



LEGO HUNTS THE WUMPUS

- TEAM 0 -

Software-Entwicklungspraktikum (SEP)
Sommersemester 2012

P f l i c h t e n h e f t



Auftraggeber:

Technische Universität Braunschweig
Institut für Programmierung und Reaktive Systeme
Prof. Dr. Ursula Goltz
Mühlenpfordtstraße 23
38106 Braunschweig

Betreuer: Benjamin Mensing

Auftragnehmer:

Name	E-Mail-Adresse
Lars Luthmann	l.luthmann@tu-bs.de
Marcel Mast	m.mast@tu-bs.de
Christian Pek	c.pek@tu-bs.de
Yevgen Pikus	y.pikus@tu-bs.de
Tobias Maximilian Vogt	t.vogt@tu-braunschweig.de

Braunschweig, 02. Mai 2012

Versionsübersicht

Version	Datum	Autor	Status	Kommentar
0.1	23.04.2012	Luthmann, Pikus, Pek	in Bearbei- tung	erste vollständige Version
0.2	26.04.2012	Luthmann, Pi- kus, Pek, Vogt, Mast	in Bearbei- tung	Überarbeitungen in allen Berei- chen
1.0	27.04.2012	Luthmann, Pikus, Mast, Vogt	in Bearbei- tung	erster Release Candidate
1.1	30.04.2012	Luthmann, Pi- kus, Pek, Vogt, Mast	in Bearbei- tung	Verbesserungen in allen Berei- chen
1.2	01.05.2012	Luthmann, Pi- kus, Pek, Vogt, Mast	in Bearbei- tung	Fertigstellung des Dokuments
2.0	02.05.2012	Luthmann, Pi- kus, Pek, Vogt, Mast	abgenommen	fertige Version

Inhaltsverzeichnis

1 Zielbestimmung	5
1.1 Musskriterien	6
1.2 Wunschkriterien	7
1.3 Abgrenzungskriterien	7
2 Produkteinsatz	8
2.1 Anwendungsbereiche	8
2.2 Zielgruppen	8
2.3 Betriebsbedingungen	8
3 Produktübersicht	9
4 Produktfunktionen	12
5 Produktdaten	22
6 Nichtfunktionale Anforderungen	23
7 Benutzeroberfläche	25
8 Technische Produktumgebung	27
8.1 Software	27
8.2 Hardware	27
8.3 Orgware	27
8.4 Produktschnittstellen	28
9 Glossar	29

Abbildungsverzeichnis

1.1	Beispiel für Aufbau des Spielfelds	5
3.1	Use-Cases - Aktionen des Agenten	9
3.2	Uses-Cases - Wahrnehmungen des Agenten	9
3.3	Use-Cases - Server	10
3.4	Use-Cases - Server und Benutzer	10
3.5	Use-Cases - Wumpus	11
7.1	Startfenster	26
7.2	Fenster zum Eingeben der Karte	26
7.3	Fenster zum Laden der Karte von einer Datei	26

1 Zielbestimmung

Ziel dieses Projektes im Rahmen des Softwareentwicklungspraktikums 2012 ist die Simulation des Computerspiels *Hunt the Wumpus*¹ mittels Lego Mindstorms NXT Robotern, die die Aufgabe durch eigenes logisches Schließen lösen sollen. Dies findet in einem physikalisch vorhandenen Spielfeld statt. In diesem Spiel gibt es zum einen den *Wumpus* und zum anderen den *Spieler*, der aufgrund seiner *Künstlichen Intelligenz* im weiteren Verlauf Agent genannt wird. Der Agent muss das in einer Höhle mit verschiedenen Räumen versteckte Gold finden, ohne dabei vom Wumpus verspeist zu werden. In der Höhle befinden sich Falltüren, in die nur der Agent hineinfallen kann. Da die Karte der Höhle, im Gegensatz zum Wumpus, dem Agenten nicht bekannt ist, muss er sie selbstständig erkunden. Die Falltüren, der Wumpus und das Gold können an angrenzenden Felder durch bestimmte Merkmale wie zum Beispiel Gestank wahrgenommen werden. Der Agent hat zudem die Möglichkeit, den Wumpus mit einem beigegeführten Pfeil zur Strecke zu bringen.

Das Spiel soll auf einem rechteckigen Spielfeld vorgegebener Größe stattfinden. Das Spielfeld kann weitere Wände enthalten, die jedoch keinen abgeschlossenen Raum bilden dürfen. Der Wumpus und das Gold werden zufällig auf dem Spielfeld verteilt, die Falltüren auf die restlichen Felder mit einer Wahrscheinlichkeit von 0,2. Der Agent startet in der unteren linken Ecke mit der Fahrtrichtung nach rechts. Ein Server soll das Spielgeschehen kontrollieren. Er koordiniert den Spielablauf und den Punktestand des Agenten. Zudem übermittelt er an die Aktoren Wahrnehmungen, die durch die Sensorik der NXT-Roboter nicht umgesetzt werden können.

