

Praktikum aus Programmierung

Dr. Michael Hahsler

WS 2006/2007

# **Dokumentation des Projektes**

## **Texas Hold'em Poker**

Michael Röhrenbacher (9801753)

## **Inhaltsverzeichnis**

Abbildungsverzeichnis.....	2
1. Problemdefinition.....	3
1.1. Ziel.....	3
1.2. Darstellung.....	3
1.3. Spielanleitung.....	3
2. Analyse und Modellierung.....	4
2.1. Einschränkung der Problemdefinition.....	4
2.2. Use-Case.....	5
2.2.1. Use-Case Beschreibung.....	5
2.2.2. Use-Case Diagramm.....	5
2.3. Sequenzdiagramm.....	6
2.4. Klassendiagramm.....	7
3. Beschreibung der Klassen anhand einzelner Attribute und Methoden.....	7
4. Benützung des Programms.....	9
5. Installation und Wartung.....	10

## **Abbildungsverzeichnis**

Abbildung 1: Use-Case Diagramm.....	5
Abbildung 2: Sequenzdiagramm.....	6
Abbildung 3: Klassendiagramm.....	7
Abbildung 4: GUI nach Start des Programms.....	9
Abbildung 5: GUI nach Betätigung des bet Buttons.....	10

# **1. Problemdefinition**

## **1.1. Ziel**

Das Ziel dieses Projektes ist es die populärste Form des Poker – Spieles, Texas Hold'em in Java zu programmieren.

Es soll dem Spieler möglich sein, nach den allgemein gültigen Regeln gegen den Computer zu pokern. Dabei soll zunächst allen Spielern (1 Spieler – 4 Spielertypen des Computers) ein vom Spieler gewähltes Guthaben an Chips gutgeschrieben werden. Danach kann der Spieler die maximalen Limits für die Einsätze festlegen. Während des Spiels scheidet ein Spieler aus wenn er über kein Guthaben mehr verfügt. Der Spieler der schlussendlich übrig bleibt ist der Sieger des Spiels und hat somit den gesamten Einsatz gewonnen.

## **1.2. Darstellung**

Für das Spiel wird ein virtueller Kartensatz erzeugt. Dieser besteht aus 52 Karten – jeweils eine 2, 3, 4, 5, 6, 7, 8, 9, 10, Bub, Dame, König, As von jeder der vier Farben (Herz, Karo, Pik, Kreuz). Dabei soll zunächst der gesamte zugedeckte Kartensatz angezeigt werden. Nach dem Austeilen der Karten werden die aufgedeckten Karten in einer zweidimensionalen Ansicht (von oben) dargestellt. Die Spielchips werden zweidimensional in drei verschiedenen Farben (rot, grün, gelb) darstellt, wobei ein roter Chip 5, ein grüner Chip 10 und ein gelber Chip 20 Geldeinheiten wert ist.

## **1.3. Spielanleitung**

Für das Spiel gelten die allgemein gültigen Texas Hold'em Regeln. Das heißt jeder Spieler bekommt zunächst 2 Karten. Jeder Spieler sieht nur seine beiden Karten und kann danach entscheiden ob er einen Einsatz macht oder die Karten ablehnt und somit für diese Runde ausscheidet. Allerdings muss pro Runde ein Spieler einen großen Grundeinsatz und ein Spieler einen kleinen Grundeinsatz leisten. Per Zufallsprinzip wird zunächst entschieden wer zuerst den großen Grundeinsatz leisten muss. Der Spieler der in der Runde vor diesen gelosten Spieler sitzt ist damit automatisch zum kleinen Grundeinsatz verpflichtet. Bei der nächsten Runde wandert die Pflicht sowohl den großen als auch den kleinen Grundeinsatz leisten zu müssen einen Spieler weiter.

