

Card Game Durak - Report

Einleitung

Unser Anspruch in den Projektzeitraum war es nicht nur das Projekt selbst zu schaffen, sondern auch neues Wissen anzueignen. Keiner von uns hat zuvor mit einem Client-, Server-Modell gearbeitet, folglich haben wir auch dieses Projekt gewählt.

Die nächste Entscheidung war es, ein Kartenspiel auszusuchen. Als Gruppe haben wir uns auf eine Liste beschränkt, bestehend aus: Mau-Mau, Schwimmen und Durak. Aus dem gleichen Anspruch, wie bei der Auswahl des Projektes selbst, haben wir Mau-Mau und Schwimmen aus der Liste gestrichen und uns für Durak entschieden.

In diesem Report werden Denkprozesse, Lösungsansätze, Anleitungen für User und Developer als auch das ein modifiziertes Durak-Regelwerk, weiter erläutert.

Verlauf des Projekts

Woche 1:

Anfänglich haben wir uns mit dem Regelwerk befasst, um uns einen Rahmen zu setzen. Es kam die Diskussion hervor, ob wir die traditionellen bzw. eigene Regeln für das Spiel erstellen. Das Ergebnis war, dass die traditionellen Spielregeln für unsere Spielvorstellung nicht gut genug waren. Statt 2 bis 6 Spielern wollten wir bis 8 Spieler erlauben, folglich mussten wir die Anzahl der Spielkarten pro Farbe von 9 auf 13 auffüllen. So haben wir zu den traditionellen Karten: 6 bis 10, Bube, Dame, König und Ass, noch die Karten 2 bis 5 hinzugefügt, sodass man auf ein Blatt von insgesamt 52 Karten kommt.

Des Weiteren haben wir verändert, dass nicht jeder in jedem Zug angreifen kann, sondern nur die Nachbarn neben den jeweiligen Verteidiger. Die Überlegung dahinter war es, dass bei 8 Spielern jede Runde viel zu lange gehen würde sowie das trotzdem nicht jeder Angreifer angreifen könnte, weil der Verteidiger maximal nur 6-mal Verteidigen soll.

Als nächstes haben wir uns dem Layout des Spielbretts gewidmet. Hier gab es die Schwierigkeit, wie wir 8 Spieler darstellen wollen. Am Ende haben wir uns dazu entschieden, dass wir eine Liste an Spielern, ihre Anzahl an Spielkarten sowie ihren Spielstatus (Angreifer, Verteidiger, Nichts) zeigen.

Nach der Planung haben wir uns an die Arbeit gesetzt und uns in zwei Gruppen aufgeteilt. Die eine hat an dem UI gearbeitet, während die andere sich mit den Texturen befasst hat. Hier haben wir ebenfalls diskutiert, ob wir lizenzfreie Texturen verwenden wollen oder nicht. Wir haben uns für eigene Texturen entschieden.

Woche 2:

Wie in der vorherigen Woche haben wir weiter an dem UI und den Texturen gearbeitet sowie beide Elemente in einem funktionierenden Prototyp zusammengeführt. Dabei entstanden mehrere Probleme und eine Diskussion.

Beginnend mit der Diskussion: Wir haben uns überlegt, wie wir die Skalierung der Texturen, bei Änderung der Fenstergröße, vornehmen. Sollte es fest in einem Menü verstellbar sein, oder per Maus frei veränderbar? Mit der Begründung, dass es angenehmer sei für den Nutzer, das Fenster per Maus zu skalieren, haben wir uns für diese Variante entschieden.

Dies bringt uns zum zweiten Punkt die Probleme: Die meisten entstanden durch die manuelle Skalierung des Fensters, wie z.B., dass die Texturen unproportional skalieren oder dass die Scene vor der Stage aktualisiert wurde, sodass die Stage manchmal größer bzw. kleiner als die Scene war. Nichtsdestotrotz wurden diese Probleme behoben, die unproportionale Skalierungen wurden mit eigenen Funktionen gefixt, statt mit den Standardfunktionen von JavaFX und die Stage-, Scene-Probleme wurden mit Verzögerungen unterdrückt.