Smell		Breeze	TRAP
Wumpus	Breeze Smell Gold	TRAP	Breeze
Smell		Breeze	
Start	Breeze	TRAP	Breeze

Abbildung 1.1: Beispiel für Aufbau des Spielfelds

¹Gregory Yob, 1972

1.1 Musskriterien

Folgende Kriterien werden umgesetzt:

- /M10/ Der Agent muss mit Hilfe von Sensoren fahren können, damit er das Rätsel lösen kann.
- /M20/ Der Agent muss sich an die Spielregeln von *Hunt the Wumpus* halten.
- /M30/ Der Agent muss das Spiel selbst mit seiner KI² lösen.
- /M40/ Die vollständige Karte (Positionen aller Wände, Fallen, des Wumpus und des Goldschatzes) soll als XML-Datei eingelesen werden.
- /M50/ Das Spielfeld ist rechteckig und kann Wände enthalten, die jedoch keinen abgeschlossenen Raum bilden.
- /M60/ Der Agent startet in der linken unteren Ecke mit der Fahrtrichtung nach rechts.
- /M70/ Das Spiel muss in absehbarer Zeit beendet sein (eine Minute Spielzeit pro Raum).
- /M80/ Agent und Wumpus bewegen sich immer abwechselnd.
- /M90/ Die Kommunikation des Servers mit den Robotern findet über Bluetooth statt.
- /M100/ Zu Beginn des Spiels muss der Server überprüfen, ob der Agent das Spiel überhaupt gewinnen kann.
- /M110/ Das Modul der KI muss leicht austauschbar sein.
- /M120/ Der Agent besitzt nur einen Pfeil, der in Blickrichtung abgefeuert werden kann und potentiell unendlich weit fliegt.
- /M130/ Der Wumpus gibt einen Schrei ab wenn er von einem Pfeil getroffen wurde.
- /M140/ Das System wird darauf vorbereitet, dass der Wumpus sich auch bewegen kann (vgl. /W10/).
- /M150/ Der Agent nimmt die Wände durch seine Sensoren wahr.
- /M160/ Der Agent erhält nicht umsetzbare Reize, wie z.B. den Luftzug von Fallen, durch den Server.
- /M170/ Die Wahrnehmung der Reize (außer /M150/) findet nicht durch Wände und auch nicht diagonal statt.
- /M180/ Der Agent hat das Ziel eine möglichst hohe Punktzahl zu erreichen.

²Künstliche Intelligenz



1.2 Wunschkriterien

Folgende wünschenswerte Kriterien können umgesetzt werden:

- /W10/ Der Wumpus kennt die gesamte Karte und soll sich ebenfalls bewegen können, um dem Agenten das Lösen der Aufgabe schwieriger zu machen.
- /W20/ Das Labyrinth kann mit Hilfe einer GUI³ eingegeben werden.
- /W30/ Die Spielergebnisse, wie der Punktestand oder die Karte, können gespeichert werden.
- /W40/ Die eingelesene Karte wird auf Korrektheit überprüft.

1.3 Abgrenzungskriterien

- /A10/ Der Agent soll den Gestank nicht per Sensorik erkennen.
- /A20/ Die Falltür soll nicht physisch dargestellt werden.
- /A30/ Das Glitzern soll nicht per Sensorik erkannt werden.
- /A40/ Der Luftzug soll nicht per Sensorik erkannt werden.
- /A50/ Bei Spielfeldern, die größer als 6x6 Räume sind, ist es egal, ob das Rätsel in absehbarer Zeit gelöst werden kann.

³Graphical User Interface

2 Produkteinsatz

2.1 Anwendungsbereiche

Das Produkt, die Simulation des Spiels *Hunt the Wumpus*, wird für den Forschungsbereich entwickelt und kann von Forschern mit entsprechenden Fachkenntnissen analysiert und weiterentwickelt werden.

Das Produkt wird im Forschungsbereich eingesetzt. Die Anwendung dient zur Erforschung der künstlicher Intelligenz am Spiel *Hunt the Wumpus*.

2.2 Zielgruppen

Das Produkt ist ausschließlich für KI-interessierte Forscher und Entwickler gedacht. Fortgeschrittene Programmierkenntnisse und Grundkenntnisse im Bereich der künstlichen Intelligenz werden vorausgesetzt.

2.3 Betriebsbedingungen

Der Spielablauf wird durch einen Server koordiniert, der auf einem Java-6-fähigen PC bzw. Laptop betrieben wird. Jede Spielrunde wird von einem Anwender gestartet und ggf. beendet. Der Agent und der Wumpus werden durch einen Lego Mindstorms NXT Roboter umgesetzt, der mit einer leJOS Firmware betrieben wird. Die unmittelbaren Bedingungen müssen bezüglich der Agentenhardware eingehalten werden, da sonst der reibungslose Einsatz der Software nicht gewährleistet werden kann.

Es wird vorausgesetzt, dass das Spiel in einem rechtwinkligen physikalischen Labyrinth abläuft.

