



UNIVERSITÄT PADERBORN

Die Universität der Informationsgesellschaft

Heinz Nixdorf Institut
Fachgruppe Softwaretechnik
Zukunftsmeile 1
33102 Paderborn

Pflichtenheft

im Rahmen des Softwaretechnikpraktikums 2017

Team 10

s m a r t e n
professional software development

Betreuer: Christin Löer

Paderborn, den 22. Mai 2017

Autoren:

David Bock	Niklas Doppelstein
Max Krömker	Anke Küstner
Björn Luchterhandt	Sebastian Pranger
Jost Rossel	Wolfgang Schaperdot
René Scherf	Hanna Siek
Moritz Thiele	Robin Wulfes

Pflichtenheft

Inhaltsverzeichnis

1 Zielbestimmung	1
2 Produkteinsatz	2
2.1 Beschreibung des Problembereichs	2
2.2 Glossar	2
2.3 Modell des Problembereichs	5
2.4 Geschäftsprozesse	7
2.4.1 Partie ausrichten	7
2.4.2 Spielzug durchführen	8
2.4.3 Spielkonfiguration erstellen	9
3 Produktfunktionen	10
3.1 Use Cases	10
3.1.1 Registrieren	13
3.1.2 Wegfelder organisieren	15
3.1.3 Spielzug durchführen	16
3.1.4 Partie spielen	18
3.1.5 Partie ausrichten	19
3.1.6 Turnier ausrichten	21
3.1.7 Spielkonfiguration bearbeiten	23
4 Produktcharakteristiken	25
4.1 Systemanforderungen	25
4.1.1 Hardwareumgebung - Smartphone	25
4.1.2 Softwareumgebung - Smartphone	25
4.1.3 Hardwareumgebung - PC	25
4.1.4 Softwareumgebung - PC	26
4.2 Nicht funktionale Anforderungen	26
A GUI Skizzen	28

1 Zielbestimmung

Mobile Games werden immer beliebter. Ein Großteil der Bevölkerung besitzt ein Smartphone und nutzt dieses auch zur Unterhaltung. Dabei wachsen mit der steigenden Popularität auch die Anforderungen an Anwendungen aus dem Entertainment-Bereich. Mobile Spielanwendungen müssen ästhetisch ansprechend, stabil und sicher sein und außerdem eine hohe Benutzerfreundlichkeit, übersichtliche Gestaltung und leichte Bedienbarkeit aufweisen. Unter Beachtung dieser Qualitätsstandards wird eine netzwerkfähige Softwareversion des beliebten Brettspiels „Tsuru“ für Android-Smartphones und den Computer umgesetzt.

Die zu entwickelnde Anwendung besteht aus einem Server und einem Client, mithilfe dessen man über ein Android-Endgerät an Spielen teilnehmen kann. Ein Computer kann verwendet werden, um Partien zu beobachten.

Der Nutzer kann gegen bis zu vier andere Personen oder computergesteuerte Teilnehmer in digitalen „Tsuru“-Spielen antreten.

Größe und weitere Einstellungen des Spielfelds werden individuell vom Ersteller getroffen.

Die Anwender haben ein grundlegendes Verständnis für Android-Anwendungen und kennen die Regeln des Spiels. Unter diesen Voraussetzungen verwenden sie die Software intuitiv ohne Einarbeitungszeit.

2 Produkteinsatz

In diesem Kapitel wird der Einsatzbereich der Software erläutert. Zunächst wird auf den Problembereich eingegangen, dieser wird beschrieben und dabei auftretende Fachbegriffe werden im Glossar erläutert. Anschließend werden der Einsatzbereich und die Funktionen der Software in einem grafischen Modell und durch Darstellung der Geschäftsprozesse vorgestellt.

2.1 Beschreibung des Problembereichs

Der Problembereich umfasst das Spiel Tsuru. Während des Spiels versuchen zwei bis vier Personen durch taktisches Legen von Wegfeldern nicht den Rand des Spielfeldes zu erreichen oder mit einer anderen Spielfigur zu kollidieren.

Ein Spieler besitzt eine Spielfigur und eine bestimmte Anzahl an Wegfeldern. Auf den Wegfeldern sind Pfade gekennzeichnet, über die sich eine Spielfigur fortbewegen kann. Weiterhin gibt es ein Spielfeld, welches in einzelne Felder eingeteilt ist.

Zu Beginn einer Partie setzt jeder Spieler seine Spielfigur an einen Eintrittspunkt am Rand des noch leeren Spielfeldes. Die Spieler legen abwechselnd Wegfelder. Nach jedem Legen muss jeder Spieler seine Spielfigur bis zum Ende des Pfades vorziehen. Wenn ein Pfad zum Rand des Spielfeldes führt, scheidet der Spieler aus. Außerdem gibt es nicht bespielbare Felder, die vor dem Beginn der Partie festgelegt werden können. Diese führen ebenfalls zum Ausscheiden eines Spielers, falls der Weg zu einem solchen Feld führen sollte. Das Spiel wird beendet, wenn sich nur noch eine Spielfigur auf dem Spielfeld befindet oder bereits alle Spieler ausgeschieden sind.

Des Weiteren kann man dem Spiel auch als Beobachter beiwohnen. Dieser kann das Spielfeld inklusive aller Wegfelder jedes Spielers sehen.

In einem Turnier können KI-Teilnehmer gegeneinander antreten. In mehreren Partien wird hier der beste KI-Teilnehmer ermittelt.

2.2 Glossar

Das Glossar erläutert die in diesem Dokument verwendeten Begriffe im Zusammenhang mit dem Problembereich. Falls für das Verständnis förderlich, steht hinter dem Begriff in Klammern die englische Übersetzung.

Ausrichter

Der Ausrichter ist der Bediener der Spiel-Engine, somit lädt er eine Spielkonfiguration in die Spiel-Engine, startet das Spiel, wählt die Teilnehmer und setzt den beginnenden Teilnehmer.

Beobachter

Ein Beobachter des Spiels kann das Geschehen auf dem Spielbrett beobachten und hat Einsicht in die Wegfelder aller Teilnehmer.

Client

Ein Client ist ein Programm, das auf dem Endgerät eines Spielers ausgeführt wird und mit einem

Server kommuniziert.

Deck

Als Deck bezeichnen wir den Ablageort der bisher noch nicht gespielten Wegfelder eines Teilnehmer.

Eintrittspunkt

Der Eintrittspunkt beschreibt die Position am Spielfeldrand, an die ein Teilnehmer seine Spielfigur vor Beginn des Spiels setzt, sowie die Positionen am Rand eines Wegfelds an denen die Pfade beginnen bzw. enden.

Feld

Ein Feld ist ein Teil eines Spielfelds auf dem ein Wegfeld abgelegt werden kann.

Graphische Benutzeroberfläche (GUI - graphical user interface)

Eine graphische Benutzeroberfläche zeigt die Informationen (*hier: Deck, Spielfeld, Teilnehmer, etc.*) graphisch auf dem Client an.

KI

Eine KI ist eine künstliche Intelligenz, die an einem Tsuru-Spiel teilnehmen kann.

Partie

Eine Partie besteht aus mehreren Runden, die von zwei bis vier Teilnehmer gespielt werden. Das Ende einer Partie tritt ein, wenn einer oder mehrere Teilnehmer die Partie gewinnen oder der Ausrichter sie beendet.

Pfad

Ein Pfad verbindet zwei Eintrittspunkte auf einem Wegfeld miteinander.

Runde

Eine Runde ist ein Teilabschnitt einer Partie. Während einer Runde führt jeder Teilnehmer einen Spielzug aus.

Server

Ein Server ist ein Programm, mit dem sich mehrere Teilnehmer verbinden können um an einer Partie oder einem Turnier teilzunehmen.

Spielfeld

Ein Spielfeld ist ein quadratisches und rasterförmiges Feld auf dem das Tsuru-Spiel gespielt werden kann. Die Größe des Spielfelds entscheidet darüber, in wie viele Felder ein Spielfeld eingeteilt wird. Des Weiteren sind am Rand des Spielfelds Eintrittspunkte gekennzeichnet.

Spielfigur

Eine Spielfigur ist eine Figur, die einen Teilnehmer während einer Partie auf dem Spielfeld repräsentiert. Sie hat eine bestimmte Farbe und wird am Anfang einer Partie an einen Eintrittspunkt gesetzt. Sie bewegt sich im Laufe des Spiels über die durch die Wegfelder entstandenen Pfade.

Spielzug

Ein Spielzug ist die kleinste Einheit im Ablauf des Spiels. In einem Spielzug legt ein Teilnehmer ein Wegfeld auf das Spielfeld und die betroffenen Spielfiguren werden entlang ihrer Pfade weiterbewegt. Ein Spielzug kann eine Zeitbegrenzung haben. Nach Beendigung eines Spielzugs führt der nächste Teilnehmer seinen Spielzug aus.

Teilnehmer

Ein Teilnehmer kann sowohl ein menschlicher Spieler als auch eine KI sein. Er kann eine Partie gewinnen oder verlieren und besitzt eine Spielfigur. Des Weiteren wird ein Teilnehmer auch als Spieler bezeichnet.

Tsuro-Spiel

Ein Tsuro-Spiel besteht aus einem Spielfeld, vier Spielfiguren und mehreren Wegfeldern. Es wird zum Spielen einer Partie benötigt.

Turnier

Ein Turnier besteht aus mehreren Partien, durch die ein Gesamtsieger ermittelt wird. An einem Turnier nehmen nur KI-Teilnehmer teil.

Wegfeld

Ein Wegfeld besitzt an jeder Seite zwei Eintrittspunkte, die durch Pfade miteinander verbunden sind. Wegfelder werden während einer Partie von den Teilnehmern aus ihrem Deck auf ein einzelnes Feld des Spielfelds gelegt und können dabei auch gedreht werden.

2.3 Modell des Problembereichs

Das in Abbildung 1 dargestellte Modell des Problembereichs wurde anhand der im Lastenheft beschriebenen Vorgaben entwickelt und soll einen Überblick über den Problembereich bieten. Zur besseren Übersicht werden im Diagramm Kardinalitäten von 1 nicht dargestellt.

Eine *Tsuro-Partie* besteht aus einem Spiel (*TsuroSpiel*), zwei bis vier *Teilnehmern* und einer Zahl an *Spielzügen*, die den Verlauf der Partie modellieren. Weiter kann die Partie von einer beliebigen Zahl von *Beobachtern* betrachtet werden.

Die Teilnehmer besitzen zur Identifikation je einen Namen, eine Spielfigur und halten dem Lastenheft entsprechend $\lceil (\frac{\#Felder}{2}) \rceil$ *Wegfelder*. Zusätzlich wird der *Teilnehmer*, der als Nächstes seinen Spielzug machen darf, als „aktiver Spieler“ bezeichnet. Nach einem *Spielzug* wird diese Rolle an den nächsten Teilnehmer weitergegeben.

Die Komponente *TsuroSpiel* selbst besteht wiederum aus einem quadratischen Spielfeld, wobei *Größe* die Länge einer Spielfeldseite beschreibt, sowie einer Reihe von *Eintrittspunkten*, die das Spielfeld umranden. Das Spielfeld besitzt somit $[Größe^2]$ Felder, die ihre Position auf dem Spielfeld kennen und als bespielbar oder nicht bespielbar markiert werden können und $[Größe * 4 * 2]$ Eintrittspunkten am Rand des Feldes.

Auf die bespielbaren Felder kann ein Teilnehmer während seines Zugs je ein von ihm gehaltenes *Wegfeld* platzieren. Die Wegfelder können je eine von vier Ausrichtungen haben und bestehen selbst aus acht Eintrittspunkten, die je paarweise durch *Pfade* untereinander verbunden sind. Das Anlegen eines neuen Wegfeldes wird somit durch eine Verbindung der nun aneinandergrenzenden Eintrittspunkte dargestellt.

