Fachhochschule Nordwestschweiz

Projektarbeit IT-Projekt

Frühling 2017

Kamisado

The duel of the dragon towers

Gruppe Wind Chuckers

Lukas Weber

Cyrill Füglister

Lukas Kunz

Robin Lehner

Thomas Bosshard

Supervision by Lukas Frey

Responsible professor: Brad Richards

Olten, 10. März 2017

Inhaltsverzeichnis

1 Einleitung

Das Spiel Kamisado wurde im Jahr 2009 vom HUCH & friends-Verlag lanciert. Das abstrakte Brettspiel kann von zwei Personen gespielt werden.

Das Spielfeld ist ähnlich wie ein Schachbrett aufgebaut. Es gibt 8 Felder in der Breite und 8 Felder in der Tiefe. Insgesamt sind es 64 Felder, welche nach einem bestimmten Muster eingefärbt sind. Dieselben Farben werden auch für die Spielfiguren verwendet. Jeder Spieler ist im Besitz von 8 Türmen.

In einem Zug muss jeder Spieler eine seiner Spielfiguren entweder vorwärts oder diagonal ziehen, ohne dabei andere Figuren zu überspringen. Die Farbe, auf dem der Spielzug endet, bestimmt die Farbe der gegnerischen Spielfigur, mit welcher dieser seinen nächsten Zug machen muss. Ziel ist es die Grundlinie des Gegners zu erreichen.

Es ist auch möglich das Spiel über mehrere Runden zu führen. Dazu wird die Spielfigur, welche die Grundlinie des Gegners erreicht hat mit einem Drachenzahn versehen. Spielfiguren, welche einen Drachenzahn besitzen, sogenannte „Sumos“, verfügen über spezielle Fähigkeiten. Zum Beispiel können Sie den Gegner mit einem „Sumo-Stoss“ zurückdrängen. Sumo-Figuren bringen Punkte, wobei die Anzahl der Punkte von der Anzahl der Drachenzähne abhängt.

Das Spiel wird beendet, wenn ein Spieler die festgelegte Punktezahl erreicht.

1.1 Ziele

Es ist vorgesehen das Spiel virtuell spielbar zu machen mit einer Server-Client-Variante. Es soll dabei die Option geben das Spiel als Multiplayer oder Singleplayer zu starten.

1.2 Umsetzungs- und Projektziele

Umsetzungsziele	Projektziele
Multiplayer-Spiel möglich	- Auf dem Server können sich zwei Parteien einloggen via Client - Der Server wechselt die Spieler automatisch ab
Singleplayer-Spiel möglich	- Der Server hat eine Logik, welche selbst Spielzüge vornehmen kann - Der Server wechselt die Spieler automatisch ab
Etc.	-

Tabelle 1: Umsetzungs- und Projektziele

2 Stakeholder Analyse

2.1 Stakeholder in den Systemgrenzen

Die ermittelten Stakeholder stehen wie folgt in Beziehung mit dem System:

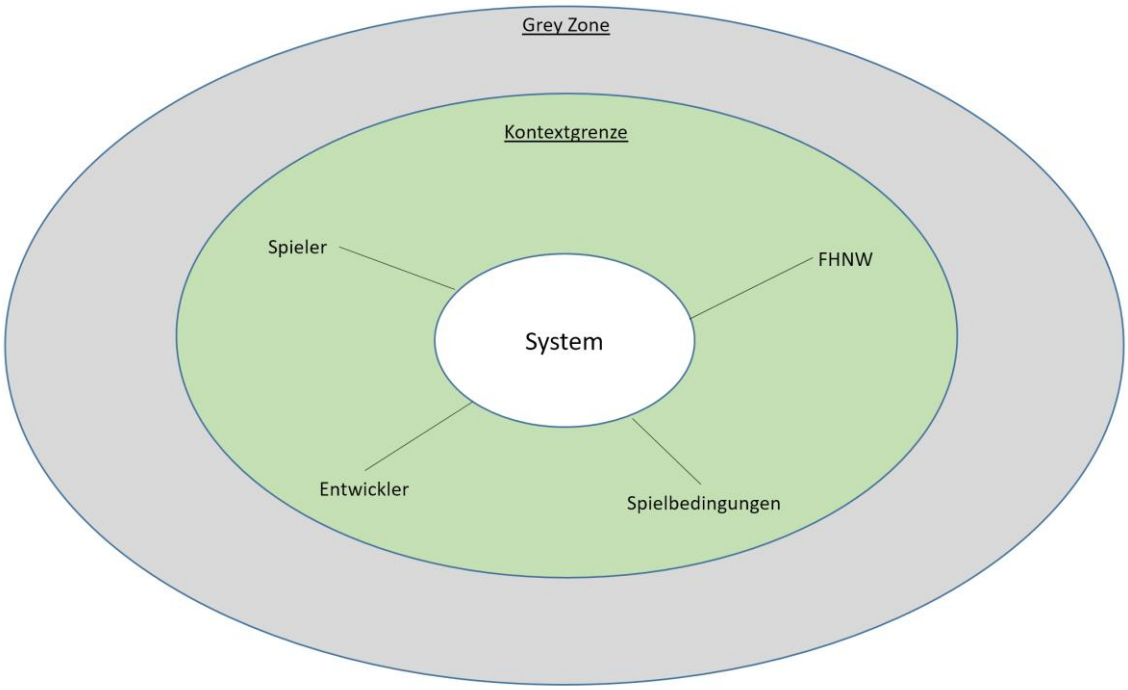


Abbildung 1: Stakeholder in der Systemgrenze

2.2 Stakeholder im Detail

Stakeholder	**Spieler**	
Position	Alle Spieler	
Rolle	Benutzer des Spiels über den Client	
Ziele	Spiel als Single-, oder als Multiplayer zu starten und zu spielen	
Wissen	Spielregeln	
Wichtigkeit	Hoch	Es ist der Endbenutzer. Der Spieler hat eine Vorstellung wie das System laufen sollte.
Einfluss	Niedrig	Hat auf die Entwicklung einen geringen Einfluss
Motivation	Hoch	Wird am Ende mit dem Spiel spielen

Tabelle 2: Stakeholdergruppe: Spieler

Stakeholder	Entwickler	
Position	Entwickelt das Spiel	
Rolle	Erstellt das spielbare Spiel	
Ziele	Möchte eine möglichst gute Applikation erstellen, welche im Minimum die Minimalanforderungen erfüllt, jedoch wäre es besser weitere Funktionen einzubauen	
Wissen	Weiss wie das Spiel entwickelt werden muss	
Wichtigkeit	Hoch	Kennt das Spiel und weiss wie es entwickelt werden muss
Einfluss	Hoch	Bestimmt welche Anforderungen umgesetzt werden
Motivation	Hoch	Will am Schluss eine gute Note erzielen und ein gutes, funktionierendes System abgeben

Tabelle 3: Stakeholdergruppe: Entwickler

Stakeholder	FHNW	
Position	Gibt den Auftrag für das IT-Projekt	
Rolle	Gibt Auftrag und bewertet Endergebnis	
Ziele	Will mindestens ein funktionierendes System, besser jedoch wenn noch mehr Funktionen eingebaut wurden	
Wissen	Weiss wie das Spiel aussehen soll und welche die Minimalforderungen sind	
Wichtigkeit	Hoch	Kennt das Spiel und erstellt die Benotung
Einfluss	Mittel	Gibt Leitplanken vor, jedoch liegt die Entwicklung komplett beim Entwickler
Motivation	Hoch	Wird am Schluss das Spiel spielen und bewerten

Tabelle 4: Stakeholdergruppe: FHNW

Stakeholder	Spielbedingungen	
Position	Die Regeln des Spiels	
Rolle	Gibt die einzuhaltenden Vorschriften bezüglich der Spielregeln vor	
Ziele	Spielregeln einhalten	
Wissen	Spielregeln	
Wichtigkeit	Mittel	Gibt die Spielregeln vor, auf den Rest der Umsetzung hat es keinen Einfluss
Einfluss	Hoch	Die Spielregeln sind dringend einzuhalten
Motivation	Niedrig	Die Spielregeln sollen eingehalten werden

Tabelle 5: Stakeholdergruppe: Spielbedingungen

2.3 Diagramme

2.3.1 Einteilung nach Wichtigkeit / Einfluss

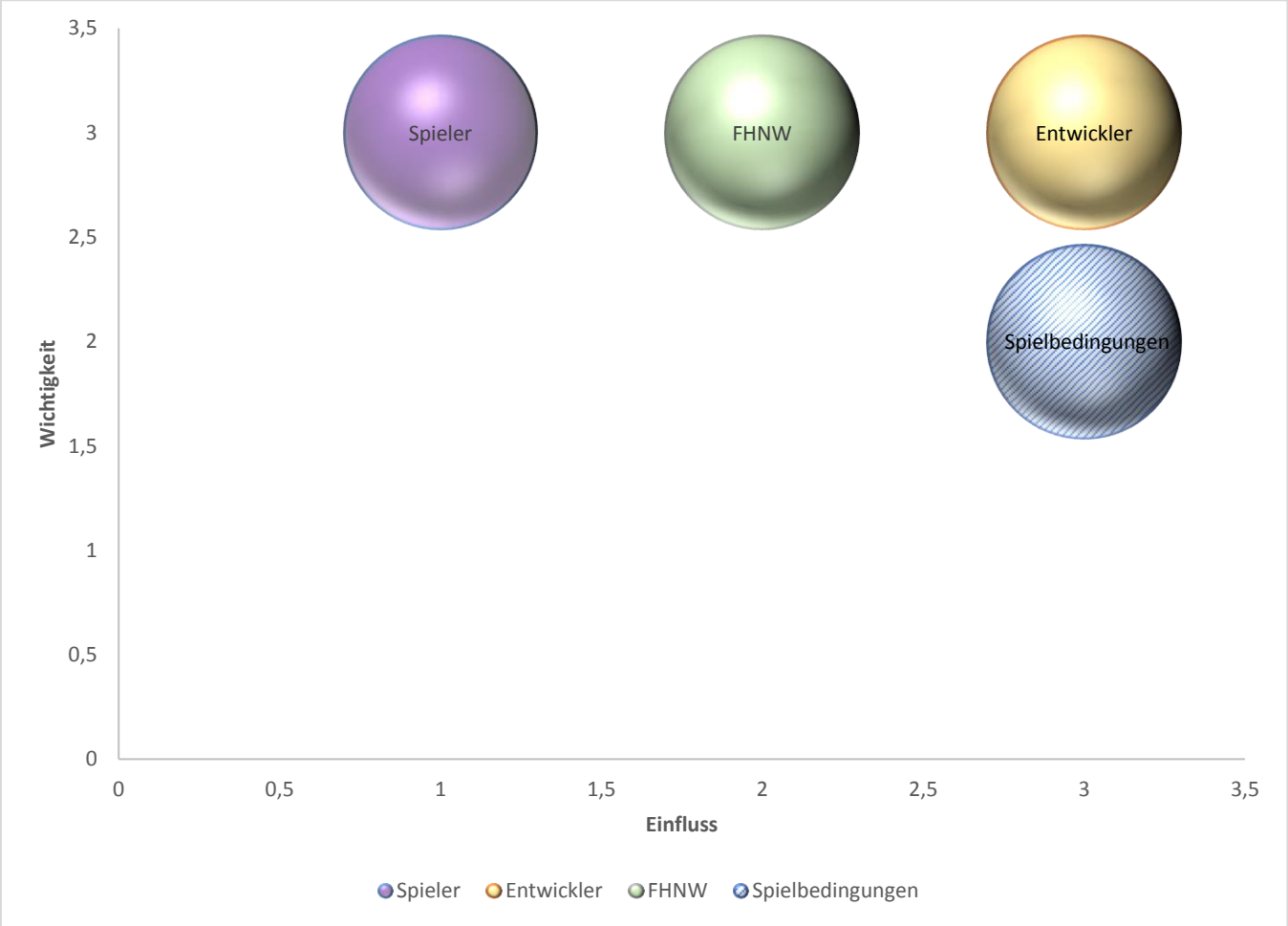


Abbildung 2: Stakeholder: Wichtigkeit und Einfluss

Die Y- Achse zeigt die Wichtigkeit eines Stakeholders. Die Wichtigkeit indexiert, wieviel der Stakeholder beitragen kann, um die Anforderungen an das System konkret zu spezifizieren. Die X- Achse zeigt den Einfluss des Stakeholders. Der Einfluss-Wert sagt aus, über wie viel Macht und Kompetenzen eine Person verfügt.

2.3.2 Einteilung nach Wichtigkeit / Motivation

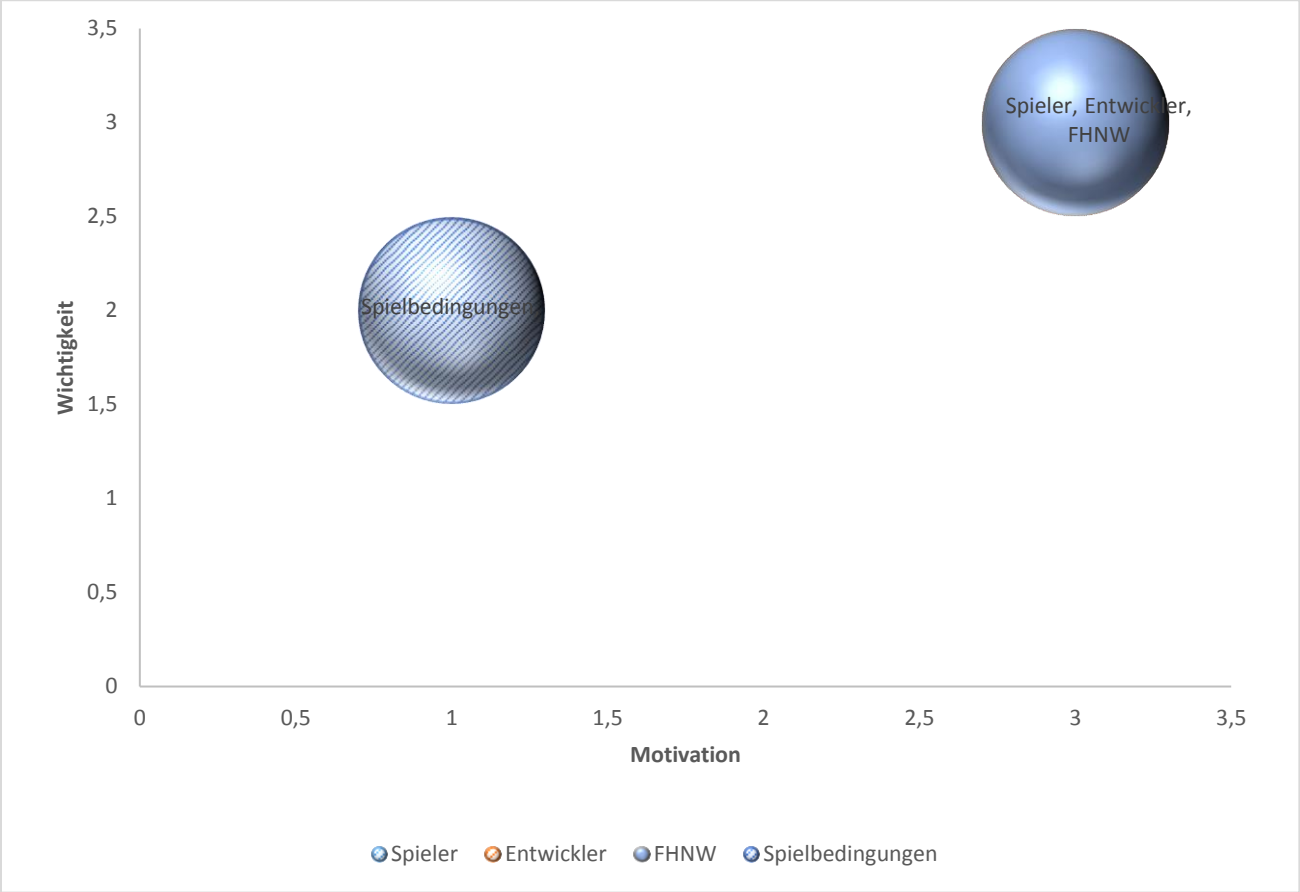


Abbildung 3: Stakeholder: Wichtigkeit und Motivation

Die Y- Achse zeigt die Wichtigkeit eines Stakeholders. Die Wichtigkeit indexiert, wieviel der Stakeholder beitragen kann, um die Anforderungen an das System konkret zu spezifizieren. Die X- Achse zeigt die Motivation des Stakeholders. Die Motivation unterstreicht das Interesse der unterschiedlichen Stakeholder für das Projekt.

3 Kontextanalyse

3.1 Systemabgrenzung

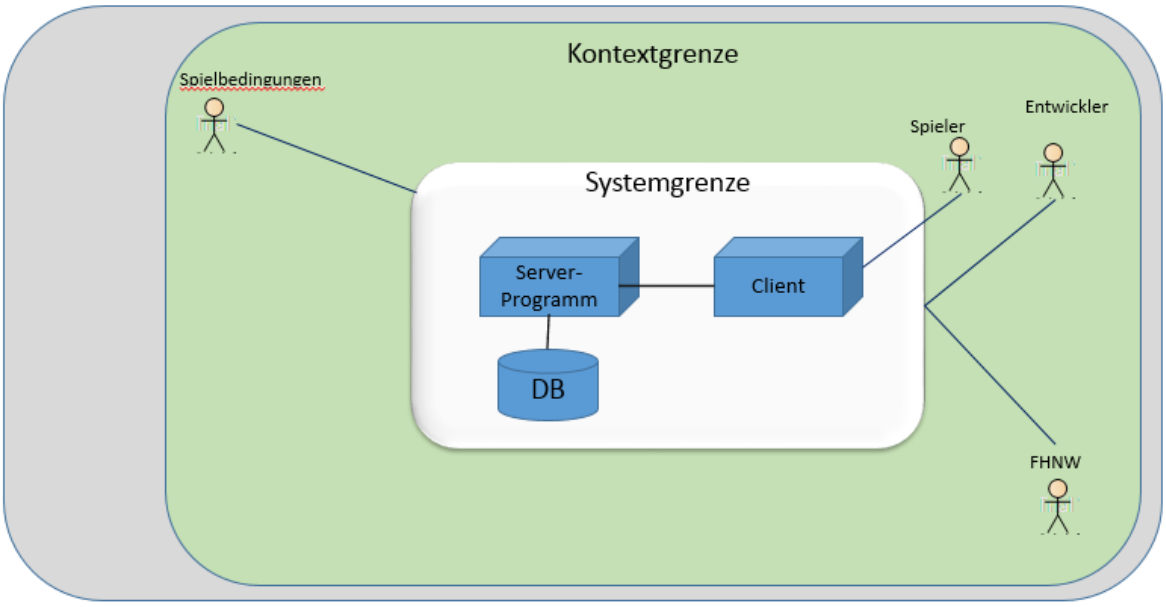


Abbildung 4: Systemabgrenzung

3.2 Systemgrenze

Das System besteht aus einem Server- und einem Client-Programm welches zur Benutzung zur Verfügung steht.

3.3 Kontextgrenze

Innerhalb der Kontextgrenze stehen die Stakeholder mit folgenden Tätigkeiten:

- Der Spieler beginnt das Spiel im Single- oder Multi-Player-Modus auf dem Client
- Der Entwickler entwickelt das System nach erhobenen Use Cases
- Die FHNW erhält das Spiel am Ende des Projektes und benotet dieses
- Die Spielbedingungen geben die Spielregeln vor, welche im Spiel implementiert sein müssen

3.4 Ausserhalb der Kontextgrenze

Ausserhalb der Kontextgrenze befindet sich kein Stakeholder.

4 Anforderungsermittlung

Um die Anforderungen korrekt ermitteln zu können, wurde hauptsächlich die Dokumentation der FHNW benutzt. Des Weiteren wurden zusätzliche Informationen aus dem Internet und von der offiziellen Spiel-Seite herbeigezogen um mehr Details zu erfahren. Grundsätzlich soll beim Projekt die „Lifecycle-Methode“ angewendet werden, da in der Grösse dieses Projektes eine iterative Methode wenig Sinn ergibt. Damit das Projekt am Schluss den Systemtest wie auch den Akzeptanztest problemlos besteht, soll das System von Zeit zu Zeit einer unbeteiligten Person gezeigt werden. Dadurch soll verifiziert werden, dass das Spiel leicht verständlich ist.

5 Use Case Modellierung

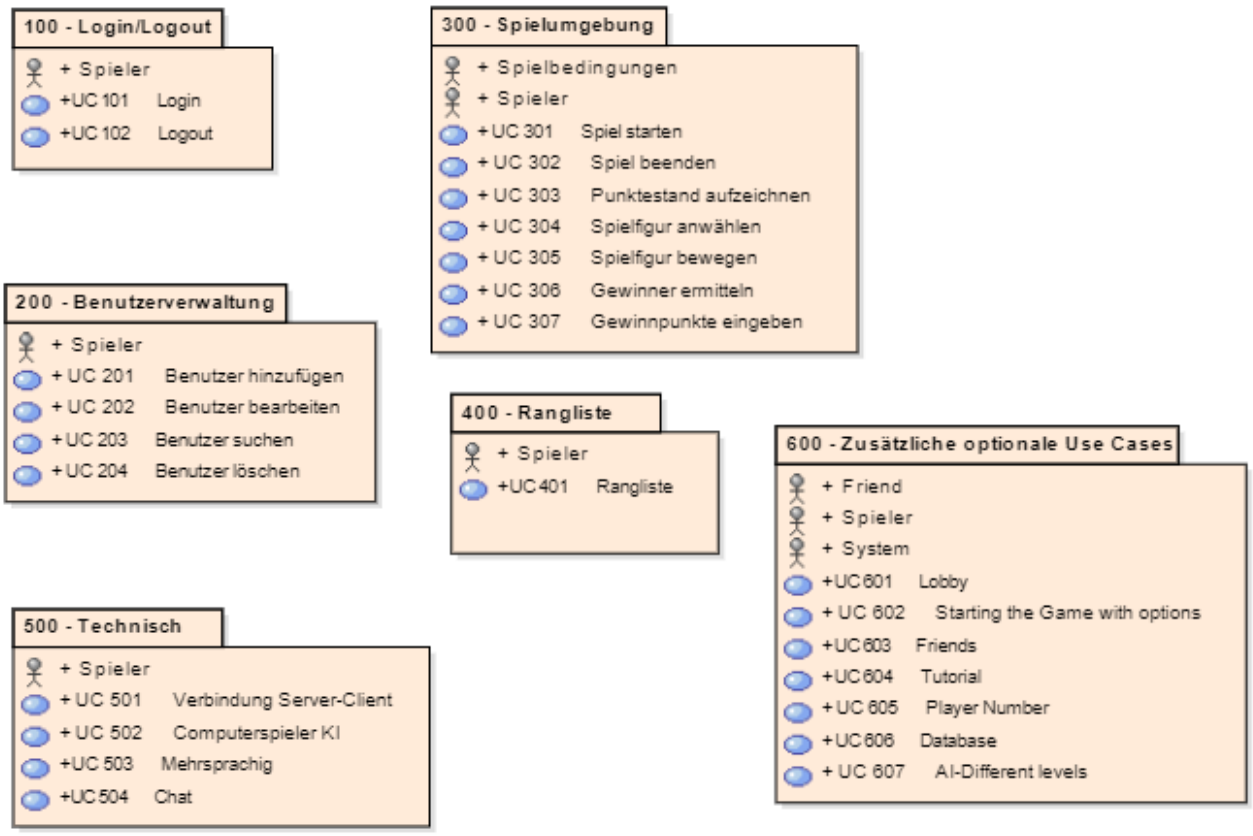


Abbildung 5: Use Case Modellierung Übersicht

5.1 100 - Login / Logout

UC Nr. 101	Login	optional	
Beschreibung	Das System muss dem Benutzer ermöglichen, seinen Benutzernamen und sein Passwort über die Tastatur auf der Benutzeroberfläche einzugeben und sich damit einzuloggen.		
Verantwortlicher	Spieler		
Beteiligte	-		
Auslöser	Der Benutzer loggt sich ein.		
Vorbedingungen	UC 201 -> Benutzer muss im System erfasst sein UC 501 -> Verbindung muss hergestellt sein		
Eingaben	Benutzername, Passwort		
Ablauf (Szenario)	**Standard**	**Alternativ**	
	1. Aufruf des Loginfensters		„Datei“ und dann „Ende“ anklicken. Ausstieg aus dem Use Case und zurück zur Startseite
	2. Benutzername eingeben	2.a.1 Hinweis: Benutzername falsch eingegeben	
	3. Passwort eingeben	3.a.1 Hinweis: Passwort falsch eingegeben	
	4. „Anmelden“ klicken		
	5. System überprüft eingaben		
	6. Eingaben korrekt	6.a.1 Eingaben nicht korrekt	
		6.a.2 Benutzer wird zur erneuten Eingabe aufgefordert	
	7. Benutzer angemeldet		
Ausgaben	Logdaten für das Einloggen werden geschrieben		
Nachbedingung	Der Benutzer ist eingeloggt		

Tabelle 6: UC NR 101: Login

UC Nr. 102	Logout	optional
Beschreibung	Das System muss dem Benutzer ermöglichen sich abzumelden	
Verantwortlicher	Spieler	
Beteiligte	-	
Auslöser	Der Benutzer loggt sich aus.	
Vorbedingungen	UC 101 -> Benutzer muss im System angemeldet sein UC 201 -> Benutzer muss im System erfasst sein UC 501 -> Verbindung muss hergestellt sein	
Eingaben	-	
Ablauf (Szenario)	**Standard**	**Alternativ**
	1. Im Main Menu über „Datei“ – „Beenden“ sich ausloggen	
Ausgaben	-	
Nachbedingung	Der Benutzer ist ausgeloggt	