3 Produktübersicht

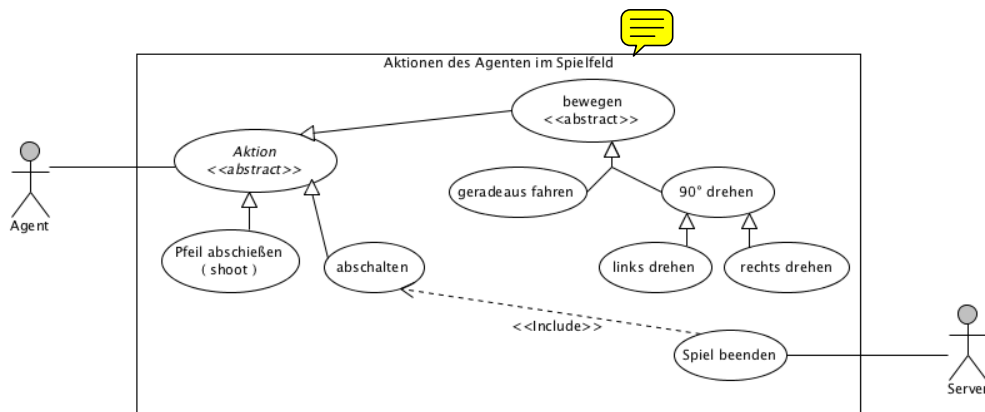


Abbildung 3.1: Use-Cases - Aktionen des Agenten

Der Agent hat mehrere Aktionen zur Auswahl (eine Aktion benötigt einen ganzen Zug, der Agent kann im gleichen Zug also keine zweite Aktion ausführen). Zum einen ist das eine 90 Drehung nach links oder rechts (/F10/) oder das Geradeausfahren (/F20/). Und zum anderen hat er die Möglichkeit einen Pfeil abzuschießen (/F150/). Wird das Spiel vom Server beendet, wird der Agent abgeschaltet (/F130/).

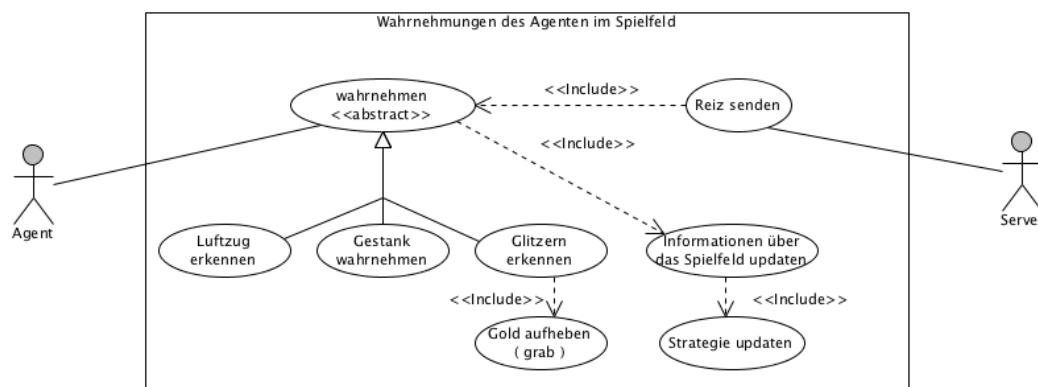


Abbildung 3.2: Uses-Cases - Wahrnehmungen des Agenten

Der Agent kann verschiedene Sachen wahrnehmen: den Gestank des Wumpus (/F190/), den Luftzug einer Falldtür (/F180/) oder das Glitzern des Goldes (/F170/) was das Aufheben des Goldes nach sich zieht (/F160/). Diese Reize bekommt er vom Server übermittelt.

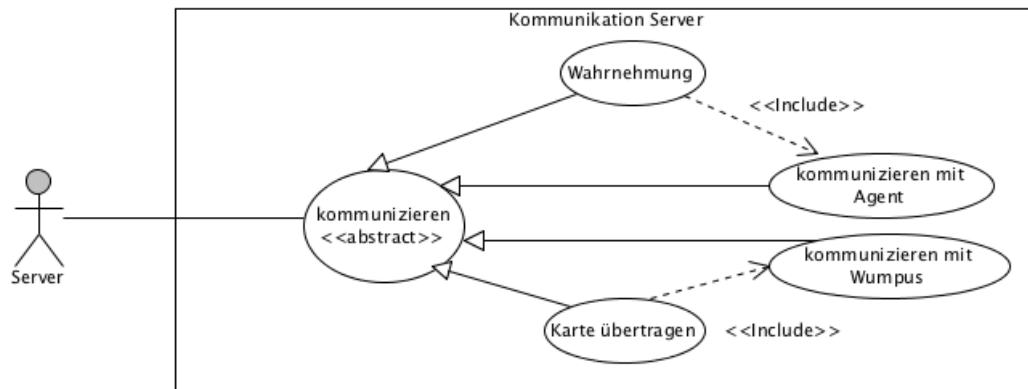


Abbildung 3.3: Use-Cases - Server

Die wichtigste Aufgabe des Servers während des Spieles ist die Kommunikation mit Wumpus und Agent. So uebertraegt er am Anfang die Karte an den Wumpus und teilt dem Agenten seine Wahrnehmungen mit (/F90/).