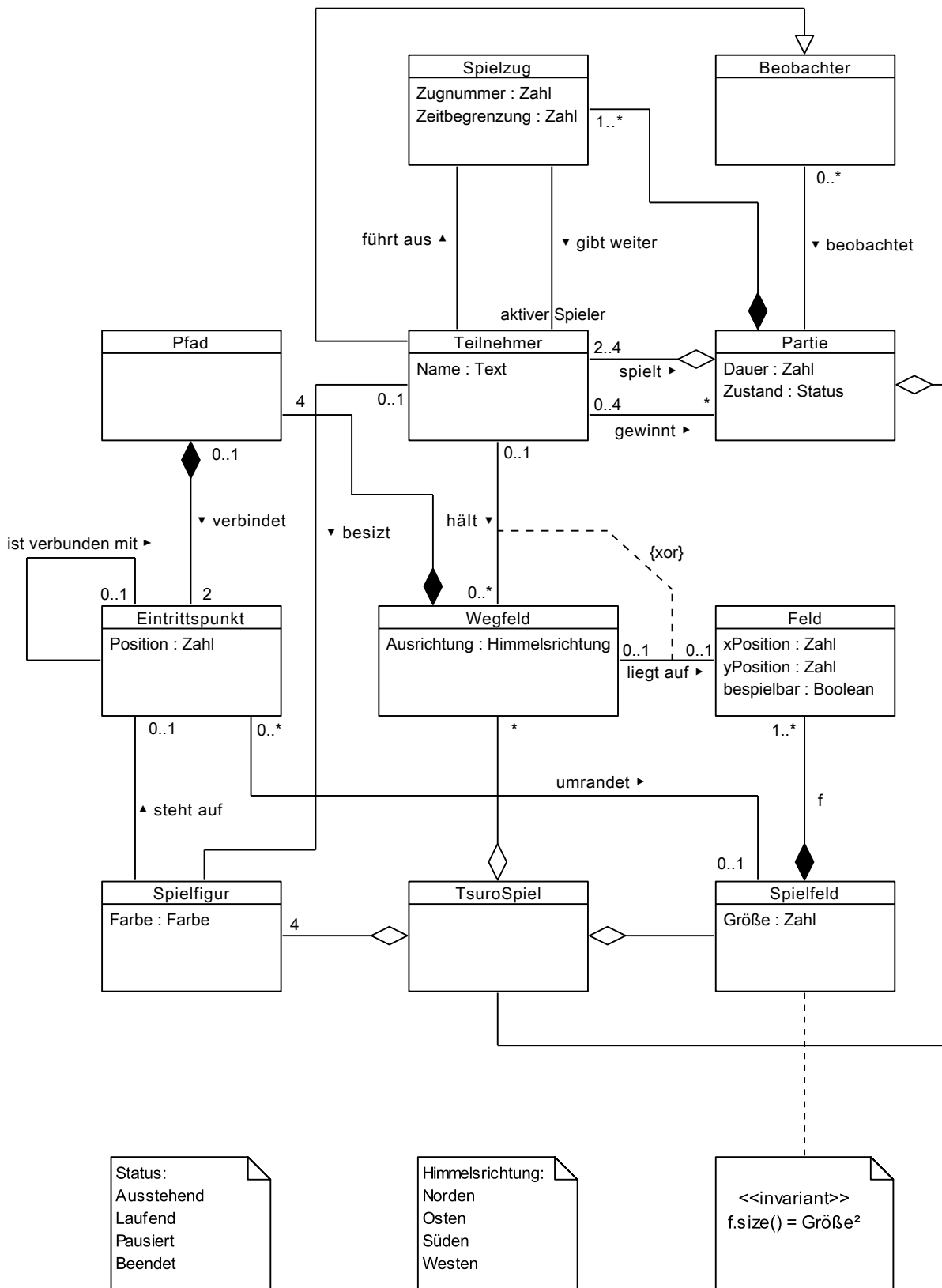


Abbildung 1: Modell des Problembereichs

2.4 Geschäftsprozesse

Im Folgenden werden die Abläufe der wesentlichen Geschäftsprozesse erläutert und beschrieben.

2.4.1 Partie ausrichten

Der Ablauf einer Partie sieht vor, dass nach dem Starten des Servers auf die Anmeldung der Teilnehmer gewartet wird. Haben sich ausreichend viele Spieler als Teilnehmer registriert und hat der Ausrichter eine Spielkonfiguration geladen, so kann die Partie nach Auswahl von zwei bis vier Teilnehmern gestartet werden. Nachdem ein Teilnehmer als erster aktiver Spieler festgelegt oder von der Spiel-Engine zufällig ermittelt wurde, wird die Partie als "laufend" bezeichnet. Während die Partie läuft, kann der Ausrichter das Spiel pausieren und/oder abbrechen. Bei einem Abbruch zählt die Partie als abgeschlossen und wird mit unentschieden gewertet.

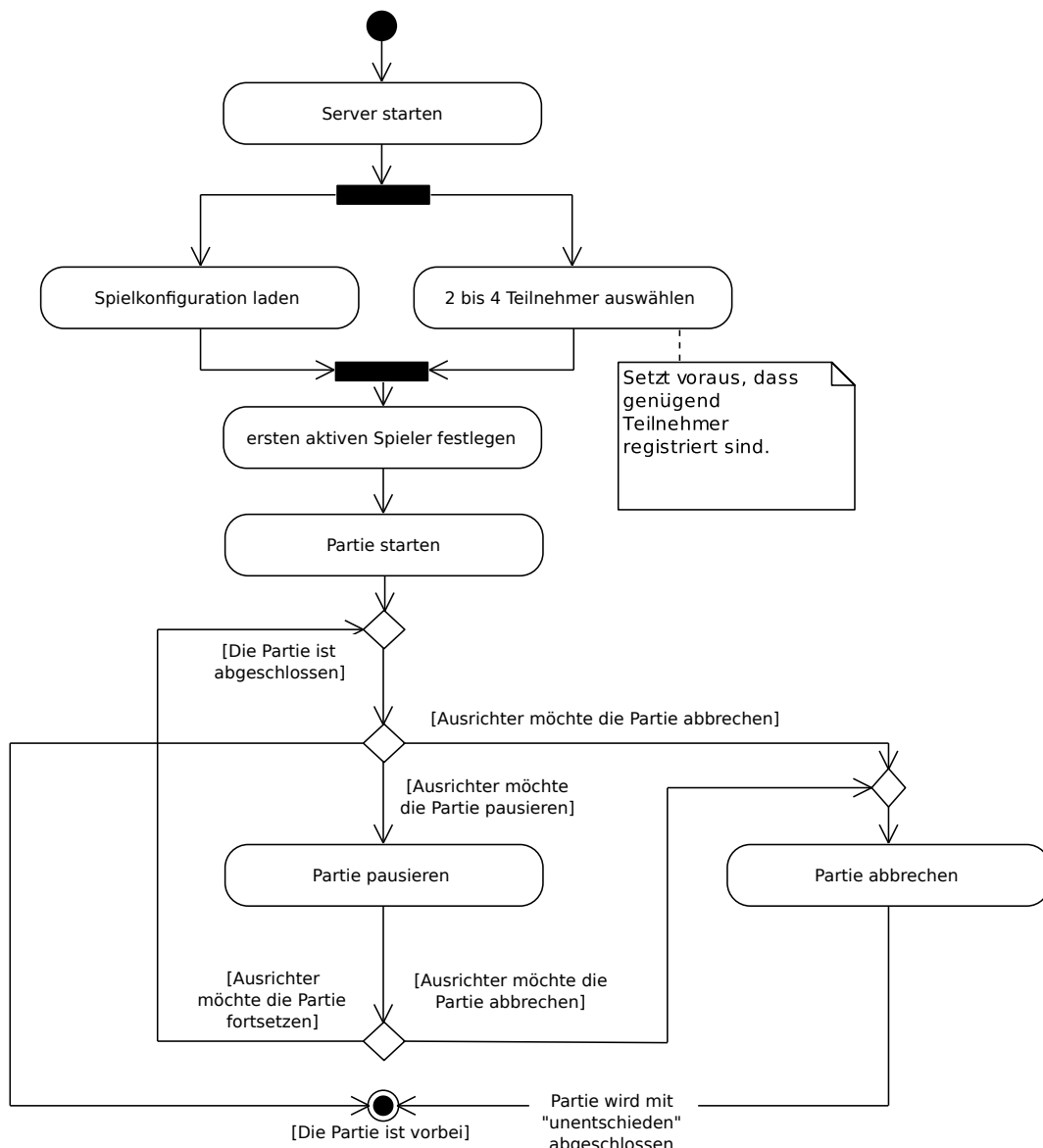


Abbildung 2: Geschäftsprozess - Partie ausrichten

2.4.2 Spielzug durchführen

Ist ein Spieler an der Reihe beziehungsweise wird der Spieler zu einem Spielzug aufgefordert, besteht sein Spielzug im Wesentlichen nur aus der Platzierung eines der von ihm gehaltenen Wegfelder. Sofern der Spielzug gültig ist, werden anschließend die Spielfigur des Spielers sowie alle anderen Spielfiguren, die an Eintrittspunkten des neu angelegten Wegfeldes liegen, weitergezogen. Bei einem ungültigen Spielzug wird dieser nicht ausgeführt und der betreffende Spieler aus dem Spiel entfernt.

