

Pflichtenheft

Projekttitel: **CardClub**
 Auftraggeber: **DI Günther Hölzl**
 Auftragnehmer: **Bernd Reither, Lisa-Marie Hörmann, Mattias Burkard**
 Schuljahr: **2021/22** Klasse: **2CI**

| VERSION | DATUM | AUTORIN/AUTOR | ÄNDERUNG |
|---------|-----------|--------------------|-------------------------------|
| v1.0 | 9.2.2022 | Bernd Reither | Erstellung des Pflichtenhefts |
| v1.1 | 21.3.2022 | Lisa-Marie Hörmann | Überarbeitung der Muss-Ziele |

Inhalt

| | | |
|-----|------------------------------------------|----|
| 1 | Vorwort..... | 2 |
| 2 | Ausgangssituation | 2 |
| 3 | Ziele | 2 |
| 3.1 | Muss-Ziele..... | 2 |
| 3.2 | Optionale Ziele..... | 3 |
| 4 | Know-How..... | 3 |
| 5 | Aufgabenstellung | 3 |
| 6 | Richtlinien | 4 |
| 7 | Allgemeines..... | 4 |
| 7.1 | Identifikation des Projekts | 4 |
| 7.2 | Leitgedanke | 4 |
| 7.3 | Veränderungsmanagement | 4 |
| 8 | Use Cases | 5 |
| 8.1 | Name und Identifier | 5 |
| 8.2 | Beschreibung des Use Case/Funktion | 5 |
| 8.3 | Anforderungen | 5 |
| 8.4 | Aktoren / Anwender..... | 5 |
| 8.5 | Voraussetzungen | 5 |
| 8.6 | Ablauf des Use Case | 6 |
| 8.7 | Logisches Ablaufdiagramm | 8 |
| 8.1 | Skizzen des User Interface | 12 |
| 9 | Server..... | 14 |
| 10 | Deliverables | 14 |
| 11 | Testen | 14 |

1 Vorwort

Bei „CardClub“ handelt es sich um einen digitalen Nachbau des bekannten Kartenspiels UNO. Ziel der Software ist es, alle Karten und Regeln wie im Originalspiel zu integrieren und somit maximalen Spielspaß zu garantieren. Im Singleplayer-Modus ist es möglich gegen einen Bot zu spielen und im optionalen Multiplayer-Modus soll man gegen andere Spieler spielen können.

2 Ausgangssituation

Im SEW-Unterricht werden bis zum 19.4.2022 Projekte umgesetzt. Unser Team hat sich für einen Nachbau des Kartenspiels UNO mithilfe der Programmiersprache Java und der GUI-Library JavaFX entschieden.

3 Ziele

3.1 Muss-Ziele

| Aufgabe | Fertigstellung in % |
|----------------------------------------------------------------------------------------------------------------------------------|-------------------------------|
| Klasse für eine Karte (Symbol, Farbe, ...) → Bernd Reither | 100% |
| Klasse zur Speicherung der Karten des Spielers → Bernd Reither | 100% |
| Begrenzte Dauer für einen Zug (festlegen durch User) → Mattias Burkard | 100% |
| Überprüfen des Regelwerks von UNO - Ist der Zug gültig? → Mattias Burkard (Checker.java) | 80% (+2 und +4 Karten ziehen) |
| Zusätzliche Regeln sollen wählbar sein → Lisa-Marie Hörmann | 30% |
| Einbauen eines Bots → Mattias Burkard | 80% (Schwierigkeitsstufen) |
| Hauptmenü: Auswahl zwischen Singleplayer, Optionen, About, Beenden → Lisa-Marie Hörmann | 100% |
| Singleplayer: Wählen von Zugdauer, Anzahl der Startkarten, Anzahl der Bots, Zusatzregeln mittels Checkboxes → Lisa-Marie Hörmann | 80% (Zusatzregeln) |
| Spieltisch: Eigene Karten anzeigen, Abhebestapel, Ablegestapel, Verbleibende Zeit → Lisa-Marie Hörmann | 100% |
| Optionen: Lautstärke, Reset → Bernd Reither | 70% (noch keine Musik etc.) |
| About: Infos über den aktuellen Versionsstand, die Entwickler, Websites, etc. → Bernd Reither | 100% |

3.2 Optionale Ziele

- Globales Leaderboard
 - Registrierung notwendig?
 - Zählen der „Wins“
- (Privater) Multiplayer
 - Eigene Serversoftware, die die Spieler miteinander „verbunden“ werden
 - Ein Spieler ist der „Host“, d.h. er verwaltet die Spieleinstellungen (ähnlich wie beim Singleplayer) und stellt einen Code für die Mitspieler bereit
 - Die anderen Spieler treten mithilfe des Codes bei
 - Server soll in den Optionen gewählt werden können

4 Know-How

- Objektorientierte Entwicklung
- Exceptions
- Arrays
- Java-FX
- FileIO/Database
- Eventhandler
- Style-Guide Überprüfung mit Checkstyle
- Effizienz
- Zusätzlich
 - Scene Builder
 - Serverkommunikation

5 Aufgabenstellung

Umgesetzt werden soll ein Nachbau des Spiels UNO, alle Karten und Regeln sind so wie im Originalspiel zu integrieren. Es soll einen Singleplayer-Modus geben, in dem man gegen sogenannte „Bots“ spielen kann und optional noch einen Multiplayer-Modus, um auch gegen Freund spielen zu können.