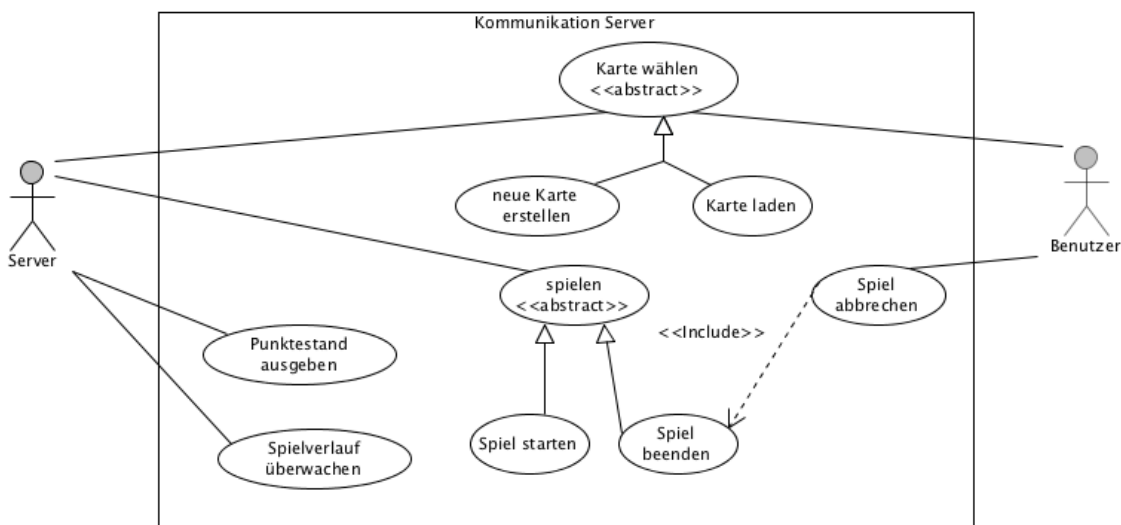


Abbildung 3.4: Use-Cases - Server und Benutzer

Ein möglicher Anwendungsfall des System Interaktion Server und Benutzer ist „Punktestand ausgeben“ durch den der aktuelle Punktestand des Agenten ausgegeben wird (/F120/). Ein weiterer Anwendungsfall ist „Spielverlauf überwachen“ der es ermöglicht die Punkte es Agenten mit zu zählen und das Ende des Spieles fest zu stellen (/F100/). „Spiel abbrechen“ (/F80/) erlaubt es dem Benutzer das Spiel vorzeitig zu beenden oder der Server beendet das Spiel (/F110/) nachdem „Spielverlauf überwachen“ das Ende des Spieles festgestellt hat. Wird das

Spiel beendet, wird automatisch der Punktestand ausgegeben (/F120/). Es ist dem Benutzer möglich eine Karte zu wählen (/F50/), wobei er die Wahl hat eine Karte zu laden (/F40/) oder eine neue Karte zu erstellen (/F30). Dem Server ist es möglich die Spielinformationen sowohl dem Wumpus als auch dem Server zu kommunizieren (/F90/). Nimmt der Agent etwas wahr oder soll dem Wumpus die Spielfeldkarte übertragen werden ist das kommunizieren mit einem der beiden Aktive notwendig.

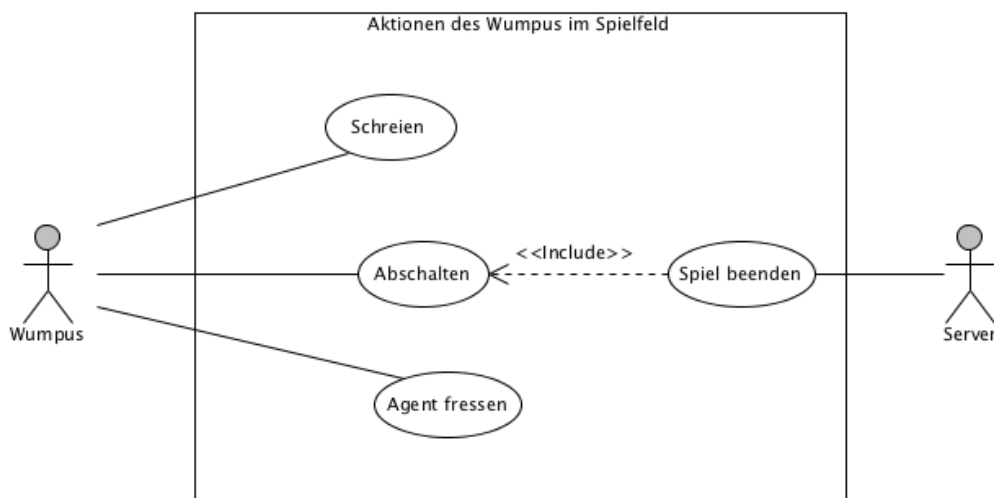


Abbildung 3.5: Use-Cases - Wumpus

Der Wumpus kann den Agenten fressen, sobald sich beide in demselben Raum befinden (F210). Sobald der Wumpus vom Pfeil des Agenten getroffen wird, schreit er so laut, dass man es im ganzen Labyrinth hören kann (F140). Wenn der Wumpus die Meldung erhält, dass das Spiel beendet ist, schaltet er sich ab (F130).

4 Produktfunktionen

In den Produktfunktionen können keine Bezüge zu einem Lastenheft gesetzt werden, da kein Kundeninterview stattfand und es deshalb kein Lastenheft gibt.

/F10/

- Geschäftsprozess: drehen
- Ziel der Funktion: Der Akteur dreht sich um 90 Grad nach links oder rechts.
- Vorbedingung: Der entsprechende Akteur ist am Zug.
- Nachbedingung Erfolg: Der Akteur hat sich um 90 Grad gedreht.
- Nachbedingung Fehlschlag: Der Akteur hat sich zu weit oder nicht weit genug gedreht.
- Akteur: Agent
- Auslösendes Ereignis: Die KI des Akteurs möchte den entsprechenden Roboter drehen.
- Beschreibung: Der Akteur dreht sich um 90 Grad nach links oder rechts.

