

Entwicklung einer Java Simulationsumgebung für Lego Mindstorms NXT Roboter

Exposé - Version 2

Pamina Maria Berg
Matr.Nr. 6087438

5. Januar 2016

Motivation

Die Arbeit mit Robotern bereichert den Informatikunterricht und das Nachmittagsprogramm vieler Schulen seit Jahren. Auf spielerische Art und Weise sollen Schülerinnen und Schüler (im Folgenden SuS abgekürzt) mit einfachen Konstrukten der objektorientierten Programmierung umzugehen lernen. Dies geschieht zum einen mit Drag-and-Drop Softwareangeboten, wie das standardmäßig mit ausgelieferte NXT Mindstorms Tool oder auch Enchanting. Zum anderen bietet sich ab der Mittelstufe (Klasse 7 – 10) die Arbeit mit BlueJ zur Erstellung erster selbstgeschriebener Programme an. Hierzu kann eine BlueJ Extension genutzt werden, die mit der Java Virtual Machine leJOS NXJ für NXT Robotern arbeitet.

Doch immer wieder stoßen Lehrkräfte in den Schulen auf Hardwareprobleme jeglicher Art, wie zum Beispiel das Fehlen von einer ausreichenden Anzahl an Robotern im Unterricht, fehlende Firmwareupdates oder defekte Sensoren.

Auch ist das Zusammen- und wieder Auseinanderbauen der Roboter ein Aufwand, der nicht für den alltäglichen Unterricht geeignet sind. Um kleine Aufgaben mit Java zu lösen, wie zum Beispiel das Fahren einer S-Kurve oder das Anhalten auf einem bestimmten Punkt, muss bisher immer erst der Roboter gestartet, der Code in BlueJ verfasst, auf den Roboter übertragen und dieser dann zu einem geeigneten Parcours gebracht werden.

Um diese Probleme zu umgehen und den Einstieg in die objektorientierte Programmierung über die Benutzung von LEGO Mindstorm Robotern zu ermöglichen, soll im Rahmen dieser Masterarbeit eine Simulationsumgebung für die Arbeit mit LEGO Mindstorms NXT Roboter entstehen.

Hintergrund / Entwicklung der NXT Roboter

Die ersten computergesteuerten LEGO Produkte wurden bereits 1986 veröffentlicht. In einer Zusammenarbeit von LEGO Education und dem Massachusetts Institute of Technology (MIT) wurde LEGO TC LOGO entwickelt. Dies war eine spezielle Abwandlung der Programmiersprache LOGO, mit der zusammengesetzte LEGO-Modelle gesteuert werden konnten [Rol14].

Die Entwicklung eines programmierbaren LEGO-Steins begann 1988 und erreichte ihren Höhepunkt mit der Vorstellung des ersten MINDSTORMS Systems im Januar 1998, bei der der LEGO MINDSTORMS RCX Intelligent Brick, ein Microcomputer und somit das Kernstück des RCX-Systems, und das Robotics Invention System im Museum of Modern Art in London vorgestellt wurden.

Bereits zwei Monate nach Verkaufsstart wurde die FIRST LEGO League (FLL) gegründet – eine Zusammenarbeit zwischen LEGO und FIRST (For Inspiration and Recognition of Science and Technology), die den Grundstein für die heute noch bestehende Wettbewerbsliga legte [Rol14].

Die Vorstellung und der Verkaufsstart der Nachfolge-Roboter des RCX-Systems, den LEGO MINDSTORMS NXT Robotern, fand im August 2006 statt. Diese damals neu entwickelten Roboter sind auch, dank eines Updates in 2009, fast zehn Jahre später noch in den Schulen und Universitäten zu finden. Im April 2005 fand die erste FLL Weltmeisterschaft in Atlanta, Georgia, statt und bis heute bieten die Weltmeisterschaften einen Anlaufpunkt für Jugendliche auf der ganzen Welt, die ihr Können und ihre Roboter auf die Probe stellen wollen [Lego].

Fragestellung

In dieser Masterarbeit soll der Prototyp einer Simulationsumgebung entwickelt werden. Diese Simulationsumgebung soll auf Basis der von leJOS NXJ angebotenen API den SuS eine Möglichkeit zur schnellen Überprüfung ihrer Lösungen durch eine Simulation am Computer geben.

Des Weiteren soll im Laufe der Implementation festgestellt werden, inwiefern eine Integration der Simulationsumgebung in BlueJ mithilfe einer Extension sinnvoll und realisierbar ist.

Abschließend wird die Arbeit einen kleinen Anteil an didaktischen Inhalten umfassen. Dieser soll das Forschungsthema in einen schulischen Kontext einordnen und somit die Anforderungen, die an eine solche Simulationsumgebung gestellt werden, aus zwei Perspektiven darstellen.

Forschungsfragen:

-
- Welche Anforderungen werden an eine solche Simulationsumgebung gestellt - aus Schülerperspektive, aus Lehrerperspektive, aus Entwicklerperspektive?
 - Welche Integrationsmöglichkeit bietet BlueJ und wie kann diese sinnvoll genutzt werden?

Lösungsidee

Die generelle Idee ist es, eine eigene API für den Simulator zu programmieren, die exakt die gleichen Methoden anbietet, wie die leJOS NXJ API. Somit wäre gewährleistet, dass die SuS eine einheitliche Syntax lernen und nicht zwischen Methoden wechseln müssen, um zum Beispiel die beiden Motoren zu starten oder die Sensoren anzusteuern. Hierbei soll ein Roboter als Objekt in einer Bitmap, die aus einer Pixelgrafik erzeugt wird und einen Parcours darstellt, auf dem Bildschirm angezeigt werden und die programmierte Lösung der SuS auf diesem Parcours simulieren.