6 Richtlinien

- Bis 22.2.2022: Erstellen eines Repos
 - Das Repository wurde auf dem Schul-Git-Server, aber auch privat auf Github.com angelegt, da hier die Verwaltung des Projekts deutlich einfacher ist (z.B. Implementieren von Features in eigenen Branches, Tracken von Issues, ...).
 - Es wurden sich Gedanken über die genaue Umsetzung des Projektes gemacht.
 - Es wurde sich über notwendige Librarys informiert.
 - Abschließend wird das Pflichtenheft im PDF-Format auf beide Git-Remotes gepusht.
- Bis 22.3.2022: Abgabe eines lauffähigen Prototyps
 - Die meisten Basisfunktionen sollten bis zu diesem Zeitpunkt makellos umgesetzt sein.
- Bis 19.4.2022: Abnahme des Projekts
 - Zu mindestens alle Basisfunktionen sollten bis zu diesem Zeitpunkt implementiert sein, wenn genug Zeit ist, auch alle optionalen Funktionen.

Richtlinien für Beurteilung siehe „*Beurteilung 2CI SEW Projekt - Sommersemester 2022.pdf*“

7 Allgemeines

Das Ziel dieses Projekts ist es, das Kartenspiel UNO mithilfe der Programmiersprache Java und der GUI-Library JavaFX nachzubauen.

7.1 Identifikation des Projekts

Das Projekt trägt den Namen „CardClub“, das Kürzel ist somit „CC“.

7.2 Leitgedanke

Unsere Motivation war es, ein schnelles und einfaches Spiel zu entwickeln, dass sowohl in einer Gruppe als auch allein gespielt werden kann. Denn es gibt nicht immer noch jemanden, der gerade Zeit und Lust hat, gemeinsam zu spielen. Da die meisten Leute bereits die (Grund-)Regeln von UNO kennen, wird die Komplexität des Spiels stark verringert und ein schneller Einstieg ermöglicht. Durch die vielen Anpassungsmöglichkeiten der Regeln werden verschiedenste Spielweisen unterstützt, was für Abwechslung und ein spannendes Spielerlebnis sorgt.

7.3 Veränderungsmanagement

Das Pflichtenheft wird am 22.2.2022 von DI Günther Hölzl bewertet, danach ist eine überarbeitete Version (V2) hochzuladen.

8 Use Cases

8.1 Name und Identifier

- SP: Singleplayer
- MP: Multiplayer

8.2 Beschreibung des Use Case/Funktion

- SP: Man spielt im sogenannten Singleplayer-Mode gegen „Bots“
- MP: Man spielt im sogenannten Multiplayer-Mode gegen andere Spieler (optionales Ziel)

8.3 Anforderungen

| Kürzel | Anforderung |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| SP | Bot, der die Regeln des Spiels versteht und anwenden kann. Optional sollte dieser auch versuchen zu gewinnen, anstatt einfach nur mitzuspielen. |
| MP | Serversoftware, durch die Instanzen des Spiels gehostet werden können. Mittels Invite-Codes muss es möglich sein, einer solchen Instanz beizutreten. |

8.4 Aktoren / Anwender

- SP: max. 1 Anwender
- MP: max. 4 Anwender (über Server verbunden)

8.5 Voraussetzungen

- SP: Anwendung muss heruntergeladen und ausgeführt werden.
- MP: Anwendung muss heruntergeladen und ausgeführt werden, zudem muss sie mit einem Server verbunden sein (Registrierung?)