/F20/

- Geschäftsprozess: geradeaus fahren
- Ziel der Funktion: Der entsprechende Akteur bewegt sich ein Feld vor.
- Vorbedingung: Der entsprechende Akteur ist am Zug.
- Nachbedingung Erfolg: Der entsprechende Akteur ist genau ein Feld vorgerückt.
- Nachbedingung Fehlschlag: Der entsprechende Akteur hat sich nicht weit genug oder zu weit bewegt.
- Akteur: Agent
- Auslösendes Ereignis: Die KI des Akteurs möchte den entsprechenden Roboter nach vorne bewegen.
- Beschreibung: Der entsprechende Roboter rückt ein Feld vor.

/F30/

- Geschäftsprozess: neue Karte erstellen
- Ziel der Funktion: Der Benutzer kann sich eine eigene Karte erstellen.
- Vorbedingung: Der Benutzer hat gerade ein neues Spiel gestartet und wählte 'neue Karte erstellen'.
- Nachbedingung Erfolg: Der Benutzer hat sich eine eigene Karte erstellt.
- Nachbedingung Fehlschlag: Der Benutzer konnte sich keine eigene Karte erstellt.
- Aktor: Benutzer
- Auslösendes Ereignis: Der Benutzer hat ein neues Spiel gestartet und beschlossen eine neue Karte zu erstellen.
- Beschreibung: Die Funktion erlaubt es dem Benutzer eine neue Karte zu erstellen:
 1. neue Karte erstellen wählen
 2. Kartengröße wählen
 3. Wände platzieren
 4. Fallen platzieren
 5. Gold platzieren
 6. bestätigen

/F40/

- Geschäftsprozess: Karte laden
- Ziel der Funktion: Der Benutzer kann eine Karte laden.
- Vorbedingung: Der Benutzer hat gerade ein neues Spiel gestartet und wählte 'Karte laden'.
- Nachbedingung Erfolg: Der Benutzer hat eine Karte geladen.
- Nachbedingung Fehlschlag: Der Benutzer konnte keine Karte laden.
- Aktor: Benutzer
- Auslösendes Ereignis: Der Benutzer hat ein neues Spiel gestartet und beschlossen eine Karte zu laden.
- Beschreibung: Die Funktion erlaubt es dem Benutzer eine Karte zu laden:
 1. Karte laden auswählen
 2. Dateipfad angeben

3. bestätigen

/F50/

- Geschäftsprozess: Karte wählen
- Ziel der Funktion: Der Benutzer entscheidet sich für eine Kartenauswahl.
- Vorbedingung: Der Benutzer hat gerade ein neues Spiel gestartet.
- Nachbedingung Erfolg: Der Benutzer hat sich entschieden.
- Nachbedingung Fehlschlag: Der Benutzer konnte keine Auswahl treffen.
- Aktor: Benutzer
- Auslösendes Ereignis: Der Benutzer hat ein neues Spiel gestartet.
- Beschreibung: Die Funktion erlaubt es dem Spieler zwischen 'neue Karte erstellen' (*/F30/*) und 'Karte laden' (*/F40/*) zu wählen:
 1. zwischen 'Karte laden' und 'neue Karte erstellen' wählen
 2. bestätigen

/F60/

- Geschäftsprozess: Spiel starten
- Ziel der Funktion: Das Spiel wird gestartet und die Verbindungen zu den Robotern aufgebaut.
- Vorbedingung: Der Benutzer startet gerade das Spiel.
- Nachbedingung Erfolg: Das Spiel wurde gestartet und die Verbindungen erfolgreich.
- Nachbedingung Fehlschlag: Das Spiel wurde/konnte nicht gestartet werden.
- Aktor: Benutzer
- Auslösendes Ereignis: entsprechende Eingabe durch den Benutzer
- Beschreibung: Die Funktion startet das Spiel:
 1. das Programm starten
 2. Spieloptionen wählen (siehe */F70/*)
 3. bestätigen

/F70/

- Geschäftsprozess: Spieloptionen wählen
- Ziel der Funktion: Festlegung der Spieleinstellungen
- Vorbedingung: Der Spieler will die Standard-Einstellungen nicht verwenden.
- Nachbedingung Erfolg: Der Spieler konnte die Einstellungen ändern.
- Nachbedingung Fehlschlag: Der Spieler konnte keine Einstellungen ändern.
- Akteure: Benutzer, Server
- Auslösendes Ereignis: Der Spieler hat ein Spiel gestartet.
- Beschreibung: Die Funktion erlaubt es dem Spieler die Spieleinstellungen zu wählen:
 1. bewegbarer Wumpus (on/off)
 2. Karte wählen (siehe /F50/)
 3. bestätigen

Die Standardoptionen sind:

- 'neue Karte erstellen' (/F50/)
- kein bewegbarer Wumpus

/F80/

- Geschäftsprozess: Spiel abbrechen
- Ziel der Funktion: Das Spiel wird beendet.
- Vorbedingung: Es läuft ein Spiel, welches der Spieler abbrechen will.
- Nachbedingung Erfolg: Das Spiel wurde beendet.
- Nachbedingung Fehlschlag: Das Spiel konnte nicht beendet werden.
- Akteure: Agent, Benutzer
- Auslösendes Ereignis: Der Benutzer klickt den entsprechenden Button.
- Beschreibung: Die Funktion bricht das Spiel ab und teilt dies dem Agenten und dem Wumpus mit:
 1. klicken des entsprechenden Buttons
 2. bestätigen

