*Problemdefinition*

Datum

26.Oktober 2004

Projektbezeichnung

Softwareversion von Würfelpoker

Autor

Name: Ing. Andreas Schlichtinger

Matrikelnummer: 9251410

E-mail: [**a.schlichtinger@vkmb.at**](mailto:a.schlichtinger@vkmb.at)

Kurzbeschreibung

In diesem Projekt wird das Würfelspiel „Würfelpoker“ in eine Softwareversion gebracht. In der Grundversion wird die Ein- und Ausgabe textbasiert sein. In einer Upgradeversion soll es möglich sein, das Spiel über ein komfortables User-Interface bedienen zu können. Die Upgradeversion wird voraussichtlich im Rechnerpraktikum aus Programmierung nicht mehr realisiert werden können.

Projektziele

Erstellung einer funktionsfähigen, leicht zu bedienenden Software und einer Anleitung für den im Würfelpoker nicht so versierten Benutzer. Des weiteren Aufbau von Know-how im Bereich Design und Implementierung von Java.

Projektumfang

Das Projekt läuft im Rahmen der Lehrveranstaltung Rechnerpraktikum aus Programmierung, das ein Semester dauern wird. Somit wird der Zeitaufwand in etwa 12 Wochen betragen.

Lösungsvorschlag

Kern des Spieles Würfelpoker wird ein Programm in der Programmiersprache Java sein. Dieses Programm kommuniziert mit dem Benutzer über eine textbasierte Eingabe. In einen weiteren Schritt könnte die textbasierte Eingabe durch eine HTML-Oberfläche ersetzt werden, sodass die Würfel visualisiert werden können und die Eingabe komfortabler wäre.

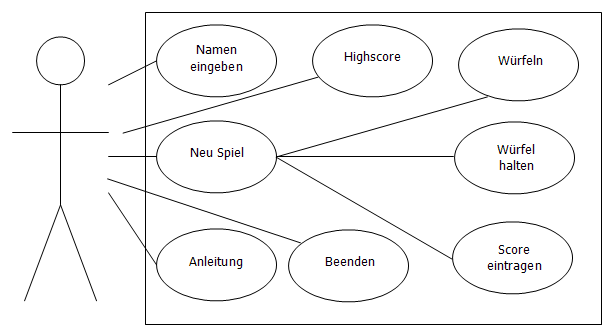
Das Spiel Würfelpoker

Fünf Würfel werden geworfen, die Würfel können dann jeweils „Neun“, „Zehn“, „Bub“, „Dame“, „König“ oder „Ass“ zeigen (Werte zählen in dieser Reihenfolge von 1 bis 6 aufwärts). Ist das Ergebnis nicht ausreichend, kann es durch nochmaliges Werfen aller oder einiger Würfel noch verbessert werden (maximal zwei zusätzliche Würfe).

Strassen sind zusammenhängende Würfelketten (Neun-Zehn-Bub-Dame-König oder Zehn-Bub-Dame-König-Ass) und zählen 20 Punkte. Full House nennt man das Ergebnis aus zwei Würfel eines Werts und drei Würfel eines anderen Werts (30 Punkte). Poker und Grande sind jeweils vier bzw. fünf gleichwertige Würfel (40 bzw. 50 Punkte).

*Analyse*

**Use Case Diagramm**

****

Das Use Case „Namen eingeben“ beschreibt die Anmeldeprozessur eines neuen Spieler, die darin besteht, dass der Spieler seinen Namen eingibt.

Fehlszenario: Spieler gibt keinen Namen ein

Das Use Case „Neu Spiel“ beschreibt den Spielablauf. Durch Drücken des „Würfel-Buttons wird gewürfelt. Nach jedem Wurf darf der Spieler 1 bis 5 Würfel halten. Dieser Vorgang wiederholt sich dreimal. Dann trägt sich der Spieler seinen Score ein.

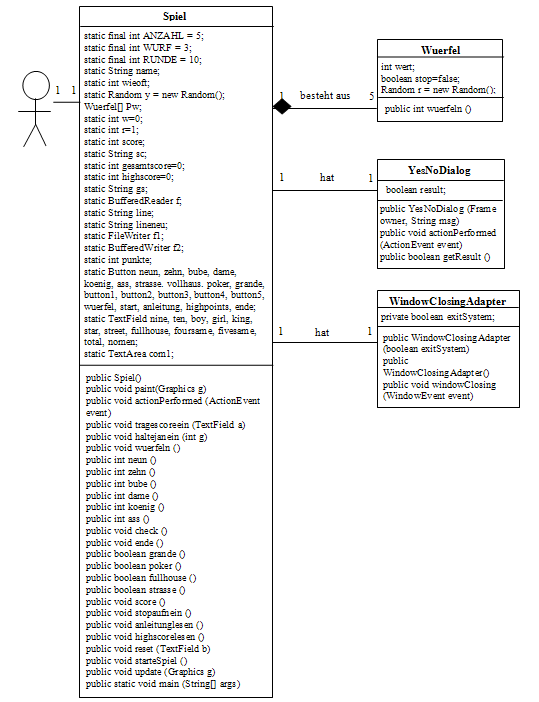
Fehlszenario: Spieler trägt sich den Score falsch ein

Das Use Case „Anleitung“ gibt Hilfe zu den Spielregeln.