Woche 3:

In dieser Woche haben wir uns mit der Server-, Client-Verbindung auseinandergesetzt. Wir wollten, dass der Client ausschließlich für das Visuelle verantwortlich ist, während der Server die Logik als auch die Verbindung zwischen den ganzen Clients verwaltet.

Die Kommunikation selbst haben wir mit Strings und ihren dementsprechenden Parsern implementiert. Ursprünglich kam der Gedanke auf die Kommunikation per JSON-Dateien zu implementieren, dies wurde aber schnell verworfen, da wir keine Kompatibilitätsprobleme haben wollten.

Letztlich wurde noch die UI, durch die Nutzung von CSS, aufgehübscht.

Woche 4:

Auch hier wurde weiter an der Server-, Client-Kommunikation gearbeitet, zusätzlich zur Spiellogik. In dieser Logik haben wir mehrerer Ansätze gebraucht. Zuerst wollten wir alles als Arrays darstellen. Dies hat sich jedoch als umständlich und kompliziert herausgestellt, folglich haben wir diese auf Arraylisten umgestellt. Danach haben wir überlegt, wie wir die Zustände der Spieler darstellen wollen. Unser erster Ansatz war es die einzelnen Spieler mit Flags zu füllen. Aus den Gründen der Übersichtlichkeit, wurden diese mit Arrays ersetzt.

In dieser Woche haben wir auch weiter mit der automatischen Skalierung zu kämpfen gehabt. Das unterdrücken der Fehler hat nicht immer funktioniert und war auch eher eine Notlösung. Um diese Probleme nun endgültig zu beheben haben wir unsere Methoden noch einmal von grund auf neu gestaltet.

Woche 5/6:

In den letzten zwei Wochen haben wir die Zeit genutzt, um das Spiel zu polieren und den Report anzufertigen. Beim Testen sind uns eine Menge Bugs aufgefallen, so hat sich das Spiel häufiger gesoftlocked, wenn ein Spieler keine Karten mehr auf der Hand hatte oder wenn ein anderer Spieler für den jeweilig anderen etwas genommen oder gepasst hat.

Auch wurde noch ein letztes Feature hinzugefügt. Sobald ein Spieler ein laufendes Spiel verlässt, wird dieser mit einer AI ausgetauscht. Und es wurde eine Kommandozeile für den Server hinzugefügt damit das Spiel nach Beendigung neu gestartet werden kann.

Individuelle Beiträge:

Bernard, Yann Pablo:

- Verantwortlich für Texturen.
- Mitarbeiter an der Spiellogik.

Kiel, Sebastian:

- Mitbegründung der Texturen
- Layout und Design des Screens
- Server-Client-Verbindung
- Konzipierung der Spiellogik
- Qualityassurance

Schewe, Patrick:

- Grundbausteine der UI
- Hauptentwickler der Spiellogik
- Hauptentwickler der Serverkonsole
- Entwickler der KI
- Qualityassurance

Witteck, Robert:

- JavaDoc
- CSS
- BasisStruktur der UI
- Ergänzungen an der Client-Server-Verbindung

User-Anleitung

Bevor Sie irgendetwas starten wollen: Installieren Sie sich bitte java Version 11 oder höher, zu finden unter <https://jdk.java.net/java-se-ri/11>

Unter Windows:

Nach der Installation fügen Sie java zu Ihren Path-Variablen hinzu, wie hier beschrieben:

https://www.java.com/de/download/help/path_de.html#:~:text=W%C3%A4hlen%20Sie%20Start%20und%20dann,PATH%20und%20w%C3%A4hlen%20Sie%20sie.

