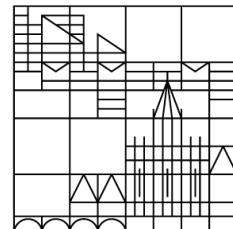


UNIVERSITY OF CONSTANCE

Software Project SS21

Universität
Konstanz



GAME DESIGN DOCUMENT

Group 03

Maria Fernanda Barrios Herrera

Jan de Boer

Andreas Kahabka

Dennis Römmich

Maik Tietz

Supervisor: Jason Oliver

May 7, 2021

Inhaltsverzeichnis

1 Änderungsliste	4
2 Übersicht	5
3 Zeitplan	5
3.1 Beginn	5
3.2 Meilenstein 1	5
3.3 Meilenstein 2	6
3.4 Meilenstein 3 und 4	6
3.5 Übersicht	7
4 Framework	8
4.1 Gamecontroller, Player	8
4.1.1 Class Gamecontroller	8
4.1.2 Spielmenü	8
4.1.3 Netzwerk-Funktion	9
4.1.4 Bibliotheken	9
5 Spiele	10
5.1 Schach	10
5.1.1 Regeln	10
5.1.2 Anforderungsanalyse	14
5.1.3 Das Benutzerinterface	15
5.1.4 Die Software-Architektur	18
5.1.5 Bildquellen	19
5.2 Rummikub	20
5.2.1 Grundsätzliches	20
5.2.2 Spielvorbereitung	20
5.2.3 Spielablauf	21
5.2.4 Anforderungsanalyse	23
5.2.5 Das Benutzerinterface	24
5.2.6 Die Software-Architektur	28
5.2.7 Bildquellen	29
5.3 Skat	30
5.3.1 Regeln und Spielablauf	30
5.3.2 Design & Architektur	33
5.3.3 Anforderungen & Anspruch	39
5.4 Die Siedler von Konstanz	40
5.4.1 Einführung	40
5.4.2 Spielelemente	40
5.4.3 Spielvorbereitung	41

5.4.4 Spielblauf	41
5.4.5 Bauregeln	42
5.4.6 Spielende	44
5.4.7 Anforderungen	44
5.4.8 Benutzeroberfläche	44
5.4.9 Formale Anforderungen	45
5.4.10 Software Architektur	45
5.5 TicTacToe	47
6 Design, Architektur and technische Anforderungen	47
7 Benutzeroberfläche	47
8 Anforderungen an die Software	48

1 Änderungsliste

Bis zur finalen Version dieses GDDs werden noch einige Veränderungen vorgenommen werden. Diese Änderungen werden laufend an dieser Stelle festgehalten.

- **10. Mai**

Abgabe der Final-Version des GDDs in den Zeitplan integriert.

- **3. Juli**

Schach Spielende Definition angepasst, Benutzerinterface angepasst, Software-Architektur geändert und Bilder eingefügt.

- **4. Juli**

Aktualisierung des Unterkapitels "Skat":

Kleine Änderungen in der ursprünglich vorgesehenen Spielweise und Aktualisierung der Architektur.

Aktualisierung und Konkretisierung des Kapitels "Framework".

Aktualisierung des Kapitels Übersicht

Aktualisierung und Konkretisierung des Unterkapitels "Die Siedler von Konstanz":

Entwurf des konkreten Spielablaufs. Aufteilung des Spiels in Komponenten. Änderung der Priorisierung: Räuber statt Spielerhandel als Haupt-Feature.

2 Übersicht

Die Software stellt eine Spielesammlung aus den 4 Spielen "Schach", "Rummikub", "Skat" und "Die Siedler von Konstanz" ("Die Siedler von Catan") dar. Mit "Schach" als strategisches Brettspiel, "Rummikub" als Legespiel, "Skat" als Kartenspiel und "Die Siedler von Konstanz" als Fantasie-Brettspiel ist eine große Diversität geboten.

Alle Spiele, bis auf Schach, welches in der Konsole gespielt wird, werden über ein *Graphical User Interface* gespielt und je nach Spiel von 2-4 Spielern spielbar sein.

3 Zeitplan

3.1 Beginn

Die Arbeit an der Software beginnt mit Woche 1 am Montag, den 10. Mai 2021. Dann wird in kleinen Gruppen zu zweit oder dritt an Teilprojekten gearbeitet. Ein bis zwei Mal pro Woche werden untereinander die Fortschritte und Errungenschaften präsentiert und Ideen und Vorschläge ausgetauscht, besprochen und diskutiert.

Das Spiel "Schach" und auch das Spiel "Rummikub" beginnen implementiert zu werden. Ebenso das Menü, als GUI, welches am Ende alle 4 Spiele auf einer Oberfläche bündeln wird.

3.2 Meilenstein 1

Meilenstein 1 ist Ende der vierten Woche am 13. Juni fällig. Bis dahin sind bereits die beiden Spiele "Schach" und "Rummikub" vollständig spielbar und auch das Grundgerüst der Gesamtanwendung funktioniert. Der Fokus liegt am Anfang in erster Linie auf "Schach", sodass bei Komplikationen auf jeden Fall ein Spiel für den ersten Meilenstein fertig zur Abgabe ist.

Durch die fast halbwöchentlich Treffen der Gruppe und das Nutzen verschiedener Organisations-Tools herrscht eine gute Kommunikation, weshalb bereits eine Woche vor Deadline der erste Meilenstein nahezu abgabebereit sein soll. Diese Woche wird genutzt um erneut sämtliche Code-Stücke zu reviewen und zu korrigieren, sodass mögliche übersehene Fehler oder Ungenauigkeiten entdeckt werden können.

3.3 Meilenstein 2

Zwischen den Meilensteinen 1 und 2 findet die erste Vorstellung der Spielesammlung statt. Präsentiert wird der aktuelle Stand des GUI mit den fertigen Spielen "Schach" und "Rummikub".

Gleichzeitig wird mit der Implementierung von "Skat" begonnen, um Zeitrückstände zu vermeiden. Sobald die Abgabe für den Milestone 1 getätigt ist, wird auch mit dem vierten und letzten Spiel begonnen: "Die Siedler von Konstanz" ist das komplexeste der Spiele, weshalb für dieses am meisten Zeit eingeplant wird. In den Wochen bis zum zweiten Meilenstein werden bei den bereits fertigen Spielen die nötigen Features implementiert, falls das nicht schon geschehen sein sollte, um den Meilenstein 2 bereits früh abgeschlossen zu haben und auch hier nicht unter Zeitdruck zu geraten. Auch bei diesem Meilenstein wird die letzte Woche genutzt um den geschriebenen Code zu lesen und zu optimieren.

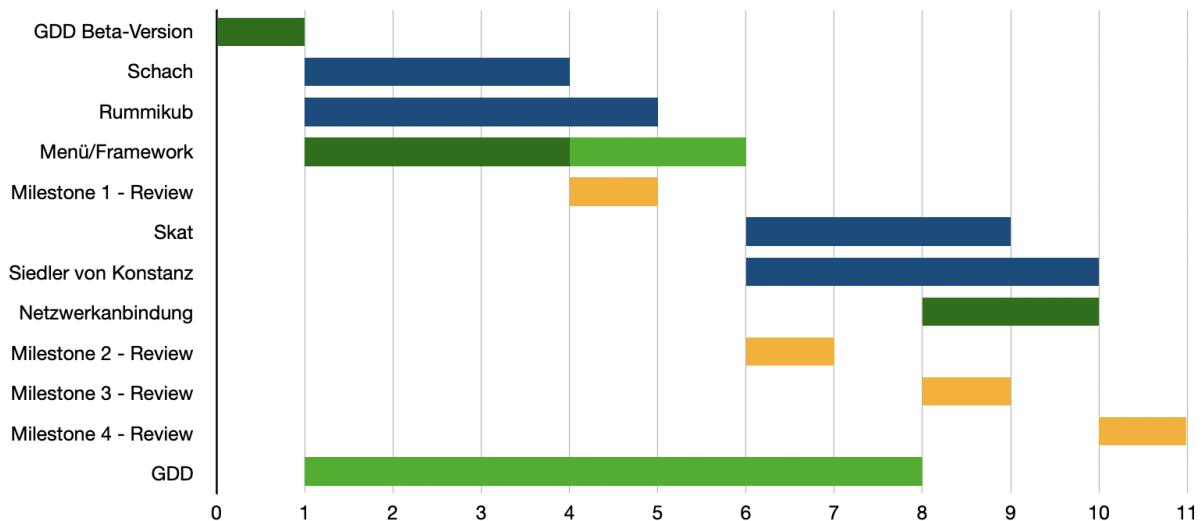
Außerdem wird das GDD, welches laufend aktualisiert wurde, erneut korrigiert und sowohl inhaltlich als auch ästhetisch fertiggestellt.

3.4 Meilenstein 3 und 4

Bis zum Meilenstein 3 am 11. Juli wird intensiv an den Spielen "Skat" und "Die Siedler von Konstanz" gearbeitet. Sämtliche Features sind in allen Spielen fertiggestellt bzw. werden bald fertiggestellt sein. Die endgültige Frist rückt näher und das Feature Netzwerkanbindung wird in Angriff genommen um eine gute Benotung zu erzielen. Geplant ist, diese in vorerst ein Spiel zu integrieren um dieses auch online spielen zu können.

In den letzten Wochen werden erneut sämtliche Code-Abschnitte reviewed, korrigiert, optimiert und verschönert, sodass alle Spiele korrekt in das Framework eingebunden und von Menü aus spielbar sein werden. Am 25. Juli 2021 wird letztendlich die finale Version der Spielesammlung präsentiert.

3.5 Übersicht



4 Framework

4.1 Gamecontroller, Player

Eine abstrakte Klasse GameController und das Interface Player und bilden den Rahmen für die implementierung der einzelnen Spiele.

4.1.1 Class GameController

Diese Klasse ist die Superklasse der Controller-Klassen die die Kommunikation zwischen den einzelnen Komponenten der Spiele übernehmen und dafür sorgen, dass Zugzurücknahme, das Starten der Spiele mit einer festen Zugreihenfolge und das Vereinheitlichen von Zügen von GUI, KI und über Netzwerk möglich ist.

Attribute:

- List<String> moveHistory

Methoden:

- init() [abstract]
- init(List<String> moves) [abstract]
- void logMove(String move) [vorimplementiert]
- void makeMove(String move) [abstract]
- void takeBackMove() [vorimplementiert]
- String getSettings() [abstract]
- void setSettings() [abstract]

4.1.2 Spielmenü

Es wird ein Spielmenü mit GUI geben, aus dem alle Spiele bequem gestartet werden könnenn.

4.1.3 Netzwerk-Funktion

Die Funktionalität von (Online-)Multiplayer wird auch über das Framework implementiert werden.

Hierfür wird die Class NetworkController zuständig sein. Wie werden *Hamachi* benutzen um ein lokales Netzwerk zu simulieren, sodass auch das Spielen mit Spielern möglich sein wird, die sich nicht im gleichen Netzwerk befinden.

4.1.4 Bibliotheken

- [JSONsimple](#)

Diese Bibliothek ermöglicht das einfache verwenden von json-Objekten, die unter anderem dazu benötigt werden, Spielzüge und damit Spiele abspeichern zu können.

- [JavaFX](#)

Diese Bibliothek bietet umfangreiche Möglichkeiten unter Java eine professionelle GUI zu erstellen.

5 Spiele

5.1 Schach

5.1.1 Regeln

Das Gesellschaftsspiel Schach ist ein Zweispieler Spiel, welches auf einem Schachbrett mit 8x8 Feldern und den darauf befindlichen 2x16 Schachfiguren ausgetragen wird.

Jeder Spieler wird durch die Farbe Schwarz bzw. Farbe Weiß repräsentiert und kontrolliert die Figuren in der jeweiligen Farbe. Dabei hat jeder Spieler zu Beginn des Spiels die gleichen 16 Figuren zur Verfügung.

Die Spieler sind abwechselnd am Zug und versuchen das Spiel zu gewinnen, indem sie die gegnerische Königs-Figur Schachmatt setzen, d.h. den König so in die Enge zu treiben, dass er aus dem gegnerischen Angriffsradius nicht mehr entkommt und somit der gegnerische Spieler keinen gültigen Spielzug mehr machen kann.

Um dieses Ziel zu erreichen, kontrollieren beide Spieler 16 Spielfiguren, welche unterschiedliche Zugmuster und Startpositionen aufweisen.