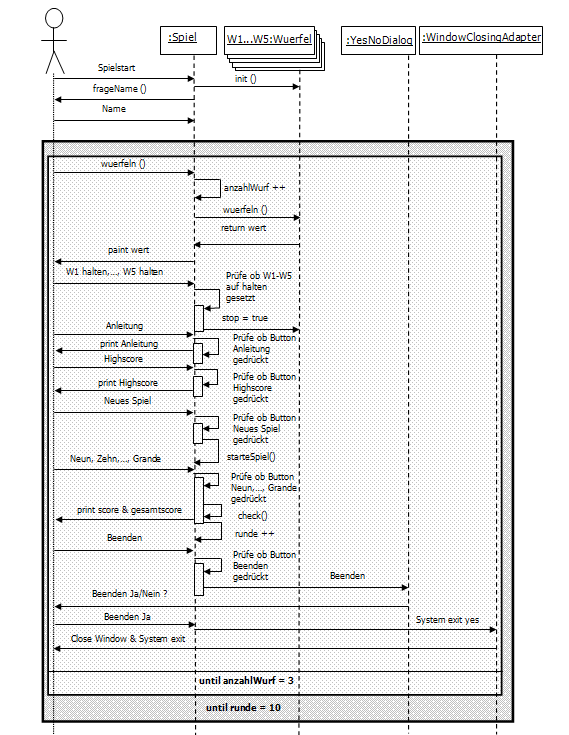
Das Use Case „Highscore“ gibt die Spielernamen mit deren Punktescores zurück, die die bisher höchsten Punkte erreicht haben.

Das Use Case „Beenden“ beendet das Spiel.

**Klassendiagramm**

****

**Sequenzdiagramm**

****

*Klassenbeschreibung*

**public class Spiel implements ActionListener extends Frame:**

Die Klasse Spiel ist Hauptklasse. Sie beinhaltet neben der main-Methode die Implementierung der benutzerfreundlichen Graphikoberfläche (GUI) in Gestalt eines Windows. Bei der Instanzierung werden das GUI mit seinen Schaltflächen und fünf Würfel erzeugt. Es können die Anleitung und der Highscore abgefragt, ein neues Spiel begonnen und das Spiel beendet werden. Das Interface ActionListener wird als Eventhandler für die Buttons und TextFields in der Methode actionPerformed() implementiert. Die Klasse ist abgeleitet von der Superklasse Frame.

Konstruktor:

public Spiel ()

               Der Konstruktor erzeugt fünf Würfel und ein GUI mit der Aufschrift „Würfelpoker”, platziert es auf dem Bildschirm, erzeugt einen WindowClosingAdapter und registriert diesen. Zusätzlich wird das Erscheinungsbild des GUI am Bildschirm definiert.

Methoden:

public void paint (Graphics g)

Diese Methode überlagert die Methode paint() der Klasse Component, um die einzelnen Würfel zu zeichnen. Dabei werden die einzelnen Graphiken implementiert.

public void actionPerformed(ActionEvent event)

               Die Ereignismethode actionPerformed() überschreibt die entsprechende Methode von ActionListener. Sie wird durch das Anklicken eines Buttons bzw. durch Eingabe des Namens in das entsprechend TextField aufgerufen.

               Wurde „Würfeln!“ angeklickt, wird eine Nachricht an die Klasse Wuerfel geschickt, die über einen Zufallsgenerator die einzelnen Werte der fünf Würfel zurückgibt.

               Wurde „Neues Spiel“ angeklickt, wird die Methode starteSpiel() aufgerufen.

               Wurde „Anleitung“ angeklickt, wird die Methode anleitunglesen() aufgerufen.

               Wurde „Highscore“ angeklickt, wird die Methode highscorelesen() aufgerufen.

               Wurde „Beenden“ angeklickt, wird die Klasse YesNoDialog instanziert und der Konstruktor dieser Klasse liefert ein Fenster, in dem entschieden werden kann, ob das Spiel wirklich beendet werden soll.

               Wurden die Buttons „W1 halten“,…, „W5 halten“ angeklickt, wird eine Nachricht an die Klasse Wuerfel geschickt, dass die einzelnen Würfel gehalten werden sollen und somit die stop-Variable der Klasse Würfel auf true gesetzt.