Tabelle 7: UC Nr. 102: Logout

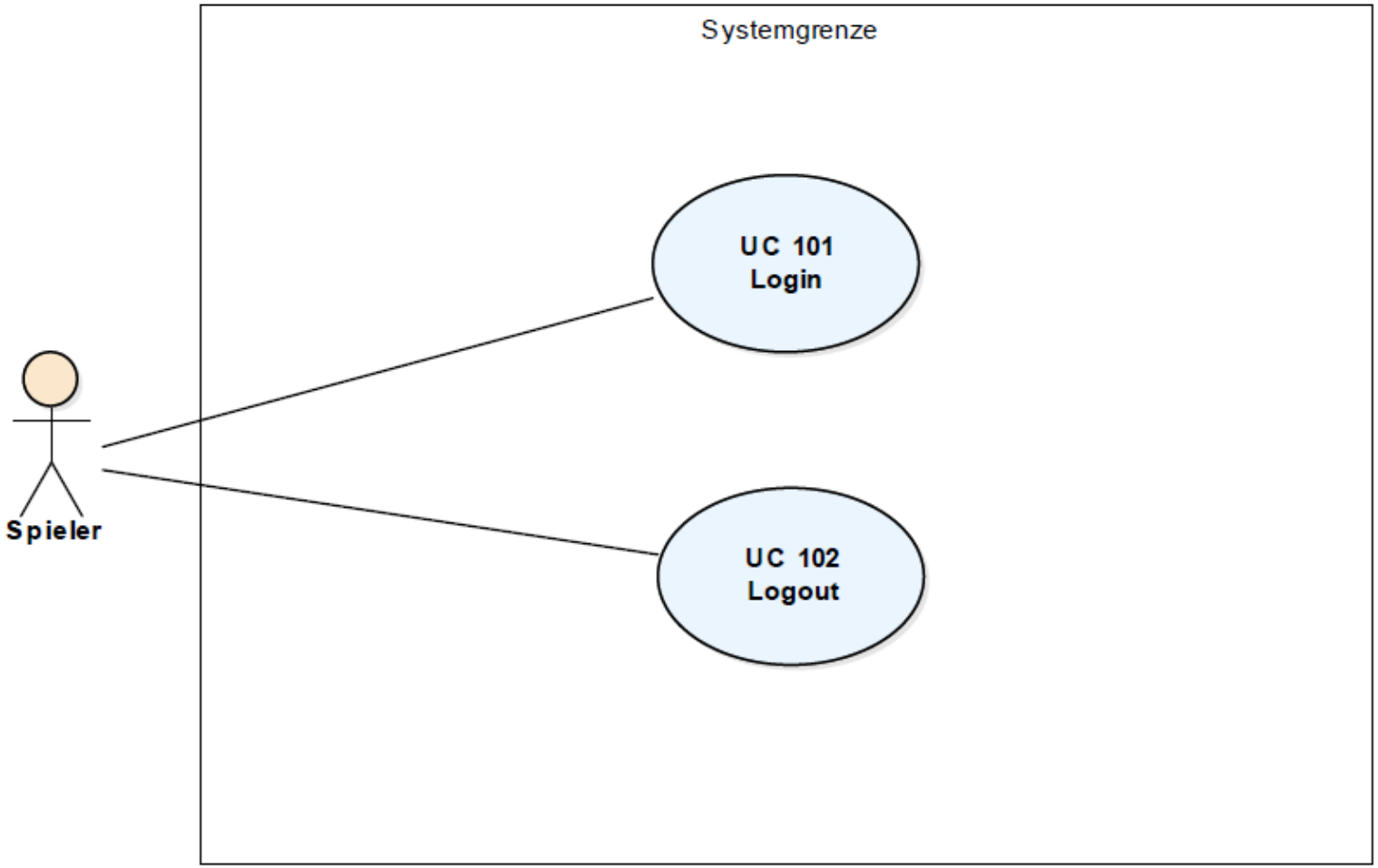


Abbildung 6: UC Diagramm 101 - 102

5.2 200 - Benutzerverwaltung

UC Nr. 201	**Benutzer hinzufügen**		**optional**
Beschreibung	Das System muss das Erstellen eines Benutzers über die Tastatur ermöglichen		
Verantwortlicher	Spieler		
Beteiligte	-		
Auslöser	Neuer Benutzer wird erstellt. Im System wird ein neuer Benutzer erfasst.		
Vorbedingungen	UC 501 -> Verbindung muss hergestellt sein		
Eingaben	• Id • Vorname • Nachname	• Points • SumoTower • Wins	
Ablauf (Szenario)	**Standard**	**Alternativ**	
	1. „Benutzer hinzufügen“ anklicken		„Datei“ und dann „Ende“ anklicken. Ausstieg aus dem Use Case und zurück zur Startseite
	2. Alle Benutzerdaten eingeben		
	3. „Benutzer erstellen“ anklicken		
	4. Eingaben korrekt und vollständig	4.a.1 Meldung „Eingaben unvollständig“ (alle Felder sind Pflichtfelder) / „Eingaben nicht korrekt“ (Vorname ist nicht ausgefüllt)	
		4.a.2 Eingaben vervollständigen / korrigieren. Weiter mit 4.	
	5. Benutzer erstellt	5.a.1 Meldung „Benutzer bereits erfasst“	
		5.a.1 Zurück zur Startseite	
Ausgaben	-		
Nachbedingung	Ein neuer Benutzer ist erstellt		

Tabelle 8: UC Nr. 201: Benutzer hinzufügen

UC Nr. 202	Benutzer bearbeiten	optional
Beschreibung	Das System muss dem Benutzer die Bearbeitung von Vornamen, Nachnamen über die Tastatur ermöglichen	
Verantwortlicher	Spieler	
Beteiligte	-	
Auslöser	Benutzer will die Benutzerdaten korrigieren / ändern.	
Vorbedingungen	UC 201 -> Der Benutzer, welcher die Änderungen vornimmt muss erfasst sein UC 101 -> Der Benutzer muss im System angemeldet sein UC 501 -> Verbindung muss hergestellt sein	
Eingaben	• Vorname • Nachname	
Ablauf (Szenario)	**Standard**	**Alternativ**
	1. Benutzer suchen <<includes Use Case 203>>	„Datei“ und dann „Ende“ anklicken. Ausstieg aus dem Use Case und zurück zur Startseite
	2. „Change User / create User“ anklicken	
	3. Personendaten ändern und altes Benutzerpasswort eingeben	
	4. „Ändern“ drücken	
	5. Eingaben korrekt und vollständig	5.a.1 Meldung „Eingaben unvollständig“ (alle Felder sind Pflichtfelder) / „Eingaben nicht korrekt“ (Vorname ist nicht ausgefüllt)
		5.a.2 Eingaben vervollständigen / korrigieren. Weiter mit 4.
	6. Benutzer bearbeitet	
Ausgaben	-	
Nachbedingung	Die aktualisierten Personendaten werden im System beim ausgewählten Benutzer gespeichert	

Tabelle 9: UC Nr. 202: Benutzer bearbeiten

UC Nr. 203	Benutzer suchen				optional
Beschreibung	Das System muss dem Benutzer die Suche von aktiven und inaktiven Spielern ermöglichen				
Verantwortlicher	Spieler				
Beteiligte	-				
Auslöser	Ein Benutzer soll gesucht werden.				
Vorbedingungen	UC 201 -> Der Benutzer muss im System erfasst sein UC 101 -> Der Benutzer muss im System angemeldet sein UC 501 -> Verbindung muss hergestellt sein				
Eingaben	• Id • Vorname • Nachname				
Ablauf (Szenario)	**Standard**		**Alternativ**		
	1. User Menu aufrufen				„Datei“ und dann „Ende“ anklicken. Ausstieg aus dem Use Case und zurück zur Startseite
	2. Eingabe der Personendaten zur Suche				
	3. „Suchen“ anklicken	3.a.1 Benutzer nicht gefunden			
		3.a.2 Eingaben korrigieren / vervollständigen weiter mit 4.	3.b.1 Benutzer nicht gefunden		
			3.b.2 UC 201 -> „Benutzer hinzufügen “ anklicken	3.c.1 Auftritt eines Verbindungsfehlers	
			3.b.3 Weiter mit UC 201	3.c.2 Das System erzeugt die Nachricht „Verbindung konnte nicht aufgebaut werden, bitte überprüfen Sie die Einstellungen und kontaktieren Sie den Systemadministrator.“	
	4. Benutzer gefunden				
	5. Benutzer und dessen Punkte werden angezeigt				
Ausgaben	-				
Nachbedingung	Der Benutzer wurde gefunden und der Account aufgerufen				

Tabelle 10: UC Nr. 203: Benutzer suchen

UC Nr. 204	Benutzer löschen		optional
Beschreibung	Das System muss dem Benutzer das Löschen von einem eindeutig definierten Spielern ermöglichen		
Verantwortlicher	Spieler		
Beteiligte	-		
Auslöser	Der Spieler spielt das Spiel nicht mehr und der Account soll gelöscht werden		
Vorbedingungen	UC 201 -> Der Benutzer, welcher die Löschung vornimmt und der Benutzer, welcher gelöscht werden soll, müssen beide erfasst sein UC 101 -> Der Benutzer muss im System angemeldet sein UC 501 -> Verbindung muss hergestellt sein		
Eingaben	Benutzername, Passwort		
Ablauf (Szenario)	**Standard**	**Alternativ**	
	1. Benutzer suchen <<includes Use Case 203>>		„Datei“ und dann „Ende“ anklicken. Ausstieg aus dem Use Case und zurück zur Startseite
	2. Altes Benutzerpasswort eingeben und „Löschen“ anklicken		
	3. Benutzer gelöscht		
Ausgaben			
Nachbedingung	Benutzer ist gelöscht.		

Tabelle 11: UC Nr. 204: Benutzer löschen

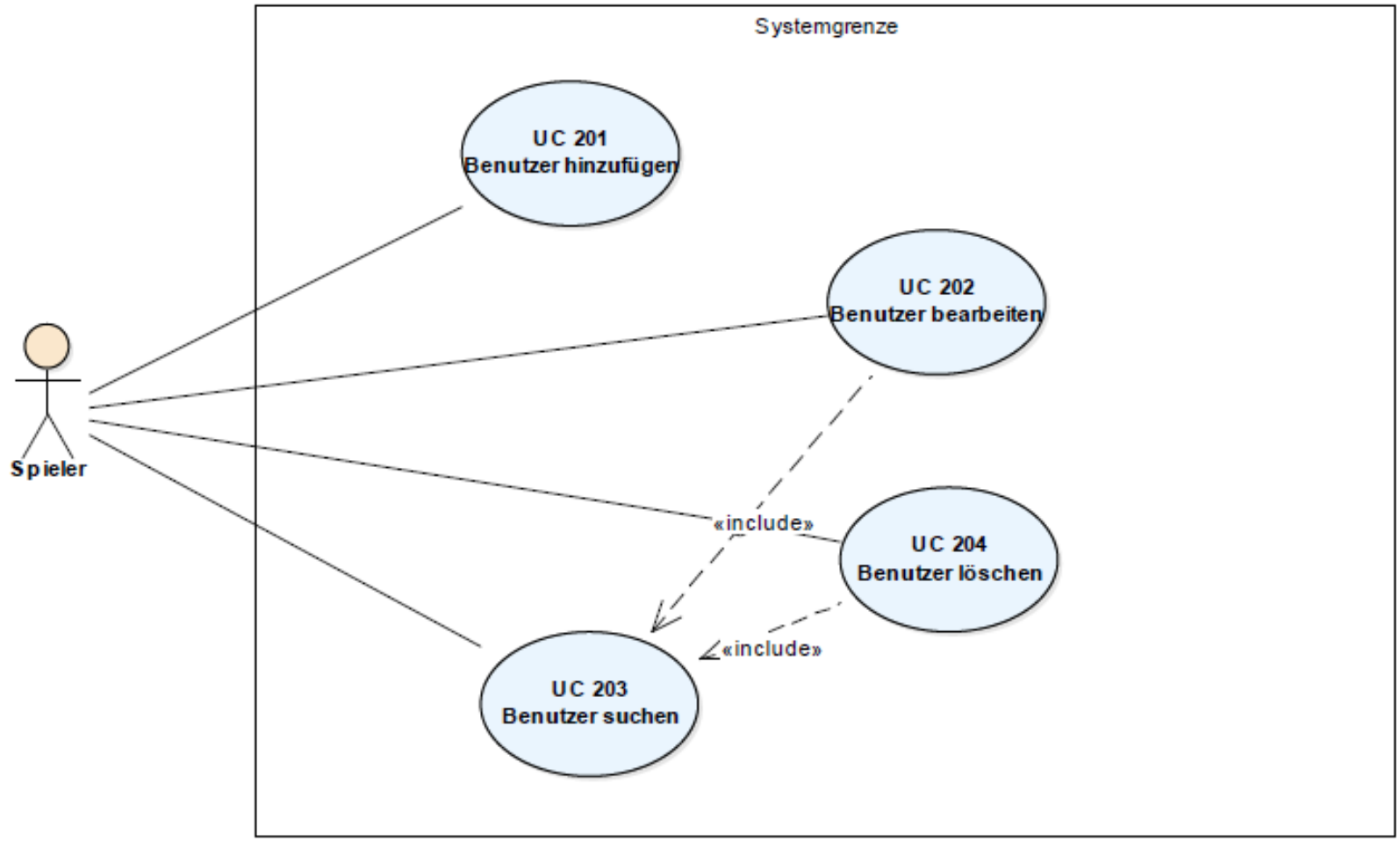


Abbildung 7: UC Diagramm: 201 - 204

5.3 300 - Spielumgebung

UC Nr. 301	Spiel starten		core
Beschreibung	Das System muss dem Spieler ermöglichen ein neues Spiel zu starten		
Verantwortlicher	Spieler		
Beteiligte			
Auslöser	Der Spieler erstellt ein neues Spiel.		
Vorbedingungen	UC 201 -> Benutzer muss im System erfasst sein UC 501 -> Verbindung muss hergestellt sein		
Eingaben	Name und Vorname für die Benutzersuche		
Ablauf (Szenario)	**Standard**	**Alternativ**	
	1. Einloggen im Login Menu	1.a.1 „Ohne Server“ spielen (als Gast)	„Datei“ und dann „Ende“ anklicken. Ausstieg aus dem Use Case und zurück zur Startseite
	2. Auswahl eines Spielmodus		
Ausgaben	-		
Nachbedingung	Das Spiel wurde gestartet		

Tabelle 12: UC Nr. 301: Spiel starten

UC Nr. 302	Spiel beenden	core
Beschreibung	Das System muss dem Spieler ermöglichen das Spiel zu beenden	
Verantwortlicher	Spieler	
Beteiligte		
Auslöser	Der Benutzer beendet das Spiel.	
Vorbedingungen	UC 201 -> Benutzer muss im System erfasst sein UC 301 -> Das Spiel muss gestartet sein UC 501 -> Verbindung muss hergestellt sein	
Eingaben		
	Standard	**Alternativ**
Ablauf (Szenario)	1. „Datei“ und dann „Ende“ anklicken.	
Ausgaben	-	
Nachbedingung	Das Spiel wurde beendet	

Tabelle 13: UC Nr. 302: Spiel beenden

UC Nr. 303	Punktestand aufzeichnen	optional
Beschreibung	Das System muss das Aufzeichnen der erzielten Punkte ermöglichen.	
Verantwortlicher	Spieler	
Beteiligte		
Auslöser	Der Benutzer hat das Spiel erfolgreich beendet	
Vorbedingungen	UC 201 -> Benutzer muss im System erfasst sein UC 301 -> Das Spiel muss gestartet sein UC 501 -> Verbindung muss hergestellt sein	
Eingaben		
Ablauf (Szenario)	**Standard**	**Alternativ**
	1. System schreibt nach Beendigung eines Spiels den Punktestand nieder	
Ausgaben	Punktestand	
Nachbedingung	Der Spieler kann den neuen Punktestand im User Menu abrufen	

Tabelle 14UC Nr. 303: Punktestand aufzeichnen

UC Nr. 304	Spielfigur anwählen	core
Beschreibung	Das System muss dem Spieler ermöglichen die Spielfigur im Spiel auszuwählen	
Verantwortlicher	Spieler	
Beteiligte		
Auslöser	Der Benutzer wählt die Figur aus.	
Vorbedingungen	UC 201 -> Benutzer muss im System erfasst sein UC 301 -> Das Spiel muss gestartet sein	
Eingaben		
Ablauf (Szenario)	**Standard**	**Alternativ**
	1. „Spielfigur“ anklicken.	„Datei“ und dann „Ende“ anklicken. Ausstieg aus dem Use Case und zurück zur Startseite
	2. Die Spielfigur ist ausgewählt	2.a.1 Die Spielfigur kann nicht ausgewählt werden
		2.a.2 Kontaktieren des Supports
Ausgaben	Angewählte Spielfigur	
Nachbedingung	Ein Spielzug wird gemacht	

Tabelle 15 UC Nr. 304: Spielfigur anwählen

UC Nr. 305	Spielfigur bewegen		core
Beschreibung	Das System muss dem Spieler ermöglichen die Spielfigur im Spiel zu bewegen		
Verantwortlicher	Spieler		
Beteiligte			
Auslöser	Der Benutzer hat die Figur angewählt und klickt auf ein freies Feld.		
Vorbedingungen	UC 201 -> Benutzer muss im System erfasst sein UC 301 -> Das Spiel muss gestartet sein UC 304 -> Die Figur muss angewählt sein UC 501 -> Verbindung muss hergestellt sein		
Eingaben			
Ablauf (Szenario)	**Standard**	**Alternativ**	
	1. „Freies Feld“ anklicken.		„Datei“ und dann „Ende“ anklicken. Ausstieg aus dem Use Case und zurück zur Startseite
	2. Die Spielfigur verschiebt sich auf das angeklickte Feld	2.a.1 Die Spielfigur verschiebt sich nicht auf das angeklickte Feld	
		2.a.2 Kontaktieren des Supports	
Ausgaben	Neuer Standort der Spielfigur		
Nachbedingung	Der andere Spieler ist am Zug		

Tabelle 16 UC Nr. 305: Spielfigur bewegen

UC Nr. 306	Gewinner ermitteln	core
Beschreibung	Das System muss den Gewinner ermitteln	
Verantwortlicher	Spieler	
Beteiligte		
Auslöser	Der festgelegte Punktestand ist erreicht.	
Vorbedingungen	UC 201 -> Benutzer muss im System erfasst sein UC 301 -> Das Spiel muss gestartet sein UC 307 -> Punktestand eingeben UC 501 -> Verbindung muss hergestellt sein	
Eingaben		
Ablauf (Szenario)	**Standard**	**Alternativ**
	1. Gewinner wird ausgegeben nachdem die Punktezahl von einem Spieler erreicht wurde	1.a.1 Der Gewinner wird nicht ausgegeben
	2. Das Spiel terminiert -> UC 302	2.a.1 Kontaktieren des Supports 2.a.2 Neuer Punktestand eingeben -> UC 307
Ausgaben	Gewinner	
Nachbedingung	Das Spiel ist beendet oder wird mit dem neuen Punktestand weitergeführt	

Tabelle 17 UC Nr. 306: Gewinner ermitteln

UC Nr. 307	Gewinnpunkte eingeben	core
Beschreibung	Das System muss dem Spieler ermöglichen auf eine unterschiedliche Anzahl an Gewinnpunkten zu spielen	
Verantwortlicher	Spieler	
Beteiligte		
Auslöser	Spielmodus wird ausgewählt oder Spiel gegen Freund gestartet.	
Vorbedingungen	UC 101 -> Login UC 201 -> Benutzer muss im System erfasst sein UC 501 -> Verbindung muss hergestellt sein	
Eingaben		
Ablauf (Szenario)	**Standard**	**Alternativ**
	1. Gewinn-Punkte werden je nach Spielmodus gesetzt (vordefinierte Standards)	1.a.1 Die Gewinnpunkte können manuell eingegeben werden bei einem Spiel gegen einen Freund
	2. Das Spiel startet -> UC 301	2.a.1 Kontaktieren des Supports
Ausgaben	Logging, dass Verbindung hergestellt wurde	
Nachbedingung	Das Spiel kann gestartet werden, weiter mit UC 301	

Tabelle 18 UC Nr. 307: Gewinnpunkte eingeben

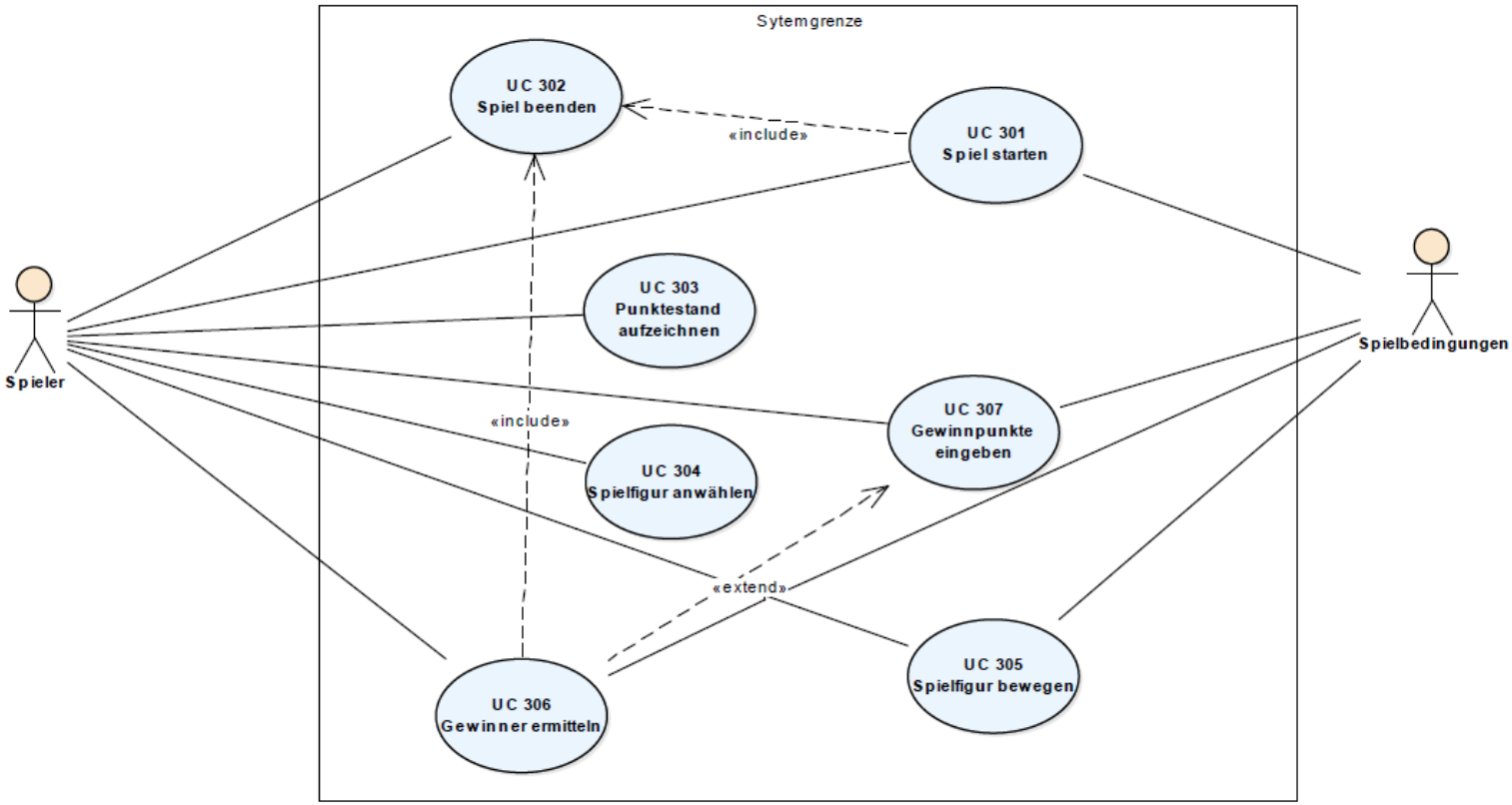


Abbildung 8: UC Diagramm: 301 - 307

5.4 400 - Rangliste

UC Nr. 401	Rangliste	optional
Beschreibung	Das System soll dem Spieler ermöglichen eine Punkteübersicht der Spieler zu haben.	
Verantwortlicher	Spieler	
Beteiligte	-	
Auslöser	Der Benutzer möchte die Punkte der Benutzer sehen.	
Vorbedingungen	UC 101 -> Login UC 201 -> Benutzer muss im System erfasst sein UC 501 -> Verbindung muss hergestellt sein	
Eingaben	Erstellung eines Reports	
Ablauf (Szenario)	**Standard**	**Alternativ**
	1. Benutzer suchen <<includes Use Case 203>>	„Datei“ und dann „Ende“ anklicken. Ausstieg aus dem Use Case und zurück zur Startseite
	2. Das System gibt die Punktezahl des gesuchten Benutzers aus	
Ausgaben	Punkte	
Nachbedingung	Die Anzahl Punkte ist ausgegeben.	

