Modul 226

Wizard

Ein Bild, das Baum, Natur, draußen, Nachthimmel enthält.

Automatisch generierte Beschreibung

**Produzent: Lukas Wider Lehrperson: Sven Schirmer**

Inhaltsverzeichnis

[Game Idee 3](#_Toc93955067)

[Grund für die Wahl 3](#_Toc93955068)

[Gangster Squad Origins 3](#_Toc93955069)

[Umsetzung 3](#_Toc93955070)

[Dankeschön 3](#_Toc93955071)

[Projekt Aufbau 4](#_Toc93955072)

[Controller 4](#_Toc93955073)

[Model 4](#_Toc93955074)

[View 4](#_Toc93955075)

[Ressources 4](#_Toc93955076)

[Vererbung 5](#_Toc93955077)

[Elternklasse: 5](#_Toc93955078)

[Kindklasse 6](#_Toc93955079)

[Klassendiagramm 7](#_Toc93955080)

[Tagesjournal 8](#_Toc93955081)

[Fazit 12](#_Toc93955082)

[Gelernt 12](#_Toc93955083)

[Schwierigkeiten 12](#_Toc93955084)

[Hilfe/ Quellen 13](#_Toc93955085)

[Eidesstattliche Erklärung 14](#_Toc93955086)

[Bewertungsraster 15](#_Toc93955087)

# Game Idee

## Grund für die Wahl

Ich habe früher sehr viele Handygames gespielt. Hab viel rumprobiert und auch immer die Spiele, welche in Werbungen abgespielt wurden, herunter geladen und gespielt. Gangster Squad Origin war eines davon und es hat mir Spass gemacht. Vom Spielprinzip und Aufbau fand ich, dass es sehr einfach ist und habe es deswegen für das Projekt im Modul 226 gewählt.

Screenshot Gangster Squad Paradise

1 Screenshot aus Gangster Squad Paradise

## Gangster Squad Origins

Gegner kommen von der linken oder rechten Seite und greifen die Person in der Mitte an. Der Spieler muss nun nach rechts oder links drücken um diese Gegner anzugreifen. Dazu gibt es verschiedene Arten von Gegnern mit gewissen Vorteilen in verschiedenen Statistiken. Zum Beispiel mit mehr Leben, Schaden oder Geschwindigkeit mit der sie zum Spieler kommen.

## Umsetzung

In meinem Spiel werden die Gegner wie in Gangster Squad Origins von rechts und links kommen. Meine Spielfigur ist ein Magier welcher Feuerbälle verschiesst, welche im Falle sie einen Gegner treffen diesen sofort töten.

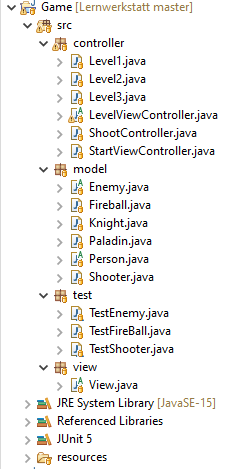
Es gibt 2 Gegner, welche in 3 Levels aufgeteilt werden. Im ersten Level kommt der Ritter welcher eine normale Geschwindigkeit hat. Im 2. Level kommt ein Paladin welcher ein Bisschen schneller ist und im dritten kommen beide Gegnertypen aber jeder einzelne Gegner hat unterschiedlichen Geschwindigkeiten.

Im Start Screen kann man diese 3 Levels auswählen (Einfach, Mittel und Schwer)

## Dankeschön

Ich danke Eric Schneider, Wendelin Haller und Sasha Kokot, welche mir zum Schluss hin, als ich riesige Probleme hatte, sehr viel geholfen haben. Ein Spezielles Dankeschön geht an Wendelin Haller welcher mich sehr viel unterstützt was man auch an dem Projektaufbau und der Struktur erkennt.

# Projekt Aufbau



2 Projektstruktur Screenshot aus Eclipse

## Controller

Die Darstellung des Spiels wir im Controller gemacht. Im ShootController wird alles ausgeführt, er dient als Main-Controller.

## Model

Im Package "model" werden alle Figuren, wie der Feuerball, der Shooter als Magier und die Gegner als Paladin und Knight/Ritter erstellt. Die Person dient als Elternklasse für alle Figuren und von Enemy. Enemy ist die Elternklasse von Paladin und Knight.

## View

View.java kümmert sich um alle Darstellungen des Controllers.