Jene Spieler die einen Einsatz leisten bleiben in dieser Runde dem Spiel erhalten und haben die Möglichkeit den Einsatz nochmals zu erhöhen, wobei bei dieser Erhöhungsrunde wieder ein Spieler aussteigen darf, wobei sein bereits geleisteter Einsatz mit den Einsätzen der

anderen Spieler und den Grundeinsätzen im Pot bleibt. Danach werden drei Karten offen auf den Tisch aufgelegt. Es folgt für die Spieler wieder die Möglichkeit zu setzen, zu erhöhen oder auszusteigen. Danach wird eine 4. Karte aufgedeckt, außer ein Spieler geht „All in“ was bedeutet dass dieser seinen gesamten Einsatz setzt – dann werden nämlich jeweils die beiden Karten der sich noch im Spiel befindlichen Spieler aufgedeckt und zu den drei Karten am Tisch 2 weitere aufgedeckt. Aus den 5 Karten am Tisch und jeweils den beiden Karten der Spieler werden die 5 besten Karten zum Blatt zusammengefasst und der Spieler mit dem höchsten Blatt gewinnt. Wurde die „All in“ Option von keinem der Spieler genutzt erfolgt die gleiche Möglichkeit nachdem die 4. Karte am Tisch aufgedeckt wurde. Es kann wieder gesetzt, erhöht oder ausgestiegen werden. Wurde die „All in“ Option auch in dieser Runde nicht ausgenutzt wird eine 5. Karte am Tisch aufgedeckt. Nun kann jeder Spieler nochmals setzen, erhöhen, aussteigen und „All in“ gehen. Danach werden die beiden Karten der jeweiligen Spieler aufgedeckt und die 5 besten der 7 Karten ergeben somit wieder das Blatt und der Spieler mit dem besten Blatt bekommt den gesamten Einsatz gutgeschrieben. Dann kann eine neue Spielrunde gestartet werden. Die Güte des Blattes erfolgt in aufsteigender Reihenfolge von höchster Karte, Paar, Doppel-Paar, Drilling, Straße, Flush, Full House, Poker, Straight Flush, Royal Flush.

## **2. Analyse und Modellierung**

### **2.1. Einschränkung der Problemdefinition**

Die Grundeinsätze sollen entfallen und das Spiel auf no-limit Texas Hold'em eingeschränkt werden. Dadurch steht es jedem Spieler frei die Höhe seiner Einsätze entsprechend seinem Guthaben fest zu legen. Grundsätzlich gliedert sich das Spiel in fünf Phasen:

- Preflop: Jeder Spieler kennt seine beiden Karten, aber die Gemeinschaftskarten bleiben verdeckt
- Flop: Die ersten drei Gemeinschaftskarten werden aufgedeckt
- Turn: Die vierte Gemeinschaftskarte wird aufgedeckt
- River: Die fünfte Gemeinschaftskarte wird aufgedeckt
- Showdown: Nach dem alle Einsätze gemacht wurden erfolgt die Auswertung zur Ermittlung des Siegers der Runde

Aus Gründen der Vereinfachung soll die Ausführung in diesem Projekt auf 3 Phasen beschränkt werden, indem die Phasen Flop und Turn übersprungen werden. Weiters ist keine Erhöhung der Einsätze möglich nachdem diese getätigt wurden. Außerdem soll auf eine grafische Darstellung der Chips verzichtet werden und das Guthaben wertmäßig dargestellt werden. Die Anzahl der Spieler wird auf 4 (1 Poker-Spieler und 3 Computer-Spieler, die sich durch unterschiedliche Strategien kennzeichnen) beschränkt. Der Poker

Spieler kann setzen oder aussteigen. Eine Beschreibung der Strategien der 3 Computer-Spieler erfolgt im 3. Kapitel.

## 2.2. Use-Case

### 2.2.1. Use-Case Beschreibung

*Use-Case Name:* Pokern

*Akteur:* Poker-Spieler

*Vorbedingungen:* Aufgedrehter PC; eclipse installiert

*Auslöser:* Spaß am Poker-Spiel

*Ablaufbeschreibung:*

- Poker-Spieler hat Lust auf Spiel
- Der Spieler macht seine Einsätze und spielt solange er Lust hat
- Ist das Guthaben des Spielers aufgebraucht startet dieser ein neues Spiel
- Verlässt den Spieler die Freude am Spiel beendet er dieses

*Instanz:* Michael hat Langeweile und möchte wieder einmal Pokern. Da alle seine Poker-Freunde durch Lernstress verhindert sind dreht er den Computer auf und startet das Pokerprogramm in eclipse. Er spielt solange er Lust hat, danach beendet er das Spiel.