               Wurden die Buttons „Neun“, „Zehn“,…, „Grande“ angeklickt und sind die Texte der dazugehörigen TextFields nine, ten,…,fivesame noch  „0“ wird entweder die Methode score() aufgerufen, um die Anzahl der gleichen Würfeln abzufragen und den Score zurückzugeben, oder abgefragt ob es sich um eine spezielle Würfelzusammensetzung handelt und der Score festgelegt. Danach wird die Methode tragescoreein (TextField a) aufgerufen.

               Wird im TextField nomen der Name eingegeben, wird dieser Namen an die Variable name übergeben und der Spieler über die TextArea com1 aufgefordert zu würfeln.

public void tragescoreein (TextField a)

               Es wird die Methode score() aufgerufen und in das mit der Methode übergebene TextField der String sc, der sich aus dem errechneten Score ergibt, eingetragen. Danach wird in das TextField total der String gs geschrieben, der sich aus dem zu diesem Zeitpunkt errechneten Gesamtscore ergibt.

public void haltejanein (int g)

               Hier wird beim Aufrufen der Methode der durch das Übergabe der Ziffer entsprechende Würfel die stop-Variable des Würfel auf true gesetzt, wenn diese vorher false war, oder auf false gesetzt, wenn diese vorher true war. Danach werden durch aufrufen der repaint()-Methode, die Würfel neu gezeichnet, um darzustellen, ob ein Würfel gehalten (mit farbigen Rand) oder nicht gehalten (ohne Rand) wird.

public void wuerfeln ()

               In dieser Methode werden alle fünf Würfel durch senden einer Nachricht an die Klasse Würfel sooft gewürfelt, wie es die Randomvariable wieoft vorgibt. Die Methode wuerfeln() der Klasse Wuerfel liefert danach die Werte der einzelnen Würfel zurück.

public int neun ()

               Hier wird die Anzahl der gewürfelten „Neun” ermittelt und diese zurückgegeben.

public int zehn ()

               Hier wird die Anzahl der gewürfelten „Zehn” ermittelt und diese zurückgegeben.

public int bube ()

               Hier wird die Anzahl der gewürfelten „Bube” ermittelt und diese zurückgegeben.

public int dame ()

               Hier wird die Anzahl der gewürfelten „Dame” ermittelt und diese zurückgegeben.

public int koenig ()

               Hier wird die Anzahl der gewürfelten „König” ermittelt und diese zurückgegeben.

public int ass ()

               Hier wird die Anzahl der gewürfelten „Ass” ermittelt und diese zurückgegeben.

public void check ()

               In dieser Methode wird unter der Voraussetzung, dass die dazugehörigen TextFields noch „0” anzeigen, überprüft, ob eine „Grande“, ein „Poker“, ein „Full House“ oder eine „Strasse“ gewürfelt wurde. Ist dies der Fall, wird der entsprechende Text über die TextArea com1 angezeigt.

public void ende ()

               Wenn in den Score-TextFields andere Strings als „0“ eingetragen sind und insgesamt 10 Runden gespielt wurde, wird dem Spieler über die TextArea com1 nochmals der Gesamtscore angezeigt und ihm über Anzeige von „Rekord!“ mitgeteilt, dass er die bisher höchste Punkteanzahl erreicht.

public boolean grande ()

               Hier werden die einzelnen Würfelwerte überprüft und true zurückgegeben, wenn die Würfelzusammensetzung einer „Grande“ entspricht.

public boolean poker ()

               Hier werden die einzelnen Würfelwerte überprüft und true zurückgegeben, wenn die Würfelzusammensetzung einem „Poker“ entspricht.

public boolean fullhouse ()

               Hier werden die einzelnen Würfelwerte überprüft und true zurückgegeben, wenn die Würfelzusammensetzung einem „Full House“ entspricht.

public boolean strasse ()

               Hier werden die einzelnen Würfelwerte überprüft und true zurückgegeben, wenn die Würfelzusammensetzung einer „Strasse“ entspricht.

public void score ()

               Hier wird der Gesamtscore errechnet, die Variable w, die der Anzahl der Würfe entspricht, auf null gesetzt und die Rundenvariable r um eins erhöht. Durch Aufrufen der Methode stopaufnein() werden alle stop-Variablen der einzelnen Würfel auf false gesetzt. Der Score und der Gesamtscore werden in einen String umgewandelt. Entspricht der Score dem Wert null wird dies durch den String „---„ dargestellt, um zu visualisieren, dass in dieser Zeile nichts mehr eingetragen werden kann. Danach werden durch den Aufruf der repaint()-Methode die Würfel neu gezeichnet (ohne Rand!).

public void stopaufnein ()

               Hier werden alle stop-Variablen der einzelnen Würfel auf false gesetzt.

public void anleitunglesen ()