Tabelle 19: UC Nr. 401: Rangliste

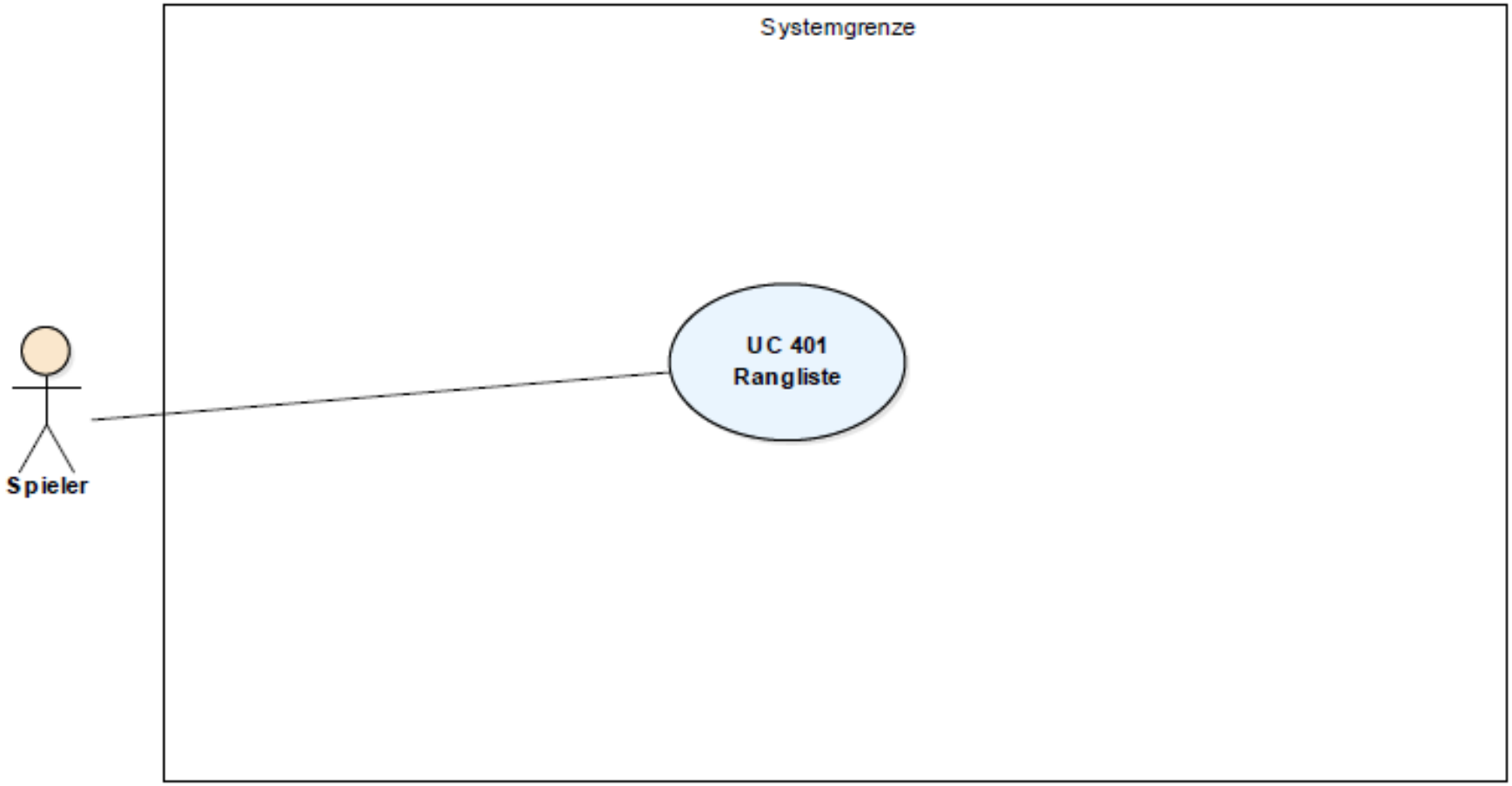


Abbildung 9: UC Diagramm 401

5.5 500 - Technisch

UC Nr. 501	Verbindung Server-Client	core
Beschreibung	Das System muss dem Spiel ermöglichen eine Verbindung vom Client zum Server herzustellen	
Verantwortlicher	Spieler	
Beteiligte		
Auslöser	Der Benutzer will den Client mit dem Server verbinden.	
Vorbedingungen		
Eingaben	Netzwerkadresse, Port	
Ablauf (Szenario)	**Standard**	**Alternativ**
	1. „Verbinden“ anklicken	„Datei“ und dann „Ende“ anklicken. Ausstieg aus dem Use Case und zurück zur Startseite
	2. Der Client verbindet sich über das Netzwerk zum Server	2.a.1 Der Client kann sich nicht mit dem Server verbinden
		2.a.2 Erneut mit anderer Adresse versuchen
Ausgaben	-	
Nachbedingung	Client ist mit Server verbunden	

Tabelle 20 UC Nr. 501: Verbindung Server-Client

UC Nr. 502	Computerspieler (KI)	optional
Beschreibung	Das System muss dem Spiel ermöglichen mittels programmierter Logik selbständig Spielzüge zu berechnen und vorzunehmen	
Verantwortlicher	Spieler	
Beteiligte		
Auslöser	Der Benutzer will ein Single-Player-Spiel starten.	
Vorbedingungen	UC 101 -> Login UC 201 -> Benutzer muss im System erfasst sein	
Eingaben		
Ablauf (Szenario)	**Standard**	**Alternativ**
	1. Einloggen und Spielmodi auswählen.	„Datei“ und dann „Ende“ anklicken. Ausstieg aus dem Use Case und zurück zur Startseite
	2. Weiter mit UC 301, mit Computer-Gegner	
Ausgaben		
Nachbedingung	Spiel wird gegen den Computer gespielt	

Tabelle 21 UC Nr. 502: Computerspieler (KI)

UC Nr. 503	Mehrsprachig	optional
Beschreibung	Das System muss dem Benutzer ermöglichen die Sprache des Spiels zu ändern	
Verantwortlicher	Spieler	
Beteiligte		
Auslöser	Der Benutzer will eine andere Sprache auswählen.	
Vorbedingungen		
Eingaben	Gewünschte Sprache	
Ablauf (Szenario)	**Standard**	**Alternativ**
	1. Im Spiel „Datei“ und dann „Sprache“ anklicken	
	2. Gewünschte Sprache auswählen	2.a.1 Die Sprache kann nicht geändert werden
		2.a.2 Kontaktieren des Supports
	3. Sprache ist geändert	
Ausgaben	Text erscheint in gewünschter Sprache	
Nachbedingung	Sprache geändert	

Tabelle 22 UC Nr. 503: Mehrsprachig

UC Nr. 504	Chat	optional
Beschreibung	Das System muss dem Benutzer ermöglichen mit dem anderen Spieler via Chat zu kommunizieren	
Verantwortlicher	Spieler	
Beteiligte		
Auslöser	Der Benutzer will mit dem anderen Spieler kommunizieren.	
Vorbedingungen	UC 101 -> Login UC 201 -> Benutzer muss im System erfasst sein UC 501 -> Verbindung muss hergestellt sein	
Eingaben		
Ablauf (Szenario)	**Standard**	**Alternativ**
	1. Den gewünschten Text im Text-Feld eingeben	
	2. „Absenden“ drücken	
	3. Der andere Benutzer kann den Text sehen und antworten	3.a.1 Der Text kann nicht übermittelt werden
		3.a.2 Kontaktieren des Supports
Ausgaben	Der Übermittelte Text wird auf dem anderen Client angezeigt	
Nachbedingung	Antwort schreiben	

Tabelle 23 UC Nr. 504: Chat

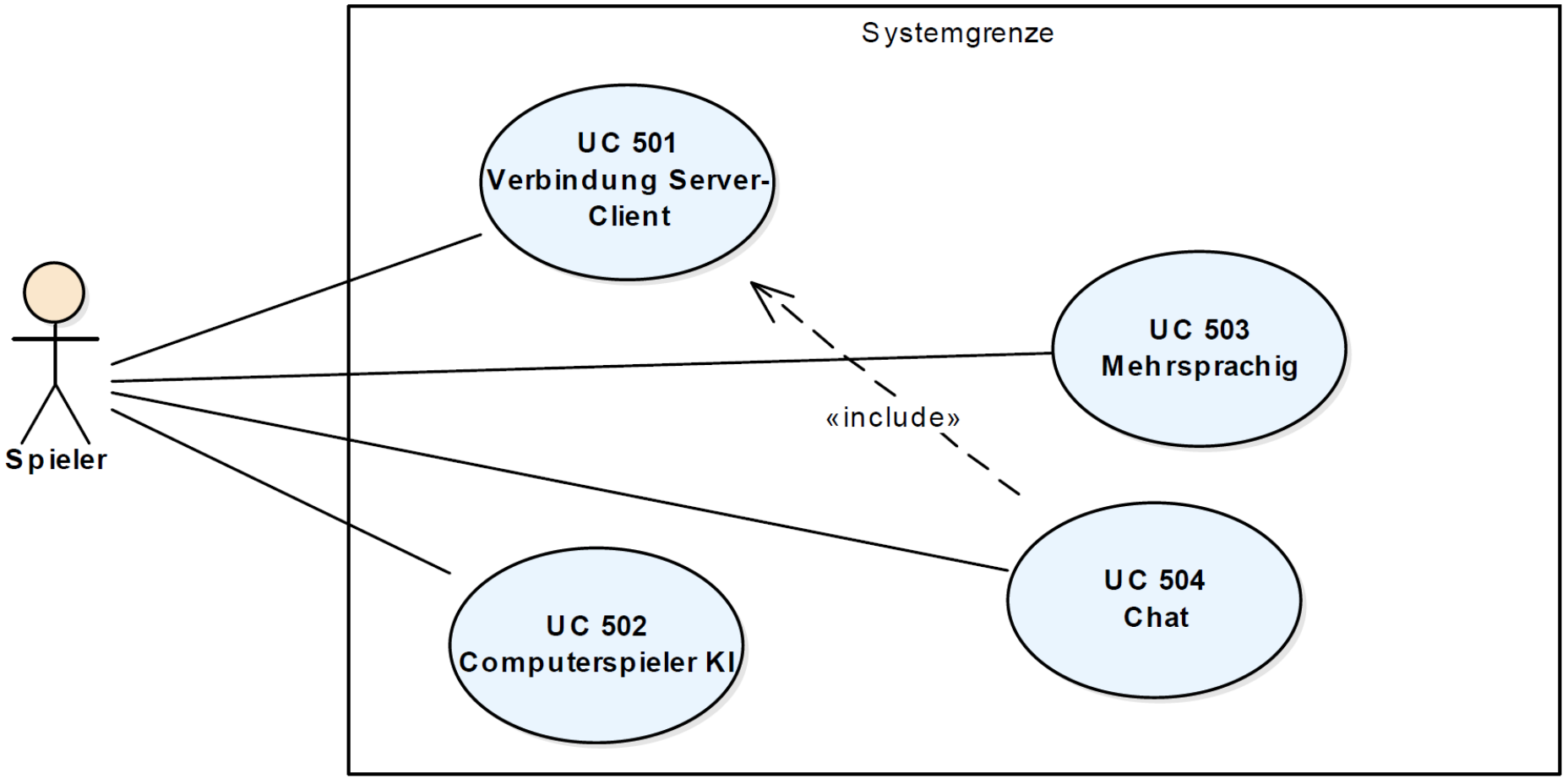


Abbildung 10 UC Diagramm 501 - 504

5.6 600 - Zusätzliche optionale Use Cases

UC Nr. 601	**Lobby**		**optional**
Beschreibung	Das System muss dem Benutzer ermöglichen, mit einem Freund direkt vor dem Spiel zu chatten, ohne dass dies die anderen Benutzer sehen und seine Bereitschaft zum Spielen anzumelden		
Verantwortlicher	Spieler		
Beteiligte	-		
Auslöser	Der Benutzer möchte ein Freunde-Spiel starten		
Vorbedingungen	UC 201 -> Benutzer muss im System erfasst sein UC 501 -> Verbindung muss hergestellt sein		
Eingaben	Spiel starten		
Ablauf (Szenario)	**Standard**	**Alternativ**	
	1. Aufruf der Lobby		„Datei“ und dann „Ende“ anklicken. Ausstieg aus dem Use Case und zurück zur Startseite
	2. Den Button „Go“ drücken		
	3. Das Spiel mit dem Freund startet	3.a.1 Das Spiel startet nicht	
		3.a.2 Den Button nochmals abwählen und erneut drücken	
		6.a.1 Eingaben nicht korrekt	
		6.a.2 Benutzer wird zur erneuten Eingabe aufgefordert	
Ausgaben	Log Daten für das Freunde-Spiel werden geschrieben		
Nachbedingung	Das Freunde Spiel wurde gestartet		

Tabelle 24: UC Nr. 601: Lobby

UC Nr. 602	Starting the Game with options	optional
Beschreibung	Das System muss dem Benutzer ermöglichen, das Spiel mit verschiedenen Modi zu starten	
Verantwortlicher	Spieler	
Beteiligte	-	
Auslöser	Der Benutzer startet ein Spiel	
Vorbedingungen	UC 201 -> Benutzer muss im System erfasst sein UC 501 -> Verbindung muss hergestellt sein / Bei dem Modus „Ohne Server“ trifft diese Vorbedingung nicht zu	
Eingaben	Spiel starten	
Ablauf (Szenario)	**Standard**	**Alternativ**
	1. Aufruf des Haupt-Menus	„Datei“ und dann „Ende“ anklicken. Ausstieg aus dem Use Case und zurück zur Startseite
	2. Spiel-Variante auswählen und den entsprechenden Button drücken	
	3. Spiel wird gestartet	3.a.1 Das Spiel wird nicht gestartet
		3.a.2 Kontaktieren Sie den Administrator
Ausgaben	Logdaten für das starten des Spiels werden geschrieben	
Nachbedingung		

Tabelle 25: UC Nr. 602: Starting the Game with options

UC Nr. 603	Friends	optional
Beschreibung	Das System muss dem Benutzer ermöglichen, seine Freunde zu verwalten. Dies bedeutet Freunde hinzufügen, diese löschen, Anfragen zu versenden und nachzuschauen wer alles im Spiel registriert ist.	
Verantwortlicher	Spieler	
Beteiligte	-	
Auslöser	Der Benutzer verwaltet seine Freunde	
Vorbedingungen	UC 201 -> Benutzer muss im System erfasst sein UC 501 -> Verbindung muss hergestellt sein	
Eingaben	1. Add Friends -> hinzufügen eines Freundes 2. Name in Add Friends eingeben -> Freund suchen 3. Remove drücken - >Freund entfernen 4. Play drücken -> Freunde-Spiel starten 5. Accept drücken -> Anfrage annehmen 6. Refuse drücken -> Anfrage abweisen	
Ablauf (Szenario)	**Standard**	**Alternativ**
	1. Aufruf des Freunde-Menus	„Datei“ und dann „Ende“ anklicken. Ausstieg aus dem Use Case und zurück zur Startseite
	2. Die entsprechende Eingabe tätigen – siehe Eingabe	
	3. Die Aktion wie in Eingabe beschrieben wird ausgelöst.	3.a.1 Die Aktion wird nicht ausgelöst
		3.a.2 Kontaktieren Sie den Administrator
Ausgaben	Logdaten für das Einloggen werden geschrieben	
Nachbedingung	Freunde wurden verwaltet	

Tabelle 26: UC Nr. 603: Friends

UC Nr. 604	Tutorial	optional
Beschreibung	Das System muss dem Benutzer ermöglichen, sich über die möglichen Spiele zu orientieren	
Verantwortlicher	Spieler	
Beteiligte	-	
Auslöser	Der Benutzer informiert sich über die Spiel-Möglichkeiten	
Vorbedingungen	UC 201 -> Benutzer muss im System erfasst sein UC 501 -> Verbindung muss hergestellt sein	
Eingaben	Help >> Tutorial drücken	
Ablauf (Szenario)	**Standard**	**Alternativ**
	1. Aufruf des Tutorial Grundfensters im Spiel unter „Hilfe“ – „Einführung“	„Datei“ und dann „Ende“ anklicken. Ausstieg aus dem Use Case und zurück zur Startseite
	2. Basic Rules auswählen	2.a.1 Long Play auswählen 2.b.1 Marathon Play auswählen
	3. Basic Rules werden erklärt	2.a.2Long Play Rules werden erklärt 2.b.2 Marathon Play Rules werden erklärt
	4. „Ok“ anklicken	2.a.3 „Ok“ anklicken 2.b.3 „Ok“ anklicken
	5. Zurück kehren ins Spiel	2.a.4. Zurück kehren ins Spiel 2.b.4 Zurück kehren ins Spiel
Ausgaben	Logdaten für das Tutorial werden geschrieben	
Nachbedingung	Der Benutzer hat das Tutorial gelesen	

Tabelle 27: UC Nr. 604: Tutorial

UC Nr. 605	Player-Number	optional
Beschreibung	Das System muss dem Benutzer anzeigen, welcher Spieler er ist	
Verantwortlicher	Spieler	
Beteiligte	-	
Auslöser	Der Benutzer startet ein Spiel	
Vorbedingungen	UC 201 -> Benutzer muss im System erfasst sein UC 501 -> Verbindung muss hergestellt sein	
Eingaben	Spiel-Start	
Ablauf (Szenario)	**Standard**	**Alternativ**
	1. Aufruf der Spieloberfläche	„Datei“ und dann „Ende“ anklicken. Ausstieg aus dem Use Case und zurück zur Startseite
	2. Auf der rechten Seite des Spielbretts wird angezeigt welcher Spieler der User ist	2.a.1 Es wird nicht angezeigt welcher Spieler der User ist
		2.a.2 Das Spiel erneut starten
Ausgaben	Anzeige welcher Spieler der Benutzer ist	
Nachbedingung	-	

Tabelle 28: UC Nr. 605: Player-Number

UC Nr. 606	Database	optional
Beschreibung	Das System muss ermöglichen, die Daten ordnungsgemäss zu verwalten.	
Verantwortlicher	Programm	
Beteiligte	-	
Auslöser	Benutzerverwaltung, Freundeverwaltung, Spiel gewonnen	
Vorbedingungen	UC 201 -> Benutzer muss im System erfasst sein UC 501 -> Verbindung muss hergestellt sein	
Eingaben	-	
Ablauf (Szenario)	**Standard**	**Alternativ**
	1. Die Daten werden richtig in der Datenbank gespeichert	
	2. Die Daten können wieder abgerufen werden	
Ausgaben	Logdaten für das abrufen oder speichern der Daten in der Datenbank werden geschrieben	
Nachbedingung	Daten sind persistent gespeichert	

Tabelle 29: UC Nr. 606: Database

UC Nr. 607	AI-Different levels	optional
Beschreibung	Das System muss dem Benutzer ermöglichen, den KI-Gegner in verschiedenen stärken einzustellen	
Verantwortlicher	Spieler	
Beteiligte	-	
Auslöser	Der Benutzer startet ein AI-Game	
Vorbedingungen	UC 201 -> Benutzer muss im System erfasst sein UC 501 -> Verbindung muss hergestellt sein	
Eingaben	Spiel start	
Ablauf (Szenario)	Standard	Alternativ
	1. Aufruf des Haupt-Menus	„Datei“ und dann „Ende“ anklicken. Ausstieg aus dem Use Case und zurück zur Startseite
	2. Auswahl vom Gamemodi	
	3. AI Controller auswählen	
	4. Player two, resp. Player one einstellen	
	5. Single Player Button resp. Double AI Button anklicken	
	6. Die stärke des AI- Gegners wurde angepasst	6.a.1 Die Stärke des AI-Gegners konnte nicht angepasst werden
		6.a.2 Verbindung zum Server überprüfen
Ausgaben	Logdaten für den AI-Gegner werden geschrieben	
Nachbedingung	Der Benutzer spielt gegen den AI-Gegner	

Tabelle 30: UC Nr. 607: AI-Different levels

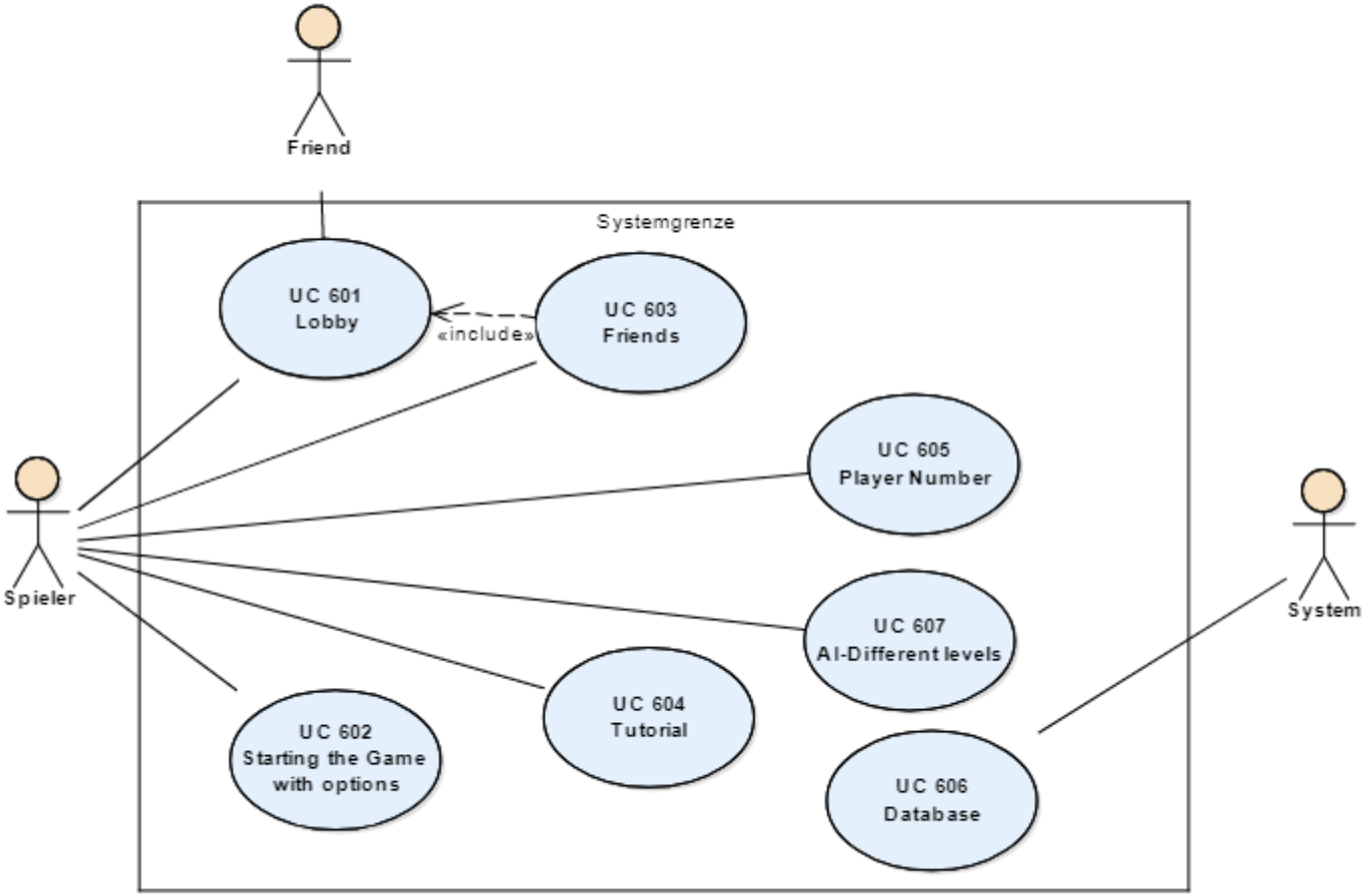


Abbildung 11: Diagramm 601 - 607

6 Endprodukt Use Cases

6.1 Einzel- oder Mehrplayer

Dieses Storyboard zeigt die derzeitige Situation auf, dass ein Spiel zu zweit begonnen werden könnte.

Sollte dies aber mal nicht möglich sein, kann ein Spiel auch im Singleplayer-Modus gespielt werden.

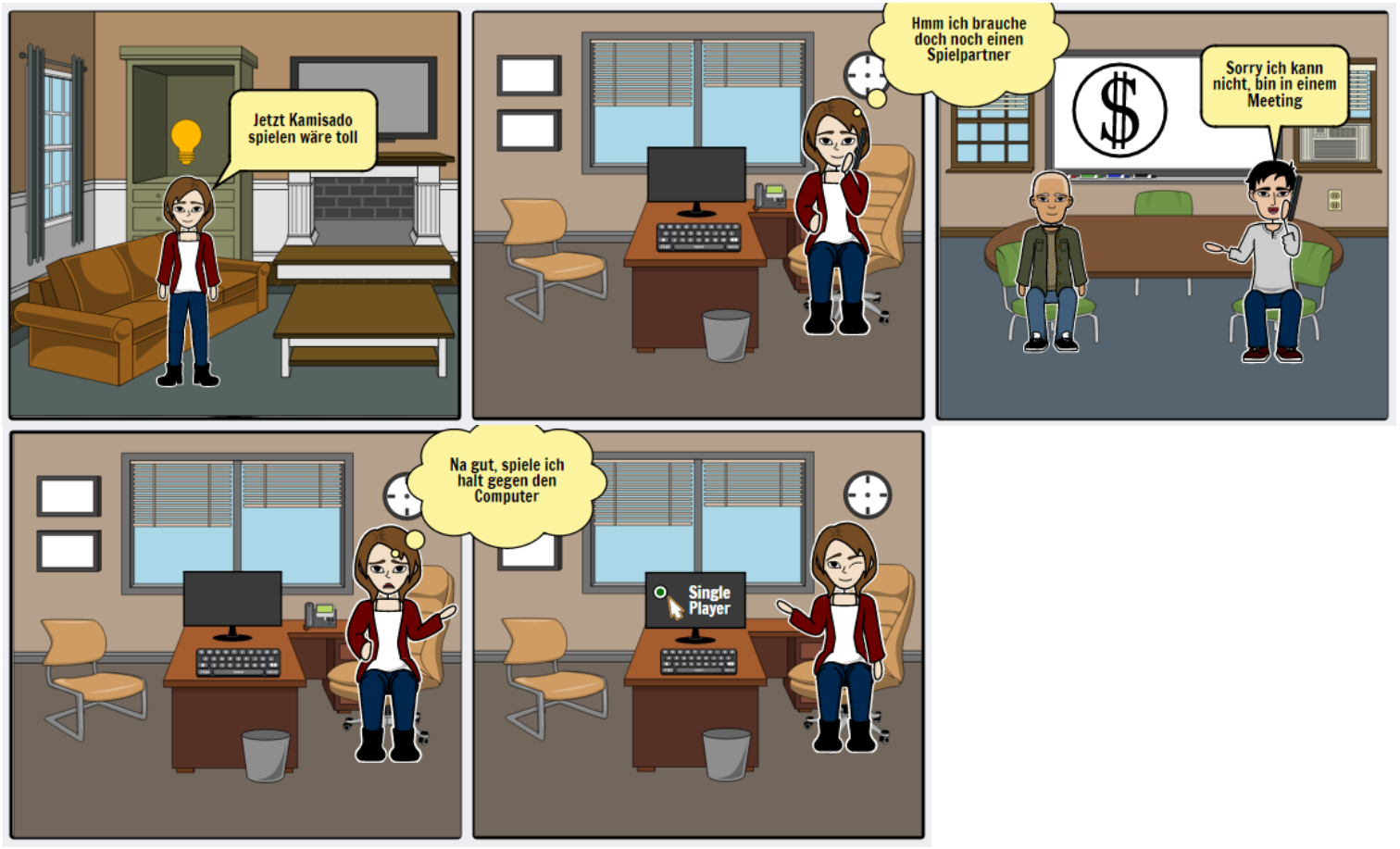


Abbildung 12: StoryBoard möglicher Ablauf

7 Klassendiagramm

Das Klassendiagramm musste aufgrund der Übersichtlichkeit und auch der Grösse auseinander genommen werden. Dies in die verschiedenen einzelnen Ansichten, welche programmiert wurden. Sämtliche Klassen sind mit dem GameMenu_Model als Model selbst verbunden. Die Beziehungen hierzu können in dieser Aufsplittung nicht mehr angezeigt werden.

7.1 Abstract MVC

Abgeleitet sind die Klassen mit Controller, View oder Model jeweils von den abstrakten Klassen

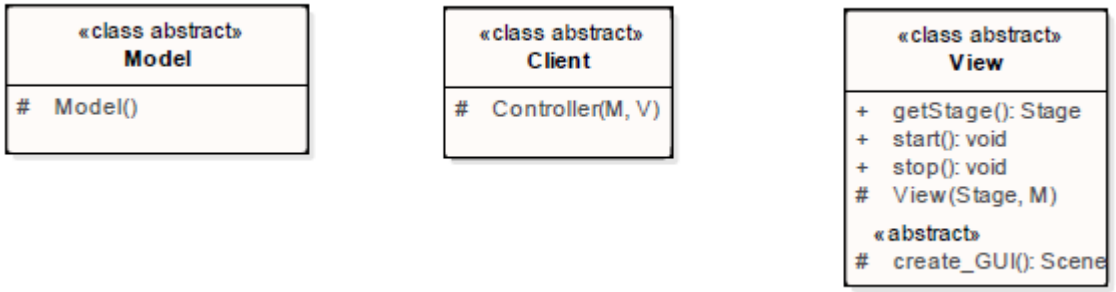


Abbildung 13: Klassendiagramm - abstrakte Klassen

7.2 Allgemeine Klassen

Klassen, welche überall im ganzen Programm gebraucht werden, vor allem aber auch im Ladeprozess des Programmes.