/F90/

- Geschäftsprozess: Spielinformationen kommunizieren
- Ziel der Funktion: Die Funktion lässt das Spiel synchron ablaufen.
- Vorbedingung: Es läuft ein Spiel und die Verbindungen mit Wumpus und Agent sind aufgebaut.
- Nachbedingung Erfolg: Das Spiel läuft synchron.
- Nachbedingung Fehlschlag: Das Spiel läuft nicht synchron.
- Akteure: Benutzer, Agent
- Auslösendes Ereignis: Ein Spiel wird gestartet.
- Beschreibung: Die Funktion regelt den Spielablauf:
 - regelt das abwechselnde Ziehen von Wumpus und Agent
 - informiert den Agenten über die Eigenschaften eines Feldes (Gestank, Luftzug, Glitzern)
 - gibt die aktuellen Positionen von Agent und Wumpus an den Server

/F100/

- Geschäftsprozess: Spielverlauf überwachen
- Ziel der Funktion: Ziel der Funktion ist es, die Punkte des Agenten zu zählen und festzustellen, ob das Spiel zu Ende ist.
- Vorbedingung: Es läuft ein Spiel.
- Nachbedingung Erfolg: Die Punkte wurden korrekt gezählt und das Ende des Spieles richtig festgestellt.
- Nachbedingung Fehlschlag: Die Punkte wurden nicht korrekt gezählt oder das Ende des Spieles zu früh oder gar nicht festgestellt.
- Akteure: Server, Agent
- Auslösendes Ereignis: Beginn des Spieles
- Beschreibung: Die Funktion zählt die Punkte und stellt fest, wann der Agent gewinnt oder verliert. Für folgende Aktionen werden Punkte vergeben:
 - Agent sammelt das Gold: +1000 Punkte
 - Agent wird vom Wumpus gefressen: -1000 Punkte
 - Agent fällt durch eine Falltür: -1000 Punkte

- Agent dreht sich nach links oder rechts: -1 Punkt
- Agent bewegt sich geradeaus: -1 Punkt
- Agent schießt dem Pfeil ab: -10 Punkte
- Pfeil des Agenten trifft: +10 Punkte

Das Spiel endet unter folgenden Bedingungen:

- Der Agent entkommt mit dem Gold aus der Höhle.
- Der Agent ist in eine Falltür gefallen.
- Der Wumpus hat den Agenten gefressen.

/F110/

- Geschäftsprozess: Spiel beenden
- Ziel der Funktion: Das Spiel wird durch den Server beendet und der Punktestand ausgegeben. (/F120/)
- Vorbedingung: Es läuft ein Spiel.
- Nachbedingung Erfolg: Das Spiel wurde beendet.
- Nachbedingung Fehlschlag: Das Spiel wurde nicht korrekt beendet.
- Akteure: Server, Wumpus, Agent
- Auslösendes Ereignis: Die Funktion 'Spielverlauf überwachen' (/F100/) hat das Ende des Spieles festgestellt.
- Beschreibung: Die Funktion beendet das Spiel im Falle eines Sieges oder einer Niederlage des Agenten und teilt dies dem Agenten und dem Wumpus mit.

/F120/

- Geschäftsprozess: Punktestand ausgeben
- Ziel der Funktion: Der Punktestand wird ausgegeben.
- Vorbedingung: Die Funktion 'Spielverlauf überwachen' (/F100/) hat das Ende des Spieles festgestellt.
- Nachbedingung Erfolg: Der Punktestand wurde korrekt ausgegeben.
- Nachbedingung Fehlschlag: Der Punktestand wurde nicht korrekt ausgegeben.
- Akteur: Server

- Auslösendes Ereignis: Die Funktion 'Spielverlauf überwachen' (*/F100/*) hat das Ende des Spieles festgestellt.
- Beschreibung: Die Funktion gibt den Punktestand aus.

/F130/

- Geschäftsprozess: abschalten
- Ziel der Funktion: Der Akteur schaltet sich ab.
- Vorbedingung: Der Akteur ist angeschaltet.
- Nachbedingung Erfolg: Der Akteur wurde abgeschaltet.
- Nachbedingung Fehlschlag: Der Akteur wurde nicht abgeschaltet.
- Akteure: Wumpus, Agent
- Auslösendes Ereignis: Der Server teilt das Ende des Spiels/den Tod des Wumpus mit.
- Beschreibung: Die Funktion schaltet den entsprechenden Roboter ab.

/F140/


- Geschäftsprozess: schreien
- Ziel der Funktion: Der Wumpus gibt einen Ton (=Schrei) aus.
- Vorbedingung: Der Wumpus ist eingeschaltet und wurde durch den Agenten erschossen.
- Nachbedingung Erfolg: Ein Ton wurde ausgegeben.
- Nachbedingung Fehlschlag: Es wurde kein Ton ausgegeben.
- Akteur: Wumpus
- Auslösendes Ereignis: Der Server teilt dem Wumpus mit, dass er erschossen wurde.
- Beschreibung: Der Wumpus bekommt vom Server die Mitteilung, dass er getötet wurde und gibt anschließend einen Ton als Schrei aus.

