

Broccoli

Egloff, Pinto, Teixeira da Silva, Zehnder

Datum	Version	Änderung	Autor
	0.1.0	Erste Version	Egloff, Pinto, Teixeira da Silva, Zehnder
	0.2.0		Egloff, Pinto, Teixeira da Silva, Zehnder
	0.3.0		Egloff, Pinto, Teixeira da Silva, Zehnder
	0.4.0		Egloff, Pinto, Teixeira da Silva, Zehnder
	1.0.0	Projektabschluss	Egloff, Pinto, Teixeira da Silva, Zehnder

1. Informieren

1.1 Ihr Projekt

Unser Ziel ist es, Unity zu lernen. Wir haben uns ebenfalls für das entschieden, weil wir lernen wollen, wie man sich selbst Technologien beibringt. Wir wollen Grundwissen ansammeln, im Sinne von, wie man Spiele kreiert und wie man diese designt. Auch ist es unser Ziel besser zu lernen, wie man Sachen gut präsentiert.

Wir wollen ein kleines Spiel in der Unity Engine programmieren. Es ist ein 2D-Jump-and-Run, das stark an der Souls-Spiele Reihe inspiriert ist. Es besteht aus einer kleinen Levelpassage und einem Bossfight. Das Spiel beinhaltet auch ein Menü mit Einstellungen, in welchen man die Lautstärke einstellen kann. Das Spiel kann einen Spielstand speichern und löschen. Das Spiel wird im «Pixelgrafik-Stil» gemacht. Je nachdem, wie viel Zeit wir noch haben, werden wir eventuell noch ein weiteres Level hinzufügen.

1.2 Quellen

<https://docs.unity3d.com/Manual/index.html>

<https://www.youtube.com/watch?v=DZbQRw-ftnU>

[🎮 Einstieg - Programmierung! || Spiele Entwickeln für Anfänger mit Unity \[02-01\] 🎮 - YouTube](#)