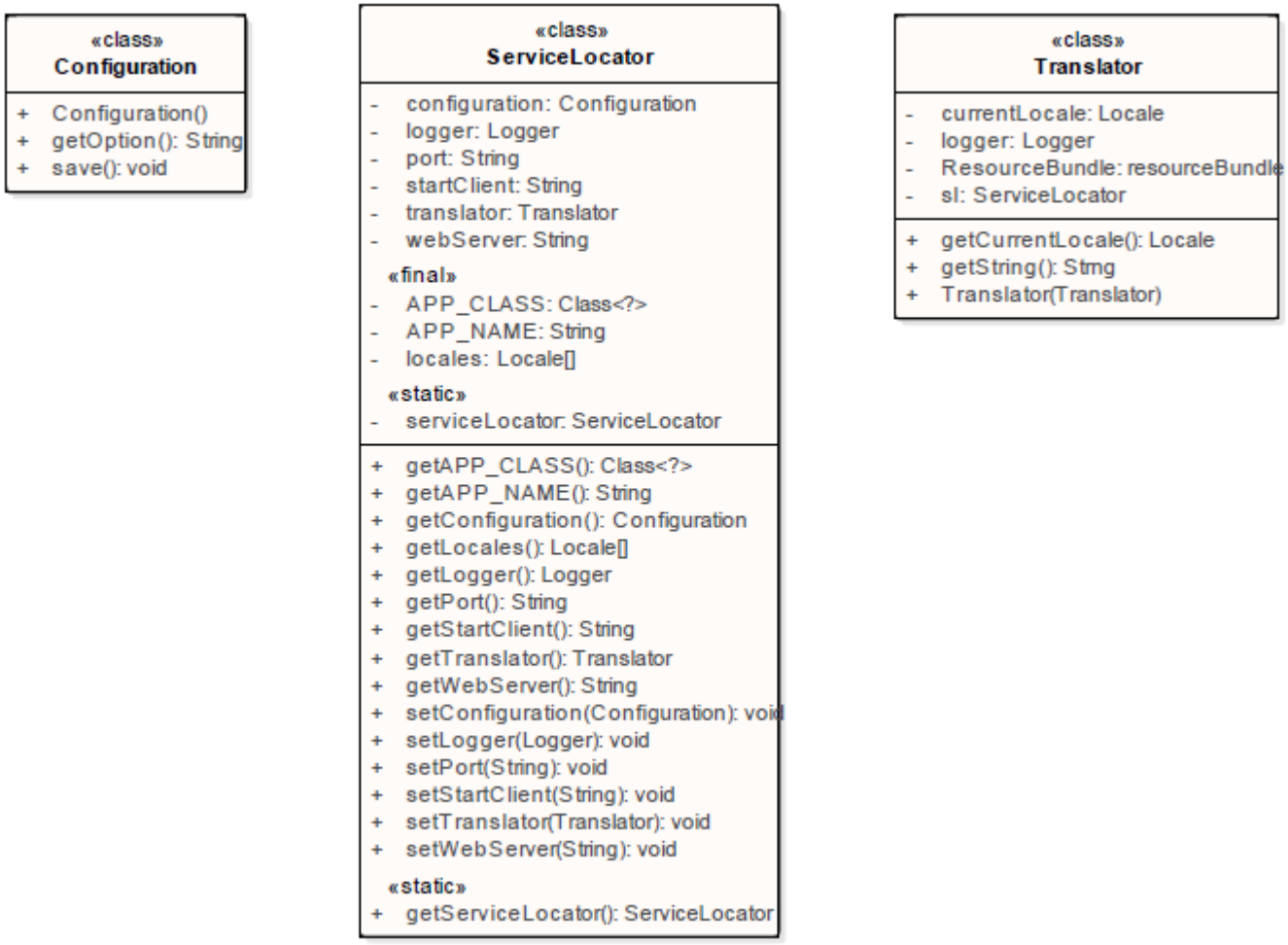


Abbildung 14: Klassendiagramm - allgemeine Klassen

7.3 Splash-Screen

Beim Lade-Prozess des Programmes wird eine Splash-Screen angezeigt, bis alles vollständig geladen wurde.

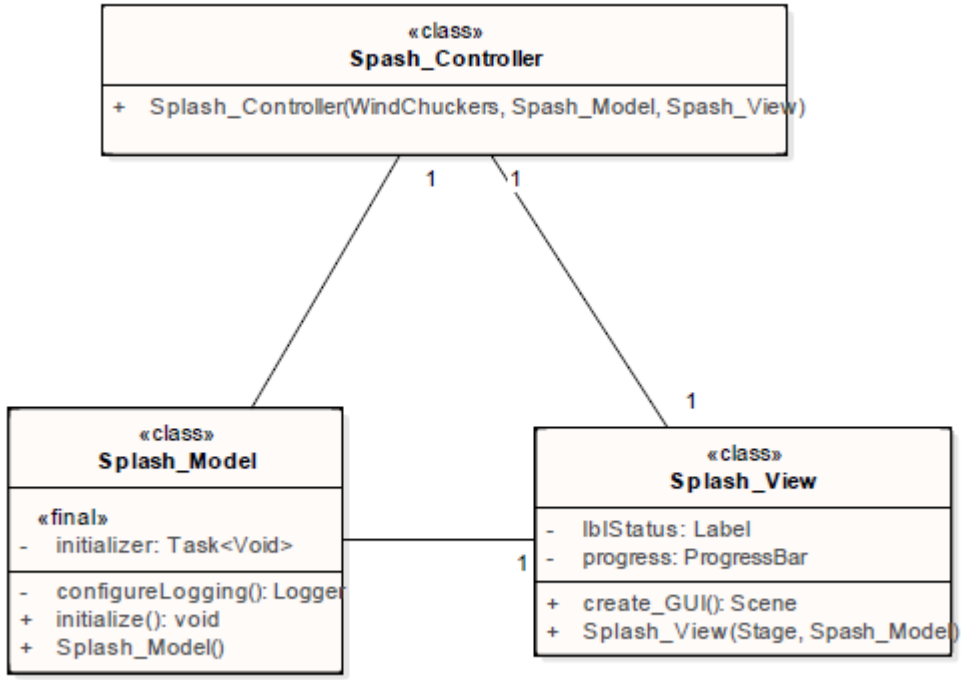


Abbildung 15: Klassendiagramm - Splash Screen

7.4 Client

Damit sich der Client mit dem Server verbinden wird der Client benötigt.

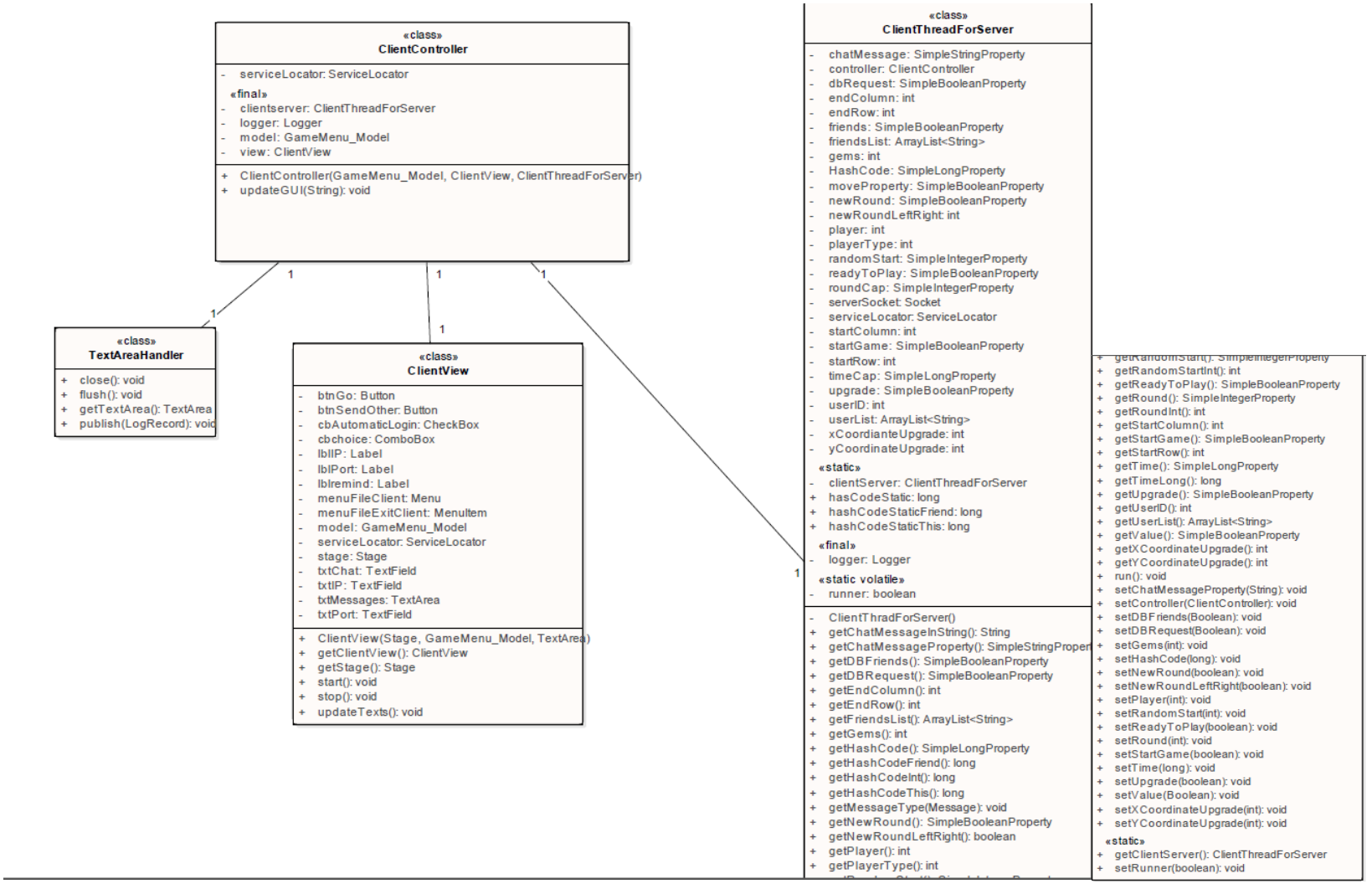
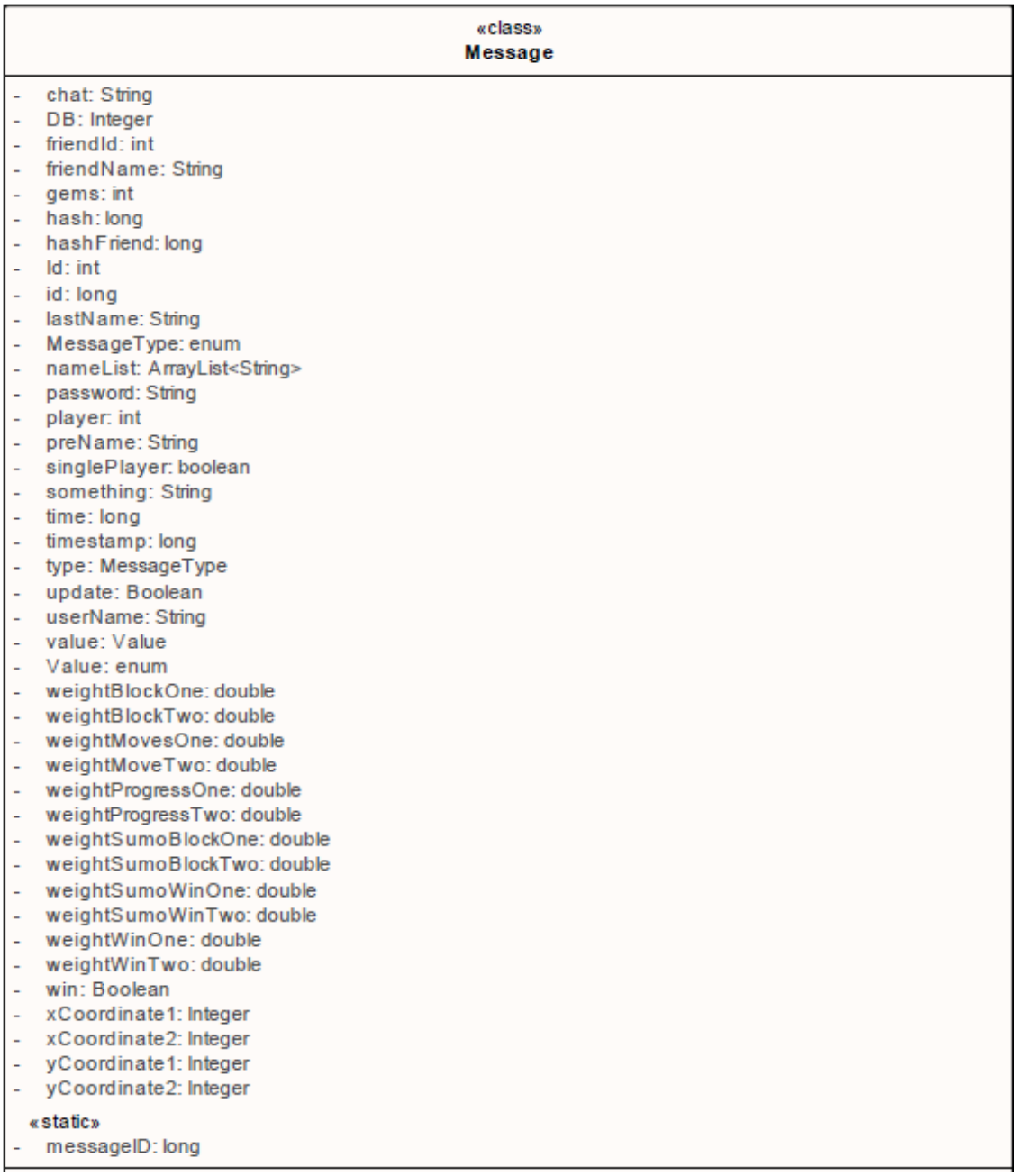


Abbildung 16: Klassendiagramm - Client

7.5 Message

Für den Meldungsaustausch zwischen Client und Server wird die Message-Klasse gebraucht. Diese wird vom Server, sowie vom Client selbst gebraucht.



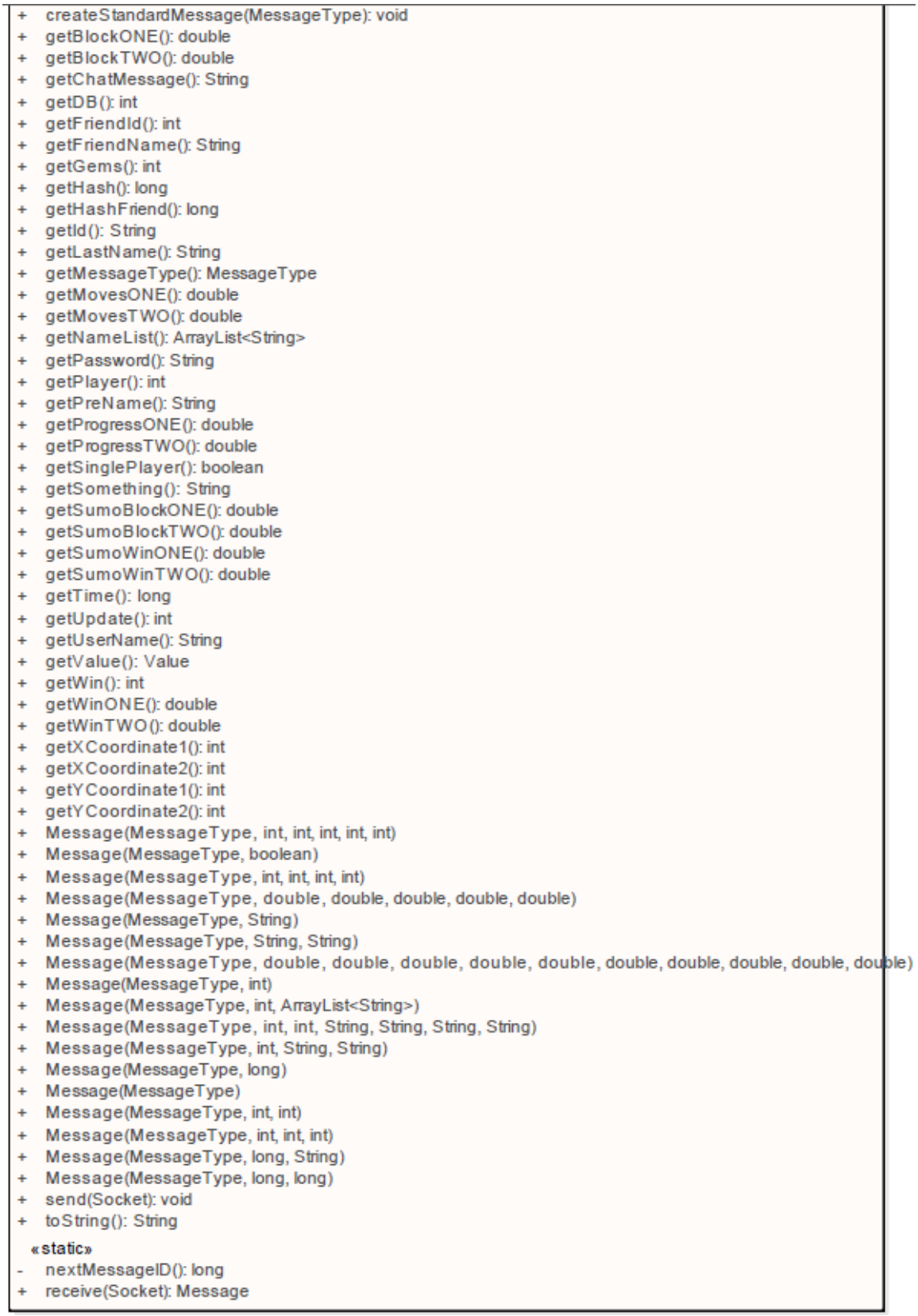


Abbildung 17: Klassendiagramm - Message

7.6 Login

Damit der Benutzer sich einloggen kann wird das Login aufgerufen. In diesem kann ebenfalls auch ein lokales Spiel, ohne Server gestartet werden.

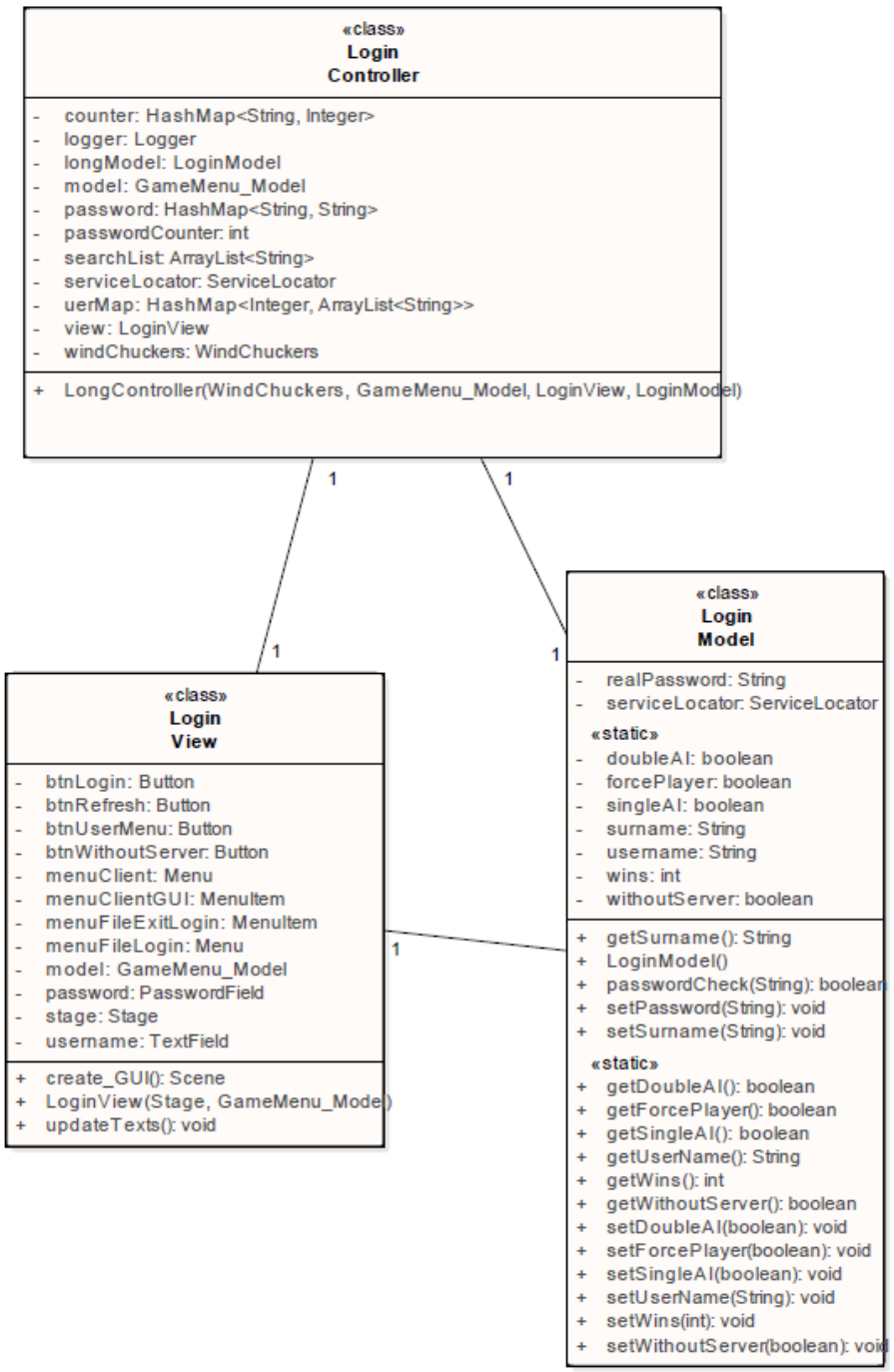


Abbildung 18: Klassendiagramm – Login

7.7 Main Menu

Für die Auswahl des Spiels wird das Main Menu aufgerufen.

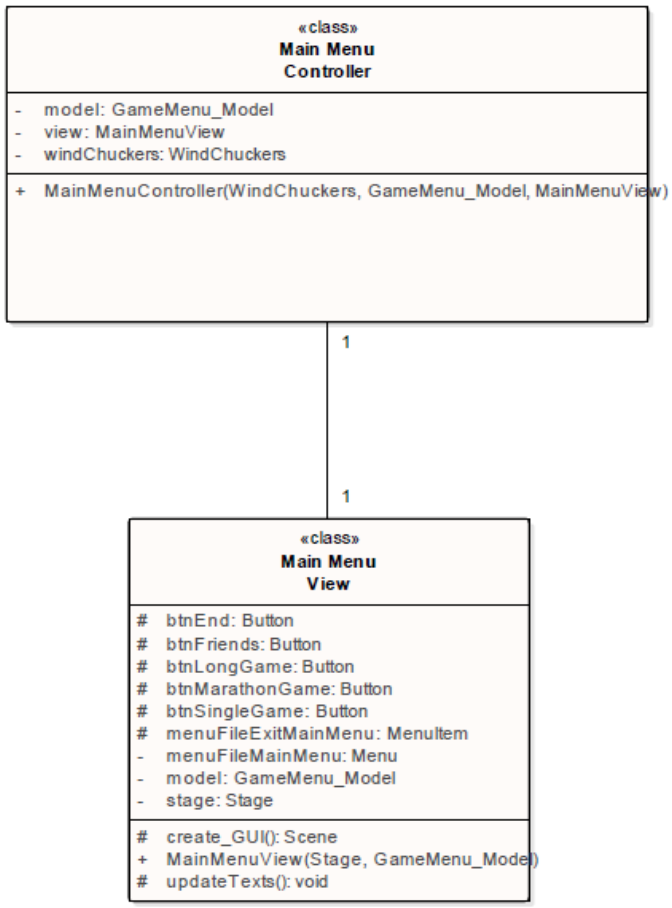


Abbildung 19: Klassendiagramm - Main Menu

7.8 Game Menu

Das Spiel selbst läuft auf dem Game Menu, hierzu werden diverse Klassen gebraucht.

7.8.1 Game Menu View und Controller

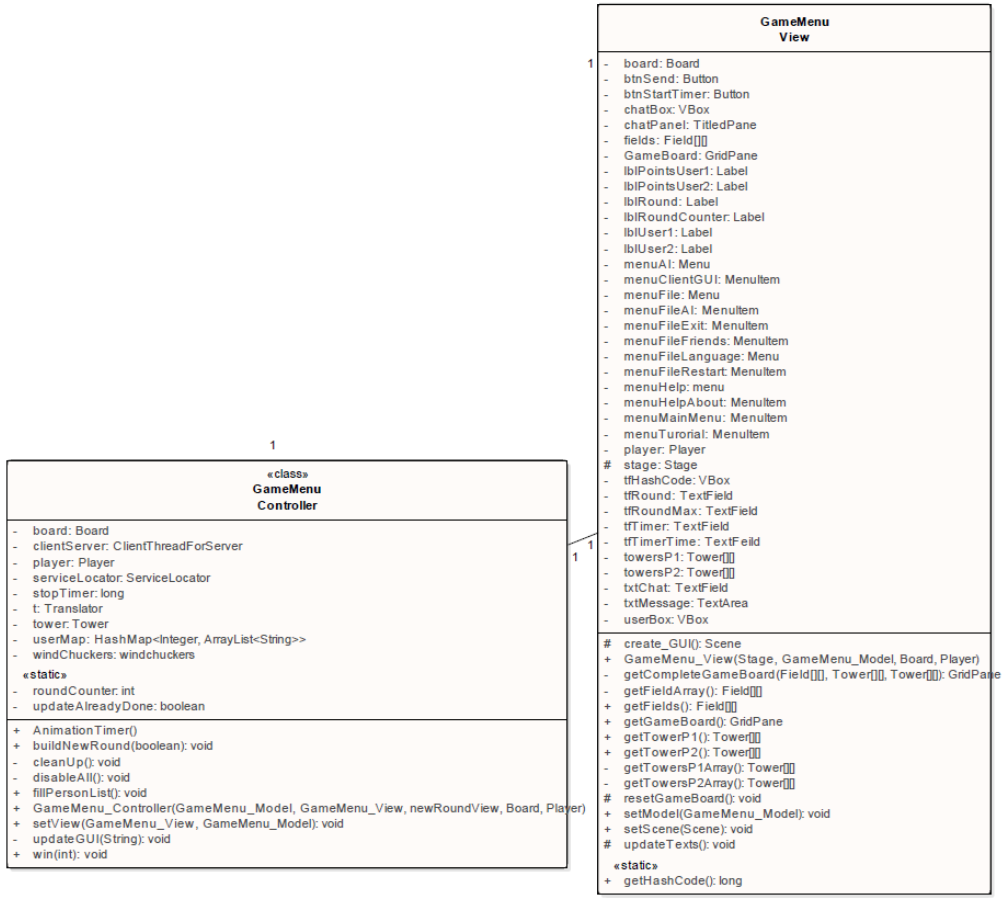


Abbildung 20: Klassendiagramm - Game Menu Client und View

7.8.2 Game Menu Tower , Field, und Player

Diese Klassen sind ebenfalls am GameMenu_Model angehängt, als Erweiterung dessen

Abbildung 22: Klassendiagramm - Tower

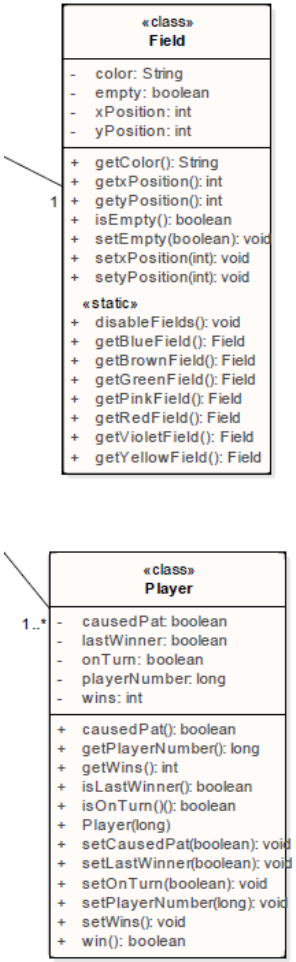


Abbildung 21: Klassendiagramm - Field und Player

7.8.3 GameMenu_Model

Diese Klasse ist bei allen Klassen auf dem Client eingebunden, muss aufgrund der grösse aber alleine Angezeigt werden.

