

# **ScribbleFight - Plattformbasiertes 2D-Brawl-Spiel mit Trainings-KI und Bildererkennung**

## **DIPLOMARBEIT**

verfasst im Rahmen der

**Reife- und Diplomprüfung**

an der

**Höheren Abteilung für Medientechnik und Informatik**

Eingereicht von:

Himmetsberger Jonas  
Rafetseder Tobias  
Weinzierl Ben

Betreuer:

Aistleitner Gerald

Projektpartner:

none

Leonding, April 2022

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Die Arbeit wurde bisher in gleicher oder ähnlicher Weise keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Die vorliegende Diplomarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

Leonding, April 2022

S. Schwammal & S. Schwammal

Zur Verbesserung der Lesbarkeit wurde in diesem Dokument auf eine geschlechtsneutrale Ausdrucksweise verzichtet. Alle verwendeten Formulierungen richten sich jedoch an beide Geschlechter.

# Abstract

Brief summary of our amazing work. In English. This is the only time we have to include a picture within the text. The picture should somehow represent your thesis. This is untypical for scientific work but required by the powers that are. Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.



# Zusammenfassung

Zusammenfassung unserer genialen Arbeit. Auf Deutsch. Das ist das einzige Mal, dass eine Grafik in den Textfluss eingebunden wird. Die gewählte Grafik soll irgendwie eure Arbeit repräsentieren. Das ist ungewöhnlich für eine wissenschaftliche Arbeit aber eine Anforderung der Obrigkeit. *Bitte auf keinen Fall mit der Zusammenfassung verwechseln, die den Abschluss der Arbeit bildet!* Suspendisse vel felis.

Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Abkürzungen . . . . .	1
1.2	Autoren der Diplomarbeit . . . . .	1
1.2.1	Himmetsberger Jonas . . . . .	1
1.2.2	Rafetseder Tobias . . . . .	1
1.2.3	Weinzierl Ben . . . . .	1
<b>2</b>	<b>Zieldefinition</b>	<b>2</b>
2.1	Projektanlass . . . . .	2
2.2	Ziele . . . . .	2
2.3	Aufgabenverteilung . . . . .	2
2.3.1	Web-Game und Deployment [R] . . . . .	2
2.3.2	Gamedesign und Lobby-System [B] . . . . .	3
2.3.3	Aufbereitung der Spielumgebung und Künstliche Intelligenz [H] . . . . .	3
2.4	Dokumente . . . . .	3
<b>3</b>	<b>Umfeldanalyse</b>	<b>4</b>
3.1	Ähnliche Spiele . . . . .	4
3.1.1	Super Smash Bros (SSM) . . . . .	4
3.1.2	Brawlhalla . . . . .	4
3.1.3	Stick Fight: The Game . . . . .	4
3.2	Ist-Zustand . . . . .	5
<b>4</b>	<b>Technologien</b>	<b>6</b>
4.1	JavaScript [R] . . . . .	6
4.1.1	p5.js / p5.play [R] . . . . .	6
4.1.2	node.js [R] . . . . .	7
4.1.3	Socket IO [R] . . . . .	7

4.2	Cloud Computing [R]	7
4.2.1	Docker [R]	7
4.2.2	Kubernetes [R]	7
4.3	Python [H]	7
4.3.1	Flask [H]	7
4.3.2	OpenCV2 [H]	7
4.3.3	PIL [H]	7
4.3.4	TensorFlow und Keras [H]	7
4.3.5	OpenAI Gym [H]	7
<b>5</b>	<b>Umsetzung</b>	<b>8</b>
5.1	Web-Game [R]	9
5.1.1	Frontend [R]	9
5.1.1.1	Architektur [R]	9
5.1.1.2	p5.js [R]	9
5.1.2	Backend [R]	9
5.1.2.1	Node.js [R]	9
5.1.2.2	SocketIO [R]	9
5.1.3	Gamephysics [R]	9
5.1.4	Hitregistration [R]	9
5.1.5	Collisiondetection [R]	9
5.1.6	Regeln und Spielablauf [R]	9
5.2	Deployment [R]	9
5.2.1	Docker [R]	9
5.2.2	Leo-Cloud [R]	9
5.3	Map-Erkennung [H]	9
5.3.1	Objekterkennung [H]	9
5.3.1.1	Beschreibung der Funktionalität [H]	9
5.3.2	Open-CV2 [H]	9
5.3.2.1	Umwandlung der Bilder in für das Spiel brauchbare Daten [H]	9
5.3.3	Kommunikation mit der Lobby via Flask und Flask SocketIO [H]	9
5.4	KI [H]	9
5.4.1	Lernen mit OpenAI-Gym [H]	9
5.4.2	Künstliche Intelligenz Definition [H]	10
5.4.3	Reinforcement Learning [H]	10

