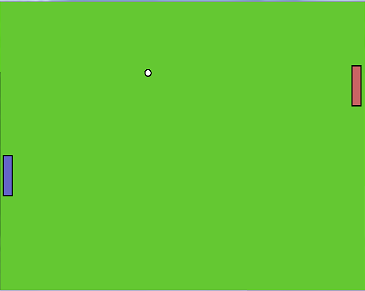


**Belegarbeit Software Engineering**

Programmcode: ‘Pong‘

Abgabetermin: 28.06.2015

****

Von Hokar Mohammed und Dimitri Kraft, 3.Fachsemester Computer Engineering

**1.0 Einführung**

In dem folgenden Projekt haben wir uns mit dem Spiel „Pong“ befasst. Hierbei handelt es sich um ein Spiel, welches in der Sprache C++ geschrieben ist. Das Spiel hat einen einfachen Aufbau und ist leicht verständlich. Das Ziel dieses Spieles ist es, den eigenen Balken (im weiteren „paddle“ genannt) mittels der Pfeiltasten hoch bzw. runter zu bewegen, sodass der Ball, welcher immer in Bewegung ist, an diesem paddle abprallt und somit wieder in Richtung des Gegners gelenkt wird. Verfehlt der Ball nun ein paddle und gelangt an den linken oder rechten Rand des Spielfensters, so verliert der jeweilige Spiele, bei dem dies eingetroffen ist.

**2.0 Ziele:**

Das Spiel hatte uns sehr viele Möglichkeiten geboten Modifikationen durchzuführen und am Programmcode zu basteln. Wir haben uns intensiv mit dem Programm befasst und sind zu dem Entschluss gekommen, uns mehrere Ziele zu setzen. Die Änderung beinhaltete die Erweiterung um einen neuen Spielmodus, eines zusätzlichen Balles, die Möglichkeit der Veränderung der Paddle und Ballgeschwindigkeit und abschließend die Festlegung der Spielfeldgröße.

**3.0 Umsetzung**

Unsere gesetzten Ziele haben wir mit der Implementierung einer Konsolenabfrage realisiert. Der Ablauf ist im Folgenden erklärt:

**3.1 Spielmodus**

Zu Anfang wird der Spielmodus gewählt. Hierbei kann der Anwender entscheiden, ob er alleine gegen den Computer oder zusammen mit einem anderen Anwender am selben PC spielen möchte.

Dies wird durch eine einfache Abfrage realisiert und anschließend ein Wert 0 oder 1 übergeben.

Durch diese Einstellung wird überprüft, ob das rechte Paddle von dem Computer oder von dem Benutzer gesteuert wird. Dies wird im Code durch eine if Bedingung realisiert.

**3.2 Anzahl der Bälle**

Diese Option ist nur verfügbar, wenn der Mehrspielermodus ausgewählt wurde.  
Hier kann man entscheiden, ob man mit einem oder zwei Bällen spielen möchte.

Durch eine Abfrage wird ein boolescher Wert auf true oder false gesetzt.

Je nach Einstellung wird im Code via Bedingung festgelegt, ob ein zweiter Ball im Spiel erscheinen soll oder nicht.

**3.3 Geschwindigkeit Paddle und Ball**

Des Weiteren ist es möglich die Geschwindigkeit des Paddles und des Balls zu bestimmen.

Durch eine Abfrage kann ein bestimmter floating- Wert auf 400 (langsam) oder 800 (schnell)

gesetzt werden.

Dies wird im Code mit einer if Bedingung realisiert.

**3.4 Spielfeldbreite**

Die Spielfeldbreite wird mit dem Eingeben einer bestimmten Zahl realisiert.

Hier wird dann einem Integer (int) ein bestimmter Wert zugeteilt.

Dieser ist verantwortlich für das Erzeugen des Spielfensters.

**3.5 Spielfeldhöhe**

Die Spielfeldhöhe wird mit dem Eingeben einer bestimmten Zahl realisiert.

Hier wird dann einem Integer (int) ein bestimmter Wert zugeteilt.

Dieser ist verantwortlich für das Erzeugen des Spielfensters.

**3.6 Bestätigung der Einstellungen**

Abschließend kann via ja oder nein Abfrage entschieden werden, ob die Einstellungen korrekt oder ggf. falsch sind.

Dann kann besteht die Möglichkeit einer Neukonfiguration oder die Übernahme der Einstellungen ins Spiel.

1. **Implementierung**

**4.1 Implementierung des Mehrspielmodus**

Die Implementierung des Mehrspielermodus erfolgt durch eine Abfrage in der Konsole und anschließender If-Bedingung im Code selbst.

Im Code wurde eine zweite Steuerung für das rechte Paddle analog zum linken Paddle erstellt und wird nur aktiviert, wenn der Wert für „singleplayer“ auf 0 gesetzt wird.

Durch die Bibliothek ist eine einfache Konfiguration der Steuerung möglich und mit wenigen Handgriffen realisierbar.

