Institut für Informatik
Lehrstuhl für Programmierung und Softwaretechnik

LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Bachelorarbeit

Minecraft in MicroPsi

Eine populäre Videospielwelt als Simulationsumgebung für eine kognitive kuenstliche Intelligenz

Jonas Kemper

Medieninformatik Bachelor

Aufgabensteller:	Prof. Dr. Martin Wirsing
Betreuer:	Joscha Bach PhD, Annabelle Klarl
Abgabetermin:	19. September 2013

Ich versichere hiermit eidesstattlich, dass ich die vorliegende Arbeit selbstständig angefertigt, alle Zitate als solche kenntlich gemacht sowie alle benutzten Quellen und Hilfsmittel angegeben habe.

München, den 19. September 2013

. . . .

(Unterschrift des Kandidaten)

Zusammenfassung

Ziel der Arbeit ist die Entwicklung und das Testen einer Simulationsumgebung für einen kognitiven Agenten auf Basis des populären Videospiels Minecraft.

Minecraft bietet sich als Grundlage für eine Simulationsumgebung aufgrund der kompositionalen Semantik der Spielwelt besonders an, da das Agentensystem durch das Erforschen seiner Umwelt Wissen über die Spielwelt aufbauen und ähnliche Strukturen wieder erkennen kann. Objekte der Spielwelt sind in Minecraft keine bloßen Hindernisse sondern werden mit variablen Eigenschaften prozedural erzeugt und ähneln so eher einer realen Umgebung als andere virtuelle Welten.

Da Minecraft von Anfang an mit einem Mehrspielermodus ausgestattet wurde, lässt es sich zudem für Multiagenten-Umgebungen und so für kollaborative Agenten verwenden. Des Weiteren sind Minecraft-Lizenzen günstig zu erwerben, erhältlich für viele Plattformen und es gibt eine äußerst große und aktive Community für selbsterstellte Spiel-Inhalte und -Modifikationen.

Angestrebt wird, dass die kognitive Architektur MicroPsi2 sich an einen Minecraft-Server anmelden kann, ihre Umgebung wenigstens in Ansätzen wahrnimmt (Objekte, Terraintypen), sich fortbewegen kann und einfache Interaktionsmöglichkeiten besitzt (z.B. Objekt aufnehmen und ablegen).

Daneben soll eine Visualisierung der Umgebung aus Sicht des Agenten entstehen, z.B. als zweidimensionale Übersicht, in deren Zentrum der Agent steht. Diese Visualisierung soll live im Browser angezeigt werden (wahlweise mit WebGL oder Canvas/D3).

Zum Schluss der Arbeit soll zum Testen der Funktionalität ein virtuelles Braitenbergvehikel in die Simulationsumgebung gesetzt werden, welches sich daraufhin auf die nächstgelegene Lichtquelle zubewegen soll. Dieses Experiment wird als Teil der Arbeit umfangreich dokumentiert.

Abstract

What I cannot create, I do not understand Richard P. Feynman, 1988

Danksagung

Ich danke meinen Betreuern Joscha Bach und Annabelle Klarl sowie Dominik und Professor Wirsing.

Contents

Chapter 1

A! Introduction

The hunt for artificial intelligence started many years ago. Dividing the subjects into the strong and weak parts means seperating it goals into usefull applications and those that try to learn about the nature of intelligence itself. The ultimate task of strong A.I. — recreating human intelligence — admitably still seems to be science fiction, though.

A new generation of cognitive scientists, psychologists and computer scientists strives to implement new Ideas for simulated cognition — building cognitive architectures. Many of them do so by simulating in one way or the other, what could be called "neural networks".

To test the functionality of such cognitive architectures and to figure out their capabilities and potential we need to research their behavior inside defined environemts. As implementing AI into the physical world (as robots for examples) requires building apropriate hardware and patience, computer-simulated environments are an important part of that research.

