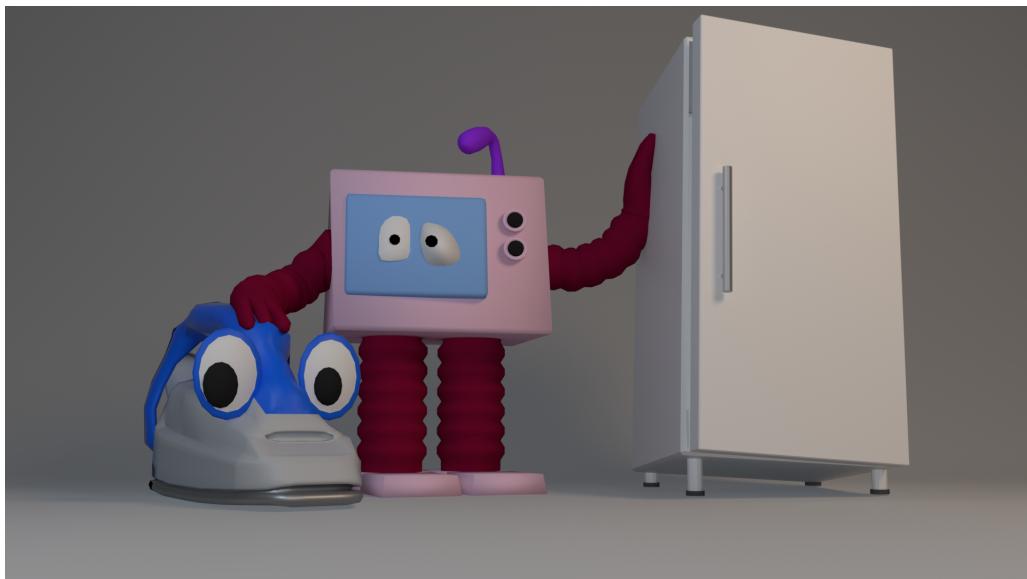


UNIVERSITÄT ULM

Robo Rally

Pflichtenheft von 0x7DA8



erstellt von

Marie Lohbeck
Misam Huynh

Pascal Schiessle
Pascal Weber

Ulrike Kulzer
Nathan Maier

Version vom: 1. Februar 2017

Inhaltsverzeichnis

1	Überblick	1
1.1	Einleitung	1
1.2	Motivation	1
1.3	Vision	1
1.4	Projektkontext	2
1.5	Domänenmodell	3
2	Anforderungsanalyse	5
2.1	Akteure	5
2.2	Funktionale Anforderungen	7
2.2.1	Spielregeln	7
2.2.2	Organisation	17
2.2.3	Darstellung	21
3	Benutzerschnittstellen	27
3.1	Dialog-Eigenschaften	27
3.2	Dialogstrukturdiagramme	29
3.3	Benachrichtigungen	33
3.3.1	Vor dem Spiel	33
3.3.1.1	Startmenü	33
3.3.1.2	ServerSuchen	33
3.3.1.3	Registrieren	34
3.3.1.4	ServerHinzufuegen	35
3.3.1.5	ServerLobby	35
3.3.1.6	Einstellungen	36
3.3.1.7	ProgrammBeenden	37
3.3.1.8	Hilfebereich	37
3.3.2	Im Spiel (Beobachter)	42
3.3.2.1	Beobachter	42
3.3.2.2	Pausebereich	42
3.3.2.3	Spielende	43
3.3.2.4	EinstellungenBeobachter	43
3.3.2.5	HilfebereichBeobachter	43
3.3.3	Im Spiel (Spieler)	45
3.3.3.1	Spieldoberfläche	45
3.3.3.2	PausebereichSpieler	46
3.3.3.3	GameOver	46
3.3.3.4	Karten	47
3.3.3.5	Disqualifikation	47
3.3.3.6	SpielVerlassen	47
3.3.4	Editor	49
3.3.4.1	Leveleditor	49

Inhaltsverzeichnis

3.3.4.2	NichtGespeichert	50
3.3.4.3	LevelSpeichern	50
3.3.4.4	LevelLaden	51
3.3.5	Server	52
3.3.5.1	Server	52
3.3.5.2	ServerNeuesSpielErstellen	53
3.3.5.3	ServerLevelLaden	53
3.3.5.4	ServerEinstellungen	53
3.3.5.5	ServerLaufendesSpiel	54
3.3.5.6	ServerPause	54
3.3.5.7	ServerSpielGestartet	54
3.3.5.8	ServerNeuesSpiel	56
3.4	Fehlermeldungen	57
3.4.1	Vor dem Spiel	57
3.4.1.1	VerbindungsaufbauFehlgeschlagen	57
3.4.2	Editor	58
3.4.2.1	LevelLoadError	58
3.4.2.2	CheckpointError	58
3.4.3	Weitere Fehlermeldungen	59
3.4.3.1	KeineSpielerMehr	59
3.4.3.2	VerbindungVerloren	59
3.5	Nutzungskonzept	60
4	Datenmodell	61
4.1	Sequenzdiagramme	61
4.1.1	Anmeldung Initiieren	61
4.1.2	Beobachter	62
4.1.3	KI	63
4.1.4	Spiel Beenden	64
4.1.5	Spieler	65
4.1.6	Disqualifikation	66
4.1.7	Spiel spielen	66
4.1.8	Level erstellen	67
4.1.9	Anforderungsabhängigkeiten	68
4.2	Zustandsdiagramme	69
4.2.1	Beobachter	69
4.2.2	KI	70
4.2.3	Mensch	71
4.2.4	Roboter	72
4.2.5	Server	73
4.2.6	Spielsitzung	73
4.2.7	Anforderungsabhängigkeiten	74
4.2.8	Bemerkungen	74

Inhaltsverzeichnis

5 Randbedingungen	75
5.1 Qualitätsanforderungen	75
5.2 Abnahmekriterien	78
Anhang	79
Glossar	79

1 Überblick

1.1 Einleitung

Wir werden im Projekt eine digitale Version des Brettspiels „Robo Rally“ entwickeln. Im Originalspiel geht es darum, die Spielfiguren mittels überlegter Vorausplanung über das Spielfeld zu führen und dabei Checkpoints abzulaufen. Das Computerspiel soll eine Nachbildung sein, die das Brettspiel an die digitale Umgebung anpasst.

Mehrere Spieler können sich mit ihren Computern verbinden und gegeneinander oder gegen eine künstliche Intelligenz antreten. Weiterhin besteht die Möglichkeit, dass weitere Personen dem Spielgeschehen auf ihrem Computer zuschauen.

1.2 Motivation

Das Brettspiel „Robo Rally“ ist spannend und auch auf lange Zeit interessant, da man mit strategischem Vorgehen sehr viel bewirken und sich verbessern kann. Es scheint also naheliegend, den Fans auch eine digitale Version anzubieten – gerade weil die Domäne Roboter ohnehin die Schnittstelle zwischen digitaler und analoger Welt behandelt. Dadurch ist die Wahrscheinlichkeit hoch, dass die Zielgruppe des Spiels auch für ein dazugehöriges Computerspiel zu begeistern ist, was sich wiederum positiv auf die Vermarktung auswirkt. Zusätzlich wird der Spielerkreis um Liebhaber von Computerspielen erweitert.

Außerdem bietet die Digitalisierung von Spielen generell einige Vorteile. So hat man keine Kleinteile mehr wie Kärtchen, Figuren und Chips, die leicht verloren gehen oder versehentlich von ihrem angewiesenen Platz geschoben werden. Spielfeldauf- und -abbau funktionieren auf Knopfdruck und die Einhaltung der Regeln wird automatisch überprüft, was Schummeln erschwert und schlichte Fehler durch die Spieler unwahrscheinlicher macht. Weiterhin stehen der virtuellen Welt Methoden zur Verfügung, um das Szenario realistisch zu gestalten, was bei einem Brettspiel schwer umsetzbar ist. Das Spielgeschehen kann zum Beispiel animiert und mit Soundeffekten kommentiert werden. Zudem ist es einfacher, eine Vielzahl unterschiedlicher Level zur Verfügung zu stellen, damit der Spielspaß auch auf lange Sicht gewährleistet wird. Ein weiterer Pluspunkt der Digitalversion ist, dass sich Spieler über das Internet verbinden können. Das ermöglicht ihnen, das Spiel sogar dann miteinander zu spielen, wenn sich nicht alle Spieler am selben Ort aufhalten.

1.3 Vision

Am Ende der beiden Semester soll ein Computerspiel stehen, das die Spieler herausfordert, Strategien zu entwickeln und Spielzüge zu optimieren, und dabei auch in hohem Maße die möglichen Pläne der Gegenspieler miteinzubeziehen. Die Aufgabe

1 Überblick

der Spieler ist es, einen Weg zu finden, um möglichst schnell alle Checkpoints in der richtigen Reihenfolge zu besuchen. Der schnellste Roboter gewinnt.

Unser Ziel ist es, ein ansprechendes Spiel zu gestalten, das kurzweilig ist und auch auf lange Sicht noch Herausforderung und Spielspaß bietet. Der besondere Antrieb liegt im Rahmen dieses Projekts aber darin, andere Gruppen in der Implementierung von „Robo Rally“ zu überbieten, um das beste Produkt abzuliefern und auf der anstehenden Messe viele Kunden zu gewinnen.

1.4 Projektkontext

Das Spiel wird im Rahmen des Softwaregrundprojekts (kurz SOPRA) entwickelt. Das SOPRA muss in den Informatikstudiengängen an der Uni Ulm absolviert werden – laut Regelstudienplan in der Mitte des Bachelorstudiums. In erster Linie geht es darum, die Studierenden zum ersten Mal in einem größeren Projekt arbeiten zu lassen. Einerseits können hierbei die in den vorangegangenen Semestern erlernten Programmierkenntnisse in einem größeren Rahmen angewandt werden, andererseits kann ein erstes Gefühl für Organisationsumfang und Dokumentationsaufwand gewonnen werden, was bei Einzelaufgaben nicht wirklich benötigt wird. Auch Teamarbeit wird trainiert. Das Spiel soll während des Wintersemesters 2016/17 und des Sommersemesters 2017 in Gruppen von jeweils sechs Personen entwickelt werden.

Am Ende des Projekts wird es ein Turnier geben, um die gebauten Spielimplementierungen zu testen und zu vergleichen. Darüber hinaus könnte es als echte Alternative zur Brettspielvariante angesehen und mit dem richtigen Support gegebenenfalls sogar vermarktet werden.

1.5 Domänenmodell

Folgendes Diagramm soll einen ersten Überblick darüber geben, wie das Spiel in den Grundzügen aufgebaut sein soll. Es ist dargestellt, wie die Domänen Spielbrett, Mauer, Checkpoint, Roboter und Karte zueinander in Verbindung stehen. Außerdem gibt es noch Drehungen und Vorwärtsbewegungen, die für die Werte der Karten verwendet werden. Drehungen werden in diesem Modell zusätzlich dazu genutzt, um die Richtung von Robotern und Mauern zu definieren. Um Checkpoints, Mauern und Roboter räumlich auf dem Spielfeld zu verorten, existiert eine weitere Domäne Vektor.

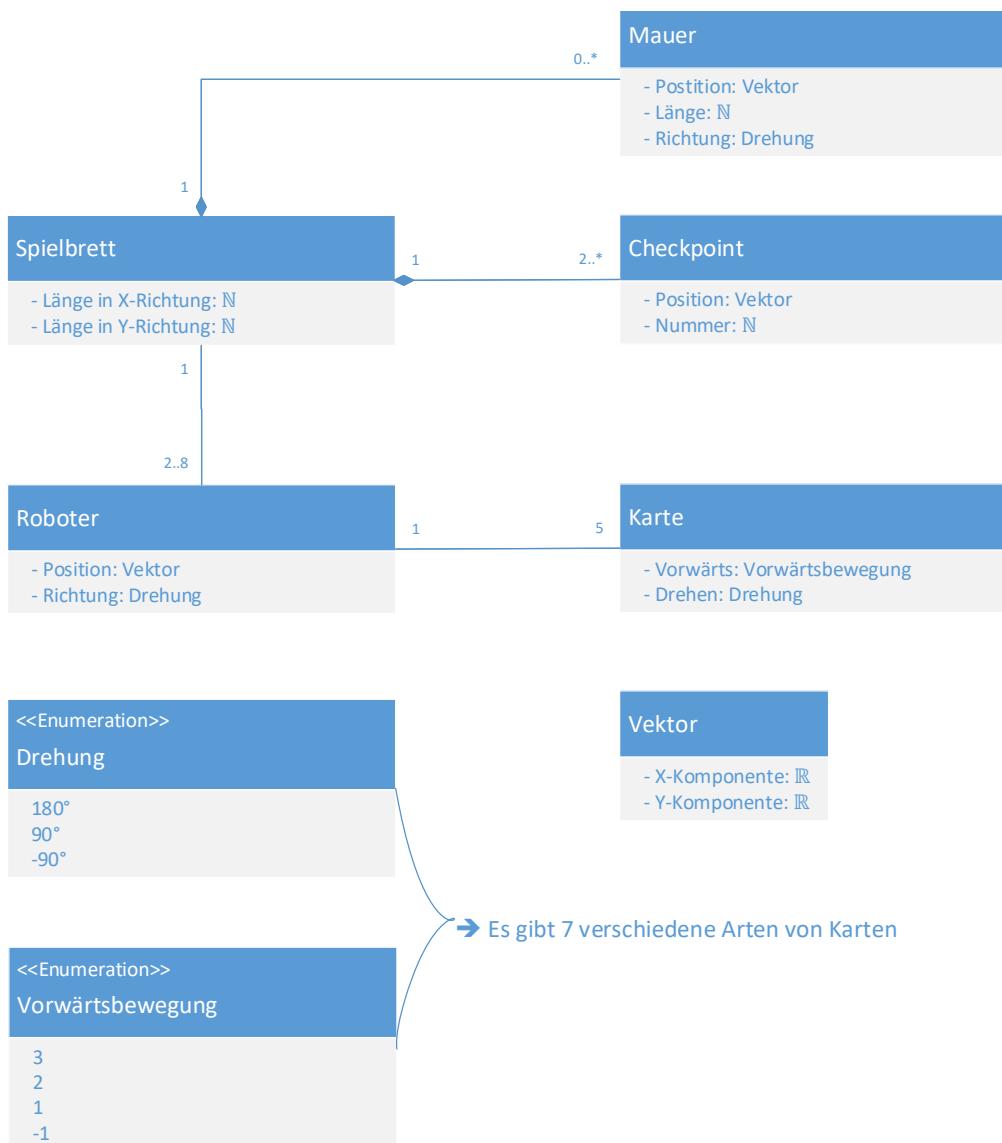


Abbildung 1.1: Domänenmodell-Grafik

1 Überblick

Invarianten:

- Der Koordinatenursprung liegt im linken unteren Eck des Spielfelds
- Die X-Komponenten aller Vektoren dürfen nicht größer als die Länge in X-Richtung des Spielfelds sein
- Die Y-Komponenten aller Vektoren dürfen nicht größer als die Länge in Y-Richtung des Spielfelds sein
- Die Nummern der Checkpoints werden in aufsteigender Reihenfolge vergeben und sind eindeutig.
- Die Mauern müssen alle komplett innerhalb des Spielfelds liegen
- Es existieren 7 verschiedene Karten, von denen der Spieler jede Runde 9 zufällige Karten bekommt, maximal 3 von einer Sorte und mindestens eine Fortbewegungskarte. Der Spieler muss sich 5 davon aussuchen.
- Es gibt die Drehungsmöglichkeiten -90° , 90° und 180°
- Es gibt die Fortbewegungskarten -1, 1, 2, 3

2 Anforderungsanalyse

2.1 Akteure

Im Folgenden werden alle Akteure, die in der Software auftauchen, genauer beschrieben. Allen funktionalen Anforderungen ist jeweils mindestens ein Akteur zugeordnet, der auf das Erfüllen der Anforderungen angewiesen ist.

NAME	Spieler: KI-Teilnehmer
ROLLE	Ist ein virtueller Spieler.
BESCHR.	Der KI-Teilnehmer kann durch die Kommandozeile gestartet werden und steuert während dem Spiel einen Roboter anhand eines geeigneten Algorithmus.

NAME	Spieler: Mensch-Teilnehmer
ROLLE	Ist eine realer Spieler.
BESCHR.	Ein Mensch-Teilnehmer steuert einen Roboter durch Befehle, die er anhand einer passenden Strategie auswählt.

NAME	Beobachter
ROLLE	Passiver Spieler
BESCHR.	Der Beobachter ist ein passiver Spieler, der durch eine visualisierte Übersicht und über die aktuell eingestellten Konfigurationen und den Spielstatus das Spiel mitverfolgt.

NAME	Server
ROLLE	Schnittstelle zwischen Teilnehmer und Beobachter
BESCHR.	Der Server kann Spiele erstellen und verwalten. Er bestimmt die Levelkonfigurationen. Teilnehmer und Beobachter können sich an einem Server registrieren. Der Server kann Registrierungen akzeptieren oder ablehnen und Spieler und Beobachter dann jeweils einem Spiel zuordnen. Während eines Spiels ist er für die Einhaltung der Spielregeln verantwortlich. Er kann Spiele pausieren und abbrechen.

2 Anforderungsanalyse

NAME	Client
ROLLE	Aktive und passive Teilnehmer
BESCHR.	Client ist eine Oberkategorie von allen Teilnehmern. Darunter fallen die menschlichen Spieler, die KI und auch die Beobachter. Der Client muss sich beim Server registrieren, um beim Spiel teilnehmen zu können.

2.2 Funktionale Anforderungen

2.2.1 Spielregeln

ID	FA-S-01
TITEL	Roboter - Bewegung
BESCHR.	Ein Roboter kann sich drehen oder eine bestimmte Anzahl von Feldern ziehen. Die Bewegungen der Roboter werden von den Programmen bestimmt, die die Spieler zusammenstellen.
BEGRÜ.	Um dem Teilnehmer das Steuern des Roboters zu ermöglichen.
ABHÄNG.	FA-S-02, FA-S-10, FA-S-11, FA-S-08
AKTEUR	Spieler: Mensch, Spieler: KI, Server
PRIORI.	++

ID	FA-S-02
TITEL	Programmierung
BESCHR.	Um einen Roboter zu programmieren wählt ein Spieler fünf seiner Karten aus und bringt sie in eine Reihenfolge.
BEGRÜ.	Um den Benutzern Kontrolle über die Roboter zu geben.
ABHÄNG.	FA-S-01, FA-D-02, FA-S-18, FA-S-03
AKTEUR	Spieler: Mensch und Spieler: KI
PRIORI.	++

2 Anforderungsanalyse

ID	FA-S-03
TITEL	verfügbare Zeit Programmierung
BESCHR.	Den Spielern steht eine begrenzte Zeit zum Auswählen seiner Karten zur Verfügung, die vor dem Spiel vom Server festgelegt wird und mindestens 15 Sekunden betragen muss. Hält ein Spieler diese Zeit nicht ein, so wird sein Roboter zerstört.
BEGRÜ.	Damit es zu keinen Spielverzögerungen kommt.
ABHÄNG.	FA-S-18, FA-S-02
AKTEUR	Spieler: Mensch, Spieler: KI, Server
PRIORI.	+

ID	FA-S-04
TITEL	Leben
BESCHR.	Jeder Roboter besitzt drei Leben. Er verliert ein Leben, wenn er zerstört wird. Besitzt ein Spieler keine Leben mehr, wird er disqualifiziert.
BEGRÜ.	Der Roboter kann auch disqualifiziert werden, wenn er zu oft zerstört wurde.
ABHÄNG.	FA-S-05
AKTEUR	Server
PRIORI.	+

ID	FA-S-05
TITEL	Disqualifikation
BESCHR.	Ein Spieler wird disqualifiziert, falls sein Roboter zum dritten Mal zerstört wird oder er eine ungültige Aktion durchführt.
BEGRÜ.	Damit sich die Spieler an die Spielregeln halten.
ABHÄNG.	FA-S-02, FA-D-06
AKTEUR	Server
PRIORI.	++

2.2 Funktionale Anforderungen

ID	FA-S-06
TITEL	Spielbrett
BESCHR.	Das Spielbrett soll rechteckig sein und aus quadratischen Feldern derselben Größe bestehen, auf denen sich die Roboter bewegen. Zwischen den einzelnen Feldern können sich Mauern befinden. Auf einigen Feldern stehen Checkpoints.
BEGRÜ.	Die Roboter benötigen eine Spielumgebung, in der sie sich bewegen können.
ABHÄNG.	FA-S-07, FA-S-08, FA-S-09
AKTEUR	Server
PRIORI.	++

ID	FA-S-07
TITEL	Checkpoint
BESCHR.	Auf dem Spielbrett befinden sich mehrere Checkpoints, mindestens aber zwei, die auf genau einem Feld stehen. Sie werden eindeutig nummeriert. Alle Roboter starten vom Checkpoint mit der Nummer 1. Die Aufgabe der Spieler ist es, mit ihren Robotern die Checkpoints in der richtigen Reihenfolge bis zum letzten Checkpoint abzulaufen.
BEGRÜ.	Damit die Spieler eine Aufgabe haben.
ABHÄNG.	FA-S-06, FA-S-09
AKTEUR	Server
PRIORI.	++

2 Anforderungsanalyse

ID	FA-S-08
TITEL	Mauer
BESCHR.	Auf dem Spielfeld befinden sich auf den Grenzen zwischen manchen Feldern Mauern, die die Roboter daran hindern, jene Grenzen zu überqueren.
BEGRÜ.	Um die Herausforderung zu erhöhen.
ABHÄNG.	FA-S-06, FA-S-09
AKTEUR	Server
PRIORI.	++

ID	FA-S-09
TITEL	Kacheln
BESCHR.	Die Kacheln auf dem Spielbrett sind quadratisch und haben eine feste Länge und Breite von einer Rastereinheit.
BEGRÜ.	Die feste Größe soll den Kacheln eine einheitliche Form geben.
ABHÄNG.	FA-S-06, FA-S-07
AKTEUR	Server
PRIORI.	o

ID	FA-S-10
TITEL	Roboter schiebt Roboter
BESCHR.	Auf jedem Feld darf höchstens ein Roboter stehen. Falls ein Roboter über/auf ein Feld ziehen soll, auf dem sich bereits ein anderer Roboter befindet, wird dieser um die entsprechende Anzahl an Feldern geschoben, falls keine Blockade auftritt.
BEGRÜ.	Um sicherzustellen, dass nie mehr als ein Roboter auf jedem Spielfeld steht und um das Spiel unberechenbarer für die Spieler zu machen.
ABHÄNG.	FA-S-01, FA-S-06, FA-S-07, FA-S-08
AKTEUR	Server
PRIORI.	++

ID	FA-S-11
TITEL	Blockade
BESCHR.	Falls ein Roboter bei einer Bewegung auf eine Mauer stößt, wird seine Bewegung blockiert und er muss davor stehen bleiben. Dies gilt sowohl, wenn der Roboter gerade selbst einen Laufbefehl ausführt, als auch, wenn er von einem anderen Roboter geschoben wird. Falls ein Roboter blockiert wird, während ein anderer Roboter ihn schiebt, wird der schiebende Roboter ebenfalls blockiert.
BEGRÜ.	Um festzulegen, was passiert, wenn ein Roboter auf eine Mauer stößt.
ABHÄNG.	FA-S-01, FA-S-06, FA-S-08, FA-S-02
AKTEUR	Server
PRIORI.	++

ID	FA-S-12
TITEL	Roboter fällt vom Spielbrett
BESCHR.	Fällt ein Roboter vom Spielbrett, so wird er zerstört.
BEGRÜ.	Um festzulegen, was passiert, wenn ein Roboter über den Spielfeldrand hinaus bewegt werden sollte; um das Spiel herausfordernder zu gestalten.
ABHÄNG.	FA-S-01, FA-S-06, FA-S-13, FA-D-06
AKTEUR	Server
PRIORI.	++

2 Anforderungsanalyse

ID	FA-S-13
TITEL	Zerstörung
BESCHR.	Falls ein Roboter zerstört wird, wird er zu Beginn der nächsten Runde bei seinem ersten Zug wieder am zuletzt erreichten Checkpoint eingesetzt.
BEGRÜ.	Teil der Spielregeln.
ABHÄNG.	FA-D-06, FA-S-04
AKTEUR	Server
PRIORI.	++

ID	FA-S-14
TITEL	Kartenblatt Zusammensetzung
BESCHR.	Ein Spieler darf von einer Karte nicht mehr als 3 Stück pro Runde erhalten. Ebenso darf er nicht nur Drehbefehle bekommen.
BEGRÜ.	Um das Spiel fair zu gestalten wird die Kartenvarianz verringert.
ABHÄNG.	FA-S-02
AKTEUR	Server
PRIORI.	++