/F150/

- Geschäftsprozess: Pfeil abschießen
- Ziel der Funktion: Der Agent schießt seinen Pfeil ab.
- Vorbedingung: Der Agent ist am Zug und hat seinen Pfeil noch nicht abgeschossen.
- Nachbedingung Erfolg: Der Pfeil wird in Blickrichtung des Agenten abgeschossen.

- Nachbedingung Fehlschlag: Der Pfeil befindet sich noch beim Agenten oder wurde nicht in Blickrichtung des Agenten abgeschossen.
- Akteur: Agent
- Auslösendes Ereignis: Die Agenten-KI möchte den Wumpus mit seinem Pfeil töten.
- Beschreibung: Der Agent schießt seinen Pfeil ab.

/F160/

- Geschäftsprozess: Gold aufheben
- Ziel der Funktion: Der Agent hebt das Gold auf.
- Vorbedingung: Der Agent ist am Zug.
- Vorbedingung: Der Agent steht auf dem Feld, auf dem sich das Gold befindet.
- Nachbedingung Erfolg: Der Agent ist nun im Besitz des Goldes.
- Nachbedingung Fehlschlag: Der Agent ist nun nicht im Besitz des Goldes.
- Akteur: Agent
- Auslösendes Ereignis: Der Agent nimmt ein Glitzern wahr. 
- Beschreibung: Der Agent hebt das Gold auf.

/F170/

- Geschäftsprozess: Glitzern erkennen
- Ziel der Funktion: Der Agent bekommt vom Server die Information, dass es auf seinem Feld glitzert.
- Vorbedingung: Das Spiel wurde gestartet.
- Nachbedingung Erfolg: Der Agent nahm ein Glitzern wahr und entdeckte die Position des Goldes.
- Nachbedingung Fehlschlag: Der Agent hat die Nachricht des Servers ignoriert, nicht verarbeitet, fehlinterpretiert oder nicht erhalten.
- Akteur: Agent
- Auslösendes Ereignis: Der Agent entdeckt ein neues Feld.
- Beschreibung: Der Agent nimmt ein Glitzern wahr und erfährt die Position des Goldes.

/F180/

- Geschäftsprozess: Luftzug erkennen
- Ziel der Funktion: Der Agent bekommt vom Server die Information, dass es auf seinem Feld einen Luftzug gibt.
- Vorbedingung: Das Spiel wurde gestartet.
- Nachbedingung Erfolg: Der Agent nahm einen Luftzug wahr und aktualisierte ggf. seine Spielstrategie.
- Nachbedingung Fehlschlag: Der Agent hat die Nachricht des Servers ignoriert, nicht verarbeitet, fehlinterpretiert oder nicht erhalten.
- Akteur: Agent
- Auslösendes Ereignis: Der Agent entdeckt ein neues Feld.
- Beschreibung: Der Agent nimmt einen Luftzug wahr und updatet ggf. seine Spielstrategie.

/F190/

- Geschäftsprozess: Gestank wahrnehmen
- Ziel der Funktion: Der Agent bekommt vom Server die Information, dass es auf seinem Feld einen Gestank gibt.
- Vorbedingung: Das Spiel wurde gestartet.
- Nachbedingung Erfolg: Der Agent nahm einen Gestank wahr und aktualisierte ggf. seine Spielstrategie.
- Nachbedingung Fehlschlag: Der Agent hat die Nachricht des Servers ignoriert, nicht verarbeitet, fehlinterpretiert oder nicht erhalten.
- Akteur: Agent
- Auslösendes Ereignis: Der Agent entdeckt ein neues Feld.
- Beschreibung: Der Agent nimmt einen Gestank wahr und updatet ggf. seine Spielstrategie.

/F200/

- Geschäftsprozess: korrigieren
- Ziel der Funktion: Der Agent korrigiert seine Position.
- Vorbedingung: Der Agent steht nicht mittig im Raum.
- Nachbedingung Erfolg: Der Agent konnte seine Position korrigieren.

- Nachbedingung Fehlschlag: Der Agent konnte sich nicht in eine mittige Position bewegen.
- Akteur: Agent
- Auslösendes Ereignis: Der Agent misst seine Position mit der Sensorik.
- Beschreibung: Wenn der Agent beim Messen der Position feststellt, dass er nicht mittig im Raum steht, muss er seine Position korrigieren.

/F210/

- Geschäftsprozess: fressen
- Ziel der Funktion: Der Wumpus frisst den Agenten.
- Vorbedingung: Wumpus und Agent befinden sich in demselben Raum.
- Nachbedingung Erfolg: Der Wumpus hat den Agenten gefressen.
- Nachbedingung Fehlschlag: Der Wumpus konnte den Agenten nicht fressen.
- Akteur: Wumpus
- Auslösendes Ereignis: Der Agent wechselt in den Raum des Wumpus oder umgekehrt.
- Beschreibung: Sobald sich Wumpus und Agent in demselben Raum befinden, frisst der Wumpus den Agenten.

