

# LA1500: Projektdokumentation

---

Legloff

Matarese, Pfister, Frey, Kistler

Datum	Version	Änderung	Autor
08.03	0.1.0	IPE	Matarese, Pfister, Frey, Kistler
15.03	0.2.0	Realisieren	Matarese, Pfister, Frey, Kistler
22.03	0.3.0	Realisieren	Matarese, Pfister, Frey, Kistler
29.03	0.4.0	Realisieren	Matarese, Pfister, Frey, Kistler
05.04	1.0.0	Projektabschluss	Matarese, Pfister, Frey, Kistler

## 1. Informieren

### 1.1 Ihr Projekt

---

Unser Ziel ist es, eine Jump n Run Applikation zu entwickeln. Die Applikation sollte mit Unity erstellt werden. Die verwendete Programmiersprache wird C# sein. Das Game sollte eine angemessene Schwierigkeit haben und nicht zu einfach sein. Es sollte aber auch nicht unmöglich sein. In unserem Spiel ist das spezielle, dass man in einer tropischen Inselumgebung Level besiegen muss. Ausserdem hat man drei Leben und kann Münzen einsammeln, die den Score bestimmen.

### 1.2 Quellen

---

[https://www.youtube.com/playlist?list=PLf\\_j4Db1v3RkEknk50NWuOVL96xv6if-7](https://www.youtube.com/playlist?list=PLf_j4Db1v3RkEknk50NWuOVL96xv6if-7)

für spezifische Themen

<https://unity.com/de/learn>

offizielle Website mit Kurs für Beginner vielleicht ein wenig zu lang

[https://www.youtube.com/watch?v=DZbQRw-ftnU&list=PL\\_pgkvxZ6ho1g\\_e56fct7Cm6bgQBhmAqN](https://www.youtube.com/watch?v=DZbQRw-ftnU&list=PL_pgkvxZ6ho1g_e56fct7Cm6bgQBhmAqN)