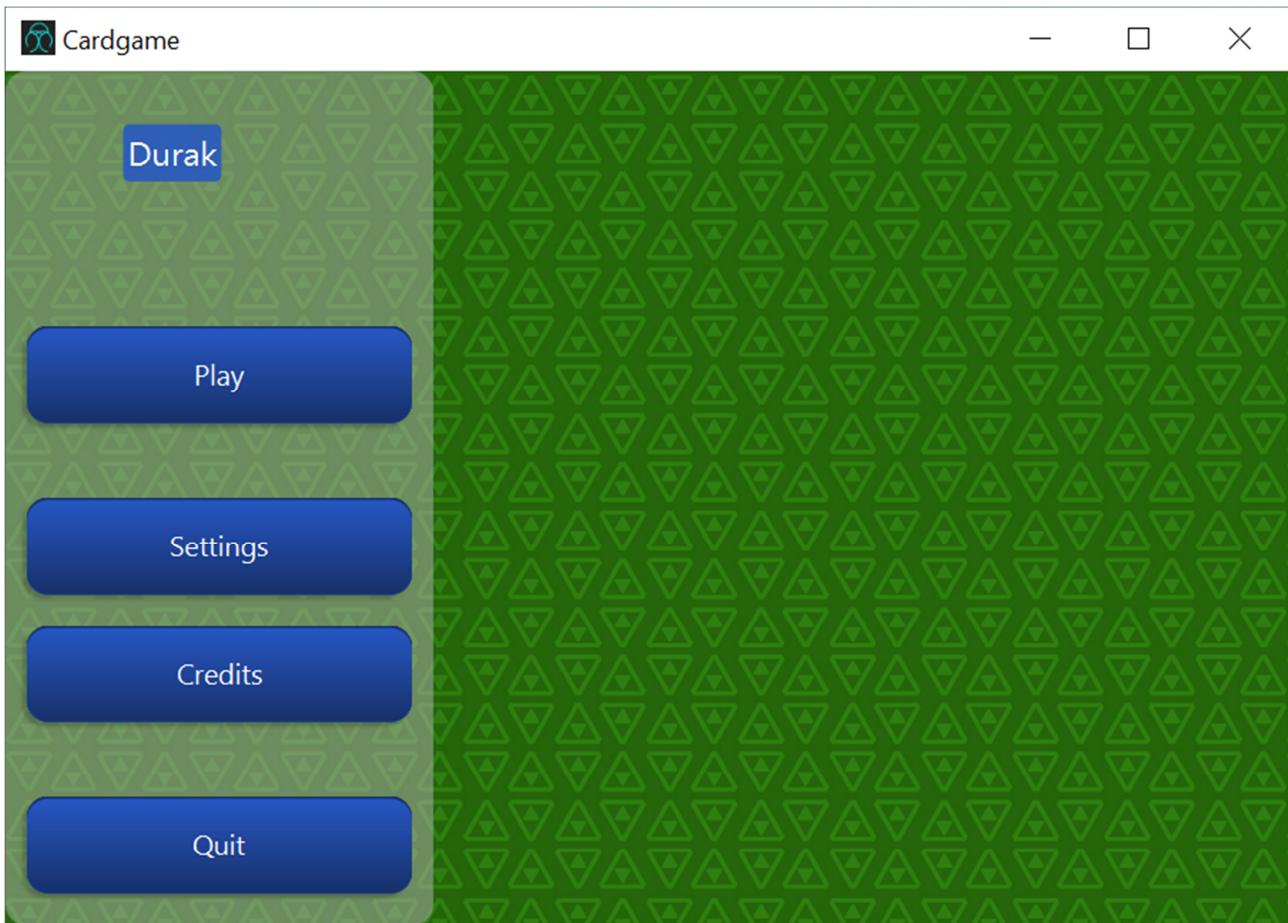
Wenn Sie java nicht zu Ihren Path-Variablen hinzufügen können müssen Sie in der Start_Server.bat den Pfad von Java angeben. Dazu bearbeiten Sie die Datei mit Notepad++ oder einem anderen, geeigneten Editor und ersetzen Zeile 1 mit folgendem:

'JAVA_HOME="C:\Program Files\Java\jdk11.0.2" java -jar BuildTools.jar'

Beachten Sie: Der Pfad zum tatsächlichen Java-Verzeichnis kann hierzu abweichen, bitte gehen Sie sicher, dass Sie den Pfad zum tatsächlichen Oberordner angeben!