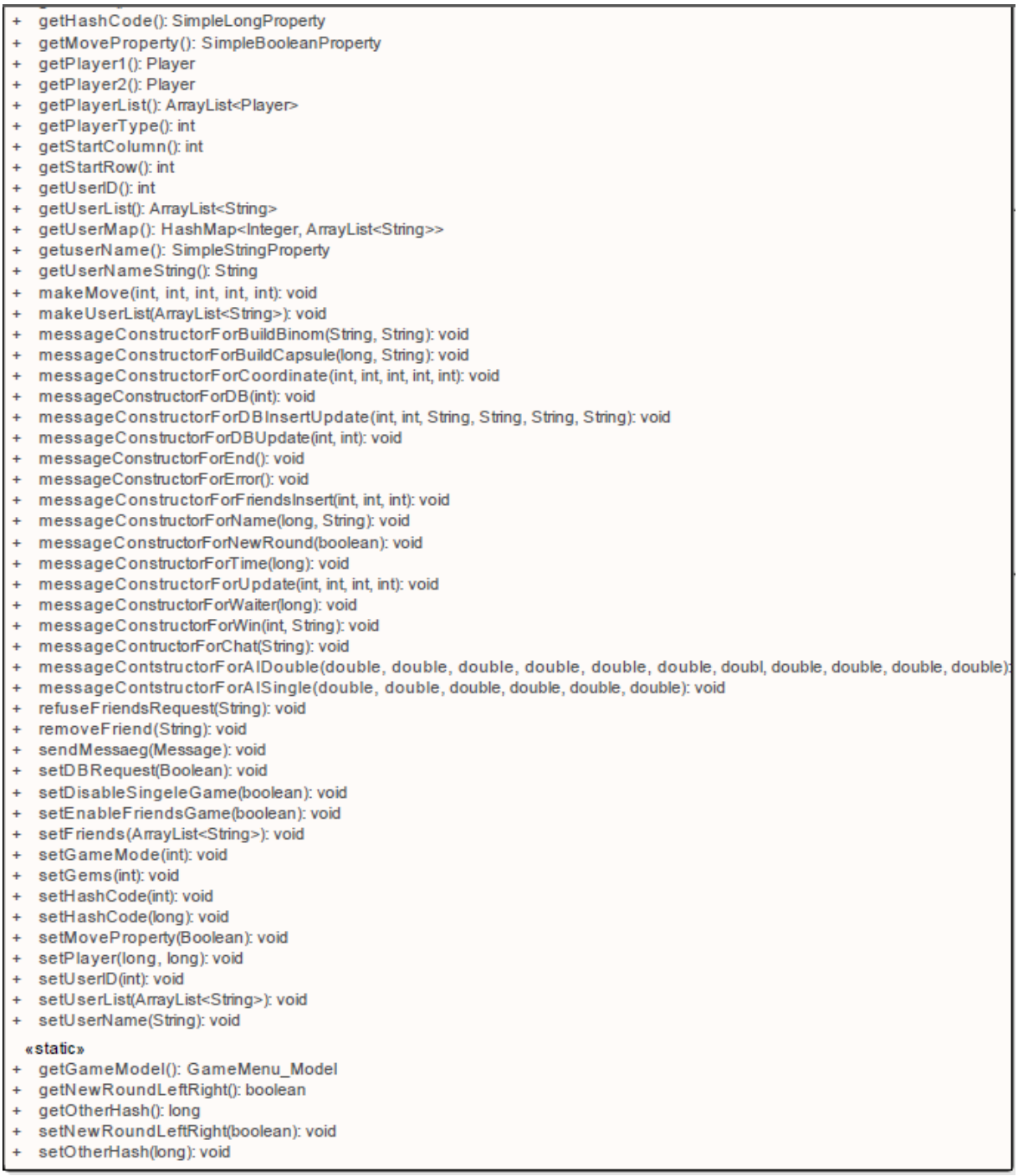


Abbildung 23: Klassendiagramm - GameMenu_Model

7.9 AI Steuerung

Für die Steuerung der AI aus dem Game Menu.

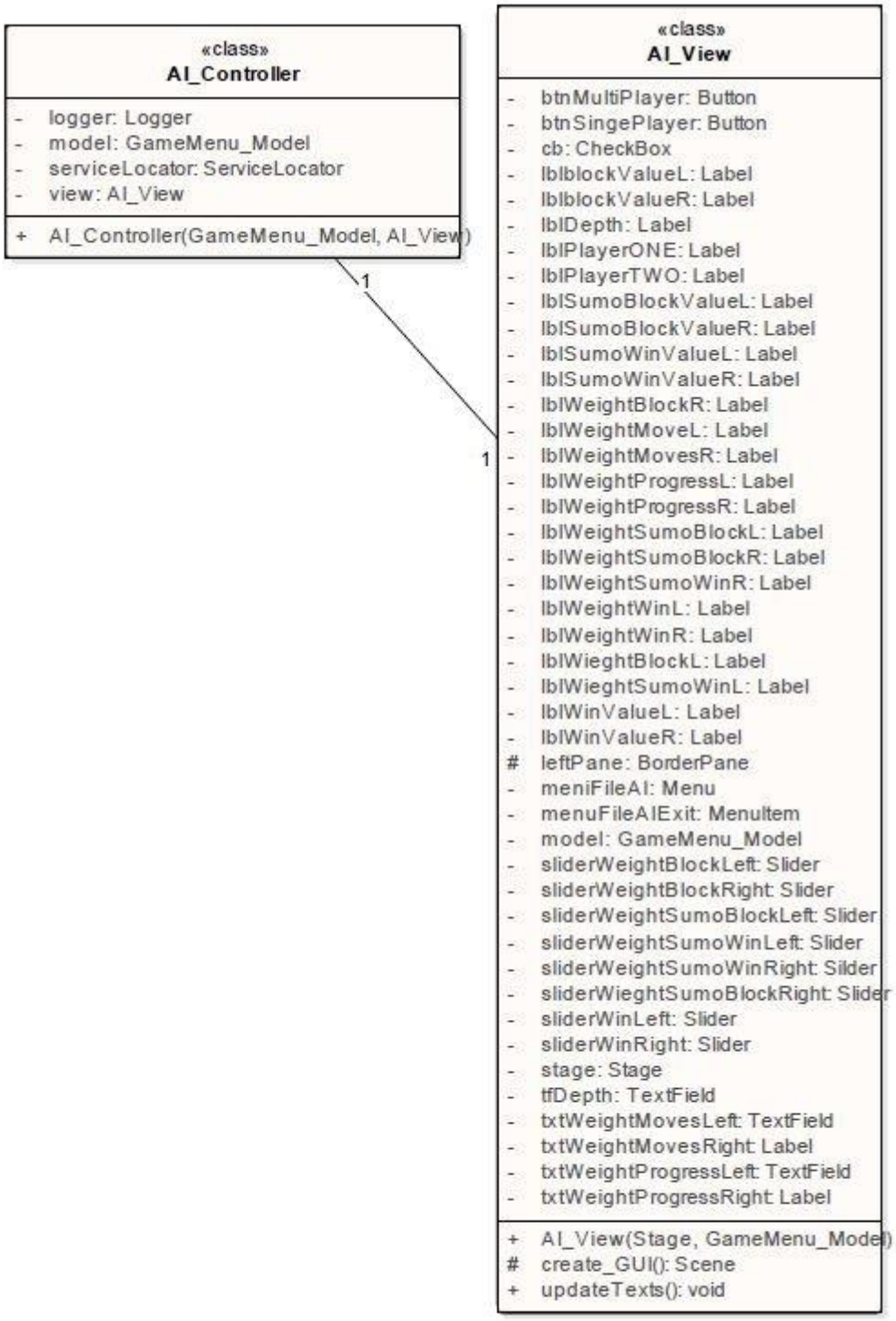


Abbildung 24: Klassendiagramm - AI Controlling

7.10 Friends

Für das Spiel mit Freunden wird das Friends-Menu benötigt.

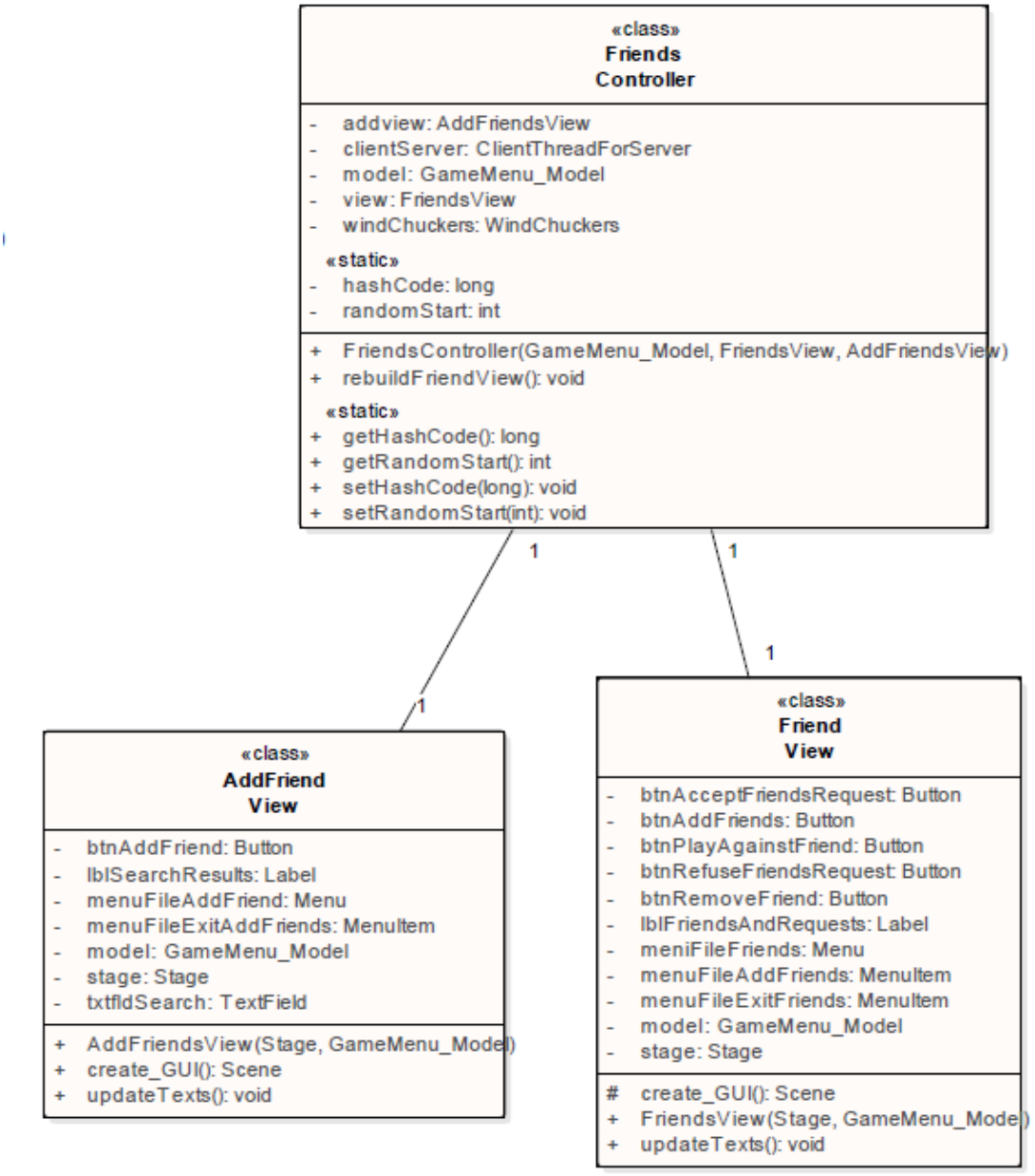


Abbildung 25: Klassendiagramm - Friends

7.11 Lobby

Sobald sich zwei Freunde entschieden haben miteinander zu spielen, wartet das Programm das „Go“ beider Spieler in der Lobby ab.

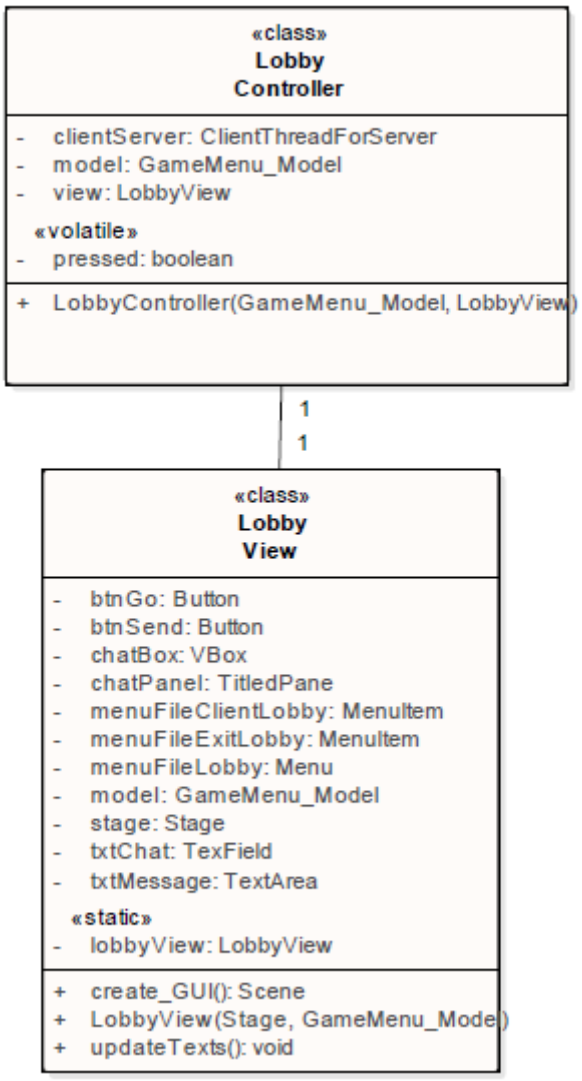


Abbildung 26: Klassendiagramm - Lobby

7.12 Turorial

Möchte der Spieler wissen was den einzelnen Spielen für Regeln zu Grunde liegt, öffnet er das Tutorial.

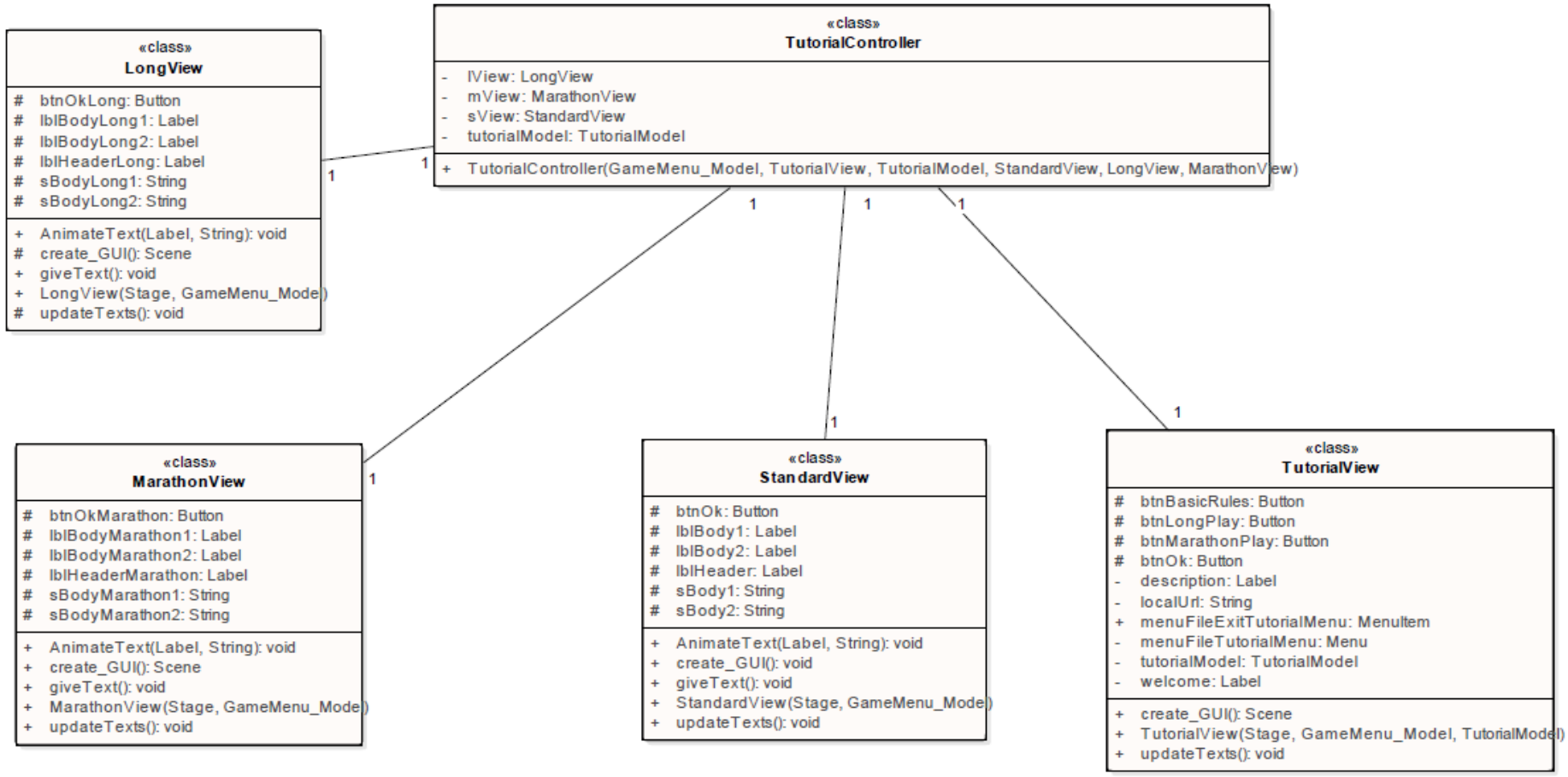


Abbildung 27: Klassendiagramm – Tutorial

7.13 User Menu

Damit neue Users angelegt, oder bestehende Users verwaltet werden können benötigt es das User Menu.

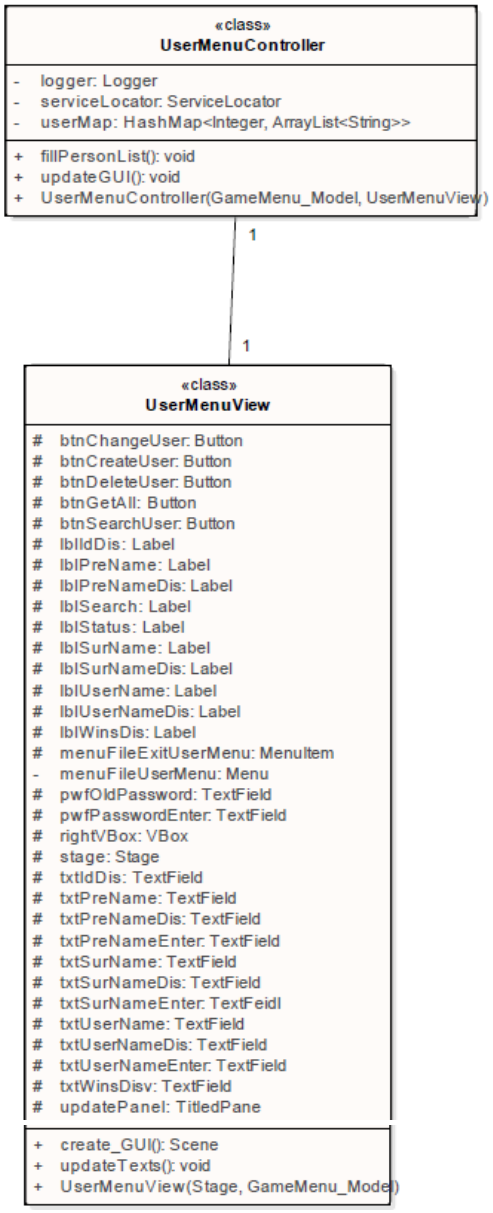


Abbildung 28:Klassendiagramm - User Menu

7.14 Aufforderung zur neuen Runde

Wenn eine Runde zu Ende ist, wird der Spieler aufgefordert zu wählen wie er die neue Runde gestalten möchte.

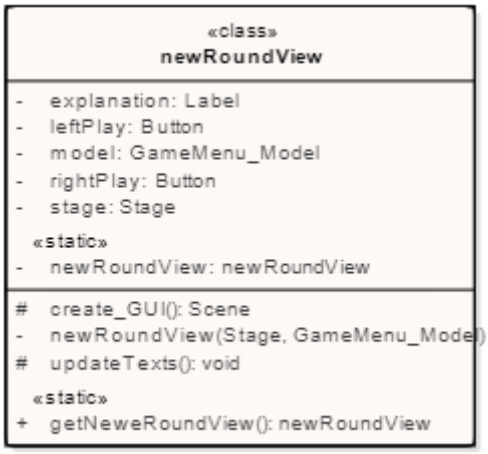


Abbildung 29: Klassendiagramm - neue Runde

7.15 Steuerung der Menus

Für die Steuerung der Menu's, sowie das Start-Programm des Clients.

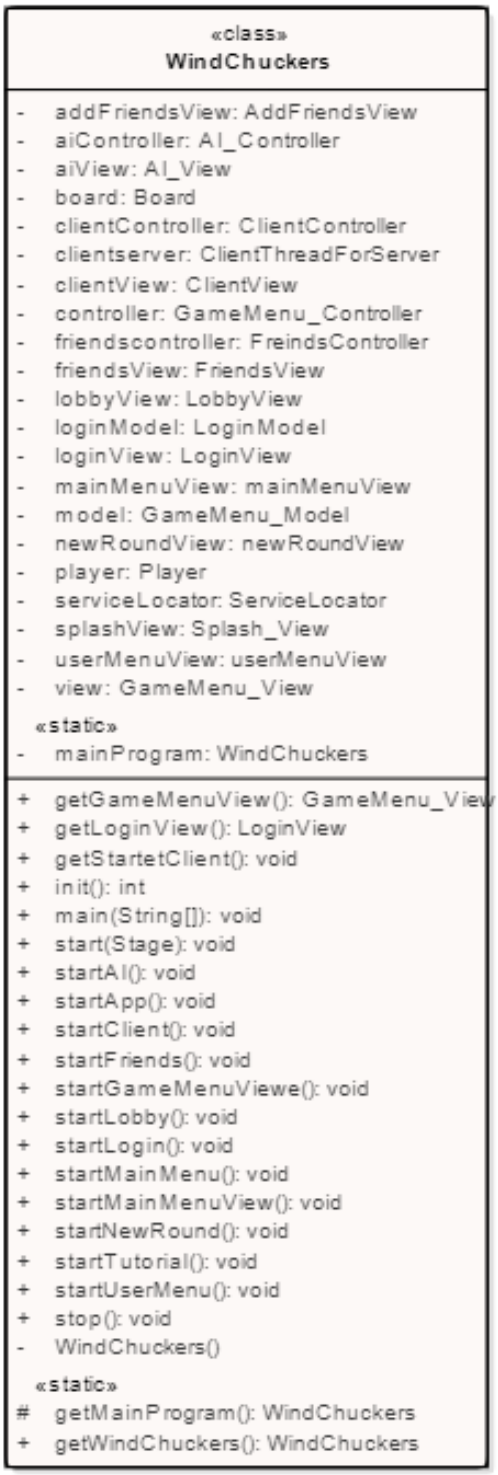


Abbildung 30: Klassendiagramm - Steuerung Programm

7.16 Server

Der Server als Gegenseite verwaltet das Spiel auf dem Server selbst.

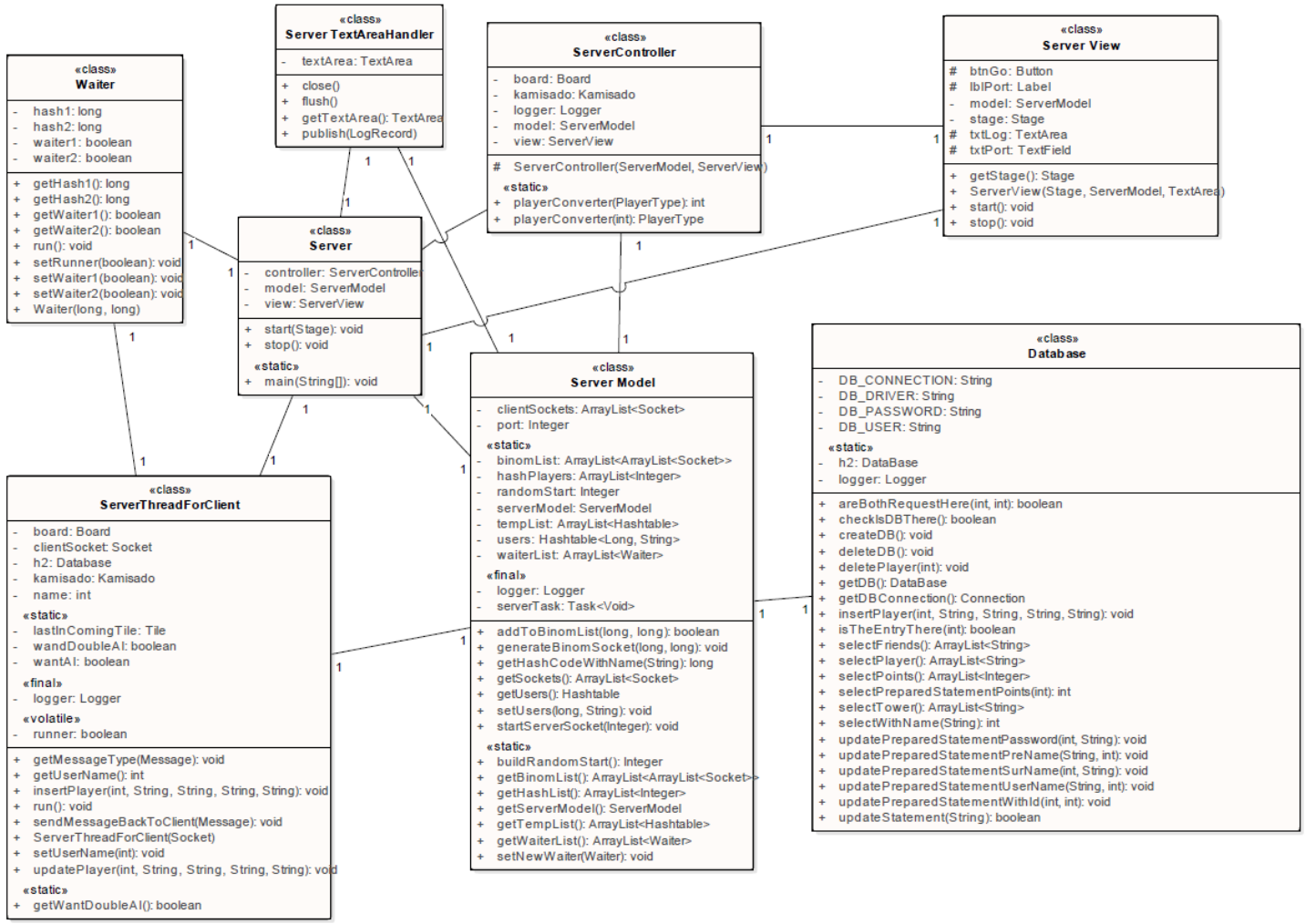


Abbildung 31: Klassendiagramm - Server

Die Kommunikation zwischen Client und Server teilt sich die Klasse „Message“. Das Format für die Kommunikation ist XML, dazu wird die „Simple XML“-API verwendet.