               Durch Aufruf dieser Methode wird die im File anleitung.txt hinterlegte Anleitung zeilenweise ausgelesen und in der TextArea com1 ausgegeben.

public void highscorelesen ()

               Durch Aufruf dieser Methode wird der im File highscore.txt hinterlegte Highscore zeilenweise ausgelesen und in der TextArea com1 ausgegeben.

public void reset (TextField b)

               Hier wird der Text des übergebenden TextField auf den String „0“ gesetzt.

public void starteSpiel ()

               Beim Aufruf dieser Methode wird die reset (TextField b)-Methode für alle TextFields aufgerufen, der Spieler aufgefordert im neuen Spiel zu würfeln, die Variable gesamtscore auf null gesetzt und die Methode stopaufnein() aufgerufen. Danach wird die Methode repaint() aufgerufen.

public void update (Graphics g)

               Diese Methode überlagert die entsprechende Methode der Klasse Component damit, um Bildflackern zu verringern. Es wird nur die nur die paint (Graphics g)-Methode aufgerufen und nicht wie bei der überlagerten Methode auch noch einzelne Hintergründe des Frames neu gezeichnet.

public static void main (String[] args)

               Main-Methode, die den Konstruktor Spiel () aufruft.

**public class Wuerfel:**

Die Klasse Wuerfel repräsentiert einen einzelnen Würfel, der die Werte 1-6 annehmen kann und diesen Wert „halten“ kann, d.h. beim nächsten mal würfeln nicht mitgewürfelt wird.

Methode:

public int wuerfeln ()

               Hier wird der Wert wer des Würfels  anhand des Zufallsgenerators r, der die Werte eins bis sechs ausgeben kann, ermittelt.

**public class WindowClosingAdapter extends WindowAdapter:**

Diese Klasse ist eine Subklasse von WindowAdapter.

Methoden:

public WindowClosingAdapter (boolean exitSystem)

               Der Konstruktor erzeugt ein WindowClosingAdapter-Objekt zum Schließen des Fensters. Ist exitSystem true, wird das komplette Programm beendet.

public WindowClosingAdapter()

               Erzeugt einen WindowClosingAdapter zum Schließen des Fensters. Das Programm wird nicht beendet.

public void windowClosing (WindowEvent event)

               Diese Methode überlagert die Methode „windowClosing“ der Klasse WindowAdapter. Sie wird aufgerufen, wenn der Anwender das Fenster über die Titelseite oder die Tastenkombination [ALT]+[F4] schließen will.

**class YesNoDialog extends Dialog implements ActionListener:**

Diese Klasse ist eine Subklasse von Dialog und implementiert das Interface ActionListener.

Konstruktor:

public YesNoDialog (Frame owner, String msg)

               Der Konstruktor erzeugt einen Frame, aus dem per Buttonklick ein modaler Dialog aufgerufen werden kann. Der Dialog fragt den Spieler, ob das Spiel beendet werden soll, und wartet, bis einer der Buttons „Ja“ oder „Nein“ gedrückt wurde.

Methoden:

public void actionPerformed (ActionEvent event)

               Wird der Button „Ja“ gedrückt, wird das Spiel beendet, andernfalls läuft es weiter.

public boolean getResult ()

               Die Methode liefert nach drücken der Buttons „Ja“ oder „Nein“ einen booleschen Rückgabewert, der nach schließen des Dialogs zurückgegeben wird.

*Beschreibung für den Anwender*

Das Spiel Würfelpoker wird in der Klasse Spiel gestartet. Als erstes muss man den Namen eingeben. Danach wird man aufgefordert zu würfeln. Es werden zehn Runden und pro Runde maximal dreimal gewürfelt. Nach jedem Wurf können die Würfel gehalten werden. Diese Würfel werden beim nächsten Wurf nicht mitgewürfelt. Es kann zu jedem Zeitpunkt der Score in einer der Zeilen (Neun, Zehn, …, Grande) eingetragen werden. Des Weiteren kann jederzeit die Anleitung oder der Highscore abgefragt werden. Zusätzlich kann ein neues Spiel begonnen oder das Spiel  beendet werden. Nach Abschluss der 10 Runden erfährt man, ob man einen Highscore erreicht hat.

*Wartung*

Grundsätzlich ist keine Wartung des Programms vorgesehen. Treten jedoch unerwartet Fehler auf, kann man mir gerne eine e-Mail schicken, in der eine genaue Fehlerbeschreibung des aufgetreten Problems mitgeschickt wird. Ich werde dann versuchen, den Fehler in einer neueren Version zu beheben und dem Sender die überarbeitete Version zurückzuschicken.

**Quellcode:** [**Würfelpoker**](http://michael.hahsler.net/JAVA_praktikum/projekte/WS04/schlichtinger/quellcode/)