8.6 Ablauf des Use Case

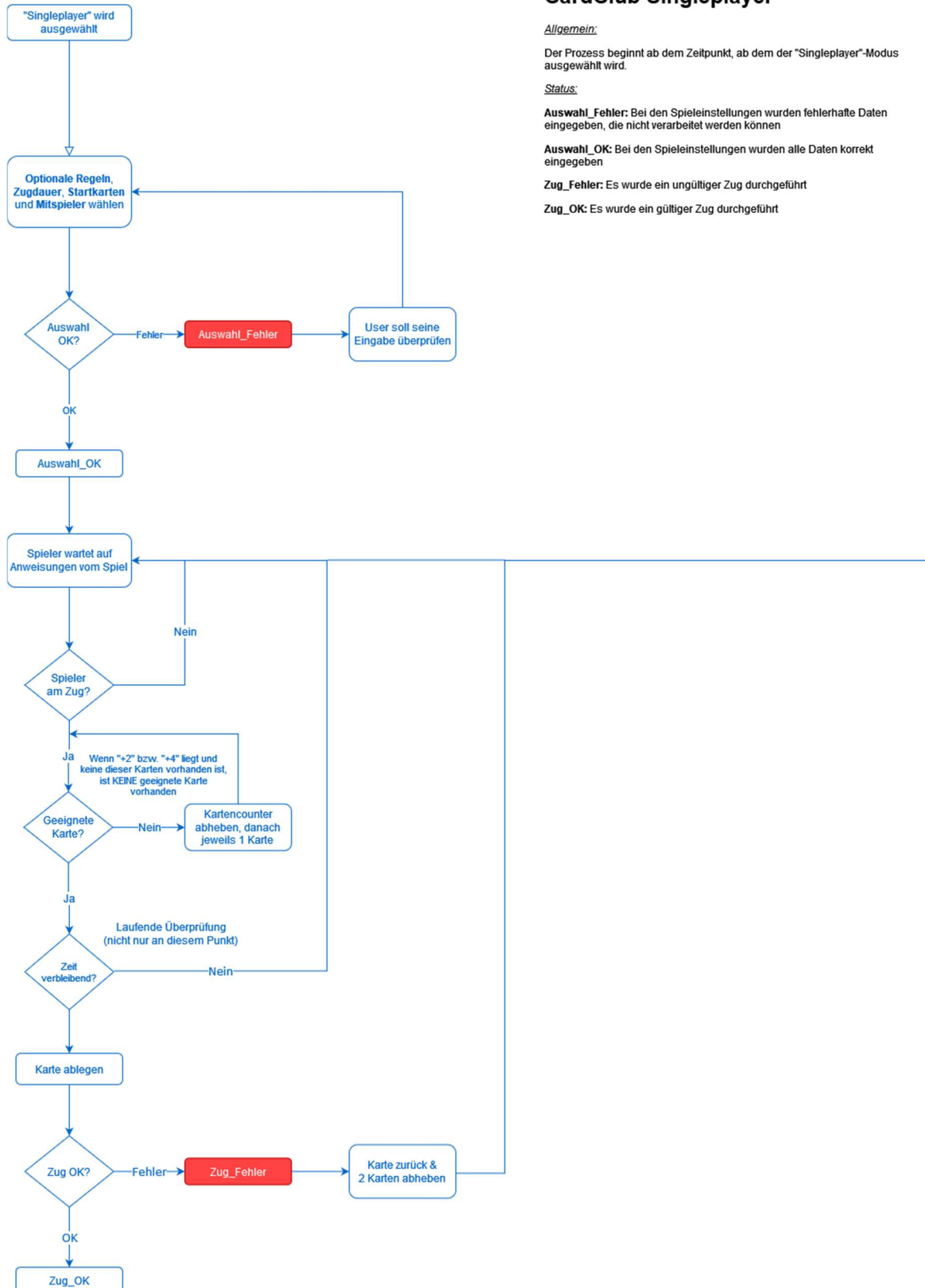
Für SP:

| Step | Name | Trigger | Input | Beschreibung | Output |
|------|----------------------|--------------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| 1 | Anwendung starten | Link/JAR | Mausklick | Start der Anwendung mittels eines Links bzw. einer JAR-Datei | Spiel öffnet sich |
| 2 | Klick „Singleplayer“ | Button | Mausklick | Auswählen des Singleplayer-Mode | Singleplayer-Spieleinstellungen öffnen sich |
| 3a | Optionale Regeln | Checkboxes | Mausklick | Auswählen von optionalen Regeln für das Spiel | Checkbox ist ausgewählt und das zusätzliche Regelwerk wird im Spiel angewendet |
| 3b | Zugdauer | Button/Feld | Mausklick Tastatur | Wählen der maximalen Zugdauer | Evtl. Fenster zur Eingabe |
| 3c | Startkarten | Button/Feld | Mausklick Tastatur | Wählen der Anzahl an Startkarten | Evtl. Fenster zur Eingabe |
| 3d | Mitspieler | Button/Feld | Mausklick Tastatur | Wählen der Anzahl an Mitspielern (Bots) | Evtl. Fenster zur Eingabe |
| 4 | Play | Button | Mausklick | Spiel starten | Das Spielfeld öffnet sich |
| 5a | Karten ablegen | Karte | Mausklick | Ablegen einer Karte am Stoß, indem man zuerst auf die Karte klickt und dann auf den Stoß | Eine Karte wird am Stoß abgelegt |
| 5b | Karten abheben | Karte(nstoß) | Mausklick | Abheben einer Karte vom Stoß, wenn man keine passende Karte hat bzw. keine ablegen möchte | Eine Karte wird zu den Spielerkarten hinzugefügt |
| 5c | „UNO!“ | Button | Mausklick | Wenn man nur mehr eine Karte hat, muss man innerhalb von 3s den „UNO!“ Button klicken, ansonsten müssen 2 Karten abgehoben werden | „UNO!“-Button wird ausgegraut |