In der Datenbank werden hauptsächlich die Spieler-Daten gespeichert. Also Id, Name, Vorname, Punkte, SumoTower und wie viele Spiele bereits gewonnen wurden für das Ranking.

7.17 AI

Der Computer-Gegner selbst berechnet sich in der AI. Verbunden sind die beiden Teile durch die Klassen PlayerType und MiniMaxAlphaBeta.

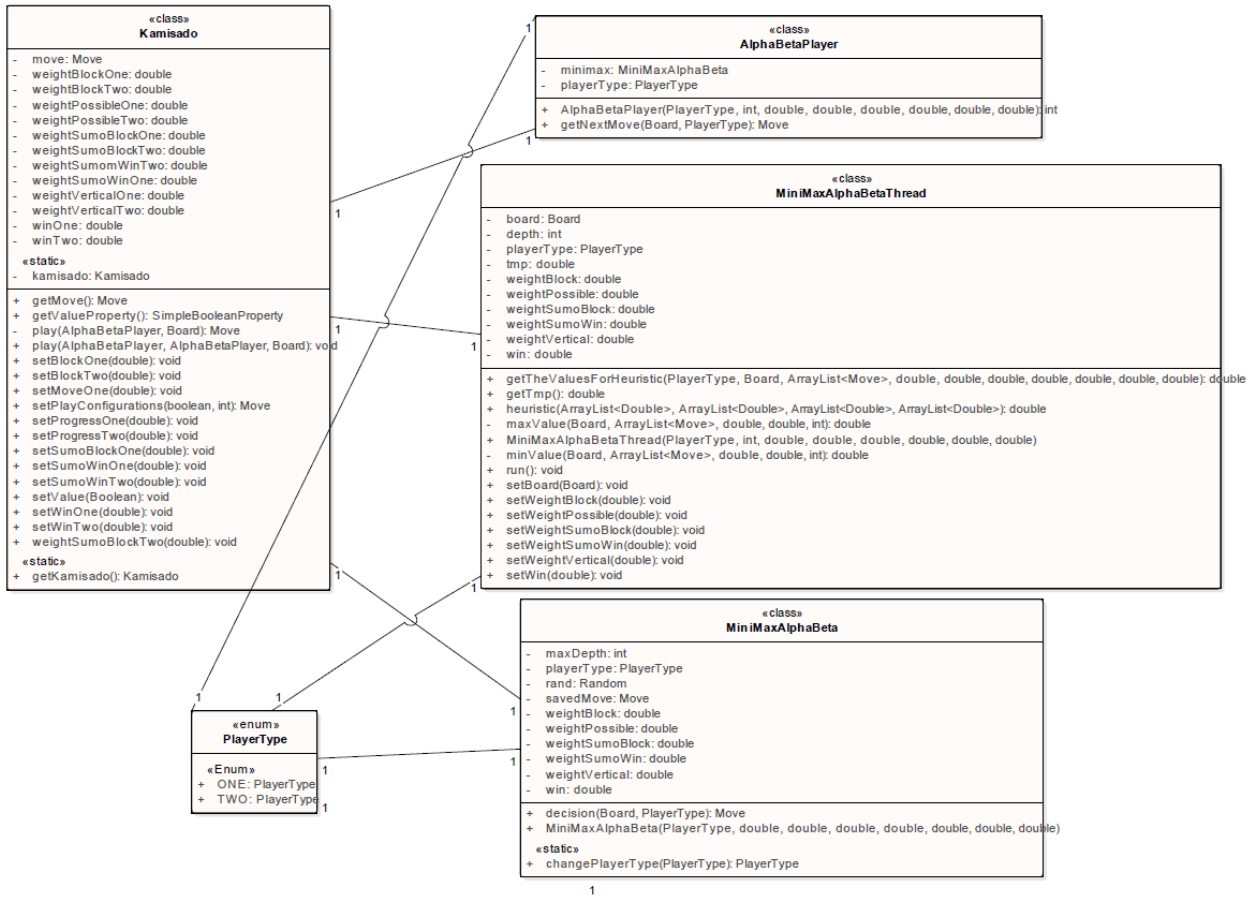


Abbildung 32: Klassendiagramm - AI Teil 1

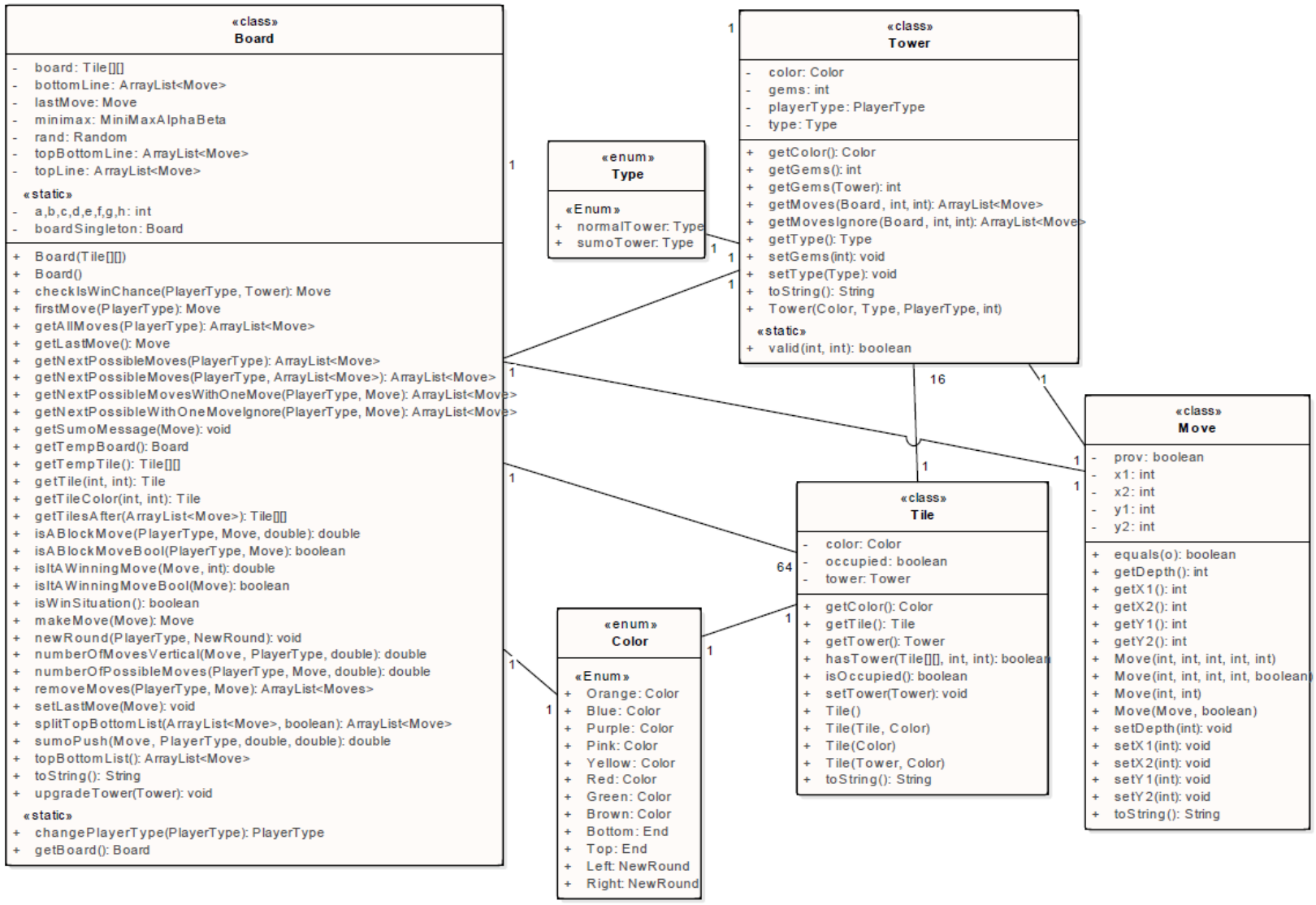


Abbildung 33: Klassendiagramm AI – Teil 2

8 Entity-Relationship-Modell

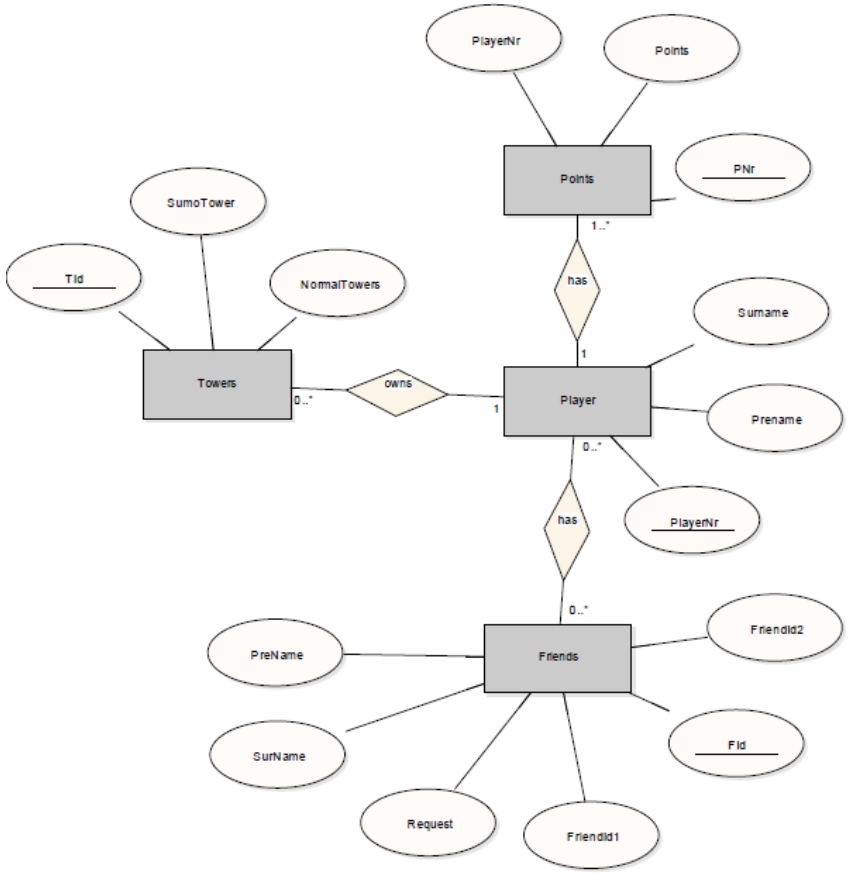


Abbildung 34: ERM-Model

9 Relationales-Modell

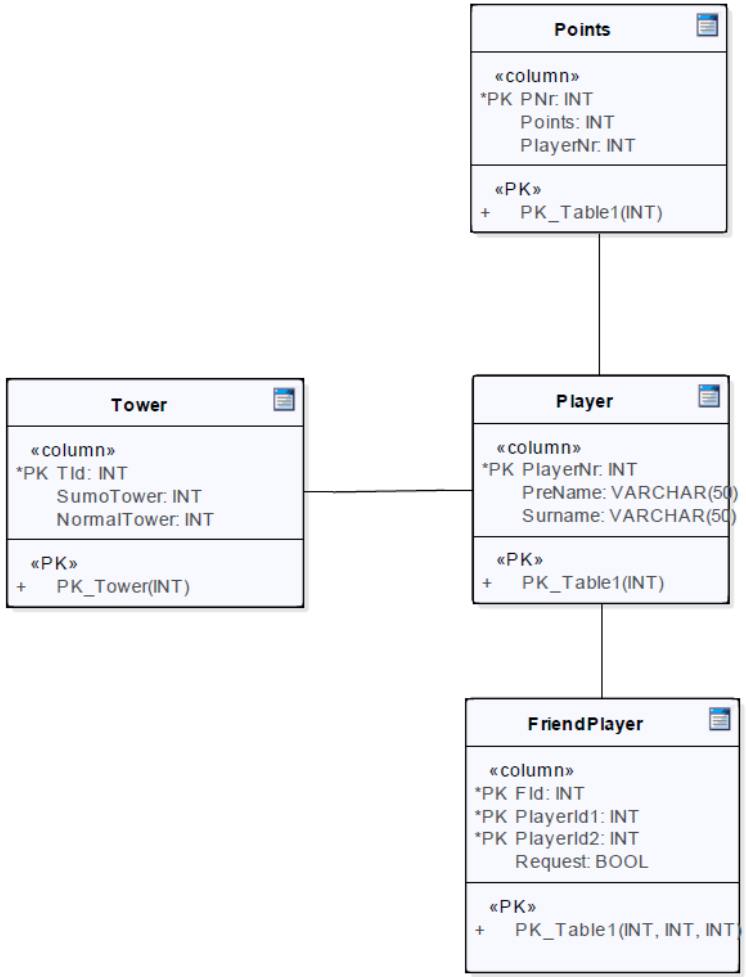


Abbildung 35: Relationales Modell

10 Verteilungsdiagramm

Das System benötigt eine 3-Tier-Aufteilung, damit sämtliche Funktionen korrekt ausgeführt werden. Auf dem Client selbst läuft die View, welcher die Rückgabe des Webservers anzeigt. Auf dem Webserver wird die gesamte Logik ausgeführt. Der Server kann die Ansicht an mehrere Clients senden, sprich mehrere User können gleichzeitig auf das Programm zugreifen.

Es gibt eine interne Datenbank. In der internen werden die Benutzerdaten und der Punktestand abgelegt.

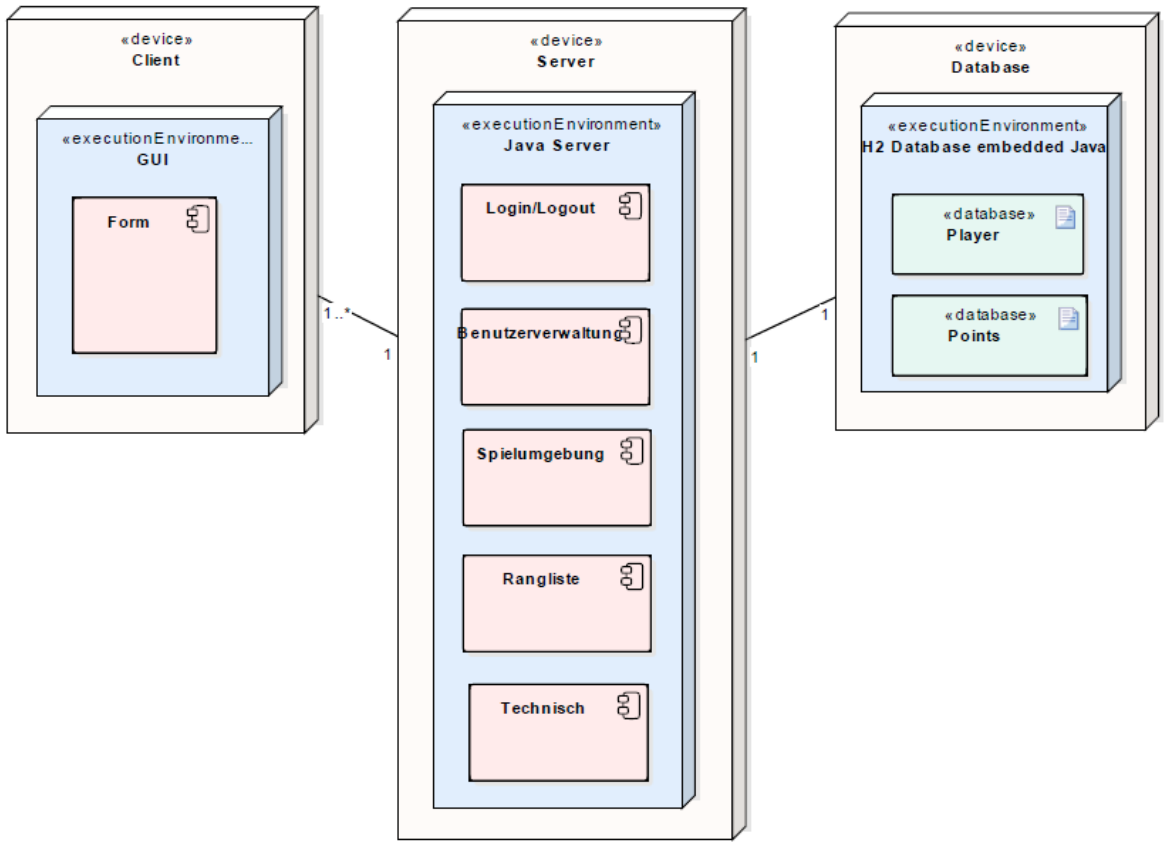


Abbildung 36: Verteilungsdiagramm

11 Qualitätsanforderungen

Zur Ergänzung der funktionalen Anforderungen von Kapitel 5 ergeben sich folgende nicht funktionale Anforderungen:

Anforderungen	Messkriterien
Verfügbarkeit	Das System muss eine jährliche Verfügbarkeit von 90% aufweisen
Abänderbarkeit	Eine neue Komponente soll in weniger als einem Monat designt, entwickelt und getestet werden können.
Effizienz	Die Reaktion des Systems soll unter 5 Sekunden sein. Eine Änderung resp. ein Update soll in weniger als 10 Sekunden erledigt sein.
Erlernbarkeit	Der Benutzer soll das System innert 10 Minuten erlernt haben.
MTTF (Mean time to failure)	Das System soll min. 50 Stunden laufen, bis der erste Fehler auftritt.
MTTR (Mean time to repair)	Der aufgetretene Fehler muss innert 5 Arbeitstagen repariert sein.

Tabelle 31: Qualitätsanforderungen

12 GUI-Bestandteile

12.1 Login

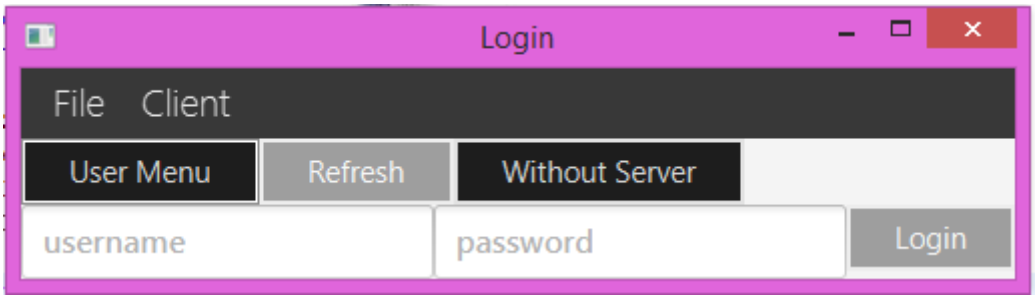


Abbildung 37: GUI Login

12.2 Lobby

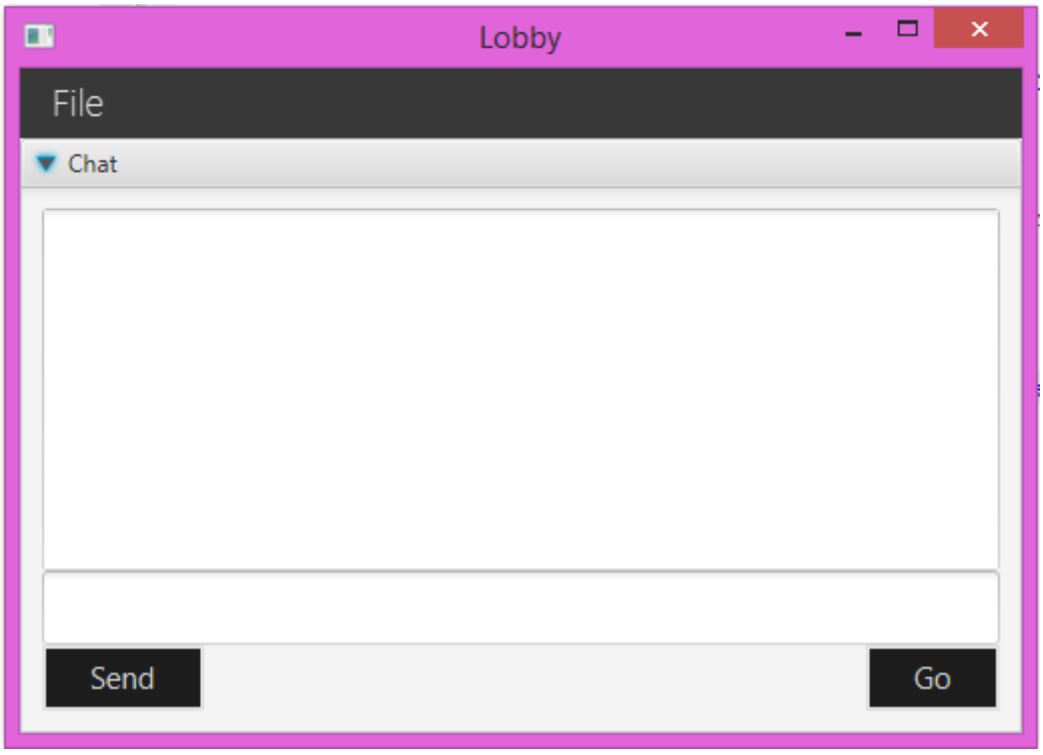


Abbildung 38: GUI Lobby

12.3 Main Menu

Abbildung 39: GUI Main Menu

12.4 Friends Menu

Abbildung 40: GUI Friends Menu

12.5 Add Friends

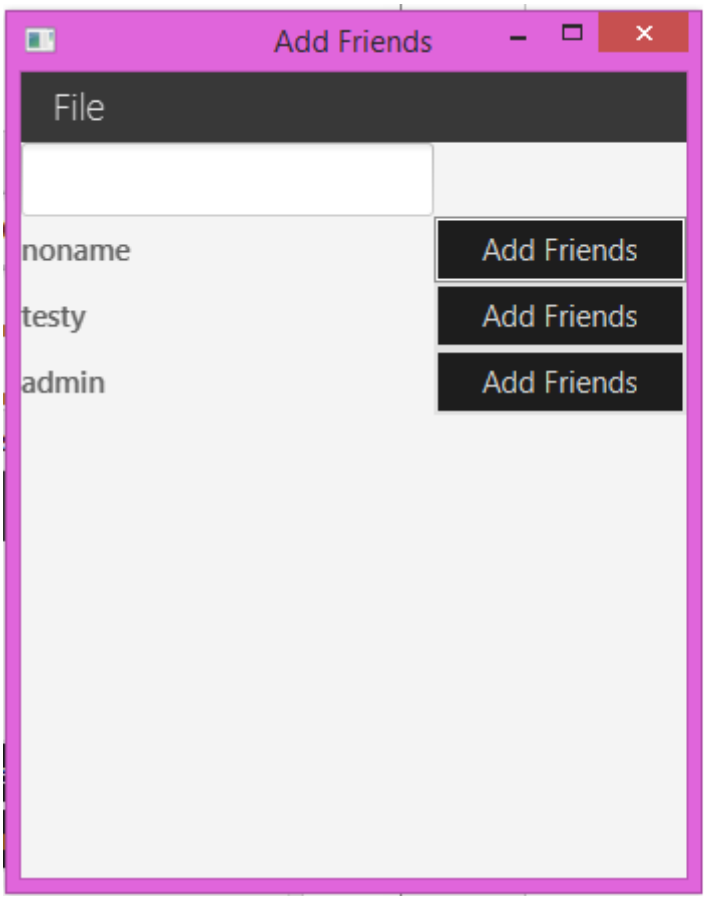


Abbildung 41: GUI Add Friends

12.6 Game Menu

Abbildung 42: GUI Game Menu

13 Testkonzept

13.1 Einführung

Bei dem vorliegenden Dokument handelt es sich um den Testplan der Gruppe WindChuckers, welcher im Rahmen des IT-Projektes das an der FHNW durchgeführt wird, erstellt wird.

13.2 Zweck

Zweck des Testplans ist es, ein organisiertes und systematisches Testen des IT-Projektes zu ermöglichen. Es werden die zu testenden Komponenten des Systems benannt und dazu adäquate Testfälle entworfen. Auch werden Kriterien gesetzt, wann der Test als erfolgreich durchgeführt gilt und wann er abgebrochen werden muss.

13.3 Bezeichnen der Testfälle

Die Testfälle sind nach folgendem Prinzip aufgebaut:

- Die Reihenfolge der Testfälle stimmt mit der Reihenfolge der Use Cases überein
- Sämtliche Ausprägungen werden aufgelistet
- Die Testfälle erhalten immer eine Testfallnummer

13.4 Abnahme- und Testkriterien

Wann wird das Testen beendet? Die angegebenen Kriterien müssen objektiv prüfbar sein.

Die „wahre“ Fehlerzahl einer Software ist unbekannt. Daher müssen im Vorfeld Kriterien bestimmt werden, die festlegen, in welchem Umfang das Testen stattfindet und wann die Tests beendet werden können.

Es gibt für jeden Test ein erwartetes Ergebnis. Trifft dieses ein gilt der Test als erfolgreich und ist abgeschlossen. Bei einem unerwarteten Ergebnis wird der Test zu einem späteren Zeitpunkt wiederholt.

13.4.1 Erfassung Testfälle und Anweisungsabdeckung

Anweisungsabdeckung: Tests wird die Anweisungsabdeckung gemessen. Wir streben eine Anweisungsabdeckung von 90 Prozent an. Dies wird bestimmt dadurch, dass von jedem Use Case sämtliche aufgeführte Ausprägungen als Testfall erfasst werden. Ebenfalls kann damit festgestellt werden, welche Zeilen des Codes vom Testfall durchlaufen wurden und welche nicht. Die Tests müssen die angegebene Anweisungsabdeckung erreichen, vorher darf nicht abgebrochen werden.

13.4.2 Unit-Tests

Zusätzlich werden durch Unit-Tests mit jUnit die Testfälle aufgebaut und eingerichtet. Folgende Ausprägungen werden pro Klasse erfasst:

- @BevorClass (Ausführung vor den Tests, Vorbereitung der Daten)
- @BevorMethods (Ausführung vor den einzelnen Tests)
- @AfterMethods (Nach jedem einzelnen Testfall)
- @AfterClass (Wenn alle Testfälle durchgelaufen sind)

Dies stellt sicher, dass die Klasse in sich komplett ist und funktioniert.

Die einzelnen Testfälle werden alle als separierte Tests mit jUnit erfasst, dies nach den oben beschriebenen Ausprägungen, welche bereits schriftlich festgehalten wurden.

Diese Methode stellt sicher, dass bereits geschriebener und implementierter Code nicht später Fehler auswirft, welche nicht erkannt werden bei neuen Implementierungen.