5.4.4	Die ScribbleFight-KI [H]	10
5.4.4.1	Beschreibung der Funktionalität [H]	10
5.4.4.2	Warum Python? [H]	10
5.4.5	Tensorflow und Keras [H]	10
5.4.5.1	Tensorflow [H]	10
5.4.5.2	Keras [H]	10
<b>6</b>	<b>Evaluation des Projektverlaufs</b>	<b>11</b>
6.1	Meilensteine	11
6.2	Gelerntes	11
6.3	Was würden wir anders machen?	11
	<b>Literaturverzeichnis</b>	<b>VII</b>
	<b>Abbildungsverzeichnis</b>	<b>VIII</b>
	<b>Tabellenverzeichnis</b>	<b>IX</b>
	<b>Quellcodeverzeichnis</b>	<b>X</b>
	<b>Anhang</b>	<b>XI</b>

# **1 Einleitung**

## **1.1 Abkürzungen**

## **1.2 Autoren der Diplomarbeit**

### **1.2.1 Himmetsberger Jonas**

### **1.2.2 Rafetseder Tobias**

### **1.2.3 Weinzierl Ben**



## **2 Zieldefinition**

### **2.1 Projektanlass**

Die Möglichkeit seiner Kreativität freien Lauf zu lassen ist bei den meisten populären Online-Spielen sehr eingeschränkt, da man wenig Einfluss auf die Spielumgebung hat. Unsere Arbeit soll diesem Problem entgegenwirken. Der Spieler kann selbst entscheiden, wie die 2D-Spielumgebung auszusehen hat, indem er diese auf ein Blatt Papier zeichnet, welche dann via Bilderkennung als spielbare Welt aufbereitet wird.

### **2.2 Ziele**

Bis zum Abgabetermin sollen folgende Ziele erreicht werden:

- Das Spiel soll als Browserspiel funktionsfähig sein.
- Für die Benutzer soll es möglich sein ihre eigenen Kampfumgebungen zu erschaffen.
- Das Spiel soll als online-Multiplayer "Player versus PlayerSpiel funktionieren.
- Ein eigener Modus, in welchem der Spieler als Singleplayer gegen eine funktionsfähige KI antreten kann, soll umgesetzt werden.

### **2.3 Aufgabenverteilung**

Dadurch, dass die Diplomarbeit drei mitwirkende Schüler hat, wurde das Thema in drei ähnlich anspruchsvolle Teile unterteilt. Diese werden im Folgenden genauer erläutert:

#### **2.3.1 Web-Game und Deployment [R]**

Die wichtigsten Punkte, die im Bezug auf das Web-Game umzusetzen zu waren, sind:

- Die Spielphysik, also wie sich Spieler und Objekte verhalten

- Die Collisiondetection von Spielern mit der Umgebung und mit Objekten
- Die Hitregistration, falls ein Spieler von etwas getroffen wurde
- Ab wann ist das Spiel zu Ende, beziehungsweise wann hat jemand gewonnen

Das Deployment des Projekts in die Leocloud, ein Cloud-System der HTL-Leonding, soll mittels Kubernetes erfolgen.

### **2.3.2 Gamedesign und Lobby-System [B]**

### **2.3.3 Aufbereitung der Spielumgebung und Künstliche Intelligenz [H]**

Die Aufbereitung der Spielumgebung sollte folgende Funktionen erfüllen:

- Erkennung der Konturen in einer Live-View
- Aufnahme soll in brauchbare Daten umgewandelt werden

Folgende Forderungen waren an die KI gestellt:

- Die KI soll auf einem herausfordernden Niveau agieren
- Im Zuge der Forschung sollte ein Vergleich zwischen brauchbaren Lernalgorithmen gemacht werden

Im Laufe der Diplomarbeit und der damit zusammenhängenden Forschung änderte sich oft die Vorstellung darüber, wie das Endprodukt auszusehen hat.