ID	FA-S-15
TITEL	Zerstörung bei Einsetzung
BESCHR.	Wenn ein Roboter bei Neueinsetzung auf dem von ihm zuletzt erreichten Checkpoint erscheint, so zerstört er gegebenenfalls einen anderen Roboter der sich bereits auf dieser Kachel befindet.
BEGRÜ.	Um sicherzustellen, dass nicht zwei Roboter zur selben Zeit auf einer Kachel stehen.
ABHÄNG.	-
AKTEUR	Server
PRIORI.	++

2.2 Funktionale Anforderungen

ID	FA-S-16
TITEL	Karten
BESCHR.	<p>Es gibt sieben verschiedene Karten, die aber jeweils mehrfach vorkommen. Auf jeder Karte steht ein Befehl für Roboter. Folgende Befehle gibt es:</p> <ul style="list-style-type: none"> • 1 Feld vorwärts • 2 Felder vorwärts • 3 Felder vorwärts • 1 Feld rückwärts • Vierteldrehung gegen den Uhrzeigersinn • Vierteldrehung mit dem Uhrzeigersinn • Halbe Drehung (180°)
BEGRÜ.	Teil der Spielregeln.
ABHÄNG.	-
AKTEUR	Server
PRIORI.	++

2 Anforderungsanalyse

ID	FA-S-17
TITEL	Karten - Priorität
BESCHR.	<p>Jede Karte hat eine individuelle Priorität. Allerdings ist die Rangfolge unter verschiedenen Befehlen immer fest:</p> <ul style="list-style-type: none"> • Halbe Drehung (180°) (1) • Vierteldrehung mit dem Uhrzeigersinn (2) • Vierteldrehung gegen den Uhrzeigersinn (3) • 1 Feld rückwärts (4) • 1 Feld vorwärts (5) • 2 Felder vorwärts (6) • 3 Felder vorwärts (7) <p>Die Zahlen in Klammern geben die Priorität der Befehle untereinander an. Bei der Programmausführung werden höher priorisierte Befehle zuerst ausgeführt.</p>
BEGRÜ.	Teil der Spielregeln.
ABHÄNG.	FA-S-14, FA-S-02
AKTEUR	Server
PRIORI.	++

ID	FA-S-18
TITEL	Kartenverteilung
BESCHR.	Jeder Spieler erhält zu Beginn jeder Spielrunde neun Karten mit Befehlen für die Programmierung des Roboters.
BEGRÜ.	Teil der Spielregeln.
ABHÄNG.	FA-S-02, FA-S-03
AKTEUR	Server
PRIORI.	+

2.2 Funktionale Anforderungen

ID	FA-S-19
TITEL	Spielziel
BESCHR.	Der Spieler, der das Ziel, also den Checkpoint mit der höchsten Nummer, zuerst erreicht, hat das Spiel gewonnen. Der Spieler muss vorher alle anderen Checkpoints in der richtigen Reihenfolge besucht haben.
BEGRÜ.	Das Ziel des Spiels wird eindeutig festgelegt.
ABHÄNG.	FA-S-02
AKTEUR	Server
PRIORI.	++

ID	FA-S-20
TITEL	Nichtsichtbarkeit der Karten anderer Spieler
BESCHR.	Jeder Spieler sieht nur seine eigenen Karten. Auch die Beobachter sehen die Karten der Spieler nicht.
BEGRÜ.	Teil der Spielregeln.
ABHÄNG.	FA-S-02
AKTEUR	Server
PRIORI.	++

2 Anforderungsanalyse

ID	FA-S-21
TITEL	Erster Zug nach Neueinsetzung
BESCHR.	In der ersten Runde und bei jeder Neueinsetzung auf einem Checkpoint muss der Spieler zusätzlich zu seinem Roboterprogramm die gewünschte Blickrichtung für den Roboter angeben. Der erste Befehl des Programms muss dann auch immer ein Laufbefehl sein. In der ersten Runde starten alle Roboter auf Checkpoint 1.
BEGRÜ.	Damit die Roboter nicht auf dem Checkpoint stehen bleiben und die Ausrichtung eindeutig bestimmt ist.
ABHÄNG.	FA-S-02
AKTEUR	Spieler: Mensch, Spieler: KI, Server
PRIORI.	++

2.2.2 Organisation

ID	FA-O-01
TITEL	Spiel verlassen
BESCHR.	Es gibt eine Möglichkeit für Spieler und Beobachter, das Spiel vorzeitig über eine entsprechende Schaltfläche zu verlassen.
BEGRÜ.	Um das Spiel abbrechen zu können.
ABHÄNG.	-
AKTEUR	Client
PRIORI.	+

ID	FA-O-02
TITEL	Begrenzung der Zeit für visuelle Darstellung
BESCHR.	Die Beobachter- und Menschspielerkomponente bekommen nach jeder Runde pro Spielzug fünf Sekunden Zeit, um das Spielgeschehen zu visualisieren. Ist die Darstellung nach Zeitallauf noch nicht beendet, muss sie sofort abgebrochen werden.
BEGRÜ.	Damit der Spielfluss nicht gestört wird.
ABHÄNG.	FA-O-02
AKTEUR	Server, Beobachter, Spieler: Mensch
PRIORI.	++

2 Anforderungsanalyse

ID	FA-O-03
TITEL	Registrierung am Server
BESCHR.	Spieler und Beobachter müssen sich an einem Server registrieren, um an einem Spiel teilnehmen zu können. Der Server darf Registrierungen freigeben oder ablehnen. Der Server trägt Spieler und Beobachter nach seinen Wünschen für Spiele ein.
BEGRÜ.	Damit man am Spiel teilnehmen kann.
ABHÄNG.	FA-O-02
AKTEUR	Client, Server
PRIORI.	++

ID	FA-O-04
TITEL	Start via Kommandozeile
BESCHR.	Der KI-Teilnehmer muss über die Kommandozeile gestartet werden können.
BEGRÜ.	Damit man den KI-Teilnehmer starten kann.
ABHÄNG.	-
AKTEUR	Spieler: KI
PRIORI.	++

ID	FA-O-05
TITEL	Konfiguration der KI
BESCHR.	Nachdem die KI gestartet wurde, erwartet sie über die Kommandozeile IP-Adresse und Port des Servers, auf dem sie spielen soll. Weiterhin erwartet die KI einen Spielernamen, unter dem sie erscheinen soll.
BEGRÜ.	Damit die KI weiß, auf welchem Server sie spielen soll.
ABHÄNG.	FA-O-03
AKTEUR	Spieler: KI
PRIORI.	++

2.2 Funktionale Anforderungen

ID	FA-O-06
TITEL	Berechnung Spielzüge KI
BESCHR.	Die KI verfügt über einen Algorithmus, nach dem sie den nächsten Spielzug berechnet und an den Server weitergibt. Für sie gelten die selben Regeln wie für einen Mensch-Teilnehmer.
BEGRÜ.	Damit die KI den Roboter steuern kann.
ABHÄNG.	FA-O-03
AKTEUR	Spieler: KI
PRIORI.	++

ID	FA-O-07
TITEL	Einbettung der KI in den Client für den menschlichen Spieler
BESCHR.	Die KI kann auch über den menschlichen Spielerclient gestartet werden, um statt des Spielers zu spielen. Der Spieler kann nicht ins Spielgeschehen eingreifen, solange die KI aktiv ist. Möchte der Spieler wieder selbst spielen, kann er die KI auch wieder deaktivieren, allerdings erst nach Ablauf der aktuellen Runde.
BEGRÜ.	Damit der Spieler auch die KI spielen lassen kann.
ABHÄNG.	-
AKTEUR	Spieler: Mensch, Spieler: KI
PRIORI.	-

ID	FA-O-08
TITEL	Einlesen und Bearbeitung von bereits erstellten Leveln
BESCHR.	Der Leveleditor muss bereits erstellte Level einlesen und bearbeiten können.
BEGRÜ.	Damit man bereits erstellte Level bearbeiten kann.
ABHÄNG.	FA-D-08
AKTEUR	Server
PRIORI.	++

2 Anforderungsanalyse

ID	FA-O-09
TITEL	Pause
BESCHR.	Der Server kann das Spiel pausieren.
BEGRÜ.	Vom Kunden gefordert.
ABHÄNG.	FA-D-01, FA-O-01
AKTEUR	Server
PRIORI.	+

ID	FA-O-10
TITEL	Überprüfung der Spielregeln
BESCHR.	Der Server ist für die Einhaltung der Spielregeln verantwortlich. Ist eine Aktion eines Spielers nicht regelkonform, so wird dieser disqualifiziert.
BEGRÜ.	Teil der Spielregeln.
ABHÄNG.	FA-D-06
AKTEUR	Server
PRIORI.	+

2.2.3 Darstellung

ID	FA-D-01
TITEL	Musik
BESCHR.	Während des Spiels läuft eine Hintergrundmusik. Spieler und Beobachter können die Lautstärke einstellen.
BEGRÜ.	Die Hintergrundmusik soll die Beobachter und Spieler zusätzlich unterhalten.
ABHÄNG.	FA-O-09
AKTEUR	Beobachter, Spieler: Mensch
PRIORI.	--

ID	FA-D-02
TITEL	Countdown
BESCHR.	Während der Programmierphase wird Beobachtern und Spielern ein Countdown angezeigt, der die Zeit bis zum Ende der Phase herunterzählt.
BEGRÜ.	Der Spieler weiß, wie viel Zeit ihm noch bleibt, um sich zu entscheiden. Dadurch wird vermieden, dass der Spieler sich verschätzt und ggf. seine Aktionen nicht vollständig durchführen kann. Dies gibt dem Spiel einen Spannungsfaktor.
ABHÄNG.	FA-S-01, FA-S-02
AKTEUR	Beobachter, Spieler: Mensch
PRIORI.	o

2 Anforderungsanalyse

ID	FA-D-03
TITEL	Benachrichtigungsfeld
BESCHR.	Während des Spiels werden Nachrichten über Spielereignisse in einem dafür vorgesehenen Feld eingetragen. Solche Ereignisse sind zum Beispiel Zerstörungen, Disqualifikationen und das Erreichen von Checkpoints von allen Spielern.
BEGRÜ.	Damit Spielteilnehmer keine wichtigen Ereignisse verpassen.
ABHÄNG.	FA-S-06, FA-O-09
AKTEUR	Beobachter, Spieler: Mensch
PRIORI.	o

ID	FA-D-04
TITEL	Spieloberfläche
BESCHR.	Während des Spiels wird das Spielbrett mit alle darauf befindlichen Robotern, Mauern und Checkpoints angezeigt, eine Liste der teilnehmenden Spieler mit der Anzahl der Checkpoints, die sie bisher erreicht haben, und das Benachrichtigungsfeld für das aktuelle Spiel. Außerdem wird der Countdown für Programmierphasen angezeigt.
BEGRÜ.	Der Benutzer hat so einen Überblick über das Spielgeschehen.
ABHÄNG.	FA-S-06, FA-O-09, FA-D-02, FA-D-03
AKTEUR	Beobachter, Spieler: Mensch
PRIORI.	o

2.2 Funktionale Anforderungen

ID	FA-D-05
TITEL	Kartenbereich
BESCHR.	Spielern wird auf der Spieloberfläche zusätzlich ein Kartenbereich angezeigt. Dort kann er die Karten einsehen, die ihm ausgeteilt wurden, und zu einem Programm zusammenstellen, indem er sie per drag-and-drop auf dafür vorgefertigte Felder zieht.
BEGRÜ.	Spieler haben so die Möglichkeit zu programmieren.
ABHÄNG.	FA-D-04, FA-S-16
AKTEUR	Spieler: Mensch
PRIORI.	o

ID	FA-D-06
TITEL	Game-Over-Dialog
BESCHR.	Wenn das Spiel vorbei ist, wird den Teilnehmern ein Game-Over-Dialog angezeigt. Der Gewinner erhält die Information, dass er gewonnen hat. Alle anderen Spieler und die Beobachter erhalten die Information, welcher Spieler gewonnen hat.
BEGRÜ.	Das Spiel wird abgeschlossen.
ABHÄNG.	-
AKTEUR	Beobachter, Spieler: Mensch
PRIORI.	-

2 Anforderungsanalyse

ID	FA-D-07
TITEL	GUI für KI
BESCHR.	Man kann die KI auch über eine GUI starten und bedienen. Hier können dann Serveradresse, Port und Name und Intelligenz der KI eingetragen werden.
BEGRÜ.	Damit man die KI angenehmer bedienen kann.
ABHÄNG.	FA-O-05
AKTEUR	Spieler: KI
PRIORI.	--

ID	FA-D-08
TITEL	Leveleditor
BESCHR.	Es gibt einen Leveleditor, in welchem man über eine geeignete GUI verschiedene Level erstellen kann. Die Level müssen im Jason-Format abgespeichert werden können, damit der Server erstellte Level einlesen und verwenden kann.
BEGRÜ.	Damit man neue Level erstellen kann.
ABHÄNG.	-
AKTEUR	-
PRIORI.	++

ID	FA-D-09
TITEL	Abbruch-Dialog
BESCHR.	Möchte ein Beobachter oder ein Spieler das Spiel abbrechen, erscheint ein Dialog, in dem er das Abbrechen bestätigen muss. Spieler werden darauf hingewiesen, dass ein Abbruch als Niederlage gewertet wird.
BEGRÜ.	Damit Spieler und Beobachter nicht versehentlich abbrechen.
ABHÄNG.	FA-O-01
AKTEUR	Beobachter, Spieler: Mensch
PRIORI.	-

2.2 Funktionale Anforderungen

ID	FA-D-10
TITEL	Logfiles
BESCHR.	Der Server soll alle getätigten Spieleraktionen protokollieren und in einen Log speichern. Man soll das Spiel mithilfe dieses Logs nochmals wiedergeben können.
BEGRÜ.	Damit man sich bereits gespielte Partien nochmals anschauen kann.
ABHÄNG.	-
AKTEUR	Server
PRIORI.	--

3 Benutzerschnittstellen

3.1 Dialog-Eigenschaften

Es gibt Dialoge, welche die grafische Schnittstelle zwischen Beobachter, Spieler, Serverhost und Levelersteller bilden und den Informationsaustausch zwischen dem Mensch und dem Programm gewährleisten. Zu diesen Dialogen gehören:

- Startmenü-Dialog (siehe 3.3.1.1)
- Server-Suchen-Dialog (siehe 3.3.1.2)
- Server-Lobby-Dialog (siehe 3.3.1.5)
- Hilfebereich-Dialog (siehe 3.3.1.8)
- Einstellungen-Dialog (siehe 3.3.1.6)
- Beobachter-Dialog (siehe 3.3.2.1)
- Pausebereich-Dialog (siehe 3.3.2.2)
- Einstellungen-Beobachter-Dialog (siehe 3.3.2.4)
- Hilfebereich-Beobachter-Dialog (siehe 3.3.2.5)
- Spieloberfläche-Dialog (siehe 3.3.3.1)
- Pausebereich-Spieler-Dialog (siehe 3.3.3.2)
- Leveleditor-Dialog (siehe 3.3.4.1)
- Server-Dialog (siehe 3.3.5.1)
- Server–Laufendes-Spiel-Dialog (siehe 3.3.5.5)
- Server-Pause-Dialog (siehe 3.3.5.6)
- Server-Spiel-Gestartet-Dialog (siehe 3.3.5.7)
- Server-Neues-Spiel-Dialog (siehe 3.3.5.8)

Alle nun folgende dargestellten Dialoge sind modal, außer es wird explizit genannt, dass dieser Dialog nicht modal ist. Die Dialoge sind weiterhin noch in bestimmte Kategorien eingeordnet, wie man im Folgendem sehen kann.

Es gibt es Dialoge, welche den Benutzer darauf hinweist, dass eine bestimmte Aktion gerade ausgeführt wird:

- Registrieren-Dialog (siehe 3.3.1.3)

3 Benutzerschnittstellen

Oder es wird eine wichtige Information vom Benutzer abgefragt:

- Server-Hinzufügen-Dialog (siehe 3.3.1.4)
- Programm-Beenden-Dialog (siehe 3.3.1.7)
- Spiel-Verlassen-Dialog (siehe 3.3.1.7)
- Nicht-Gespeichert-Dialog (siehe 3.3.4.2)
- LevelSpeichern-Dialog (siehe 3.3.4.3)
- LevelLaden-Dialog (siehe 3.3.4.4)
- Server-Neues-Spiel-Erstellen-Dialog (siehe 3.3.5.2)
- Server-LevelLaden-Dialog (siehe 3.3.5.3)
- Server-Einstellungen-Dialog (siehe 3.3.5.4)
- VerbindungsauftbauFehlgeschlagen-Dialog (siehe 3.4.1.1)

Es gibt auch wichtige Meldungen, welche dem Benutzer sofort mitgeteilt werden und dieser Bestätigen muss:

- Spielende-Dialog (siehe 3.3.2.3)
- GameOver-Dialog (siehe 3.3.3.3)
- Disqualifikation-Dialog (siehe 3.3.3.5)
- LevelLoadError-Dialog (siehe 3.4.2.1)
- CheckpointError-Dialog (siehe 3.4.2.2)
- KeineSpielerMehr-Dialog (siehe 3.4.3.1)
- VerbindungVerloren-Dialog (siehe 3.4.3.2)

3.2 Dialogstrukturdiagramme

Die folgenden Diagramme sollen einen grafischen Überblick darüber geben, wie die einzelnen Dialoge aufeinander verweisen und ineinander übergehen. Es gibt eine Dialogstruktur für Spieler und Beobachter, eine für den Leveleditor und eine dritte für den Spielserver. Zur Aufschlüsselung dient die Legende (siehe: 3.1).