*Ergebnisse:* Langeweile wurde beendet oder zumindest unterbrochen

*Nicht-funktionale Anforderungen:* Zufriedenheit mit Spiel

### 2.2.2. Use-Case Diagramm

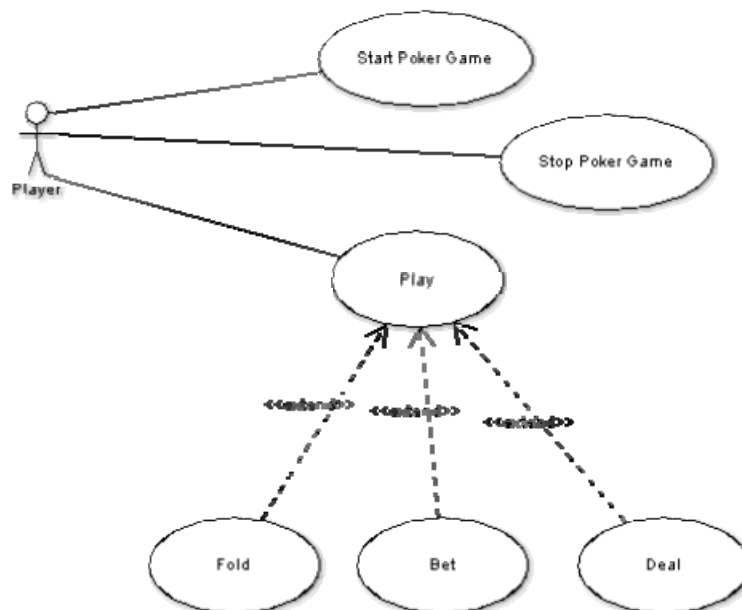


Abbildung 1: Use-Case Diagramm

2.3. Sequenzdiagramm

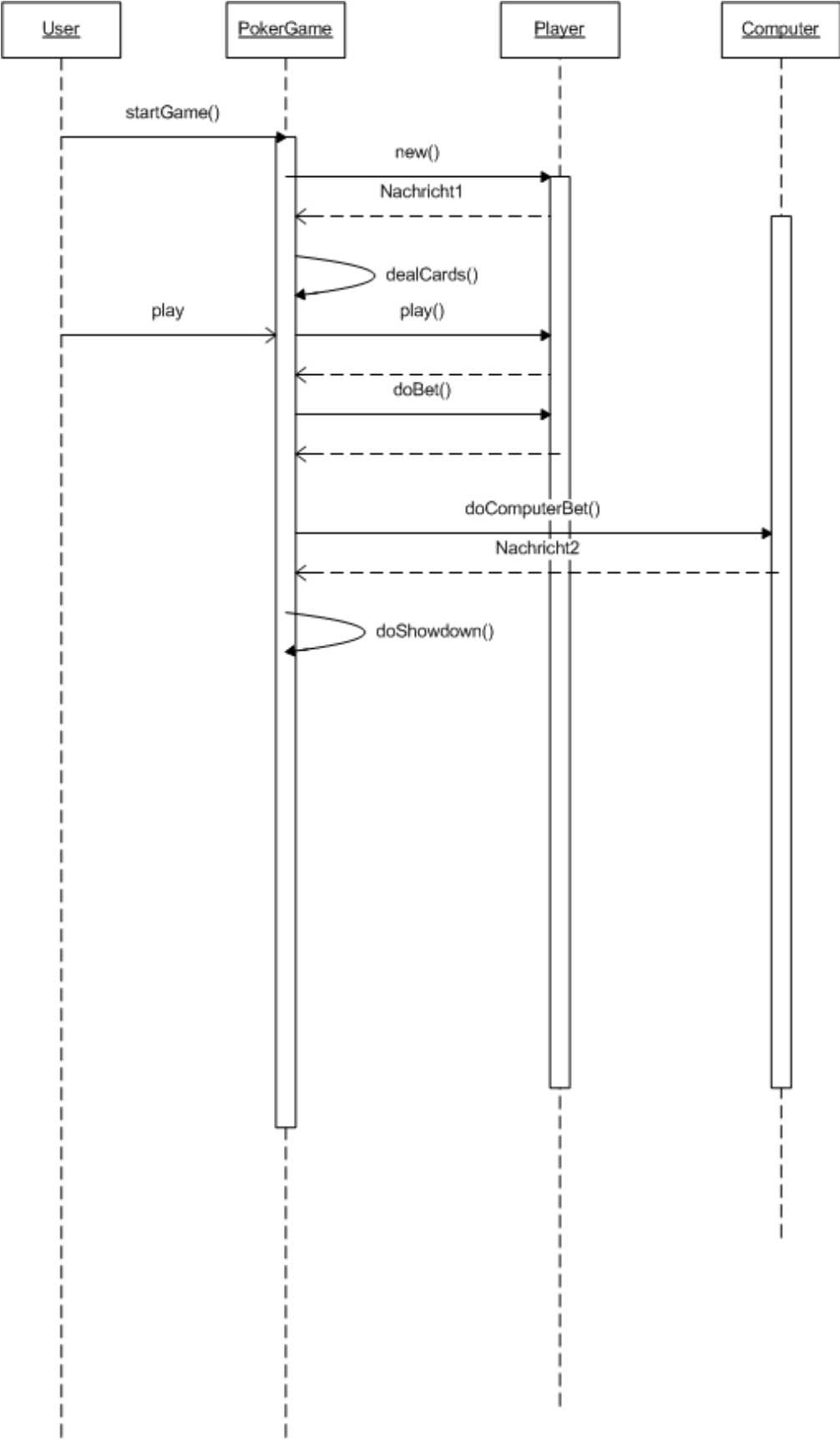


Abbildung 2: Sequenzdiagramm

## 2.4. Klassendiagramm

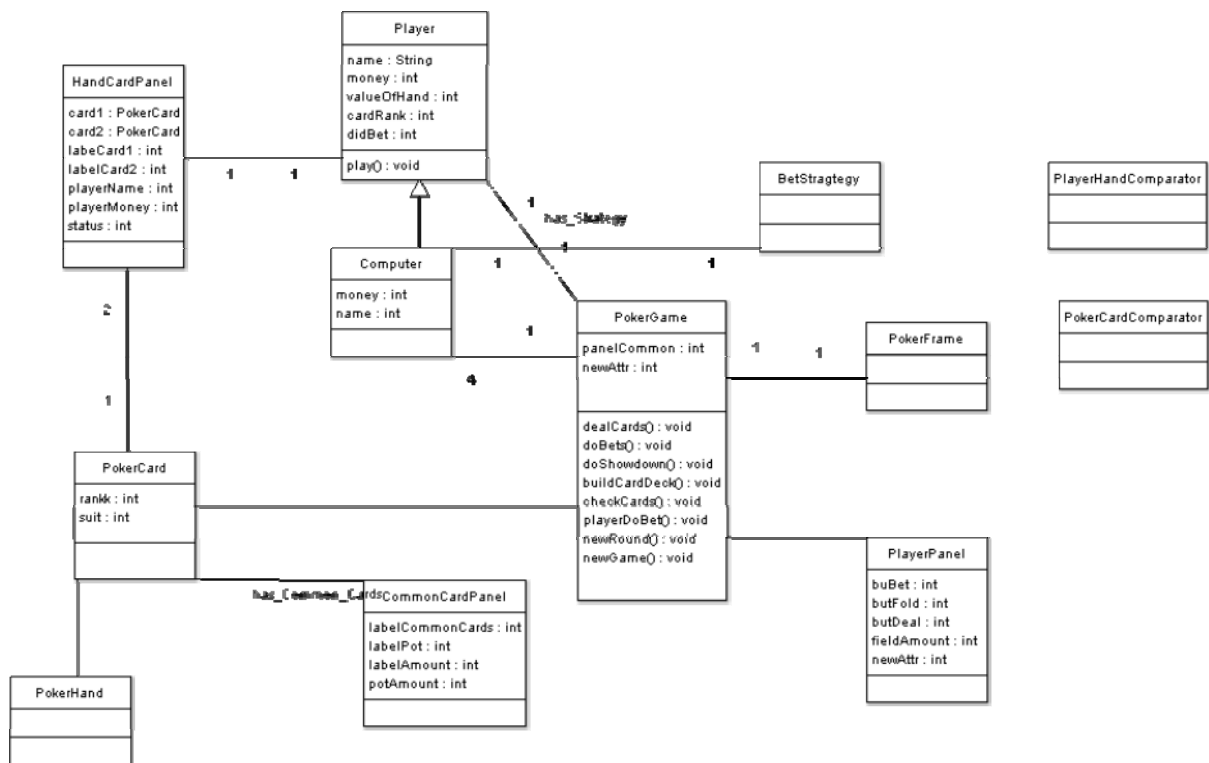


Abbildung 3: Klassendiagramm

## 3. Beschreibung der Klassen anhand einzelner Attribute und Methoden

**BetStrategy:** Für jeden Computer-Spieler wird eine Strategie implementiert die sein Verhalten in verschiedenen Situationen festlegt. *stratRandom* nimmt am Spiel teil wenn die Summe des Wertes beider Karten gerade ist, *stratAlways* nimmt immer am Spiel teil und *stratGoodHand* nimmt am Spiel teil wenn mindestens eine der beiden Karten größer als 10 (Bub, Dame, König, As) ist.

