

## Programmwurf Gruppe B

Sie wollen ihre Chancen im Casino verbessern, indem Sie mit dem Computer ihre Chancen beim Poker errechnen.

Die Gewinnchancen der Karten beim Poker finden sie in der Wikipedia unter:

<http://de.wikipedia.org/wiki/Poker>

In den Spielcasinos wird die Variante **Seven Card Stud** gespielt. Die allgemeine Erklärung des Spiels finden sie bei:

[http://de.wikipedia.org/wiki/Seven\\_Card\\_Stud](http://de.wikipedia.org/wiki/Seven_Card_Stud)

Sie sollen nun nicht ein komplettes Pokerspiel programmieren, sondern nur auf Basis der sichtbaren Karten die Chancen für die ersten vier Gewinnchancen (Ein Paar, Zwei Paare, Drilling und Straße) für Sie und ihre Gegner berechnen.

Die Vorgehensweise (Kartenausgabe, abweichend von den Regeln in der Wikipedia)

- In der ersten Runde erhält jeder Spieler drei Karten: eine nur für ihn sichtbare, zwei offen sichtbare. Für sie selbst sind natürlich alle Karten bekannt, da sie ihre eigene verdeckte Karte sehen können. Bei den Gegnern dagegen nur die offen sichtbaren.
- In den Runden zwei bis vier kommt noch jeweils eine offene Karte dazu
- In der letzten Runde erhält jeder Spieler eine verdeckte Karte
- In jeder Runde kann ein Spieler aussteigen.
- Es wird ein Kartenspiel mit 52 Karten verwendet. Das limitiert die Anzahl der auszugebenden Karten auf  $\text{Int}(52/\text{Spielerzahl})$ , z.B. bei 9 Spielern auf maximal 5.

In ihr Computerprogramm sollen die bekannten offenen Karten der gegnerischen Spieler und alle Karten, die sie haben, eingetragen werden. In jeder Runde sollten die Gewinnchancen der Gegner und Ihnen für die obigen Kombinationen ermittelt werden. Sie können auch die Regeln nach Wikipedia einsetzen wenn sie dies wollen.

Dies geschieht auf Basis der Informationen die sie von den offen sichtbaren Karten und ihren eigenen Karten haben. Das könnten z.B. die nach drei Runden vorliegenden Informationen sein:

**Beispiel:** 6 Spieler, 24 Karten wurden ausgegeben. Von 19 Karten ist ihnen der Wert bekannt (4 eigene und 3x5 offen sichtbare). Darunter sind drei Buben. Zwei bei Ihnen. Einer bei einem Gegner. Eine Karte pro Spieler sind Verdeckt. Noch drei Karten werden gezogen. (Ende Runde 2)

**Gesucht:** Chance für drei Buben bei Ihnen und zwei Buben beim Gegner mit einem Buben.

Von 33 Karten ist der Wert nicht bekannt. (5 verdeckte der anderen Spieler aus der ersten Runde und noch 28 auszugebende). Diese enthalten den Buben. Es gibt noch drei Karten für Sie zu ziehen. Die Wahrscheinlichkeit beträgt also  $3/33$ . Der Gegner mit dem Einen Buben hat noch eine verdeckte Karte auch diese könnte den Buben enthalten. Seine Chance ist also höher und liegt bei  $4/33$ . Bei Karten bei denen auch die Farbe wichtig ist, muss natürlich auch nach dieser als zweites Kriterium differenziert werden.

## Optionales

Nicht notwendig, aber wünschenswert wären folgende Punkte:

Berücksichtigen Sie, dass Gegner aussteigen können. (keine neuen Karten werden für diese Personen ausgegeben)

Grafische Gestaltung: In der Vorlesung sind wir nicht mehr dazu gekommen Bilder in Canvas Objekte einzubinden. Dies ist jedoch im Skript beschrieben. So ist es möglich Abbildungen der Karten einzuladen, um sie zu visualisieren. Entsprechende Abbildungen gibt es im Internet, z.B. unter <http://www.artpoker.net/poker-icons/poker-cards.zip>

(diese gifs müssen, wie andere Dateitypen in das bmp Format konvertiert werden. Das kann mit dem Irfanview der auf der DHBW installiert ist erfolgen (Menüpunkt Datei → Batch Konvertierung. Achten sie darauf, dass die Bilder sich auch im Archiv befinden, das sie schicken)

Sie können das Programm um die Ermittlung der Gewinnchancen für die anderen Kombinationen (Flush ... Royal Flush) ergänzen.

## Anforderungen an das Programm

Neben der primären korrekten Lösung der Aufgabe sollte ihr Programm auch so gestaltet sein, dass sowohl User mit ihm einfach umgehen können, wie auch andere Personen, die ihren Quelltext verwenden müssen, leicht die Funktionsweise nachvollziehen können, bzw. ihn einfach wiederverwenden können.

- Der Code sollte gut strukturiert und übersichtlich sein. Verlagern sie Funktionalitäten in eigene Funktionen und Prozeduren, tauschen sie Daten über Parameter aus.
- Die Funktionen sollten ausreichend kommentiert und dokumentiert sein.
- Das Programm ist gegen Fehleingaben abzusichern (z.B. nicht numerische Werte, Eingaben sind auf Plausibilität zu prüfen.)
- Das Programm sollte den Kriterien für benutzerfreundliche Anwendungen genügen und intuitiv zu bedienen sein.
- Der Quellcode sollte auch ohne Kommentierung leicht lesbar und verständlich sein.
- Umlaute in Bezeichnern sind nicht erlaubt.

## Abgabe

Programmwürfe sind wie folgt zu kennzeichnen:

- Erstellen sie einen eigenen Ordner für das Programm.
- Speichern sie alle Projektdateien nur in diesem Ordner. So vermeiden Sie, dass sie später fehlende Teile haben oder Verknüpfungen sich auf absolute Verzeichnispfade beziehen, die eine Compilierung unmöglich machen.
- Benennen sie das Projekt nach dem Schema „**B\_Vorname\_Nachname**“.
- Benennen Sie niemals Dateien außerhalb Delphi um. Das Programm ist dann nicht mehr compilierbar.
- Packen sie alle zum Programm gehörenden Dateien in ein Zip Archiv. Dazu gehören mindestens:
  - Die Projektdatei mit der Endung .dpr
  - Pro Fenster eine Formulardatei mit der Endung .dfm
  - Pro Fenster eine Quelltextdatei mit der Endung .pas
  - Die Ressourcen Datei mit der Endung .res
- Testen Sie das Programm vor der Abgabe eingehend. Sie haben nur eine Abgabe, die benotet wird.
- Senden sie das Zip Archiv an [bl@delphi-vorlesung.de](mailto:bl@delphi-vorlesung.de)

**Sollte der Programmwurf nicht eigenständig erarbeitet sein und mehrere Studenten denselben Programmwurf abgeben, so gilt die Prüfung bei diesen Studenten als nicht bestanden.**