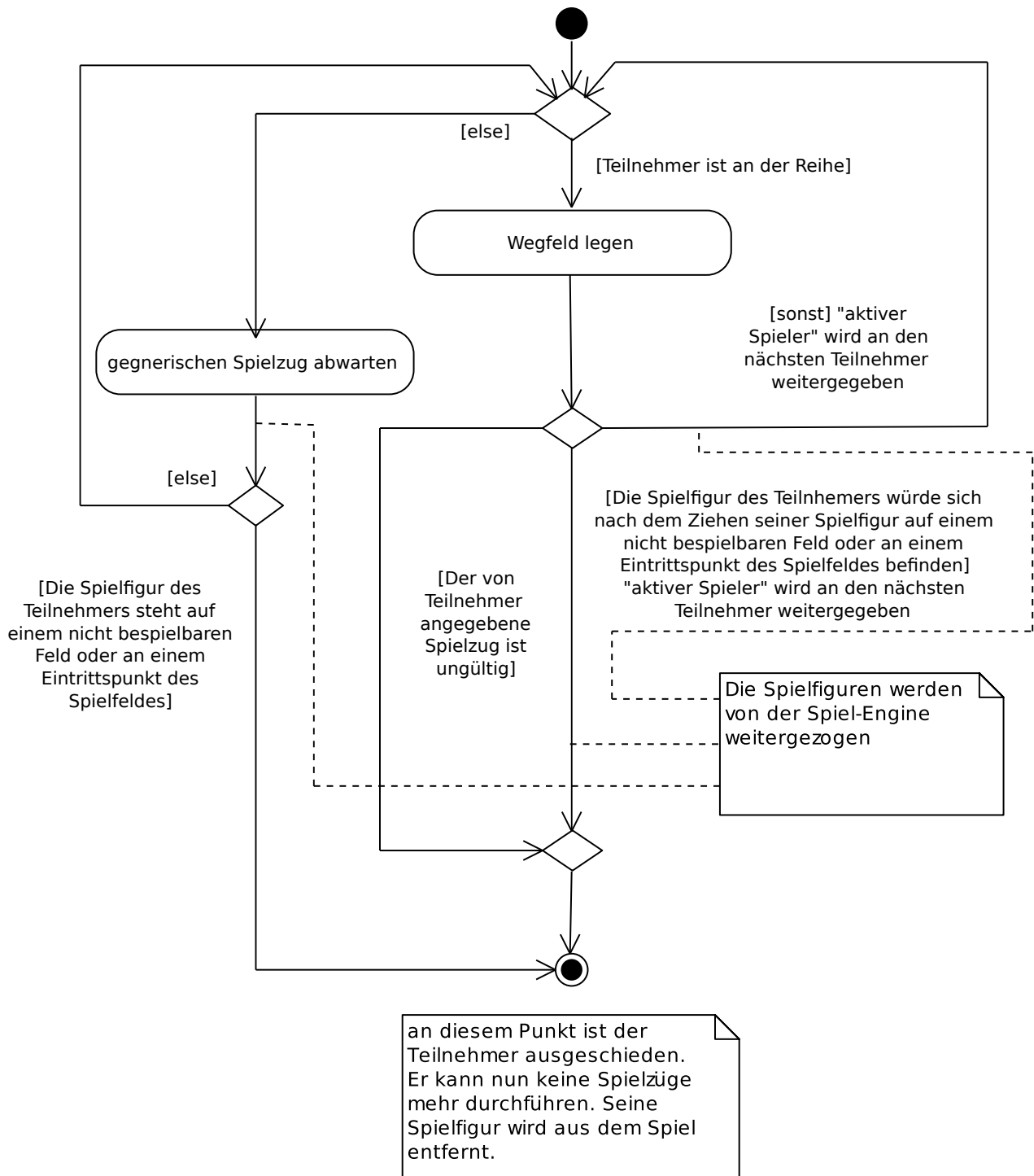


Abbildung 3: Geschäftsprozess - Spielzug durchführen

2.4.3 Spielkonfiguration erstellen

Für die Erstellung einer Spielkonfiguration ist vorgesehen, dass entweder eine bereits erstellte oder eine mit Standardwerten initialisierte Konfiguration als Ausgangspunkt genutzt werden kann. Ist der Ausgangspunkt gewählt, soll der Benutzer die Möglichkeit haben, die Größe des Spielfeldes zu ändern, Felder als bespielbar oder nicht bespielbar zu definieren, aus einer Liste an vorgeschlagenen Wegfeldern ein Deck zusammenzustellen oder die Pfade einzelner Wegfelder zu bearbeiten. Das Deck bezeichnet in diesem Fall die Wegfelder, die jedem Spieler zu Beginn der Partie ausgeteilt werden. Vor dem Speichern soll der Konfigurator prüfen, ob die eingestellte Konfiguration nur gültige Werte enthält und dem Nutzer gegebenenfalls eine Fehlermeldung ausgeben. Ist die Konfiguration gültig, wird sie gespeichert. Die Konfiguration soll nach dem Speichern weiter bearbeitet werden können.

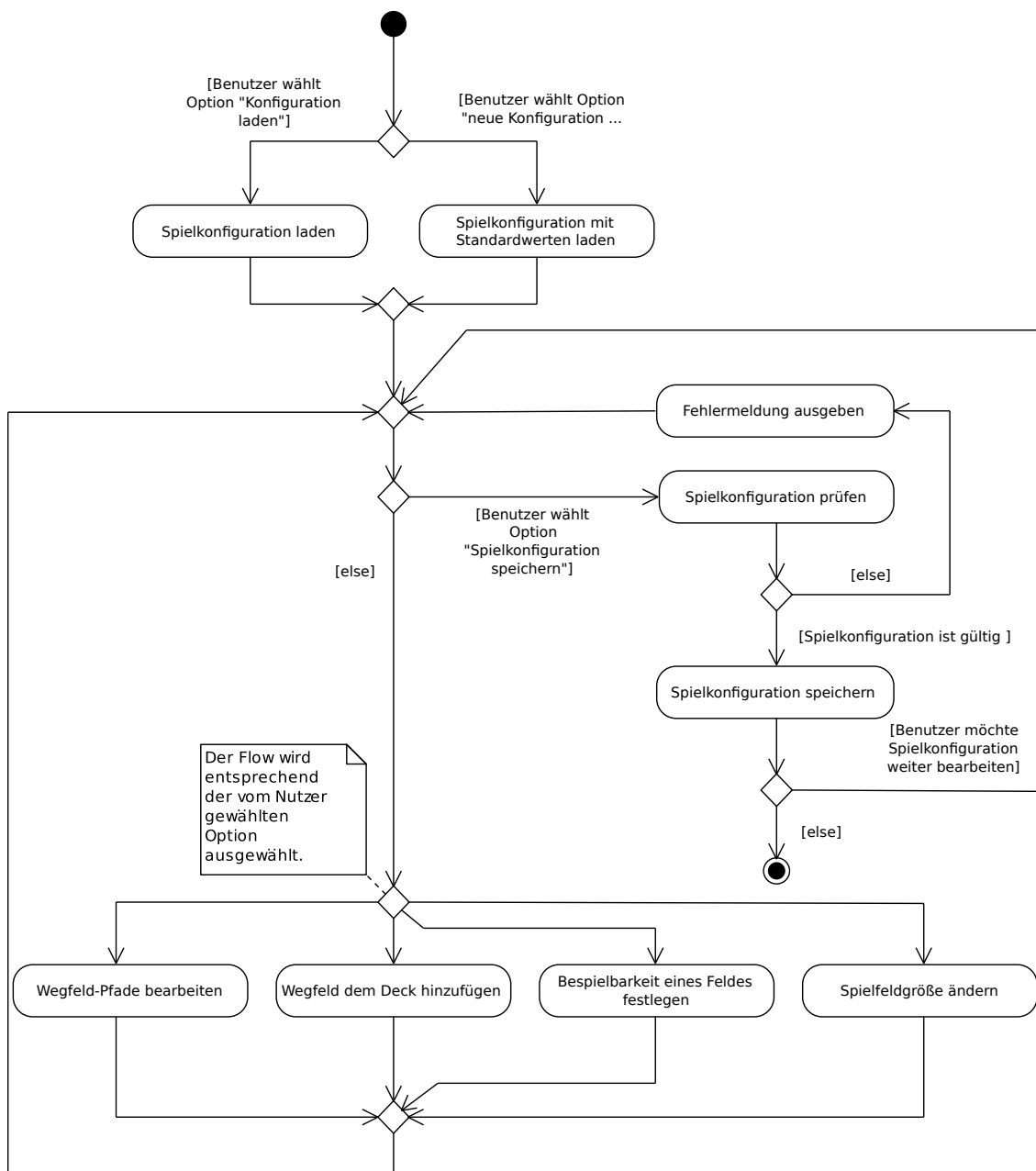


Abbildung 4: Geschäftsprozess - Spielkonfiguration erstellen

3 Produktfunktionen

3.1 Use Cases

In diesem Abschnitt werden die Produktfunktionen des verteilten Tsuru-Spiels dargestellt. Einführend wird die Funktionalität der zu entwickelnden Software durch vier Use-Case-Diagramme abgebildet, wobei Abbildung 6, Abbildung 8 und Abbildung 7 lediglich Teile von Abbildung 5 sind um eine bessere Übersicht zu gewährleisten. Anschließend wird der Ablauf der einzelnen Produktfunktionen (Use-Cases) im Detail behandelt.

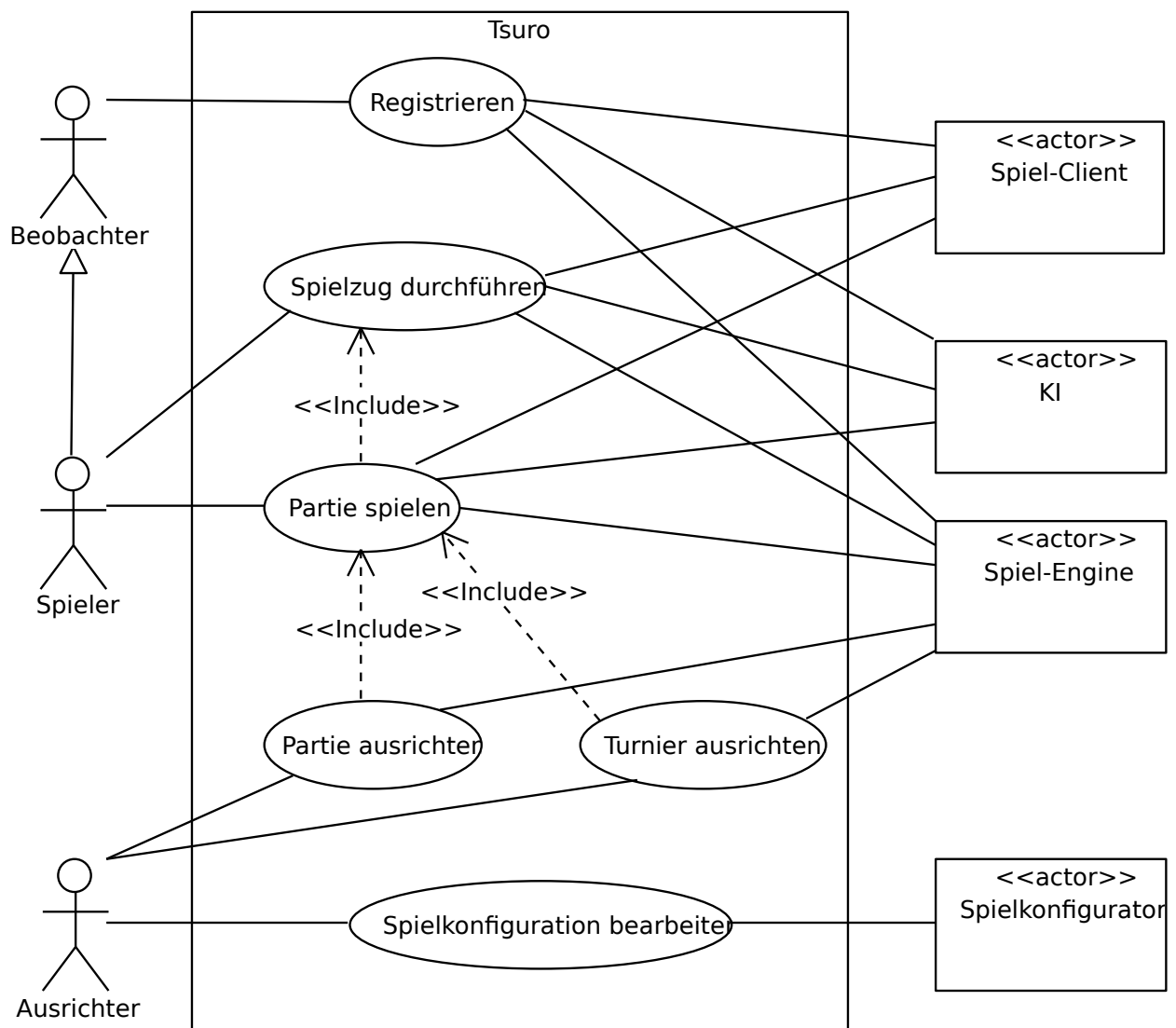


Abbildung 5: UML-Use-Case Diagramm - gesamte Übersicht.

Hier wurden alle Use-Cases aufgelistet und mit den Akteuren verbunden, die die jeweiligen Use-Cases ausführen können.