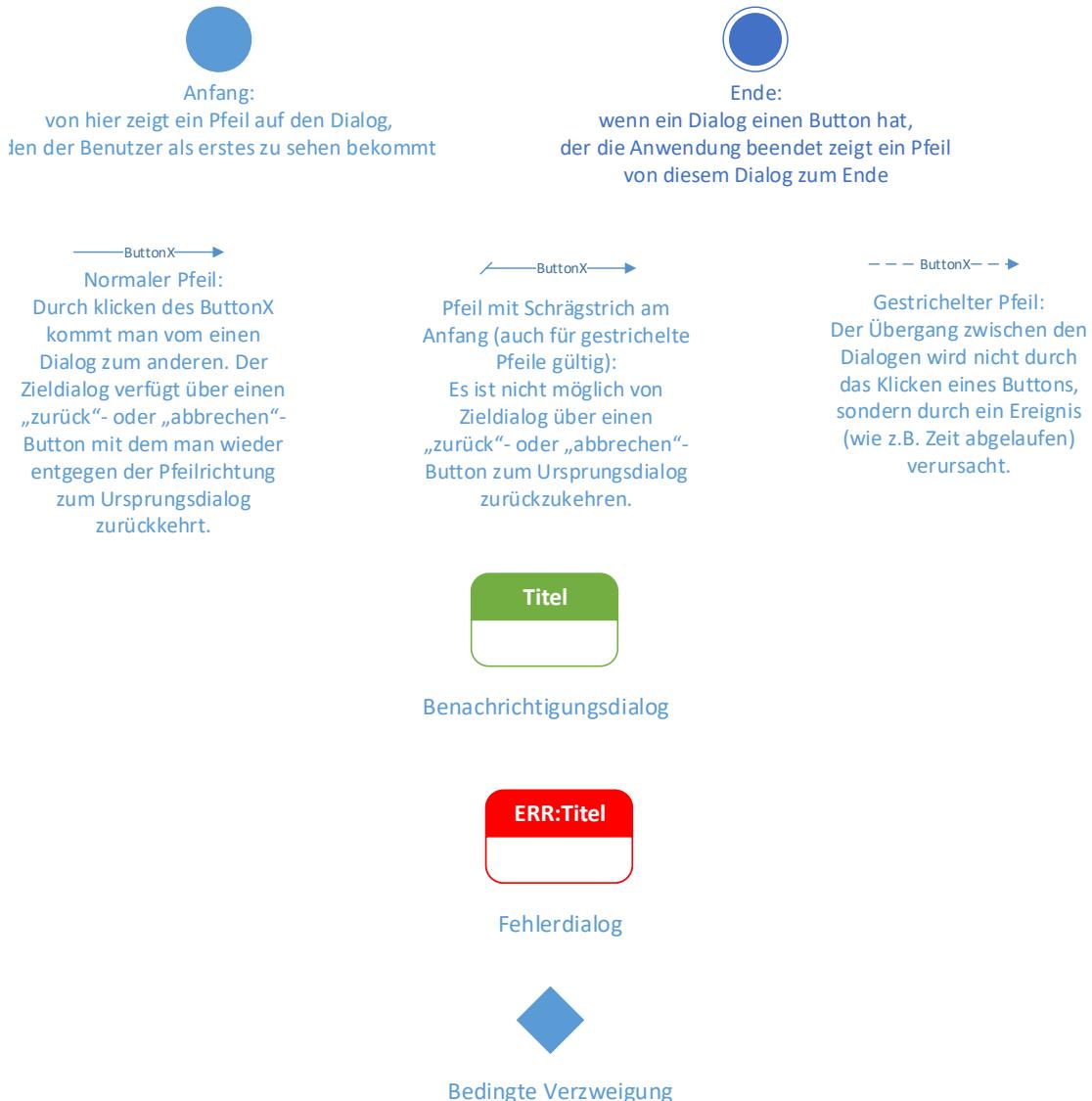


Abbildung 3.1: Dialogstrukturdiagramm-Legende

3 Benutzerschnittstellen

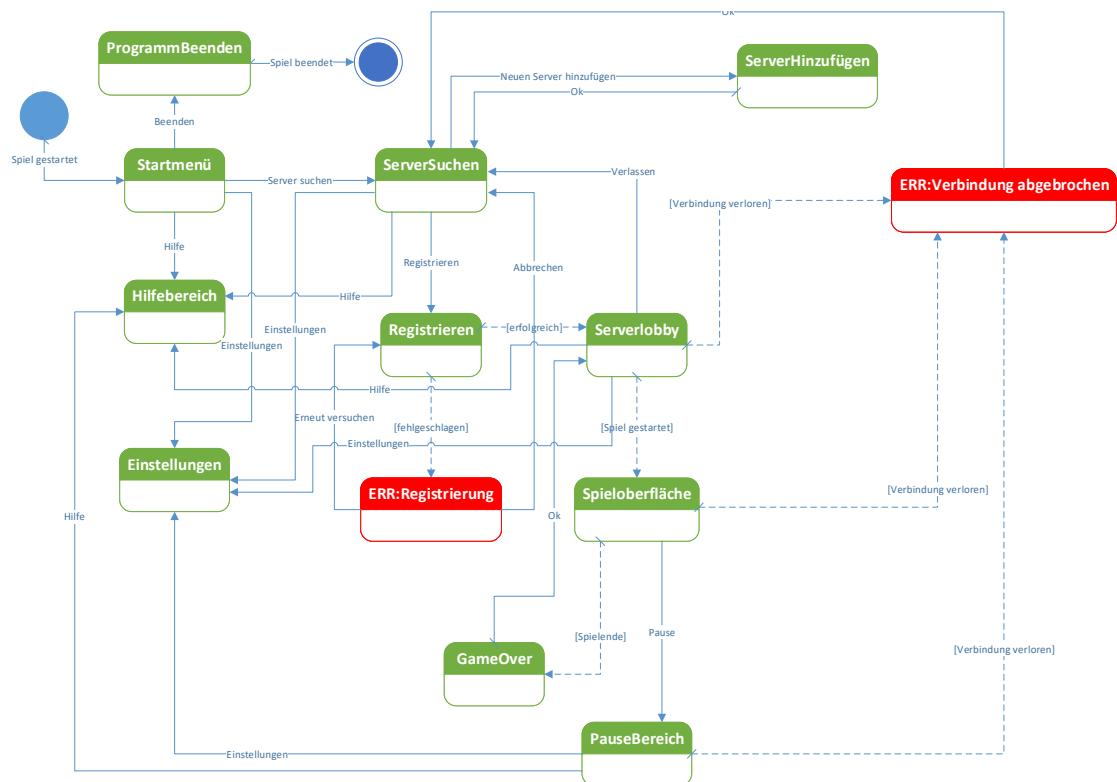


Abbildung 3.2: Dialogstrukturdiagramm-Spieler und Beobachter

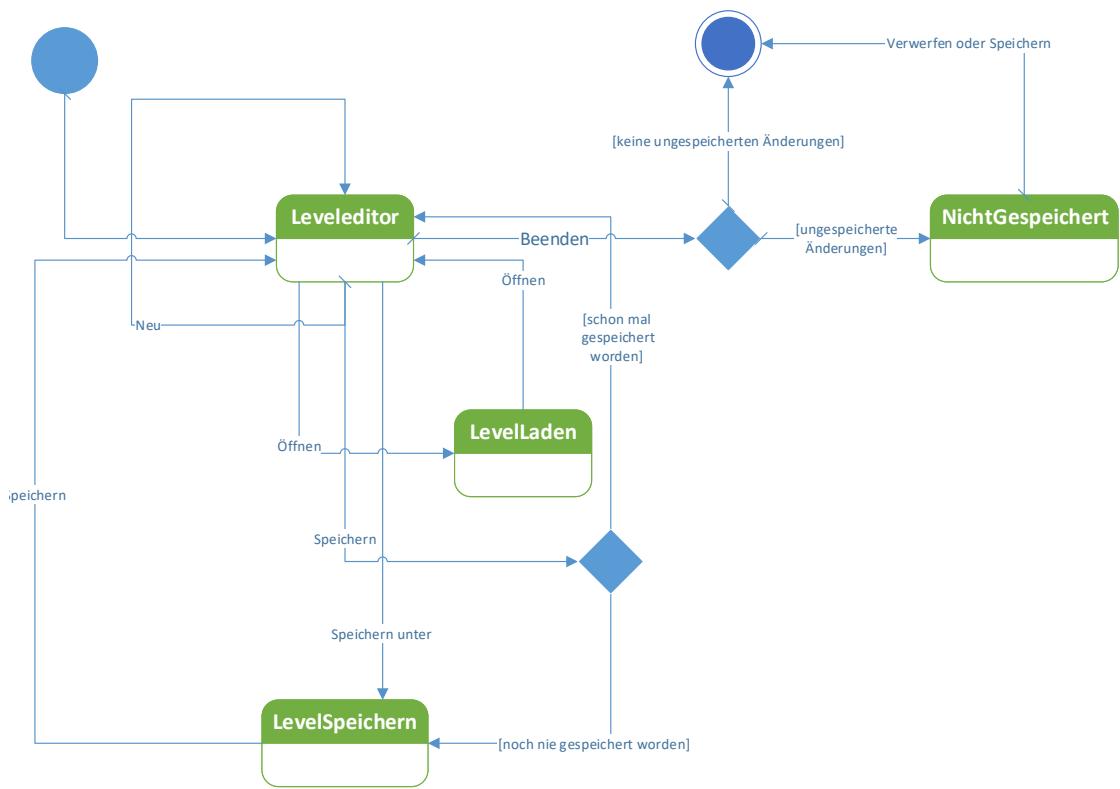


Abbildung 3.3: Dialogstrukturdiagramm-Leveleditor

3 Benutzerschnittstellen

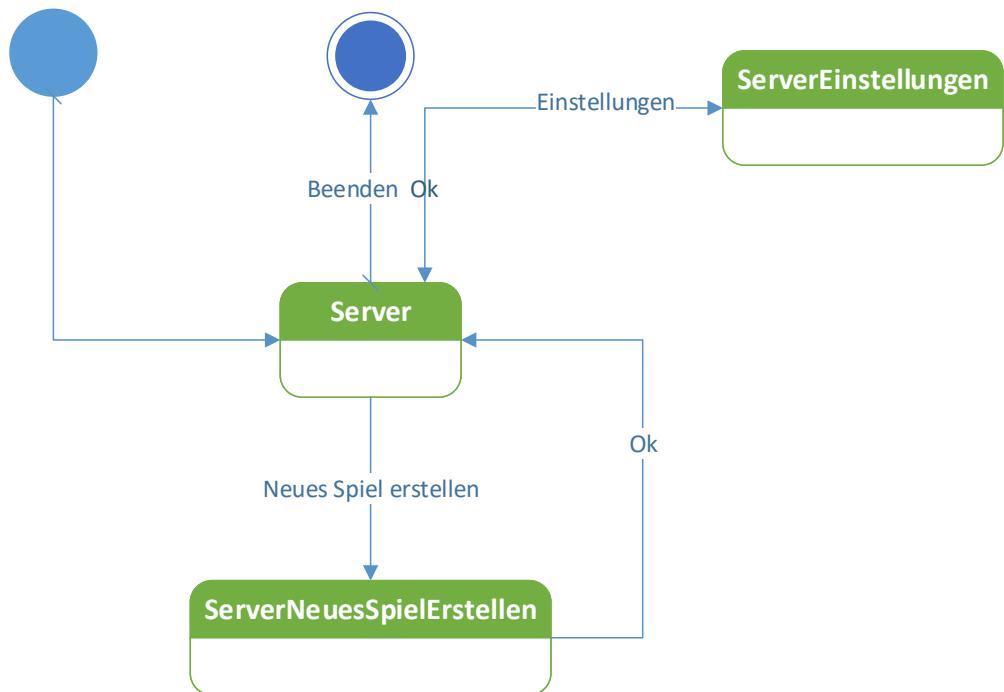


Abbildung 3.4: Dialogstrukturdigramm-Server

3.3 Benachrichtigungen

3.3.1 Vor dem Spiel

3.3.1.1 Startmenü

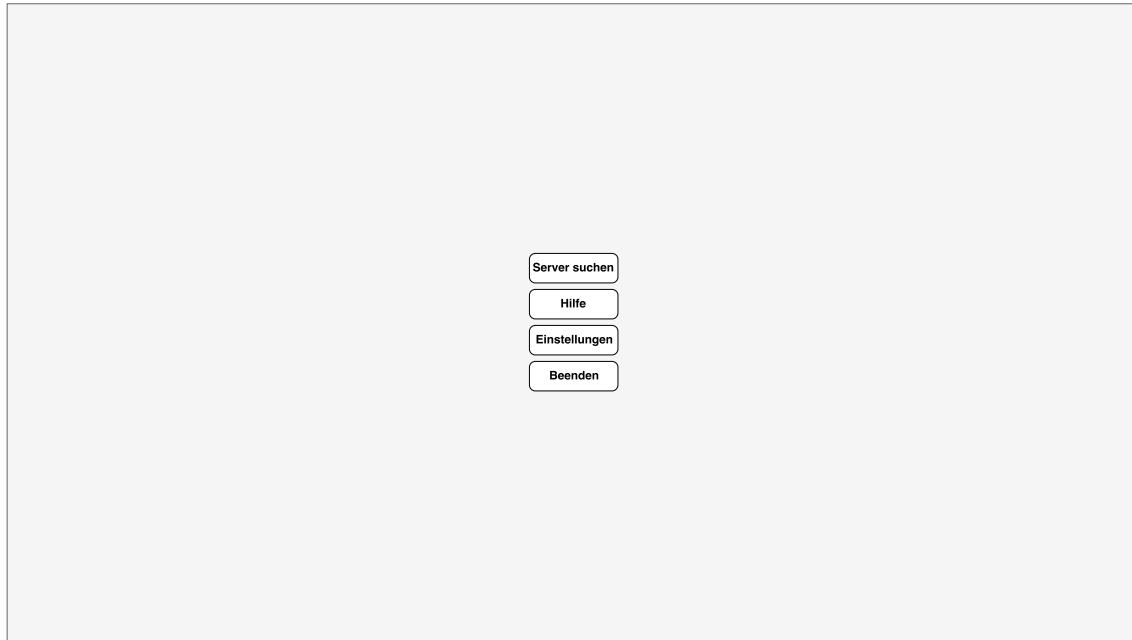


Abbildung 3.5: Startmenü-Dialog

Das Startmenü erscheint nachdem der Nutzer das Programm gestartet hat. Der Nutzer hat nun vier Buttons zur Wahl: „Server suchen“, „Hilfe“, „Einstellungen“ und „Beenden“.

Über „Server suchen“ kommt der Spieler in den *Server-Suchen-Dialog* (siehe 3.3.1.2), und bekommt Server angezeigt, auf denen er spielen kann. Der „Hilfe“-Button öffnet den *Hilfebereich-Dialog* (siehe 3.3.1.8), wo der Nutzer Informationen über die Bedienung der Software und die Spielregeln erhält. Über den „Einstellungen“-Button kommt der Spieler in den *Einstellungen-Dialog* (siehe 3.3.1.6), wo er die Einstellungen für das Spiel ändern kann. Mithilfe des „Beenden“-Buttons lässt sich das Programm nach einer Bestätigung im *Programm-beenden-Dialog* (siehe 3.3.1.7) ordnungsgemäß schließen.

3.3.1.2 ServerSuchen

Dieser Dialog erscheint, wenn der Spieler im *Startmenü-Dialog* (siehe 3.3.1.1) auf den Button „Server suchen“ geklickt hat. Hier kann man aus einer Liste von Servern seinen gewünschten Server auswählen. Durch Anklicken eines Servers erhält man Informationen über IP-Adresse, Portnummer und Status. Unter den Informationen ist ein Textfeld, in das der Nutzer den Namen eintragen kann, unter dem er spielen

3 Benutzerschnittstellen

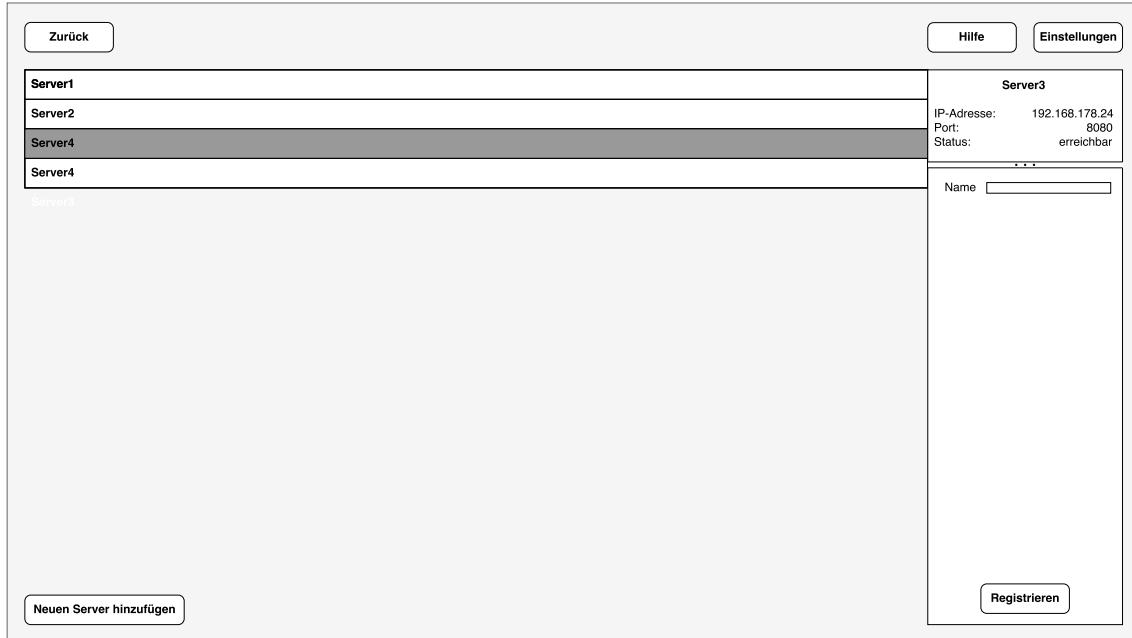


Abbildung 3.6: Server-Suchen-Dialog

möchte, und darunter dann ein Button zur Registrierung. Wenn der Nutzer noch keinen Namen eingegeben hat ist der Button ausgegraut. Nach Klick auf „Registrieren“ erscheint der *Registrieren-Dialog* (siehe 3.3.1.3).

Falls der gesuchte Server nicht in der Liste steht, hat der Spieler die Möglichkeit einen weiteren Server hinzuzufügen, in dem er auf den Button *Server-Hinzufuegen-Dialog* (siehe 3.3.1.4) klickt. Falls der Nutzer mehr Informationen zum Spiel benötigt, kann er hier ebenfalls mit dem „Hilfe“-Button auf den *Hilfebereich-Dialog* (siehe 3.3.1.8) gelangen. Durch den „Einstellungen“-Button gelangt der Spieler auf den *Einstellungen-Dialog* (siehe 3.3.1.6) um entsprechende Spieländerungen vorzunehmen. Zudem gibt es ein „Zurück“-Button, der den Nutzer wieder auf den *Startmenü-Dialog* (siehe 3.3.1.1) bringt.

3.3.1.3 Registrieren

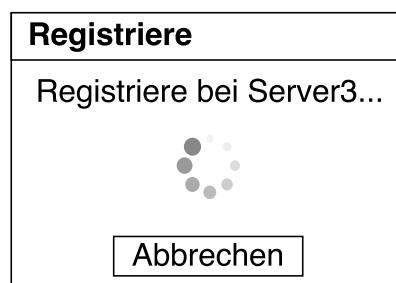


Abbildung 3.7: Registrieren-Dialog

Dieses Textfeld erscheint, nachdem der Nutzer im *Server-Suchen-Dialog* (siehe 3.3.1.2) registrieren gedrückt hat. Es wird angezeigt, bei welchem Server die Registrierung vorgenommen wird. Dass gerade versucht wird, eine Verbindung aufzubauen, wird durch einen runden Ladebalken dargestellt. Der Nutzer kann während des Registriervorgangs auf „Abbrechen“ klicken und die Registrierung beenden. Damit gelangt er wieder in den *Server-Suchen-Dialog* (siehe 3.3.1.2).

Ansonsten wird der Dialog automatisch verlassen, wenn die Registrierung entweder gelungen oder gescheitert ist. Falls sie gelingt, gelangt der Nutzer nun in die *Server-Lobby* (siehe 3.3.1.5). Falls nicht, taucht ein *Verbindungsauftbau-Fehlgeschlagen-Dialog* (siehe 3.4.1.1) auf.

3.3.1.4 ServerHinzufuegen

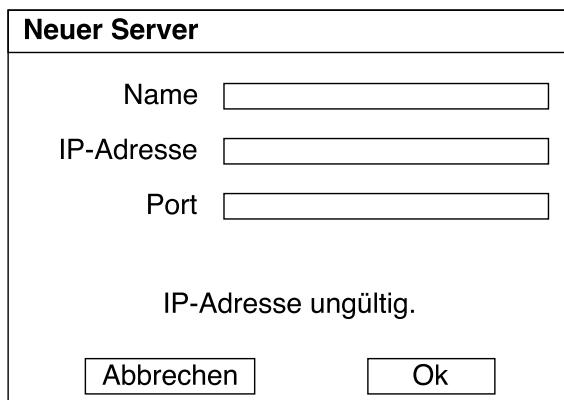


Abbildung 3.8: Server-Hinzufügen-Dialog

Wenn das Programm den vom Spieler gewünschten Server nicht automatisch erkennt, hat der Spieler die Möglichkeit, durch Klicken auf den entsprechenden Button einen neuen Server hinzuzufügen. Dann erscheint das Fenster, in dem der Spieler den Namen, die IP-Adresse und den Port des gewünschten Servers eintragen kann. Der „Ok“-Button ist ausgegraut und kann nicht angeklickt werden, bis der Spieler alle Felder ausgefüllt hat. Betätigt der Spieler dann den „Ok“-Button, wird überprüft, ob die Eingaben valide sind. Sollte etwas nicht richtig sein, wird der Spieler im unteren Teil des Fensters darüber informiert. In dem Beispiel-Fenster steht z. B. „IP-Adresse ungültig“. Weitere mögliche Meldungen sind „Portnummer ungültig“ und „Name ungültig“. Über den „Abbrechen“-Button kommt man wieder zurück zum *Server-Lobby-Dialog* (siehe 3.3.1.5).

3.3.1.5 ServerLobby

Wenn die Registrierung beim ausgewählten Server erfolgreich war, wird der Nutzer einem ausstehenden Spiel zugeteilt und kommt in die „Server Lobby“, wobei das obige Fenster (Abbildung 3.9) angezeigt wird. Oben im Dialog steht der Name des Servers, dem die Lobby angehört. In der linken Spalte „Ausstehende Spiele“ kann der

3 Benutzerschnittstellen

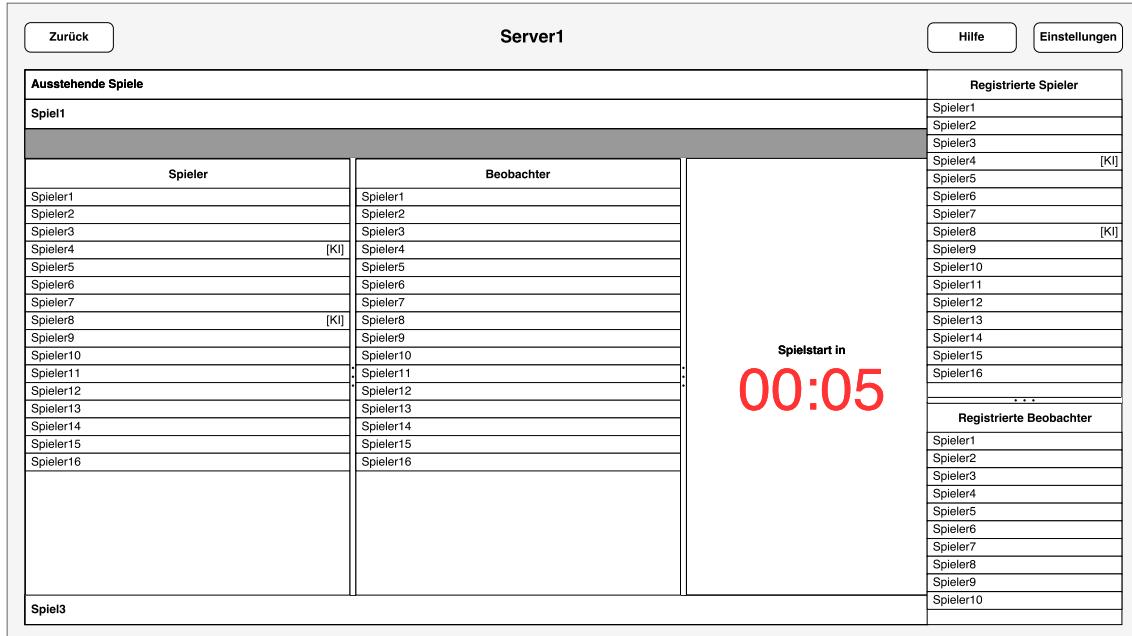


Abbildung 3.9: Server-Lobby-Dialog

Spieler alle aktuellen Spiele und in der rechten Spalte „Registrierte Spieler“ alle bei diesem Server angemeldeten Spieler sehen. Klickt der Nutzer auf ein Spiel, werden ihm weitere Informationen angezeigt: Im linken Feld unter dem Spiel alle Teilnehmer des Spiels und im rechten Feld ein Timer, der die verbleibende Zeit bis zum Spielstart herunterzählt.

Ganz normal führen die Buttons „Hilfe“ und „Einstellungen“ in die *Hilfebereich-Dialog* (siehe 3.3.1.8) und *Einstellungen-Dialog* (siehe 3.3.1.6). Der „Zurück“-Button führt den Nutzer zurück in den *Server-Suchen-Dialog* (siehe 3.3.1.2), wobei die Registrierung rückgängig gemacht wird.

3.3.1.6 Einstellungen

Dieser Dialog erscheint, nachdem ein Beobachter im *Pausebereich-Dialog* (siehe 3.3.2.2) auf den „Einstellungen“-Button geklickt hat. Der Dialog ist in drei Tabs aufgeteilt:

Im Tab „Bedienung“ kann man Bedienelemente wie Tastaturkürzel einstellen.

Im Tab „Grafik“ kann der Beobachter zwischen einer 2D- und 3D-Ansicht wechseln und die Auflösung anpassen. Im rechten Bereich sieht der Beobachter eine Vorschau seiner Einstellungen.

Im „Sound“-Tab kann der Beobachter die Musik- und Soundeffektlautstärke regulieren.

Über den „Zurück“-Button gelangt der Beobachter zurück in den *Pausebereich-Dialog* (siehe 3.3.2.2).

3.3 Benachrichtigungen

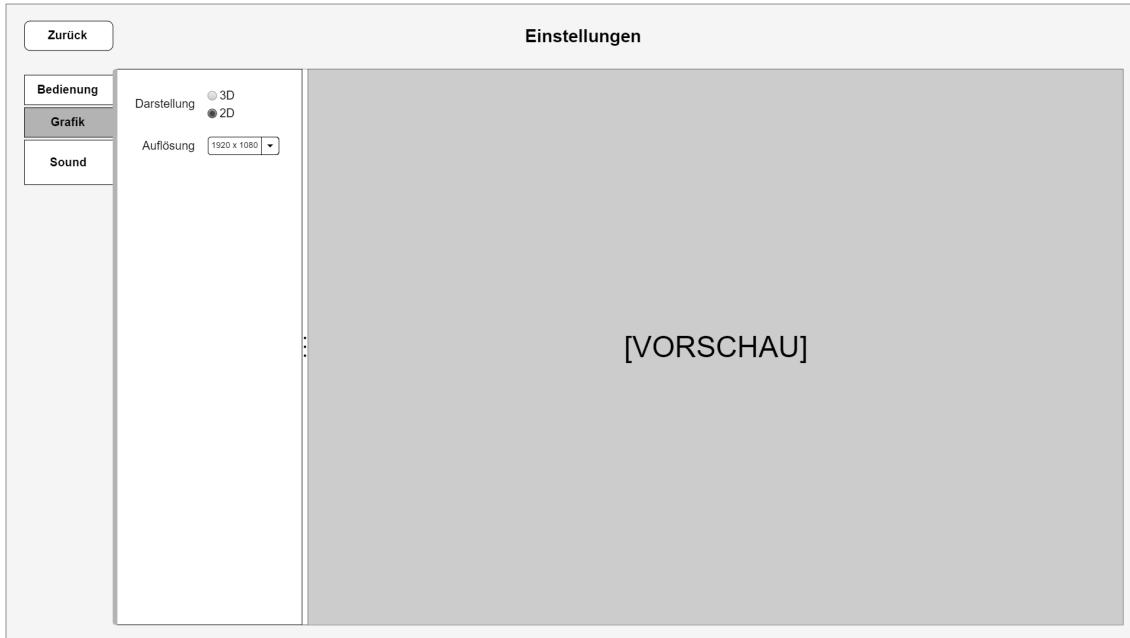


Abbildung 3.10: Einstellungen-Dialog

3.3.1.7 ProgrammBeenden

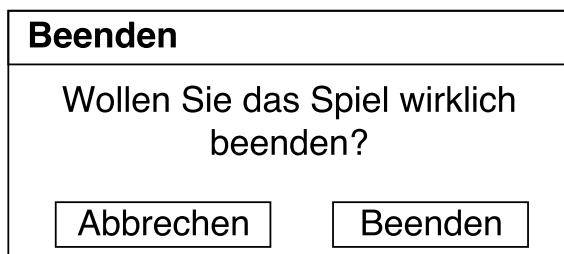


Abbildung 3.11: Programm-Beenden-Dialog

Der *Programm-Beenden-Dialog* erscheint, wenn der Nutzer im *Startmenü-Dialog* (siehe 3.3.1.1) auf den Button „Beenden“ geklickt hat. Der Nutzer muss in diesem Dialog bestätigen, dass er das Programm wirklich beenden möchte. Wenn er auf „Beenden“ klickt, beendet sich das Programm. Wenn der Spieler den „Abbrechen“-Button drückt, kehrt der Nutzer zum *Startmenü-Dialog* (siehe 3.3.1.1) zurück. Dies ist notwendig, da sich der Nutzer unter Umständen verklickt hat und das Programm eigentlich nicht beenden wollte.

3.3.1.8 Hilfebereich

Im Hilfebereich soll sich der Nutzer in den Grundzügen über die Spielregeln und die Funktionsweise der Software informieren können. Dieser Dialog ist in unterschiedliche Tabs unterteilt, die abwechselnd aufgerufen werden können. Der Tab Bedienung

3 Benutzerschnittstellen

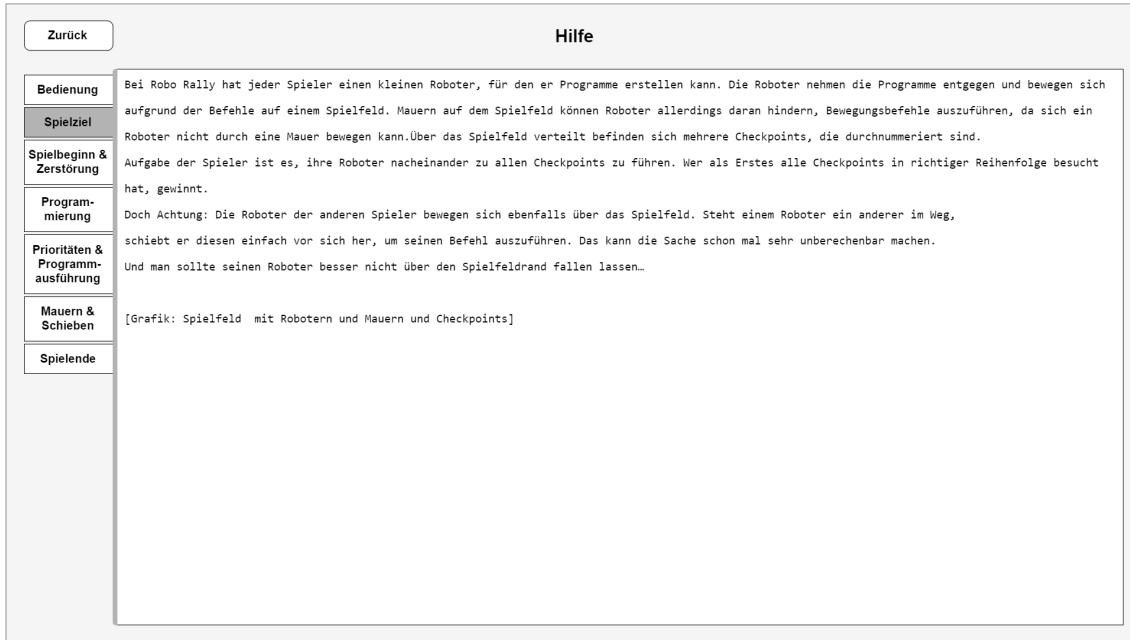


Abbildung 3.12: Hilfebereich-Dialog

bezieht sich auf die Software, die anderen Tabs sollen hauptsächlich die Spielregeln erklären. Der „Zurück“-Button zu dem Dialog zurück, aus dem heraus man den Hilfebereich aufgerufen hat.

