

## Dokumentation zu BlackJack (Prog Prüfungsleistung sem1/2020)

### Einleitung und kurze Erklärung des Programms

Der Zweck des Programmes sollte ein kleines Glücksspiel sein, das einen menschlichen Spieler gegen den Computer antreten lässt, indem Texteingaben in die Konsole vom Programm verarbeitet und als Spielzüge ausgeführt werden. Der Spieler sollte dabei ständig von einer Textausgabe des Programms auf dem aktuellen Stand gehalten werden. Das Spiel, das ich für dieses Projekt ausgewählt habe, ist BlackJack<sup>1</sup>.

Sobald das Programm gestartet wird, fragt es nach dem Wetteinsatz, mit dem der Spieler in der ersten Runde antreten möchte. Der Minimaleinsatz beträgt hierbei 25 Casinochips, und der Spieler kann nicht mehr setzen, als er besitzt (am Anfang standardmäßig 1000 Chips, kann im Konstruktor der Klasse Spieler geändert werden). Hat der Spieler eine Eingabe getätigt, werden die Karten ausgeteilt; der Dealer und der Spieler bekommen beide jeweils zwei Karten. Gleichzeitig werden dem Spieler die aktuellen Punktzahlen mitgeteilt. Nachdem die Karten verteilt wurden, wird der Spieler wieder um eine Eingabe gebeten, um den nächsten Spielzug auszuwählen. Im Folgenden sind die möglichen Spielzüge aufgeführt:

Karte ziehen: Eine weitere Karte auf die Hand ziehen. Der Spielzug endet, falls der Spieler über 21 Punkte kommt.

Halten: Zieht keine Karte und bleibt auf der Punktzahl, dies beendet den Spielzug des Spielers.

Verdoppeln: Falls der Spieler risikobereit ist, kann er die Wette verdoppeln und eine Karte ziehen. Dies beendet ebenfalls den Spielzug.

Split: Falls der Spieler zu Beginn des Spiels ein Paar (3-3, König-König, etc.) ausgeteilt bekommen hat, kann er seine Hand für einen möglicherweise höheren Gewinn aufteilen, da jede Hand den anfänglichen Einsatz bekommt. Macht er dies, werden beide Karten in einer separaten Hand abgelegt, die jetzt unabhängig voneinander gespielt werden können. Ab jetzt kann der Spieler für Hand 1 und Hand 2 jeweils solange Karten ziehen, bis er entweder über 21 Punkte kommt oder der Spieler als nächsten Spielzug „Halten“ wählt.

Der Spieler kann in diesem Menü außerdem die Gewinne und Verluste aus den vorherigen Runden der gleichen Spielsitzung sehen, falls er am Ende des Spiels auf „Nochmal spielen?“ mit Ja geantwortet hat.

---

<sup>1</sup> Das Programm wurde basierend auf den im Quellenverzeichnis angegebenen Regeln geschrieben.

Die Punktzahl, die eine Karte wert ist, ist bei Zahlenkarten (2-10) immer ihr Zahlenwert, bei Bildkarten (B-K) immer 10, und bei einem Ass entweder 11 oder 1, je nachdem, was für den Spieler (oder Dealer) vorteilhafter ist.

Der Spieler gewinnt das Spiel, wenn er mehr Punkte als der Dealer hat, ohne die Hand überkauft zu haben, er darf also mit seiner Hand nicht über 21 Punkte kommen.

Der Spieler gewinnt auch, falls der Dealer überkauft. Dieser zieht solange Karten, bis er mindestens 17 Punkte hat, und hört dann auf, Karten zu ziehen. Der Gewinn an Casinochips wird normalerweise im Verhältnis 1:1 ausbezahlt, falls der Spieler jedoch genau 21 Punkte hatte und gewonnen hat, wird er im Verhältnis 3:2 entlohnt.

Der Gewinn wird dem Spieler gutgeschrieben, er kann in der nächsten Runde wiederverwendet werden. Ein Unentschieden tritt dann auf, wenn der Spieler nicht überkauft hat und mit dem Dealer gleichauf ist.

## Erläuterung der main()-Funktion

Zum besseren Verständnis über die genaue Funktionsweise des Programms soll hier der Ablauf der Funktion `int main()` (Zeile 347 der Hauptdatei) durchgegangen und erläutert werden. Diese lässt sich in sechs Phasen (GameState) einteilen, die in Zeile 11 der Hauptdatei deklariert werden. Diese sind Initialisierung (INIT), Wette platzieren (BET), erstmaliges Austeilen der Karten (DEAL), das Erfragen des nächsten Spielzugs und das Ausführen von diesem (NEXTMOVE), den Gewinner festlegen und unter Umständen den Gewinn auszuschütten (PAYOUT) und Spiel beenden (QUIT).

In der INIT-Phase werden zwei Objekte der Klasse Spieler erstellt, einer davon menschlich (`Spieler s`), einer computergesteuert (`Spieler c`). Ein Objekt der Klasse Spieler wird in diesem Programm meistens zum Zwischenspeichern von Daten wie der Hand des Spielers, dem verbleibenden Geld oder dem Spielverlauf der letzten Runden benutzt. Außerdem verfügt die Klasse Spieler über einige Funktionen, die verschiedene Dinge tun, wie eine Karte auf die Hand zu ziehen, oder die Punkte der Hand zusammenzuzählen. Es gibt noch eine weitere Klasse Karte, diese speichert welches Zeichen und welche Farbe eine Karte hat, und kann diese (pseudo)zufällig generieren<sup>2</sup>. Dann wird noch die Funktion `void drawScreen(GameState state, Spieler s, Spieler c)` aufgerufen, die dafür da ist, die Textausgabe zu übernehmen.

In der darauf folgenden BET-Phase wird mit Hilfe von `int askForBetAmount(Spieler s)` der Einsatz des Spielers eingelesen, und in der darauf folgenden DEAL-Phase bekommen jeweils der Spieler und der Dealer ihre Karten ausgeteilt.

---

<sup>2</sup> Dies wird mithilfe einer `uniform_int_distribution` erreicht. Siehe auch Quellen.

Die NEXTMOVE-Phase besteht aus einem `do {} while ()`, der Spieler wird solange mit `PlayerMove askForNextMove()`; nach seinem nächsten Zug gefragt, bis er keinen weiteren Zug mehr machen kann, und der gewählte Zug wird mit `bool doMove(Spieler &s, PlayerMove next)` ausgeführt.

Falls der Spieler sich eben entschieden hat, das Spiel zu beenden, endet das Programm hier. Falls nicht, geht es über in die PAYOUT-Phase. Erst wird die verdeckte Karte des Dealers aufgedeckt, dann spielt sich der Dealer nach den oben erklärten Regeln hoch, und dann wird mit `void determineResult(Spieler& s, Spieler& c)` der Gewinner bestimmt und die tatsächlich gewonnenen Casinochips (der Gewinn ist bei einer verlorenen Runde negativ) ausgerechnet. Danach wird der Rundenzähler für die Spielstatistik um 1 erhöht und der Spieler wird gefragt, ob er die nächste Runde spielen oder das Spiel beenden will.