Duck Around

Das Spieleprojekt

Name: David Noskov

Abgabedatum: 18.06.2024

Klasse: 11BGPI1

Fach: Praktische Informatik

Lehrkraft: Herr Magdeburg

Inhalt

[1. Einleitung 1](#_Toc167784381)

[2. Planung 1](#_Toc167784382)

[Produktskizze 1](#_Toc167784383)

[Prototypen 1](#_Toc167784384)

[Versionierung 1](#_Toc167784385)

[3. Entwurf 1](#_Toc167784386)

[Klassendiagramm 1](#_Toc167784387)

[Struktogramm einer Methode 1](#_Toc167784388)

[4. Durchführung 1](#_Toc167784389)

[Beschreibung Ablauf 1](#_Toc167784390)

[Beschreibung des Spielablaufs mit Verweis zur Methode 1](#_Toc167784391)

[Fehlerprotokoll und Maßnahmen zur Behebung 1](#_Toc167784392)

[Kommentierter Quelltext 1](#_Toc167784393)

[Begründung von Abweichungen gegenüber Planung 1](#_Toc167784394)

[Klassendokumentation 1](#_Toc167784395)

[Screenshots 1](#_Toc167784396)

[5. Fazit 1](#_Toc167784397)

# Einleitung

Im Rahmen einer Projektarbeit im Fach „Praktische Informatik“ wurde die Aufgabe erteilt, ein Spiel zu programmieren in einer beliebigen Game-Engine. Diese Projektarbeit erfolgt unter Leitung der Lehrkräfte, Herrn Magdeburg und Frau Engel. Die Projektarbeit fand vom 7. Mai bis zum 18. Juni statt, wo am Ende das vollendete Spiel sowie die dazugehörige Dokumentation abgegeben werden soll. Zur Verfügung standen in der Schule die Räumlichkeiten U40 und U42. Ebenfalls wurde an diesem Projekt von Zuhause aus gearbeitet. Ziel des Projekts war es sich mit der Game Engine eigener Wahl zu befassen und die Grundlagen der Spieleentwicklung kennenzulernen sowie das Konzept von Objektorientierte Programmierung.

# Planung

## Produktskizze

Inspiriert von Spielen wie „Vampire Survivors“ und „Brotato“ wurde sich entschieden ein Rouge-like Spiel zu erstellen. Für die Grafik wurde sich auf Pixel Art entschieden. Der Spieler spielt einen Charakter, der in einer Spielwelt spawned und sich darin bewegen kann. Der Spieler hat die Möglichkeit anzugreifen und seine Fähigkeiten verbessern, wie Angriffsschaden, Defensive, Bewegungsgeschwindigkeit und Kritischen Schaden. Die Spielwelt ist von Zäunen umgeben. Rund um die Spielwelt sollen Gegner spawnen die auf den Spieler zulaufen und angreifen. Wenn alle Gegner besiegt worden sind, kommt eine neue Welle mit größerer Anzahl an stärkeren Gegnern. Jeder Gegner soll Erfahrung nach dem Tod an den Spieler geben. Nach genug Erfahrung soll der Spieler aufleveln können und hat dann die Möglichkeit seine Fähigkeit zu verbessern. Wenn der Spieler keine Leben mehr hat endet das Spiel und man muss dann von neu starten.

## Prototypen

Skizzen wurden gefertigt, um eine Grobe Idee zu haben, wie das Spiel aussehen soll am Ende des Projektes. Gefordert war es

ル / ル 0 
う 
引 0 〇 つ ク 0 ク 
( 0 レ e 
日 ノ 5C0 / し : 
ル ク 000 ワ 00 

Abb. 1: Spielanfang

/ 0 / ル 0 
0 
VUI し 
グ 
〇 
〇 
0 
( 0 レ e こ 
司 0 ク 000 ワ 00 
h) 5C0 / し 。 
レ 
0 0 
ナ 7 レ 
気 1 ム 
ワ 
フ フ 0 

Abb. 2: Typische Szene im Spiel

0 し / ル 0 
0 
0 
う 
引 〇 つ ク 0 ク 
( 0 レ e 
日 易 。 尾 : ル 00 ク ク 0 
〇 〇 ワ 
ク ク つ つ 
し 〇 ク レ を 
、 フ ワ り CD 0 
し 0 レ 08 
凵 7 じ 

Abb. 3: Spielende

## Versionierung

|  |  |
| --- | --- |
| 07. Mai | * Entwicklung der Spieleidee * Zeichnen des Prototyps |
| 14. Mai | * Vorläufige Spielewelt erstellt * Zeichnen der Spielerfigur, Gegner und Tiles * Erstellen einer Spielerfigur mit Bewegung |
| 21. Mai | * Spieler kann angreifen * Spieleroberfläche erstellt * Erstellung einer Gegnerfigur * Erstellen eines Titel Screen * Erstellen eines Gameover Screen |
| 28. Mai | * Spieler greift Richtung Maus an * Gegner kann Schaden bekommen * Gegner kann dem Spieler Schaden zufügen * Spieler kann verlieren |
| 30. Mai | * Gegner können spawnen * Gegner Wellen Logik implementiert * Kompletter Game Loop fertig |
| 04. Juni | * Gegner verfolgt den Spieler mit Rücksicht auf die Spielewelt * Erstellung von Musik * Zeichnen von Tiles für die Spielewelt |
| 11. Juni |  |

# Entwurf

## Klassendiagramm

## Struktogramm einer Methode

# Durchführung

## Beschreibung Ablauf

Am 7. Mai hat offiziell das Spieleprojekt begonnen und am gleichen Tag wurde die Idee entwickelt. Vor dem Projekt war das Spieleprojekt bekannt und im Vorhinein wurde entschieden ein Roguelike Spiel zu programmieren im Style von bekannten Titeln wie z.B. „Vampire Survivors“. Zunächst wurde eine Skizze

Entschieden wurde sich das Spiel auf der Godot Game Engine mit der Programmiersprache C# zu programmieren. Eine alternative Wahl wäre die Unity Game Engine gewesen, doch aufgrund einer zuvor diskutierten Runtime-Gebühr von Unity wurde sich dagegen entschieden.

Es wurde sich entschieden C# anstatt GDScript zu verwenden, aufgrund der vorhanden Familiarität und Erfahrung.

## Beschreibung des Spielablaufs mit Verweis zur Methode

## Fehlerprotokoll und Maßnahmen zur Behebung

## Kommentierter Quelltext

## Begründung von Abweichungen gegenüber Planung

## Klassendokumentation

## Screenshots

# Fazit