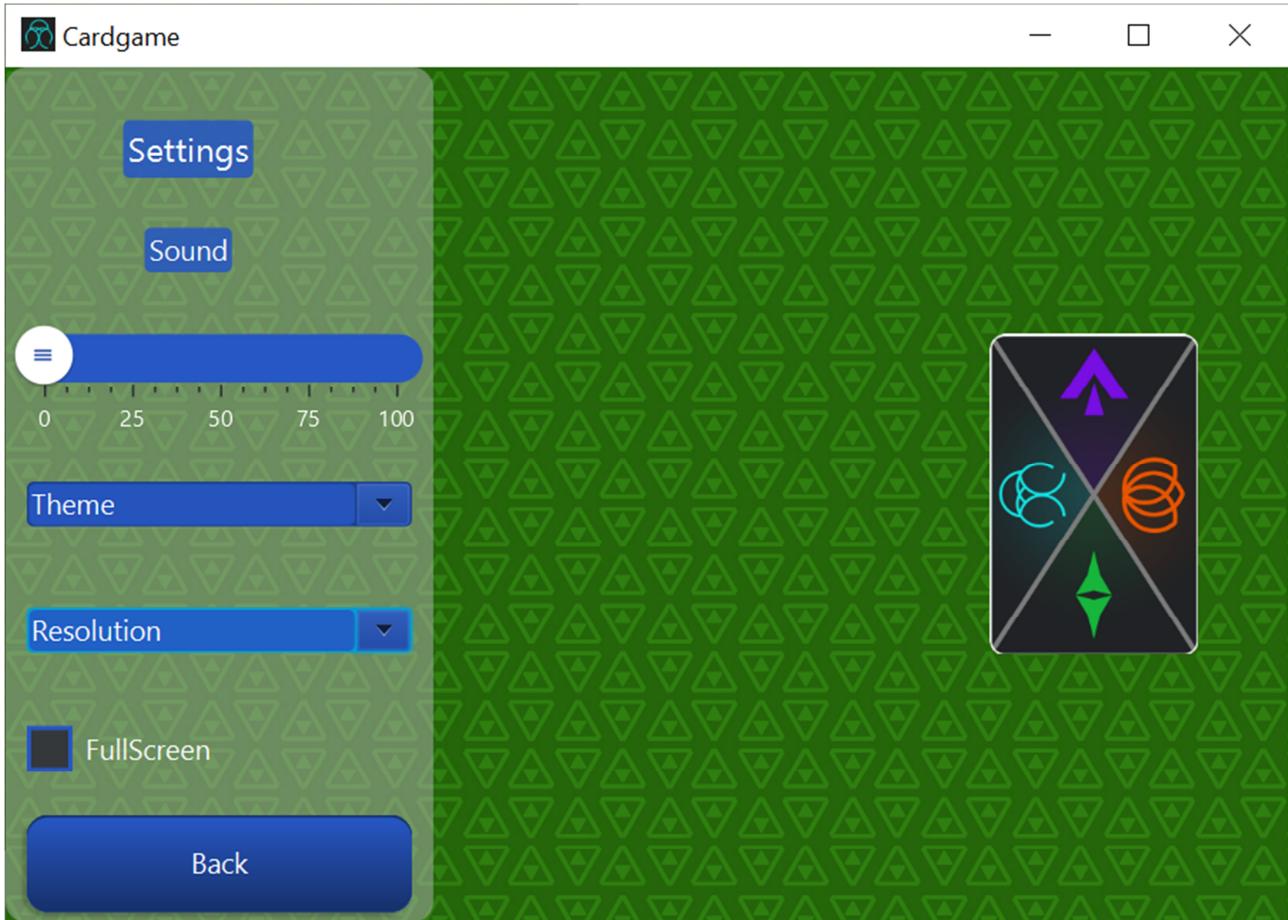
Client:

Starten Sie das Client-Programm. Auf dem Start Splashscreen klicken Sie auf Start. Warten Sie bis der Ladevorgang abgeschlossen ist. Sie befinden sich nun im Hauptmenü des Programms. Von hier haben Sie verschiedene Möglichkeiten:



Play	Klicken Sie hier um zur Spieldaten auswahl zu gelangen.
Settings	Hier gelangen Sie zu den Einstellungen, welche Audiovisuelle Optionen enthalten.
Credits	Hier gibt es weitere Informationen über die wunderbaren Entwickler.
Quit	Klicken Sie hier um das Spiel zu beenden.