Ein typisches 8x8 Schachbrett

Im Folgenden werden die Figuren und ihre Zugmuster genauer erläutert:

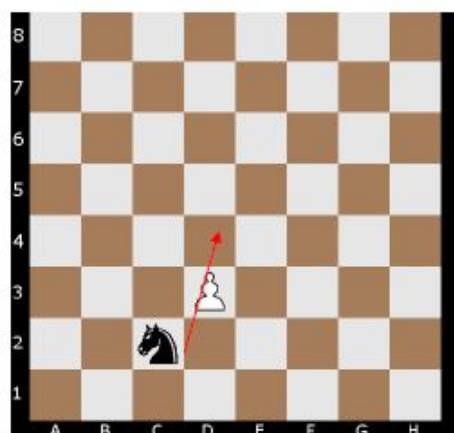
- **Der Bauer**

Jeder Spieler beginnt das Spiel mit 8 Bauern, welche die gesamte Reihe 2, bzw. Reihe 7 füllen. Jeder Bauer darf in seinem allerersten Zug ein oder zwei Felder nach vorne bewegt werden. In jedem weiteren Zug jedoch nur einen einzigen Zug. Befindet sich auf dem zu ziehenden Feld eine andere Figur kann der Bauer nicht dorthin gezogen werden.

Der Bauer kann jedoch ein Feld diagonal nach links oder rechts eine gegnerische Figur schlagen, d.h. aus dem Spiel entfernen, sofern sich dort eine gegnerische Figur befindet. Damit ist der Bauer die einzige Figur, die nicht mit ihrem normalen Laufmuster gegnerische Figuren schlagen kann.

- **Der Springer**

Jeder Spieler beginnt mit 2 Springern auf den Feldern B1 und G1 bzw. B8 und G8. Der Springer bewegt und schlägt gleichzeitig. Er darf auf eines der Felder ziehen, die seinem Standfeld am nächsten, aber nicht auf gleicher Reihe, Linie oder Diagonale mit diesem liegen (deutsche FIDE Beschreibung) - sprich: Er bewegt sich ein Schritt nach vorne, hinten, links oder rechts und anschließen 2 Felder nach links oder rechts, aus der Richtung gesehen, aus der der Springer das erste Feld betreten hat. Dabei darf der Springer über andere (auch eigene Figur springen).



Ein möglicher Springerzug

- **Der Läufer**

Jeder Spieler beginnt mit 2 Läufern auf den Feldern C1 und F1 sowie C8 und F8. Der Läufer bewegt sich immer diagonal und darf dabei so viele Felder wie möglich in die festgelegte Zugrichtung passieren. Eine gegnerische Figur darf dabei geschlagen werden (der Läufer beendet auf der Position der geschlagenen Figur seinen Zug), die eigenen Figuren dürfen aber nicht passiert werden.

Die Feldfarbe, auf der sich der Springer befindet, legt auch gleichzeitig die Farbe fest, auf der dieser sich bewegen darf (Diagonale Felder haben alle dieselbe Farbe).

- **Der Turm**

Jeder Spieler beginnt mit 2 Türmen, welche sich auf den Feldern A1 und H1 sowie A8 und H8 befinden. Die Türme dürfen sich so viele Felder nach vorne, nach hinten, links oder rechts bewegen, wie passierbar sind, d.h. keine eigenen Figuren oder gegnerische Figuren stehen im Weg. Wenn eine Figur geschlagen wird, beendet der Turm seinen Zug an der Position der geschlagenen Figur.

- **Die Dame**

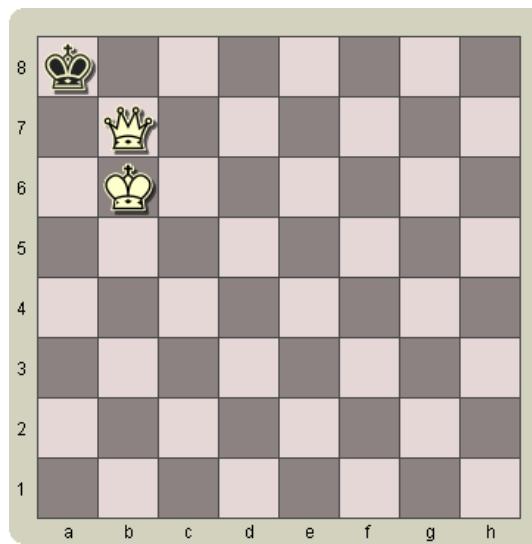
Jeder Spieler beginnt mit einer Dame auf Feld D1 sowie D8. Die Dame darf sich so viele Felder wie möglich sowohl diagonal, wie ein Läufer, als auch nach vorne, nach hinten, nach rechts und nach links, wie ein Turm, bewegen. Wenn eine Figur geschlagen wird, beendet die Dame ihren Zug an der Position der geschlagenen Figur.

- **Der König**

Jeder Spieler hat einen König auf den Feldern E1 bzw. E8. Der König darf nur ein Feld in jede Richtung tätigen (sowohl diagonal als auch, nach vorne, nach hinten und seitwärts). Der König kann auch Figuren schlagen. Diese Figur entscheidet maßgeblich über Sieg und Niederlage.

Befindet sich der König im sogenannten Schach, d.h. es besteht die Möglichkeit, dass er im nächsten Zug vom Gegner geschlagen werden kann, so muss dieser aus der Gefahrensituation gebracht werden (Wegschieben des Königs, Blocken mit anderen Figuren etc.). Falls dies nicht möglich ist, verliert der entsprechende Spieler, womit der andere Spieler ihn Schachmatt gesetzt hat und das Spiel gewinnt.

Es ist jedoch auch möglich, dass sich der König zwar nicht in einer Gefahrensituation befindet, jedoch keine zulässigen Spielzüge mehr für den Spieler möglich sind (König umringt von Feldern mit Schachmatt Gefahr, keine Figuren mehr und Bauern nicht Zug fähig, weil blockiert). Diese Spielposition nennt sich Patt und lässt das Spiel in einem Unentschieden enden.



Schachmatt mit nur einer Dame

Zusätzlich zu den normalen oben aufgeführten gibt es auch noch spezielle Spielzüge mit gewissen Bedingungen:

- **Die Rochade**

Ein Turm und ein König werden gleichzeitig bewegt und tauschen ihre Positionen. Dies ist nur möglich, wenn sowohl der jeweilige Turm als auch der König im bisherigen Spiel nicht bewegt wurden. Wenn der König mit dem näherliegenden Turm tauscht, so steht der König anschließend auf G1 bzw. G8 und der Turm auf C1 und C8.

- **Die Umwandlung**

Falls es ein Bauer auf ein Feld der letzten Reihe des Schachbrettes schafft (für Schwarz Reihe 1, für Weiß Reihe 8), so wird dieser in eine andere Figur (außer König) umgewandelt. Dabei entscheidet der Spieler, in welche Figur umgewandelt wird.

- **Die En-passant Regel**

Wenn ein Bauer in seinem ersten Zug 2 Felder nach vorne bewegt wird und dieser dadurch ein vom gegnerischen Bauern beherrschtes Feld übersprungen, so darf dieser gegnerische Bauer den 2 Felder Bauern so schlagen, als wäre er nur ein 1 Feld nach vorne bewegt worden. Dies darf er nur im direkt nächsten Zug ausführen, ansonsten verfällt diese Zugmöglichkeit.

Das Spielende:

Das Spiel ist zu Ende, wenn einer der Spieler Schachmatt gesetzt wurde, eine Patt Situation vorliegt (Spieler ist nicht im Schach aber hat keinen möglichen Zug) oder wenn die vorhanden Figuren kein Matt mehr erlauben (maximal ein Springer oder ein Läufer pro Spieler).

5.1.2 Anforderungsanalyse

Schach ist ein komplexes Gesellschaftsspiel, welches Figuren mit sehr unterschiedlichen Zugmustern aufweist. Es müssen diverse Eigenheiten der Figuren und spezielle Züge bedacht werden. Auch die Möglichkeit bei Verwandlungen zwischen Figuren zu wählen, muss entsprechend implementiert werden. Zudem soll nicht nur das Spiel zwischen menschlichen Spielern, sondern auch mit einer kompetenten KI möglich sein, welche einem den Sieg nicht ohne Weiteres überlässt.

Die Umsetzung einer guten KI ist hierbei besonders aufwendig. Die Entwicklung einer guten Schach KI war jahrzehntelang eines der beherrschenden Themen der KI Entwicklung und ist selbst nach den Erfolgen der KIs gegen menschliche Spieler noch immer ein beliebtes Forschungsgebiet. An traditionellen KI Systeme wie „Stockfish“ waren unzählige Entwickler beteiligt. Im Bereich des Machine Learning spielen Schach KIs ebenfalls eine große Rolle. So entwickelt bspw. Google die AlphaZero AI, welche es nicht nur mit menschlichen Spielern sondern auch mit traditioneller KI, wie dem eben erwähnten „Stockfish“ aufnehmen kann.

Aufgrund der genannten Punkte ist es sehr anspruchsvoll eine gute KI zu entwickeln, weshalb unsere KI nicht allzu stark sein wird. Unsere KI zielt deswegen vor allem darauf ab, möglichst wenige offensichtlich schlechte Züge zu spielen, sodass sie wenigstens im Duell mit Schachanfänger bestehen kann.

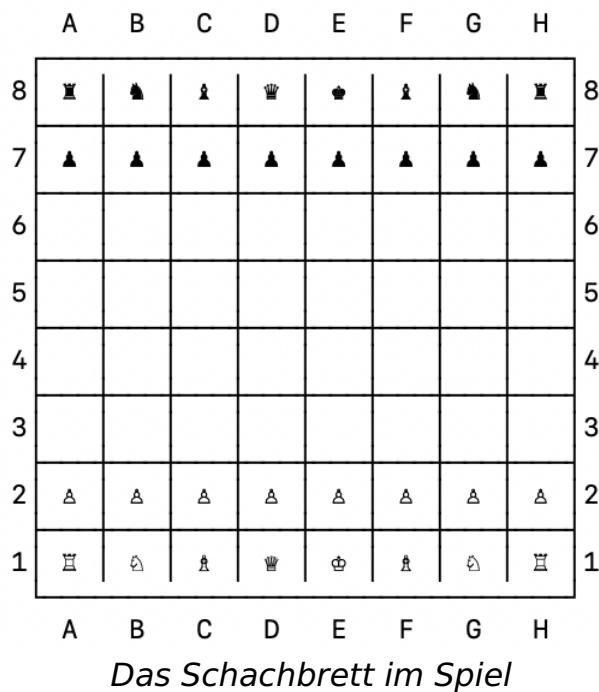
Des Weiteren ist die Implementierung einer Netzwerkfähigkeit vorgesehen, welche auch ein Spiel mit anderen Spielern im lokalen Netzwerk ermöglicht, was ebenfalls technisches Know How im Bereich der Netzwerkimplementierung voraussetzt.

Dennoch sind wir überzeugt dieses Spiel umsetzen zu können, da trotz höherer Komplexitätstiefe als beispielsweise Tic Tac Toe, die Implementation eines Schachspiels nicht wesentlich anspruchsvoller sein sollte, zumal es sich auch hier wieder um ein Grid-basiertes Spiel mit Zügen ist, in welchem sich Figu-

ren mit Restriktionen bewegen und es eine klar definierte Siegvoraussetzung gibt (König steht im Schach und hat keine freien Züge mehr).

Verschiedene Varianten (wie z.B. Torpedo Schach) bieten den Raum das Spiel beliebig zu erweitern.

5.1.3 Das Benutzerinterface



Nach Auswahl des Spiels Schach werden einem folgende Möglichkeiten geboten:

- Neues Spiel in klassischem Schach (Hotseat)
- Neues Spiel in klassischem Schach (vs KI)
- Neues Spiel in Torpedo Schach (Spielmodus mit Bauern, die in jedem Zug 2 Felder anstatt nur im ersten ziehen können. En Passant Regel bleibt bestehen.)
- Laden eines Spielstandes
- Statistik
- Figuren in Unicode oder Buchstaben
- Spiel beenden

In der Statistik werden Metastatistiken über die Anzahl ausgeführter Züge gezeigt.

Zusätzlich gibt es zur besseren Kompatibilität die Funktion die Schachfiguren als Buchstaben anstatt als Unicode Symbole anzeigen zu lassen.