MicroPsi is both: a cognitive architecture and also a set of and an interface to simulation environments.

1.1 Motivation

Even though there have been more complex simulation enviroments (e.g. 3D-worlds) for previous implementations of Psi-architectures, the relatively new MicroPsi 2 has only two fairly simple ones: a 2D-Island and a map of the public transportation system of Berlin. Instead of building a new 3D-world, we set out for something a little more innovative.

Video games are natural applications of artificial intelligence. The quality of a games A.I. can make all the difference in between a great title and an unenternaining demo of computer graphics.

One computer-game in particular stood out in the last years — not for A.I. reasons though. It is called Minecraft and is benefiting of high popularity ever since its first release in 2009. Since copies of the game can be obtained commercially for the first time in 2011, the different versions of the game sold more than 26 million times - the PC version priced at about 20 Euros. It should be noted, that the games developer studio Mojang is a so called "Indie" developer that is not associated with any classical game publisher but distributes copies of their game exclusively via their own website.

Even though the game can be downloaded and played as a single packet of software, many scenarios of playing the game consist of running a Minecraft server software as well as one client per player. It is possible to mimic the official client by implementing the reverse-engineered Client-Server-Protocol and build artificial players that way.

Minecraft is a complex yet easily accesible virtual world. It is constantly developed and new features are added regularly. It is a massive fanbase and a huge community of gamemodifications.

Another interesting aspect about Minecraft is the procedural semantic the game world is generated with and. Trees in Minecraft, for example, may share a similar structure that consists of a trunk and branches and leaves spreading out as fractals, but the particular charecteristics of each tree are generated randomly. This makes a Minecraft world somewhat more realistic than most other videogames.

1.2 Objective

The objective of this thesis is to build and test an interface in between MicroPsi and Minecraft, so that a Minecraft world (e.g. server) can be used as a simulation environment for the MicroPsi 2 Framework, which will act as an artificial player.

That being said, a big part of the project is about visualization. Inside the MicroPsi Core Application, a 3D-visualization of the Minecraft world and the agent within is aimed for. There are two main reasons for this goal. The first reason is, that the agents behavior within the simulation environment is supposed to be monitored from the MicroPsi webinterface in an aesthetically appealing way. The secons reason is, that the image data is supposed to be processed by the agent as one of it's senses.

The functionality is to be tested with a simple Braitenberg-vehicle experiment.

In the future, multiple agents shall interact with the same environment and collaborate with each other.

1.3 Approach

Do obtain these goals, the Minecraft Client-Server-Protocol had to be researched, learned and imitated.

Then, artificial Minecraft players (written in Python) had to be searched, found and researched.

Eventually, building upon an existing Bot framework led to an integration with the MicroPsi framework.

1.4 Outcome

After several iterations and trying out different approaches and technologies the Interface is now functional.

The experiment with the simulated Braitenbergvehikel resulted in proofing that Minecraft is usable as a simulation environment.

1.5 Outline

...

Chapter 2

A! Background

... short History of A.I. ... ... DeepBlue / Watson ... ... Cognitive A.I. ...

2.1 A! Artificial Intelligence / Examples for A.I. applications

... examples for A.I. ... ... related work ...

2.1.1 weak AI

2.1.2 strong AI

2.2 A! Cognitive AI / cognitive AI

... examples for cognitive AI ...

2.2.1 Concepts

... Approaches to cognitive AI ... ... neural node nets ... ... related work ...

2.3 A! Psi Theory

... Basics of Psi Theorie of Dörner ...

2.3.1 Joschas Contribution: MicroPsi

... explanation of Joschas Dissertation ...

2.4 Summary

... still a lot to do in AI ...

Chapter 3

A! Foundations / Psi Implementationen

... Psi has been implemented by different groups ...

3.1 A! Psi Implementations

3.1.1 Dörners Implementation

... Dörner implemented it in Pascal ...