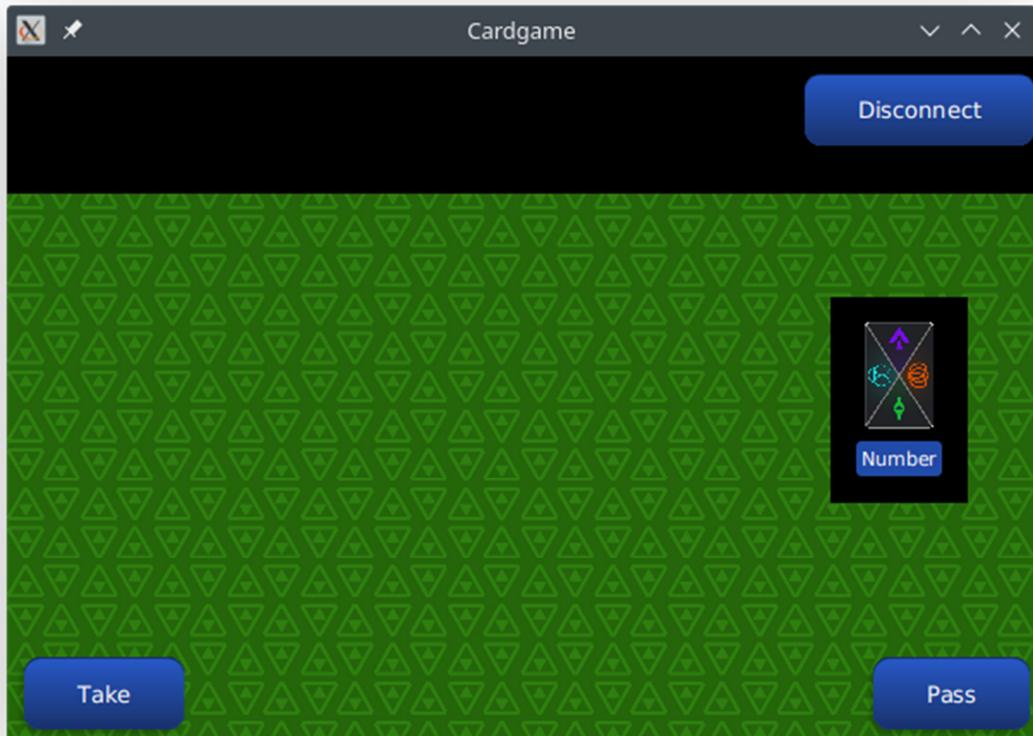
Settings:



Sound	Hier können Sie die Lautstärke der Musik und Spezialeffekte einstellen.
Theme	Diese Auswahl bietet Ihnen verschiedene Farbeinstellungen für die Menüelemente.
Resolution	Hier können Sie die Mathe ihres Fensters auf vordefinierte Werte einstellen.
Full Screen	Dieses Element wechselt zwischen dem Vollbild- und Fenstermodus.
Back	Hier gelangen Sie zurück zum Hauptmenü

Spielfeld nach Beitritt und vor Spielbeginn:

Wenn Sie dem Spiel beitreten müssen Sie unter Umständen noch auf Mitspieler warten. Das Spielfeld sieht dann wie in der folgenden Abbildung aus.