Die konkreten Inhalte sind unten aufgeführt. Grafiken werden an den gekennzeichneten Stellen eingefügt, sobald die Grafiken für die GUI erstellt sind. Der Hilfebereich ist vor allem auf Mensch-Spieler und Beobachter ausgelegt, ist aber auch aus dem Server heraus aufrufbar.

TAB BEDIENUNG

Nach Programmstart befindet man sich im Startmenü. Um als Beobachter oder Spieler an einem Spiel teilzunehmen, muss man im Startmenü auf Serversuche klicken. Daraufhin wird eine Liste an verfügbaren Servern angezeigt. Ist der gewünschte Server nicht aufgeführt, kann man auch auf *Neuen Server hinzufügen* klicken. Dort hat man die Möglichkeit, mit Name, IP-Adresse und Port, direkt einen konkreten Server zu beschreiben um so vielleicht doch noch eine Verbindung aufzubauen. Nachdem „Ok“ gedrückt wird, kommt man wieder in die Serversuche. Dort kann ein Server ausgewählt, ein Name, unter dem gespielt/beobachtet wird, eingetragen, und auf Registrieren geklickt werden.

Wenn die Verbindung steht, gelangt man in den Lobby-Bereich. Hier muss gewartet werden, bis das Spiel beginnt. Das Spiel startet dann von selbst.

Wenn das Spiel läuft, kann nur der Server eine Pause veranlassen. Auf dem Spielfeld wird dennoch ein „Pause“-Button angezeigt. Damit gelangt man in einen Bereich, aus dem heraus die Einstellungen geändert oder dieser Hilfebereich aufgerufen werden kann. *Das Spiel läuft aber währenddessen weiter!*

Spielern wird auf der Spieloberfläche eine Schaltfläche namens KI angezeigt. Wenn man nicht mehr spielen möchte, kann dieser Schalter umgelegt werden. Dann spielt eine künstliche Intelligenz das Spiel weiter. Durch erneutes Drücken des Schalters kann die KI wieder abgeschaltet werden, allerdings erst am Ende der Runde.

TAB SPIELZIEL (VGL. BEISPIELTEXT IM DIALOG-PROTOTYPEN)

Bei Robo Rally hat jeder Spieler einen kleinen Roboter, für den er Programme erstellen kann. Die Roboter nehmen die Programme entgegen und bewegen sich aufgrund der Befehle auf einem Spielfeld. Mauern auf dem Spielfeld können Roboter allerdings daran hindern, Bewegungsbefehle auszuführen, da sich ein Roboter nicht durch eine Mauer bewegen kann.

Über das Spielfeld verteilt befinden sich mehrere Checkpoints, die durchnummieriert sind. Aufgabe der Spieler ist es, ihre Roboter nacheinander zu allen Checkpoints zu führen. Wer als Erstes alle Checkpoints in richtiger Reihenfolge besucht hat, gewinnt. Doch Achtung: Die Roboter der anderen Spieler bewegen sich ebenfalls über das Spielfeld. Steht einem Roboter ein anderer im Weg, schiebt er diesen einfach vor sich her, um seinen Befehl auszuführen. Das kann die Sache schon mal sehr unberechenbar machen. Und man sollte seinen Roboter nicht über den Spielfeldrand fallen lassen.

TAB SPIELBEGINN UND ZERSTÖRUNG

Jeder Roboter wird bei der Ausführung seines ersten Befehls im Spiel am ersten Checkpoint eingesetzt. Dafür muss ein Spieler seinem Roboter vor dem ersten Zug eine Richtung vorgeben. Dies geschieht während der Zeit für die Programmierung. Neben den Befehlskarten, die den Spielern im unteren Spielfeldbereich angezeigt werden (siehe Programmierung), erscheint eine Art Kompass, in dem die Richtung eingestellt werden kann.

Der Roboter kann während des Spiels zerstört werden. Dies geschieht entweder,

- wenn der Roboter über den Spielfeldrand fällt,
- der Roboter auf einem Checkpoint steht, während ein anderer Roboter genau dort eingesetzt wird, oder
- das Programm nicht rechtzeitig zusammengestellt wird.

Wenn der Roboter zerstört wurde, setzt er für die restliche Runde aus. In der nächsten Runde wird er mit der ersten Befehlausführung genau wie in der ersten Runde an dem Checkpoint eingesetzt, den er zuletzt erreicht hat. Also muss auch wieder eine Richtung angegeben werden.

Wird ein Roboter jedoch zum dritten Mal zerstört, wird der zugehörige Spieler disqualifiziert und scheidet aus dem Spiel aus.

TAB PROGRAMMIERUNG

Zu Beginn jeder Runde erhält jeder Spieler zufällige neun Karten. Auf jeder Karte steht ein Roboterbefehl. Es gibt sieben verschiedene Befehle:

3 Benutzerschnittstellen

- 1 Feld vorwärts
- 2 Felder vorwärts
- 3 Felder vorwärts
- 1 Feld rückwärts
- Vierteldrehung gegen den Uhrzeigersinn
- Vierteldrehung mit dem Uhrzeigersinn
- Halbe Drehung (180°)

Die Karten werden den Spielern im unteren Bildschirmbereich aufgefächert angezeigt. Man kann sie mit der Maus oder den Pfeiltasten auf der Tastatur durchblättern.

Jetzt folgen einige Sekunden Zeit, um das Roboterprogramm zusammenzustellen. Man muss fünf der neun Karten auswählen und in eine Reihenfolge bringen. Eine Karte wird durch gedrückte Maustaste ausgewählt und auf den gewünschten der fünf freien Plätze im eigenen erstellten Programm eingefügt. Alternativ kann der im Kartenfächer vorne liegenden Karte auch mit der Tastatur mit den Ziffern 1 bis 5 der entsprechenden Platz im Roboterprogramm zugewiesen werden.

In der ersten Runde und nach einer Zerstörung müssen Roboter mit Beginn der Befehlsausführung neu an einem Checkpoint eingesetzt werden. Wie unter „Spielbeginn und Zerstörung“ beschrieben, musst der zugehörige Spieler dann während der Programmierzeit auch eine Bewegungsrichtung für den ersten Befehl vorgeben.

Um zu verhindern, dass reihenweise Roboter zerstört werden, muss der erste Befehl nach dem Einsetzen (egal, ob in der ersten Runde oder nach einer Zerstörung) ein Laufbefehl sein, also den Roboter vorwärts oder rückwärts bewegen. So ist gewährleistet, dass sich erst einmal alle Roboter von den Checkpoints weg bewegen.

Wie viel Zeit den Spielern in der aktuellen Runde zum programmieren bleibt, ist am Countdown erkennbar. Den Spielern wird er oberhalb des Kartenbereichs angezeigt, den Beobachtern oben mittig. Ist die Zeit abgelaufen, müssen die Programme fertig sein; ein Roboter ohne vollständiges Programm wird zerstört.

TAB PRIORITÄTEN UND PROGRAMMAUSFÜHRUNG

Nach Ablauf der Zeit werden Stück für Stück die Roboterprogramme abgearbeitet. Zuerst wird der jeweils erste Befehl jedes Spielers ausgeführt, dann alle zweiten Befehle und so weiter. Welcher erster, zweiter, ..., fünfter Befehl welches Spielers aber zuerst ausgeführt wird, entscheidet sich durch Prioritäten.

Prioritäten sind Zahlen, die auf den Befehlskarten vermerkt sind, welche den Spielern ausgeteilt wurden. Die Prioritäten sind in einem Spiel über alle Karten hinweg einzigartig. Die Befehle der Spieler werden also in absteigender Reihenfolge der Prioritäten auf den Karten ausgeführt.

Beispiel: Sie und Rosi spielen gegeneinander Robo Rally.

3.3 Benachrichtigungen

Sie haben ein Programm aus Befehlen mit den Prioritäten (1), (5), (7), (2), (15) zusammengestellt, Rosi eines mit den Prioritäten (12), (3), (4), (9), (16).

Also hier wird erst Rosis Befehl mit der Priorität (12) ausgeführt, dann der Befehl von Ihnen, da 12 höher ist als 1. Dann kommen nach dem selben Prinzip nacheinander die Befehle mit den Prioritäten (5), (3), (7), (4), (9), (2), (16), (15).

Bemerkung: Die Prioritäten sind zwar zufällig, allerdings hat jede Art von Befehl einen Rang bezüglich den Prioritäten. Jede 90°-Drehung hat zum Beispiel immer eine höhere Priorität als jede 180°-Drehung. Die Rangfolge ist absteigend wie folgt:

- 3 Felder vorwärts
- 2 Felder vorwärts
- 1 Feld vorwärts
- 1 Feld rückwärts
- Vierteldrehung gegen den Uhrzeigersinn
- Vierteldrehung mit dem Uhrzeigersinn
- Halbe Drehung (180°)

TAB MAUERN UND SCHIEBEN

Roboter können nicht durch Mauern gehen. Führt dein Roboter eine Bewegung aus, der eine Mauer im Weg steht, so wird die Bewegung an der Mauer abgebrochen und der Roboter bleibt davor stehen.

Grundsätzlich können niemals zwei Roboter gleichzeitig auf demselben Feld stehen. Führt ein Roboter eine Bewegung aus, der ein anderer Roboter im Weg steht, so wird dieser einfach um die entsprechende Anzahl an Feldern mitgeschoben. Ein Roboter kann auch mehrere Roboter voreinander herschieben.

Natürlich kann auch ein geschobener Roboter nicht durch Wände gehen. Wird ein Roboter gegen eine Wand geschoben, bleibt er stehen, sowie alle schiebenden Roboter, die hinter ihm kommen.

TAB SPIELENDE

Das Spiel endet, sobald ein Roboter den letzten Checkpoint erreicht, und zuvor alle anderen Checkpoints in der richtigen Reihenfolge besucht hat. Dabei ist es egal, ob der Roboter auf dem Feld des Checkpoints stehen bleibt oder nur darüberzieht; es macht auch nichts wenn der Roboter hinterher in der selben Bewegung über den Spielfeldrand hinaus fällt. Egal ist auch, ob der Roboter selbst die Bewegung als Befehl ausführt oder ob er von einem anderen Roboter geschoben wird. Andere Roboter, die den Checkpoint hinterher im Laufe der selben Runde ebenfalls erreicht hätten, haben nicht gewonnen.

3 Benutzerschnittstellen

3.3.2 Im Spiel (Beobachter)

3.3.2.1 Beobachter

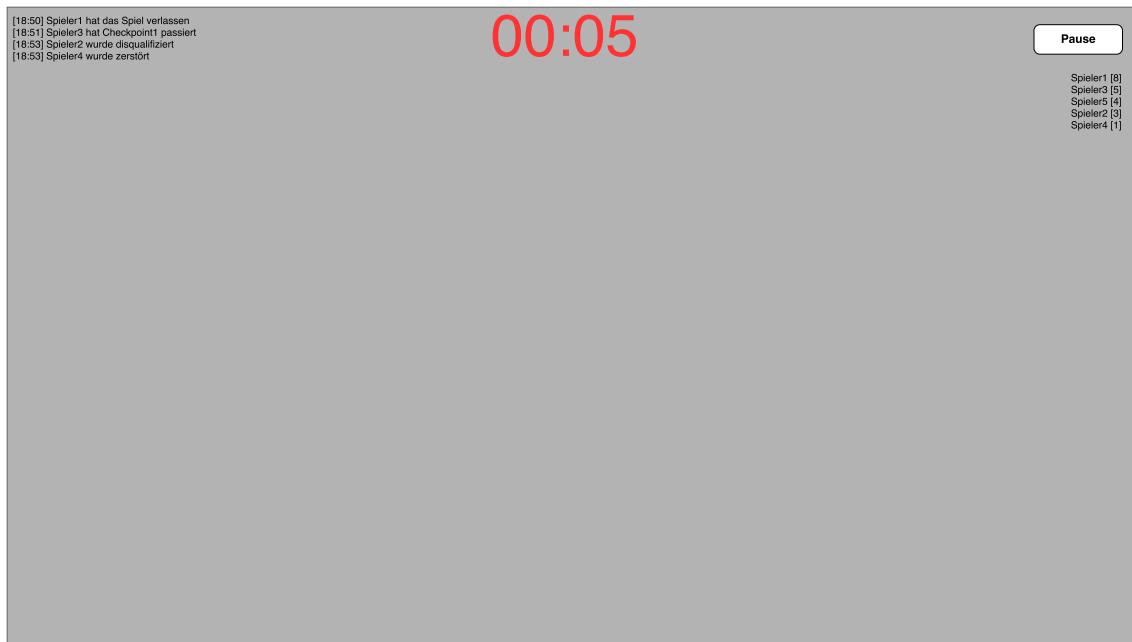


Abbildung 3.13: Beobachter-Dialog

Auf der Spieloberfläche des Beobachters werden während des Spiels oben links Beachrichtigungen über das Spiel eingeblendet. Dies beinhaltet Informationen wie: Runde X hat begonnen, Spieler XY wurde zerstört, Spieler XY hat Checkpoint * passiert, Spieler XY wurde disqualifiziert. Der rote Counter oben mittig zeigt während der Programmierphase die Zeit an, die den Spielern verbleibt, um ihre Roboter zu programmieren, bis die nächste Visualisierung beginnt. Oben rechts befindet sich ein „Pause“-Button, welcher jederzeit angeklickt werden kann. Man gelangt dadurch zum *Pausebereich-Dialog* (siehe 3.3.2.2). Man beachte, dass dieser Button das Spiel keineswegs pausiert, sondern lediglich dem Beobachter erlaubt, die Spielfläche zu verlassen, um beispielsweise Einstellungen oder den Hilfebereich aufzurufen. Rechts an der Seite befindet sich ein Ranking, welches die Spieler mit der Nummer des jeweils höchsten erreichten Checkpoints zusammen aufführt.

3.3.2.2 Pausebereich

Klickt der Beobachter auf der *Spieloberfläche* (siehe 3.3.2.1) auf „Pause“, öffnet sich dieser *Pausebereich-Dialog*. Hier kann der Beobachter durch Klicken des „Spiel fortsetzen“-Buttons wieder zurück auf die *Spieloberfläche* (siehe 3.3.2.1), über den „Hilfe“-Button in den *Hilfebereich* (siehe 3.3.2.5), und durch Drücken des „Einstellungen“-Buttons zu den *Einstellungen* (siehe 3.3.2.4). Wenn der Beobachter den „Spiel beenden“-Button betätigt, verlässt er das Spiel und befindet sich wieder im *Startmenü* (siehe 3.3.1.1).

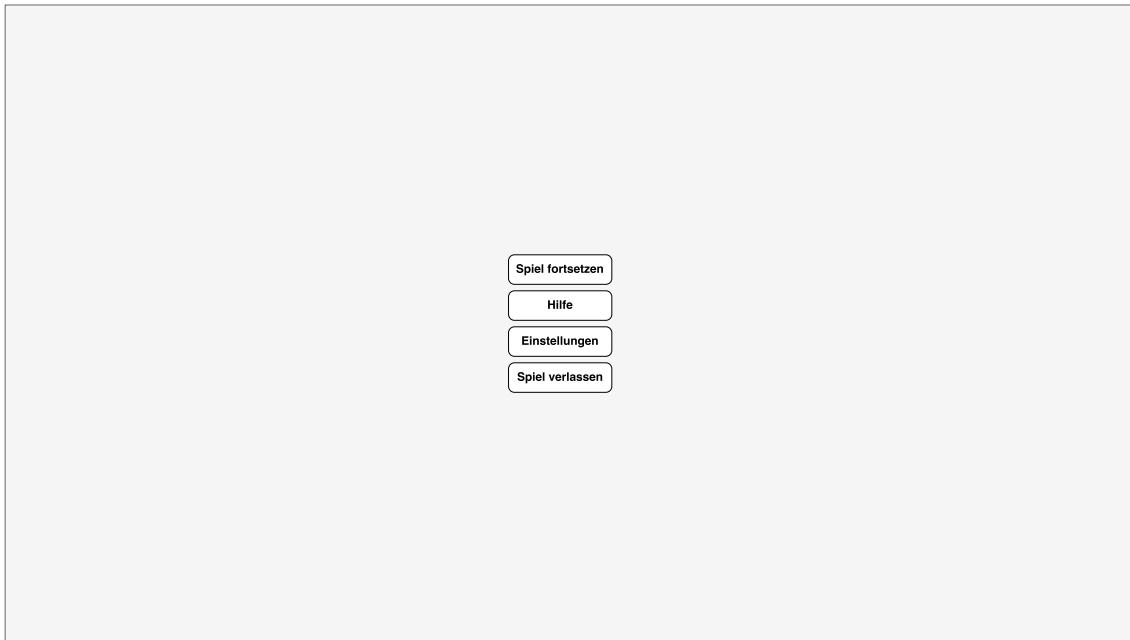


Abbildung 3.14: Pausebereich-Dialog

3.3.2.3 Spielende



Abbildung 3.15: Spielende-Dialog

Am Ende jedes Spiels erscheint ein Text, der den Beobachtern Auskunft gibt, wer gewonnen hat. Nach dem Drücken des „Ok“-Buttons kommen die Beobachter wieder ins *Startmenü* (siehe 3.3.1.1).

3.3.2.4 EinstellungenBeobachter

Der Dialog „Einstellungen“, der vom Beobachter während des Spiels aufgerufen werden kann, ist identisch zu dem, den Beobachter und Spieler vor dem Spiel aufrufen können (siehe 3.3.1.6).

3.3.2.5 HilfebereichBeobachter

Durch den „Hilfe“-Button im *Pausebereich-Dialog* (siehe 3.3.2.2) gelangt man in den Hilfebereich. Der Hilfebereich ist identisch zu dem Hilfebereich, der von Spieler,

3 Benutzerschnittstellen

Beobachter und Server vor dem Spiel einesehen werden kann. Über den „Zurück“-Button gelangt der Beobachter in den *Pausebereich-Dialog* (siehe 3.3.2.2).

3.3.3 Im Spiel (Spieler)

Alle Dialoge des Benutzers „*Im Spiel (Beobachter)*“ (siehe 3.3.2) existieren ebenfalls für den Spieler. Manche Dialoge sind allerdings abgewandelt, was im Folgenden beschrieben wird.

3.3.3.1 Spieloberfläche

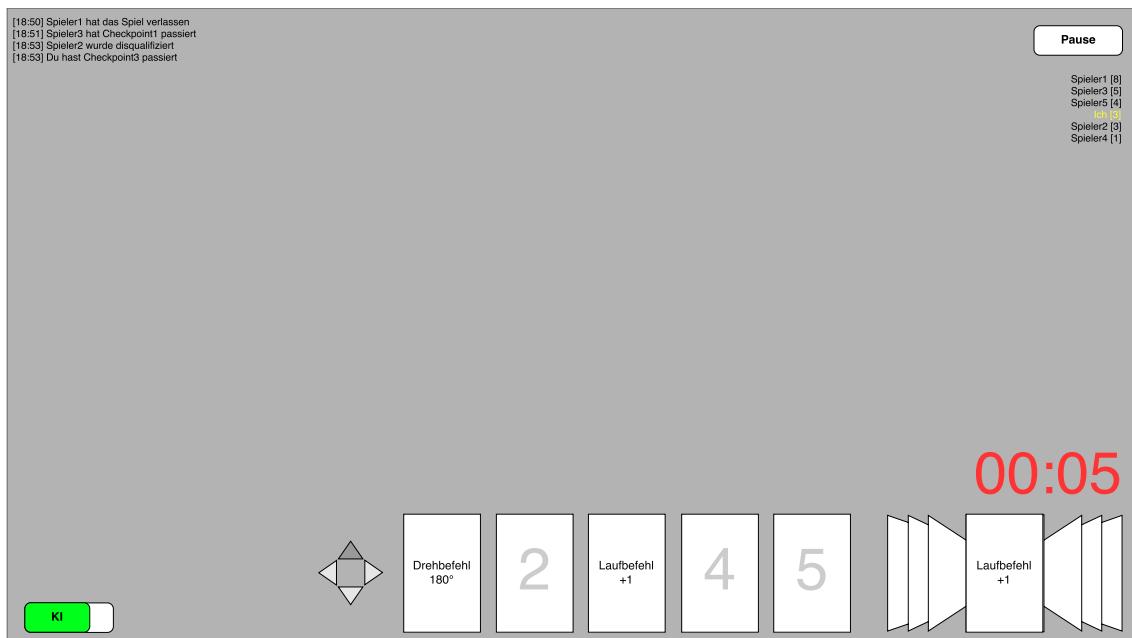


Abbildung 3.16: Spieloberflaeche-Dialog

Die Spieloberfläche des Spielers ist eine Erweiterung der Spieloberfläche des Beobachters. Identisch sind die Benachrichtigungen im linken oberen Spielfeldbereich, der „Pause“-Button oben rechts, der in den *Pausebereich* (siehe 3.3.2.2) führt (wobei auch hier das Spiel nicht echt pausiert) und die Rangliste an der rechten Seite. In der Rangliste besteht der kleine Unterschied, dass der Spieler selbst nicht mehr mit dem Spielernamen, sondern mit „Ich“ aufgeführt ist.

Der Countdown befindet sich im Gegensatz zu oben mittig beim Beobachter nun beim Spieler unten rechts über dem Kartenbereich, erfüllt aber dieselbe Funktion. Neue Komponenten sind besagter Kartenbereich rechts im unteren Spielfeldbereich sowie der Kompass links daneben und der Schalter „KI“ im linken unteren Eck.

Der Kartenbereich umfasst unter anderem die Karten, die dem Spieler zu Rundenbeginn ausgeteilt werden. Sie werden aufgefächert im rechten unteren Eck angezeigt. Mit der Maus oder den Pfeiltasten der Tastatur können die Karten durchblättert werden, sodass jede Karte nach vorne geholt werden kann. Links neben dem Kartenstapel befinden sich fünf nummerierte Platzhalter, auf denen der Spieler während der Programmierphase sein Programm erstellen kann. Per drag-and-drop oder Tastatur kann der Spieler die Karten aus dem Stapel auf die Platzhalter ziehen, wodurch sie im

3 Benutzerschnittstellen

Kartenstapel nicht mehr zu sehen sind. Eine Auswahl kann korrigiert werden, indem eine neue Karte auf einen bereits belegten Platzhalter gezogen wird. Die alte Karte wird dann dem Kartenstapel wieder hinzugefügt. Während der Visualisierungsphase ist der Kartenstapel leer und auf den Platzhaltern wird das, soweit vorhandene, aktuell ausgeführte Programm angezeigt. Eine Besonderheit tritt in der ersten Runde, sowie wenn der Roboter in der vorherigen Runde zerstört wurde, auf: Auf den ersten Platzhalter darf dann kein Drehbefehl gezogen werden. Versucht der Spieler dies, geht die Karte einfach in den Kartenstapel zurück.

Der Kompass kommt ebenfalls in der Programmierphase der ersten Runde und nach einer Zerstörung zum Tragen. Indem der Spieler auf eine Pfeilrichtung klickt, legt er die Bewegungsrichtung fest, mit der der Roboter beim ersten Befehl eingesetzt wird. Er kann diese Richtung auch während der Programmierphase ändern. Per Default ist der Pfeil Richtung Norden aktiviert. In Runden, in denen der Kompass nicht benötigt wird, ist er ausgegraut.

Der Schalter „KI“ ermöglicht dem Spieler, eine Runde oder das ganze Spiel von einer künstlichen Intelligenz für ihn zu Ende spielen zu lassen. Diese Entscheidung kann der Spieler erst nach Beendigung einer Runde wieder rückgängig machen. Drückt der Spieler einmal auf den Schalter, wird die KI aktiviert, der Schalter bewegt sich nach rechts und wird rot und Kartenbereich und Kompass werden ausgeblendet. Beim erneuten Drücken des Schalters wird der Schalter erst einmal grau und bewegt sich zurück nach rechts. Dieser Zustand dauert an, bis die aktuelle Runde beendet ist. Dann wird der Schalter wieder grün, Kartenbereich und Kompass werden eingeblendet und der Spieler kann wieder selbst weiterspielen.