	A	B	C	D	E	F	G	H	
8	R	N	B	Q	K	B	N	R	8
7	P	P	P	P	P	P	P	P	7
6									6
5									5
4									4
3									3
2	p	p	p	p	p	p	p	p	2
1	r	n	b	q	k	b	n	r	1

Das Schachbrett mit Buchstaben als Figuren

Sobald das Spiel beginnt, hat man Zugriff auf folgende Funktionen:

- Hilfe-Screen
- Auto-Verwandlung aus/an
- Zug zurücknehmen
- Spiel beenden

Sobald das Spiel zu Ende ist, wird das Spielergebnis ausgegeben und das Spiel beendet. Während des Spiels wird automatisch eine Save-Datei erzeugt (Autosave mit Datum), welche man zu Beginn des Spiels ausführen lassen kann. Falls das Spiel noch kein Ergebnis hat, kann dieses auch weitergespielt werden.

	A	B	C	D	E	F	G	H	
8	■		■	■	■	■	■	■	8
7	■	■	■			■	■	■	7
6			■	■					6
5					■				5
4			■		■				4
3						■			3
2	■	■	■	■		■	■	■	2
1	■	■	■	■	■	■	■	■	1
	A	B	C	D	E	F	G	H	

Eine mögliche Stellung im Spielverlauf

Sobald man am Zug ist, hat man folgende Möglichkeiten:

- Ausführung eines Zuges per offizieller algebraischer Kurzschreibweise.
Für das obige Spielfeld ist das:

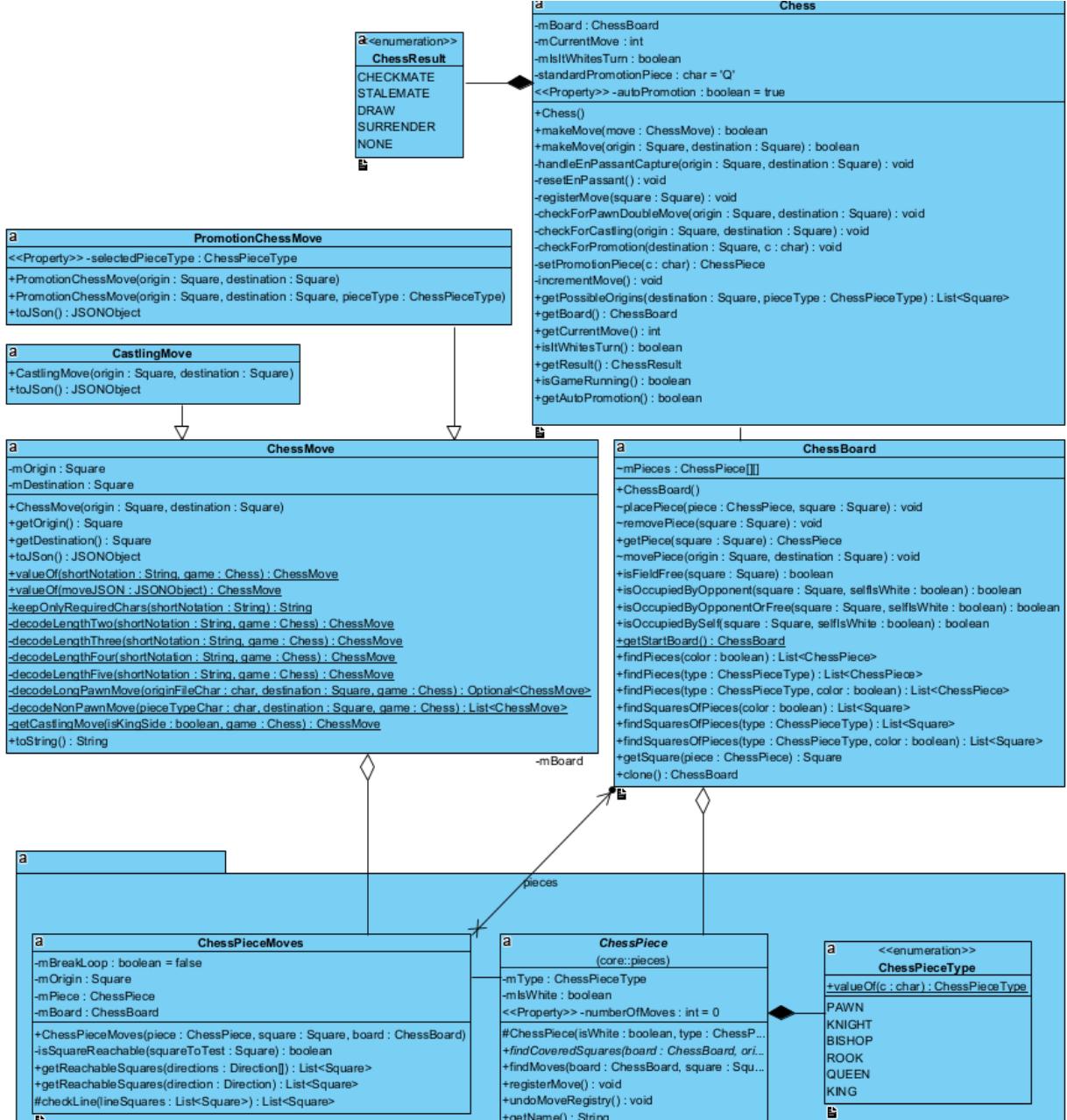
1. e4 - e5
2. nf3 - nc6
3. bc4 - d6
4. 0-0 (kurze Rochade)

Sonderzeichen für Schlagen (x) und Schachgeben (+) sind redundant und sind daher nicht erforderlich, jedoch erlaubt.

- Ausführung einer oben beschriebenen Funktion.

5.1.4 Die Software-Architektur

So sieht die Klassenstruktur für das Spiel aus:



5.1.5 Bildquellen

- <https://www.schachbund.de/downloads.html?file=files/dsb/srk/2019/FIDE-Regeln-2018-Final-DEU.pdf&cid=43465>.
- <https://www.kiknet-schachbund.org/unterrichtsmaterial/2-zyklus-3-zyklus/>
- <http://www.ich-lerne-schach.de/node/12/>

5.2 Rummikub

5.2.1 Grundsätzliches

Rummikub ist ein Spiel basierend auf Spielsteinen, das mit mindestens 2 und bis zu 4 Spielern gespielt werden kann. Es gibt 106 Spielsteine im Spiel, von denen 104 nummeriert und 2 Joker sind.

• Spielsteine

Die nummerierten Spielsteine haben einen Wert von 1 bis 13 und sind in 4 Farben unterteilt (blau, rot, gelb und schwarz), wobei es jeweils 2 Sets mit 13 Spielsteinen gibt. Jeder Spieler hat ein Sack zum Lagern der Spielsteine.

5.2.2 Spielvorbereitung

Die Spielsteine werden mit der Vorderseite nach unten auf dem Spielbrett verteilt. Jeder Spieler zieht einen Spielstein und deckt ihn auf. Nachdem alle Spieler einen Spielstein aufgedeckt haben, beginnt der Spieler, dessen Spielstein den höchsten Zahlenwert hat, das Spiel. Die Spielsteine kommen zurück in den Spielsteinstapel/Sack. Jeder Spieler erhält 14 zufällige Spielsteine, die

auf seinem Stapel angeordnet werden. Das Spiel beginnt mit dem Startspieler und wird im Uhrzeigersinn fortgesetzt.

5.2.3 Spielablauf

Damit ein Spieler einen ersten Zug machen kann, muss ein Set (oder Sets) mit einem Wert von mindestens 30 Punkten gespielt werden. Die Punktwerte werden vom Nennwert jedes gespielten Spielsteins genommen, wenn der Joker gespielt wurde, wird der Wert angenommen, von dem der Spielstein, an dessen Stelle der Joker verwendet wird, ausgeht. In diesem ersten Zug kann der Spieler keine Spielsteine eines anderen Spielers verwenden. Wenn der Spieler den ersten Zug nicht ausführen kann, muss er ein einzelnen Spielstein vom Spielsteinstapel nehmen und in seine Ablage legen. Wenn dies der Fall ist, ist sein Zug vorbei und der nächste Spieler ist am Zug.

Sobald ein Spieler seinen ersten Zug gemacht hat, kann er in seinem nächsten und den folgenden Zügen einen oder mehrere Spielsteine aus seinem Stapel ausspielen, um Gruppen zu bilden oder zu ergänzen. Wenn der Spieler keine Spielsteine ausspielen kann oder will, muss er den obersten Spielstein vom Spielsteinstapel nehmen und ihn zu seiner Ablage hinzufügen und damit seinen Zug beenden.

Wenn ein Spieler den ersten Zug nicht macht, kann er keine Spielsteine durch Hinzufügen oder Bilden von Gruppen spielen. Ein Spieler kann nur Spielsteine spielen, nachdem er den ersten Zug gemacht hat.

Die Spieler können Spielsteine spielen, indem sie sie zu bereits auf dem Spielbrett befindlichen Sets hinzufügen (entweder von ihnen selbst oder von anderen Spielern). Die einzige Grenze für die Länge eines Sets sind die Extremwerte der Spielsteine. Es kann nur 4er-Gruppen geben, da sich Farben innerhalb einer Gruppe nicht wiederholen dürfen.

Sätze

Alle Spielsteine im Spiel müssen in Sätzen von mindestens 3 Steinen angeordnet sein. Die 2 gültigen Satztypen werden Straßen und Gruppen genannt. Eine Straße besteht aus drei oder mehr gleichfarbigen Spielsteinen in fortlaufender Nummernreihenfolge. Eine Gruppe besteht aus drei oder vier gleichwertigen Spielsteinen in unterschiedlichen Farben.



Vorhandene Spielsteine manipulieren

Während des Zuges eines Spielers dürfen bereits gespielte Spielsteine in Sätzen manipuliert werden, damit weitere Steine gespielt werden können. Am Ende des Zuges müssen alle gespielten Spielsteine in gültigen Sätzen liegen.

- **Verschieben einer Straße**

Die Spieler können das entsprechende Plättchen an einem Ende einer Straße hinzufügen und einen Spielstein vom anderen Ende entfernen, um ihn anderswo zu verwenden.

- **Aufteilung einer Straße**

Die Spieler dürfen lange Straßen teilen und die entsprechenden Spielsteine in die Mitte legen.

- **Ersetzen in einer Gruppe**

Die Spieler können jeden beliebigen Spielstein in einer Gruppe mit drei Spielsteinen durch einen Spielstein der vierten Farbe und des gleichen Werts ersetzen.

- **Entfernen von Spielsteine**

Solange die verbleibenden Spielsteine eine gültige Straße bilden, können Spielsteine von den Enden der Läufe entfernt werden. Aus einer Gruppe mit vier Spielsteinen kann ein beliebiger Spielstein entfernt werden.

- **Joker-Ersatz**

Ein Joker kann vor dem ersten Zug zurückgeholt werden. Ein Joker kann aus einem Satz entnommen werden, indem er durch einen Spielstein

mit demselben Zahlenwert und derselben Farbe, die er repräsentiert, ersetzt wird. Der Joker kann aus der Ablage des Spielers oder vom Tisch genommen werden. Im Falle einer 3er-Gruppe kann der Joker durch den Spielstein einer der fehlenden Farben ersetzt werden. Ein ersetzter Joker muss im selben Zug des Spielers als Teil eines neuen Sets verwendet werden. Ein Satz, der einen Joker enthält, kann um Spielsteine erweitert, geteilt oder aus ihm entfernt werden. Der Joker kann auf jede mögliche Weise verschoben oder ersetzt werden, solange Sie einen Satz von mindestens drei Spielsteinen beibehalten.

• **Geerntete Fliesen**

Im Laufe einer Runde muss ein Spielstein, der aus einem bestehenden Satz "geerntet" wird, während des Zuges gespielt werden; er kann nicht zur späteren Verwendung aufbewahrt werden.

Ende des Spiels

Eine Runde endet, wenn ein Spieler keine Steine mehr hat. Die Werte aller Steine, die die anderen Spieler noch haben, werden addiert und dem jeweiligen Spieler als negative, dem Sieger der Runde als positive Punkte gutgeschrieben. Der Joker zählt dabei 30 Punkte. Der Spieler mit der höchsten Punktzahl gewinnt.