13.4.3 Testergebnisse

Testergebnisse: Erfolgreiche Tests (d.h. Tests, die einen Fehler gefunden haben), werden in Fehlerklassen entsprechend Tabelle kategorisiert. Pro 100 Zeilen Code dürfen nur Fehler mit einer Wertigkeitssumme von 100 auftreten. Das Testen darf nicht beendet werden, solange noch Fehler der Klassen 3 oder 4 auftreten. Die zur Verfügung stehende Zeit ist begrenzt. Es ist sinnvoll, frühzeitig und regelmassig den Fortschritt der Testphase zu beurteilen.

Fehlerklasse	Beschreibung	Wertigkeit
1	unwesentlich: Fehler, der keine sichtbaren Schäden verursacht, aber trotzdem korrigiert werden muss	1
2	gering: Fehler, der das System in seiner Leistung einschränken oder dessen Verfügbarkeit reduzieren könnte	10
3	kritisch: Fehler, der schwere Schäden am System verursachen kann	20
4	katastrophal: Fehler, der die Unterbrechung des Systems verursachen kann bzw. das System unbrauchbar macht	100

13.5 Vorgehen

13.5.1 Beginn Projekt

Ein Projekt wird auf GitHub.com eröffnet. Dazu wird ein leerer Model-View-Controller erstellt und direkt „gepusht“, sodass die Entwickler, welche am Projekt arbeiten einen Branch von der Repository auf ihren Laptop holen können und von dort aus arbeiten. So starten alle von derselben Basis.

13.5.2 Aufbau Testing

Wie in Abschnitt 16.4.1 beschrieben werden sämtliche Ausprägungen der erfassten Use Cases als Testfälle erfasst und festgehalten.

13.5.3 Unit-Tests

Wie in Abschnitt 16.4.2 beschrieben werden zu allen erfassten Ausprägungen, parallel zum entstehenden Code für die Umsetzung der Use Cases, Unit-Tests geschrieben. Somit wird am Schluss jede Methode einen Unit-Test haben, welcher garantiert, dass keine Fehler im Programm vorhanden sind.

13.5.4 Implementierung des neuen Codes

Jeweils bevor ein „commit“ gemacht wird, muss mittels Unit-Tests sichergestellt werden, dass das Programm läuft und sich so verhält wie es die Anforderungen definieren. Es darf kein Unit-Test auf „Fehler“ gelaufen sein.

Erst dann darf das Programm gepusht werden und in die Repository aufgenommen werden.

13.5.5 Neuster Stand

Sobald dies der Fall ist, sollen die anderen Benutzer nun die aktuellste Repository via „fetch“ abholen, sodass ihre Version auf dem neusten Stand ist und dass auch bezüglich dem Testing keine Lücken entstehen.

13.5.6 Freigabe Komplettpacket

Die einzelnen Use Cases werden mittels Testprotokoll, sowie Unit-Tests getestet und auch geführt. Erst wenn das Testprotokoll angibt, dass sämtliche Testfälle ok sind kann eine Version des Programms freigegeben werden.

13.6 Testprotokoll

Auftraggeber	FHNW, Olten
Projektleiter	Weber Lukas
Autor	Weber Lukas
Klassifizierung	Intern
Status	In Arbeit

Änderungsverzeichnis

Datum	Version	Änderung	Autor
08.03.2017	1.0	Erstellung	Wel
28.03.2017	1.1	Komplettierung der Ausprägungen	Wel
02.08.2017	1.2	Testdurchführung	Wel

Testfallsammlung:

UseCase-Gruppe	UC Nr	Bezeichnung	Testfall	Kurzbeschreibung	Resultat
100 Login/Logout					
	101	Login	Standard	Einloggen des Users im System	Ok
			Alternativ 1	Passwort / Username falsch eingeben	Ok
			Alternativ 2	Abbruch	Ok
	102	Logout	Standard	Abmelden des Users	Ok
			Alternativ 1	Abbruch	Ok
200 Benutzerverwaltung					
	201	Benutzer hinzufügen	Standard	Benutzer hinzufügen	Ok
			Alternativ 1	Benutzerdaten unvollständig	Ok
			Alternativ 2	Benutzer ist bereits erfasst	Ok
			Alternativ 3	Abbruch	Ok
	202	Benutzer bearbeiten	Standard	Bearbeiten des Benutzers	Ok
			Alternativ 1	Keine oder unvollständige Daten eingeben	Ok
			Alternativ 2	Abbruch	Ok
	203	Benutzer suchen	Standard	Benutzer suchen	Ok
			Alternativ 1	Benutzer wird nicht gefunden (nicht erfasst)	Ok
			Alternativ 2	Nicht gefundener Benutzer direkt hinzufügen	Ok
			Alternativ 3	Verbindungsfehler	Ok
			Alternativ 4	Abbruch	Ok
	204	Benutzer löschen	Standard	Benutzer löschen	Ok
			Alternativ 1	Abbruch nach bereits eingebenen Daten	Ok
			Alternativ 2	Abbruch	Ok

UseCase-Gruppe	UC Nr	Bezeichnung	Testfall	Kurzbeschreibung	Resultat
300 Spielumgebung					
	301	Spiel starten	Standard	Spiel starten	Ok
			Alternativ 1	Benutzer wird nicht gefunden	Ok
			Alternativ 2	Als Gast spielen	Ok
			Alternativ 3	Abbruch	Ok
	302	Spiel beenden	Standard	Spiel beenden	Ok
			Alternativ 1	Abbruch der Beendigung	Ok
			Alternativ 2	Abbruch	Ok
	303	Punktestand aufzeichnen	Standard	Nach Beendigung wird Punktestand aufgezeichnet	Ok
	304	Spielfigur auswählen	Standard	Spielfigur kann angewählt werden	Ok
			Alternativ 1	Spielfigur kann nicht ausgewählt werden	Ok
			Alternativ 2	Abbruch	Ok
	305	Spielfigur bewegen	Standard	Freies Feld kann angeklickt werden und Spielfigur verschiebt sich	Ok
			Alternativ 1	Spielfigur verschiebt sich nicht	Ok
			Alternativ 2	Abbruch	Ok
	306	Gewinner ermitteln	Standard	Der Gewinner wird festgelegt	Ok
			Alternativ 1	Der Gewinner wird nicht ausgegeben	Ok
			Alternativ 2	Weiterspielen nach Gewinnerausgabe	Ok
	307	Gewinnpunkte eingeben	Standard	Gewinn-Punkte eingeben	Ok
			Alternativ 1	Gewinn-Punkte können nicht eingegeben werden	Ok

UseCase-Gruppe	UC Nr	Bezeichnung	Testfall	Kurzbeschreibung	Resultat
400 Rangliste					
	401	Rangliste	Standard	Gesammelte Punkte der Spieler ersichtlich	Ok
			Alternativ 1	Abbruch	Ok
500 Technisch					
	501	Verbindung Server-Client	Standard	Die Verbindung kann hergestellt werden	Ok
			Alternativ 1	Die Verbindung kann nicht hergestellt werden	Ok
			Alternativ 2	Abbruch	Ok
	502	Computerspieler (KI)	Standard	Das Spiel wird mit Gegener gestartet	Ok
			Alternativ 1	Abbruch	Ok
	503	Mehrsprachig	Standard	Die Sprache kann geändert werden	Ok
			Alternativ 1	Die Sprache kann nicht geändert werden	Ok
	504	Chat	Standard	Die Chatnachrichten können gesendet und emfpangen werden	Ok
			Alternativ 2	Die Chatnachrichten können nicht gesendet und empfangen werden	Ok

UseCase-Gruppe	UC Nr	Bezeichnung	Testfall	Kurzbeschreibung	Resultat
600 Zusätzlich optionale Use Cases					
	601	Lobby	Standard	Chat Funktion verwenden können	Ok
			Alternativ 1	Abbruch	Ok
	602	Starting the game with options	Standard	Spielvariante auswählen und Button drücken	Ok
			Alternativ 1	Das Spiel wird nicht gestartet	Ok
			Alternativ 2	Abbruch	Ok
	603	Friends	Standard	Friendsmenu aufrufen und entsprechende Eingabe tätigen	Ok
			Alternativ 1	Die Aktion wird nicht ausgelöst	Ok
			Alternativ 2	Abruch	Ok
	604	Tutorial	Standard	Aufrufen des Tutorial Fensters	Ok
			Alternativ	Entsprechender Modi auswählen	Ok
	605	Player-Number	Standard	Anzeige welche Nummer der User ist	Ok
			Alternativ 1	Nummer vom User wird nicht angezeigt	Ok
	606	Database	Standard	Die Daten werden in der Datenbank gespeichert und sind wieder abrufbar	Ok
	607	AI-Different levels	Standard	AI Controller aufrufen und Stärke einstellen	Ok
			Alternativ 1	Stärke nicht anpassbar	Ok

Testdurchführung und Testergebnis

Tester	L.Weber
Datum Testdurchführung	02.08.2017
Fehlerklasse (Testergebnis)	Nun alles i.O
Fehlerbeschreibung	Keine Fehler mehr gefunden

14 Abbildungsverzeichnis

15 Tabellenverzeichnis

16 Anhang

16.1 AI

Für dieses Spiel wurde eigens eine künstliche Intelligenz geschaffen, damit das Spiel auch im Single-Player-Modus gespielt werden kann. Ebenfalls möglich ist es zwei künstliche Intelligenzen gegen einander antreten zu lassen.

Die Parameter für die hier erhobene Heuristik können angepasst werden auf die eigenen Bedürfnisse. Ebenfalls können die Parameter für beide künstliche Intelligenzen geändert werden. So könnte z.B. auch im Feldversuch evaluiert werden welche der Einstellungen in welcher Situation stärker wäre.

Da die Türme im Spiel Kamisado immer gleich wertig respektive gleich stark sind, können nicht die Türme selbst mit Punkten gewichtet werden, sondern es muss jeweils die gesamte Situation betrachtet und die daraus entstehenden Möglichkeiten gewichtet werden.

16.1.1 Heuristik

Dazu wurden sich folgende Heuristiken überlegt:

1. Umso weiter vorne sich ein Turm befindet (mit diesem Zug), desto besser für seine Gewinnchancen
2. Umso mehr Züge ein Turm noch zur Verfügung hat (nach diesem Zug), desto besser steht der Turm
3. Wenn der Zug ein Gewinn erzielt werden kann, muss dieser Zug gemacht werden
4. Wenn der Zug die Möglichkeit bietet den Gegner in seinem nächsten Zug an einem Gewinn zu hindern, wird der Zug besser bewertet
5. Wenn der Turm ein Sumo ist können folgende Situationen zum Vorteil führen:
    a. Der Sumo-Push ermöglicht für den nächsten Zug einen Sieg
    b. Der Sumo-Push ermöglicht für den nächsten Zug einen Block zu erzielen

16.1.2 Gewichtung

Die Standard-Gewichtung:

Nummer	Gewichtung	Einheit	Was wird gemessen
1	100	Prozent	Anzahl Schritte von der Home-Line
2	100	Prozent	Anzahl mögliche Schritte
3	Maximal	Punkte	Kann dieser Zug direkt Siegen
4	15	Punkte	Kann dieser Zug blocken
5.a	25	Punkte	Kann dieser Sumo-Push einen Sieg im nächsten Zug erringen
5.b	15	Punkte	Kann dieser Sumo-Push einen Block im nächsten Zug erringen

16.1.3 Steuerung

Single Player

Abbildung 43: Single Player

Soll ein Spiel gestartet werden, indem der Spieler gegen einen Computer-Gegner spielen soll, kann der Single-Player-Modus aktiviert werden, indem die Parameter beim Spieler zwei eingestellt werden und der Button „Single Player“ gedrückt wird.

Double AI – BetaVersion

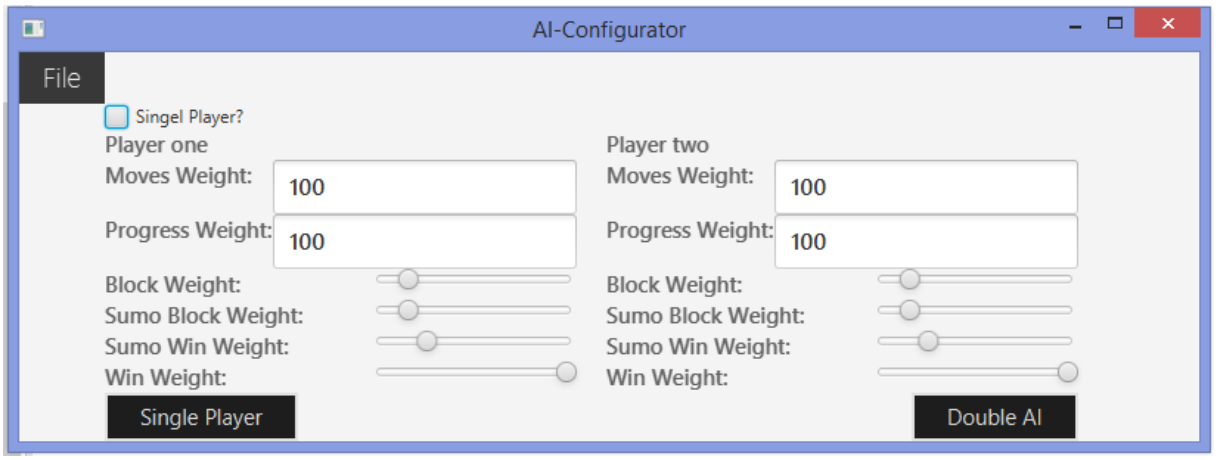


Abbildung 44: Double AI

Sollen zwei künstliche Intelligenzen gegen einander antreten, dann kann das Häkchen bei „Single Player“ entfernt werden, die Parameter bei Spieler eins und Spieler zwei eingestellt werden und auf den Button „Double AI“ gedrückt werden.

16.2 Client

Der „Client“ wird direkt zu Beginn des Spiels aufgerufen, damit eine Verbindung zum Server hergestellt werden kann.

Das Client-Menu bietet folgende Funktionen:

1. Eingabe der Host-Adresse
2. Eingabe des Portes
3. Log-Einträge werden für die Nachvollziehbarkeit der Übertragung im Fenster angezeigt
4. Chat-Nachrichten können gesendet werden, dies auch ohne Login, falls Probleme auftreten und man jemanden, welcher bereits im Spiel ist um Hilfe bitten möchte
5. Test-Modus für den „admin-User“. Dieser User kann diverse Nachrichten via Knopfdruck an den Server übermitteln um die Übertragungen nachzuverfolgen (Troubleshooting). Solange man nicht als „admin-User“ eingeloggt ist, werden diese Funktionen nicht freigeschalten
6. Die Verbindung kann sich „gemerkt“ werden, sodass beim nächsten Start des Programmes direkt auf eine Verbindung aufgebaut wird.

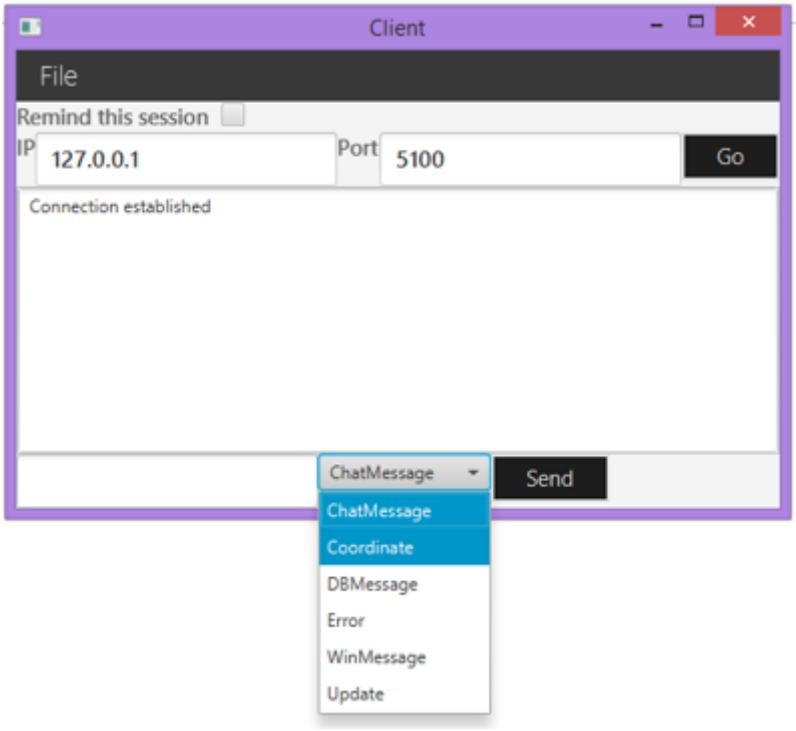


Abbildung 45: Client-Menu, Ansicht als "admin-User" mit allen Message-Typen

16.3 User Menu

Das User Menu kann aus dem Login Menu aufgerufen werden und ermöglicht die Abfrage und Erfassung von Spielern. Man kann Spieler mit dem Vornamen, Nachnamen und Usernamen suchen. Wenn ein User Fokussiert wurde kann dieser mit dem Button „Change“ geändert werden, oder mit dem Button „Delete“ gelöscht werden. Um Änderungen beim User vorzunehmen müssen die Änderungen entsprechend in die Felder eingetragen werden.

Der Button „Search All“ zeigt alle Einträge in der Datenbank an.

Damit etwas geändert werden kann, muss zwingend das Passwort (Old Password) richtig eingegeben sein, ansonsten dürfen keine Änderungen vorgenommen werden.

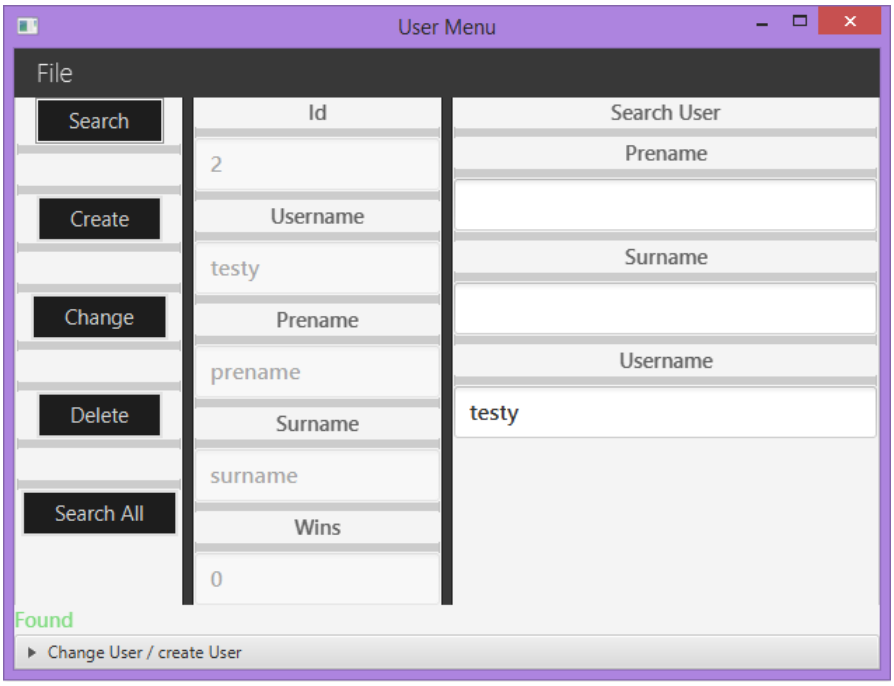


Abbildung 46: Suchen nach dem Spieler "testy"

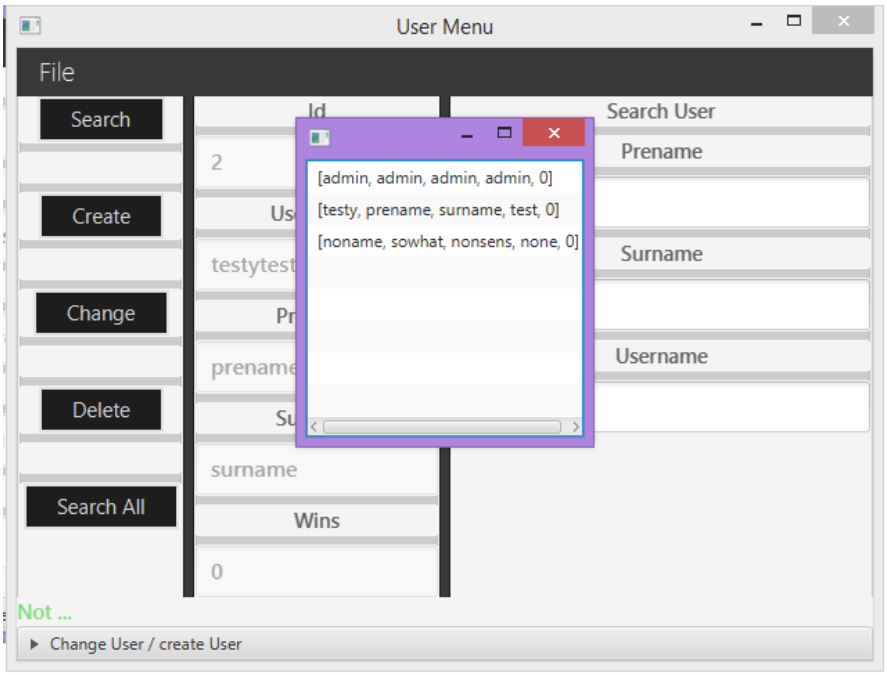


Abbildung 47: Alle Spieler anzeigen

Sollte ein neuer User erzeugt werden, kann dies mit dem Button „Create“ gemacht werden. Dafür müssen die Felder ausgefüllt sein.

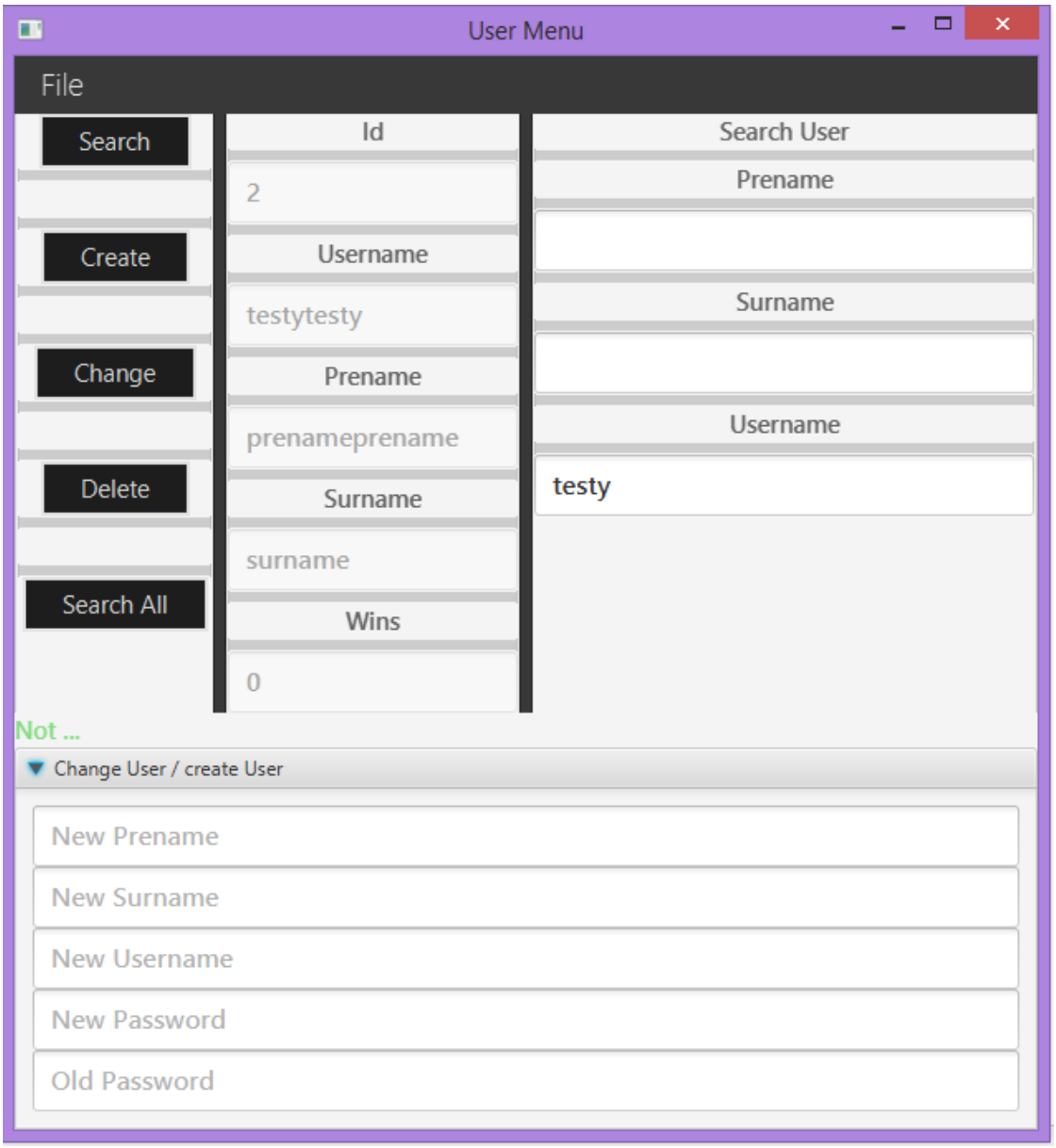


Abbildung 48: Das Menu der Felder für Änderungen oder Neuerzeugungen ist aufgeklappt

16.4 Login

Das Login Menu ist dazu hier sich mit seinem Benutzer anzumelden, damit der Benutzer identifiziert werden kann und auch seine Punkte entsprechend vergeben werden können.

Die Funktionen des Login Menu:

1. Überprüft ob der Benutzer bereits in der Datenbank vorhanden ist oder nicht. Vorhandene werden grün angezeigt, nicht vorhandene werden rot angezeigt. Ein Login ist nur möglich, wenn der Benutzer auch angelegt ist
2. Der Button „Refresh“ ermöglicht eine komplette Aktualisierung der User-Daten von der Datenbank, sollte etwas nicht funktionieren respektive nicht aktuell sein
3. Der Button „User Menu“ ruft das User Menu auf
4. Im Menu kann das Client Menu wieder aufgerufen werden, sofern dieses gebraucht wird
5. Nach drei falschen Versuchen das Passwort einzugeben, wird der Account gesperrt. Um diesen wieder zu aktivieren, muss der Administrator kontaktiert werden

Über den Button „Without Server“ kann ein Spiel-Modus eröffnet werden indem zwei Spieler auf demselben Computer gegeneinander antreten können. Hierzu sollte keine Server-Verbindung hergestellt sein, so wie es der Name des Buttons bereits verrät.