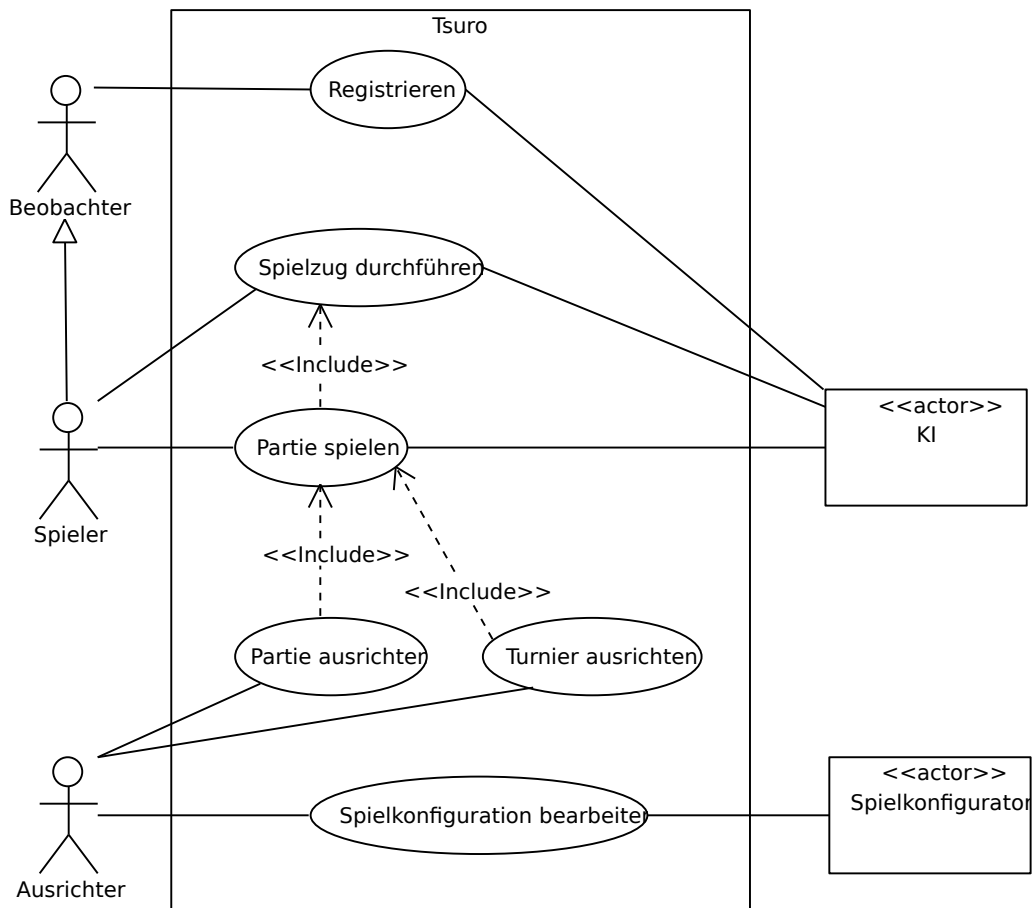


Abbildung 6: UML Use Case Diagramm - KI

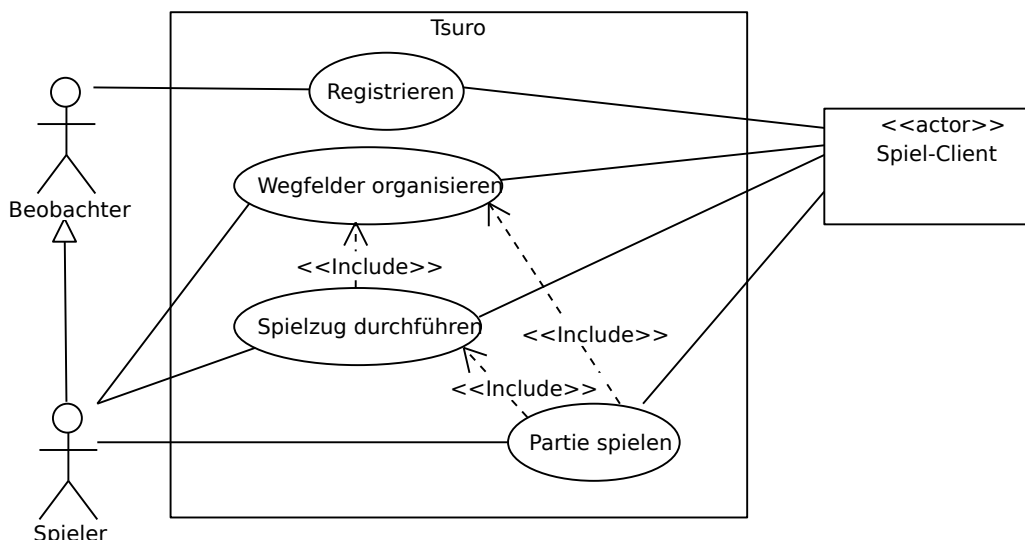


Abbildung 7: UML Use Case Diagramm - Client

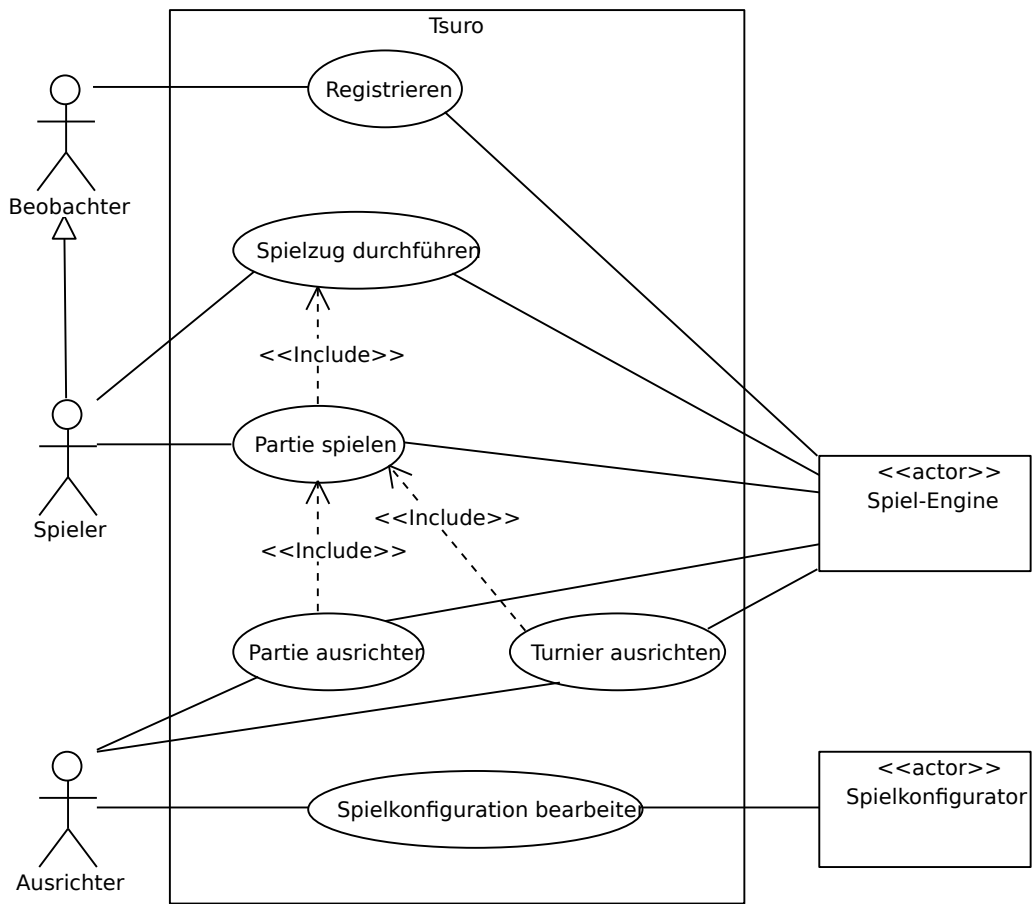


Abbildung 8: UML Use Case Diagramm - Engine

BEMERKUNG: *Skizzen für sowohl den Smartphone (Abbildung 18), also auch den PC Client (Abbildung 16) befinden sich in Anhang A.*

3.1.1 Registrieren

Der Use-Case "Registrieren" beschreibt den Vorgang, bei dem sich ein Nutzer mit einem Server verbindet, um einer auf diesem Server gehosteten Partie beizutreten.

BEMERKUNG: *Da jeder Spieler auch ein Beobachter ist, wird hier der Begriff "Beobachter" als Stellvertreter für einen menschlichen Akteur benutzt.*

Charakterisierende Informationen	
Name	Registrieren
Ziel des Nutzers	Verbindung zwischen Beobachter und Spiel-Engine
Vorbedingung	Beobachter ist im Hauptmenü
Nachbedingung	Verbindung zwischen Beobachter-Client und Spiel hergestellt
Auslösendes Ereignis	Beobachter wählt "Registrieren"
Nutzer	Beobachter, Spiel-Engine, Client
Umgebende Systemgrenze	Tsuro

Szenario: Registrieren		
Schritt	Akteur	Beschreibung der Aktivität
1	Beobachter	wählt "Registrieren"
2	Client	fragt nach Netzwerkadresse
3	Beobachter	gibt Netzwerkadresse ein
4	Client	stellt Verbindung mit Server her
5	Client	zeigt Liste der auf dem Server verfügbaren Spiele
6	Client	fragt nach Name, Spiel und Rolle
7	Beobachter	wählt Name, Spiel und Rolle
8	Beobachter	bestätigt Eingabe
9	Client	übermittelt Daten an Spiel-Engine
10	Spiel-Engine	fügt Beobachter zu Spiel hinzu
11	Client	benachrichtigt Beobachter über erfolgreichen Beitritt

Alternativszenario: Registrieren		
Bedingung: Keine Antwort vom Server		
Schritt	Akteur	Beschreibung der Aktivität
4.1	Client	zeigt "Fehler bei der Verbindung" an
4.2		zurück zu Schritt 2

Alternativszenario: Registrieren		
Bedingung: Beobachter wählt kein Spiel		
Schritt	Akteur	Beschreibung der Aktivität
9.1	Client	übermittelt Daten an Spiel-Engine
9.2	Spiel-Engine	fügt Beobachter zum Beobachterpool hinzu
9.3		zurück zu Schritt 10

Alternativszenario: Registrieren		
Bedingung: Kein beitrebares Spiel gewählt oder kein beitrebares Spiel vorhanden		
Schritt	Akteur	Beschreibung der Aktivität
10.1	Client	zeigt dem Beobachter an, dass keinem Spiel beigetreten werden konnte und den zugehörigen Grund
10.2		zurück zu Schritt 5