**CommonCardPanel:** Legt neuen Vector mit Gemeinschaftskarten an. Durch die Methode *showCard* werden die Karten aufgedeckt. Die Methode *hideCard* verdeckt die Karten und zeigt stattdessen ein X an.

**Computer:** Diese Klasse erbt von der Klasse *Player* und legt die Strategie für die Computerspieler fest.

**HandCardPanel:** Erzeugt einen Rahmen und fügt dem Panel die *labels*, *playerName*, *playerMoney*, *labelCard1*, *labelCard2* und *status* hinzu.

**Player:** Legt ein neues *HandCardPanel* an. Die Methode *doBet* ermöglicht dem Spieler am Spiel teilzunehmen, während die Methode *doFold* den Ausstieg ermöglicht. Die Methode *resetStatus* erneuert den Status und *resetPlayer* setzt alle Werte zurück wenn ein neues Pokerspiel gestartet wird.

**PlayerHandComparator:** Diese Klasse vergleicht die Spieler nach den sortierten Karten und bewertet diese nach dem *valueOfHand* – also dem Wert des Blattes (so wird beispielsweise ein Poker höher bewertet als ein Drilling). Weisen die Karten beider Spieler den gleichen Wert auf (beispielsweise beide Spieler haben einen Poker) wird der *cardRank* – also der Rang der Karten - verglichen. Dabei wird geprüft welcher Spieler Karten höheren Ranges besitzt (beispielsweise wird ein As-Poker höher bewertet als ein Bub-Poker).

**PlayerPanel:** Diese Klasse erzeugt die Buttons *bet*, *fold* und *deal*. Weiters legt sie fest was passiert wenn diese angeklickt werden. Wird der *fold* Button geklickt wird der Schritt des Spieler setzen ausgelassen und jeder Computer-Spieler setzt einen Einsatz von 100. Wird auf *deal* geklickt startet eine neue Runde.

**Poker:** Diese Klasse beinhaltet die main Methode und legt die Klasse *PokerFrame* an.

**PokerCard:** Diese Klasse legt den Wert (durch den Rang 2, 3, 4, 5, 6, 7, 8, 9, 10, Bub, Dame, König, As) und die Farbe (Kreuz, Karo, Herz, Pik) der einzelnen Karten fest.