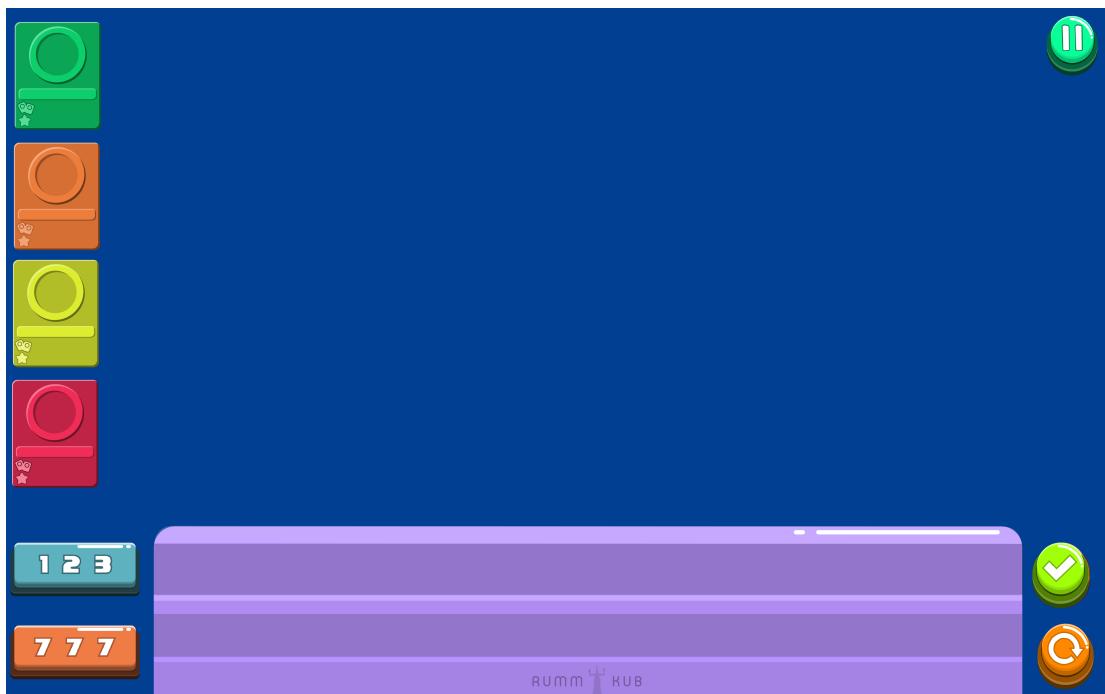
5.2.4 Anforderungsanalyse

Rummikub ist ein Spiel, das mindestens von 2 Spielern gespielt werden muss. Die Umsetzung für 4 macht es aus 2 Gründen anspruchsvoller: die Benutzeroberfläche und eine gute KI. Die verschiedenen Möglichkeiten, verschiedene Plättchen zu verwenden, die Implementierung, dass ein Spieler Straßen aufteilen kann und auch, dass ein Spieler gegen 3 Spieler spielt, die von der Maschine gesteuert werden, macht es zu einem anspruchsvollen Spiel.

Dieses Spiel wird mit der Option implementiert, mit anderen Spielern zu spielen, die über eine lokale Verbindung verbunden sind. Diese zusätzliche Implementierung erfordert viel Zeitaufwand für die Umsetzung.

Obwohl wir dieses Spiel als anspruchsvoll betrachten, haben wir den Anspruch, das Spiel so zu implementieren, dass es den Kunden überzeugt.

5.2.5 Das Benutzerinterface



Die Pause-Taste enthält einige der folgenden Optionen:

- Zurück zum Hauptmenü
- Aktuelles Spiel speichern
- Statistik
- Spiel beenden
- Neues Spiel starten
- Einstellungen

Beim Start eines neuen Spiels wird der/die Benutzer/in gefragt, ob er/sie gegen die Maschine oder gegen andere Spieler spielen möchte. Nach diesen beiden Optionen muss der/die Benutzer/in wählen, wie viele Spieler es geben soll. Danach beginnt das Spiel mit der eigentlichen Spieloberfläche

In der Statistikoberfläche kann der Benutzer/die Benutzerin die Top 10 der Spieler sehen, die dieses Spiel gespielt haben.

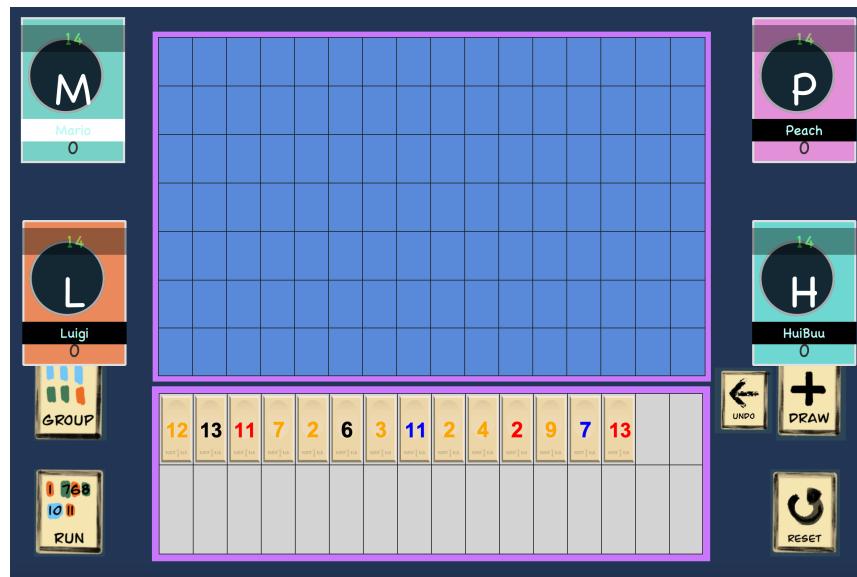
Die Spieloberfläche hat das Spielbrett in der Mitte, die Spieler werden jeder

Kante des Bildschirms zugewiesen, in einer Ecke befindet sich der Spielsteinstapel und in der rechten oberen Ecke befindet sich die Schaltfläche für das In-Game-Menü. Dieses Menü enthält die folgenden Optionen:

- Aufgeben
- Spiel speichern
- Speichern und zurück ins Hauptmenü
- Zurück ins Hauptmenü ohne Speichern.

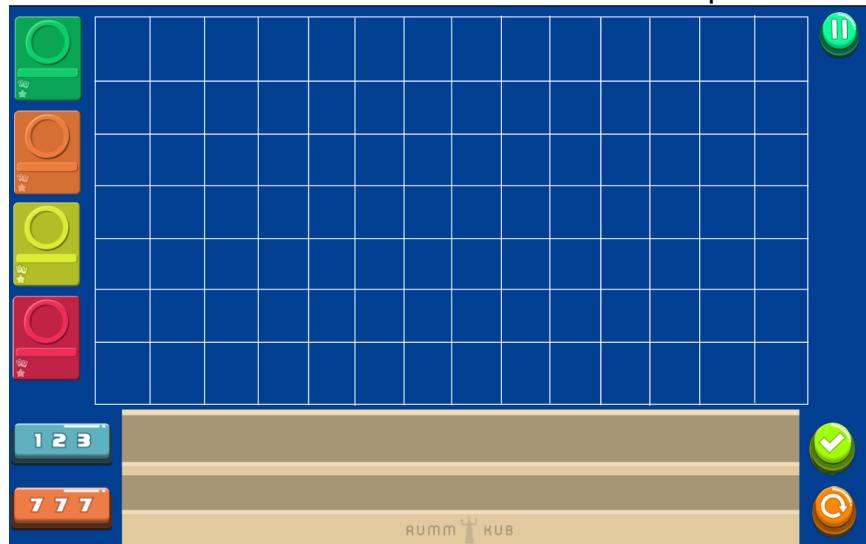
Wenn das Spiel beendet ist, wird eine Benutzeroberfläche mit der Rangliste der Spieler (vom ersten bis zum zweiten/dritten/vierten) angezeigt. Der Benutzer hat zwei Möglichkeiten, entweder erneut zu spielen oder zum Hauptmenü zurückzukehren.

Wenn der Benutzer am Zug ist, kann er die Spielsteine über das Spielbrett ziehen und sie innerhalb des Spielbretts platzieren.

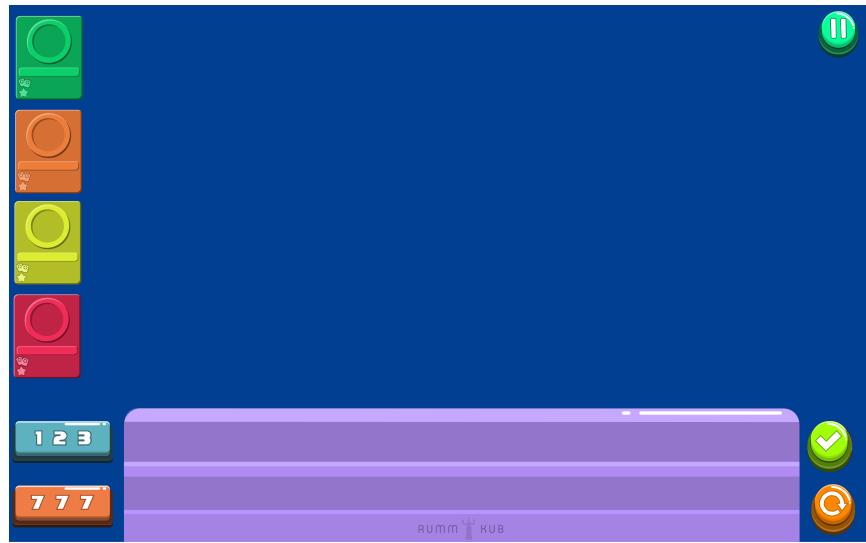




Die GUI bei Start und während eines Spiels

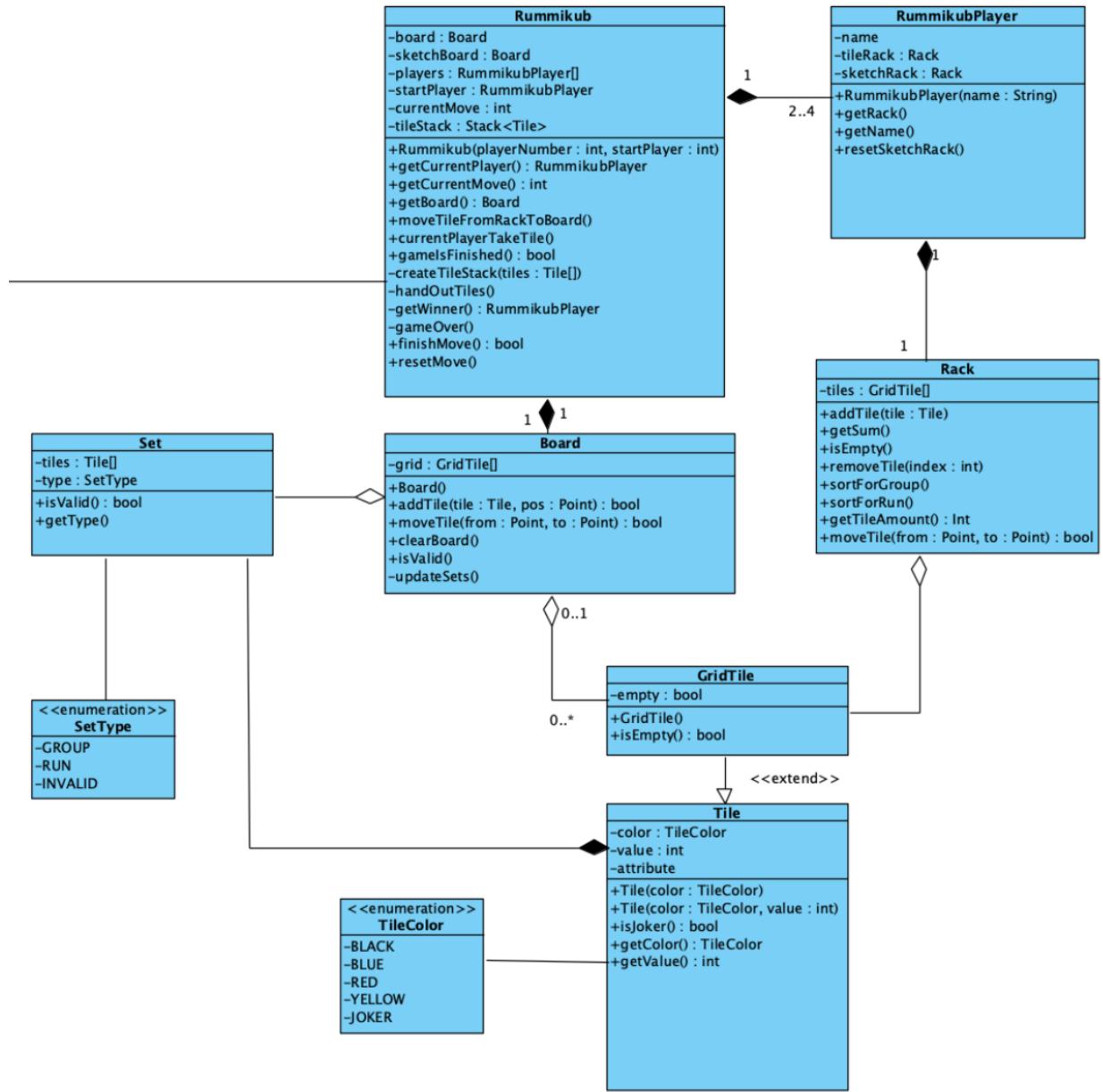


Entwurf einer GUI



Endgültiger Entwurf der GUI(einige Änderungen können sich im Gitter/Board befinden.)