Disconnect

Hier können Sie das Spiel wieder verlassen

Spielfeld nach Beitritt und vor Spielbeginn:

Wenn alle Spieler vollzählig sind beginnt das Spiel und das Spielfeld sieht wie folgt aus:



Disconnect	Hier können Sie das Spiel wieder verlassen
Take	Karten aufnehmen
Pass	Zug abgeben
Trumpf	Farbe der Trumpfkarten
Status	Angreifer oder Verteidiger des Spielers
Mitspieler	Teilnehmende Spieler und Anzahl der Handkarten jedes Spielers
Kartenstapel	Kartenstapel und verbleibende Anzahl an Karten in dem Stapel
Handkarten	Klicken Sie auf eine entsprechende Handkarte um diese auszuspielen

Server:

Unter Windows:

Starten Sie das ServerProgramm mithilfe der Start_Server.bat, ein simpler doppelklick sollte ausreichen.

Unter Linux:

Starten sie ihre bash und gehen Sie zur Directory, an der Sie die Durak_Server.jar und die start_server.sh abgelegt haben. Starten sie die start_server.sh mit dem Befehl 'sh start_server.sh'

Die Konsole/Terminal öffnet sich und jetzt müssen Sie nur noch den Anweisungen des Servers befolgen.

Insgesamt gibt es drei Abfragen für die Erstellung des Server:

1. Enter Maxplayer Number <2-8 | Blank for 2>:
⇒ Geben Sie hier die gewünschten Anzahl an Spielern ein, die Sie für Ihren Server zulassen möchten. Der Wert muss zwischen 2 und 8 liegen (inklusiv).
2. Enter Password <blank for none>:
⇒ Sie können Ihren Server mit einem Passwort vor unerwünschten Mitspieler schützen.
Lassen Sie die Zeile frei sollten Sie keine Passwort wünschen
3. Enter server Port <1-65535> (Leave blank for Standard Port 8000):
Port Einstellungen für die Verbindung.. Standard ist 8000.

Nachdem Sie die Einstellungen für den Server gemacht haben steht nun folgendes in der Konsole:
'waiting for clients0'

Ab nun nimmt der Server Verbindungen von Clients an, diese benötigen aber auch das korrekte Passwort.

Jetzt, nachdem alle Spieler beigetreten sind, ist die Serverkonsole die sichtbare Schnittstelle dafür, welche Aktionen welcher Spieler tätigt und bedingt dafür, was der Server gerade versucht zu machen.

Nach Beendigung des Spiels taucht in der Konsole eine Abfrage auf, was nun geschehen soll:

Wenn Sie nun 'n' eingegeben wird das Spiel neugestartet und Sie können wieder wählen, wie viele Spieler im neuen Spiel teilnehmen, was für ein Passwort gesetzt werden soll und welcher Port verwendet werden soll, danach geht der Server wieder in den 'waiting for clientsX'-Status.

Wenn Sie stattdessen 'r' eingegeben haben startet der Server mit exakt den selben Einstellungen wie für das Spiel davor.

Wenn Sie 'q' eingegeben haben schließt sich der Server.

Technisches Manual

Server -> Client

Kommunikation (String)	Erklärung
Gamestate Als einziger String separiert mit Leerzeichen	<p>drawPileSize Int Größe des Nachziehstapels</p> <p>trumpColor Int Trumpf Farbe</p> <p>playerCount Int Anzahl der spielenden Spieler</p>
Variabler langer String, abhängig von der Spielerzahl.	<p>playerID Int Die ID des jeweiligen Spielers.</p> <p>playerName String Name des jeweiligen Spielers.</p> <p>amountOfHandCard Int Anzahl der Handkarten des jeweiligen Spielers.</p> <p>isAttacker Int Gibt an, ob der jeweilige Spieler ein Angreifer ist.</p> <p>isDefender Int Gibt an, ob der jeweilige Spieler ein Verteidiger ist.</p> <p>isActive Int Gibt an, ob der jeweilige Spieler aktiv ist.</p>
countVisibleCards Int visibleCards Array Int	<p>Anzahl der sichtbaren Karten</p> <p>Die Art der Sichtbaren Karten. Jede Karte hat seine eigene Karten-ID.</p>
Spezifischer Spieler. amountOfHandCard Int handCards Array Int	<p>Anzahl der Handkarten</p> <p>Art des Kartentypes, beschrieben als Karten ID</p>
wasSuccessful Int	1 falls der transfer erfolgreich verlaufen ist, 0 wenn nicht.
connection_stopped	Verbindung wurde unterbrochen.

error	Jegliche Art von Fehler.
disconnect	Der Server kickt den Client.
GameEnded	Beendigung eines Spieles.
LoggedIn	Client ist mit dem Server verbunden.