**PokerCardComparator:** Diese Klasse enthält die Methode *compare*, welche Objekte vom Typ *PokerCard* vergleicht und sortiert.

**PokerFrame:** Diese Klasse setzt das ausgegebene Layout fest. Dabei werden die Karten der Computer-Spieler nach Art des Spielers auf den linken, oberen und rechten Rand gesetzt. Die Karten des Spielers werden am unteren Rand und die Gemeinschaftskarten zentral angeordnet.

**PokerGame:** Dies ist die umfangreichste Klasse des Programms. Es werden die Spieler angelegt. Weiters wird das Fenster für die Gemeinschaftskarten erzeugt und der Konstruktor für das *CommonCardPanel* aufgerufen. Weiters wird das Karten Deck mit 52 Karten (jede Farbe 2 bis As) erzeugt in dem ein Vector mit Objekten des Typs *PokerCard* angelegt wird. Die Methode *dealCards* teilt die Karten aus. Zuerst werden die Karten gemischt und dann wird die 1. Karte für jeden Spieler ausgegeben. Danach erhält jeder Spieler die 2. Karte. Im Anschluss werden die Gemeinschaftskarten ausgeteilt. Die Methode *checkCards* erzeugt einen Vector mit 7 Karten und filtert die 5 besten Karten daraus aus. Dabei werden alle 7 Karten nach dem Wert des Blattes überprüft. Ein *flush* wird überprüft indem kontrolliert wird ob 5 Karten die gleiche Farbe haben. Eine *straight* wird kontrolliert indem die Karten nach Höhe sortiert werden und dann überprüft wird ob der Wert der 2. Karte um 1 höher ist als der Wert der zweiten Karte. Wenn dies der Fall ist wird weiter geprüft ob die 3. Karte um 1 höher ist als die 2. Karte, usw. Trifft die Wertsteigerung um 1 für 5 hinter einander folgende Karten zu ist die Straße verwirklicht. Besteht die Straße aus den Karten 10, Bub, Dame, König, As realisiert wird *royal* zugewiesen. Haben die Karten der Straße die gleiche Farbe ist ein *straightFlush* realisiert. Einem Royal Flush wird daher *straightFlush* && *royal* zugewiesen. Weiters wird überprüft ob die 1. Karte gleich den Karten 2-7, die 2. Karte gleich den Karten 3-7, usw. ist. Enthält der Vector 4 gleiche Karten wurde ein Poker ausgeteilt und es wird *four*

zugewiesen. Enthält der Vector 3 gleiche Karten wird *three* zugewiesen. Wurden nur 2 gleiche Karten ausgeteilt wird *pair* zugewiesen. Wurden 3 gleiche Karten ausgeteilt wird in weiterer Folge noch überprüft ob der Vector ein zusätzliches Pärchen enthält. Ist dies der Fall wird *pair && three* zugewiesen. Sind in dem Vector 2 Pärchen enthalten wird *twoPairs* zugewiesen. Danach wird den Blättern ein Wert nach folgender Verteilung zugeordnet: *straightFlush && royal* = 9; *straightFlush* = 8; *four* = 7; *pair && three* = 6; *flush* = 5; *straight* = 4; *three* = 3; *twoPairs* = 2; *pair* = 1. Anderen Blättern wird „*high card*“ und der Wert 0 zugeordnet. Die Methode *doShowDown* zeigt alle Karten an und rechnet für jeden setzenden Spieler den Wert des Blattes aus. Sortiert wird durch den *PlayerHandComparator*. Die Methode *newGame* startet ein neues Spiel.