5.2.6 Die Software-Architektur



Klassendiagramm der Rummikub-Game Engine

- TileColor (Enum)

Beinhaltet alle möglichen Farben eines Steins. Der Joker wird als Farbe behandelt.

- SetType (Enum)

Beinhaltet die möglichen SetTypen: Straße, Gruppe oder keine von bei-

den.

- Rummikub

Diese Klasse ist die zentrale Klasse. Von hier aus werden die Spieler und auch das Spielfeld gesteuert.

- Board

Die Brett-Klasse stellt das Brett dar. Spieler geben ihre Steine dieser Klasse, wenn sie sie ablegen wollen.

- RummikubPlayer

Jeder Spieler wird durch eine Instanz dieser Klasse abgebildet. Jeder Spieler hat seinen eigenen Rack mit seinen Steinen.

- Rack

Diese Klasse beinhaltet die jeweiligen Steine eines Spielers.

- GridTile

Jedes GridTile kann ein Tile enthalten. Die Klassen "Board" und "Rack" sind beide durch GridTile-Arrays definiert.

- Tile

Diese Klasse ist für die Spielsteine zuständig. Jeder Stein hat eine Farbe und einen Wert. Jedes Tile befindet sich entweder auf dem Spielfeld oder auf einem Rack in einem gridTile.

- Set

Diese Klasse ist eine Erweiterung der Board-Klasse. Sie checkt die Korrektheit des Spielfeld indem sie Sets von Spielsteinen erstellt und schaut, ob diese korrekt sind.

5.2.7 Bildquellen

- https://www.printableboardgames.net/preview/Rummy_Cube_Tiles
- <https://de.wikipedia.org/wiki/Rummikub>

5.3 Skat

Typ	Kartenspiel
Spieler	3 Spieler
Material	Kartendeck mit 32 Karten

5.3.1 Regeln und Spielablauf

Grundsätzliches

- *Spielparteien*

Es spielt in einem Spiel immer ein Spieler, der "Alleinspieler", gegen die zwei anderen Spieler, die "Gegenspieler". In einem Spiel kann der Alleinspieler als einziger Spielpunkte verdienen oder verlieren.

- *Spiel und Runde*

Ein Spiel besteht aus 10 Runden. In jeder Runde eines Spiels spielen nacheinander alle Spieler jeweils eine Karte. Der Spieler mit der stärksten Karte gewinnt die Runde und damit die ausgelegten Karten und die mit ihnen verbundenen Kartenpunkte. Später mehr dazu, wie die stärkste Karte bestimmt wird.

- *Kartenpunkte und Sieg*

Es gibt insgesamt 120 Kartenpunkte pro Spiel zu erspielen. Die Punkte sind wie folgt verteilt:

[Ass]:	11P
[10]:	10P
[König]:	4P
[Dame]:	3P
[Bube]:	2P
[9], [8], [7]:	0P

Der Alleinspieler braucht 61 Kartenpunkte um zu gewinnen, seine Gegenspieler brauchen zusammen nur 60 Punkte, gewinnen also bei Gleichstand.

- *Spielpunkte*

Gewinnt der Alleinspieler das Spiel, so bekommt er eine gewisse Anzahl an Spielpunkten. Verliert er, werden ihm doppelt so viele Punkte abgezogen, wie ihm bei einem Sieg angerechnet worden wären. Diese Spielpunkte sind nicht die Kartenpunkte die über Sieg und Niederlage entscheiden, sondern werden anders bestimmt.

Spielpunkte können über die Spiele hinweg auf einem Art Punktekonto angesammelt bzw. verloren werden. Somit kann auch über mehrere Spiele ein Gewinner bestimmt werden. Mehrere Spiele zusammen werden ein Set genannt. Spielpunkte können auch in den negativen Bereich gehen.

Je öfter man Alleinspieler ist, desto mehr Chancen hat man, Punkte anzuhäufen um das Set zu gewinnen. Der Alleinspieler wird zu Beginn der Runde durch das Reizen bestimmt (s. Reizen).

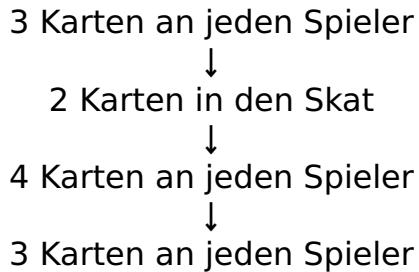
Kartenvergabe

- Hände und Skat

Jeder Spieler erhält 10 Karten, die nur für den jeweiligen Spieler sichtbar sind. 2 Karten kommen verdeckt in die Mitte, den sog. "Skat", womit alle 32 Karten ausgeteilt sind.

- Ausgabereihenfolge

Es wird in folgender Reihenfolge im Uhrzeigersinn ausgeteilt:



Reizen

Wer das Spiel als Alleinspieler bestreiten möchte, muss zu Beginn des Spiels die anderen Spieler beim Reizen überbieten.

Der Wert, bis zu dem ein Spieler bieten ("reizen") kann, hängt von seinem Blatt (Handkarten) ab und ergibt sich aus der Konstellation der eigenen Buben und dem Farbwert.

Auswahl der Werte der Buben:

♣ ♠ ♥ ♦ : 5	♣ ♠ ♥ ♦ : 2	♣ ♠ ♥ ♦ : 2
♣ ♠ ♥ ♦ : 3	♣ ♠ ♥ ♦ : 3	♣ ♠ ♥ ♦ : 2
♣ ♠ ♥ ♦ : 2	♣ ♠ ♥ ♦ : 4	♣ ♠ ♥ ♦ : 4
♣ ♠ ♥ ♦ : 2	♣ ♠ ♥ ♦ : 2	♣ ♠ ♥ ♦ : 3
♣ ♠ ♥ ♦ : 3	♣ ♠ ♥ ♦ : 2	♣ ♠ ♥ ♦ : 5

Hierbei zählt man von links aus die Anzahl der Buben, welche man in der eigenen Hand hat bis zum ersten Buben den man nicht hat bzw. die Anzahl der Buben die man nicht hat bis zum ersten Buben den man hat und addiert zu dieser Zahl eins.

Die Farbenwerte:

Kreuz: 12 Pik: 11 Herz: 10 Karo: 9 | Grand: 24

Der Farbenwert, den man sich entscheidet zu spielen, stellt den Trumpf dar. Wenn ein Spieler Kreuz als Trumpf haben möchte, so reizt er mit dem Farbwert Kreuz, wenn Herz als Trumpf besser scheint, reizt er mit dem Farbwert Herz.

"Grand" ist ein spezieller Farbwert. Bei ihm sind nur die Buben Trumpf und keine andere Karte irgendeiner Farbe. Da nur 4 Trümpfe im Spiel sind, ist ein Grand schwerer zu gewinnen als ein normales Farbspiel, daher auch der höhere Farbenwert.

Nullspiel:

Nullspiel ist eine andere Weise zu gewinnen. Wenn der Alleinspieler "Null" spielt, darf er keine einzige Runde gewinnen, sonst verliert er. Bei Nullspiel gibt es keinen Trumpf. Der maximale Reizwert ist 23. Die Bubenkonstellation hat keinen Einfluss auf diesen Wert.

Beispiel:

Eine Spielerin hat als einzigen Buben den Herz Buben auf der Hand. Es fehlen ihr von links gezählt 2 Buben, Kreuz Bube & Pik Bube. Der Faktor der sich durch die Buben ergibt ist also $2+1 = 3$. ("Ohne 2, Spiel 3"). Die Spielerin will am liebsten Karo (Farbenwert 9) als Trumpf spielen, Pik (Farbenwert 11) wäre für sie aber eine Alternative um höher reizen zu können. Sie reizt zunächst bis $3 \times 9 = 27$ um Karo spielen zu können. Allerdings reizt ein anderer Spieler höher, sodass sie nicht mehr auf Karo reizen kann. Nun reizt sie weiter bis $3 \times 11 = 33$ und gewinnt das Reizen und ist in dem folgenden Spiel die Alleinspielerin. Sie kann jetzt nur noch Pik (bzw. auch Kreuz) als Trumpf spielen, weil sie für Herz und Karo zu hoch gereizt hat. Grand wäre eine weitere Option.

Gereizt wird folgender Maßen:

Es gibt bei 3 Spielern die Positionen Vorhand, Mittelhand und Hinterhand. Diese wechseln jedes Spiel um eine Position.

Die Mittelhand beginnt die Vorhand zu reizen. D.h. sie sagt einen Wert an, den sie spielen möchte. Wenn die Mittelhand nicht passt, hat die Vorhand nun die Möglichkeit den genannten Wert der Mittelhand 'mitzugehen'. Wenn dies

geschieht, muss die Mittelhand erneut einen höheren Wert nennen, muss dabei natürlich auf Karten und den möglichen Spielwert achten. Vorhand und Mittelhand reizen sich gegenseitig bis einer der beiden Spieler passt. Meist geschieht das, wenn die Trümpfe nicht optimal sind und der Spielwert bei Weiterreizen somit überschritten werden könnte. Wenn die Vorhand passt, reizt die Mittelhand die Hinterhand, anderenfalls, reizt die Rückhand die Vorhand. Fortgesetzt wird mit dem nächsthöheren Wert von dem Wert der als letztes genannt wurde.

Spielverlauf

Nach dem Reizen steht der Alleinspieler fest. Dieser bestimmt nun den Trumpf. In der ersten Runde eines Spiels beginnt die Vorhand. Danach immer derjenige, der den letzten Stich (3 Karten) gewonnen hat. Da die Vorhand jedes neue Spiel wechselt, beginnt auch jedes Spiel ein anderer Spieler.

In jeder Runde bestimmt die Farbe der ersten Karte, die Farbe des Stichs. Sie muss wenn möglich von den anderen Spielern "bedient werden", d.h. die Spieler müssen Karten der gleichen Farbe spielen, es sei denn sie haben keine solchen Karten. Buben zählen zur Trumpffarbe dazu. Wird eine Farbe nicht bedient, so ist die gespielte Karte - unabhängig von Wertigkeit - immer schwächer, es sei denn es ist eine Trumpfkarte. Trumpfkarten sind immer stärker als Nicht-Trumpfkarten. Die stärkste Karte, die gelegt wurde entscheidet die Runde. Der Gewinner der Runde beginnt die nächste.

Die Karten der Runde kommen auf den Stapel des Gewinners und werden am Ende des Spiels, nach 10 Runden, in ihren Punkten summiert um den Gewinner des Spiels zu bestimmen.