3.3.3.2 PausebereichSpieler

Klickt der Spieler auf der *Spieloberfläche* (siehe 3.3.3.1) auf „Pause“, öffnet sich der *Pausebereich-Dialog* (siehe 3.14). Hier kann der Spieler durch Klicken des „Spiel fortsetzen“-Buttons wieder zurück auf die *Spieloberfläche* (siehe 3.3.3.1), über den „Hilfe“-Button in den *Hilfebereich* (siehe 3.3.2.5), und durch Drücken des „Einstellungen“-Buttons zu den *Einstellungen* (siehe 3.3.2.4). Wenn der Spieler den „Spiel beenden“-Button betätigt, erscheint der *SpielVerlassen-Dialog* (siehe 3.3.3.6).

3.3.3.3 GameOver

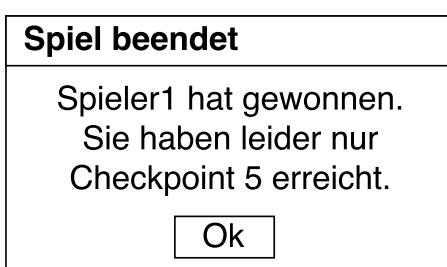


Abbildung 3.17: GameOver-Dialog

Dieser Dialog wird bei Spielende allen Spielern angezeigt, die nicht gewonnen haben. Im oberen Bereich steht, dass das Spiel beendet wurde. Ist der Spieler nicht selbst der Gewinner, steht darunter welcher Spieler gewonnen hat (hier Spieler1). Außerdem wird angegeben, welchen Checkpoint der Spieler selbst als letztes erreicht hat. Wenn der Spieler selbst gewinnt, steht im Dialog „Sie haben gewonnen. Herzlichen Glückwunsch!“.

Klickt der Spieler auf „Ok“, kommt er zurück ins Startmenü und der *Startmenü-Dialog* (siehe 3.3.1.1) wird angezeigt.

3.3.3.4 Karten

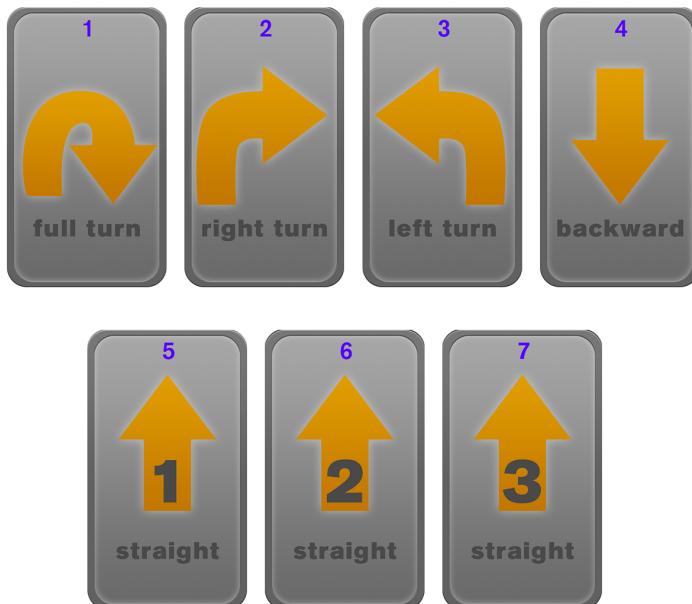


Abbildung 3.18: Karten

Abbildung 3.18 zeigt die 7 verschiedenen Karten im Spiel. Die blaue Zahl am oberen Rand jeder Karte gibt die Priorität an die jede Karte besitzt. Des weiteren gibt der Text verbunden mit dem Pfeil die Art der Karte an.

3.3.3.5 Disqualifikation

Wird ein Spieler disqualifiziert, wird dieser Dialog angezeigt. Wenn der Spieler auf „Ok“ klickt, kommt er zurück ins Startmenü und der Spieler sieht wieder den *Startmenü-Dialog* (siehe 3.3.1.1).

3.3.3.6 SpielVerlassen

Klickt der Spieler im *Pausebereich-Dialog* (siehe 3.3.2.2) auf „Spiel verlassen“, wird dieses Fenster angezeigt. Im oberen Bereich steht „Spiel verlassen“ und darunter

3 Benutzerschnittstellen

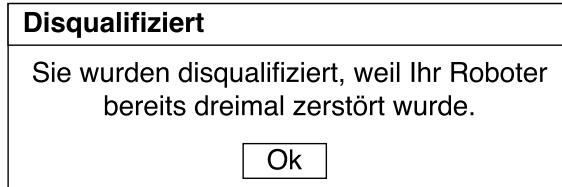


Abbildung 3.19: Disqualifikation-Dialog

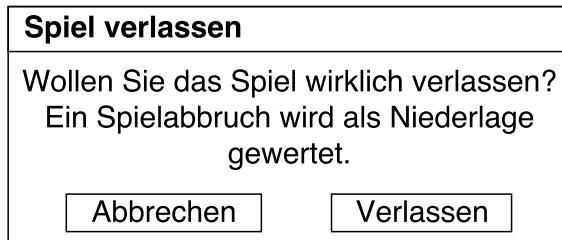


Abbildung 3.20: Spiel Verlassen-Dialog

„Wollen Sie das Spiel wirklich verlassen? Ein Spielabbruch wird als Niederlage gewertet.“. Der Spieler kann den Vorgang entweder durch Klicken auf den „Abbrechen“-Button abbrechen oder das Verlassen des Spiels durch Drücken auf den „Verlassen“-Button bestätigen. Im ersten Fall wird das Fenster geschlossen und der Spieler kommt wieder in den Pausebereich, wobei der *Pausebereich-Dialog* (siehe 3.3.2.2) wieder angezeigt wird. Im zweiten Fall beendet der Spieler das Spiel und kommt zurück ins Startmenü, in dem der *Startmenü-Dialog* (siehe 3.3.1.1) zu sehen ist.

3.3.4 Editor

3.3.4.1 Leveleditor

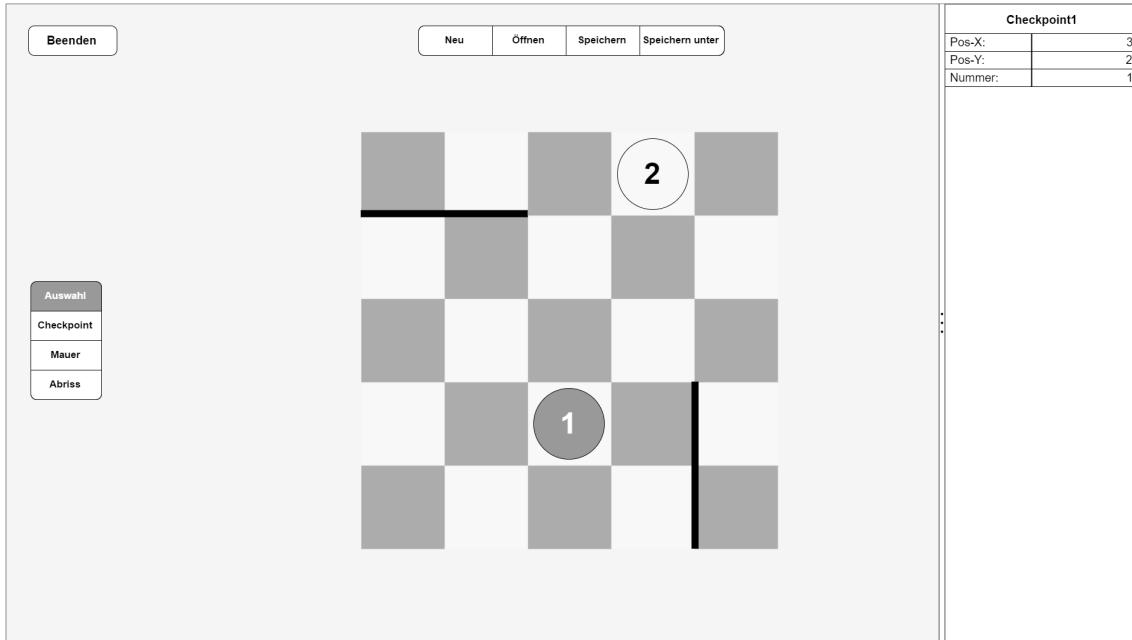


Abbildung 3.21: Leveleditor-Dialog

Im Leveleditor werden die Spielfelder für das Spiel erstellt. Links kann auf die Optionen Checkpoint, Mauer und Abriss geklickt werden. Wählt man „Checkpoint“ aus, kann man auf dem Spielfeld eine beliebige Stelle aussuchen um seinen Checkpoint zu setzen. Auf der rechten Seite erscheinen dann zusätzliche Informationen wie X- und Y-Position des Checkpoints und die Nummer. Wählt man „Mauer“ aus, kann auf dem Spielfeld eine beliebige Kante ausgewählt werden, um eine Mauer zu setzen. Wird „Abriss“ ausgewählt, können Checkpoints und Mauern aus dem Spielfeld entfernt werden. Der „Neu“-Button erstellt ein neues Spielfeld, das noch keine Mauern oder Checkpoints besitzt.

Der „Öffnen“-Button öffnet ein gespeichertes Spielfeld, damit darauf weitergearbeitet werden kann. Unter *LevelLaden-Dialog* (siehe 3.3.4.4) kann ein bereits gespeichertes Spielfeld geöffnet werden. Das erstellte Spielfeld kann mit dem „Speichern“-Button gespeichert werden. Der „Speichern unter“-Button ermöglicht es, ein Spielfeld in einem bestimmten Ordner zu speichern. Dabei öffnet sich ein *LevelSpeichern-Dialog* (siehe 3.3.4.3). Dieser Dialog öffnet sich ebenfalls beim „Speichern“, falls ein Spielfeld zum ersten Mal gespeichert werden soll. Wenn auf „Beenden“ geklickt wird, wird das Programm beendet, falls zuvor gespeichert wurde. Ansonsten erscheint der *NichtGespeichert-Dialog* (siehe 3.3.4.2) und fragt nach, ob das Spielfeld gespeichert oder verworfen werden soll.

3 Benutzerschnittstellen

3.3.4.2 NichtGespeichert

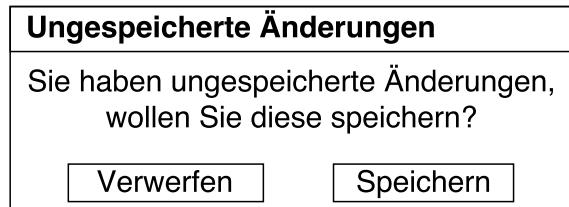


Abbildung 3.22: NichtGespeichert-Dialog

Falls der Leveleditor beendet, aber das Spielfeld zuvor nicht gespeichert wird, erscheint eine Abfrage ob die Änderungen an dem Spielfeld gespeichert werden soll oder nicht. Klickt man auf „Verwerfen“, schließt sich das Programm ohne seine Änderungen zu sichern. Wird „Speichern“ ausgewählt, erscheint beim ersten Mal Speichern ein *LevelSpeichern-Dialog* (siehe 3.3.4.3). Ansonsten wird ganz normal gespeichert ohne den Speicherort abzufragen.

3.3.4.3 LevelSpeichern

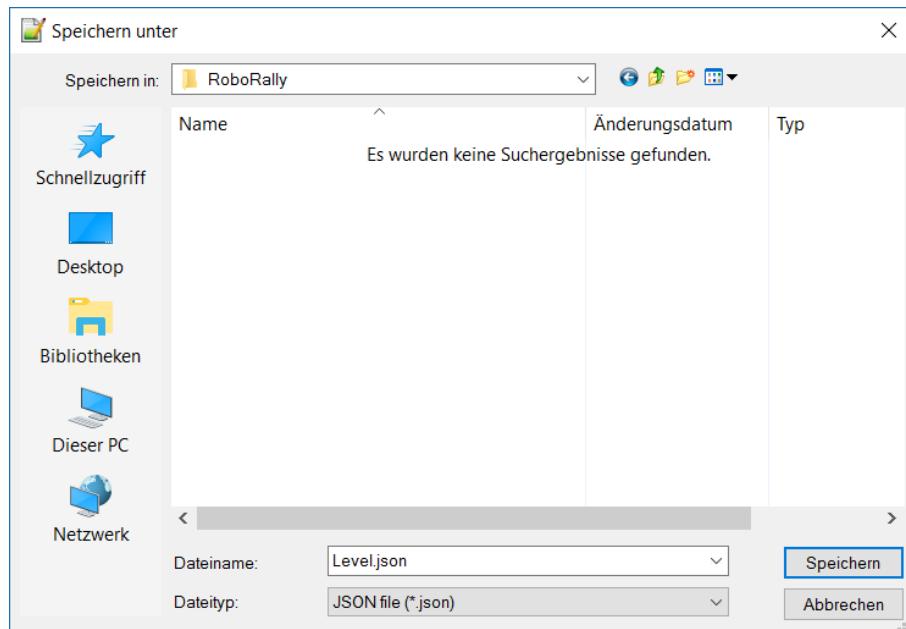


Abbildung 3.23: LevelSpeichern-Dialog

Wenn der Nutzer sein erstelltes Level speichern möchte und daher auf den „Speichern“-Button klickt, öffnet sich der *Level-Speichern-Dialog*. Hier kann der Nutzer den Ort auswählen, an dem die Datei gespeichert werden soll. Außerdem kann der Nutzer im unteren Eingabefeld einen Dateinamen festlegen. Wenn der Nutzer auf „Speichern“ drückt, wird das Level am ausgewählten Ort gespeichert. Falls der Nutzer keinen Dateiname eingetragen hat, wird ein Default-Name vergeben.

3.3.4.4 LevelLaden

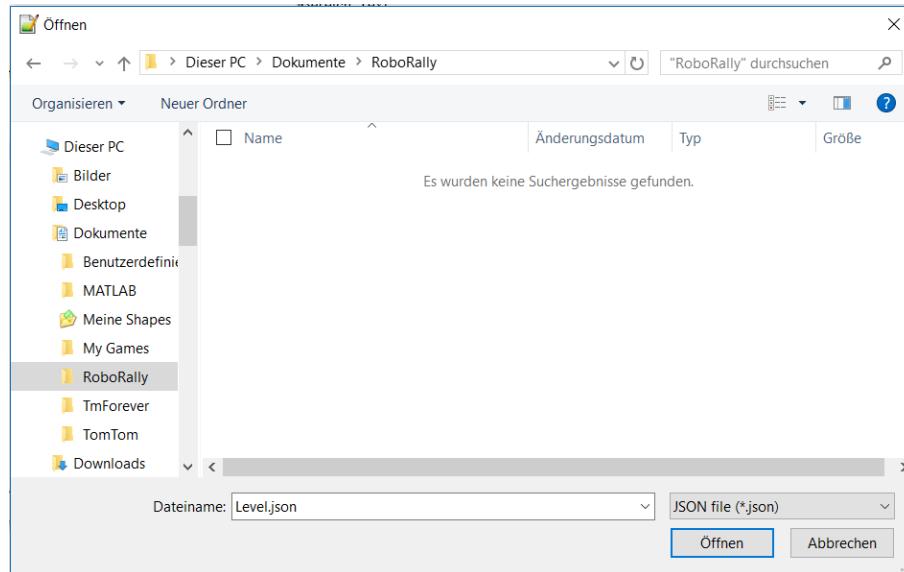


Abbildung 3.24: LevelLaden-Dialog

Dieser Dialog wird geöffnet, wenn ein Nutzer an einem zuvor erstellten Level weiter bauen möchte und auf den „Öffnen“-Button drückt. Hier kann der Nutzer die Leveledatei suchen und zum Öffnen auswählen. Durch das Klicken des „Öffnen“-Buttons wird die Datei bestätigt und im Editor geöffnet. Das Dropdown Menu in der rechten unteren Hälfte ermöglicht dem Nutzer, die Dateien nach bestimmten Endungen zu filtern.

3 Benutzerschnittstellen

3.3.5 Server

3.3.5.1 Server

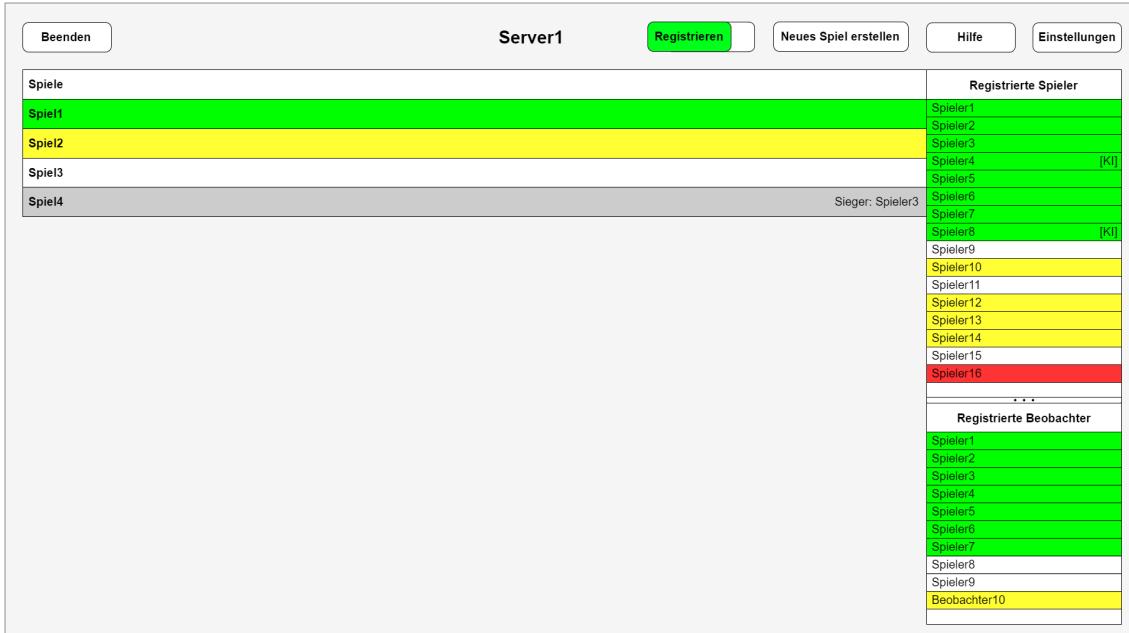


Abbildung 3.25: Server-Dialog

Im *Server-Dialog* wird oben mittig der Name des Servers angezeigt. Ein „Beenden“-Button im oberen linken Eck beendet den Server auf Knopfdruck. Dann folgen am oberen Rand von links nach rechts ein Schalter „Registrierung“, mit dem man einstellen kann, ob der Server zur Zeit Registrierungen akzeptiert oder zurückweist (grün bedeutet akzeptierend, rot zurückweisend), ein „Neues-Spiel-erstellen“-Button, der den *Neues-Spiel-Erstellen-Dialog* (siehe 3.3.5.2), ein „Hilfe“-Button, der den *Hilfebereich* aufruft (siehe 3.3.1.8) und ein „Einstellungen“-Button, der in die *Einstellungen* des Servers wechselt (siehe 3.3.5.4).

Am rechten Rand werden alle Spieler und Beobachter aufgelistet, die momentan am Server registriert sind. Falls es sich dabei bei einem Spieler um eine KI handelt, wird dies in eckigen Klammern hinter dem Namen vermerkt. Im Hauptbereich des Dialogs befindet sich eine aufklappbare Liste aller Spieler, die der Server erstellt hat. Die unterschiedlichen Farben haben dabei folgende Bedeutung:

- grün: Das Spiel wird momentan durchgeführt
- gelb: Das Spiel läuft noch nicht, wird aber in Kürze beginnen; der Countdown bis Spielbeginn läuft bereits
- weiß: Das Spiel hat noch nicht begonnen und kein Countdown läuft, es können noch weitere Spieler für das Spiel angemeldet werden
- grau: Das Spiel ist bereits vorbei. Es wird der Sieger des Spiels vermerkt

Die Spieler und Beobachter werden entsprechend der Farbe eingefärbt, die das Spiel besitzt, für das sie angemeldet sind. Spieler und Beobachter, die bei keinem Spiel angemeldet sind, sind ebenfalls weiß. Disqualifizierte Spieler sind rot.

Klickt man auf ein Spiel in der Liste, klappt es sich auf und weitere Details werden angezeigt; Näheres im Folgenden.

3.3.5.2 ServerNeuesSpielErstellen

Neues Spiel	
Name	<input type="text"/>
<input type="button" value="Abbrechen"/>	<input type="button" value="Ok"/>

Abbildung 3.26: Server-Neues-Spiel-Erstellen-Dialog

Soll ein Server für ein neues Spiel erstellt werden, gibt man seinen Namen ein und drückt anschließend auf „Ok“. Davor ist das „Ok“-Feld ausgegraut. Wird der Name angenommen erscheint ein *Server-Dialog* (siehe 3.3.5.1). Durch „Abbrechen“ wird kein Server und somit kein neues Spiel erstellt.

3.3.5.3 ServerLevelLaden

Hier wird das selbe Bild, welches auch schon beim Leveleditor verwendet wurde, benutzt: Dies ist Abbildung 3.24.

Der Server lädt ein Level aus den erstellten Leveln heraus. Durch Anklicken des „Öffnen“-Buttons wird das gewünschte Level in den *ServerNeuesSpiel-Dialog* (siehe 3.3.5.8) importiert. Er kann diese Aktionen aber auch durch den „Abbrechen“-Button schließen.

3.3.5.4 ServerEinstellungen

Einstellungen	
Servername	<input type="text"/>
Port	<input type="text"/>
<input type="button" value="Abbrechen"/>	<input type="button" value="Ok"/>

Abbildung 3.27: Server-Einstellungen-Dialog

In den Server Einstellungen kann man Einstellen, wie der Server heißen soll und auf welchem Port er freigegeben wird. Der Server startet nach der Bestätigung mit dem „Ok“-Button neu. Mit dem „Abbrechen“-Button kommt der Benutzer wieder in den *Server-Dialog* (siehe 3.3.5.1) zurück.

3 Benutzerschnittstellen

3.3.5.5 ServerLaufendesSpiel

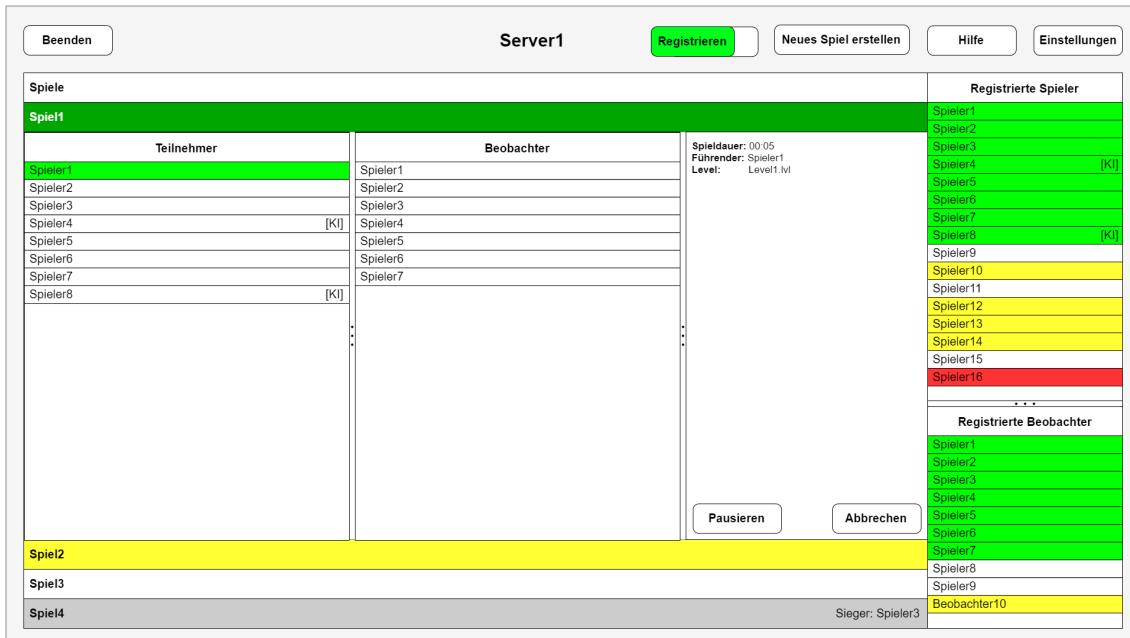


Abbildung 3.28: Server-LaufenSpiel-Dialog

Bei einem laufenden Spiel kann man mit Hilfe des „Pausieren“-Button das Spiel pausieren. Das Spiel wird dann für alle Spielteilnehmer und Beobachter eingefroren. Diese können erst wieder weiterspielen, wenn der Spielleiter im *Server-Pause-Dialog* (siehe 3.3.5.6) den „Fortsetzen“-Button gedrückt hat. Man kann das Spiel, wie auch schon zuvor über den „Abbrechen“-Button abbrechen. Außerdem werden alle teilnehmenden Spieler und Beobachter in der Mitte links und rechts angezeigt.

3.3.5.6 ServerPause

Wenn der Server das Spiel pausiert, erscheint rechts vom gewählten Spiel „[Pausiert]“. Es erscheint ebenfalls die Spieldauer, der führende Spieler und das Spielfeld, in dem die Teilnehmer spielen. Über den „Fortsetzen“-Button, wird das Spiel weitergeführt. Mit dem „Abbrechen“-Button wird das Spiel ganz beendet.