Crashkurs

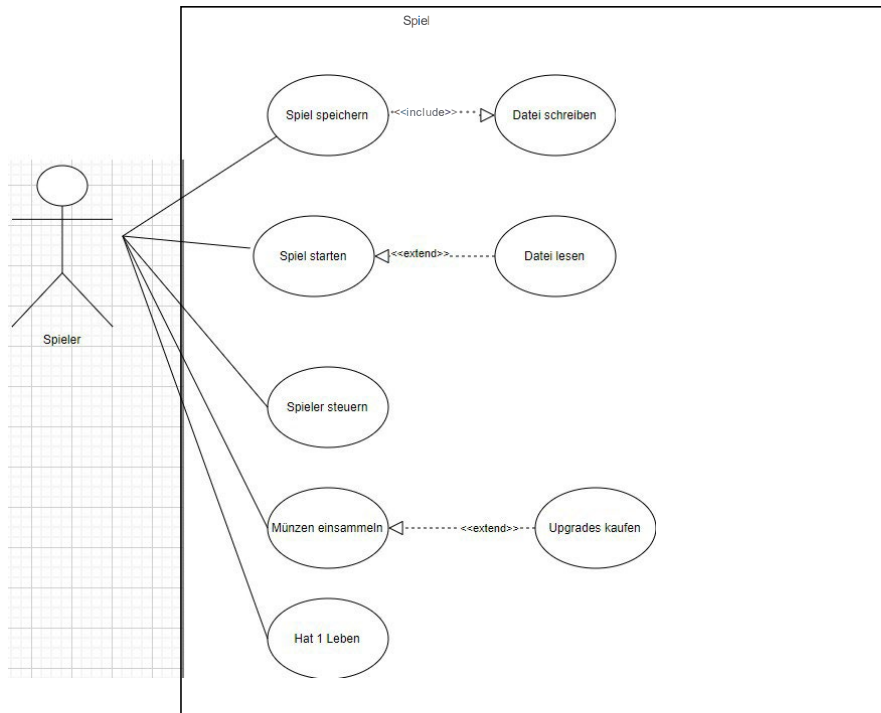
### 1.3 Anforderungen

---

Nummer	Muss/ Kann?	Funktional? Qualität? Rand?	Beschreibung
1	Muss	Funktional	Der Charakter muss sich bewegen können
2	Muss	Funktional	Es gibt Gravitation
3	Muss	Funktional	Das Level hat ein Ziel
4	Muss	Funktional	Man kann Münzen einsammeln
5	Muss	Funktional	Man hat 3 Leben
6	Muss	Funktional	Man verliert Leben
7	Muss	Funktional	Bei 0 Leben stirbt man
8	Muss	Funktional	Es gibt 3 Level
9	Muss	Funktional	Es gibt einen Endboss
10	Muss	Funktional	Es gibt verschiedene Grafiken für Charakter und Objekte, sowie Landschaften
11	Muss	Funktional	Das Programm hat einen Münzzähler
12	Muss	Funktional	Das Programm hat ein Scoreboard
13	Muss	Funktional	Das Programm schreibt die Münzen am Ende des Levels auf ein Konto gut
14	Muss	Funktional	Das Programm errechnet einen Score mit den eingesammelten Münzen.
15	Kann	Funktional	Es gibt ein verstecktes Easter Egg
16	Muss	Funktional	Das Programm spielt in einer tropischen Inselwelt
17	Kann	Funktional	Das Programm speichert den Fortschritt
18	Muss	Qualität	Das Game wird auf einem 15 Zoll Bildschirm korrekt angezeigt
19	Muss	Rand	Das Spiel ist in C# geschrieben
20	Muss	Rand	Man kann das Spiel im Browser spielen

---

## 1.4 Diagramme



## 1.5 Testfälle

Nummer	Voraussetzung	Eingabe	Erwartete Ausgabe
1.1	Das Spiel wurde gestartet	"Leertaste"	Der Charakter springt
2.1	Das Spiel wurde gestartet	"Leertaste"	Der Charakter fällt wieder, falls er gesprungen ist.
2.2	Das Spiel wurde gestartet	"A"	Der Charakter fällt ins Void, falls er einen Sprung nicht schafft.
3.1	Der Spieler schafft ein Level	"A" "Leertaste"	Der Spieler schafft ein Level
4.1	Der Spieler springt	"A" "Leertaste"	Der Spieler sammelt Münzen ein.
5.1	Das Spiel wurde gestartet	-	Der Spieler hat 3 Leben
6.1	Der Spieler fällt	-	Der Spieler verliert ein Leben
7.1	Der Spieler fällt	-	Der Spieler hat keine Leben mehr und muss das Level neu starten.
8.1	Der Spieler spielt das Spiel durch	-	 Der Spieler hat das Spiel geschafft
9.1	Der Spieler gewinnt das Spiel	-	Der Endboss taucht auf
10.1	Das Spiel wurde gestartet	-	Es gibt versch. Grafiken
11.1	Das Spiel wurde gestartet	-	Das Spiel zählt die Münzen
12.1	Das Spiel wurde gestartet	-	Das Spiel zählt den Score
13.1	Das Spiel wurde gestartet	Der Spieler sammelt Münzen auf	Das Spiel zählt die Münzen aus allen Leveln zusammen
14.1	Das Spiel wurde gestartet	Der Spieler sammelt Münzen auf	Das Spiel zählt die gesammelten Münzen zusammen
15.1	Das Spiel wurde gestartet	Der Spieler läuft gegen eine bestimmte Wand	Der Spieler findet ein geheimes Easter Egg
15.2	Der Spieler findet das geheime Easter Egg	Der Spieler sieht das geheime Easter Egg	-

16.1	Das Spiel wurde gestartet	-	Das Spiel spielt in einer tropischen Inselwelt
17.1	Das Spiel wurde gestartet	Der Spieler speichert den Fortschritt und beendet das Spiel	Das Spiel speichert seinen Fortschritt