3.2 Joschas Implementation

"The cognitive architecture MicroPsi builds on a framework for simulating agents as neuro-symbolic spreading activation networks. These agents are situated in a simulation environment or fitted with robotic bodies. The current implementation of MicroPsi has been re-implemented from the ground up and is described here." [Bac12]

3.2.1 microPsi in Java

"The first implementation of the MicroPsi framework spanned the years 2003 to 2009, and was built in Java as a set of plugins for the Eclipse IDE. The graphical edi- tor was built on SWT. It comprised about 60000 lines of code, and although a lot of effort went towards platform independence (with the exception of a DirectX/.Net based 3D viewer component), deployment on the various operating systems and across several versions of Eclipse became support intensive, especially after its adop- tion by teams outside of our group." [Bac12]

... Joscha implemented it in Java ...

3.2.2 microPsi in Python

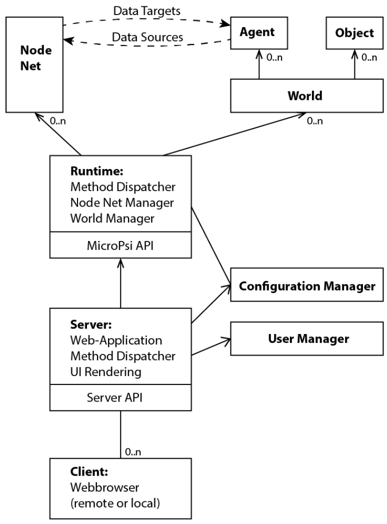
"Gradual changes in the formalization of MicroPsi and the emergence of new soft- ware development methodologies and tool chains, especially the move from Java design patterns and XML tools towards lightweight and agile Python code, prompted a complete rewrite of the MicroPsi framework, starting in 2011. The following sec- tion describes the overall structure of the framework, followed by detailed definitions of the node net formalism and the structure of simulation worlds that enable running MicroPsi agents.” [Bac12]

The MicroPsi 2 user interface is rendered completely inside a web browser and the simulation is deployed as a web application. The UI components are based upon HTML/Javascript and ”and facilitates the communication between the browser based renderer and the agent simulator via JSON and JSON remote procedure calls. Rendering is supported by Twitter’s widget library Bootstrap (2012) and the Javascript library PaperJS (Lehni and Puckey, 2011).” [Bac12]

... then in Python with a Webinterface ...

3.2.3 Module Overview / Architecture

... it consists of a Core an a Server module with different threads running ...



3.2.4 Core

... the core runs the heart of the simulation ...

3.2.5 Server

... the server provides the interface ...

3.2.6 Simulation Environments

... so far, there are an ”Island” and an ”Berlin” worlds ...

3.3 A! Minecraft

... a more complex simulation environment could be fun ...

... the story of Minecraft ... ... Minecrafts poularity (and demographics) ...

3.3.1 What is Minecraft?

... brief description of the basic mechanisms ...

3.3.2 The Cient Server Protokoll

... the language an external client needs to speak, to take place in a Minecraft world ...

3.3.3 Suitability of Minecraft as a simulation environment

... cheap licenses ... ... developer friendly community and game-studio ... ... sandbox game with many possibilieties but no pre-defined goals ... ... procedural semantic ...

3.4 A! Minecraft and MicroPsi

... a more complex simulation environment could be fun ...

Chapter 4

A! Approach / Minecraft as a Simulation Environment for MicroPsi 2

4.1 A! Overview / What has been there so far?

... Minecraft Bots with simple as well as sophisticated AI ... ... MicroPsi 2 with Island and Berlin world ...

4.2 A! Architecture / Building the interface in between Minecraft and the simulation environment

... result: a Minecraft Bot that implements MicroPsi AI and is controlled and monitored via the MicroPsi webinterface ... ... the Webinterface holds its own visualization of the Agents worldview ...