Für MP:

| Step | Name | Trigger | Input | Beschreibung | Output |
|--------------|---------------------|--------------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| 1 | Anwendung starten | Link/JAR | Mausklick | Start der Anwendung mittels eines Links bzw. einer JAR-Datei | Spiel öffnet sich |
| 2 | Klick „Multiplayer“ | Button | Mausklick | Auswählen des Multiplayer-Mode | Auswahl zw. Hosten und Beitreten öffnet sich |
| 3a | Klick „Host Game“ | Button | Mausklick | Multiplayer-Spiel als Host starten | Multiplayer-Spieleinstellungen öffnen sich |
| 3b | Klick „Join Game“ | Button | Mausklick | Multiplayer-Spiel beitreten | Box zur Eingabe des Codes öffnet sich |
| 4.1a Host | Optionale Regeln | Checkboxen | Mausklick | Auswählen von optionalen Regeln für das Spiel | Checkbox ist ausgewählt und das zusätzliche Regelwerk wird im Spiel angewendet |
| 4.1b Host | Zugdauer | Button/Feld | Mausklick Tastatur | Wählen der maximalen Zugdauer | Evtl. Fenster zur Eingabe |
| 4.1c Host | Startkarten | Button/Feld | Mausklick Tastatur | Wählen der Anzahl an Startkarten | Evtl. Fenster zur Eingabe |
| 4.1d Host | Mitspieler | Button/Feld | Mausklick Tastatur | Wählen der Anzahl an Mitspielern (Bots) | Evtl. Fenster zur Eingabe |
| 4.2 Join | Bereit | Button | Mausklick | Mitspieler ist bereit, wenn alle den Button klicken, ist der „Play“-Button für den Host freigegeben | Bereit-Button wird ausgegraut und evtl. ein grüner Haken neben dem Namen |
| 4 | Play | Button | Mausklick | Spiel starten | Das Spielfeld öffnet sich |
| 5a | Karten ablegen | Karte | Mausklick | Ablegen einer Karte am Stoß, indem man zuerst auf die Karte klickt und dann auf den Stoß | Eine Karte wird am Stoß abgelegt |
| 5b | Karten abheben | Karte(nstoß) | Mausklick | Abheben einer Karte vom Stoß, wenn man keine passende Karte hat bzw. keine ablegen möchte | Eine Karte wird zu den Spielerkarten hinzugefügt |
| 5c | „UNO!“ | Button | Mausklick | Wenn man nur mehr eine Karte hat, muss man innerhalb von 3s den „UNO!“ Button klicken, ansonsten müssen 2 Karten abgehoben werden | „UNO!“-Button wird ausgegraut |