## 2. Planen

Nummer	Frist	Beschreibung	Zeit (geplant)	Zeit (effektiv)
1.1	15.03	Der Charakter kann sich bewegen	45	30
1.2	15.03	Man kann hüpfen	45	10
1.3	15.03	Man kann sich nach rechts bewegen	45	10
1.4	15.03	Man kann sich nach links bewegen	45	10
2.1	15.03	Gravitation funktioniert	45	10
2.2	15.03	Gravitation ist passend	45	10
3.1	22.03	Es gibt ein Ziel	45	10
4.1	22.03	Es gibt Münzen	45	10
4.2	22.03	Man kann die Münzen einsammeln	45	20
5.1	22.03	Man hat drei Leben	45	20
6.1	22.03	Man verliert Leben	45	20
6.2	22.03	Man kann nicht weniger als 0 Leben haben	45	20
6.3	22.03	Bei einem Sprung ins Leere, verliert man ein Leben	45	10
6.4	22.03	Falls man von einem Gegner getroffen wird, verliert man ein Leben	45	20
7.1	22.03	Bei 0 Leben ist man tot	45	20
7.2	22.03	Das Level startet bei 0 Leben neu	45	20
8.1	29.03	Es gibt 3 Level	3x45	120
8.2	22.03	Jedes Level hat ein anderes Design	3x45	120
9.1	29.03	Es gibt einen Endboss	3x45	120
9.2	29.03	Der Endboss bewegt sich intelligent	3x45	120
9.3	29.03	Man kann dem Endboss Schaden zufügen	45	
10.1	22.03	Es gibt verschiedene Grafiken für Charakter und Objekte sowie Landschaften	5x45	10
11.1	29.03	Das Programm hat einen Münzzähler	45	40
12.1	29.03	Das Programm hat ein Scoreboard	45	
13.1	29.03	Das Programm schreibt die Münzen am Ende des Levels auf ein Konto gut	45	
14.1	29.03	Das Programm errechnet einen Score mit den eingesammelten Münzen.	2x45	30
15.1	29.03	Es gibt ein verstecktes Easteregg	2x45	30

16.1	29.03	Das Programm spielt in einer tropischen Inselwelt	2x45	45
17.1	29.03	Das Programm speichert den Fortschritt ab	3x45	
18.1	29.03	Das Game wird auf einem 15 Zoll Bildschirm korrekt angezeigt	3x45	45
18.2	29.03	Alle Objekte haben eine annehme Grösse	45	45
19.1	29.03	Man kann das Spiel im Browser spielen	2x45	
19.2	29.03	Das Spiel hat die gleichen Mechaniken im Browser	45	

### 3. Entscheiden

---

Wir haben als Gruppe entschieden, dass wir unser 2D Game in Unity programmieren werden.

Wir haben uns am 15.03.22 dazu entschieden früher aufzuhören und dafür am Wochenende eine Stunde länger zu arbeiten. Wir treffen uns dazu entweder gemeinsam oder über Teams.

### 4. Realisieren

---

Siehe Arbeitspäcklein bei Punkt 2.

Das Projekt befindet sich auf github unter <https://github.com/oli-kis/Foxy-Moxy>

Falls sie das Spiel selber testen wollen: <https://play.unity.com/mg/other/foxy-moxy-web>

## 5. Kontrollieren

### 5.1 Testprotokoll

Nummer	Datum	Resultat	Durchgeführt
1.1	05.04.2022	OK	Matarese
2.1	05.04.2022	OK	Matarese
2.2	05.04.2022	OK	Matarese
3.1	05.04.2022	OK	Matarese
4.1	05.04.2022	OK	Matarese
5.1	05.04.2022	OK	Matarese
6.1	05.04.2022	OK	Matarese
7.1	05.04.2022	OK	Matarese
8.1	05.04.2022	OK	Matarese
9.1	05.04.2022	OK	Matarese
10.1	05.04.2022	OK	Matarese
11.1	05.04.2022	OK	Matarese
12.1	05.04.2022	NOK	Matarese
13.1	05.04.2022	NOK	Matarese
14.1	05.04.2022	NOK	Matarese
15.1	05.04.2022	OK	Matarese
15.2	05.04.2022	OK	Matarese
16.1	05.04.2022	OK	Matarese
17.1	05.04.2022	OK	Matarese

Das Wichtigste haben wir in der Zeit geschafft. Nur das Münzenzusammenzählen in den Levels und das Scoreboard haben wir zeitlich nicht geschafft.

## 6. Auswerten

Das Projekt ist sehr gut gelaufen. Wir konnten mithilfe des Tutorials sehr schnell die wichtigsten Punkte des Games abschliessen. Wir hätten vielleicht noch mehr geschafft, wenn wir in der zweiten Lektion unser Projekt nicht fast verloren hätten. Wir haben probiert ein Teil eines Projektes einzufügen und hatten viele Compile Errors und konnten das Spiel nicht mehr starten. Glücklicher und komischer Weise funktionierte das Programm nach einem Mal zippen dann wieder.