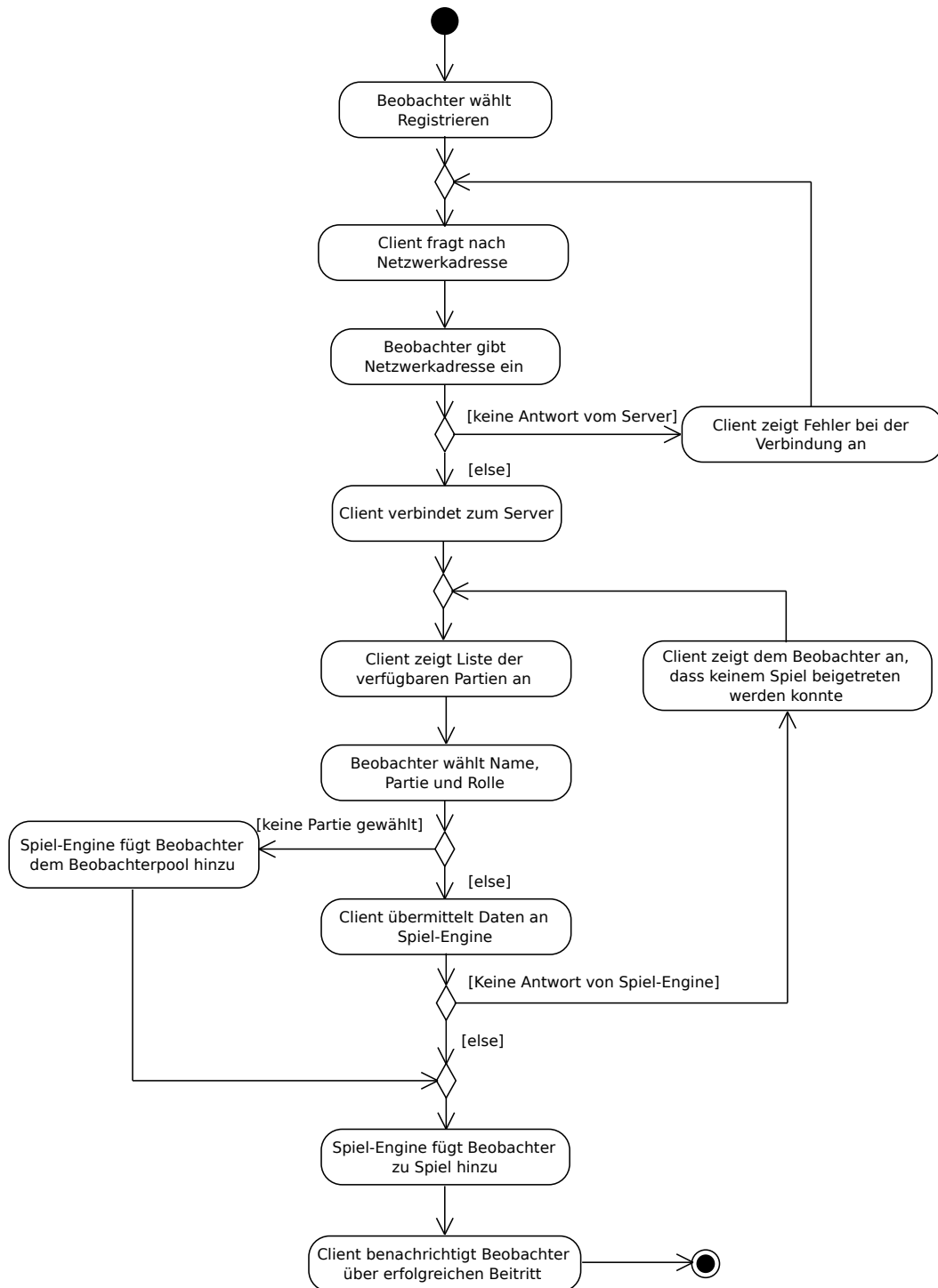


Abbildung 9: Aktivitätsdiagramm - Registrieren

3.1.2 Wegfelder organisieren

Charakterisierende Informationen	
Name	Wegfelder organisieren
Ziel des Nutzers	Wegfelder auf dem eigenen Deck drehen oder umsortieren
Vorbedingung	Client ist in laufendem Spiel registriert
Nachbedingung	Spieler ist mit Organisation seiner Wegfelder zufrieden
Auslösendes Ereignis	Spieler möchte Wegfelder umsortieren
Nutzer	Client, Spieler
Umgebende Systemgrenze	Tsuro

Szenario: Wegfelder organisieren		
Schritt	Akteur	Beschreibung der Aktivität
1	Spieler	zieht Wegfeld in seinem Deck an eine andere Position
2	Client	ändert die Position des umgelegten Wegfeldes
3	Spieler	tippt auf ein Wegfeld auf seinem Deck
4	Client	dreht das angetippte Wegfeld um 90° im Uhrzeigersinn
5		Springe zu Schritt 1

BEMERKUNG: Die beiden Schritte 1 und 2 bzw. 3 und 4 sind optional. Um den Use-Case abzuschließen, müssen jedoch entweder die Schritte 1 und 2, oder 3 und 4 mindestens einmal ausgeführt werden. Schritt 5 ist ebenfalls optional.

Alternativszenario: Wegfelder organisieren		
Bedingung: Die Zielposition des Wegfeldes ist schon besetzt		
Schritt	Akteur	Beschreibung der Aktivität
2.1	Client	vertauscht das Wegfeld an der neuen Position mit dem an der alten
2.2		zurück zu Schritt 3

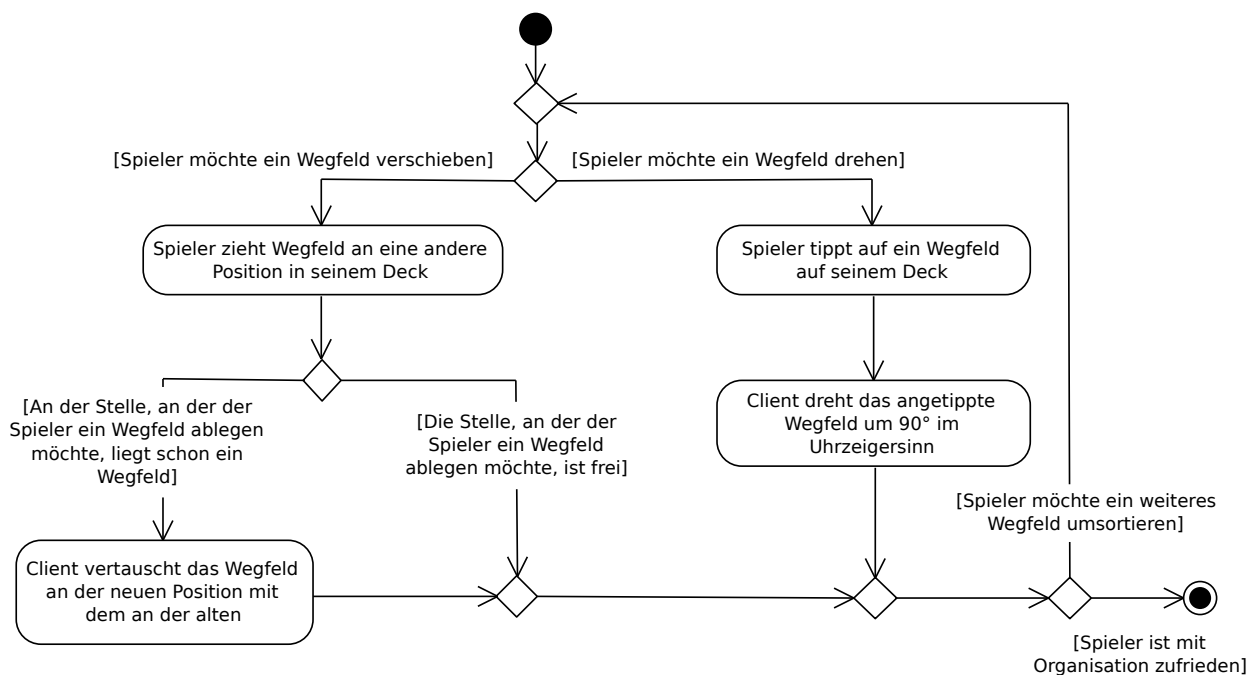


Abbildung 10: Aktivitätsdiagramm - Wegfelder Organisieren

3.1.3 Spielzug durchführen

Charakterisierende Informationen	
Name	Spielzug durchführen
Ziel des Nutzers	Wegfeld legen
Vorbedingung	Spieler X ist an der Reihe
Nachbedingung	Neues Wegfeld liegt auf dem Spielfeld / Spieler X hat verloren
Auslösendes Ereignis	Spieler X ist an der Reihe
Nutzer	Spieler
Umgebende Systemgrenze	Tsuro

Szenario: Spielzug durchführen		
Schritt	Akteur	Beschreibung der Aktivität
1	Spiel-Engine	benachrichtigt alle Clients, dass Spieler X am Zug ist
2	Client X	fordert Spieler X zum Spielzug auf
3	Client	zeigt an, dass Spieler X am Zug ist und startet den Timer
4	Spieler X	zieht ein Wegfeld aus seinem Deck auf das Spielfeld
5	Client X	sendet das Spielfeld und seine Drehung an die Spiel-Engine
6	Spiel-Engine	sendet das Ergebnis des Spielzugs an alle Clients
7	Client	setzt den Timer zurück
8	Client	aktualisiert die Anzeige

Alternativszenario: Spielzug durchführen		
Bedingung: Server sendet ein "Kick Command"		
Schritt	Akteur	Beschreibung der Aktivität
8	Client	aktualisiert die Anzeige und meldet seinem Benutzer, welche Spieler verloren haben

BEMERKUNG: Wenn als Akteur "Client" angegeben ist, so sind alle im Spiel anwesenden Clients gemeint (ausdrücklich auch Beobachter), "Client X" bezeichnet den Client des Spielers X.

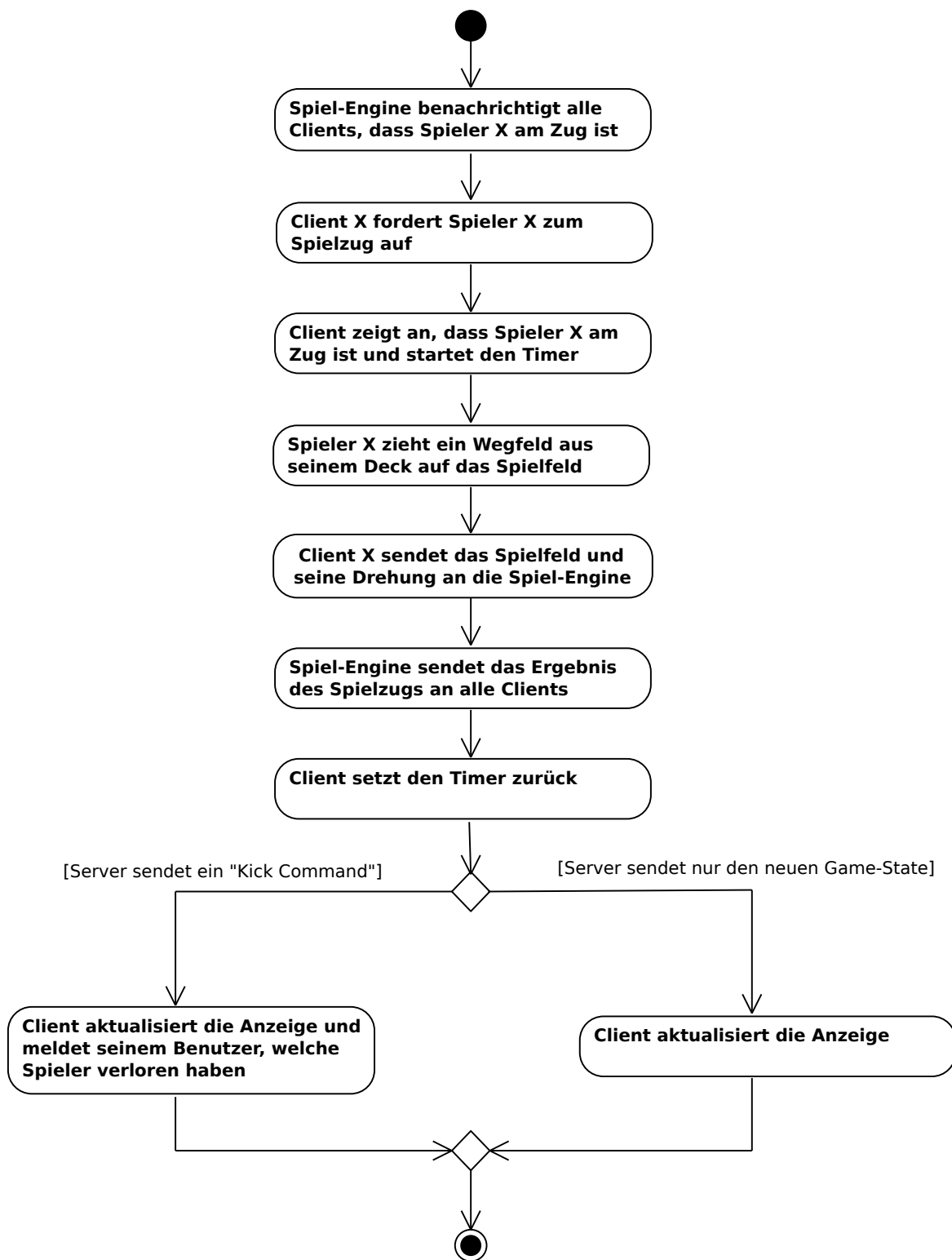


Abbildung 11: Aktivitätsdiagramm - Spielzug durchführen

3.1.4 Partie spielen

Der Use-Case "Partie spielen" beschreibt den Vorgang beim Spielen einer Partie Tsuru.

