der gescannten Umgebung) und Kollisionserkennung vorhanden, die mitverwendet werden können.



Vereinfachte Spielregeln

Jeder Roboter hat zu Beginn der Simulation eine bestimmte Energiemenge¹ zur Verfügung, die durch Bewegung, Kollisionen mit anderen Robotern oder Treffer fremder Projektilen,

¹Diese Anfangsenergie unterscheidet sich zwischen Teamrobots und dem Basis-Robot.

Abfeuern eigener Projektile etc. reduziert wird. Ist die verbliebene Energie 0, explodiert der Roboter, und das Spiel ist für ihn² damit beendet. Gewonnen hat am Ende Team mit der höchsten Punktzahl, die sich unter anderem aus der Anzahl der „überlebten“ Runden zusammensetzt.

²Wenn auch nicht unbedingt für das ganze Team.

2 Aufgabenstellung

2.1 Ziel

Ziel und Aufgabe des Projektes ist es, ein Team von genau 2 Robotern zu **planen**, zu **implementieren** und zu **dokumentieren**. Das Roboterteam soll mit einer gewinnenden Strategie sowohl gegen die von den anderen Kursteilnehmern programmierten Roboterteams, als auch gegen die in der Robocode-Umgebung vorhandenen Beispielroboter und -teams in einer ausreichend großen Arena (meist 1000×1000) erfolgreich antreten können.

2.2 Projektbearbeitung in Teams

Es werden zu Beginn des Kurses Teams von 3 (empfohlene Teamgröße), maximal 4 Personen gebildet. Innerhalb des Teams sollen einerseits alle teilnehmenden Personen mit fest zugewiesenen, speziellen Aufgaben betraut werden, andererseits sollen alle Teammitglieder auch nach Bedarf an anderen notwendigen Teilprojekten (z.B. Implementation, Dokumentation) mitarbeiten. Eine Person aus dem Team wird zum Projektleiter/zur Projektleiterin gewählt, der oder die für das Ressourcen-, Zeit- und Qualitätsmanagement verantwortlich ist. Analyse, Planung, Implementation, Tests und Präsentation können von jeweils einer oder mehreren Personen geleitet werden, speziell an der Abschlusspräsentation sollten alle Teammitglieder beteiligt sein und einen Teil vortragen.

In der Dokumentation (s.S. 5) soll ersichtlich werden, welche Person während des Projektes für welche Aufgabe(n) verantwortlich war, um neben der Gesamtwertung für die Gruppe auch eine personenbezogene Einzelbewertung des Projekts durchführen zu können.

2.3 Methoden

Es sollen die Methoden des Software Engineering angewandt werden, also

1. Planung,
2. Analyse,
3. Entwurf,
4. Programmierung,
5. Tests,
6. ... begleitende Prozesse, insbesondere Dokumentation,

mit Validierung und Verifizierung in geeigneten Phasen des Projektverlaufs.

3 Vorgehen

3.1 Projektplan/Zeitplan

Vom Projektleiter ist in einem frühen Projektstadium in Absprache mit den übrigen Teammitgliedern ein Projektplan zu erstellen, die Aufgaben sind festzulegen und zu verteilen.

Beispielsweise könnte der Projektplan für Woche 1 eine „Orientierungsphase“ vorsehen, um die Robocode Laufzeit- und Entwicklungsumgebung kennenzulernen, und einen ersten Blick in die Beispielprogramme und die Robocode-API zu werfen, Woche 2 für Analyse der Möglichkeiten, die die Robocode-API bietet, Design (Entwurf einer Gewinn-Strategie) und Planung (wer kümmert sich um welchen Teil der Implementation, welche Module sind wann fertigzustellen und zu testen?), Woche 3 für die Implementierung und Tests.

3.2 Analyse und Design

Auch wenn es den einen oder anderen Teilnehmer vielleicht reizt, gleich „draufloszuprogrammieren“, soll die Arbeit an der Software zielgerichtet ablaufen. Es wird zunächst gemeinsam die Strategie, die jeder der Teamroboter verfolgen kann, festgelegt. In der Implementationsphase soll diese Strategie nach Möglichkeit nicht mehr geändert werden. Da hierzu die Kenntnis der Möglichkeiten innerhalb der Laufzeitumgebung Voraussetzung ist, müssen sich die Teilnehmer vorher eingehend mit der Robocode-API, insbesondere der Klasse **TeamRobot** und den darin geerbten Basisklassen und deren Methoden beschäftigt haben.

3.3 Implementation

In der Implementationsphase sollen (Vorgabe), ausgehend von Objekten der Klasse **TeamRobot**, zwei Roboter programmiert werden. Diese sollen sich gegenseitig bei der Erreichung des Ziels (s.S. 3) unterstützen, was z.B. durch geeignete Kommunikations-Schnittstellen erreicht werden kann. Das Team kann in der virtuellen Arena von Robocode getestet werden.

Um fair gegenüber den anderen Projektteams zu bleiben, sollen nur die in Robocode vorhandenen Beispiele als Basis dienen und keine 1:1-Kopien aus dem Internet heruntergeladener „Killermaschinen“, die oft auch kaum verständlich programmiert und unzureichend dokumentiert sind. Sie dürfen aber durchaus Strategien wiederverwenden, die Sie bei anderen Robotern gesehen haben, sofern Sie diese durchschauen und verständlich erklären und dokumentieren können.

3.4 Abschlusspräsentation/Dokumentation

Am letzten Tag des Projektes muss jedes Team

1. Eine Dokumentation
2. Eine Präsentation
3. Einen funktionsfähigen Prototyp bzw. ein fertiges Roboter-Team (**jar**-Package!)

abliefern.

3.4.1 Dokumentation

Die Dokumentation soll, pro Team eine, in elektronischer Form (PDF + eine weiterbearbeitbare Fassung) beim Kursleiter abgegeben werden und enthält:

- Den Namen des Projektes,
- Die Namen der in diesem Team beteiligten Personen und ihre Funktionen für das Projekt,
- Eine **kurze** Einführung („Worum geht es überhaupt?“),
- eine Ausarbeitung, wobei das Pflichtenheft als Gliederungshilfe dienen kann, u.a.:
 - Eine Beschreibung des Vorgehens und der verwendeten Methoden bei der Realisierung des Projektes,
 - Eine Erläuterung der Strategie für jeden programmierten Roboter,
 - Die Umsetzung dieser Strategie, z.B. mit Beispielcode aus der Implementation,
- Ein Fazit: Wurde das Ziel erreicht? Welche Problempunkte sind noch ungelöst? Was könnte man anders/besser machen, wie hoch ist der Aufwand dafür?

3.4.2 Präsentation

Das Team stellt sein abgeschlossenes Projekt in Form einer Kurzpräsentation (max. 20 Minuten, plus 10 Minuten Diskussion) vor. Hierbei kann der Beamer oder die Tafel mitverwendet werden. Falls Live-Demonstrationen am Rechner gezeigt werden, sind diese im Vorfeld so vorzubereiten, dass keine zusätzliche Zeit für die Technik aufgewandt werden muss.

3.5 Demonstration

Nach Abschluss aller Präsentationen erfolgt (als Demonstration der Ergebnisse) eine Gegenüberstellung aller Teams in einer gemeinsamen, großen Robocode-Arena in 10 Runden, um das Team mit der „besten“ Strategie zu ermitteln. Hierzu ist dem Kursleiter ein Snapshot (**jar**-Export) des programmierten Roboter-Teams zur Verfügung zu stellen.

Während oder nach der Demonstration erfolgt eine Abschlussdiskussion mit allen Teilnehmern.