3.3.5.7 ServerSpielGestartet

Während dem Spiel erkennt der Server in welchem Spiel er sich befindet, welcher Teilnehmer und Beobachter es gibt und in welchem Level die Spieler sein werden. Mit einem Countdown werden die Sekunden heruntergezählt bis ein Spiel gestartet wird. Der Server hat dabei noch die Möglichkeit sich umzuentcheiden und über den „Abbrechen“-Button das Spiel vorzeitig zu beenden.

3.3 Benachrichtigungen

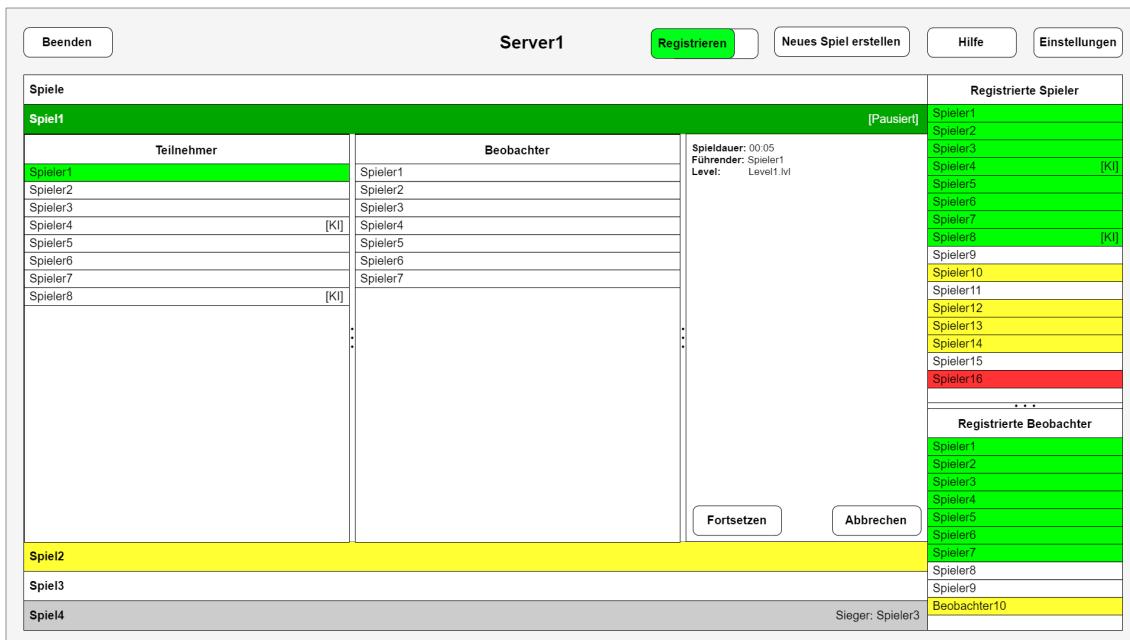


Abbildung 3.29: Server-Pause-Dialog

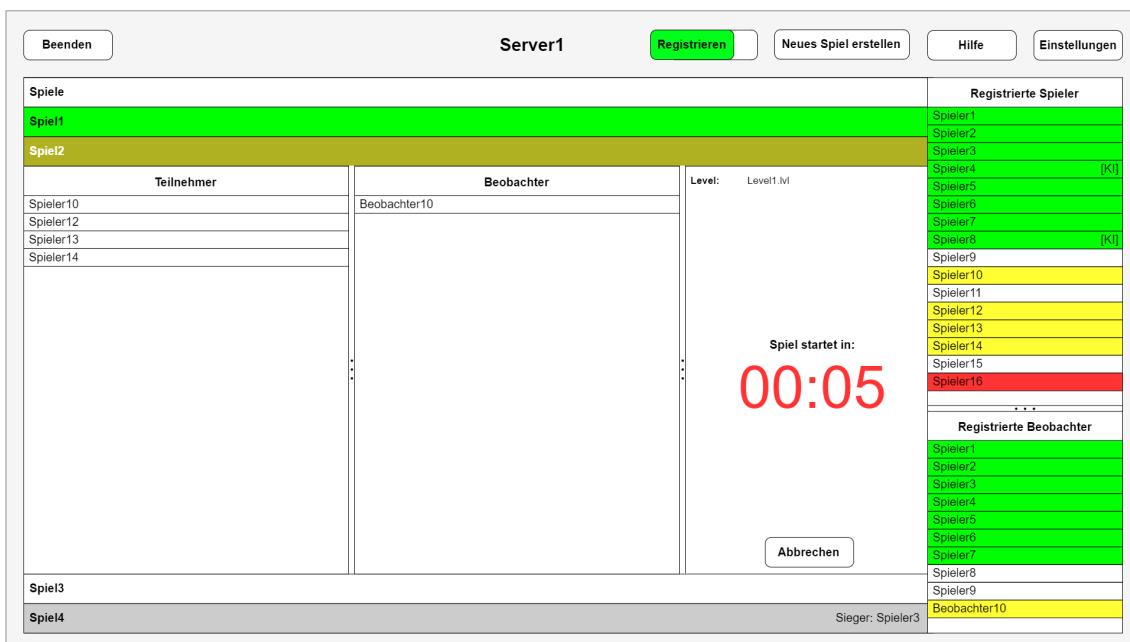


Abbildung 3.30: Server-Spiel-Gestartet-Dialog

3 Benutzerschnittstellen

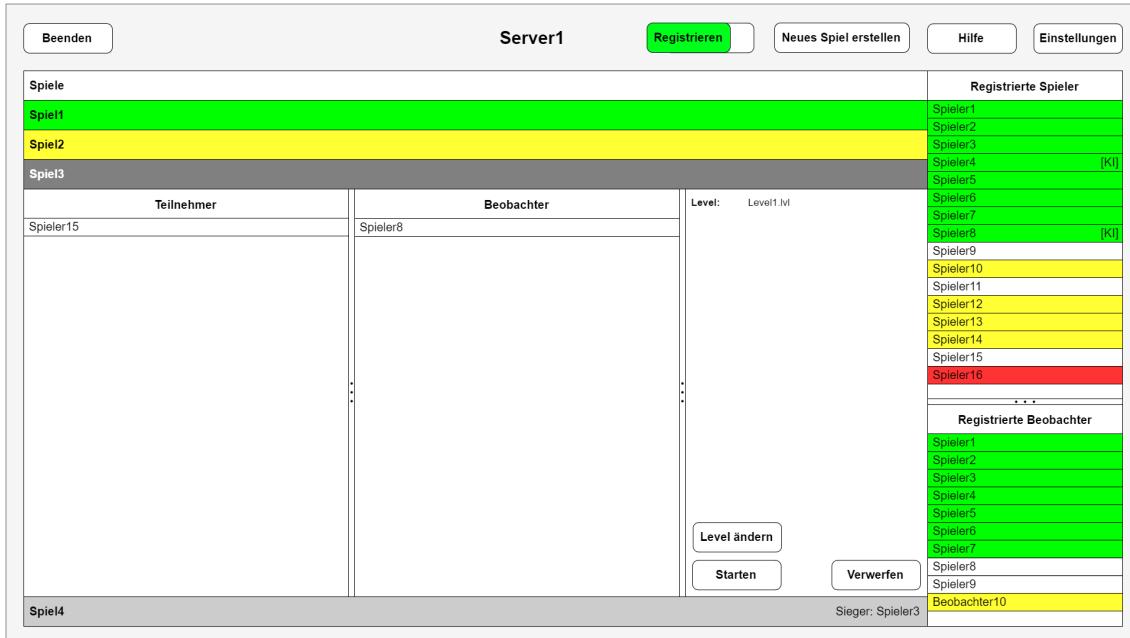


Abbildung 3.31: Server-Neues-Spiel-Dialog

3.3.5.8 ServerNeuesSpiel

Wenn der Server ein neues Spiel erstellen will, wählt er den „Neues Spiel erstellen“-Button. Danach zieht er aus der Liste, der registrierten Spieler, die Spieler, die nicht in einem Spiel beteiligt sind (hier: weiß) in seine Teilnehmer-Liste hinein. Der Server entscheidet ebenfalls welche registrierten Beobachter (hier: weiß) in die Beobachter-Liste kommen. Unter „Level ändern“ kann aus den gespeicherten Leveln eines ausgewählt werden. Wird „Starten“ ausgewählt, beginnt ein neues Spiel. Ein Spiel kann durch den „Verwerfen“-Button gelöscht werden, falls der Server doch kein neues Spiel anfangen möchte.

3.4 Fehlermeldungen

3.4.1 Vor dem Spiel

3.4.1.1 VerbindungsauftbauFehlgeschlagen

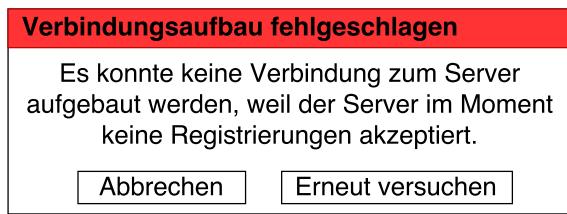


Abbildung 3.32: VerbindungsauftbauFehlgeschlagen-Dialog

Diese Fehlermeldung erscheint, wenn keine Verbindung zum Server aufgebaut werden konnte. Dies wird dem Benutzer (Beobachter oder Spieler) zusammen mit dem Grund (hier: „Server akzeptiert im Moment keine Registrierungen“) mitgeteilt. Ein weiterer Grund kann sein, dass kein Internetzugang vorhanden ist. Dann lautet die Fehlermeldung: „Es konnte keine Verbindung zum Server aufgebaut werden. Bitte überprüfen Sie Ihre Internetverbindung.“

3 Benutzerschnittstellen

3.4.2 Editor

3.4.2.1 LevelLoadError

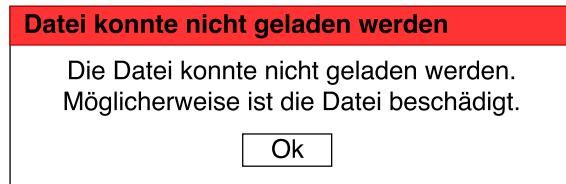


Abbildung 3.33: LevelLoadError-Dialog

Diese Fehlermeldung wird angezeigt, falls beim Laden einer Datei ein Fehler passiert. Dies wird dem Benutzer zusammen mit dem Grund mitgeteilt. Im obigen Beispiel ist der Grund: „Möglicherweise ist die Datei beschädigt.“ Ein anderer Grund könnte sein: „Möglicherweise ist die Datei zugriffsgeschützt.“.

3.4.2.2 CheckpointError



Abbildung 3.34: CheckpointError-Dialog

Dieser Dialog wird dem Editor-Benutzer angezeigt, falls er beim Spiel erstellen nur einen Checkpoint gesetzt hat und den Editor prüfen lässt, ob das Level von einem Spielserver als valide angenommen wird.

3.4.3 Weitere Fehlermeldungen

3.4.3.1 KeineSpielerMehr

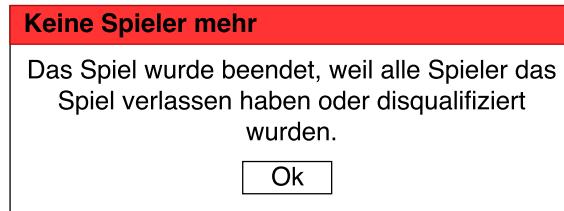


Abbildung 3.35: KeineSpielerMehr-Dialog

Diese Fehlermeldung wird allen Beobachtern und dem Server angezeigt, falls alle Spieler das Spiel verlassen haben oder disqualifiziert wurden.

3.4.3.2 VerbindungVerloren

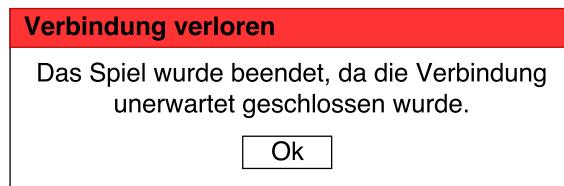


Abbildung 3.36: VerbindungVerloren-Dialog

Diese Fehlermeldung wird allen Beobachtern, Spielern und dem Server angezeigt, falls aus unbekannten Gründen die Verbindung zum Server bzw. zu den Teilnehmern unterbrochen oder sogar ganz verloren wurde.

3.5 Nutzungskonzept

Im Folgenden soll nochmals eine Zusammenfassung über die Bedienung des Systems gegeben werden. Im Wesentlichen handelt es sich um eine schriftliche Ausführung der Dialogstrukturdiagramme.

Startmenü Als Spieler oder Beobachter gelangt man nach Start des Spiels zunächst ins Startmenü (siehe 3.3.1.1). Möchte man an einem Spiel teilnehmen, drückt man „Server suchen“ (siehe 3.3.1.2). Möchte man das Spiel verlassen drückt man „Beenden“ (siehe 3.3.3.6). Mit einem Klick auf „Einstellungen“ (siehe 3.3.1.6) gelangt man zu den Spieleinstellungen. Mit einem Klick auf „Hilfe“ gelangt man in den Hilfbereich (siehe 3.3.1.8).

ServerSuchen Im Server-Suchen-Dialog (siehe 3.3.1.2) kann man sich aus einer Liste von Servern den gewünschten Server aussuchen. Ist ein gewünschter Server nicht in der Liste, kann man durch Anklicken von „Neuen Server hinzufügen“ (siehe 3.3.1.4) die Daten eines bestimmten Servers angeben. Sonst kann man einen Server auswählen und einen Namen eintragen, unter dem man dort erscheinen will, und sich via „Registrieren“-Button dort anmelden (siehe 3.3.1.3). Daraufhin wird ein Wartedialog angezeigt. Gelingt die Verbindung und nimmt der Server die Registrierung an, gelangt man in die Serverlobby (siehe 3.3.1.5). Ansonsten erhält man eine Fehlermeldung und kann die Registrierung erneut versuchen oder über „abbrechen“ zur Serversuche zurück. Aus der Serversuche zurück ins Startmenü gelangt man über einen Button „zurück“. Wie im Startmenü gelangt man aus der Serversuche heraus über entsprechende Buttons zu den Einstellungen (siehe 3.3.1.6) und zum Hilfebereich (siehe 3.3.1.8).

Serverlobby In der Serverlobby verweilt man, bis ein Spiel, dem man zugeteilt wurde, beginnt. Welches Spiel man zugeteilt wird, liegt in der Hand des Servers. Das Spiel startet dann automatisch nach Ablauf einer festgelegten Zeit. Falls die Verbindung aus irgendwelchen Gründen verloren geht, erhält man eine Fehlermeldung, von der aus man zur Serversuche zurückkehren kann. Ebenfalls gelangt man von der Serverlobby zurück zur Serversuche über den Button „verlassen“. Auch in der Serverlobby gelangt man über entsprechende Buttons zu den Einstellungen (siehe 3.3.1.6) und zum Hilfebereich (siehe 3.3.1.8).

Spieloberfläche Nach Beginn des Spiels befinden sich Spieler und Beobachter auf der Spieloberfläche (siehe 3.3.3.1). Klickt man hier auf den „Pause“-Button, gelangt man in den Pausenbereich-Dialog (siehe 3.3.2.2). Von dort erreicht man Hilfebereich (siehe 3.3.1.8) und Einstellungen (siehe 3.3.1.6) über entsprechende Buttons. Um das Spiel zu verlassen, muss der Spieler auf ‘Spiel beenden‘-Button klicken und es erscheint ein SpielVerlassen-Dialog (siehe 3.3.3.6). Mit „Spiel fortsetzen“ gelangt man aus dem Pausenbereich auf die Spieloberfläche zurück. Geht im Spiel die Verbindung verloren, erscheint wieder ein Fehlerdialog.

GameOver Bei Spielende gelangt man automatisch in den GameOver-Dialog (siehe: 3.3.3.3). Über „Ok“ kommt man von dort zurück in die Serverlobby.

4 Datenmodell

4.1 Sequenzdiagramme

4.1.1 Anmeldung Initiieren

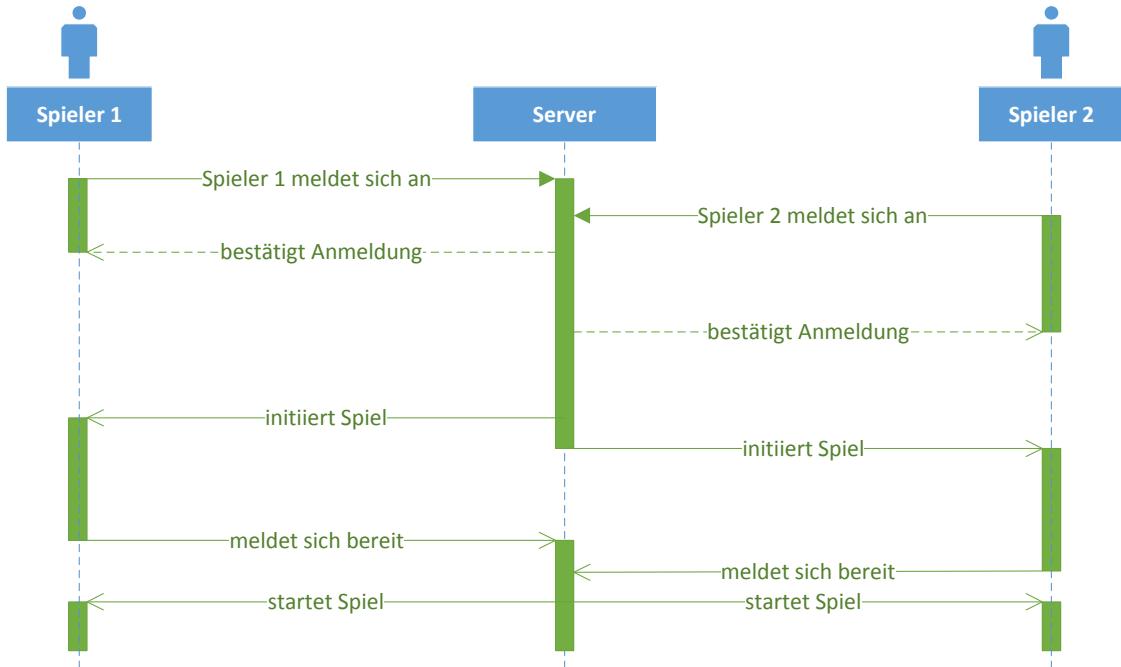


Abbildung 4.1: Sequenzdiagramm-Anmeldung Initiieren

Wenn sich ein Spieler mit dem Server verbündet, meldet sich der Spieler zuerst an und erhält dann eine Bestätigung von diesem. Das Spiel wird darauf hin vom Server initiiert, wobei der Spieler solange auf die Rückmeldung des Servers wartet. Nachdem sich alle Spieler (in diesem Fall 2) bereit gemeldet haben, wird das Spiel vom Server gestartet.

4.1.2 Beobachter

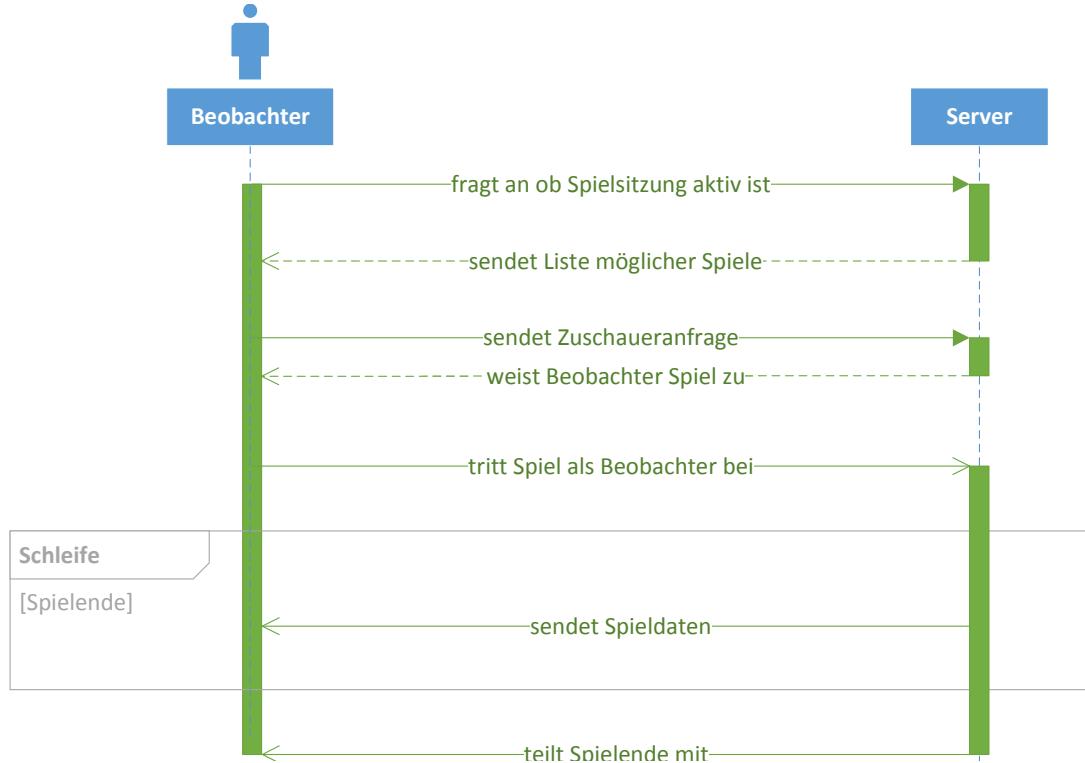


Abbildung 4.2: Sequenzdiagramm-Beobachter

Ein Beobachter fragt den Server an, ob eine Spielsitzung verfügbar ist. Daraufhin erhält er eine Liste mit allen Spielen, bei denen der Beobachter zuschauen kann. Der Beobachter wählt ein Spiel aus und schickt eine Zuschaueranfrage an den Server, worauf der Server mit der Zuteilung des Beobachters zur laufenden Spielsitzung reagiert. Nun tritt der Beobachter der Spielsitzung bei und erhält vom Server die erforderlichen Daten über das Spielgeschehen. Wenn das Spiel beendet ist teilt der Server dem Beobachter das Spielende mit.

4.1.3 KI

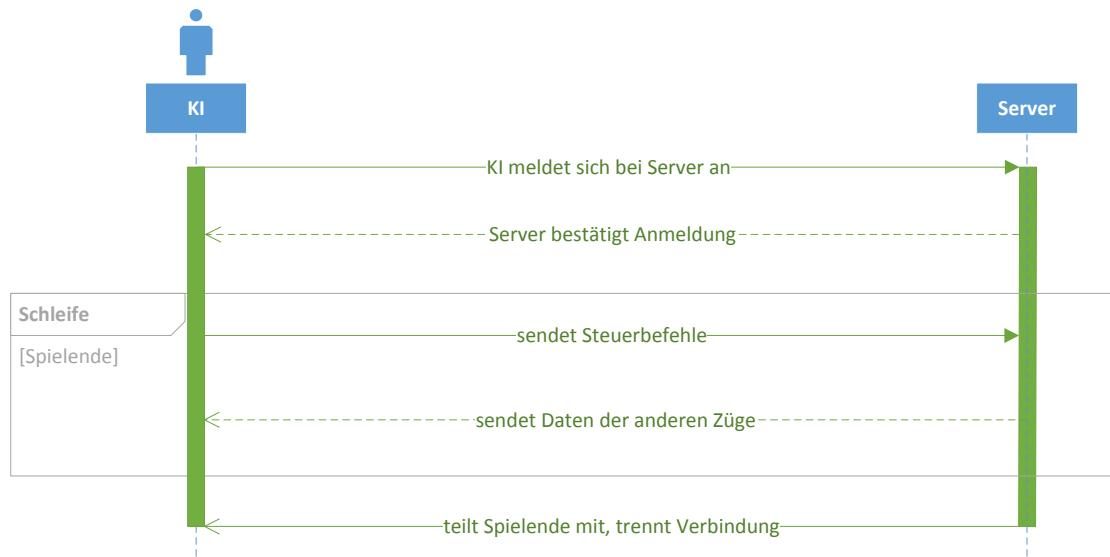


Abbildung 4.3: Sequenzdiagramm-KI

Die KI kann sich als eigenständiger Spieler bei einer Partie anmelden und wird vom Server als normaler Spieler angesehen. Alternativ kann ein Spieler ein oder mehr Züge von einer KI übernehmen lassen, allerdings werden dann die Züge weiterhin vom Spieler gesendet, wodurch der Server den Unterschied nicht kennt.