3 Ordnerstruktur vom Projekt

## Ressources

In den Ressourcen liegen alle Bilder zur Darstellung von den Figuren und vom Hintergrund

Ein Bild, das draußen, Baum, Himmel, Schnee enthält.

Automatisch generierte BeschreibungEin Bild, das Text enthält.

Automatisch generierte Beschreibung

4 resources Gameboard

5 resources Knight

Ein Bild, das Text, Verkehrsampel, ClipArt enthält.

Automatisch generierte Beschreibung

6 resources Feuerball

Ein Bild, das Spielzeug enthält.

Automatisch generierte Beschreibung

7 Wizard

8 resources Paladin

# Vererbung

## Elternklasse:

### **Ein Bild, das Text enthält. Automatisch generierte BeschreibungKlassen Definition**

**9** CodeSnippet Klasse Person

Diese Klasse setzen wir auf **abstract**, damit man keine Objekte des Typen "Person" erstellen kann. Eine andere Möglichkeit wäre es die Klasse auf **final** zu setzen, damit man die Vererbung stoppt.

### **Attribute**

In der Klasse Person machen wir wie gewohnt den Normalen Aufbau einer Klasse. Dazu geben wir zuerst die Attribute an.

### **Konstruktor**

Auch der Konstruktor ist gleich wie beim Erstellen einer normalen Klasse.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

10 CodeSnipper Klasse Person

### **Methoden/ Funktionen**

Die Methoden werden wie in normalen Klassen erstellt.

### **Grund für Vererbung**

Credo: **DRY**

**D**ONT 🡪 Copy Paste ist böse

**R**EPEAT 🡪 Wartbarkeit verbessern

**Y**OURSELF 🡪 Code- Qualität verbessern

## Kindklasse

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

11 CodeSnippet Klasse Shooter

### **Klassendefinition**

Die Klasse wird normal definiert mit der Ausnahme **extends Elternklassenname.**

### **Attribute**

Sofern keine neue Attribute dazu kommen, schreibt keine Attribute.

### **Konstruktor**

Der Konstruktor wird mit den **gleichen Attributen der Elternklasse.** Der Konstruktor wird mir **super(Attributname …)** abgeschlossen.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

12 CodeSnippet Klasse Shooter

### **Methoden/ Funktionen**

Falls eine Methode welche in der Elternklasse definiert wurde, nicht verändert wird muss man diese in der Kindklasse nicht implementieren. Falls jedoch eine Methode in der Kindklasse ,verändert , gebraucht wird muss man diese implementieren.

# Klassendiagramm

# Tagesjournal

**22.11.2021**

|  |  |
| --- | --- |
| Zeit (geplant) | 3.5h |
| Zeit (tatsächlich) | 3.5h |
| Geplante Ziele | Aufgabe 1-3 GitHub einführung |
| Erreichte Ziele | Aufgabe 1-3 GitHub einführung |
| Ausgeführte Arbeiten | Aufgabe 1  Aufgabe 2  Aufgabe 3 |
| Probleme | Aufgabestellung am Anfang nicht verstanden |
| Hilfestellung | Klassenkameraden |

**29.11.2021**

|  |  |
| --- | --- |
| Zeit (geplant) | 3.5h |
| Zeit (tatsächlich) | 3.5h |
| Geplante Ziele | GitHub einrichten, Repositoris erstellen, Game Idee definieren |
| Erreichte Ziele | GitHub einrichten, Repositoris erstellen, Game Idee definieren |
| Ausgeführte Arbeiten | GitHub Account erstellt,  Game Idee definiert |
| Probleme | Ausversehen den falschen Pull-Befehl eingegeben |
| Hilfestellung | Git --help |

**06.12.2021**

|  |  |
| --- | --- |
| Zeit (geplant) | 3.5h |
| Zeit (tatsächlich) | 3.5h |
| Geplante Ziele | Repos vom Projekt erstellen. Genauere Attribute definieren.  Grundstruktur des Projekts erstellen. |
| Erreichte Ziele | Repos erstellt.  Grösstenteils die Attribute definiert.  Klassendiagramm grösstenteils fertig (neue Attribute und Methoden möglich) |
| Ausgeführte Arbeiten | Repos erstellt, Attribute definiert, Klassendiagramm gemacht |
| Probleme | Attribute und Methoden sind noch nicht ganz klar, wird sich im Verlauf des Programmierens wahrscheinlich aufklären |
| Hilfestellung |  |