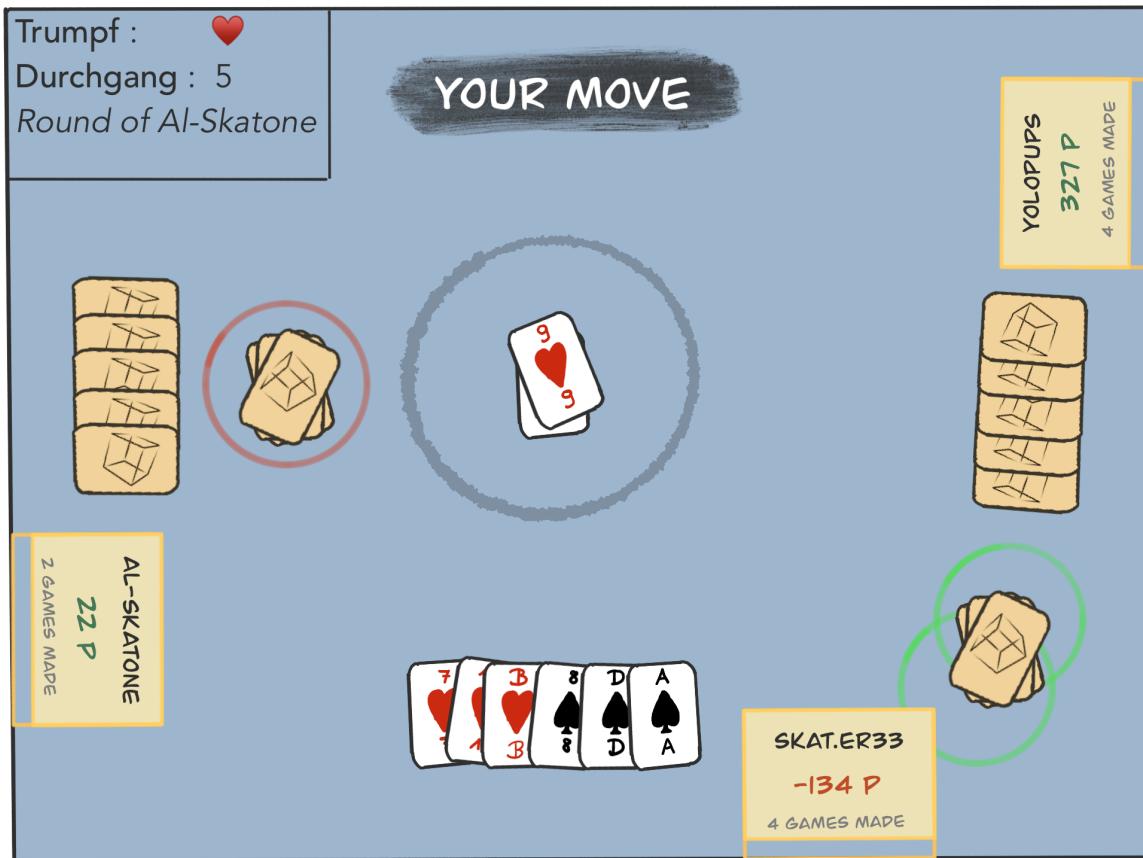
Auf das Punktekonto des Alleinspielers wird bei einem Sieg der Wert des Spiels addiert, sprich das, was der Alleinspieler maximal hätte reizen können mit seinen Buben und der Trumpffarbe.

Beispiel:

Karo ist Trumpf und die Pik Dame wird als erste Karte des Durchgangs gespielt. Der nächste Spieler spielt ein Kreuz Ass, was bedeuten muss, dass er kein Pik mehr auf der Hand hat (Pik Bube ausgeschlossen), da er sonst hätte bedienen müssen. Pik Dame ist hier stärker als Kreuz Ass. Nun wird als letzte Karte des Durchgangs Karo 8 gespielt. Diese Karte ist die stärkste, da Karo Trumpf ist, die anderen Karten äbsticht und damit den Stich gewinnt.

5.3.2 Design & Architektur

Graphical User Interface

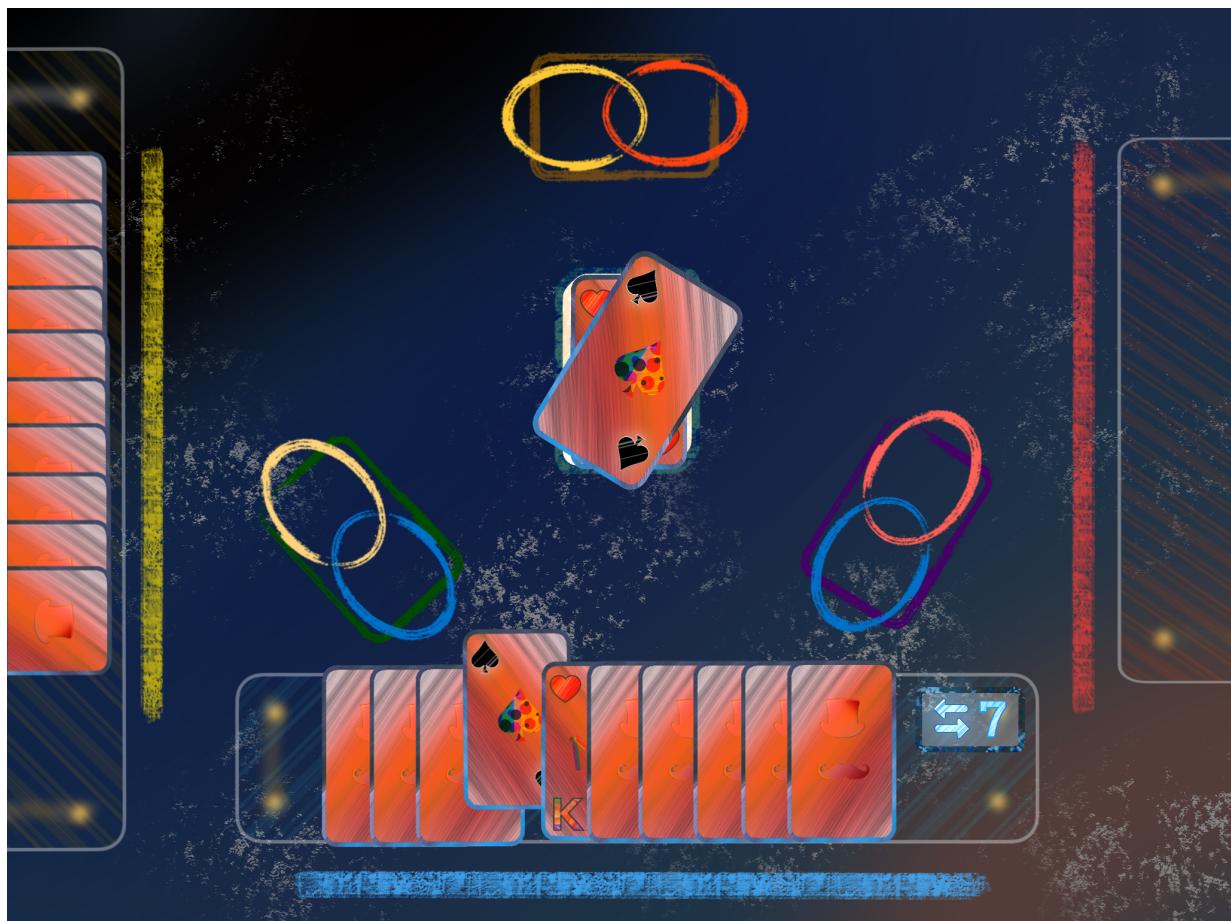


Erste, grobe Skizze der GUI

Der Spieler sieht ein Spielfeld vor sich auf dem sowohl seine als auch die Karten der Gegner (verdeckt) gezeigt werden. Am oberen Rand werden aktueller Durchgang, aktuelle Trumpffarbe und der Name des Alleinspielers angezeigt. Im restlichen Bild sieht der Spieler seine Karten offen vor sich. Die Gegner befinden sich links und rechts des Spielers. Für jeden Spieler werden Name und aktuelle Punktzahl angezeigt. Zum Spielen kann der Spieler Karten mit der Maus anklicken, woraufhin die Karte auf den Stapel in die Mitte gelegt wird. Während des Reizens stehen entsprechende Buttons zur Verfügung.

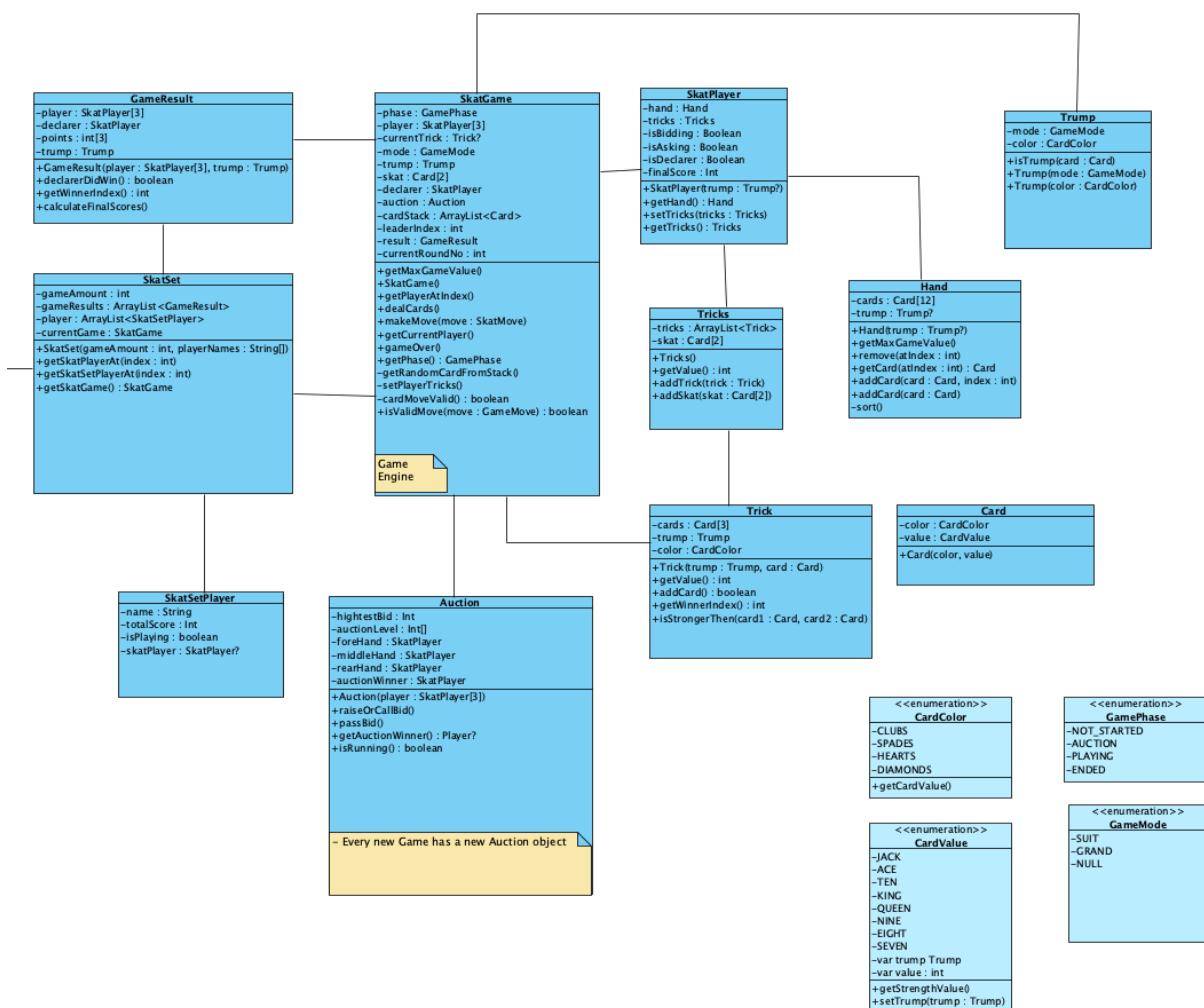
Der Spieler wird Karten durch klicken aber auch durch ziehen spielen können. Außerdem hat er die Möglichkeit Karten auf seiner Hand umzusortieren oder durch das drücken eines entsprechenden Buttons automatisch nach Farbe zu sortieren.

Am Ende einer jeden Partie und am Ende eines Sets wird eine Spielübersicht präsentiert die den Punkteverlauf über die mehreren Spiele sowie die Platzierung der Spieler anzeigt.



Entwurf der GUI, noch ohne Buttons und Spiel-Status-Anzeigen

Architektur



Klassen-Diagramm der Game Engine

• Enums

- CardValue: A, K, D, B, 10, 9, 8, 7
Enthält die möglichen Karten-Werte einer Karte und einige Funktionen zum berechnen der Kartenpunkte bzw. der Stärke einer Karte.
- Karten-Farbe: Kreuz, Pik, Herz, Karo
Enthält die möglichen Farben einer Karte und Funktionen zum Berechnen der Stärke.

- GameMode: Farbspiel, Grand, Null
Enthält die möglichen Modi eines Spiels.
- GamePhase: not started, auction, playing, ended
Enthält die verschiedenen Phasen eines Spiels.