#### 4. Benützung des Programms

Nach dem Starten des Programms wird folgendes Fenster angezeigt:

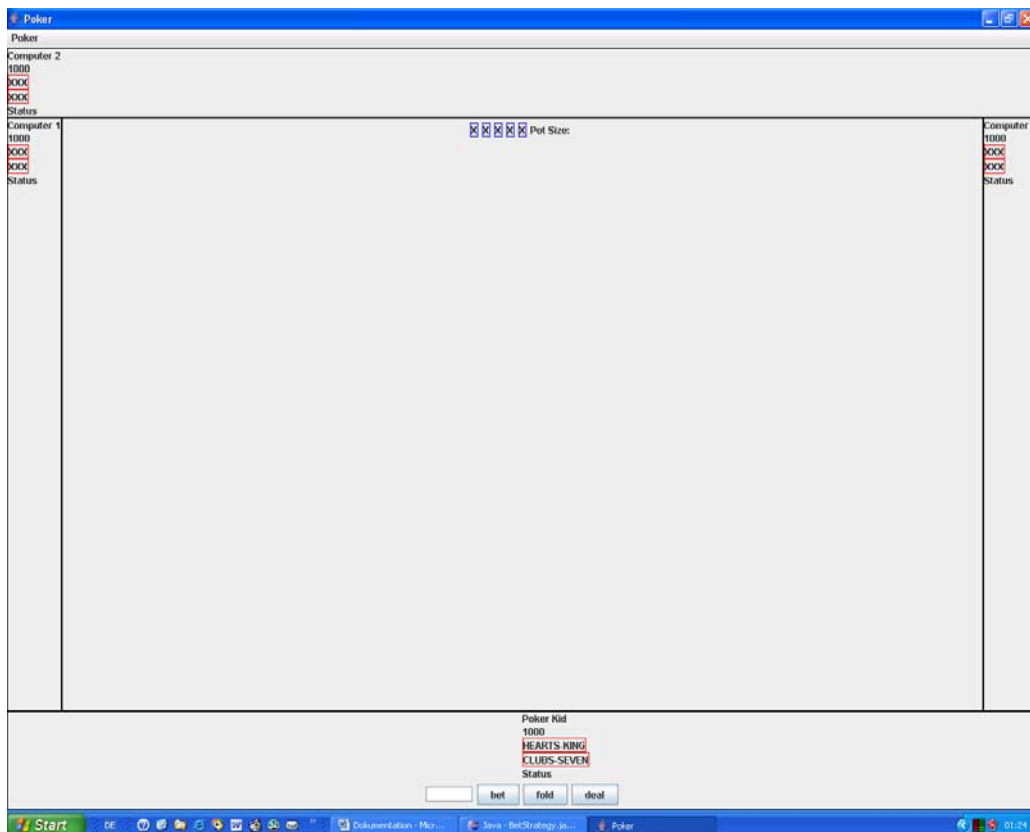


Abbildung 4: GUI nach Start des Programms

Auf der linken Seite werden die *HandCardPanels* von Computer-Spieler 1 und darüber Computer-Spieler 2 angezeigt. Auf der rechten Seite jenes von Computer-Spieler 3 und unterhalb das vom Spieler. Dabei ist der Name, das Guthaben, die Karten (nur jene des Spieler sind aufgedeckt) und der Status ersichtlich. Nun hat der Spieler die Möglichkeit zu setzen, aus zu lassen oder neue Karten zu geben. Lässt der Spieler aus muss er auf *fold*

klicken und es spielen nur die Computer-Spieler gegeneinander, wobei jeder einen Einsatz von 100 leistet. Setzt der Spieler so gibt dieser den Betrag in das linke Fenster der untersten Zeile ein und klickt auf *bet*. Der eingesetzte Betrag wird von seinem Guthaben abgebucht. Die Computer-Spieler setzen den gleichen Betrag, wenn ihre Strategie eine Beteiligung am Spiel vorgibt. Es werden nun alle Karten aufgedeckt, der Sieger ermittelt und diesem der Wert des Pots gutgeschrieben. Klickt der Benutzer auf *deal* werden die Karten neu gegeben. In den oberen beiden Menüleisten kann das Spiel beendet, oder ein neues Spiel gestartet werden.