Charakterisierende Informationen	
Name	Partie spielen
Ziel des Nutzers	Spielen einer Partie Tsuru
Vorbedingung	Use-Case "Partie ausrichten" ist erfolgreich durchgelaufen
Nachbedingung	Partie ist beendet und Startbildschirm wird angezeigt
Auslösendes Ereignis	Der Ausrichter hat eine Partie gestartet
Nutzer	Teilnehmer, Spiel-Engine, Spiel-Client, Ausrichter
Umgebende Systemgrenze	Tsuru

Szenario: Partie spielen		
Schritt	Akteur	Beschreibung der Aktivität
1	Spiel-Engine	legt Teilnehmerreihenfolge (unter Beachtung der Konfiguration des Ausrichters) fest
2	Client	Use-Case "Spielzug durchführen" wird ausgeführt
3	Spiel-Engine	bestimmt gemäß Reihenfolge nächsten Teilnehmer, der am Zug ist
4	Spiel-Engine	Solange kein Spielergebnis feststeht, gehe zu Schritt 3
5	Spiel-Engine	beende Partie

BEMERKUNG: Der Ausrichter kann zu jeder Zeit die Partie abbrechen. Im Falle eines Abbruchs werden alle Teilnehmer und Beobachter über den Abbruch benachrichtigt. BEMERKUNG: Zu jedem Zeitpunkt können Teilnehmer eine Partie aus unterschiedlichen Gründen verlassen. Im Falle eines solchen "Disconnects" wird die Spielfigur vom Spielfeld genommen. Die restlichen Teilnehmer und Beobachter werden über das Ausscheiden informiert. Danach wird der normale Ablauf fortgeführt.

Alternativszenario: Partie spielen		
Bedingung: Es gibt nur noch maximal einen nicht ausgeschiedenen Teilnehmer		
Schritt	Akteur	Beschreibung der Aktivität
3.1a	Spiel-Engine	benachrichtigt alle Teilnehmer über den bzw. die Sieger und das Partieende
3.1a		zurück zu Schritt 6

Alternativszenario: Partie spielen		
Bedingung: Der Ausrichter pausiert die Partie in der Spiel-Engine		
Schritt	Akteur	Beschreibung der Aktivität
3.1b	Spiel-Engine	benachrichtigt alle Teilnehmer, dass die Partie pausiert wird
3.2b	Spiel-Client	zeigt "Pause" an
3.3b	Ausrichter	setzt Partie fort
3.4b	Spiel-Engine	benachrichtigt alle Teilnehmer, dass die Partie fortgesetzt wird
3.5b		zurück zu Schritt 4

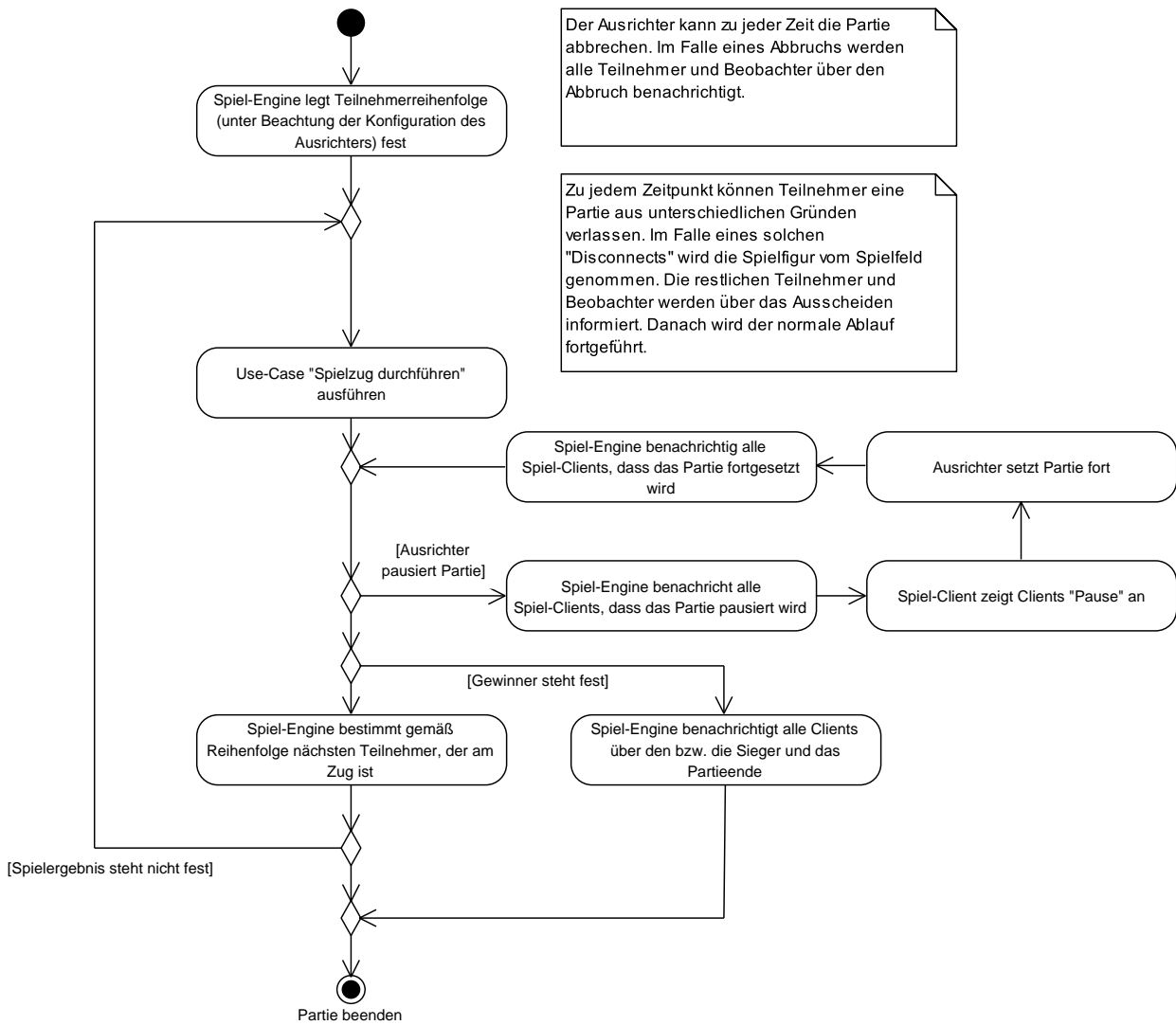


Abbildung 12: Aktivitätsdiagramm - Partie spielen

3.1.5 Partie ausrichten

Der Use-Case "Partie ausrichten" beschreibt das Ausrichten einer Partie mit ausgewählten Teilnehmern und einer bestimmten Spielkonfiguration.

Charakterisierende Informationen	
Name	Partie ausrichten
Ziel des Nutzers	Ausrichten einer Partie
Vorbedingung	Spiel-Engine ist vom Spielausrichter gestartet und es sind bereits genügend Teilnehmer registriert
Nachbedingung	Partie ist gestartet
Auslösendes Ereignis	Ausrichter möchte Partie starten
Nutzer	Ausrichter, Spiel-Engine
Umgebende Systemgrenze	Tsuro

BEMERKUNG: Sollten vor dem Partiestart registrierte Teilnehmer aus nicht näher bestimmten Gründen die Verbindung schließen oder verlieren, wird die Auswahl des Ausrichters bzgl. der Teilnehmer, die an

der Partie teilnehmen sollen, zurückgesetzt. In diesem Falle muss der Ausrichter erneut eine Auswahl über die Teilnehmer treffen.

Szenario: Partie ausrichten		
Schritt	Akteur	Beschreibung der Aktivität
1	Ausrichter	wählt Namen für Spielraum aus
2	Spiel-Engine	zeigt Startbildschirm (Liste aller verfügbaren Spieler und Konfigurationen) an
3	Ausrichter	wählt zwei bis vier Teilnehmer aus, die an der Partie teilnehmen sollen
4	Ausrichter	wählt ob er zudem Beobachter sein möchte
5	Ausrichter	wählt eine Spielkonfiguration aus
6	Ausrichter	bestimmt, welcher Teilnehmer anfängt (manuelle Auswahl oder per Zufall)
7	Ausrichter	startet Partie
8	Spiel-Engine	startet Partie und benachrichtigt Teilnehmer/Beobachter

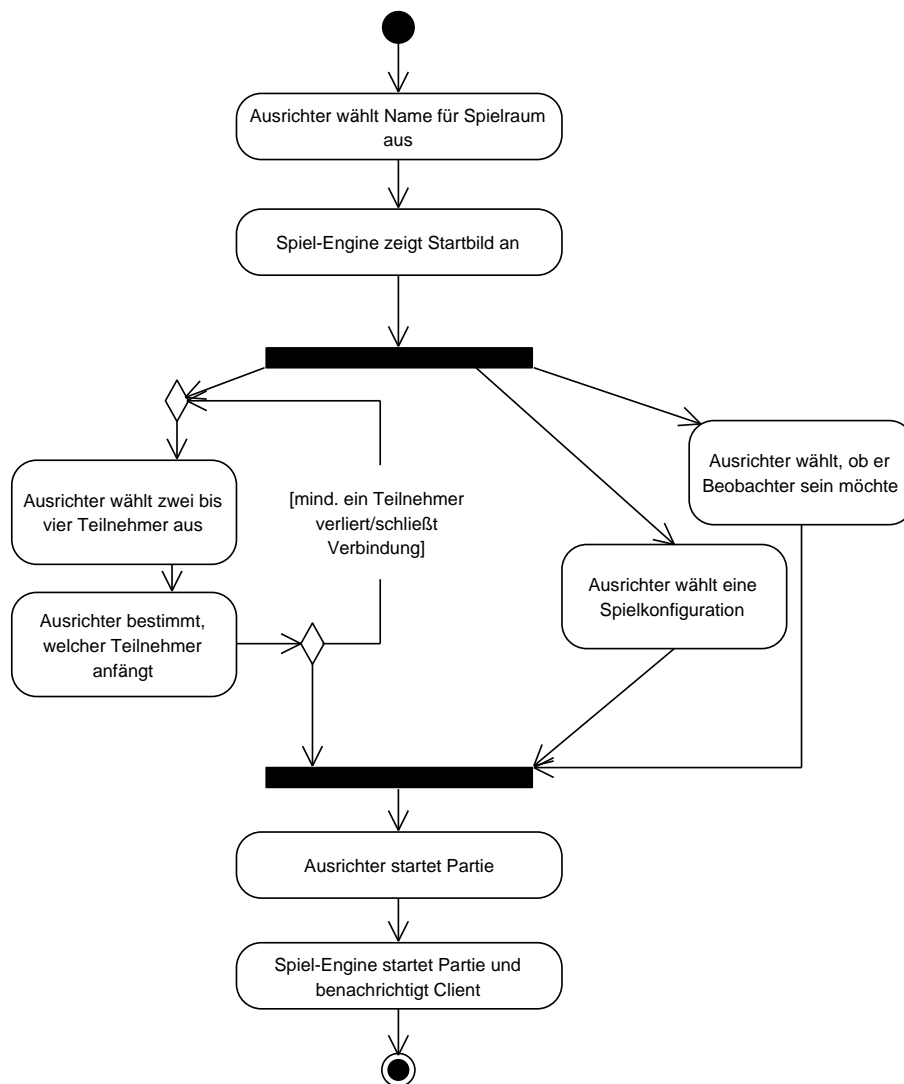


Abbildung 13: Aktivitätsdiagramm - Partie ausrichten

3.1.6 Turnier ausrichten

Charakterisierende Informationen	
Name	Turnier ausrichten
Ziel des Nutzers	Ein Turnier ausrichten
Vorbedingung	Genügend Spieler sind registriert
Nachbedingung	Alle Spiele des Turniers sind beendet
Auslösendes Ereignis	Benutzer der Spiel-Engine ("Ausrichter") wählt "Turnier ausrichten"
Nutzer	Ausrichter, Teilnehmer
Umgebende Systemgrenze	Tsuro

