Projektdokumentation

Frogger Reincarnation



|  |  |
| --- | --- |
| Autor | Kevin Sapper |
| Matrikel-Nr. | 470407 |
| Studiengang | Angewandte Informatik (B.Sc.) |
| Modul | Objektorientierte Softwareentwicklung (1110) |
| Datum | 23.01.12 |

Inhalt

[1 Spielidee 3](#_Toc315376097)

[2 Kurzanleitung 3](#_Toc315376098)

[3 Entwurf 4](#_Toc315376099)

[3.1 Bemerkung 4](#_Toc315376100)

[3.2 Main-Frame und Cards 4](#_Toc315376101)

[3.3 Gameobjects (Spielobjekte) 5](#_Toc315376102)

[3.4 Managers 7](#_Toc315376103)

[4 Quellen 8](#_Toc315376104)

# Spielidee

Die Grundidee ist, eine Spielfigur von einem Startpunkt zu einem oder mehreren Zielpunkten zu bewegen. Der Spieler benutzt hierzu die Pfeiltasten der Tastatur, um sich nach oben, links, rechts und unten zu bewegen. Die Spielfigur wird durch einen Frosch dargestellt.   
Um es dem Spieler zu erschweren, vom Start- zum Zielpunkt zu gelangen, gibt es verschiedene Hindernisse. Diese Hindernisse können in zwei Kategorien unterteilt werden. Erstens *Straßenobjekte*, wie Autos, Motorräder oder LKWs, töten die Spielerfigur bei Berührung. Zweitens *Wasserobjekte*, wie Baumstämme oder Schildkröten, transportieren die Spielerfigur in ihrer Bewegungsrichtung. Es gibt zwei verschiedene Spielmodifikationen.   
Die *Time-Mode* gibt eine bestimmte Zeit vor, in der möglichst viele Spielerfiguren vom Start- zum Zielpunkt gebracht werden müssen. Für jede Spielerfigur im Ziel wird die Punktzahl um eins erhöht. Das Spiel ist um, wenn die Zeit abgelaufen ist oder der Spieler keine Leben mehr hat. Die *Survival-Mode* hat mehrere Zielpunkte. Der Spieler muss zu jedem Zielpunkt eine Spielfigur bringen. Dazu hat er nur eine bestimmte Zeit. Ist diese abgelaufen verliert er ein Leben. Befinden sich an allen Zielpunkten Spielfiguren wird das Level um eins erhöht und das Szenario beginnt von vorne. Hat Spieler eine Spielfigur zum Zielpunkt geführt hat bekommt er in Abhängigkeit von der benötigten Zeit und des Levels Punkte. Wird der Spieler, durch Tot oder Erreichen des Ziels, an den Startpunkt zurückgesetzt startet auch die Zeit von vorne.

# Kurzanleitung

Gesteuert wird die Spielfigurfrogger_down_1.png durch drücken der Pfeiltasten. Mit Pfeiltasten hoch und runter, wird die Spielebene gewechselt. Mit den Pfeiltasten links und rechts können jeweils kleinere Schritte gemacht werden. Wird die Spielfigur auf dem Weg zum Ziel von einem Auto car_green_right.png überrollt verliert man ein Leben und wird an den Startpunkt zurückgesetzt.



Zielbereich

Startbereich

Ebene

Punktestand

Lebensanzeige

Restliche Zeit

Spieler

Autos

# Entwurf

## Bemerkung

Aufgrund des Umfangs des kompletten UML-Diagramms finden sich im Folgenden nur die wichtigsten Ausschnitte!

## Main + Cards.pngMain-Frame und Cards

Die Main Klasse leitet von JFrame ab und ist damit der Einstiegspunkt für das Programm. Zur Darstellung der Inhalte gibt es vier verschiedene JPanels (OpeningCard, StagesCard, GameCard und HighscoreCard). Die Panels werden durch das CardLayout der Main-ContentPane verwaltet und können wie Karten in einem Deck aufgedeckt werden. Alle Panels benutzen eine JLayeredPane, um ihre Inhalte auf mehreren Ebenen darzustellen. Die Helferklasse CardLayer bittet hierzu eine Auswahl an Ebenen an.

## Gameobjekts.emfGameobjects (Spielobjekte)

Ausgangspunkt für die Spielobjekte ist die Klasse GameObject, welche das Figure Interface aus der Aufgabe G implementiert. Die Klasse ist zudem abstract und dient daher nur als Vorlage für alle anderen Spielobjekte. Um neue Spielobjekte zu erzeugen wird die statische Klasse GameObjectFactory benutzt. Die Verwaltung aller Objekte übernimmt der GameObjectContainer. Dieser benutzt mehrere FigureContainer, um die Spielobjekte in Rahmen, mobile Objekte und immobile Objekte zu gliedern.

## Manager.emfManagers

Zur Spielsteuerung werden verschiedene Manager verwendet. Der Hauptmanager ist der GameManager. Er hat die Funktion das Spiel zu starten, je nach GameMode Leben abzuziehen bzw. den Level zu erhöhen und demnach auch das Spiel im geeigneten Moment zu beenden. Außerdem fällt ihm die Funktion zu, die Spielereingaben der Pfeiltasten zu interpretieren. Der HighscoreManager lädt beim Starten die existierenden Highscore-List, erzeugt und erhöht eine neue Highscore, wenn er vom GameManager dazu aufgefordert wird und speichert nach Spielende die erzielten Punkte ab. Der TimeManager verwaltet die Spielzeit und teilt dem GameManager mit wenn diese abgelaufen ist. Das Herzstück des Spiels ist der StageManager, er liest aus der aktuellen Stage die StageRow‘s aus und erzeugt auf Grundlage dessen das aktuelle Spielfeld und die Figuren die sich darauf bewegen.

# Quellen

Bild Pfeiltasten [<http://photo-dictionary.com/phrase/8534/arrow-keys.html>]

background\_opening.png [<http://fav.me/dmisml>]

background\_highscore.png [<http://fav.me/d4mnzrm>]

background\_stages.png [<http://fav.me/d3evta1>]

stage1.png [<http://fav.me/d3o6akv>]

stage2.png [<http://fav.me/d1rddjj>]

water\_1.png p[<http://www.cgtextures.com/texview.php?id=23842&PHPSESSID=nmgn8dj3cpprplnqee5a6fbvo4>]

grass.png [<http://www.cgtextures.com/texview.php?id=41195&PHPSESSID=nmgn8dj3cpprplnqee5a6fbvo4>]

highway\_1.png [<http://www.cgtextures.com/texview.php?id=53107&PHPSESSID=nmgn8dj3cpprplnqee5a6fbvo4>]

frogger\_<up, down, left, right>.png [<http://ninjanojutsu.com/index.php/sprites/show/50>]

car\_green\_<left, right>.png [<http://img.photobucket.com/albums/v171/titanomega/cargrnch.png>]