**4.2 Implementierung der Geschwindigkeit für Paddle und Ball**

Dies erfolgt durch einfache Wertzuweisung nach Konsolenabfrage im Code.

Keine weiteren relevanten Änderungen im Code notwendig.

**4.3 Implementierung 2 Bälle Modus**

Durch die Konsolenabfrage wird ein boolescher Wert auf true oder false gesetzt.

Die Implementierung erfolgt durch das Erstellen einer zweiten Balls-shape Objekts und anschließender Konfiguration von Eintrittswinkel und Startposition des Balls.

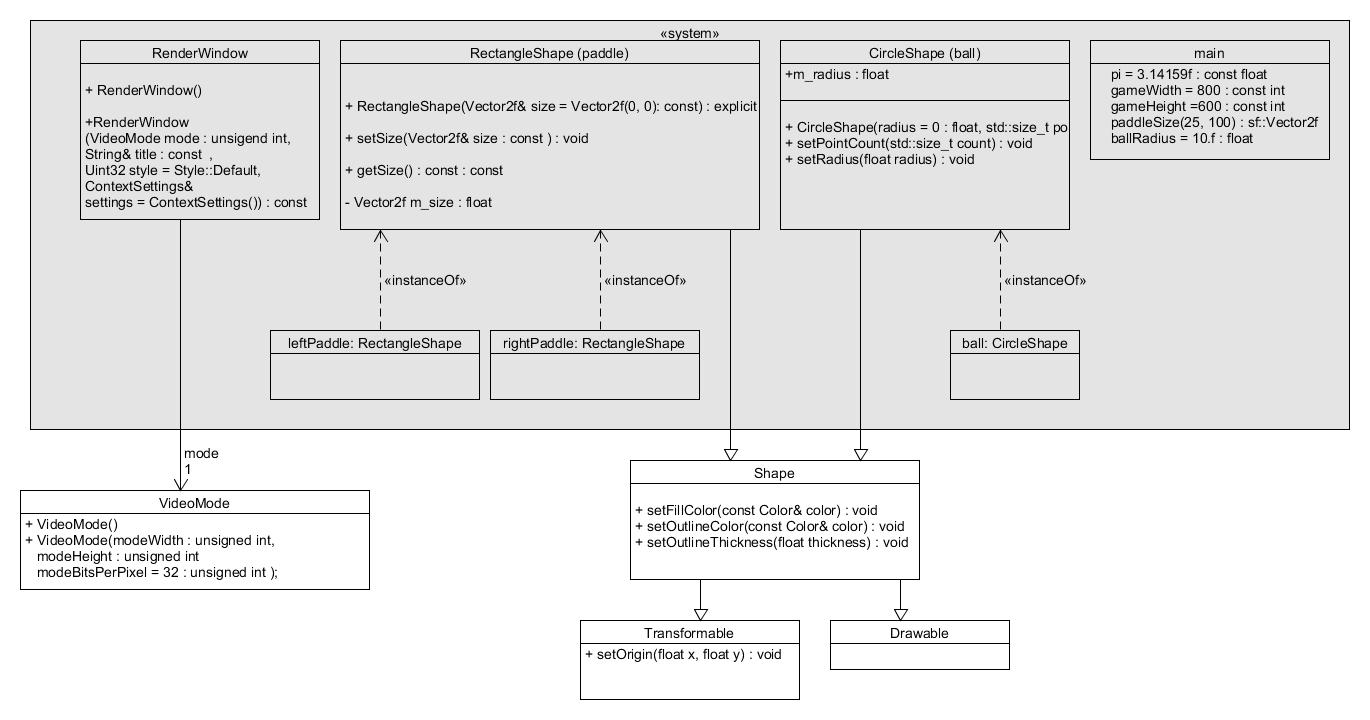
Die Bewegung erfolgt analog zu Ball 1. Hierfür wurde die Bewegung des ersten Balls kopiert und nun das Objekt ball durch ball2 ersetzt.

Die Kollisionsabfrage ist analog realisiert worden, hier wurde wieder nur ball mit ball2 ersetzt und ein zweiter Winkel für ball2 erstellt.

**4.4 Implementierung Spielfeldgröße und höhe**

Dies erfolgt wieder durch eine Wertezuweisung nach Konsolenabfrage.

Hier werden dann zwei Werte der RenderWindow Funktion übergeben und sind daher wesentlicher Bestandteil des Codes, da sie für Kollisionsabfrage, Paddle und Ballbewegung erforderlich sind.

1. **Klassendiagramm**

Das Klassenmodell des Programms „Pong“ beinhaltet die Superklassen RenderWindow, RectangleShape und Circle Shape. Hierbei handelt es sich um die Klassen, welche für die grafische Realisierung des Spiels notwendig sind. Nun gibt es noch eine kleinere Unterklassen, welche von den Superklassen erben.