## **2.4 Dokumente**

## 3 Umfeldanalyse

### 3.1 Ähnliche Spiele

Es gibt schon viele "Player vs PlayerBrawlspele. Doch solche Spiele, die man ohne Download im Browser miteinander spielen kann, findet man selten.

#### 3.1.1 Super Smash Bros (SSM)

Super Smash Bros (SSB) sind eine Reihe von plattform-basierten Videospielen. Diese sind von Nintendo entwickelt und beinhalten die bekanntesten Charakteren des Unternehmens. Figuren, wie Super Mario oder Sonic, bekämpfen sich in einer Arena mit dem Ziel sich gegenseitig von einer Plattform zu stoßen. Was jedoch fehlt, ist die Plattformunabhängigkeit, da das Spiel nur auf Nintendo-Systemen funktioniert. Außerdem besteht eine Limitierung in der Auswahl von Spielumgebungen.

#### 3.1.2 Brawlhalla

Brawlhalla ist ein von Blue Mammoth entwickeltes 2D-Kampfspiel, und wurde für alle gängigen Betriebssysteme entwickelt. Auch wie in Scribble-Fight ist es das Ziel, den Gegner von einer Plattform zu stoßen.

Der Nachteil hierbei ist, dass man sich vor dem Spielen einen Account erstellen und dann einen Download abschließen muss. Dazu kommt noch, dass man die Spielumgebung nie beeinflussen kann. Bei Scribble-Fight ist das nicht so.

#### 3.1.3 Stick Fight: The Game

In dem Spiel Stick Fight kämpft man als Strichmännchen gegeneinander, die durch eine Ragdoll-Engine gesteuert werden. Auch wie bei schon bei vorher erwähnten Spielen, kann man aber nicht einfach plattformunabhängig im Browser gegeneinander antreten. Für die Spielekonsole Nintendo Switch zum Beispiel, gibt es das Spiel nicht. Hinzu



Abbildung 1: Brawlhalla Spielumgebung

kommt, dass man auch die Umgebung nicht selbst frei erstellen kann, so wie es bei Scribble-Fight möglich ist.

## 3.2 Ist-Zustand

Dadurch das es eine Diplomarbeit ist, fängt alles bei 0 an. Es gibt jedoch Frameworks, welche die Erarbeitung erleichtern. Zum Beispiel im Falle des Spiels, welches mittels Webtechnologien umgesetzt wird, kann p5.js verwendet werden, welches die Umsetzung eines Spiels erleichtert. Ein weiteres Beispiel ist die Objekterkennung für die Spielumgebung (Map). Diese wird von Open-CV übernommen.

# 4 Technologien

## 4.1 JavaScript [R]

Für mich war es eine leichte Entscheidung JavaScript zu verwenden, da die Programmiersprache ideal für den Browser geeignet ist, und man damit nicht nur im Frontend, sondern auch im Backend programmieren kann.

### 4.1.1 p5.js / p5.play [R]

p5.js ist eine open-source JavaScript Library, die für Kreation von Spielen genutzt wird. p5.play ist eine Library für p5.js, mit der man visuelle Objekte managen kann. Außerdem beinhaltet es Features wie Animation-support, Kollisionserkennung, sowie aber auch Funktionen für Mouse und Tastatur Interaktionen. Es ist wichtig sich im Hinterkopf zu behalten, dass p5.play für barrierefreies und simples Programmieren gedacht ist, nicht für performantes. Es ist keine eigene Engine, und unterstützt auch keine 3D-Spiele.

p5.js ähnelt sich sehr stark mit Processing, eine Programmiersprache die man sich wie ein stark vereinfachte Version von Java vorstellen kann. Der Unterschied liegt darin, dass Java eine Umgebung, basierend auf der Java Programmiersprache ist, während p5.js eine Bibliothek, basierend auf der JavaScript Programmiersprache ist. Processing ist dafür geeignet, lokale Applikationen zu bauen, hingegen dazu kann p5.js nur im Browser ausgeführt werden.

p5.js ist also kurzgesagt ein direkter JavaScript Port für die Processing Programmiersprache.