- **Klassen**

- SkatSet
Diese Klasse sammelt alle SkatGames. Sie ist die oberste Instanz.
- SkatSetPlayer
Da Skat auch mit mehr als drei Spielern gespielt werden kann, die SkatGame-Klasse aber nur mit den drei tatsächlichen Spielern agiert, werden in der SkatSetPlayer-Klasse alle Mitspieler gebündelt und u.a. ihre Gesamtpunktzahl gespeichert.
- SkatGame
Diese Klasse ist das Zentrum des Spiels. Von ihr aus werden die Skat-Spieler, das Reizen zu Beginn des Spiels und die Runden gehandhabt.
- Auction
Diese Klasse kümmert sich um das Reizen am Anfang eines jeden Spiels.
- Trump
Der Trumpf wird nach dem Reizen festgelegt und spielt eine große Rolle im Spielverlauf, weswegen er eine eigene Klasse bekommt. Er beinhaltet den gespielten Spielmodus und die Spielfarbe.
- SkatPlayer
Die drei Skat-Spieler werden in jeweils einer Instanz dieser Klasse dargestellt. Jeder hat eine eigene Hand, Zugriff auf seine Stiche.
- Hand
Diese Klasse repräsentiert die Hände der Skat-Spieler.
- Trick
Diese Klasse stellt die Stiche dar. Sie wird angelegt für jede Runde, nachdem die erste Karte gespielt wurde. Nach der dritten Karte, wird der Stich auf den entsprechenden Gewinner-Stapel gelegt.

- Tricks

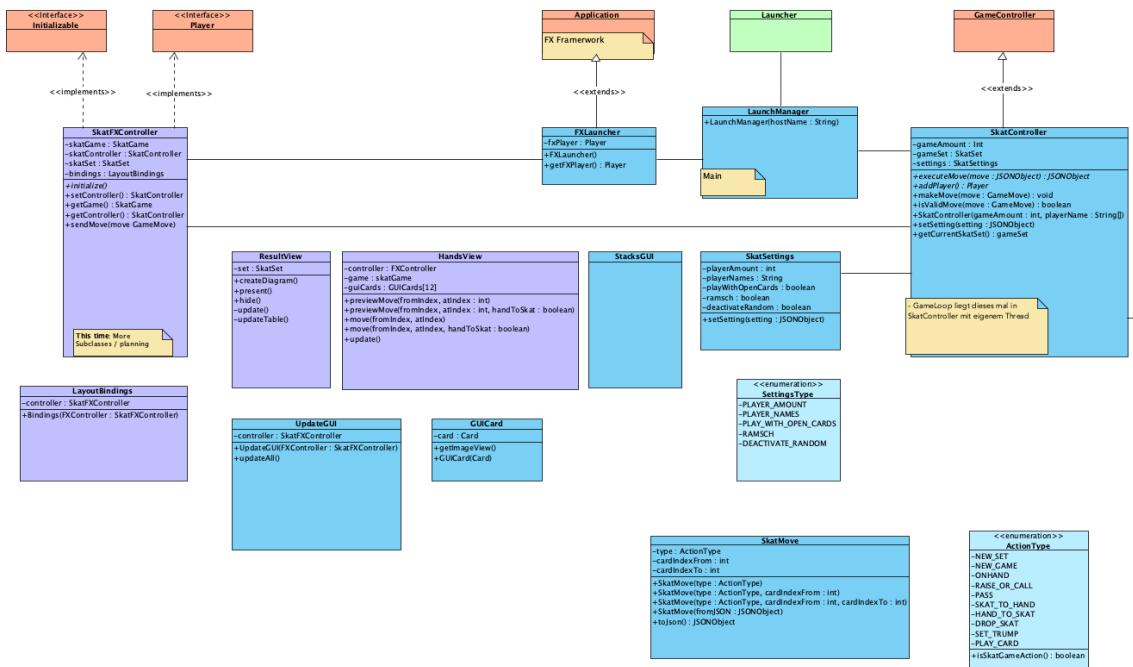
Hier werden für die zwei Parteien die Stiche der Klasse "Trick" gesammelt. Am Ende des Spiel werden die Punkte der Stiche hier berechnet.

- Card

Jede Karte ist ein Instanz dieser Klasse und hat eine Farbe und einen Wert. Jede Karte gibt es nur ein Mal.

- GameResult

Diese Klasse speichert das Endergebnis eines Spiels ab. Dies beinhaltet den Alleinspieler, den Trumpf und die Spielpunkte. Die Daten werden in der SkatSet-Klasse gespeichert.



Klassen-Diagramm JavaFX (GUI) und Controller

• Enums

- **SettingsType**: `PLAYER_AMOUNT`, `PLAYER_NAMES`, `PLAY_WITH_OPEN_CARDS`, `RAMSCH`, `DEACTIVATE_RANDOM`
Spezifiziert mögliche Einstellungen. Objekte des enums können zusammen mit einem JSON object der SkatController Klasse übergeben werden um Einstellungen zu ändern.

- ActionType: NEW_SET, NEW_GAME, ONHAND, RAISE_OR_CALL, PASS_SKAT_TO_HAND, HAND_TO_SKAT, DROP_SKAT, SET_TRUMP, PLAY_CARD
Spezifiziert mögliche Züge, die dem GameController unter Nutzung eines GameMove -Objektes übergeben werden.

- **Klassen Controller / Launcher / JavaFX**

- LaunchManager. "Main"-Klasse die extern aufgerufen wird und SkatController sowie FXLauncher initialisiert.
- FXLauncher : erbt von Application (JavaFX). Laucht JavaFX mit der gegebenen fxml-Datei.
- SkatFXController : Kontrolliert die GUI, implementiert das Player Interface. Ist dafür verantwortlich das Spiel darzustellen und GUI-Input ggbf. in Spielzüge zu übersetzen. Einige Aspekte werden in anderen Klassen ausgelagert.
- SkatController : Kümmert sich um den Spielablauf als Ganzes und kommuniziert zwischen Game-Klasse, GUI und ggbf. anderen Playern wie KI oder einer Netzwerk-Instanz
- SkatMove : Repräsentiert einen Spielzug der zwischen den Klassen übergeben werden kann.

5.3.3 Anforderungen & Anspruch

Obwohl das Spiel ein Kartenspiel ist, weist es eine ziemlich hohe Komplexität auf. Insbesondere das Reizen und die verschiedenen möglichen Spielmodi machen das Spiel spannend und durchaus herausfordernd in der Implementierung. Das Ziel ist, die zentralen Bestandteile des Spiels korrekt und anschaulich darzustellen.

Da wir außerdem eine Online-Mehrspielerfunktion und eine KI implementieren werden, sind wir der Überzeugung, dass das Spiel damit in jeder Hinsicht als "anspruchsvoll" qualifiziert ist.

5.4 Die Siedler von Konstanz

5.4.1 Einführung

Das Spiel "Die Siedler von Konstanz" nimmt sich das Spiel "Die Siedler von Catan" zum Vorbild. Hierbei handelt es sich um ein Brettspiel für 2 - 4 Spieler, indem es das Ziel ist Siedlungen und Städte zu bauen um Siegespunkte zu erhalten. Der erste Spieler mit 10 Siegespunkten hat gewonnen. Aktuell übernimmt diese Beschreibung die Bezeichnungen von „Die Siedler von Catan“. Während der Entwicklung werden wir das Szenario und die Bezeichnungen für die Spielemente dem Namen entsprechend anpassen. Die Spielweise bleibt die gleiche.

5.4.2 Spielemente

- **Map** Das Spielfeld, im Folgenden als Map bezeichnet, besteht aus 18 sechseckigen Tiles, die jeweils mit einer Glückszahl zwischen 2-12 versehen werden. Rohstofffelder. In der Mitte befindet sich ein Wüstenfeld, dieses ist neutral und dient lediglich als Startpunkt für den Räuber. Aus den 19 hexagonalen Feldern ergibt sich wiederum eine hexagonale Gesamtform.
- **Resource** Das Spiel beinhaltet die Rohstoffe Holz, Lehm, Erz, Wolle und Getreide. Sie werden für den Bau von Gebäuden und den Kauf von Entwicklungskarten verwendet.
- **Tiles** Die Map besteht aus Tiles, die man in 2 Kategorien unterscheiden kann:
Resource Tiles und *Neutral Tiles*
Resource Tiles sind mit einem Rohstoff und einer Würfelzahl (*Hit Number*) zwischen 2 und 12 versehen (Details zur Rohstoffverteilung werden im Abschnitt Spielablauf erklärt), während neutrale Felder (Wüste, Wasser) keine weitere Funktion haben.
- **(See-)Straßen** (See-)Straßen werden entlang der Kanten zwischen zwei Feldern errichtet. Es wird in Seestraßen und Landstraßen differenziert. Sie verbinden Siedlungen und Städte miteinander.
Im Folgenden sind Straßen nicht in den Begriff „Gebäude“ eingeschlossen.
- **Gebäude - Siedlungen und Städte** Siedlungen und Städte sind für die Erwirtschaftung von Rohstoffen notwendig und verleihen dem Besitzer Siegespunkte. Sie können auf den Eckpunkten zwischen drei Rohstofffeldern gebaut werden.

- **Siegespunkte** Spielende können Siegespunkte erhalten indem sie Städte (1 SP) oder Städte (2P) bauen.
- **Die Räuber** Die Räuber stehen zu Beginn des Spiels auf einem Wüstenfeld. Wenn eine 7 gewürfelt wird, darf der/die würfelnde Spieler die Räuber auf ein beliebiges anderes Feld bewegen (außer auf Wasser). Das besetzte Feld ist, sofern es sich um ein Rohstofffeld handelt, für die Zeit der Besetzung blockiert, d.h. wenn die Hitnumber gewürfelt wird, werden für dieses Feld keine Rohstoffe ausgegeben.

5.4.3 Spielvorbereitung

1. Die Rohstofffelder werden zufällig verteilt, ebenso die Glückszahlen. Lediglich die Gesamtform des Spielfelds und die Position der Wüste in der Mitte sind immer vorgegeben.
2. Die Zugreihenfolge wird für den Rest des Spiels ausgelost.
3. Die Spielenden dürfen 2 mal der Reihe nach jeweils eine Siedlung und eine Straße auf dem Spielfeld platzieren. Die zweite Siedlung muss hierbei nicht mit der ersten verbunden sein. Die allgemeinen Bauregeln werden später erklärt.
4. Alle Mitspielenden erhalten Startrohstoffe. Hierfür wählen sie eine ihrer Siedlungen und erhalten die Rohstoffe, die auf den angrenzenden Felder abgebildet sind.

5.4.4 Spielblauf

Alle Spielenden kommen nun entsprechend der Zugreihenfolge nacheinander dran:

- **Würfelwurf (erforderlich)**

Es werden 2 Würfel geworfen. Alle Spielenden (nicht nur der Würfelnde), die eine Siedlung oder eine Stadt an einem Rohstofffeld mit der gewürfelten Zahl errichtet haben, erhalten pro Siedlung einen und pro Stadt zwei Einheiten des abgebildeten Rohstoffs in Form von Rohstoffkarten.

- **Optionle Aktionen**

Folgende Aktion können anschließend mit beliebiger Reihenfolge und Häufigkeit ausgeführt werden:

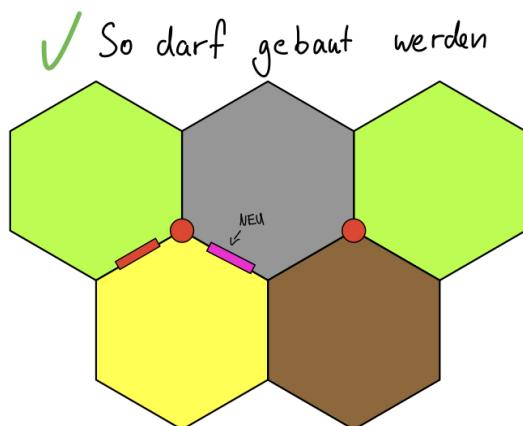
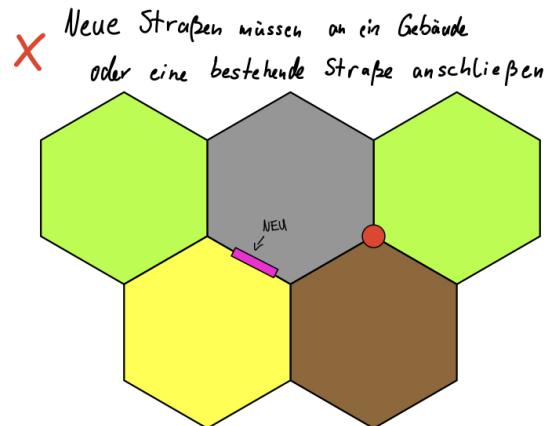
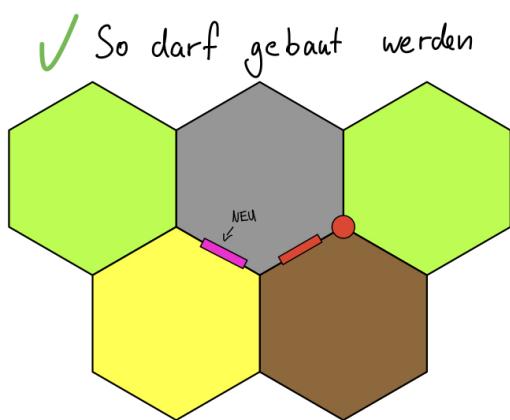
- Gebäude errichten
- Straßen errichten
- Rohstoffe tauschen mit der Bank (4 zu 1)

5.4.5 Bauregeln

Für das bauen von Gebäuden gelten folgende Kosten und Regeln:

Straßen

Neue Straßen müssen an vorhandene Straßen oder Gebäude anschließen.