Client -> Server

Action	Welche action der Spieler ausführt.
Disconnect	Der Spieler disconnected sich vom Server.
Credentials	Bestehend aus Passwort und Name

Durak-Regelwerk

Allgemein:

- Es wird mit einem 52 Kartendeck gespielt, bestehend aus 4 Farben mit jeweils 13 Karten.
- Jedes Spiel benötigt 2 bis 8 Personen.
- Gewonnen hat diejenige Person, die es schafft ihre gesamte Hand zu leeren, nachdem das Nachziehstapel ebenfalls leer ist.
- Wertigkeit der Karten verläuft aufsteigend von 2 bis 10 gefolgt von Bube, Dame, König, Ass.

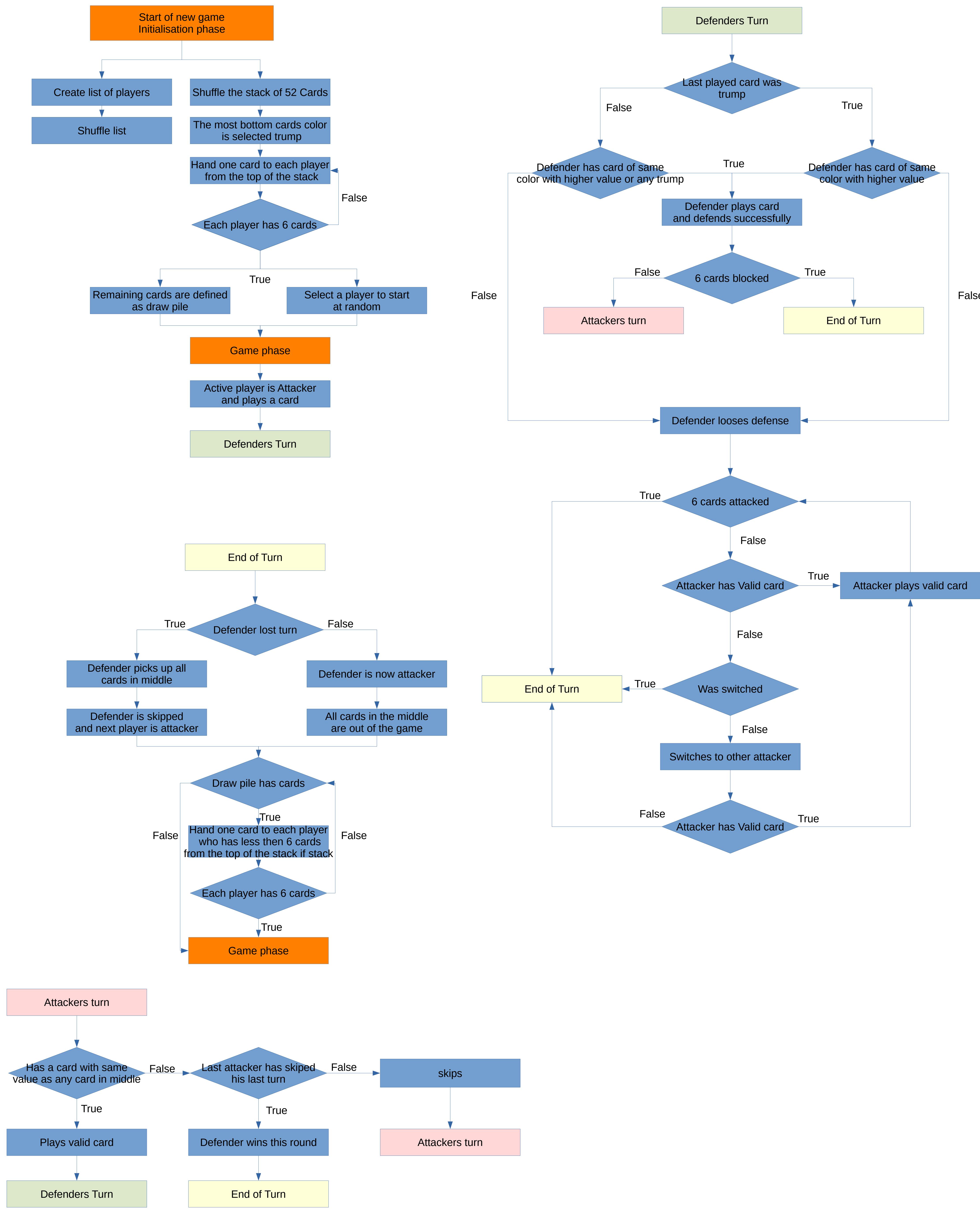
Spielbeginn:

- Aus dem Kartendeck wird ein Trumpf gezogen, dieser besteht nur aus einer der 4 Farben.
- Jeder Spieler zieht 6 Karten aus dem Nachziehstapel.
- Aus allen Spielern wird zufällig ein Spieler gewählt der anfängt.

Spielphase:

- Eröffnungszug:
 - Der aktive Spieler darf eine beliebige Karte ausspielen und ist damit der Angreifer. Der Verteidiger ist der nächste Spieler in der Reihe.
 - Gehe zur Verteidigung über.
- Verteidigung:
 - Der Verteidiger hat nun 2 Entscheidungen:
 - *Play*
 - *Take*
 - *Play*:
 - Der Verteidiger legt eine Karte gleicher Farbe mit höherer Wertigkeit oder einen Trumpf um die angreifende Karte zu Blocken.
 - Falls der Verteidiger erfolgreich verteidigt hat: Gehe zum Angriff über.
 - Wenn 6 Karten erfolgreich verteidigt wurden oder der Verteidiger keine Karten mehr auf der Hand hat, so hat der Verteidiger gesiegt. *Phasenende*.
 - Wenn der Verteidiger nichts ausspielen kann, so muss der er *Taken*.
 - *Take*:
 - Der Verteidiger nimmt alle ausgespielten Karten auf die Hand. *Phasenende*.

- Angriff:
 - Der Angreifer hat nun 2 Entscheidungen:
 - *Play*
 - *Pass*
 - Play
 - Der Angreifer darf eine weitere Karte legen, welche dieselbe Wertigkeit besitzt wie eine Bereits in der Mitte befindliche Karte.
 - Kann der Angreifer die obere Bedingung nicht erfüllen, so muss der er *Passen*.
 - Pass
 - Sofern vorhanden wird der andere Nachbar des Verteidigers nun der neue Angreifer. Ansonsten hat der Verteidiger Gewonnen.
 - Wenn beide Angreifer hintereinander *Passen*, so hat der Verteidiger gesiegt. *Phasenende*.
- Phasenende:
 - Jeder Spieler zieht aus dem Nachziehstapel, bis die Hand auf 6 Karten aufgefüllt ist. Sollte die Hand bereits 6 oder mehr Karten besitzen, so wird keine Karte gezogen.
 - Wenn das Kartendeck leer ist, werden keine neuen Karten gezogen und das Spiel wird wie gewohnt fortgesetzt.
 - Hat der Verteidiger die letzte Runde Gewonnen so wird er der neue Angreifer, anderen Falles wird er übergangen und der darauffolgende Spieler ist der neue Angreifer.
 - Zurück zum *Eröffnungszug*.



If at any point a player has no hand cards he wins and is out of the game
If its the defender the game state is changed to "End of Turn"

If at any point a player loses connection (doesn't play for 60 sec) the computer will take his place

If only one player remains he is the looser and the Game Ends