Vorteile von p5:

- Man kann interaktive Programme entwickeln, die in jedem modernen Browser funktionieren (plattformunabhängig)
- Das Programm ist nicht nur lokal auf dem eigenen Gerät, was das Teilen sehr viel leichter macht

- Man hat die Option den p5.js Editor im Web zu verwenden: Überhaupt kein Aufwand, um loszuprogrammieren

Nachteile von p5:

- Ist schneller beim Pixel manipulieren
- Man kann Java Bibliotheken verwenden

### **4.1.2 node.js [R]**

### **4.1.3 Socket IO [R]**

## **4.2 Cloud Computing [R]**

### **4.2.1 Docker [R]**

### **4.2.2 Kubernetes [R]**

## **4.3 Python [H]**

### **4.3.1 Flask [H]**

### **4.3.2 OpenCV2 [H]**

### **4.3.3 PIL [H]**

### **4.3.4 TensorFlow und Keras [H]**

### **4.3.5 OpenAI Gym [H]**



# 5 Umsetzung

## 5.1 Web-Game [R]

### 5.1.1 Frontend [R]

#### 5.1.1.1 Architektur [R]

#### 5.1.1.2 p5.js [R]

### 5.1.2 Backend [R]

#### 5.1.2.1 Node.js [R]

#### 5.1.2.2 SocketIO [R]

### 5.1.3 Gamephysics [R]

### 5.1.4 Hitregistration [R]

### 5.1.5 Collisiondetection [R]

### 5.1.6 Regeln und Spielablauf [R]

## 5.2 Deployment [R]

### 5.2.1 Docker [R]

### 5.2.2 Leo-Cloud [R]

## 5.3 Map-Erkennung [H]

### 5.3.1 Objekterkennung [H]

#### 5.3.1.1 Beschreibung der Funktionalität [H]

### 5.3.2 Open-CV2 [H]

#### 5.3.2.1 Umwandlung der Bilder in für das Spiel brauchbare Daten [H]

### 5.3.3 Kommunikation mit der Lobby via Flask und Flask SocketIO [H]

## 5.4 KI [H]



darüber, wie das Endprodukt (KI) auszusehen hat, beziehungsweise, wie dieses aussehen kann. Limitierungen, welche vor der Forschung noch nicht bekannt und bewusst waren, begrenzten die Lernfähigkeit der KI. Auf Probleme, Lösungen und Ergebnisse wird jedoch noch näher in ... eingegangen.

### **5.4.2 Künstliche Intelligenz Definition [H]**

### **5.4.3 Reinforcement Learning [H]**

### **5.4.4 Die ScribbleFight-KI [H]**

#### **5.4.4.1 Beschreibung der Funktionalität [H]**

#### **5.4.4.2 Warum Python? [H]**

### **5.4.5 Tensorflow und Keras [H]**

#### **5.4.5.1 Tensorflow [H]**

#### **5.4.5.2 Keras [H]**

# 6 Evaluation des Projektverlaufs

## 6.1 Meilensteine

Ist es uns gut dabei gegangen diese einzuhalten?

## 6.2 Gelerntes

Haben wir bei der DA was gelernt? nein

## 6.3 Was würden wir anders machen?

JOA Was würden wir anders machen?

Aufzählungen:

- Itemize Level 1
  - Itemize Level 2
    - Itemize Level 3 (vermeiden)
- 1. Enumerate Level 1
  - a. Enumerate Level 2
    - i. Enumerate Level 3 (vermeiden)

**Desc** Level 1

**Desc** Level 2 (vermeiden)

**Desc** Level 3 (vermeiden)



# Literaturverzeichnis

# Abbildungsverzeichnis

1	Brawlhalla Spielumgebung . . . . .	5
---	------------------------------------	---

# Tabellenverzeichnis

# Quellcodeverzeichnis

# Anhang