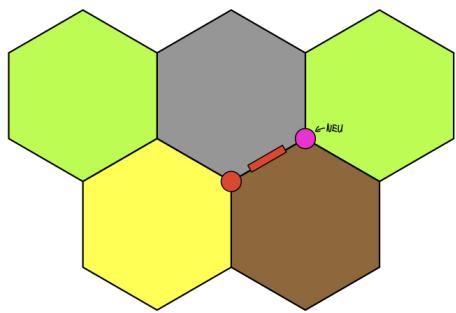


Zudem darf um eine Landstraße zu bauen maximal eines der angrenzenden Felder Wasser sein, für eine Seestraße muss mindestens eines der angrenzenden Felder Wasser sein.

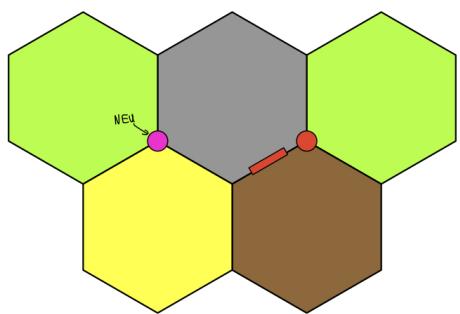
Gebäude

Gebäude müssen mindestens 2 Kreuzungen von anderen Gebäuden (egal welcher Farbe) Abstand haben und müssen an einer Straße gebaut werden.

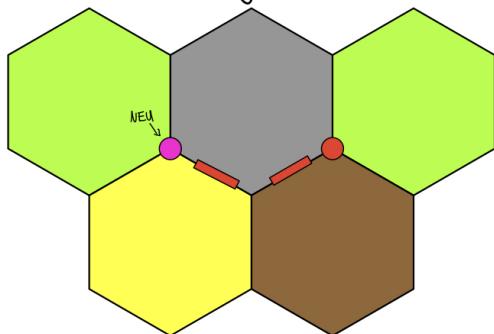
✗ Gebäude dürfen nicht nebeneinander liegen



✗ Neue Gebäude müssen an einer Straße liegen



✓ So darf gebaut werden



5.4.6 Spielende

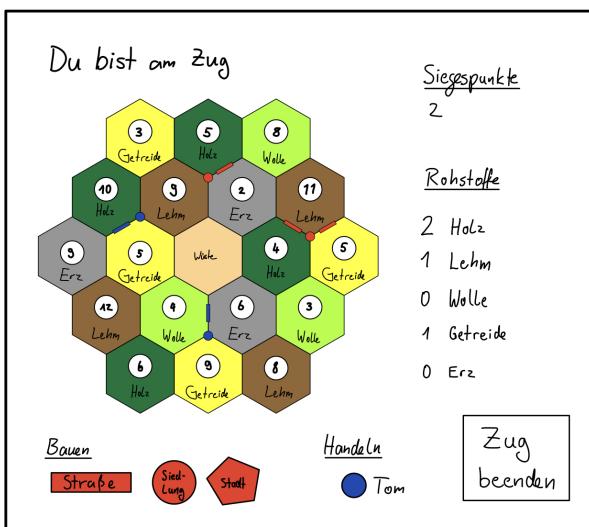
Das Spiel endet, sobald ein Spieler 10 Siegespunkte erreicht und somit das Spiel gewinnt.

5.4.7 Anforderungen

Die größte Herausforderung wird wahrscheinlich der Aufbau eines geeigneten Koordinatensystems für die hexagonale Struktur. Dieses System sollte es möglichst einfach erlauben Rohstofffelder, Straßen und Siedlungen/Städte zu kennzeichnen und zu erkennen, wie diese aneinander angrenzen. Zudem sind im originalen Spiel viele Zusatzfunktionen enthalten, die das Spiel sehr komplex machen. Falls das einfacher klappt als erwartet, existieren verschiedene Erweiterungen für das Spiel, die neue Elemente einfügen und bestehende verändern. Nach oben sind die Möglichkeiten hier dementsprechend groß. Um diese Herausforderungen zu meistern nehmen wir uns zunächst als Ziel, die Basishandlungen und Regeln des Spiels umzusetzen und lassen folgende Elemente zunächst außen vor: Handel mit anderen Spielern, Häfen, Entwicklungskarten.

5.4.8 Benutzeroberfläche

Die Benutzeroberfläche des Spiel könnte etwa folgendermaßen aussehen:



Im Zentrum steht das Spielfeld. Rechts sind die aktuellen Siegespunkte und Rohstoffe zu sehen, unterhalb des Spielfelds können Aktion ausgewählt werden.

5.4.9 Formale Anforderungen

Aufgrund des komplexen Spielfelds (Hexagonale Felder, Gebäude, Zahlen), ist eine Darstellung in der Konsole nicht sinnvoll.

5.4.10 Software Architektur

Dieses Spiel ist vielfältiger und umfangreicher als die anderen, lässt sich jedoch gut in folgende Komponenten unterteilen:

Map

Die Map-Komponente ist dafür verantwortlich, die Koordinaten aller Felder (Tiles), Gebäude (Buildings) und Straßen (Streets) abzuspeichern. Die Objekte können über ihre Koordinaten abgerufen und benachbarte Koordinaten bzw. Objekte können abgefragt werden.

Resource(-sets)

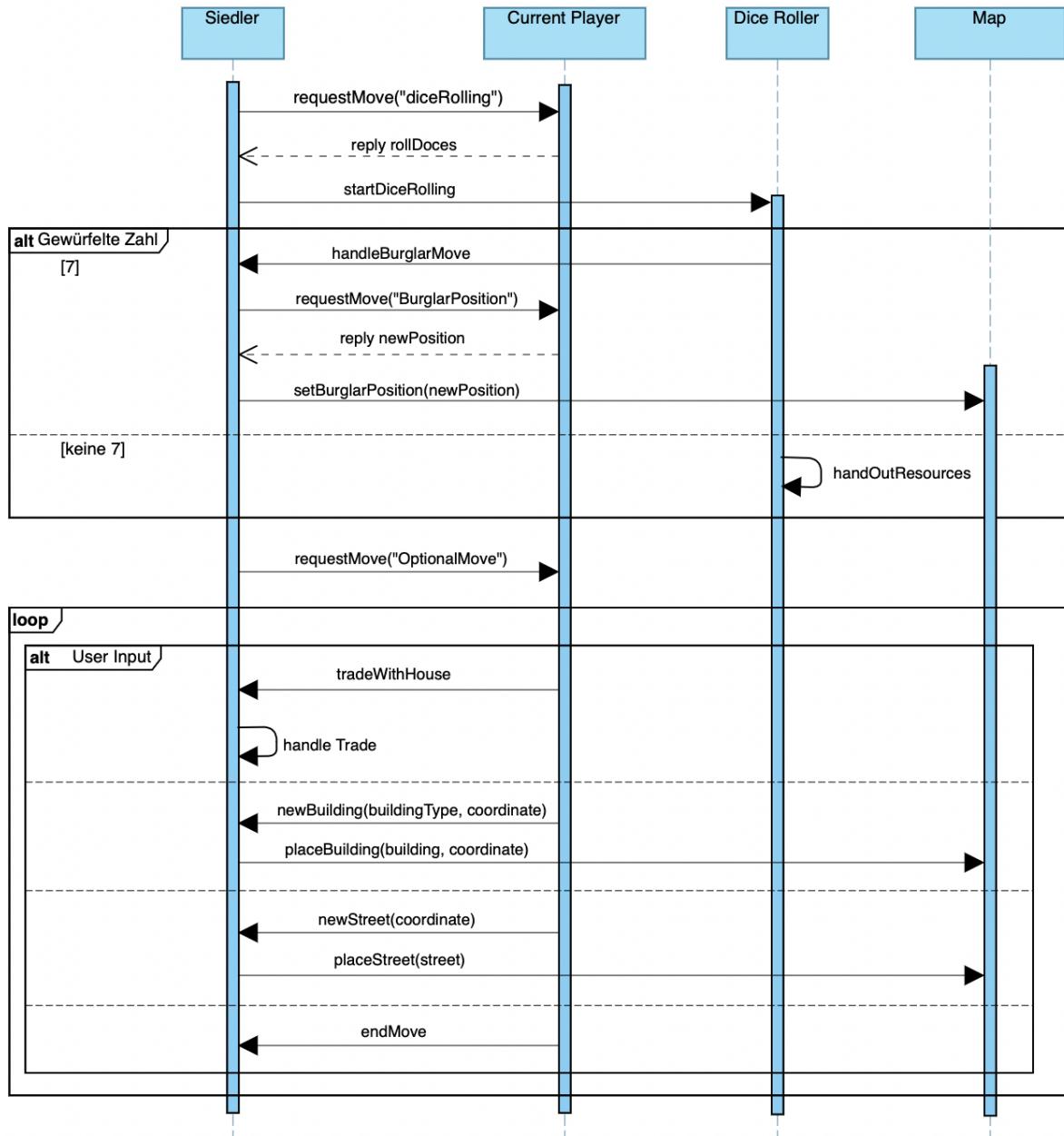
Die Resource-Komponente ist für die Bereitstellung der Rohstofftypen sowie die Verwaltung von Rohstoffmengen verantwortlich. Es stehen Methoden zur Verfügung, mit denen die Mengen auf verglichen werden können und manipuliert werden können. Dadurch kann beispielsweise überprüft werden, ob Spieler die nötigen Rohstoffe für verschiedene Bauoptionen besitzen und die Rohstoffe können ggf. vom Spieler abgezogen werden.

DiceRolling

Die DiceRolling-Komponente ist für die Zufallsgenerierung der Würfelzahlen verantwortlich sowie für ggf. für die Rohstoffverteilung. Dabei wird von den beiden bereits vorgestellten Komponenten Gebrauch gemacht.

Zugablauf

Jede Spieler führt nach einander einen Zug aus mit folgendem Schema:



5.5 TicTacToe

TicTacToe wird nur am Rande in das Framework integriert und steht bewusst nicht im Mittelpunkt wie die anderen Spiele, da die 5 Versionen nicht in Gruppen- sondern in Einzelarbeit entstanden sind.

Die verschiedenen Version werden so eingebunden, wie sie von den einzelnen Gruppenmitgliedern implementiert wurden.

6 Design, Architektur and technische Anforderungen

Da die vier Spiele an sich keine Gemeinsamkeiten haben, die sich eignen geteilt zu werden, werden keine Klassen in mehreren Spielen genutzt. In den jeweiligen Spielbeschreibungen wird näher auf Design und Architektur eingegangen. (→ Spiele)

7 Benutzeroberfläche

Das Menü der Spielesammlung wird der Ausgangspunkt sein, von dem aus alle Spiele, sowie Einstellungen und Rankings erreicht werden können. In den Einstellungen wird das Nötigste einzustellen sein, wie beispielsweise die Hintergrundfarbe. Die Rankings, welche durch das Menü zu erreichen sind, sind eine Zusammenstellung aus allen Spiele-Rankings, sodass spieleübergreifender Wettbewerb möglich ist.

Auf die Spiele kann über einen eigenen Button zugegriffen werden. Die einzelnen Spiele sind in sich äußerst unterschiedlich, weshalb die spieldaten spezifischen Optionen wie Name oder Spielmodus erst in den jeweiligen Spielen selbst einstellbar sind. Von jedem Spiel aus kann wieder ins Menü zurückgekehrt werden, sodass dieses idealerweise Beginn und Ende jeder Spielerfahrung ist.

8 Anforderungen an die Software

Die Anforderung an die Software ist alle 4 Spiele korrekt zu implementiert und in einem GUI darzustellen. Der Spieler soll von einem Menü Zugriff auf bestimmte Einstellungen haben und von dort aus auf die jeweiligen Spiel-Interfaces gelangen können.

Die Anforderungen an die jeweiligen Spiele sind in den Spiel-Kapiteln selbst erläutert. (→ [Spiele](#))