**13.12.2021**

|  |  |
| --- | --- |
| Zeit (geplant) | 3.5h |
| Zeit (tatsächlich) | 3.5h |
| Geplante Ziele | Zeitplan erstellen, Klassen mit Attributen erstellen |
| Erreichte Ziele | Zeitplan erstellen, Klassen mit Attributen erstellen |
| Ausgeführte Arbeiten | Zeitplan erstellen, Klassen mit Attributen erstellen, Dokumentationen aktualisiert |
| Probleme | Eclipse externe Library einfügen(vergessen wie) |
| Hilfestellung | Internet |

**03.01.2022**

|  |  |
| --- | --- |
| Zeit (geplant) | 3.5h |
| Zeit (tatsächlich) | 3.5h |
| Geplante Ziele | Spielfenster erstellen/ Layout |
| Erreichte Ziele | Spielfenster erstellt/ Layout |
| Ausgeführte Arbeiten | Gamefenster erstellt |
| Probleme | Man kann das Spiel nicht ausführen weil die Klasse Core.jar nicht gefunden wird obwohl diese in der referenced librarys ist. |
| Hilfestellung | Internet |

**04.01.2022**

|  |  |
| --- | --- |
| Zeit (geplant) | 3.5h |
| Zeit (tatsächlich) | 3.5h |
| Geplante Ziele | Alle Klassen mit Konstruktor erstellen |
| Erreichte Ziele | Alle Klassen mit Konstruktor erstellen |
| Ausgeführte Arbeiten | Klassen mit den Konstrukroten erstellt |
| Probleme | Konstruktor Fehler |
| Hilfestellung | Adis <3 |

**09.01.2022**

|  |  |
| --- | --- |
| Zeit (geplant) | 2h |
| Zeit (tatsächlich) | 2h |
| Geplante Ziele | Draw Funktion aller Spielfiguren erstellen |
| Erreichte Ziele | Draw Funktion |
| Ausgeführte Arbeiten | Spielerfiguren können gezeichnet werden |
| Probleme |  |
| Hilfestellung |  |

**15.01.2022**

|  |  |
| --- | --- |
| Zeit (geplant) | 2h |
| Zeit (tatsächlich) | 2h |
| Geplante Ziele | Funktionen |
| Erreichte Ziele | nichts |
| Ausgeführte Arbeiten | Rumprobiert |
| Probleme | Funktionerte nichts |
| Hilfestellung | Internet |

**17.01.2022**

|  |  |
| --- | --- |
| Zeit (geplant) | 4h |
| Zeit (tatsächlich) | 4h |
| Geplante Ziele | Klasse: Shooter Funktion: draw  Klasse: Shooter Funktion: move  Klasse: Shooter Funktion: shoot  Fehler beheben: (Klasse nicht gefunden) |
| Erreichte Ziele | Fehler behoben(Core.jar wurde falsch eingefügt)  Funktion draw und move erstellt und es funktioniert |
| Ausgeführte Arbeiten | Fehler behoben, Funktionen erstellt, Hintergrundbild endlich geschafft |
| Probleme | Feuerball wird nicht erstellt und ausgeführt…Verzweiflung |
| Hilfestellung | Internet  https://zetcode.com/javagames/movingsprites/ |

**19.01.2022**

|  |  |
| --- | --- |
| Zeit (geplant) | 4h |
| Zeit (tatsächlich) | 4h |
| Geplante Ziele | Klasse: Fireball Funktion: draw  Klasse: Shooter Funktion: shoot |
| Erreichte Ziele | Klasse: Fireball Funktion: draw |
| Ausgeführte Arbeiten |  |
| Probleme | Feuerball funktioniert immer noch nicht. VERZWEIFLUNG WIRD GRÖSSER |
| Hilfestellung |  |

**21.01.2022**

|  |  |
| --- | --- |
| Zeit (geplant) | 4h |
| Zeit (tatsächlich) | 4h |
| Geplante Ziele | Klasse: Fireball Funktion: draw  Klasse: Shooter Funktion: shoot  Klasse: Enemy Funktion : move |
| Erreichte Ziele | Klasse: Fireball Funktion: draw  Klasse: Enemy Funktion : move  Klasse: Shooter Funktion: shoot |
| Ausgeführte Arbeiten | Feuerball wird endlich gezeichnet.  Die Gegner laufen richtung des Spielers.  Wenn der Spieler schiesst wird ein Feuerball in die Richtung geschossen in welche der Spieler schaut |
| Probleme | Funktion implementieren |
| Hilfestellung | Eric Schneider |