5 Produktdaten

/D10/ Karte (siehe /M40/):


- Größe der Karte
- Positionen von Fallen (falls das Spiel exakt gleich sein soll und die Fallen nicht zufällig verteilt werden sollen)
- Positionen von Wänden
- Startposition des Wumpus (falls das Spiel exakt gleich sein soll und die Fallen nicht zufällig verteilt werden sollen)
- Einlesen der Karte findet über eine XML-Datei statt

6 Nichtfunktionale Anforderungen

Produktqualität	sehr gut	gut	normal	nicht relevant
Funktionalität				
Angemessenheit		x		
Richtigkeit		x		
Interoperabilität	x			
Ordnungsmäßigkeit				x
Sicherheit			x	
Zuverlässigkeit				
Reife	x			
Fehlertoleranz		x		
Wiederherstellbarkeit				x
Benutzbarkeit				
Verständlichkeit			x	
Erlernbarkeit			x	
Bedienbarkeit			x	
Effizienz				
Zeitverhalten		x		
Verbrauchsverhalten		x		
Änderbarkeit				
Analysierbarkeit		x		
Modifizierbarkeit	x			
Stabilität		x		
Prüfbarkeit			x	
Übertragbarkeit				
Anpassbarkeit			x	
Installierbarkeit				x
Konformität	x			
Austauschbarkeit				x



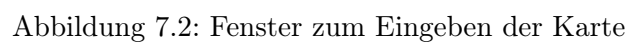
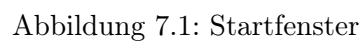
Qualitätsanforderungen

- /Q10/ Es ist eine teamübergreifend festgelegte Schnittstelle zwischen Server und Robotern nötig. Erst dadurch wird der Wettkampf mit dem anderen Team möglich. 
- /Q20/ Die Drehnung des Roboters muss möglichst genau sein.
- /Q30/ Die Übertragung von Daten via Bluetooth muss effizient sein.
- /Q40/ Die Programme der Roboter müssen effizient sein, da sie über wenig Ressourcen verfügen.

7 Benutzeroberfläche

Die Anforderungen an die Benutzeroberfläche beziehen sich auf das Wunschkriterium */W20/*.

- */B10/* Die Benutzeroberfläche muss mit der Bibliothek javax.swing erstellt werden und kann somit unabhängig vom Betriebssystem benutzt werden.
- */B20/* Mausbedienung wird vorausgesetzt.
- */B30/* Die Karte muss über die GUI eingegeben werden oder aus Dateien geladen werden.
- */B40/* Es gibt ein spezielles Fenster zum eingeben der Karte und ein Fenster zum laden der Karte.
- */B50/* Bei Eingabe der Karte steht eine Oberfläche zu Verfügung zum hinzufügen der Feldeigenschaften (z.B. Goldposition).
- */B60/* Es besteht eine Möglichkeit eine Größe der Karte einzugeben.
- */B70/* Es steht jeweils ein Button um das Spiel zu starten und ein um das Spiel zu beenden zur Verfügung.
- */B80/* Die Punkteverfolgung kann durch die bereitgestellte Oberfläche erfolgen.
- */B90/* Es steht eine Oberfläche zu Verfügung auf der die aktuelle Aktion verfolgt werden kann.
- */B100/* Es steht eine Oberfläche zur Verfügung, auf der das Spiel sowie Wumpus- und Agentenposition verfolgt werden kann.



8 Technische Produktumgebung

8.1 Software

- Server-Betriebssystem: jedes Betriebssystem mit Grafikausgabe, auf dem die 32-bit-Version von Java 6 lauffähig ist
- Agent-Betriebssystem: leJOS 0.8.5
- Wumpus-Betriebssystem: leJOS 0.8.5

8.2 Hardware

- Server: PC oder Laptop mit Grafikbildschirm und kompatiblen Bluetoothmodul
- Agent: Roboter mit LEGO NXT brick
- Wumpus: Roboter mit LEGO NXT brick

8.3 Orgware



- Das Spielfeld wird durch Holzplatten begrenzt.
 - Die Holzplatten müssen höher sein als die Sensoren der Roboter.
 - Die Holzplatten müssen die gleiche Größe haben.
 - Die Holzplatten müssen sich im rechten Winkel anordnen lassen.
 - Die Holzplatten müssen eine ebene Oberfläche haben.
 - Die Holzplatten müssen höher sein als die Sensoren der Roboter.
 - Die Holzplatten können auch einzelne Räume begrenzen.

8.4 Produktschnittstellen

Da es einen Wettkampf mit Team 1 geben soll, werden folgende Schnittstellen benötigt:

- /S10/ Kommunikation von Server mit Agent (bzw. Wumpus) über Bluetooth in einem teamübergreifend einheitlichem Format
- /S20/ Einlesen von Karten in einem teamübergreifend einheitlichem XML-Format

9 Glossar

Agent - Sucht das versteckte Gold in der Höhle, ohne dabei vom Wumpus verspeist zu werden.

Karte - virtuelle Repräsentation der Wumpus-Welt auf Rechnerseite

LEGO NXT brick - Grundbaustein von Robotern der Firma LEGO.

leJOS - Java Virtual Machine für den LEGO NXT brick.

Raum - Der Raum ist ein Teil des Spielfeldes, in dem die Roboter ihre Aktionen ausführen.
Wenn ein Roboter sich geradeaus bewegt, wechselt er den Raum.

Spieler - siehe **Agent**

Spielfeld - mit Holzplatten begrenztes Labyrinth, in dem sich die Roboter bewegen

Wumpus - Monster, das in der Höhle lebt.