

„4 gewinnt“ mit humanoidem Roboter

für 2 Bachelorstudierende als Bachelorarbeit oder einen Masterstudierenden als Masterarbeit

Seit Ende 2013 existiert in der Fachrichtung „AI“ ein „4 gewinnt“-spielender Roboterarm. Die Implementierung als Ergebnis der Zusammenarbeit von Prof. Sahn, Prof. Zerbe und Dr. Kern wurde sowohl bei der langen Nacht der Wissenschaft als auch auf der CeBIT präsentiert. Zieht der Roboterarm an, hat man als Kontrahent kaum eine Chance zu gewinnen. Bei der aktuellen Implementierung wird mit Stift auf Folie gespielt, so dass die Schwerkraft durch die Festlegung unzulässiger Felder simuliert werden muss.

Das vorliegende Thema soll prüfen, inwieweit der seit Februar 2014 vorhandene humanoide Roboter der Serie NAO die Rolle des Eigenbaus übernehmen kann. Dazu ist es erforderlich, im Wesentlichen 4 Aufgabenstellungen zu analysieren, zu strukturieren und geeignete Lösungen zu implementieren:

1. **Kommunikation des Spielerwechsels**
(Spracherkennung und Spracherzeugung)
2. **Erkennung des aktuellen Spielzustandes**
(Bilderfassung und Bildverarbeitung)
3. **Berechnung des eigenen Spielzuges**
(Modellierung und Implementierung des Spielzustandes, von Spielstrategien, Bewertungen und Suchverfahren)
4. **Ausführung des errechneten Spielzuges**
(Motionplanung und Kinematik)



Trotz der vorhandenen Vorarbeiten ist es aufgrund des Analyse- und Arbeitsumfangs möglich, die Aufgabenstellung durch zwei Bachelorstudierende zu bearbeiten. In einem solchen Fall können die Aufgabenstellungen 1 und 2 sowie 3 und 4 jeweils als Paket bearbeitet werden. Durch diese Teilung erforderliche Schnittstellen sind klar während der Designphase zu spezifizieren und festzuhalten.

Als Programmiersprache unterstützt der NAO-Roboter nativ Python und C. Python ist dabei eine recht einfache Skriptsprache, die vor allem im Umfeld von KI-Anwendungen häufig eingesetzt wird. Standardmäßig wird der NAO-Roboter mit einer sehr umfangreichen Python-Bibliothek ausgeliefert, die leicht erweitert und modifiziert werden kann.

Der theoretische Teil der Arbeit sollte sich deshalb mit verfügbaren Implementierungen bzw. Algorithmen im jeweiligen Bereich auseinandersetzen und daraus die Konzeption für die eigene Implementierung ableiten. Die Umsetzung schließt ein Software-Design, die Programmierung und eine Funktionsdemonstration ein. Sinnvoll ist es auch, zur Funktionsdemonstration rechtzeitig geeignete Testfälle – vor allem zur Integration mit dem anderen Arbeitspaket – in der Arbeit zu entwickeln und das Ergebnis der Testdurchführung zu dokumentieren.