Wird ein Betrag von 15 gesetzt erscheint folgendes Fenster:

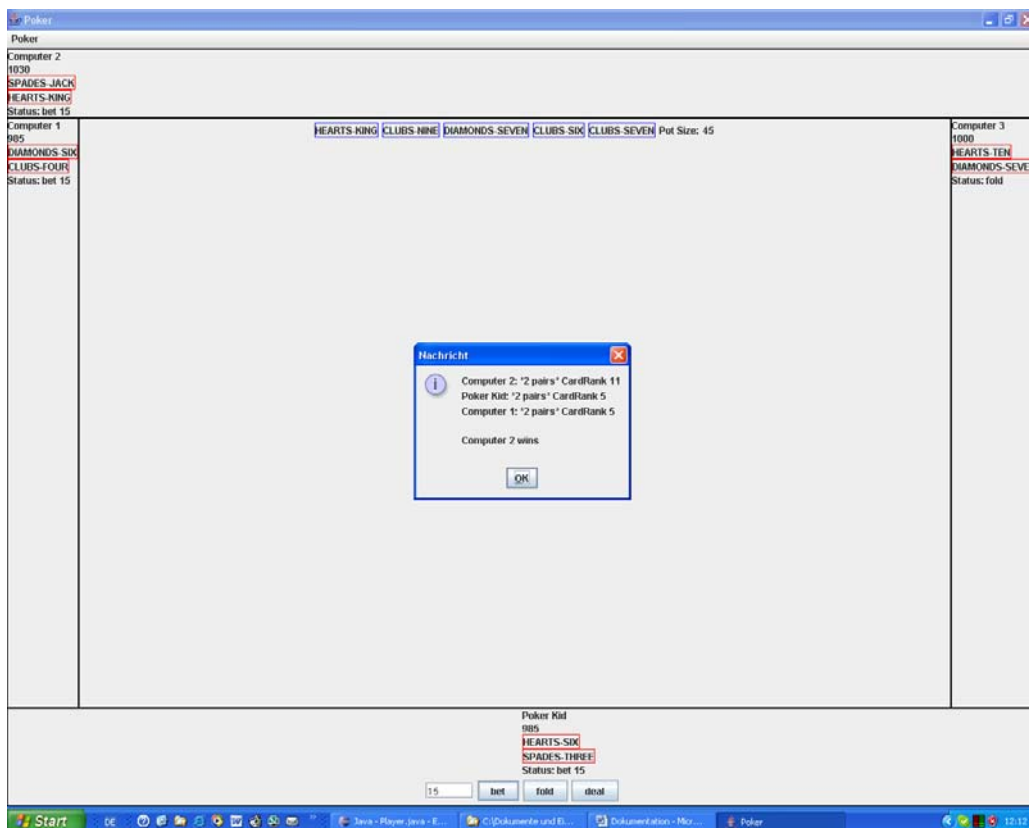


Abbildung 5: GUI nach Betätigung des bet Buttons

## 5. Installation und Wartung

Das Programm kann im Dos-Fenster bereits ausgeführt werden. Eine Wartung ist für das Spiel nicht notwendig und daher auch nicht vorgesehen. Anhand des Prototypen wurden zwei kleine Mängel (das Weiterspielen bei negativem Guthaben und die Möglichkeit auf deal zu klicken wenn die Karten bereits ausgeteilt wurden) festgestellt. Diese Mängel wurden bereits behoben – so kann jetzt nicht mehr Geld gesetzt werden als Guthaben vorhanden ist [dafür wird in der Klasse *PokerGame*, durch die Methode *playerDoBet* geprüft ob der gesetzte Betrag kleiner oder gleich dem vorhandenen Guthaben ist] und es kann erst auf

deal geklickt werden wenn zuvor bet oder fold geklickt wurde [dafür wurde in der Klasse *Player* eine Instanzvariable deklariert, welche den Status (true/false) festsetzt – dieser wird in der Klasse *PokerGame* durch die Methode *newRound* abgeprüft]. Außerdem wird jetzt mittels Nachricht der Wert des Blattes, der Rang der Karten und der Rundensieger angezeigt (vgl. Abbildung 5). Dies erfolgt durch die Variable *output* die durch die Methode *showMessageDialog* ausgegeben wird.