8.7 Logisches Ablaufdiagramm



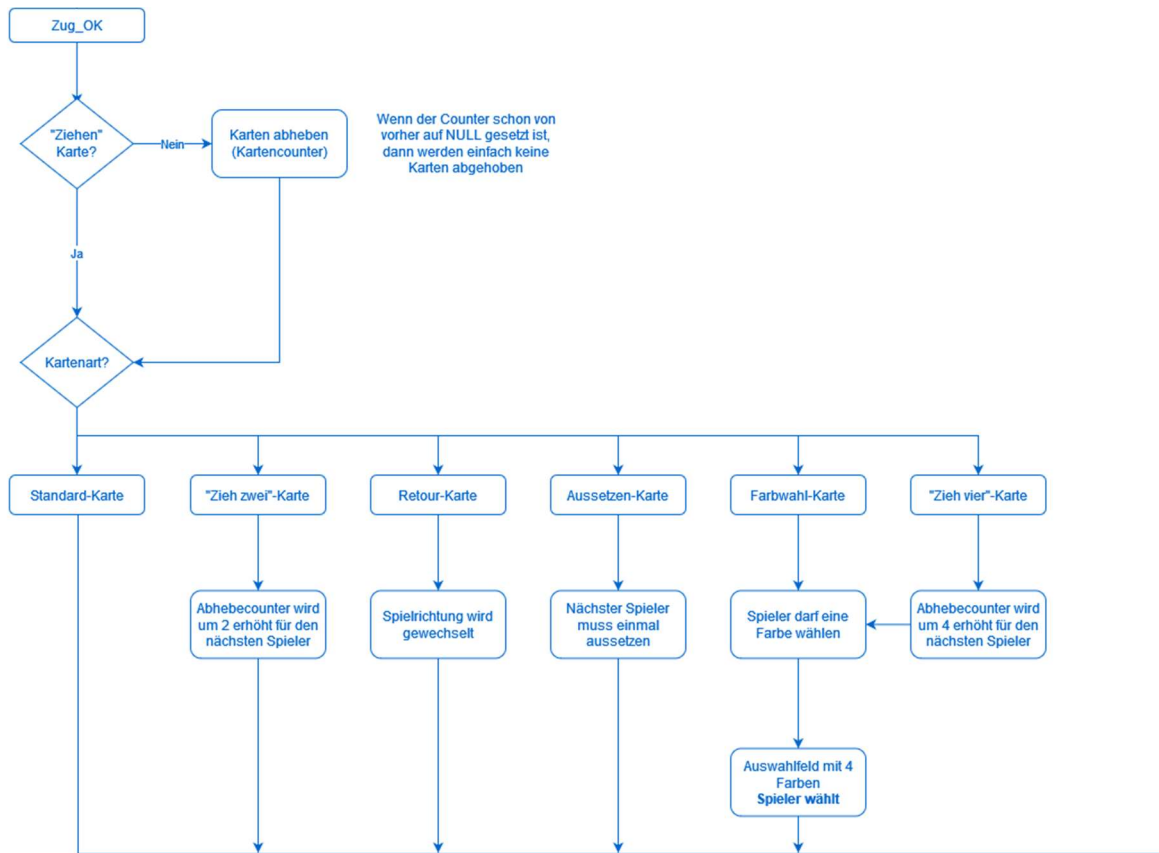


Abbildung 1 – CardClub Singleplayer

CardClub Multiplayer

Allgemein:

Der Prozess beginnt ab dem Zeitpunkt, ab dem der "Multiplayer"-Modus ausgewählt wird.

Status:

Code_Fehler: Ein ungültiger Code (nicht vorhanden) wurde eingegeben

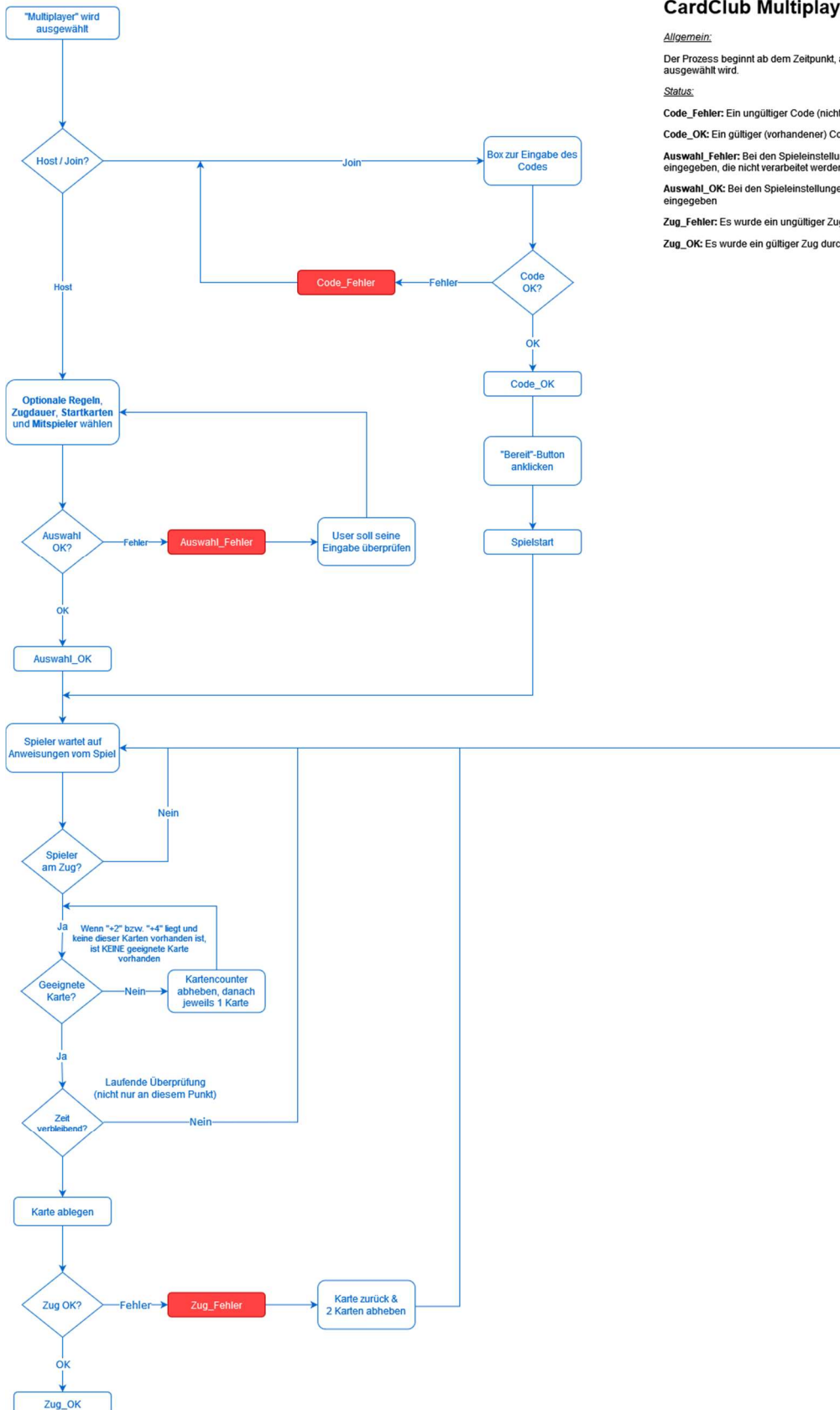
Code_OK: Ein gültiger (vorhandener) Code wurde eingegeben