4.2.1 A Minecraft Bot

4.2.1.1 Protocol Implementation

4.2.1.2 Control Structures

4.2.1.3 Previous own implementations with TwistedBot

4.2.1.4 other popular Bot projects and game modifications

4.2.1.5 Spockbot von Nickelpro

4.3 Implementation

4.4 Case Study

Chapter 5

A Minecraft Bot

5.1 Protocol Implementation

5.2 Control Structures

5.3 Previous own implementations with TwistedBot

5.4 other popular Bot projects and game modifications

5.5 Spockbot von Nickelpro

Chapter 6

Implementing the Bot in MicroPsi

6.1 The MicroPsi side

6.2 necessary Modifications and Additons in core/worldrunner

6.2.1 Data Targets and sources

6.3 necessary Modifications and Additons in server/control and monitoring interface

Chapter 7

The Visualization

7.1 Requirements and necessity of a visualization

... why a visiualization and what is it supposed to do? ...

7.1.1 monitoring the bot from the webinterface

... make it aesthetically appealing as well as easily accessible ...

7.1.2 using visualization output as a Datasource / as the bots eyes

7.2 Implementation

7.2.1 used Data

... required Data for the visualization ... ... and how to obtain it (first attempts: telnets/then sockets) ...

7.3 3D Visualisierung mit Pyglet

... foundation: that minecraft pyglet clone ...

7.4 Earlier attempts using JavaScript / AJAX

... worked well but a little slow ...

Chapter 8

Experiment

... experiment to test functionality of the system ... ... scope: only a simple test for time reasons ...

8.1 Braitenberg Vehicle

... simplest proof of concept of a microPsi agent ...

Chapter 9

Conclusion

9.1 What's next

... what has been learned ... ... what can be done with the new environment ... ... what can be improved? ... ... what other simulation environments could be of interest? ...

Appendix A

Auszug aus dem Buch

Beispielhaft wird hier gezeigt, wie Block-Zitate eingeführt werden können. Hier der Beginn des Buchs "'Per Anhalter durch die Galaxis"' von Douglas Adams [Ada98]:

> "'Das Haus stand auf einer kleinen Anhöhe genau am Rand des Ortes. Es stand alleine da und überblickte das weite Ackerland im Westen. Absolut kein bemerkenswertes Haus - es war ungefährt dreißig Jahre alt, plump, viereckig, aus Ziegelsteinen erbaut und hatte vier Fenster an der Vorderseite, der es nach Größe und Proportion mehr oder weniger mißlang, das Auge zu erfreuen."'

Appendix B

Implementierungen

Beispielhaft wird hier gezeigt, wie Code-Beispiele in den Text eingefügt werden können. Die Pseudocode-Umgebung wird von *macros.tex* bereitgestellt und kann dort entsprechend angepasst werden.

```
public static Object answeringMachine() {
    Thread.sleep(1000);
    return 42;
}
```

Figure B.1: *Implementierung einer Maschine zur Beantwortung der Fragen aller Fragen.*

List of Figures

List of Tables

Inhalt der beigelegten CD

Die beigelegte CD enthält folgenden Inhalt:

- diese Masterarbeit in PDF Format,
- Videos mit Interview von Fans,
- den Source-Code der Implementierung einer Maschine zur Beantwortung der Fragen aller Fragen. Der Source-Code ist im Ordner *src* zu finden.

Bibliography

[Ada98] In: ADAMS, Douglas: *Per Anhalter durch die Galaxis: Roman*. Heyne Verlag, 1998, S. 1

[Bac12] BACH, Joscha: MicroPsi 2: The Next Generation of the MicroPsi Framework. In: BACH, Joscha (Hrsg.) ; GOERTZEL, Ben (Hrsg.) ; IKLÉ, Matthew (Hrsg.): *AGI* Bd. 7716, Springer, 2012 (Lecture Notes in Computer Science). – ISBN 978–3–642–35505–9, 11-20