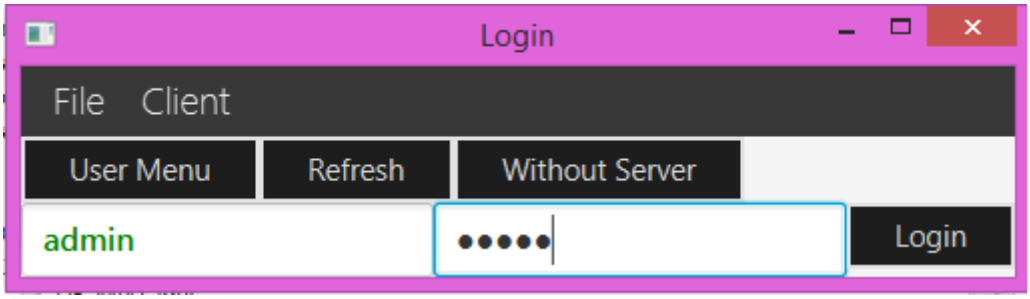


Abbildung 49: Login Menu mit korrekter Eingabe eines aktiven Benutzers

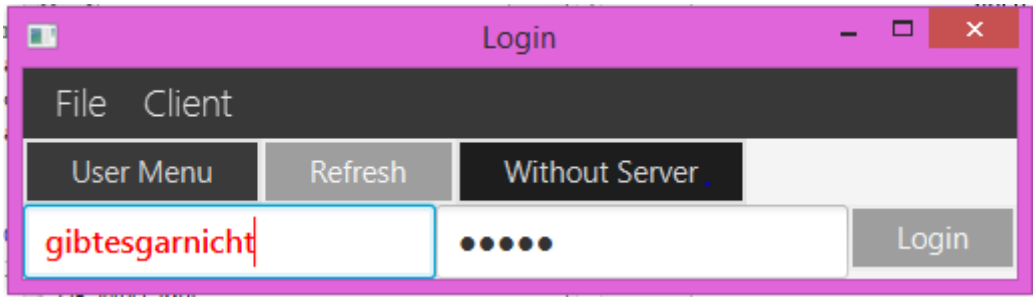


Abbildung 50: Login Menu mit eingabe eines Benutzer, den es nicht gibt

16.5 Spiel

Das Hauptmenu beinhaltet das Spielbrett, sowie einen Chat. Hier treten die Spieler gegeneinander an. Im Grund-Modus wird auf demselben Computer gegeneinander gespielt. Dieses wird gestartet indem der Button „Without Server“ im Login gedrückt wird. Es gibt weitere Spiel-Modi’s, welche im Haupt-Menu genauer erklärt werden.

In den Menus können folgende weitere Menus aufgerufen werden:

1. File
    a. Restart – Startet das Spiel neu
    b. Language – Ermöglicht eine Änderung der Sprache, die Einstellung wird sich gemerkt
    c. Friends – Startet das Friends Menu
    d. Main Menu – Startet das Main Menu
    e. Exit – Zurück kehren zum Haupt-Menu
2. Help
    a. Help about – Version und Hersteller Informationen
    b. Tutorial – Startet das Tutorial
3. Client
    a. Get Client View – Holt die Client GUI
4. AI Controller
    a. AI Controller – Startet das AI Menu

Auf der Seite werden die jeweiligen Spieler mit ihrem Punktestand angezeigt. Es ist möglich ein Timer einzustellen in Minuten. Sollte der Timer abgelaufen sein, wird das Spiel beendet. Dies ist vor allem für kurze Spiele gedacht, sodass auf Zeit gespielt wird.

Klassisch wird, wie im Tutorial angegeben, auf Runden gespielt. Sobald die angegebene Zahl Runden erreicht wurde, wird der Gewinner ermittelt.

Diese Möglichkeit besteht nur im Freunde-Modus, da in den anderen Spiel-Modi’s die Rundenanzahl vorgegeben wird.

Abbildung 51: Spiel mit Chat

Das Spiel selbst bietet folgende Möglichkeiten:

- Nur ein Spieler ist am Zug und kann wählen.
- Nur der Turm mit der Farbe, auf dem der gegnerische Turm stehen geblieben ist darf den nächsten Zug machen
- Es werden die möglichen Züge anzeigt, indem die Felder aufgeschaltet werden
- Ist es eine Patt-Situation wird dies dem Spieler anzeigt und der andere Spieler ist am Zug (mit derselben Farbe)
- Gewinne werden angezeigt und der Spieler hat beim Spielstart die Wahl zwischen einem links Auffüllen oder einem rechts Auffüllen
- Die Sumo Towers werden mit Zahlen gekennzeichnet, sodass sofort erkennbar ist wie „stark“ der Sumo Tower ist
- Die Sumo Tower können nur entsprechend ihrer „stärke“ bewegt werden (limitierter Bewegungsumfang)

16.6 Tutorial

Das Tutorial führt den Spieler durch die verschiedenen Regeln des Kamisados. Dazu können die Spielvarianten

- Basic Rules
- Long Play
- Marathon Play

angeschaut werden.

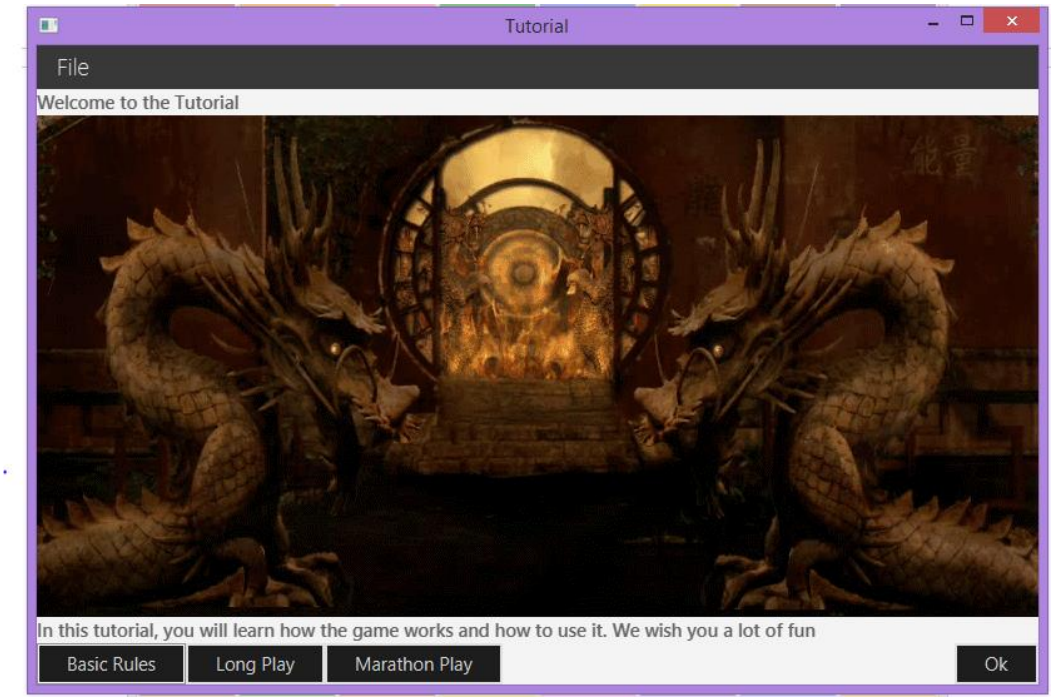


Abbildung 52: Tutorial Haupt Menu

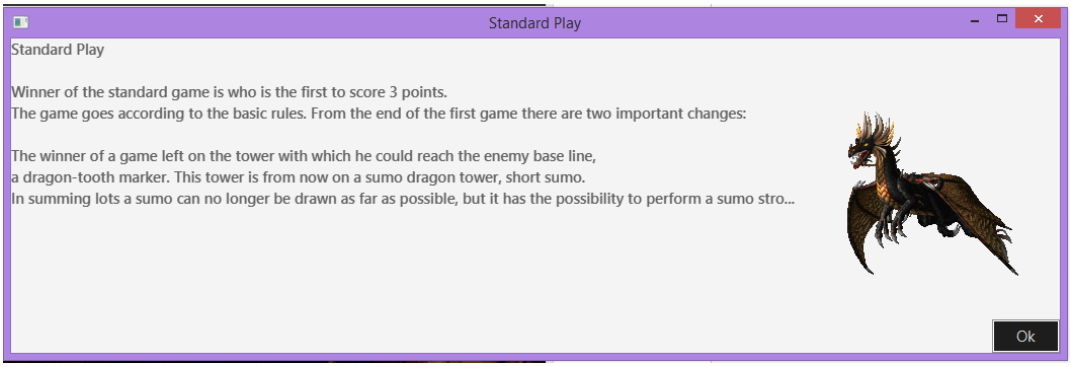


Abbildung 53: Tutorial Basic Rules

16.7 Haupt-Menu

Das Haupt-Menu ist das Menu, aus dem die verschiedenen Game-Modis gestartet werden können. Ebenfalls ist das Haupt-Menu der zentrale Punkt des Spiels, hier kehr man auch zurück, wenn man das Brettspiel selbst verlässt.

Es gibt hier folgende Auswahl:

- Single Game: Eine einfache Runde gegen den KI-Spieler
- Standard Game: 3 Runden werden gespielt gegen den KI-Spieler
- Long Game: 7 Runden werden gespielt gegen den KI-Spieler
- Marathon Game: 15 Runden werden gespielt gegen den KI-Spieler
- Friends: Ein Spiel gegen einen Freund wird gestaret
- End Game: Beendet das Spiel komplett.

Durch das Menu File >> Exit kehr man zurück zum Login, sollte man 2 Users erfasst haben ist es so möglich diesen so zu ändern, ohne aus dem Spiel auszusteigen.

Abbildung 54: Haupt-Menu

16.8 Friends-Menu

Im Friends-Menu kann der Spieler erfasste Users als Freunde anfragen oder sollte bereits eine Anfrage gekommen sein, diese annehmen, sodass gemeinsame Speile gestartet werden können.

Unter dem Menu File können folgende Punkte ausgewählt werden:

- Add Friends: Startet die Add Friends Ansicht um neue Freunde anzufragen
- Exit: Zurück zum Hauptmenu

Wenn ein Freund hinzugefügt ist hat der Spieler zwei Möglichkeiten:

- Remove: Entfernt den Freund bei beiden Spielern. Es ist möglich diesen Freund erneut anzufragen, dies muss aber wieder quittiert werden
- Play: Ein Spiel im Freunde-Modus zu starten

Wenn eine Anfrage eines Freundes besteht hat der Spieler zwei Möglichkeiten:

- Accept: Nimmt die Anfrage an und dieser Spieler wird jetzt als Freund angezeigt. Ebenso geschieht dies beim Spieler, welcher die Anfrage gesendet hat
- Refuse: Die Anfrage wird abgelehnt.

Anfragen sieht nur der Spieler, welcher angefragt wurde.

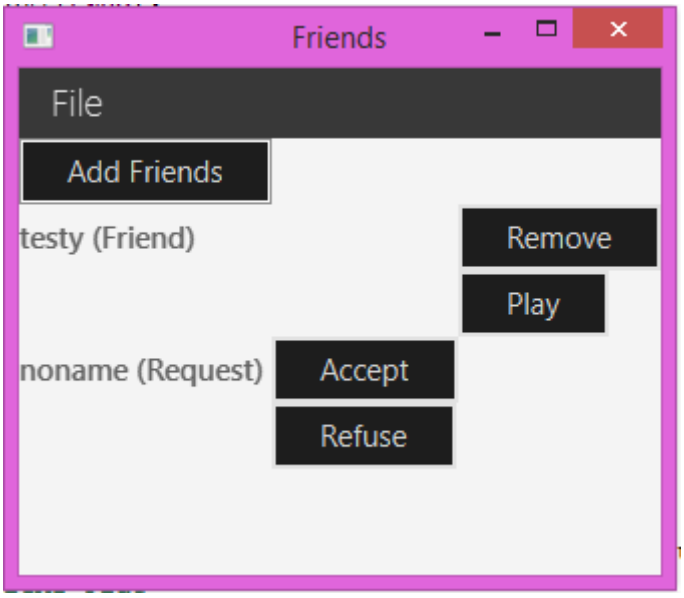


Abbildung 55: Friends-Menu mit Freund und pendenter Anfrage

16.8.1 Add Friends

Add Friends zeigt die Spieler, welche sich einen Account angelegt haben. Diese können durch das Suchfeld gefiltert werden.

Mit dem Add Friends-Button wird eine Anfrage an den entsprechenden Spieler gesendet.

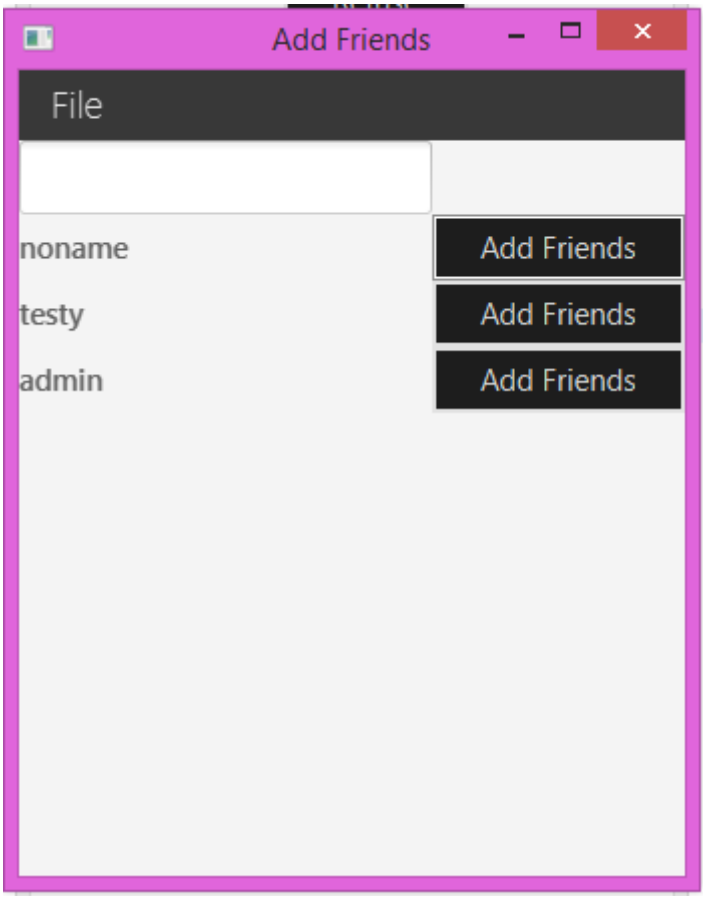


Abbildung 56: Add Friends zeigt die angemeldeten Speiler

16.9 Lobby

Die Lobby ist hier um auf einen Freund zu warten, mit dem man ein Spiel spielen möchte.

Die Lobby bietet folgende Funktionen:

- Wenn beide Spieler in der Lobby sind, kann in der Lobby untereinander gechattet werden. Dieser Chat wird nicht in den allgemeinen Chat übertragen. Der allgemeine Chat hingegen wird im Lobby-Chat angezeigt, sodass die beiden Spieler weiterhin die Informationen der anderen erhalten. Sollten die beiden Spieler in den allgemeinen Chat schreiben wollen, müssten sie via Menu den Client aufrufen und über diesen chatten.
- Sind beide Spieler in der Lobby angelangt (wichtig), und drücken dann beide Spieler auf dem Button „Go“, wird das Spiel gestartet. Es ist möglich sich wieder um zu entscheiden und die Bereitschaft zurück zu nehmen, sofern der Button „Go“ erneut gedrückt wird. Der Go-Button sollte erst gedrückt werden, wenn beide Spieler auch in der Lobby angekommen sind.
- Durch den Menu-Punkt „Exit“ wird das Haupt-Menu aufgerufen

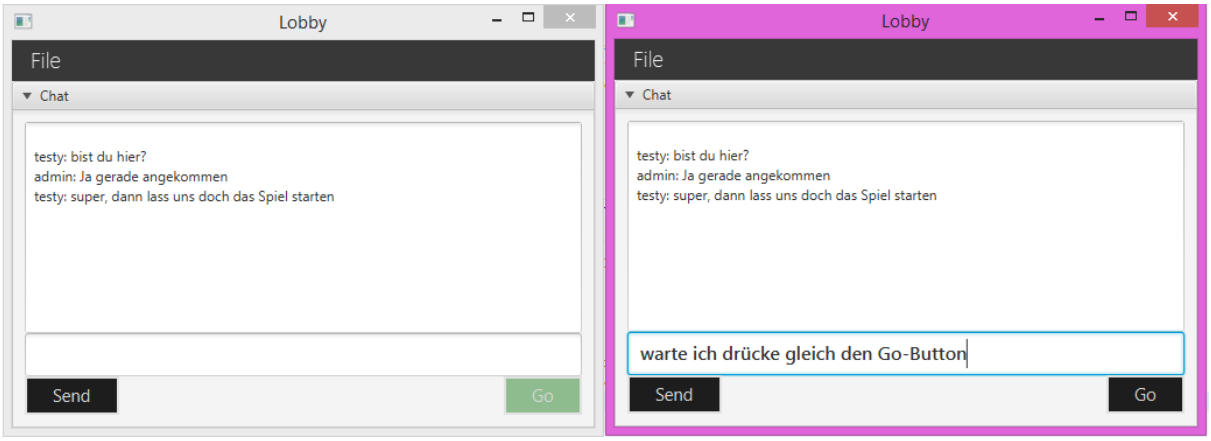


Abbildung 57: Zwei Lobby's in denen miteinander gechattet wird. Der einte Spieler hat bereits das Go gegeben

16.10 Server

Der Server wird als einfaches Feld anzeigt und in diesem wird das Log angezeigt.

Mit dem Button „Go“ wird die Verbindung auf dem angegebenen Port (TextFeld neben Port) geöffnet. Der Server wartet nun auf einen Client, welcher sich verbinden will und handelt die Verbindung.

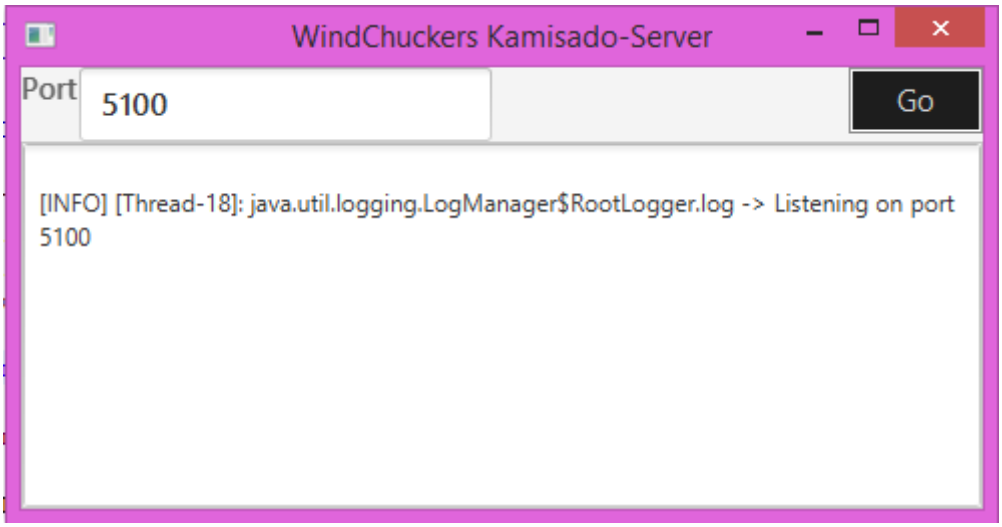


Abbildung 58: Einfaches Server-Feld

Der Server selbst verwaltet die eingehenden Meldungen in Form von XML und sendet Meldungen im Format von XML an den Client zurück. Weiteres was durch den Server verwaltet wird:

- Komplette AI
- Komplette Datenbank
- Die Koordination für das Starten eines Freunde-Spiels aus der Lobby

17 Old GUI-Mockup's

Da dieses Dokument sowohl als Requirement, als auch als Dokumentation dient, wurden die alten GUI-Mockup's wie sie ganz am Anfang in der Planungsphase angedacht wurden hier in den Anhang eingefügt, damit der Leser die Entwicklung des Spiels verfolgen kann.

17.1 Login

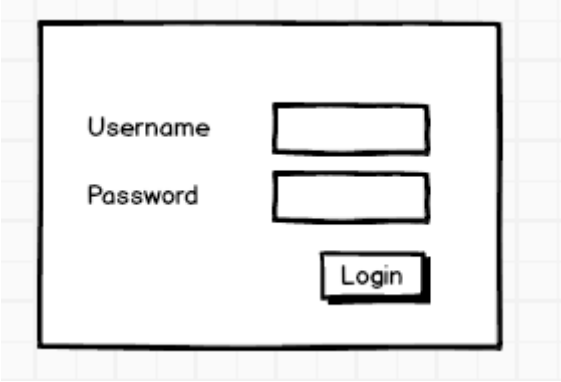


Abbildung 59: GUI-Mockup Login old

17.2 Lobby

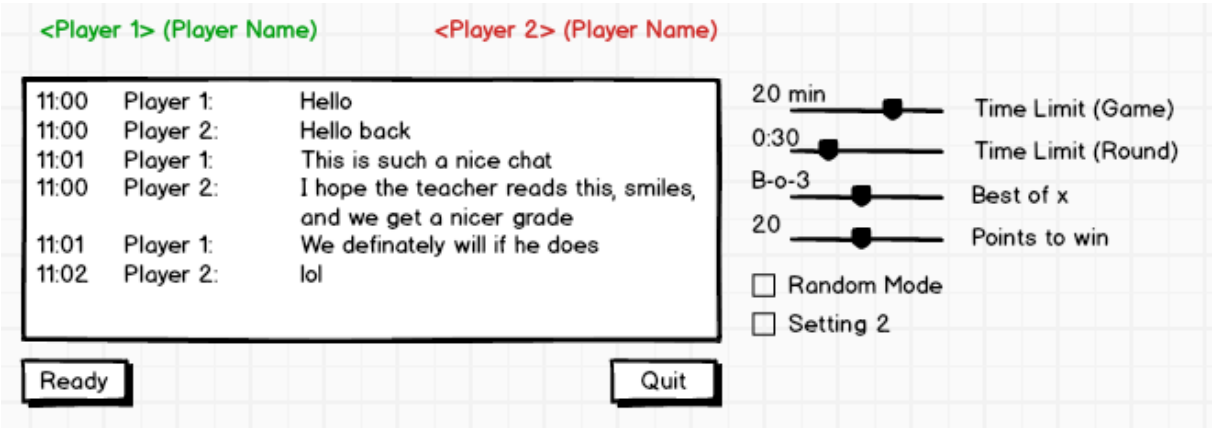


Abbildung 60: GUI-Mockup Lobby old

17.3 Main Menu

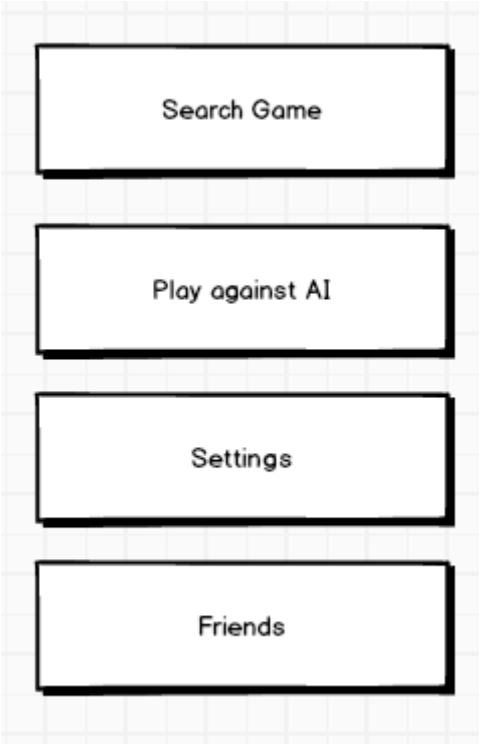


Abbildung 61: GUI-Mockup Main Menu old

17.4 Friends Menu

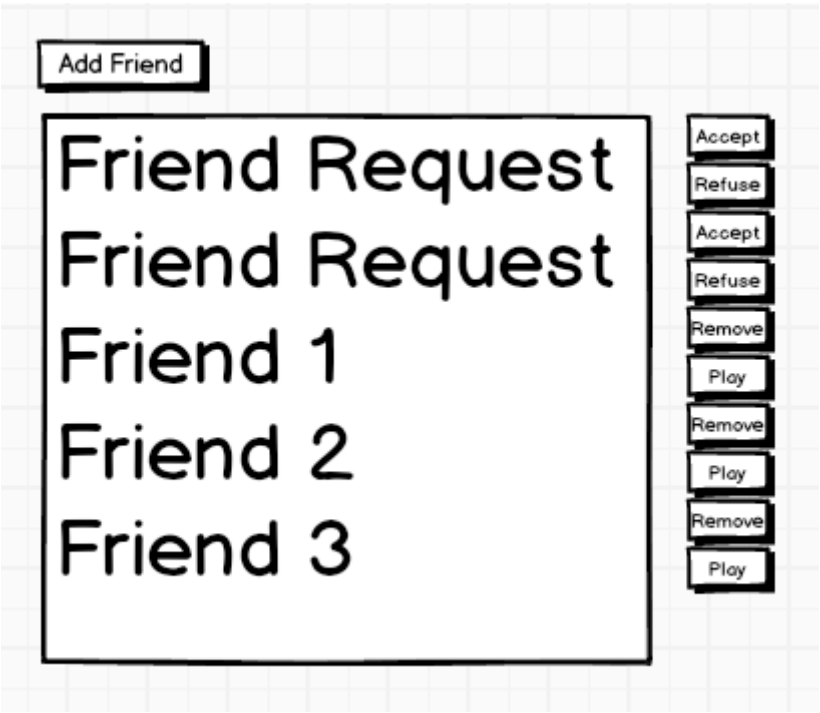


Abbildung 62: GUI-Mockup Friends Menu old

17.5 Add Friends

Abbildung 63: GUI-Mockup Add Friends old

17.6 Game Menu

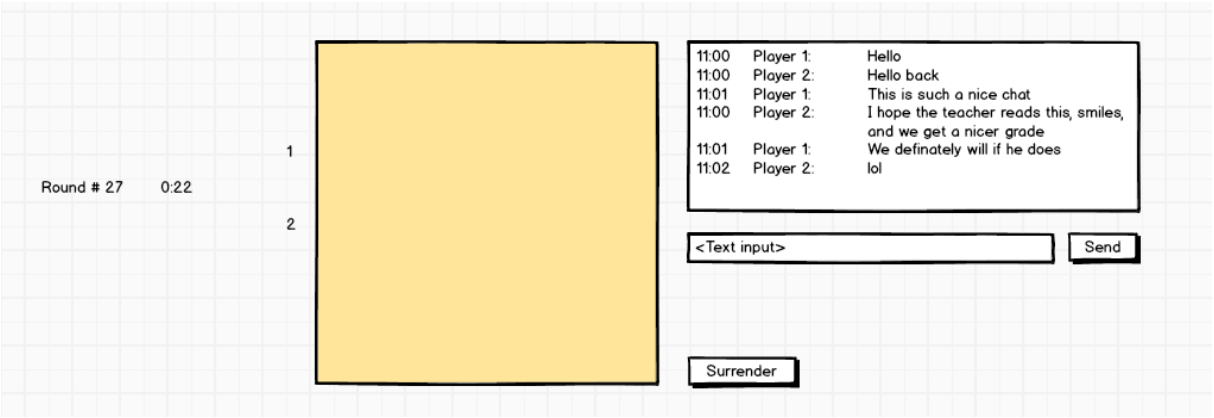


Abbildung 64: GUI-Mockup Game Menu old