**RenderWindow**

* Erzeugung des Spielsfensters

**Mit Realsierung des Ziels:**

* Erzeugung des Spielfensters mit vorhergehend eingestellten Parametern durch den Anwender

**RectangleShape**

* Erzeugung der Paddles

**CircleShape**

* Erzeugung des Balles

**Mit Realsierung des Ziels:**

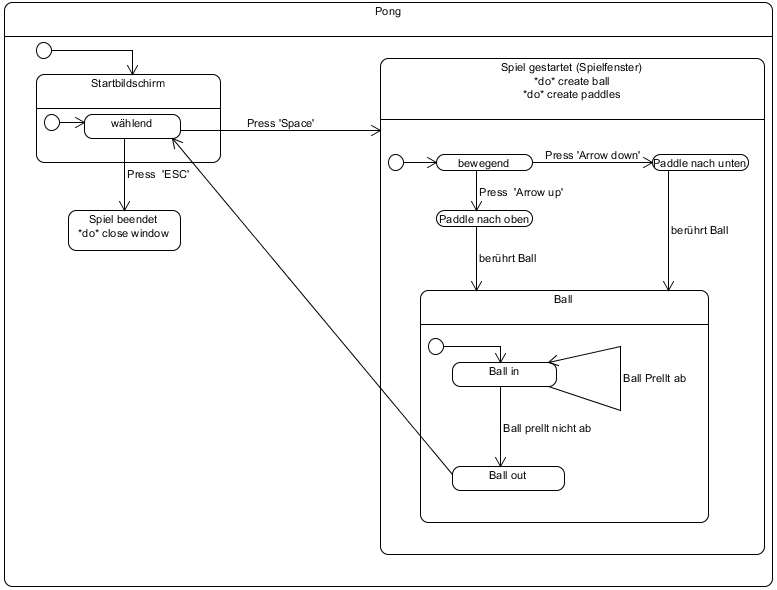
* Erzeugung eines zweiten Balles

**Main**

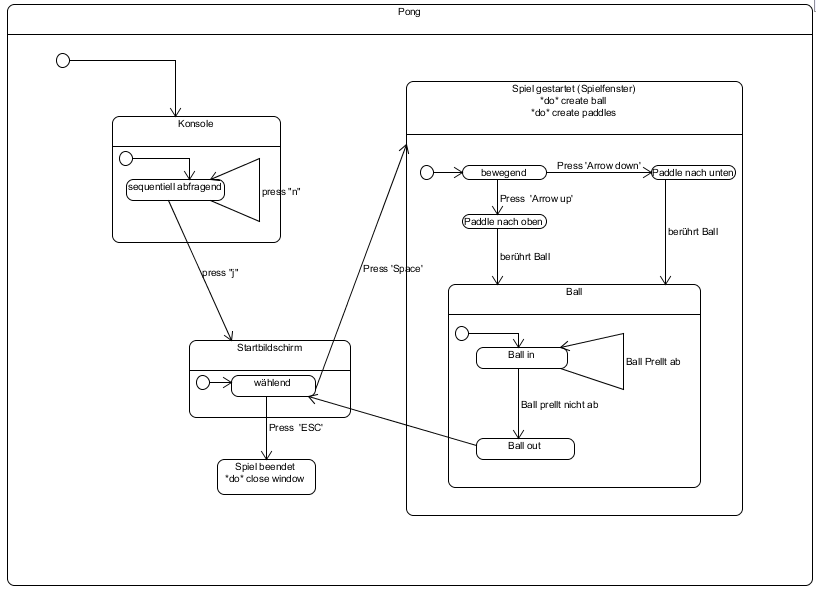
* Initialisierung der Konstanten für Ballradius, Größe des Spielfensters und Größe der Paddles

**Mit Realsierung des Ziels:**

* Initialisierung der Parameter für Ballradius, Größe des Spielfensters und Größe der Paddles
* Konsolenabfrage mit Einstellung der Parameter

1. **Zustandsdiagramm**

* Sobald der Code ausgeführt wird, erscheint ein Startfenster
* Der Anwender befindet sich im Zustand „wählend“
* Nun hat er die Möglichkeit entweder Space oder ESC auf der Tastatur zu drücken. Hierbei handelt es sich um eingehende Transitionen
* Space gedrückt: Spiel wird gestartet
* ESC gedrückt: Spiel wird beendet
* Sobald das Spiel gestartet wird, befindet sich der Anwender im Zustand „bewegend“
* Möglichkeit über die Transitionen „arrow up“ und „arow down“ in zwei verschiedene Zielzustände zu gelangen
* „arrow up“ gedrückt: Paddle nach oben
* „arrow down“: Paddle nach unten
* Sobald der Ball die Paddles berührt gelangt er in den Zustand „ball in“
* Das Spiel läuft somit weiter
* Sobald der Ball die Paddle nicht berührt und nicht abprallt wird das Spiel beendet und das Programm gelangt wieder in den Zustand „ wählend“

****

**Mit Realisierung des Ziels:**

* Beim Programmstart öffnet sich ein Konsolenfenster
* Zustand „ sequentiell abfragend“
* Über die Transition „press j“ gelangt das Programm in den Zustand Startfenster
* Über die Transition „press n“ gelangt das Programm in keinen anderen Zustand, sondern wieder in den Zustand „sequentiell wählend“