**22.01.2022**

|  |  |
| --- | --- |
| Zeit (geplant) | 4h |
| Zeit (tatsächlich) | 4h |
| Geplante Ziele | Klasse: LevelViewController  Funktion: checkCollision |
| Erreichte Ziele | - |
| Ausgeführte Arbeiten | - |
| Probleme | Die Feuerbälle treffen die Gegner aber nichts passiert |
| Hilfestellung | Finde nichts |

**23.01.2022**

|  |  |
| --- | --- |
| Zeit (geplant) | 4h |
| Zeit (tatsächlich) | 4h |
| Geplante Ziele | Klasse: LevelViewController  Funktion: checkCollision |
| Erreichte Ziele | - |
| Ausgeführte Arbeiten |  |
| Probleme | Collision funktioniert noch nicht |
| Hilfestellung | Internet  https://zetcode.com/javagames/movingsprites/ |

**24.01.2022**

|  |  |
| --- | --- |
| Zeit (geplant) | 6h |
| Zeit (tatsächlich) | 13h |
| Geplante Ziele | Klasse: LevelViewController  Funktion: checkCollision  Dokumentation fertig |
| Erreichte Ziele | Klasse: LevelViewController  Funktion: checkCollision  Dokumentation fertig |
| Ausgeführte Arbeiten | Die Funktion Collision funktionert nun. |
| Probleme | Collsion wurde nicht ausgeführt |
| Hilfestellung | Sasha Kokot, Wendelin Haller |

# Fazit

## Gelernt

**Spass:**

Zwischendurch als ich sehr gut voran kam hat es mir Spass gemacht zu programmieren. Ich sah es nicht mehr als ein muss sondern als können.

**Fragen:**

Ich bin gerne sehr selbstständig und frage nicht gerne, da es für mich ein Zeichen von "Unfähigkeit" ist. Ich habe über das Projekt gelernt, dass es nicht schadet Kollegen nach Sachen zu fragen.

## Schwierigkeiten

**Zeiteinteilung:**

Zum Schluss musste ich nochmals sehr lange Arbeiten. Ich habe mir zu viel aufgeschoben und musste dies nun nacharbeiten.

**Fixieren:**

Es fällt mir schwer kleinere Fehler zu ignorieren. Falls ich einen sehe muss ich diesen verbessern, daraus entsteht dann eine Kettenreaktion und ich vergesse mein eigentliches Ziel aus den Augen.

# Hilfe/ Quellen

**Wendelin Haller:** <https://github.com/WendelinH/M226B_SpaceShip/tree/main/src> + persönlich

Wendelin Haller hat mir die Struktur des Projektes so wie Teile des Codes zur Verfügung gestellt. Dazu hat er mir bei Fehlern oder Methoden, welche nicht funktioniert haben, geholfen. Von Wendelin hab ich auch das meiste von den **Controllern** kopiert.

**Eric Schneider:** persönlichEric Schneider hat mir am 21.01.2022 geholfen als ich gerade sehr viele Probleme mit der Funktion shoot in der Klasse Shooter hatte.

**Sasha Kokot:** persönlichSasha Kokot brachte die Idee mit der man die Gegner in einer ArrayListe speichert, diese Klont, ein Objekt daraus entfernt und diese wieder zurück auf die erste ArrayListe gibt.

**Internet:**

<https://zetcode.com/javagames/movingsprites/>

<https://forum.byte-welt.net/t/animierte-spielfigur-erstellen/10698>

<https://stackoverflow.com/questions/42146336/java-2d-game-shooting-in-different-directions>

# **Eidesstattliche Erklärung**

Der/Die Verfasser/in erklärt an Eides statt, dass er/sie die vorliegende Arbeit selbständig, ohne fremde Hilfe und ohne Benutzung anderer als die angegebenen Hilfsmittel angefertigt hat. Die aus fremden Quellen (einschliesslich elektronischer Quellen) direkt oder indirekt übernommenen Gedanken sind ausnahmslos als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden.

……………………………………… ……………………………………...

Ort, Datum Unterschrift des/der Verfassers/in

Bewertungsraster

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung