

# **Pflichtenheft**

Bauhofer Christian,  
Hofer Clemens,  
Knoflach Lukas

Klasse: **5AHWII**

Fach: **PPM**

Thema: **2D – Jump & Run – Game**

# Inhaltsverzeichnis

1. Zielbestimmung .....	3
1.1 Musskriterien .....	3
1.2 Kannkriterien .....	3
1.3 Abgrenzungskriterien .....	3
2. Produkteinsatz .....	4
2.1 Zielgruppen .....	4
2.2 Betriebsbedingungen .....	4
3. Produktumgebung .....	4
3.1 Software .....	4
3.1.1 XNA – Game Studio: .....	4
3.2 Hardware .....	4
4. Produktfunktionen .....	5
4.1 Use – Case – Diagramm .....	5
5. Benutzeroberfläche.....	7
6. Qualitäts - Zielbestimmung .....	7
7. Globale Testszenarien und Testfälle .....	7
8. Entwicklungsumgebung .....	7
9. Zeitplan.....	7
10. Arbeitsaufteilung .....	8

## 1. Zielbestimmung

Das Spiel ist ein zweidimensionales Jump & Run Spiel, das nur einzeln gespielt werden kann. Das Leveldesign, Gegenstände und der Charakter werden von der Gruppe selbst entwickelt und designed.

### 1.1 Musskriterien

- Spiel muss sich starten lassen
- Level soll geladen werden
- Charakter muss im Spiel laufen, springen können
- Charakter muss nach vorne und nach hinten laufen können
- Charakter muss auf Hindernisse springen können und diese überspringen können (Kollisionserkennung)
- Gegner im Spiel
- Gegner sind aktiv (wenn Charakter im bestimmten Sichtfeld ist, schießt der Gegner auf ihn)
- Charakter zielt mit der Maus beim Schießen
- Level abschließen
- Highscore anschaubar
- Spiel speichern und laden
- Anleitung lesbar

### 1.2 Kannkriterien

- Charakter soll schießen können
- Upgrades (schneller laufen, höher springen, mehr Leben usw.)
- verschiedene Waffenauswahl
- verschiedene Level
- verschiedene Charaktere
- verschiedene Gegner bzw. Fallen

### 1.3 Abgrenzungskriterien

- kein Browser – Game (nicht online)
- kein Multiplayer

## 2. Produkteinsatz

### 2.1 Zielgruppen

- Fans von Jump&Run – Games
- Leute, die gute Spiele mit einfacher Steuerung bevorzugen
- Jeder, der gerne Computerspiele spielt

### 2.2 Betriebsbedingungen

Um dieses Spiel spielen zu können, muss die CLR (Common Language Runtime) des .NET – Frameworks auf dem PC installiert sein.

## 3. Produktumgebung

### 3.1 Software

Verwendet wird das XNA – Game – Studio von Microsoft Visual Studio (NET Framework 4.0).  
Programmiert wird mit C#.

#### 3.1.1 XNA – Game Studio:

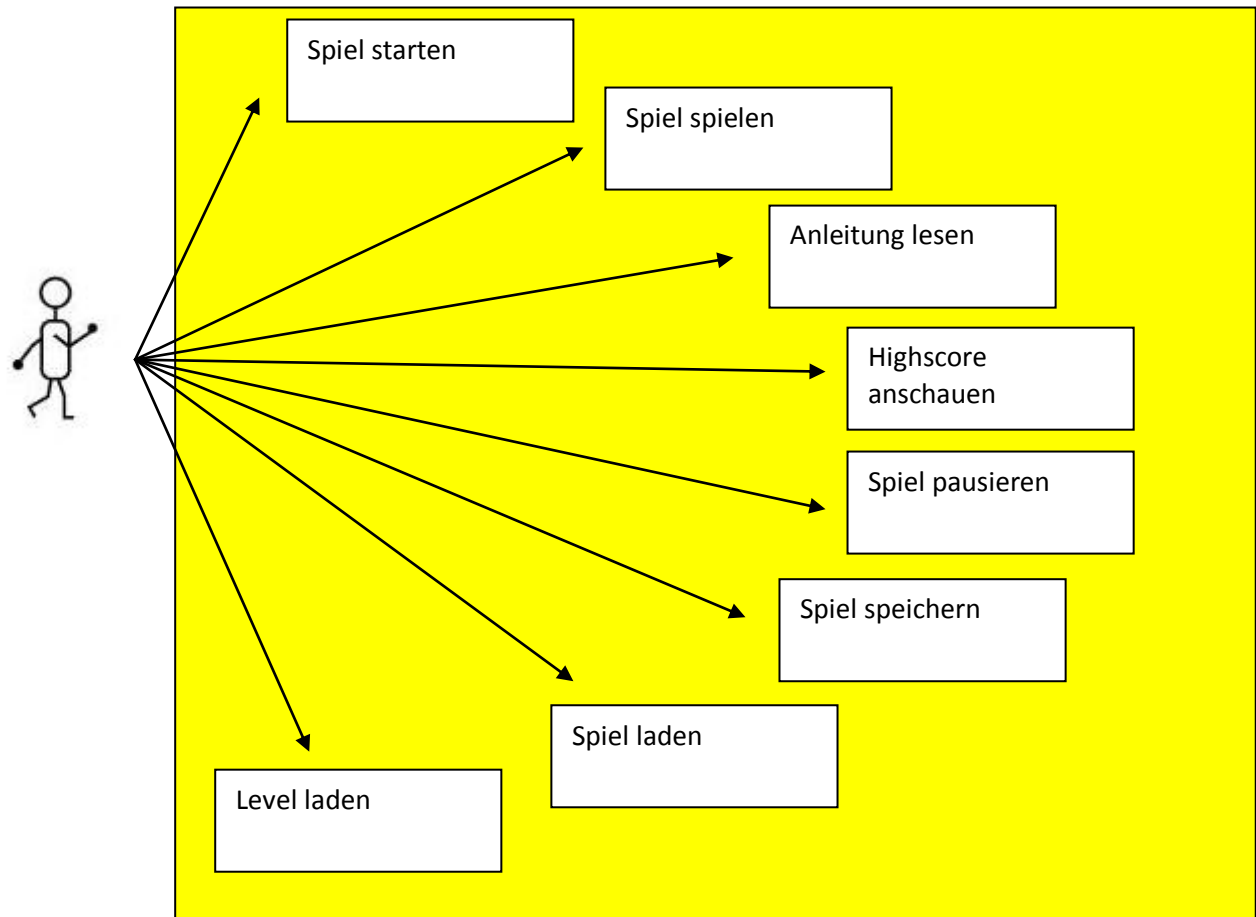
XNA Game Studio ist eine Technologie zur Spieleentwicklung für Microsoft Windows, Xbox 360 sowie Windows Phone und bietet dem Entwickler ein umfassendes Set von Werkzeugen um die Spieleentwicklung zu vereinfachen. Dabei ist zu sagen, dass XNA keine Spieleengine ist. Eine Spieleengine besteht meistens aus einer umfangreichen Suite an visuellen Editoren, zum Beispiel Level- und Shader- Editoren (Software-Module, die bestimmte Bildsynthese-Effekte bei der 3D-Computergrafik implementieren). Bei XNA stehen derartige Editoren nicht zur Verfügung, dafür bietet es Hilfsmittel unter anderem eine spezialisiertes Komponentenmodel und eine Anzahl an Bibliotheken. Diese haben den Zweck, die gängigsten Prozesse in der Spieleentwicklung wie das Ansteuern der Grafikhardware oder das Abspielen von Sounds zu erleichtern. Grundsätzlich wird bei XNA die Programmiersprache C# verwendet.

### 3.2 Hardware

Verwendet werden Laptops und Schulcomputer.

## 4. Produktfunktionen

### 4.1 Use – Case – Diagramm



- Spiel starten:

Darunter ist gemeint, dass das Spiel mit einem Doppelklick gestartet werden soll. Anschließend soll das Spielfenster mit dem Hauptmenü erscheinen und folgende Punkte zur Auswahl stehen:

- Spielen
- Anleitung
- Highscore

- Spiel spielen:

Es wird gefragt ob ein neues Spiel oder ein gespeichertes Spiel geladen werden soll.

Sollte ein neues Spiel gewählt werden, startet das Spiel ganz normal am Spielanfang und der Spieler kann mit seinem Charakter laufen, springen und schießen und bis zur Levelabschließung spielen.

- Anleitung lesen:

Bei diesem Menüpunkt erscheint ein neues Fenster, indem die Steuerung des Spiels erklärt wird. Rechts oder links unten ist ein anklickbares Symbol, das den User zurück zum Hauptmenü bringt.

- Highscore anschauen:

Wenn man Highscore auswählt, wird dem Spieler sein bestes Resultat (in Punkten) in einem neuen Fenster gezeigt. Wiederum befindet sich hier ein Symbol, mit dem man zum Hauptmenü zurück gelangt.

- Spiel pausieren:

Während des Spiels kann der User das Spiel pausieren.