Geplante Aktivitäten

Zunächst verschaffe ich mir einen Überblick darüber, welche Klassen implementiert werden müssen, um ein Mindestmaß an Funktionen der in Simulationsumgebung sicherzustellen. Hierzu gehören die Klassen der Motor- und Sensor-Ports, eine Klasse für die Motoren und jeweils eine Klasse für die Licht-, Ultraschall- und Tastsensoren.

Ich werde zunächst drei Szenarien als Pixelgrafiken erstellen, die dann von einer Bild-einleserklasse als Parcours generiert werden können. Um nun die Visualisierung des Roboters als Akteur in dem Problem-Szenario zu verwirklichen, werde ich diesen zunächst als einfaches Dreieck implementieren.

Ein Problem wird darin bestehen, dass der Roboter an sich bei der leJOS NXJ API nicht als eigenständiges Objekt gesehen wird, sondern die beiden Motoren A und B zur Steuerung des Roboters direkt angesprochen werden. Das Roboter-Objekt der Simulation muss also so mit den beiden Motoren verbunden werden, dass dieses ein visuelles

Feedback geben kann. Sobald der rechte Motor des Roboters angesprochen wird, sollte die visuelle Repräsentation auf dem Parcours direkt darauf reagieren. Zudem müssen die Sensoren an das Roboter-Objekt „angebaut“ werden, was eine weitere Schwierigkeit darstellt.

Zeitplan

Oktober 2015

- Herstellung der Parcours-Pixelgrafiken
- Übersicht über die zu verwendenden Klassen verschaffen
- Einleitung der Ausarbeitung sowie ersten geschichtlicher Hintergrund

November 2015 – Dezember 2015

- Implementation des Roboter-Objekts und der ersten Klassen
- Festhalten der Implementationsschritte mithilfe von Git
- Theoretischen Hintergrund sowie Anforderungen an die Implementation in der Ausarbeitung niederschreiben
- Anmeldung der Arbeit

Weihnachtsferien 2015

- Fertigstellung der Implementation
 - Entscheidung über Einbettung der Oberfläche in BlueJ als Extension
- Testung der Simulationsoberfläche
- Implementationsschritte sowie Beschreibung der Software in die Ausarbeitung einpflegen

Anfang Januar 2016

- Zwischenstandspräsentation im Abschlussarbeitenseminar
- Abgabe erster Entwurf der Ausarbeitung

Januar 2016 – Anfang Februar 2016

- Letzte Korrekturen in der Implementation
- Einbau des Feedbacks aus der Präsentation
- Erste Verbesserung der Ausarbeitung

Mitte Februar 2016

- Vorbesprechung zur Abgabe der Ausarbeitung
- Einbau der Verbesserungsvorschläge

Ende Februar / Anfang März 2016

- Abgabe der Masterarbeit
- Anmeldung zur mündlichen Abschlussprüfung
 - Ggf. weitere Anträge stellen

März 2016

- Vorbereitung der mündlichen Abschlussprüfung

1. April 2016

- Mündliche Abschlussprüfung zur Masterarbeit / Kolloquium

LITERATURVERZEICHNIS

- [Abe01] Michael Abend. “Robotik und Sensorik. Darstellungsschwerpunkt: Selbständige Entwicklung „unscharfer“ Algorithmen zur räumlichen Orientierung (unter Verwendung des LEGO-Mindstorms-Systems)”, *Schriftliche Prüfungsarbeit zur zweiten Staatsprüfung für das Amt des Studienrats*, Berlin, 2001
- [Ber10] Karsten Berns, Daniel Schmidt. *Programmierung mit LEGO MINDSTORMS NXT. Robotersysteme, Entwurfsmethodik, Algorithmen*, Springer Heidelberg Dordrecht London New York, 2010
- [Her12] Joachim Hertzberg, Kai Lingemann, Andreas Nüchter. *Mobile Roboter. Eine Einführung aus Sicht der Informatik*, Springer-Verlag Berlin Heidelberg, 2012
- [Hub07] Peter Hubwieser. *Didaktik der Informatik*, 3. Auflage, Springer-Verlag Berlin Heidelberg, 2007
- [Lego] o.V. URL: <http://www.lego.com/en-us/mindstorms/history>, Abgerufen am 06.11.2015, LEGO, 2015
- [leJOS] o.V. URL: <http://www.lejos.org/nxj.php>, Abgerufen am 14.12.2015, leJOS Java for Lego Mindstorms, 2015
- [Nie99] Jürg Nievergelt. *Roboter programmieren ein Kinderspiel. Bewegt sich auch etwas in der Allgemeinbildung?*, Informatik-Spektrum 22:5, S. 364–375, 1999
- [Rol14] Mark Rollins. *Beginning LEGO MINDSTORMS EV3*, Apress, Berkeley, CA, 2014

- [Sch04] Rafael Schreiber. "Der Einsatz von LEGO-Mindstorms im Informatikunterricht der 11. Klasse der Leonard-Bernstein-Oberschule. Sicherung und Transfer grundlegender algorithmischer Strukturen in NQC.", *Schriftliche Prüfungsarbeit im Rahmen der zweiten Staatsprüfung für das Amt des Studienrats*, Berlin, 2004
- [Sto01] Matthias Stolt. "Roboter im Informatikunterricht", 2001
- [Wag05] Oliver Wagner. "LEGO Roboter im Informatikunterricht. Eine Untersuchung zum Einsatz des LEGO-Mindstorms-Systems zur Steigerung des Kooperationsvermögens im Informatikunterricht eines Grundkurses (12. Jahrgang, 2. Lernjahr) der Otto-Nagel-Oberschule (Gymnasium)", *Schriftliche Prüfungsarbeit im Rahmen der zweiten Staatsprüfung für das Amt des Studienrats*, Berlin, 2005