Szenario:		
Schritt	Akteur	Beschreibung der Aktivität
1	Ausrichter	wählt die Höhe des Turnierbaums
2	Ausrichter	wählt zu jeder Turnierebene die Anzahl der an einer Partie beteiligten Spieler und die Spielkonfigurationsdatei
3	Spiel-Engine	zeigt den Turnierbaum an
4	Spiel-Engine	fordert den Ausrichter zur Auswahl der Spieler für die erste Turnier-Ebene aus einer Liste auf
5	Ausrichter	wählt für jede Partie der ersten Turnier-Ebene die Spieler aus
6	Spiel-Engine	zeigt den Turnierbaum mit Spielern der ersten Turnier-Ebene an
7	Ausrichter	startet das Turnier
8	Spiel-Engine	richtet für jede Turnier-Ebene nacheinander alle Partien aus und zeigt Ergebnisse, sowie Teilnehmer im Turnierbaum an
9	Spiel-Engine	zeigt den vollständigen Turnierbaum und die Platzierung jedes Spielers an

BEMERKUNG: Die Spiel-Engine prüft jede Nutzereingabe auf Korrektheit und lässt die nachfolgenden Schritte nur bei gültigen Eingaben zu. Der Ausrichter kann außerdem zu jedem Zeitpunkt die Auswahl/Ausführung abbrechen.

BEMERKUNG: Der Turnierbaum ist ein vollständiger Baum, der Übersicht über die Partien des Turniers gibt. Jeder Knoten des Baums stellt eine Partie zwischen den Gewinnern der Partien der Ebene darunter dar. Da der Baum keine Knoten tiefer als seine Blätter besitzt, stellen diese die ersten Partien des Turniers da. Die Wurzel repräsentiert die finale Partie.

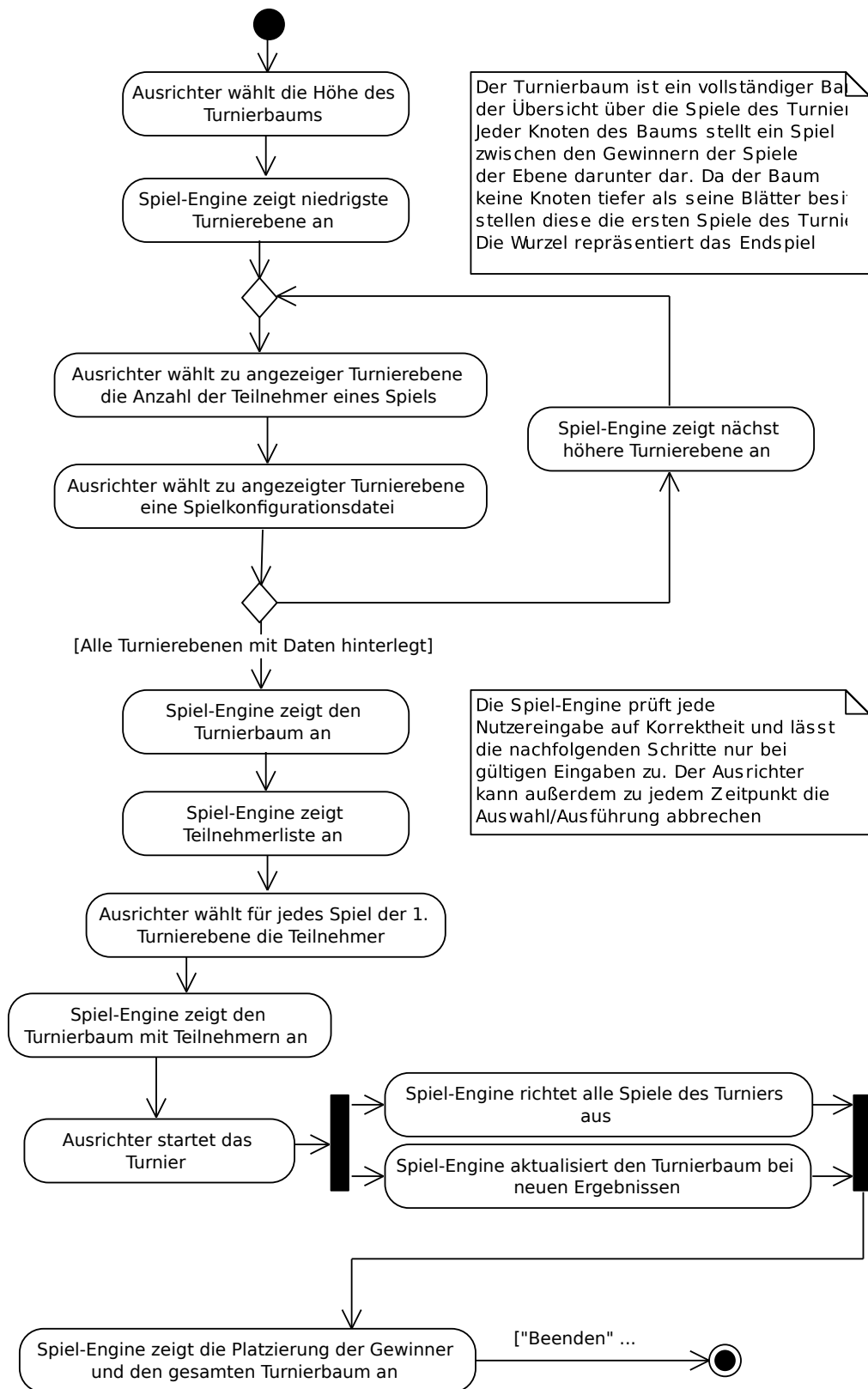


Abbildung 14: Aktivitätsdiagramm - Turnier ausrichten

3.1.7 Spielkonfiguration bearbeiten

Dieser Use Case stellt dar, wie ein Nutzer ein Tsuru Spiel konfigurieren kann. Dazu wird ein eigens entwickelter Spielkonfigurator eingesetzt, mit welchem man eigene und Standardkonfigurationen laden, bearbeiten und speichern kann.

BEMERKUNG: Eine Skizze für den Spielkonfigurator (Abbildung 17) befindet sich im Anhang A.

Charakterisierende Informationen	
Name	Spielkonfiguration bearbeiten
Ziel des Nutzers	eine Spielkonfiguration zu erstellen oder zu verändern
Auslösendes Ereignis	der Spielkonfigurator wurde geöffnet
Nutzer	Ausrichter
Umgebende Systemgrenze	Tsuru

Szenario: Spielkonfiguration bearbeiten		
Schritt	Akteur	Beschreibung der Aktivität
1	Ausrichter	lädt vorhandene oder erstellt neue Spielkonfiguration
2	Ausrichter	legt die Größe des Spielfeldes fest
3	Spielkonfigurator	schlägt einen Satz von Wegfeldern vor
4	Ausrichter	wählt Wegfelder
5	Ausrichter	konfiguriert die Pfade der Wegfelder
6	Ausrichter	platziert nicht bespielbare Felder auf dem Spielfeld
7	Ausrichter	setzt eine Zeitbeschränkung für die KI
8	Ausrichter	speichert die Spielkonfiguration

BEMERKUNG: Die Schritte 2,4,5,6 und 7 sind optional vom Ausrichter auszuführen, da entweder die Werte einer geladenen Spielkonfiguration anliegen oder, im Falle einer neuen Spielkonfiguration, Standardwerte gesetzt wurden.

Alternativszenario: Spielkonfiguration bearbeiten		
Bedingung: Das geladene Dokument ist fehlerhaft		
Schritt	Akteur	Beschreibung der Aktivität
1.1	Spielkonfigurator	zeigt "Spielkonfiguration konnte nicht geladen werden"
1.2		zurück zu Schritt 1

Alternativszenario: Spielkonfiguration bearbeiten		
Bedingung: Der Spielkonfigurator stellt fehlerhafte Konfiguration fest		
Schritt	Akteur	Beschreibung der Aktivität
8.1	Spielkonfigurator	zeigt "Spielkonfiguration ist mit inkorrekten Werten belegt"
8.2	Ausrichter	verändert ungültige Konfiguration
8.3		zurück zu Schritt 8

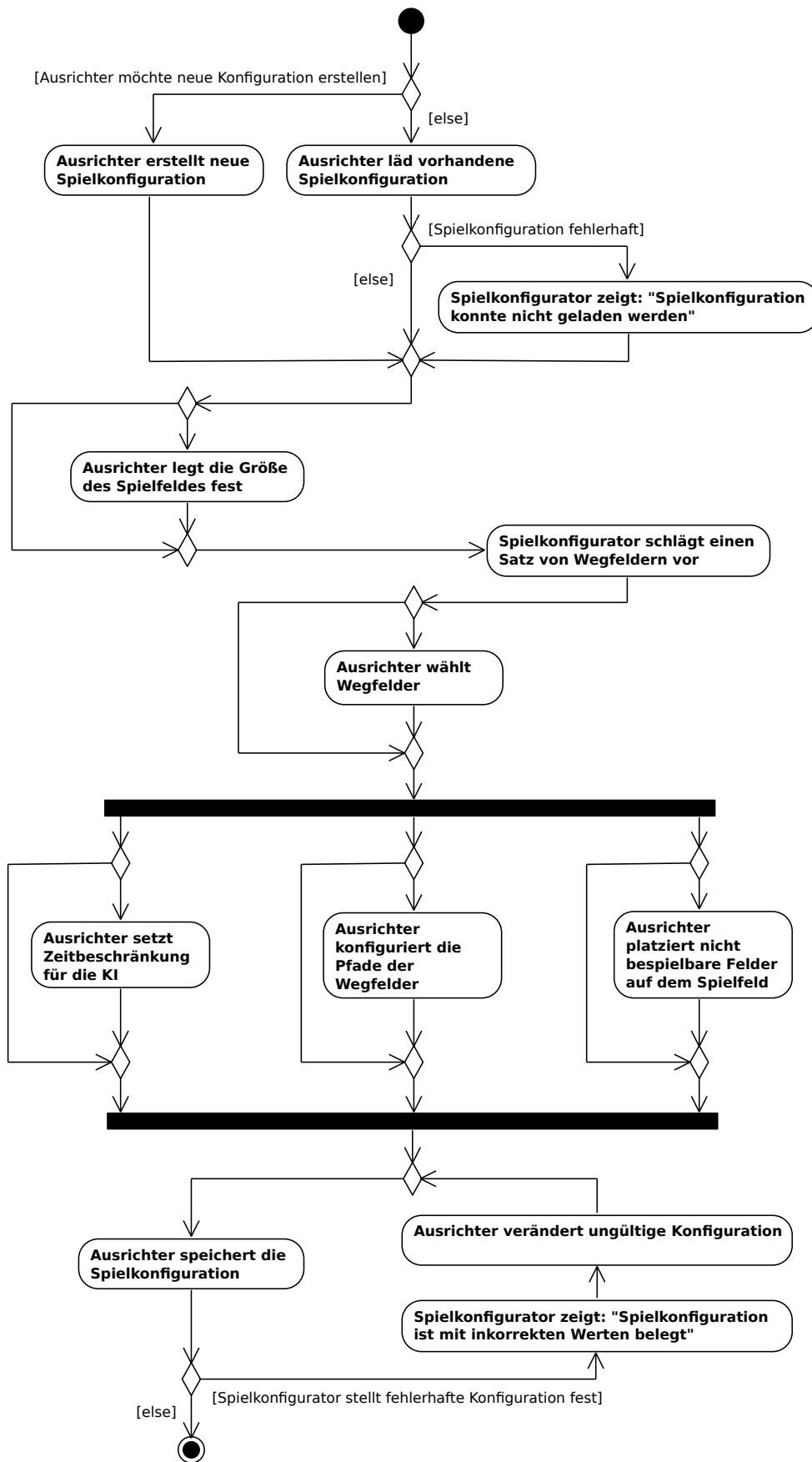


Abbildung 15: Aktivitätsdiagramm - Spielkonfiguration bearbeiten

4 Produktcharakteristiken

In diesem Abschnitt werden die Mindestanforderungen für Hard- und Softwareumgebung festgelegt sowie die nicht-funktionalen Anforderungen an die Software.