Hier kann er folgende Funktionen auswählen:

- Fortsetzen (Spiel wird dort fortgesetzt, wo der User das Spiel pausiert hat)
- Spiel speichern (siehe später)
- Anleitung lesen (die Anleitung zum Spiel wird nochmals gezeigt → siehe oben)
- Spiel beenden (das Spiel wird nicht gespeichert und der User gelangt zum Hauptmenü)

- Spiel speichern:

Wenn der User während des Spielens das Spiel pausiert, erscheint ein Button mit Spiel speichern. Die Daten werden in ein Textfile geschrieben und werden somit lokal abgespeichert.

- Spiel laden:

Wenn man im Hauptmenü auf „Spielen“ klickt, kann zwischen „neues Spiel“ (siehe Spiel spielen) oder „Spiel laden“ gewählt werden.

Das in einem Textfile abgespeicherte Spiel wird geladen und der User kann das Spiel fortsetzen.

- Level laden:

Wenn der User das Spiel spielen will, soll der Level geladen werden. Das heißt, das richtige Hintergrundbild, die Musik, die Gegner und die richtige Gegenstände (Kisten, Plattformen usw.)

## 5. Benutzeroberfläche

Beim Starten des Spiel erscheint ein Hintergrundbild und alle Punkte sind per Mausklick auswählbar.

## 6. Qualitäts - Zielbestimmung

Das Spiel ist sehr benutzerfreundlich, da es so einfach wie möglich gestaltet wird. Es gibt wenige Einstellungsmöglichkeiten, somit sollte jeder Spieler einfach mit dem Spiel zurechtkommen.

Der Spieler kann nur spielen, speicher, laden und die Anleitung lesen.

## 7. Globale Testszenarien und Testfälle

Wenn ein Programmerteil fertig gestellt ist (zB: Starten, Laufen, Schießen usw.), wird dies sofort getestet. Sollte dieser Programmcode nicht funktionieren, wird versucht, den Fehler zu beheben.

Sobald das Spiel werden

## 8. Entwicklungsumgebung

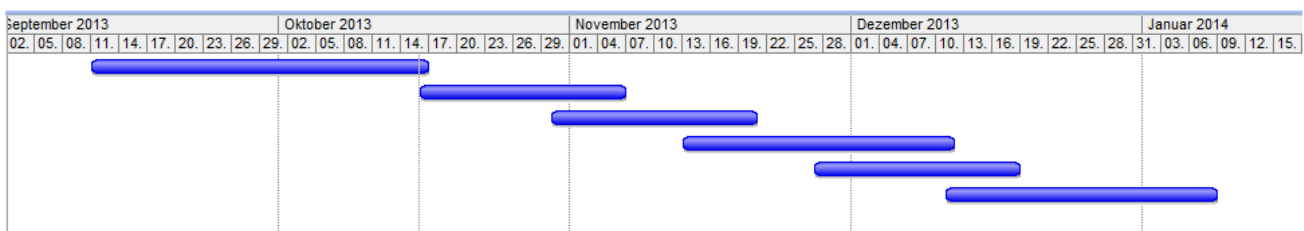
Das Spiel wird an den Laptops programmiert, da die Schulcomputer Visual Studio nicht installiert haben. Programmiert wird hauptsächlich im Unterricht.

Die Programmierschritte werden mittels GitHub abgespeichert und so für die anderen Mitarbeiter und für den Professor zugänglich gemacht.

Programmiert wird mit C#, verwendet wird das XNA – Game – Studio von Microsoft Visual Studio (siehe 3.1.1).

## 9. Zeitplan

	i	Vorgangsname	Dauer	Anfang	Ende
1		Pflichtenheft	26 Tage?	Mi 11.09.13	Mi 16.10.13
2		Level laden (Scrolling, Hintergrund, Blöcke)	16 Tage	Mi 16.10.13	Mi 06.11.13
3		Animation des Charakters (laufen, springen)	16 Tage	Mi 30.10.13	Mi 20.11.13
4		KI - Gegner	21 Tage	Mi 13.11.13	Mi 11.12.13
5		Level laden (Highscore speichern)	16 Tage	Mi 27.11.13	Mi 18.12.13
6		Menüpunkte (Anleitung, Spiel laden & speichern)	21 Tage	Mi 11.12.13	Mi 08.01.14



## 10. Arbeitsaufteilung

<b>Bauhofer</b>	<b>Hofer</b>	<b>Knoflach</b>
Rendern (Scrolling, Kollisionserkennung usw.) Allgemeines	Level (zB: zerstörbar Blöcke, dyn. Elemente) Menü (Highscores, Speichern usw.)	Spieler und Gegner (Animationen, KI usw.)
Grafik Level	Grafik User Interface	Grafik Spieler +Gegner