4.1.4 Spiel Beenden

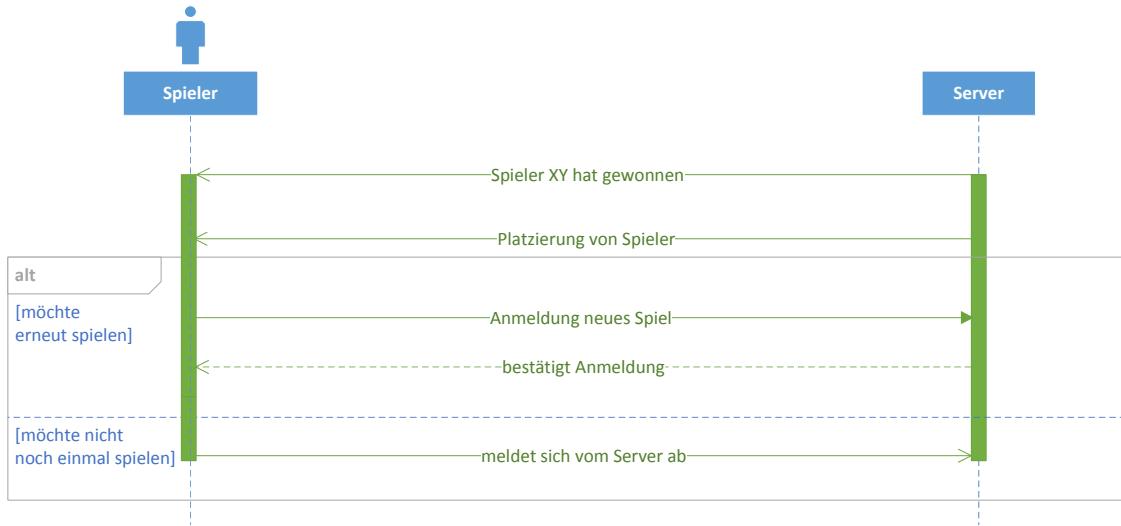


Abbildung 4.4: Sequenzdiagramm-Spiel Beenden

Nachdem das Spiel entschieden ist, teilt der Server jedem Spieler den Gewinner und zusätzlich seine eigene Platzierung mit. Daraufhin hat der Spieler die Möglichkeit sich für ein neues Spiel anzumelden oder die Verbindung abzubrechen.

4.1.5 Spieler

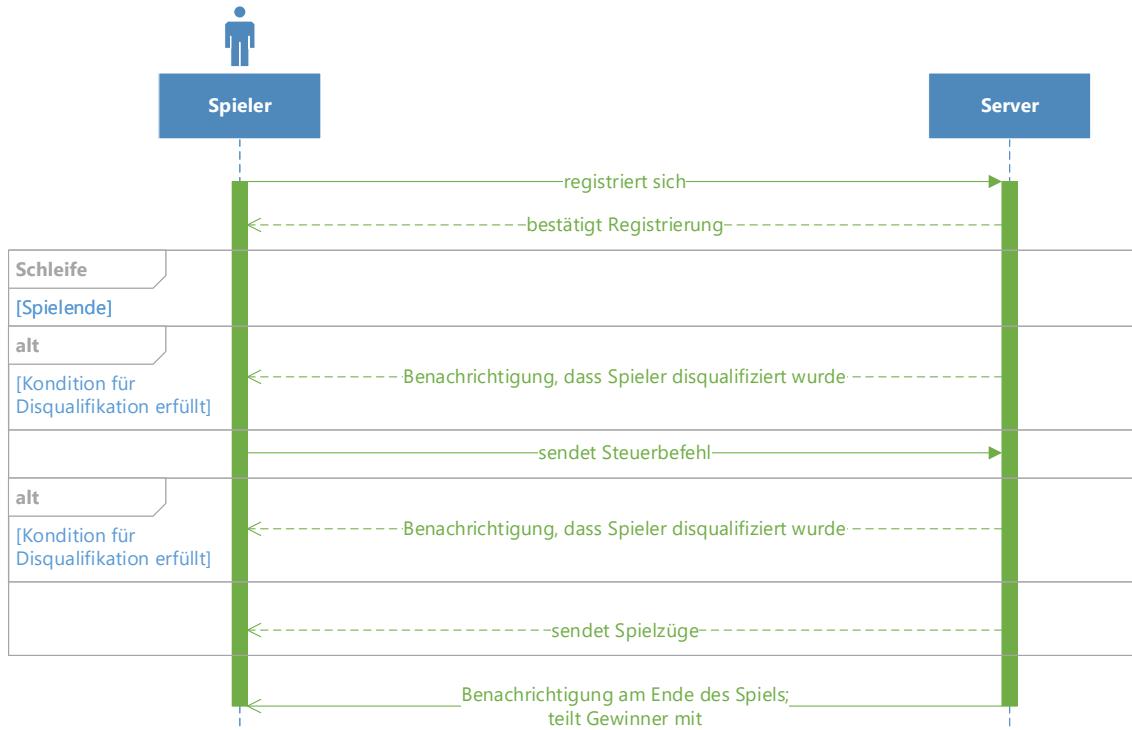


Abbildung 4.5: Sequenzdiagramm-Spieler

Der Spieler registriert sich basierend auf dem Anmeldung-Initiieren-Diagramm (siehe 4.1.1) bei dem Server. Im weiteren Verlauf tauschen Spieler und Server konstant Steuerbefehle bzw. Spielzüge aus. Falls der Spieler nicht vor Spielende durch eine Disqualifikation ausscheidet, bekommt er am Ende des Spiels die Spielinformationen und kann erneut spielen oder sich abmelden. (siehe 4.1.4)

4 Datenmodell

4.1.6 Disqualifikation

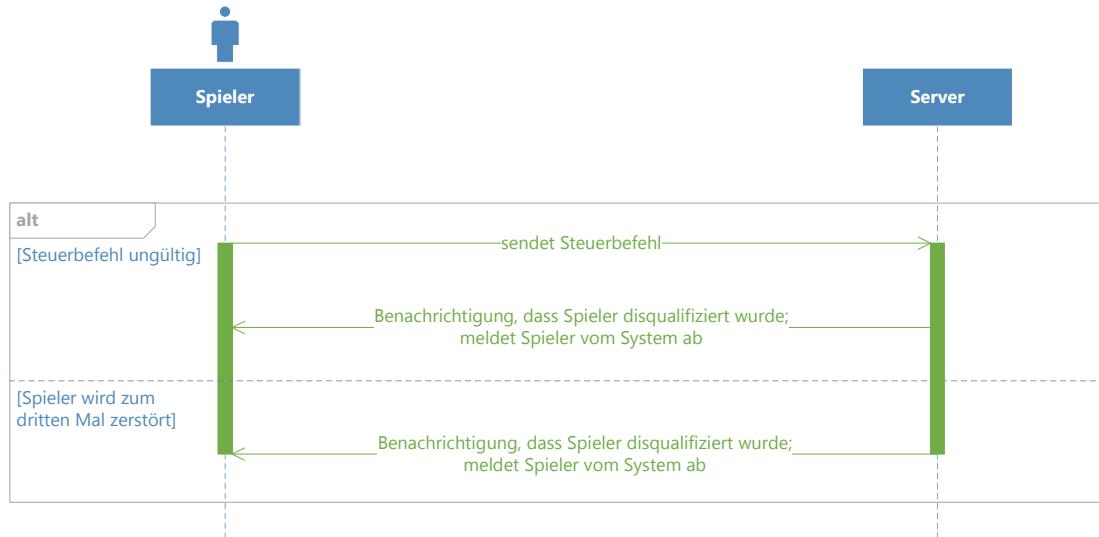


Abbildung 4.6: Sequenzdiagramm-Disqualifikation

Wenn ein ungültiger Steuerbefehl an den Server gesendet oder der Spieler dreimal zerstört wird, ist der Spieler disqualifiziert und wird direkt vom System abgemeldet.

4.1.7 Spiel spielen

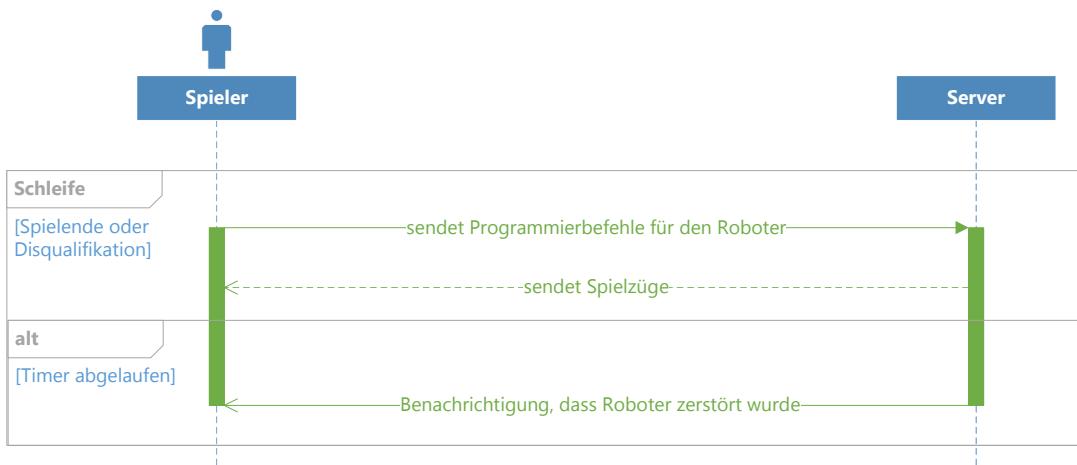


Abbildung 4.7: Sequenzdiagramm-Spiel spielen

Das Sequenzdiagramm bildet den Austausch von Daten zwischen dem Spieler und dem Server ab. Falls der Timer zur Programmierung des Roboters abläuft, wird der Roboter des Spielers zerstört. Geschieht dies zum dritten Mal oder möchte der Spieler ungültige Aktionen ausführen, wird er disqualifiziert (siehe 4.1.6).

4.1.8 Level erstellen

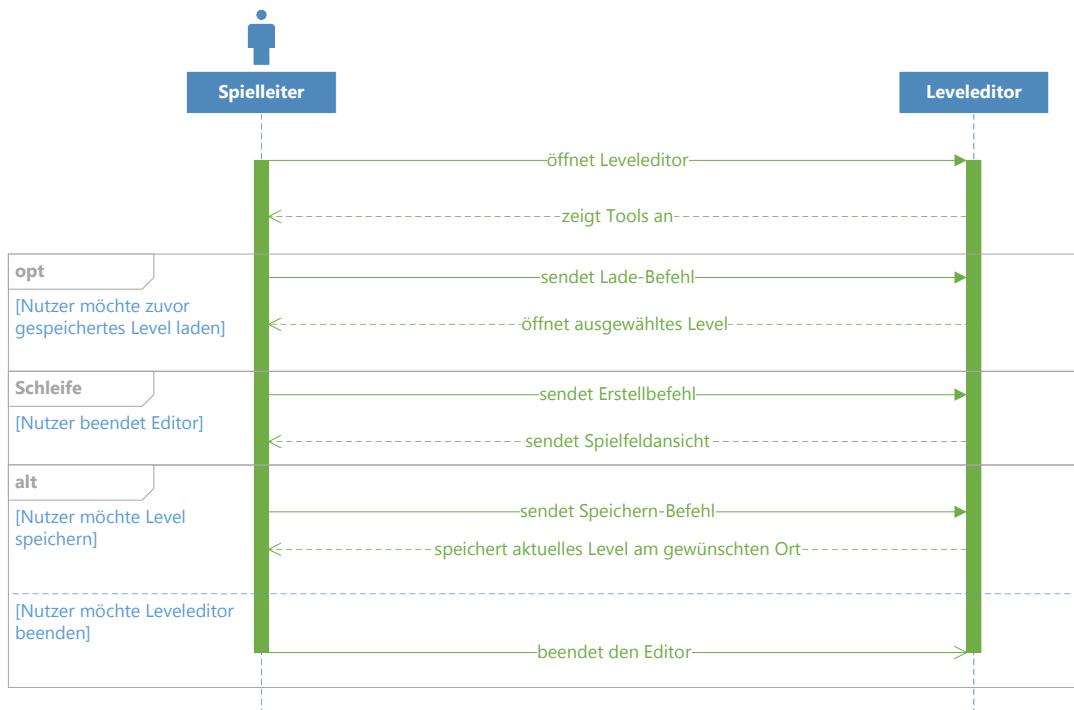


Abbildung 4.8: Sequenzdiagramm-Level erstellen

Um ein Level zu erstellen, öffnet der Spielleiter den Leveleditor. Dieser zeigt ihm die verschiedenen Tools an, mit denen er ein Level erstellen kann (siehe 3.3.4.1). Hier kann der Spielleiter ein zuvor erstelltes und gespeichertes Level öffnen und daran weiterarbeiten, oder ein neues Level erstellen. Dies erfolgt automatisch, falls keine Datei geladen wurde. Jede Änderung am Spielfeld (z.B. Checkpoint gesetzt), die der Spielleiter durchführt, wird vom Leveleditor angezeigt. Der Spielleiter kann solange daran arbeiten, bis er den Editor wieder beendet. Außerdem kann er das Level zu jeder Zeit speichern oder ein anderes Level laden.

4.1.9 Anforderungsabhängigkeiten

Beobachter: FA-B-009

KI: FA-TK-001, FA-TK-002, FA-TK-003

Mensch: FA-T-002, FA-T-003

Roboter: FA-S-001, FA-S-002, FA-S-003, FA-S-005, FA-S-006, FA-S-014,
FA-T-001

Server: FA-S-011, FA-S-017, FA-S-020, FA-S-021, FA-S-022, FA-TM-
001

Spielsitzung: FA-S-004

4.2 Zustandsdiagramme

4.2.1 Beobachter

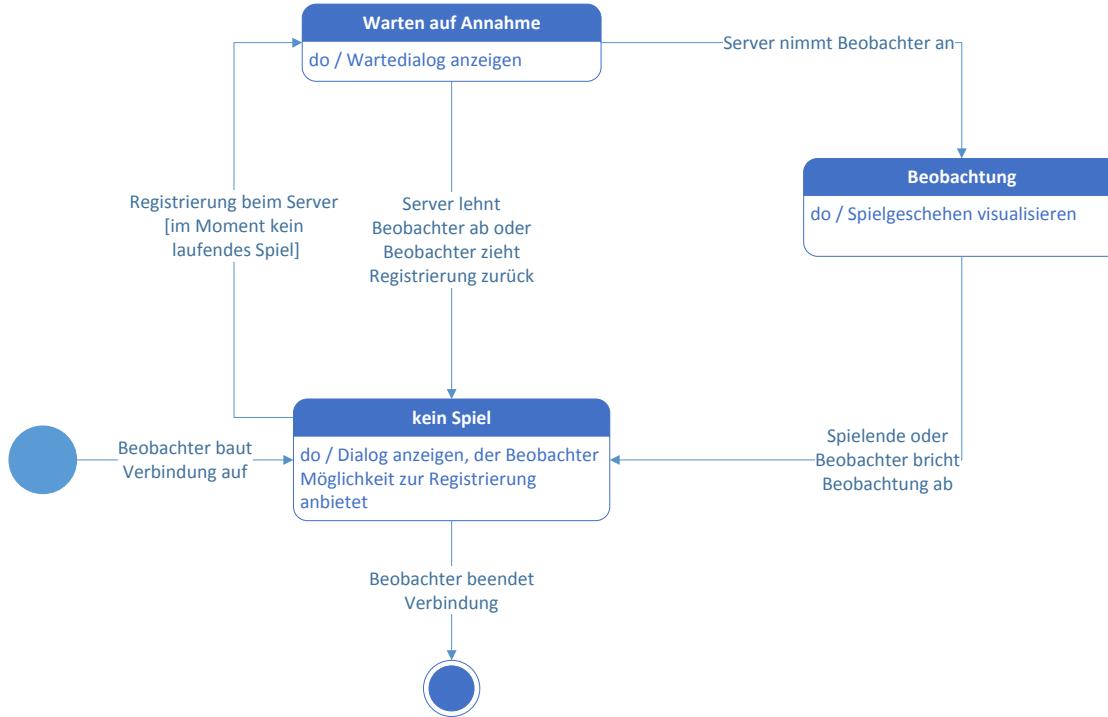


Abbildung 4.9: Zustandsdiagramme-Beobachter

Anmerkung: Der Beobachter kann die Beobachtung jederzeit verlassen (kommt zum Zustand „kein Spiel“) oder die Verbindung beenden (Endzustand). Im Regelfall baut der Beobachter eine Verbindung auf, beobachtet ein oder mehrere Spiele und beendet sie dann wieder.

Das Zustandsdiagramm Beobachter zeigt eine Registrierung beim Server. Der Server kann dem Beobachter die Visualisierung des Spielgeschehens anzeigen. Nachdem der Server den Beobachter abgelehnt hat, kann der Beobachter sich jedoch wieder beim Server registrieren. Falls der Beobachter das Spielfenster ganz schließt, kommt er von jedem Zustand aus direkt in den Endzustand.

4.2.2 KI

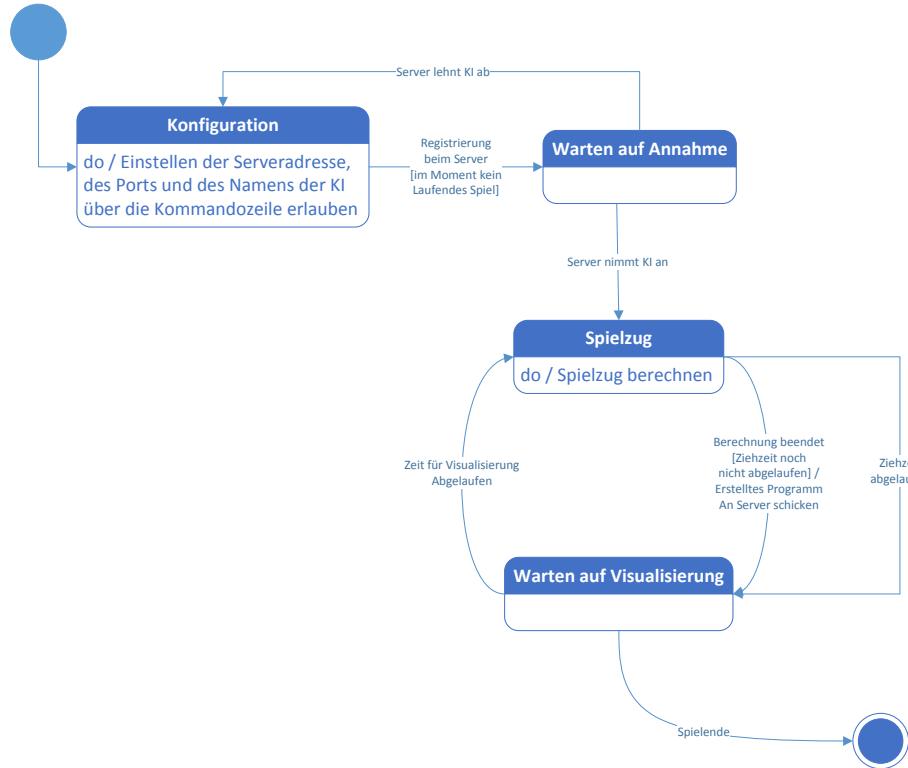


Abbildung 4.10: Zustandsdiagramme-KI

Anmerkung:

Zwischen Programmerstellung und Programmierung wird dem Spieler durch den Server der Verlauf der aktuellen Runde mitgeteilt.

Nach der Erstellung der KI kann sich dieser beim Server registrieren. Falls die KI ihn ablehnt, kann er wieder eine Registrierung versenden. Hat die KI ein Programm an den Server geschickt oder ist die Zeit abgelaufen wartet er auf eine Visualisierung. Nach dem Spielende, wird auch die KI beendet.

4.2.3 Mensch

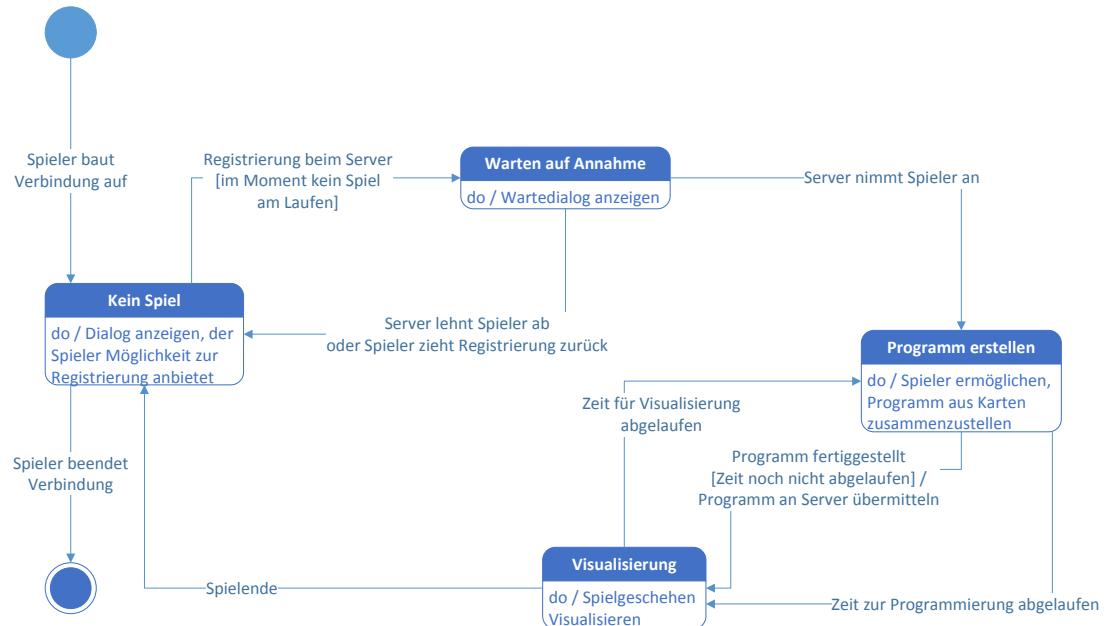


Abbildung 4.11: Zustandsdiagramme-Mensch

Anmerkung:

Zwischen Programmerstellung und Programmierung wird dem Spieler durch den Server der Verlauf der aktuellen Runde mitgeteilt.

Der Mensch-Teilnehmer kann eine Verbindung aufbauen und diese sofort wieder beenden. In jedem Zustand kann der Mensch-Teilnehmer das Spielfenster schließen und so in den Endzustand kommen. Nachdem der Spieler vom Server abgelehnt wurde oder der Spieler seine Registrierung zurückgezogen hat, kann er sich jederzeit wieder beim Server registrieren. Wenn der Mensch-Teilnehmer ein Programm fertiggestellt hat oder die Zeit abgelaufen ist, wird das Spielgeschehen visualisiert.

4.2.4 Roboter

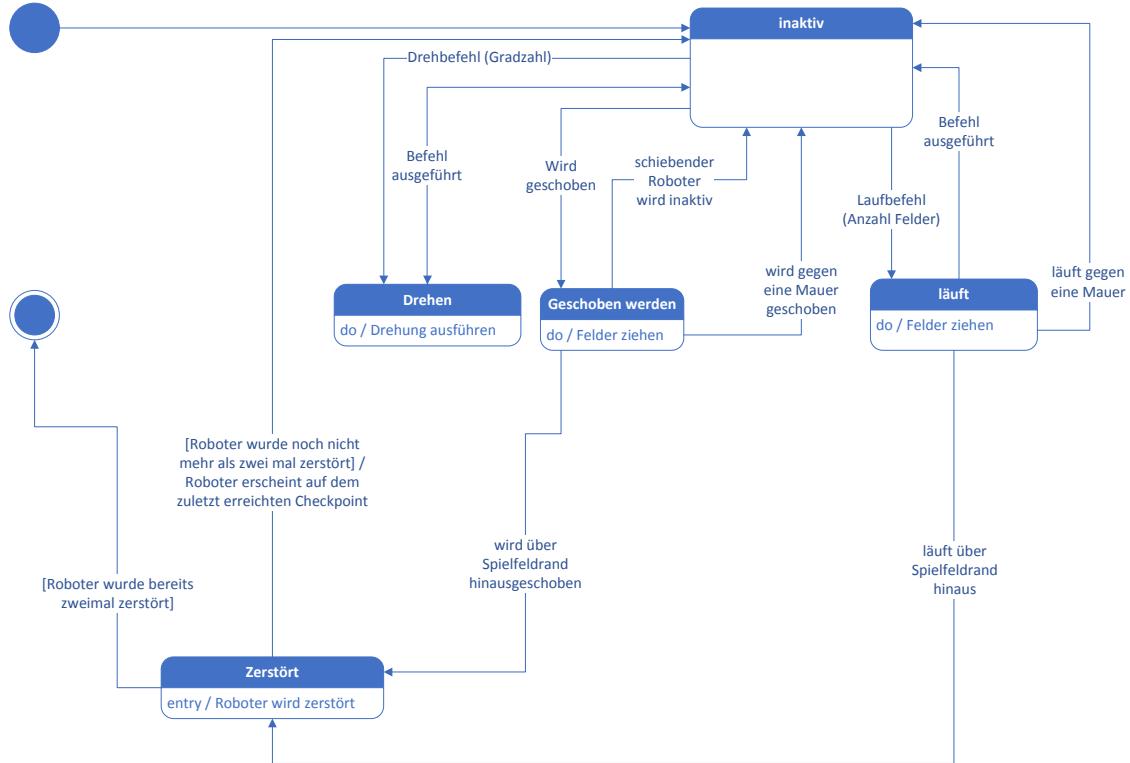


Abbildung 4.12: Zustandsdiagramme-Roboter

Der Roboter benötigt Befehle, um in den Zustand „drehen“, „geschoben werden“ oder „läuft“ zu kommen. Falls der Roboter aus dem Spielfeld hinausgeschoben wird, wird er zerstört und gelangt an seinem letzten Checkpoint im Zustand „inaktiv“. Nach dreimaliger Zerstörung, kommt der Roboter in den Endzustand.