4.1 Systemanforderungen

Für eine optimale Ausführung des Tsuru-Spiels, werden bestimmte Anforderungen an das System gestellt, auf dem unser Produkt betrieben werden kann. Um eine ausreichend gute Ausführung zu gewährleisten, müssen sowohl Hardwareumgebung als auch Softwareumgebung des Rechners sowie der Smartphones die an sie gestellten Mindestanforderungen erfüllen. *Smarten* kann eine fehlerfreie Funktionalität mit älterer Hardware bzw. älteren Softwareversionen nicht garantieren.

4.1.1 Hardwareumgebung - Smartphone

Für die auf einem Smartphone ausführbaren Komponenten unseres Produktes empfehlen wir mindestens folgende Hardware:

- Prozessor: 4x1400 MHz; Samsung Exynos 4412
- Arbeitsspeicher: 1500 MB
- Freier Festplattenspeicher: 250 MB
- Displaydiagonale: 4,8 Zoll
- Displayauflösung: 720x1280 Pixel
- Netzwerkverbindung

BEMERKUNG: *Diese Werte entsprechen einem Samsung Galaxy S3 Neo Smartphone.*

4.1.2 Softwareumgebung - Smartphone

Auf dem Smartphone muss mindestens Android 4.4 "KitKat" (API Level 19) installiert sein.

4.1.3 Hardwareumgebung - PC

Die auf einem PC ausführbaren Komponenten unseres Produktes erfordern mindestens folgende Hardwareanforderungen:

- Prozessor: 2,53 GHz Intel Core 2 Duo
- Arbeitsspeicher: 2048 MB
- Freier Festplattenspeicher: 400 MB
- Bildschirmauflösung: 1024x768 Pixel
- Netzwerkverbindung

4.1.4 Softwareumgebung - PC

An die Softwareumgebung eines Rechnersystems, welches unser Produkt ausführen soll, werden folgende Anforderungen gestellt:

- Betriebssystem: Microsoft Windows 10 (64 Bit)

BEMERKUNG: Die Lauffähigkeit wird nur für oben genanntes System garantiert; voraussichtlich wird der Betrieb aber auch unter Microsoft Windows 7 und 8, Linux (Ubuntu 10.04 oder höher) und Mac OS X möglich sein.

- Java Runtime Environment SE 8 oder neuer
- Eclipse in der Version 4.5.2 (Mars SR2)

4.2 Nicht funktionale Anforderungen

Name:	Codekommentare
Typ:	PFLEGE
Beschreibung:	Codekommentare sind im Javadoc-Format geschrieben. Dieses erleichtert die Übersicht und Wartbarkeit. Alle Codeabschnitte werden mit Kommentaren versehen, sodass diese verständlich und Veränderungen am Code schnell möglich sind. Die Kommentare sind auf Englisch, damit auch internationale Entwicklerteams das Produkt erweitern können.

Name:	Java Style Guidelines
Typ:	PFLEGE
Beschreibung:	Um einen übersichtlichen Code zu garantieren, wird der Code Einheitlich nach den Google Java Style Guidelines geschrieben.

Name:	Dokumentation
Typ:	PFLEGE
Beschreibung:	Um die Erweiterbarkeit des Produktes sicher zu stellen wird eine Dokumentation angelegt. Diese soll Entwicklern das nachträgliche Verstehen des Codes vereinfachen und damit zukünftige Änderungen und Erweiterungen vereinfachen.

Name:	Trennung von Daten, Logik und View
Typ:	PFLEGE
Beschreibung:	Um einen besseren Austausch von einzelnen Komponenten zu gewährleisten, werden wir das Produkt nach einer Model-View-Controller Architektur implementieren.

Name:	Spielerstellung
Typ:	SICHER
Beschreibung:	Das Erstellen von ungültigen Spielkonfigurationen soll nicht ermöglicht werden. Dies garantiert einen flüssigen Spielfluss und frustriert den Nutzer nicht beim Erstellen eines Spieles

Name:	Intuitive Bedienung
Typ:	USE
Beschreibung:	Um eine hohe Benutzerfreundlichkeit zu erreichen, wird das fertige Produkt intuitiv bedienbar sein, so dass nur an wenigen Stellen Erklärungen (für erweiterte/spezielle Funktionen) nötig sind. Dies soll mithilfe einer Umfrage sichergestellt werden, welche mit einer Betaversion unter typischen Nutzern stattfindet.

Name:	Handliche Bedienung
Typ:	USE
Beschreibung:	Die Smartphone App soll mit nur einem Finger zu bedienen sein. Dies macht die Bedienung handlicher.

A GUI Skizzen

BEMERKUNG: *Es wird keine Garantie gegeben, dass die hier dargestellten GUI Skizzen dem Interface des fertigen Produktes entsprechen. Die Skizzen sind Möglichkeiten, wie die Interfaces aussehen können.*

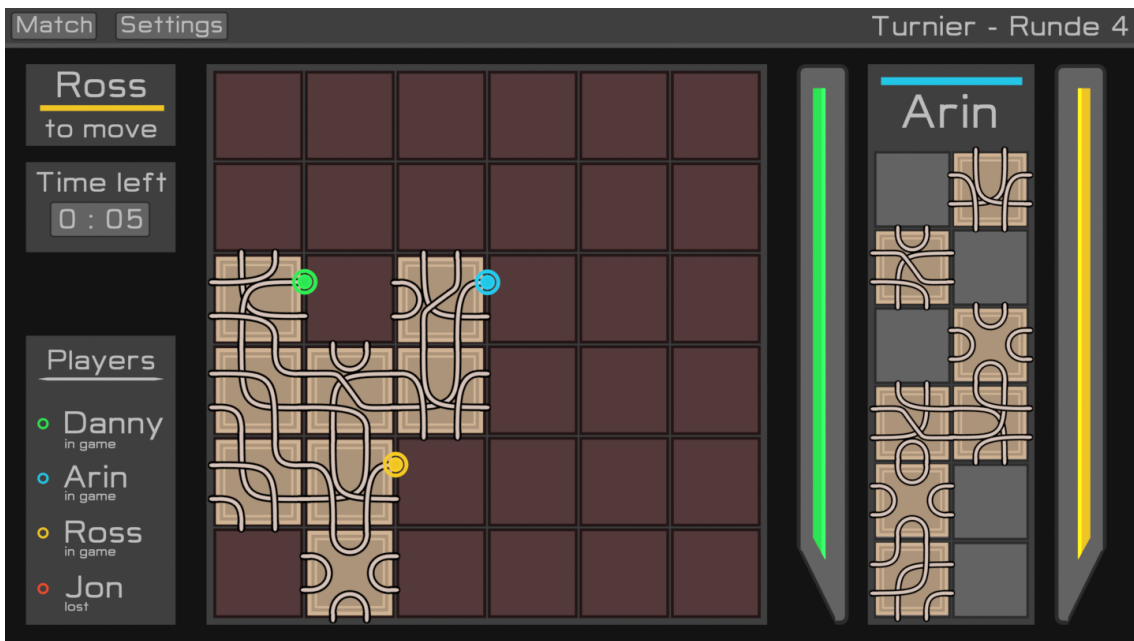


Abbildung 16: GUI Skizze PC Beobachter

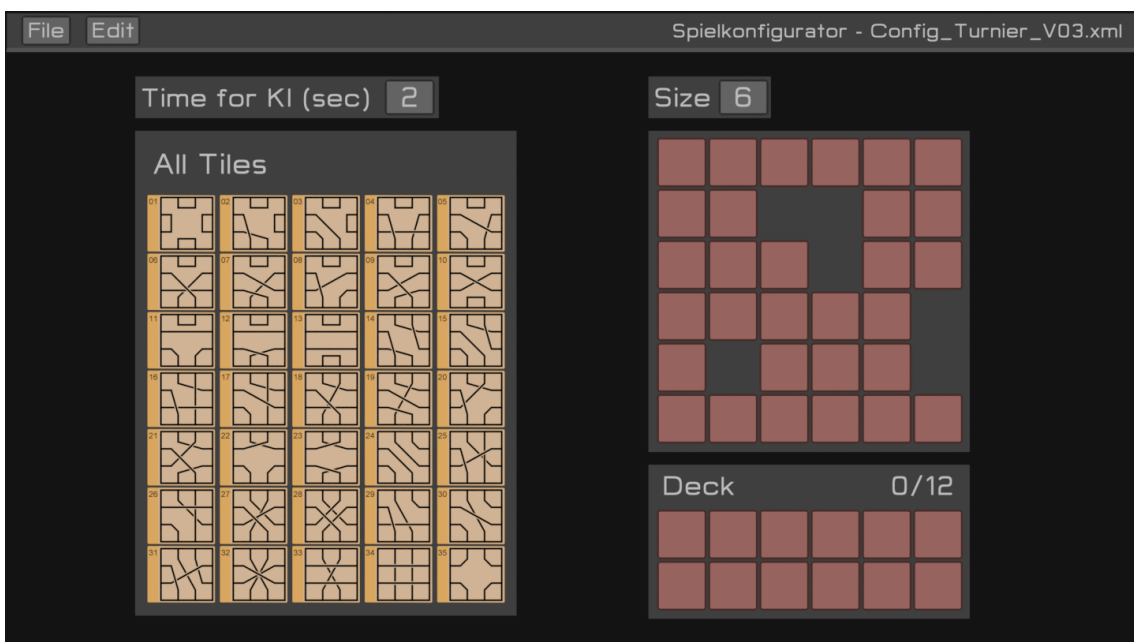


Abbildung 17: GUI Skizze Spielkonfigurator



Abbildung 18: GUI Skizze Smartphone Beobachter

Abbildungsverzeichnis

1	Modell des Problembereichs	6
2	Geschäftsprozess - Partie ausrichten	7
3	Geschäftsprozess - Spielzug durchführen	8
4	Geschäftsprozess - Spielkonfiguration erstellen	9
5	UML Use Case Diagramm - Übersicht	10
6	UML Use Case Diagramm - KI	11
7	UML Use Case Diagramm - Client	11
8	UML Use Case Diagramm - Engine	12
9	Aktivitätsdiagramm - Registrieren	14
10	Aktivitätsdiagramm - Wegfelder Organisieren	15
11	Aktivitätsdiagramm - Spielzug durchführen	17
12	Aktivitätsdiagramm - Partie spielen	19
13	Aktivitätsdiagramm - Partie ausrichten	20
14	Aktivitätsdiagramm - Turnier ausrichten	22
15	Aktivitätsdiagramm - Spielkonfiguration bearbeiten	24
16	GUI Skizze PC Beobachter	28
17	GUI Skizze Spielkonfigurator	28
18	GUI Skizze Smartphone Beobachter	29