Auswahl_Fehler: Bei den Spieleinstellungen wurden fehlerhafte Daten eingegeben, die nicht verarbeitet werden können

Auswahl_OK: Bei den Spieleinstellungen wurden alle Daten korrekt eingegeben

Zug_Fehler: Es wurde ein ungültiger Zug durchgeführt

Zug_OK: Es wurde ein gültiger Zug durchgeführt



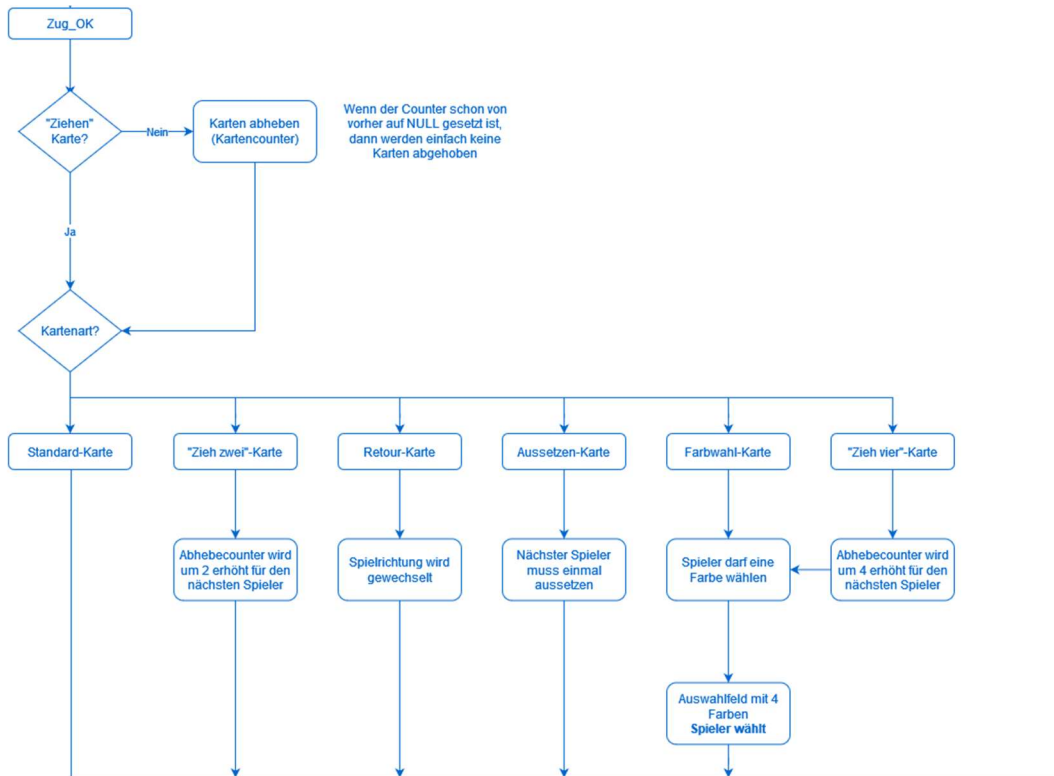
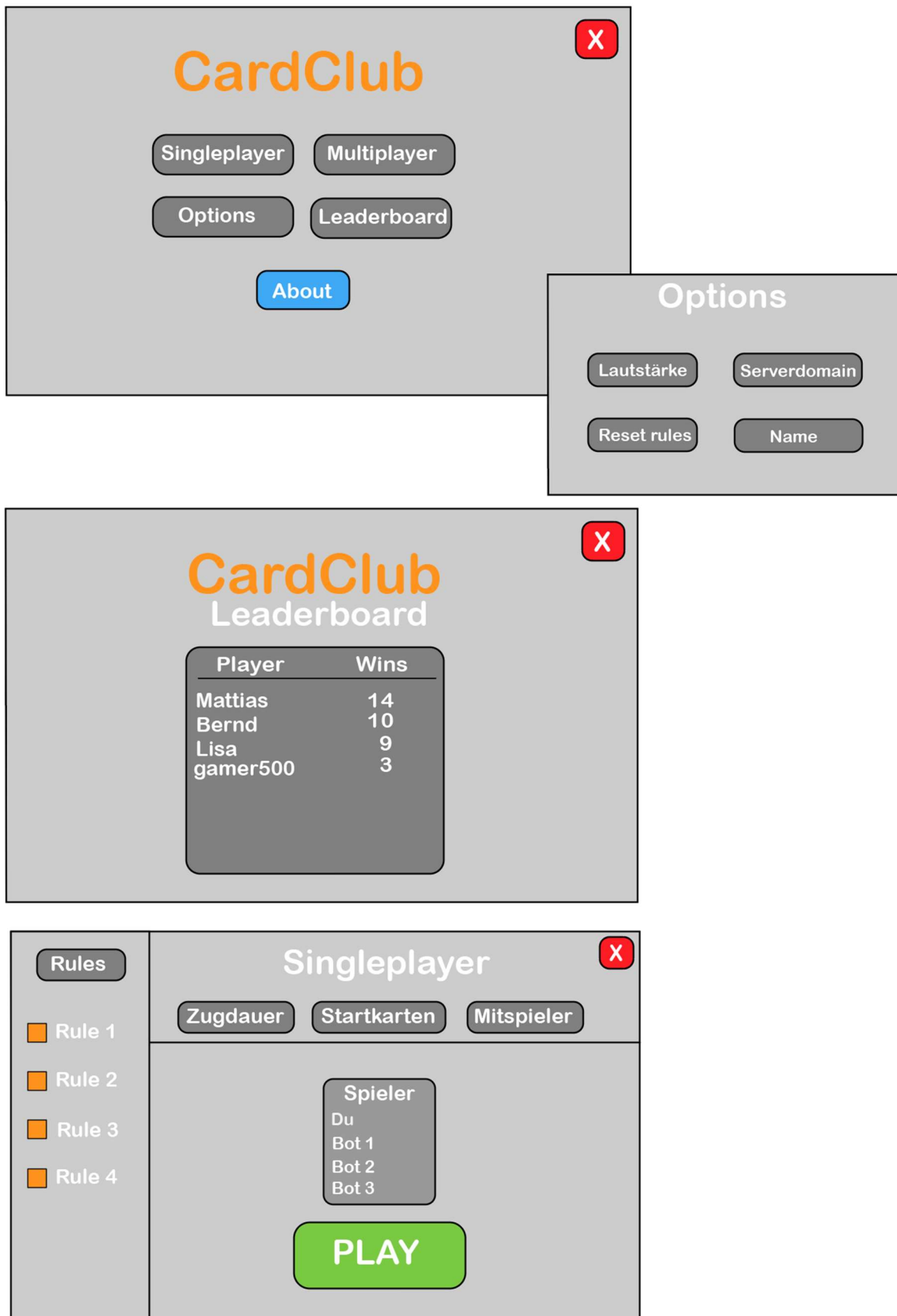
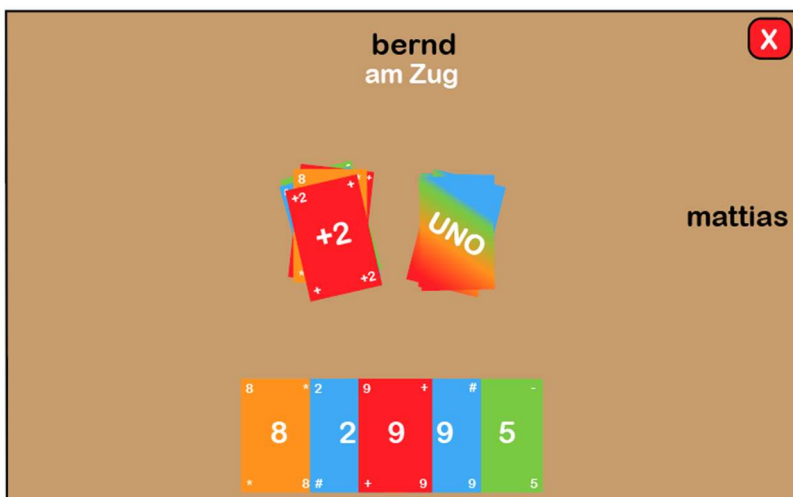