4.2.5 Server

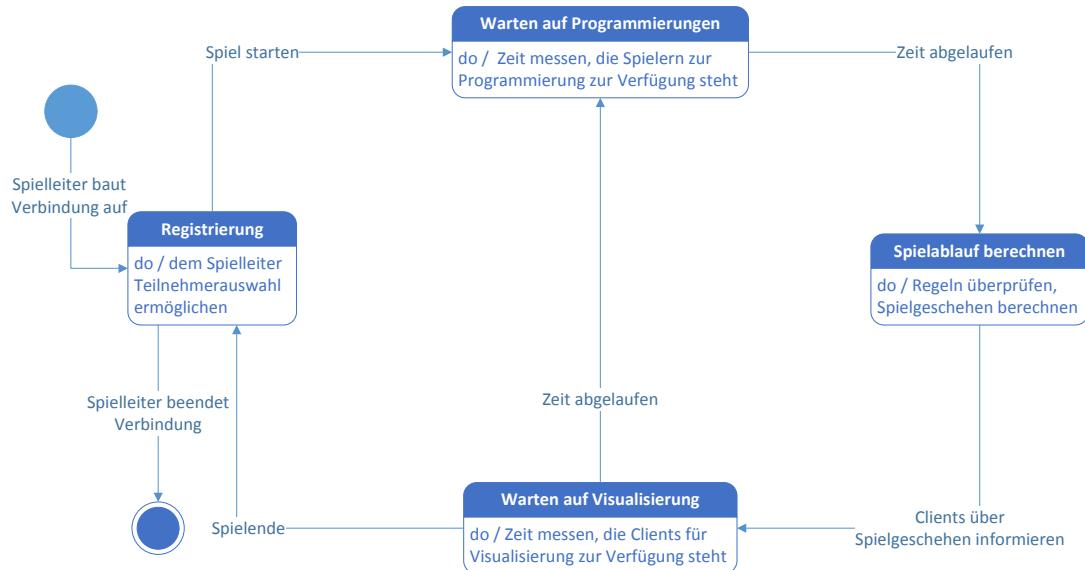


Abbildung 4.13: Zustandsdiagramme-Server

Das Zustandsdiagramm-Server beschreibt die Phasen in denen sich der Server im Programmverlauf befindet. Bis auf Spiel-starten und Spielende sind alle Übergänge Zeit basierend. Im Zustand Registrieren ist es für den Spielleiter möglich eine Verbindung zum Server auf-/abzubauen. Dieser muss dann bis zum Spielende im Spiel bleiben.

4.2.6 Spielsitzung

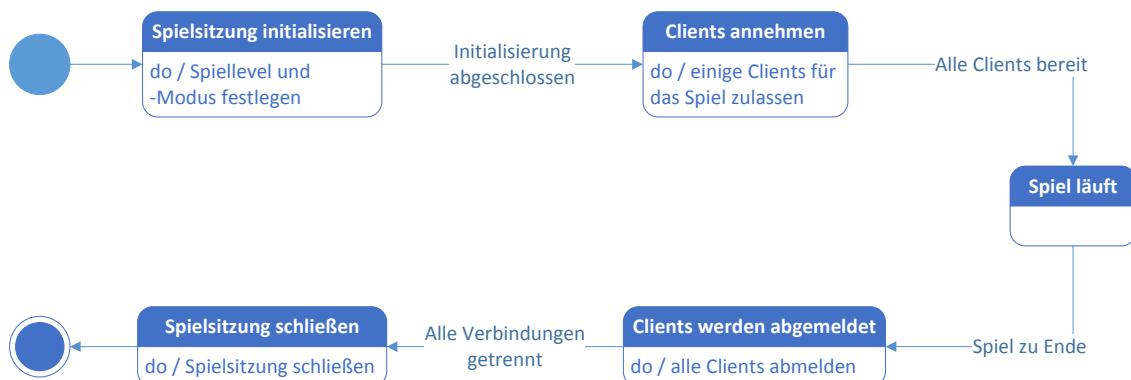


Abbildung 4.14: Zustandsdiagramme-Spielsitzung

Nachdem die Spielsitzung initialisiert wurde, werden ausgewählte Clients in dem Spiel zugelassen. Clients können sich während dem Spiel abmelden, indem sie das

4 Datenmodell

Spiel verlassen oder das Spielfenster schließen. Ansonsten werden die Clients nach dem Spiel abgemeldet und die Sitzung geschlossen.

4.2.7 Anforderungsabhängigkeiten

AnmeldungInitiiieren: FA-S-004

Beobachter: FA-S-004, FA-B-006, FA-B-009

KI: FA-S-004, FA-B-006, FA-B-009

SpielBeenden: FA-S-004, FA-TM-003, FA-TM-004

Teilnehmer: FA-S-004, FA-S-008, FA-B-006, FA-B-009

4.2.8 Bemerkungen

Bemerkungen zum Pause-Zustand:

Natürlich haben alle Clients genau wie der Server einen Pausezustand. Da dieser aber allein vom Server initiiert wird und die Clients darauf keinerlei Einfluss haben, ist es wenig sinnvoll, ihn in den Zustandsdiagrammen der Clients aufzunehmen.

5 Randbedingungen

5.1 Qualitätsanforderungen

ID	QA001
TITEL	Benutzerfreundlichkeit
BESCHR.	Es sollen maximal 10% der Bildschirmfläche von Text bedekt sein.
BEGRÜ.	Um dem Benutzer nicht die Sicht auf das Spielfeld zu versperren.
AKTEUR	Spieler: Mensch-Teilnehmer, Beobachter
PRIORI.	+
BEMERK.	Gilt für Beobachter- und Mensch-Teilnehmer-Clients.

ID	QA002
TITEL	Plattform
BESCHR.	Das Spiel sollt auf Windows 10 oder Linux ausführbar sein.
BEGRÜ.	Vom Kunden gefordert.
AKTEUR	Spieler, Beobachter, Server
PRIORI.	++

ID	QA003
TITEL	Unitest
BESCHR.	Durch Unitests soll eine Codeabdeckung von 80% erreicht werden.
BEGRÜ.	Vom Kunden gefordert.
AKTEUR	-
PRIORI.	++

5 Randbedingungen

ID	QA004
TITEL	Robustheit
BESCHR.	Von 100 Spielen soll maximal ein Spiel aufgrund eines Fehlers abgebrochen werden müssen.
BEGRÜ.	Um den Spielspaß nicht durch häufige Fehler zu beeinträchtigen.
AKTEUR	-
PRIORI.	o

ID	QA005
TITEL	Dokumentation
BESCHR.	Alle wichtigen Klassen und Methoden müssen mit JavaDoc dokumentiert werden. Welche genau, ist im Google Styleguide (siehe https://google.github.io/styleguide/javaguide.html#s7.3-javadoc-where-required) festgelegt. Insbesondere Abschnitt 7.3, 7.3.1 und 7.3.2 sind in diesem Styleguide wichtig.
BEGRÜ.	Um das Verstehen des Programmcodes zu erleichtern.
AKTEUR	-
PRIORI.	++

ID	QA006
TITEL	Netzwerkverbindung
BESCHR.	Die Client-Server-Anbindung soll sowohl im LAN als auch via Internet funktionieren.
BEGRÜ.	Vom Kunde gefordert.
AKTEUR	Spieler, Beobachter, Server
PRIORI.	++

5.1 Qualitätsanforderungen

ID	QA007
TITEL	Spielinstanzen
BESCHR.	Der Server muss mindestens drei Spiele gleichzeitig hosten können.
BEGRÜ.	Damit mehr als acht Spieler gleichzeitig spielen können.
AKTEUR	Server
PRIORI.	--

ID	QA008
TITEL	Styleguide
BESCHR.	Die gesamte Implementierung muss die Google Styleguide Richtlinien erfüllen.
BEGRÜ.	Damit der Programmcode leserlich und verständlich bleibt.
AKTEUR	Server
PRIORI.	+ +

5.2 Abnahmekriterien

Für eine erfolgreiche Abnahme müssen alle funktionalen und nichtfunktionalen Anforderungen erfüllt werden. Die Meilensteine müssen in einem bestimmten Zeitrahmen abgeschlossen sein und von einem Tutor anerkannt werden. Wenn das gesamte Softwaresystem nach Untersuchungen auf Testfällen als fehlerfrei empfunden wird und der Kunde mit dem Produkt zufrieden ist, wird dies als ein erfolgreiches Endergebnis angesehen.

Das System sollte den dokumentierten Anforderungen entsprechen. Stimmt die Implementierung nicht mit den Spezifikationen überein, wird versucht, rechtzeitig eine Lösung zu finden.

Anhang

Glossar

BEGRIFF Abbruch

Siehe Spielabbruch

BEGRIFF Ausführung eines Programms

BESCHR. Nachdem alle Spieler ihr Programm zusammengestellt haben, geht es an die Ausführung. Die Befehle werden nacheinander ausgeführt, dabei wird erst von allen Spielern der erste Befehl ausgeführt, danach von allen Spielern der zweite usw. Wessen Befehl dabei jeweils zuerst ausgeführt wird, wird durch die Prioritäten bestimmt, die auf den Befehlskarten angegeben sind. Während die Befehle ausgeführt werden, können die Spieler nicht mehr auf die Bewegungsvorgänge der Roboter Einfluss nehmen.

SYNONY. Programmausführung

BEGRIFF Ausrichtung

BESCHR. Der Roboter hat eine Ausrichtung, die bestimmt, wo bei ihm vorne und hinten ist, also seine aktuelle Bewegungsrichtung festlegt. Damit die Ausrichtung immer klar ist, legt der Spieler beim Einsetzen seines Roboters ins Spiel die Ausrichtung mit dem ersten Bewegungsbefehl fest.

IstEin Nord, Ost, Süd, West

KannSein Eigenschaft des Roboters

ASPEKT Bewegung des Roboters

BEGRIFF Befehl

Siehe Programmbebefhl

Anhang

BEGRIFF Blockierung

BESCHR. Entsteht, wenn ein Roboter in eine Richtung bewegt werden soll, in der vom Roboter aus gesehen eine Mauer steht. Dies kann geschehen, wenn der Roboter selbst einen dahingehenden Laufbefehl erhält, oder aber wenn er durch einen anderen Roboter geschoben wird. Im Fall einer Blockierung wird die Bewegung, durch die sie verursacht ist, einfach abgebrochen und das Spiel wird fortgesetzt.

ASPEKT Bewegung des Roboters

BEISPIEL *Roboter A soll drei Felder vorwärts laufen, stößt aber nach einem Feld gegen Roboter B. Roboter A schiebt Roboter B ein Feld weiter, wobei Roboter B gegen eine Wand prallt. Jetzt sind beide Roboter blockiert und Roboter A darf sein verbliebenes Feld nicht weiterziehen.*

BEGRIFF Checkpoint

BESCHR. Bestimmte Position auf dem Spielfeld. Es befinden sich mehrere (mindestens zwei) Checkpoints auf einem Spielfeld, die eine vorgebene Reihenfolge besitzen. Alle Spieler starten am ersten Checkpoint. Dabei muss jeder Spieler den Roboter mit dem ersten Befehl vom Checkpoint wegbewegen. Aufgabe der Spieler ist es, die Checkpoints mit ihren Robotern in der richtigen Reihenfolge zu erreichen. Wird ein Roboter zerstört, wird er zudem an einem Checkpoint wieder eingesetzt.

IstEin Position

KannSein Zwischenziel, Ziel

ASPEKT Spielziel

BEGRIFF Client

BESCHR. Ein Client ist ein Computersystem, das entweder von einem Menschen bedient oder durch eine KI gesteuert wird.

IstEin Computer

KannSein Beobachter, Teilnehmer

ASPEKT System

BEGRIFF Disqualifikation

BESCHR. Ein Spieler wird disqualifiziert, sobald sein Roboter zum dritten Mal zerstört wird oder eine seiner Aktionen ungültig ist. Wurden alle Spieler disqualifiziert, wird das Spiel beendet.

ASPEKT Spielregeln

BEISPIEL *Spieler A braucht mehr Zeit als ihm zur Verfügung steht, um seinen Roboter zu programmieren, weshalb er aus dem Spiel ausscheidet.*

BEGRIFF Drehbefehl

BESCHR. Befehl, mit dem die Spieler den Roboter anweisen können, sich um die eigene Achse zu drehen.

IstEin Befehl

KannSein Vierteldrehung, halbe Drehung

ASPEKT Bewegung des Roboters

BEISPIEL „Vierteldrehung gegen den Uhrzeigersinn“

BEGRIFF Engine

Siehe Server

BEGRIFF Feld

Siehe Position

BEGRIFF Gewinner

BESCHR. Gewinner ist der Spieler, dessen Roboter alle Checkpoints in der richtigen Reihenfolge passiert und als erstes das Ziel erreicht. Er wird nach Ende des Spiels für alle Teilnehmer und Beobachter sichtbar angezeigt.

IstEin Spieler

KannSein KI, Mesch

ASPEKT Spielregeln

BEGRIFF Highscore

BESCHR. Im Zusammenhang des SOPRAs ist der Highscore eine globale Teambewertung. Unter anderem fließen Bewertung von Meilensteinen und Aufgabenblättern sowie die Anzahl der vertriebenen Lizenzen auf der Messe und erreichte Punkte auf dem Turnier mit ein. Der Highscore ist öffentlich.

IstEin Teambewertung

BEGRIFF Hindernis

Siehe Mauer

BEGRIFF Kachel

Siehe Position

Anhang

BEGRIFF Karte
Siehe Programmbebefhl

BEGRIFF Konfiguration
Siehe Spielkonfiguration

BEGRIFF Künstliche Intelligenz (KI)
BESCHR. Eine KI simuliert einen Spieler und kann so am Spiel als virtueller Spieler teilnehmen.
IstEin Teilnehmer

BEGRIFF Laufbefehl
BESCHR. Befehl, mit dem die Spieler den Roboter anweisen können, eine bestimmte Anzahl von Feldern zu ziehen.
IstEin Befehl
KannSein Vorwärtsbewegung, Rückwärtsbewegung
ASPEKT Bewegung des Roboters
BEISPIEL „1 Feld rückwärts“

BEGRIFF Level
BESCHR. Ein Spielfeld mit einer bestimmten Größe und bestimmten Anzahl und Positionen von Mauern und Checkpoints. Verschiedene Levels können unterschiedlich schwierig sein. Levels können mit dem Leveleditor erstellt werden.
ASPEKT Spielregeln

BEGRIFF Leveleditor
BESCHR. Tool, um Level zu erstellen und zu konfigurieren.

BEGRIFF Mauer
BESCHR. Eine Mauer ist ein Hindernis, das auf dem Spielfeld platziert ist. Eine Mauer steht dabei auf der Grenze zwischen zwei Feldern. Roboter können eine Mauer nicht passieren. Steht der Bewegung eines Roboters eine Mauer im Weg, so wird der Roboter blockiert und bleibt stehen.
ASPEKT Bewegung des Roboters
SYNONY. Hindernis, Wand

BEGRIFF Messe
BESCHR. Treffen aller Teams im Rahmen des SOPRAs, an dem die Teams Lizenzen von Spielekomponenten voneinander erwerben können.

BEGRIFF Plattform
BESCHR. Betriebssystem, auf dem das Spiel ausgeführt wird.
KannSein Windows 10, Linux
ASPEKT System
SYNONY. Betriebssystem

BEGRIFF Position
BESCHR. Kästchen auf dem Spielfeld; durch zwei Koordinaten eindeutig bestimmt.
BEISPIEL A1
SYNONY. Feld, Kachel

BEGRIFF Priorität
BESCHR. Auf jeder Befehlskarte wird eine Priorität angegeben, die darüber entscheidet, in welcher Reihenfolge die Befehle der Spieler ausgeführt werden. Die Prioritäten sind so gesetzt, dass unter verschiedenen Befehlen insgesamt eine feste Rangfolge herrscht, und die Rangfolge unter Befehlen mit demselben Wortlaut zufällig ist.
IstEin zufällige natürliche Zahl
ASPEKT Spielregeln
BEISPIEL *Spieler1 und Spieler2 spielen gegeneinander. Spieler1 hat ein Programm aus Befehlen mit den Prioritäten (1), (5), (7), (2), (15) in dieser Reihenfolge zusammengestellt, Spieler2 eins mit den Prioritäten (12), (3), (4), (9), (16). Die Befehle werden abwechselnd abgearbeitet, aber immer so, dass der erste Befehl beider Spieler abgearbeitet wurde, bevor der zweite Befehl eines Spielers an die Reihe kommt. Also hier wird erst der Befehl mit der Priorität (12) von Spieler1 ausgeführt, dann der Befehl von Spieler2, da $12 > 1$. Dann kommen nach dem selben Prinzip nacheinander die Befehle mit den Prioritäten (5), (3), (7), (4), (9), (2), (16), (15).*

BEGRIFF Programm
BESCHR. Ein Programm ist im Zusammenhang des Spiels eine Abfolge von simplen Befehlen an einen Roboter, um ihn über das Spielfeld zu bewegen; wird von einem Spieler zusammengestellt.
IstEin Abfolge von 5 Befehlen
ASPEKT Programmierung des Roboters
BEISPIEL „1 Feld vorwärts - Vierteldrehung gegen den Uhrzeigersinn - 3 Felder vorwärts - 2 Felder vorwärts - halbe Drehung (180°)“
SYNONY. Programmierung

Anhang

BEGRIFF Programmausführung
Siehe Ausführung

BEGRIFF Programmbebefhl
BESCHR. Spielkarte mit einfacher Bewegungsanweisung für einen Roboter.
KannSein Drehbefehl, Laufbefehl
ASPEKT Programmierung des Roboters
BEISPIEL „ein Feld vorwärts“
SYNONY. Befehl, Karte

BEGRIFF Programmierung
Siehe Programm

BEGRIFF Robo Rally
BESCHR. Brettspiel, nach dessen Vorlage das Computerspiel entwickelt wird und somit auch Name des Computerspiels.

BEGRIFF Roboter
BESCHR. Virtuelle Spielfigur, die von einem Spieler durch einfache Befehle programmiert werden kann.
ASPEKT Spielfigur

BEGRIFF Runde
BESCHR. Ein Spiel besteht aus mehreren Runden. Jede Runde hat folgenden Ablauf:
1. Spieler erhalten Befehlskarten
2. Spieler stellen Programm für ihre Roboter zusammen
3. Programme werden ausgeführt
ASPEKT Spielregeln
SYNONY. Rundenablauf

BEGRIFF Rundenablauf
Siehe Runde

BEGRIFF Server

BESCHR. Der Server ist im Zusammenhang des Spiels ein Computersystem, das die Leitung über ein Spiel trägt. Er muss das Spiel verwalten, also zum Beispiel Clients ermöglichen, sich für ein Spiel anzumelden, und den Spielverlauf moderieren, also die Clients auffordern, ihre Spielzüge zu machen und auf Gültigkeit überprüfen. Der Server ist somit für die Einhaltung der Spielregeln zuständig.

IstEin Computer

ASPEKT System

SYNONY. Spieleengine, Engine

BEGRIFF Spielabbruch

BESCHR. Liegt dann vor, wenn ein Spiel manuell beendet wird.

IstEin Spielende

ASPEKT Spiel

SYNONY. Abbruch

BEGRIFF Spielbrett

BESCHR. Zweidimensionales Raster, worauf Mauern und Checkpoints positioniert sind und die Roboter bewegt werden.

ASPEKT Spielregeln

SYNONY. Spielfeld

BEGRIFF Spieleengine

Siehe Server

BEGRIFF Spielende

BESCHR. Regulär ist das Spiel beendet, sobald ein Roboter das Ziel erreicht und zuvor alle Zwischenziele in der richtigen Reihenfolge besucht hat. Ein Spiel kann aber auch manuell abgebrochen werden. Auch wenn alle Spieler disqualifiziert wurden, wird das Spiel beendet.

KannSein jemand gewinnt, Abbruch, Disqualifikation aller Spieler

ASPEKT Spielregeln

BEGRIFF Spieler

Siehe Teilnehmer

BEGRIFF Spielername

BESCHR. Jeder Spieler besitzt einen Namen, anhand dessen man ihn eindeutig identifizieren kann.

KannSein Zeichenkette aus Buchstaben und Zahlen

Anhang

BEGRIFF Spielfeld
Siehe Spielbrett

BEGRIFF Spielkonfiguration
BESCHR. Einstellung für ein Spiel
KannSein Spieleranzahl, Größe des Spielfelds, Anzahl Positionen von Mauern, Anzahl, Reihenfolge und Position von Checkpoints, Zeitlimit für Roboterprogrammierung pro Runde.
ASPEKT System
BEISPIEL *Ein Spiel hat folgende Konfogurationen: 3 Spieler auf einem 25x25-Spielfeld, 3 Mauern auf den Grenzen der Felder B5-B6, J10-K10, W3-W4, 2 Checkpoints, wobei der erste auf A1 steht und der letzte auf Y25, Zeitlimit für eine Programmierung sind 16 Sekunden.*
SYNONY. Einstellung, Konfiguration

BEGRIFF Spielregeln
BESCHR. Vorschriften, wie sich die Spieler verhalten müssen; Definition aller Spielemente wie Spielablauf, Spielende usw. Die Einhaltung der Spielregeln wird vom Server überwacht.
ASPEKT Spielregeln

BEGRIFF Teilnehmer
BESCHR. Teilnehmer sind Menschen oder KIs, die mit dem Spiel interagieren und die darin befindlichen Roboter programmieren können.
IstEin Client
KannSein Mensch, KI
ASPEKT Spiel
SYNONY. Spieler

BEGRIFF Turnier
BESCHR. Als Turnier wird der Wettbewerb bezeichnet, der 2017 im Rahmen des SOPRAs am Ende des Projekts veranstaltet werden soll. Das Turnier dient dazu, die entwickelten Programme der einzelnen Teams spielerisch zu testen.
IstEin Wettbewerb

BEGRIFF unentschieden

BESCHR. Ein Spiel geht unentschieden aus, falls alle Spieler vor Spielende disqualifiziert wurden oder bei einem manuellen Abbruch.

IstEin Spielausgang

ASPEKT Spielregeln

BEGRIFF Verlierer

BESCHR. Spieler, die zum Spielende noch nicht alle Checkpoints einschließlich dem Ziel erreicht haben. Verlierer werden nach Spielende für alle Teilnehmer und Beobachter sichtbar angezeigt.

IstEin Spieler

KannSein Mensch, KI

ASPEKT Spielregeln

BEGRIFF Wand

Siehe Mauer

BEGRIFF Zerstörung

BESCHR. Es gibt zwei Möglichkeiten für einen Roboter, zerstört zu werden: Entweder, er fällt während der Befehlsausführungsphase vom Spielfeld, oder, er befindet sich dann auf einem Checkpoint, wenn ein anderer Roboter auf ebendiesem eingesetzt wird. Wird ein Roboter zerstört, ist er für die restliche Runde nicht mehr im Spiel und wird zu Beginn der nächsten Runde an jenem Checkpoint eingesetzt, den er zuletzt erreicht hat. Wird der Roboter eines Spielers zum dritten Mal zerstört, wird der Spieler disqualifiziert.

KannSein Roboter fällt von Spielfeld, Roboter sitzt auf Checkpoint, an dem ein anderer Roboter eingesetzt wird.

ASPEKT Spielregeln

BEISPIEL *Roboter A steht bei Checkpoint 2, der sich ein Feld vom Spielfeldrand entfernt befindet und erhält den Befehl, zwei Felder vorwärts (Richtung Rand) zu ziehen. Er fällt vom Spielfeld und wird zerstört. Danach zieht Roboter B auf Checkpoint 2. Die nächste Runde beginnt, und Roboter A wird an der Stelle von Checkpoint 2 wieder auf das Spielfeld gesetzt. Damit wird Roboter 2 zerstört.*

Anhang

BEGRIFF Ziel

BESCHR. Letzter Checkpoint auf einem Spielfeld. Wer das Ziel mit seinem Roboter als erstes erreicht und zuvor alle anderen Checkpoints (Zwischenziele) in vorgegebener Reihenfolge besucht hat, gewinnt das Spiel.

IstEin Checkpoint

ASPEKT Spielziel

SYNONY. Letzter Checkpoint

BEGRIFF Zwischenziel

BESCHR. Checkpoint, der nicht das Ziel ist. Also sind alle Checkpoints außer dem letzten, Zwischenziele.

IstEin Checkpoint

ASPEKT Spielziel