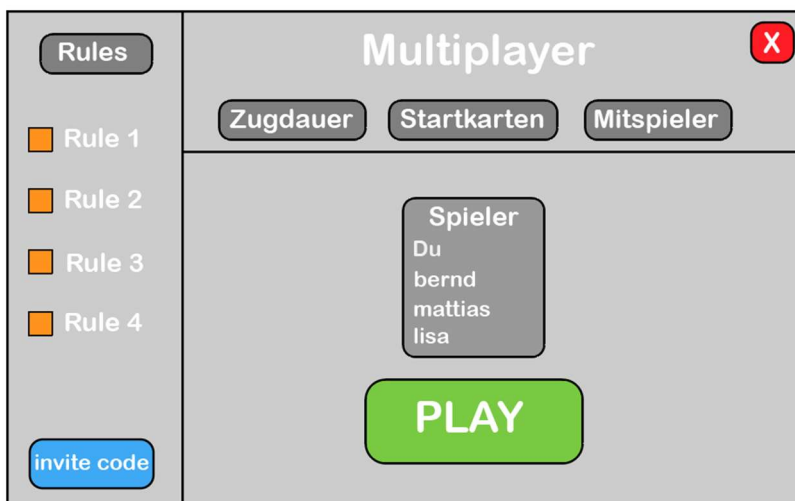
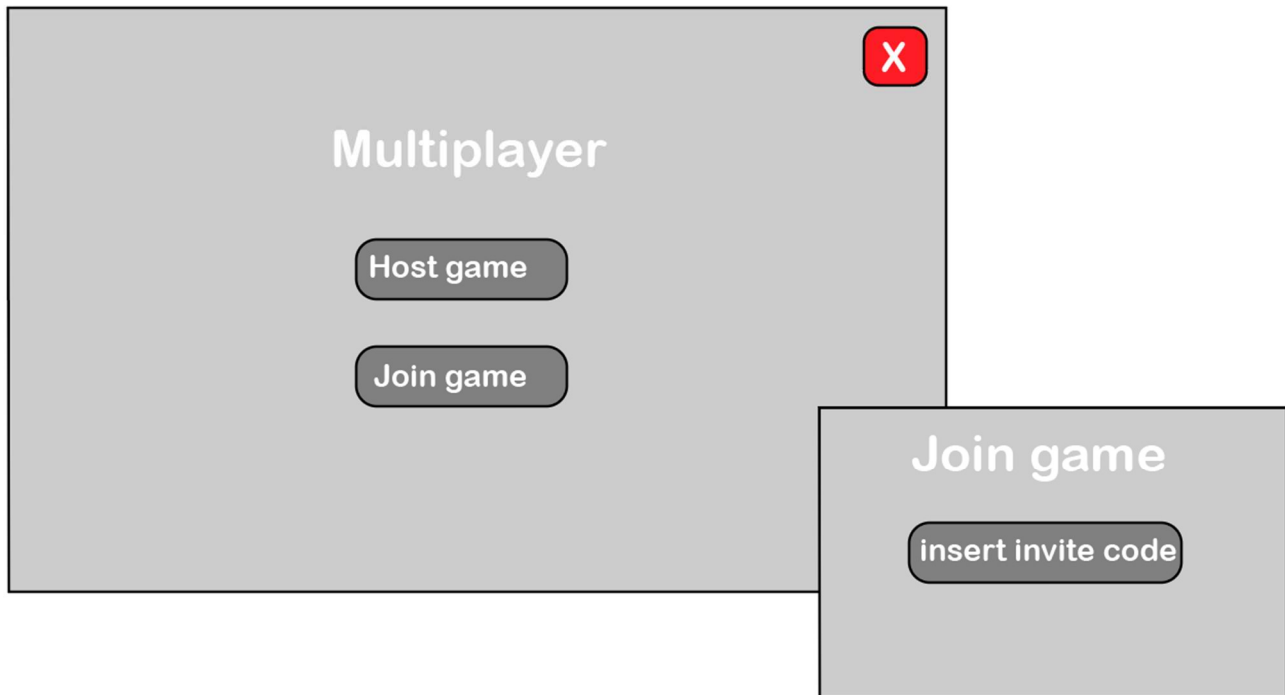


Abbildung 2 – CardClub Multiplayer

8.1 Skizzen des User Interface





9 Server

Für etwaige Serverdienste wird der Server von Bernd Reither, erreichbar unter der Domain `reither2410.duckdns.org`, verwendet.

10 Deliverables

- Client: JAR-Datei bzw. EXE-Datei
- Server: JAR-Datei

11 Testen

Um das implementierte Regelwerk zu überprüfen, wird dieses mit dem Original verglichen. Weiters wird versucht, falsche Züge durchzuführen, um zu sehen, ob diese richtig als invalide erkannt werden. Für die Prüfung der Karten werden diese in einem Testspiel ausprobiert. Alle möglichen Einstellungen und Regeln werden im Singleplayer ausprobiert. Um die Botfunktion zu testen, werden verschieden viele Bots in ein Spiel hinzugefügt.