https://www.youtube.com/watch?v=BLfNP4Sc_iA

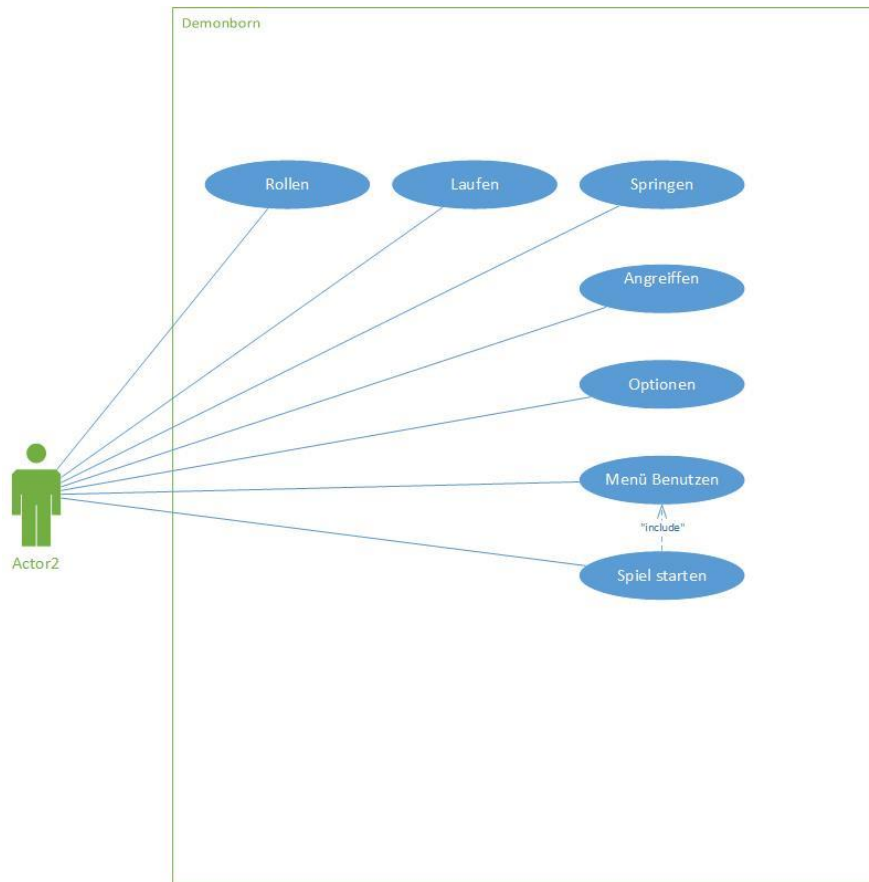
1.3 Anforderungen

Nummer	Muss / Kann?	Funktional? Qualität? Rand?	Beschreibung
1	Muss	Funktional	Der Spieler kann sich bewegen
2	Muss	Funktional	Der Spieler kann springen
3	Muss	Funktional	Der Spieler hat einen physischen Körper
4	Muss	Funktional	Der Spieler hat ein Schwert und kann damit angreifen.

5	Muss	Funktional	Der Spieler hat Schild und kann Angriffe blocken
6	Muss	Funktional	Der Spieler kann sterben
7	kann	Funktional	Der Spieler kann Parieren
8	Muss	Funktional	Der Spieler kann Rollen
9	Muss	Funktional	Der Spieler kann auf Objekten landen.
10	Muss	Funktional	Es gibt eine kleine Levelpassage
11	Muss	Funktional	Gegner sind im Level verstreut
12	Muss	Funktional	Die Gegner können den Spieler angreifen.
13	Muss	Funktional	Am Ende der Passage gibt es einen Bossfight
14	Kann	Funktional	Es gibt ein Leveling-System, welches abgespeichert wird.
15	Kann	Funktional	Wenn man Gegner tötet geben sie eine Währung
16	Kann	Qualität	Es gibt einen Shop, worin man Items kaufen kann
17	Kann	Funktional	Mit der Währung kann Level hochsteigen und Items kaufen
18	Kann	Qualität	Es gibt Checkpoints, in denen man den aktuellen Spielstand speichern kann
19	Kann	Qualität	Es gibt ein Pausenmenü
20	Muss	Funktional	Es gibt ein Hauptmenü
21	kann	Qualität	Es gibt ein Optionsmenü, in welchem der Spieler die Lautstärke einstellen kann
22	muss	Qualität	Der Spieler kann das Spiel bei jedem Checkpoint speichern

23	muss	Rand	Das Spiel wird mit Pixelart designt werden
24	muss	Rand	Ist in Unity programmiert
25	kann	Qualität	Das Spiel kann Easter Eggs beinhalten
26	kann	Funktional	Je nach Zeit, die wir noch haben, gibt es noch mehr Level.

1.4 Diagramme



1.5 Testfälle

[Erstellen Sie zu jeder Muss-Anforderung mindestens einen Testfall.]

Nummer	Voraussetzung	Eingabe	Erwartete Ausgabe
1.1	Spieler ist im Level	1. Tasten [A] [D] oder [Pfeil-Tasten]	2. Spieler bewegt sich
2.1	Spieler ist im Level	1. Der Spieler drückt [Space], [W] oder Pfeil nach oben	2. Der Spieler springt
3.1	Spieler ist im Level	1. Level starten	2. Schwerkraft zieht den Spieler nach unten und er fällt nicht durch den Boden
4.1	Spieler ist im Level Steht vor einem Gegner	1. drückt Maus-Links-Klick	2. Der Gegner bekommt Schaden
5.1	Spieler ist im Level Steht vor einem Gegner	2. Drückt Maus-Rechts-Klick	1. Der Gegner greift an 3. Spieler blockt die Attacke und er kriegt keinen physischen Schaden
5.2	Spieler ist im Level Steht vor einem Gegner Hält Maus-Rechts-Klick gedrückt	1. Drückt Maus-Links-Klick	2. Der Spieler macht keinen Angriff
6.1	Spieler ist im Level Steht vor einem Gegner	2. Der Spieler verliert Lebenspunkte	1. Der Gegner greift an 3. Der Spieler fängt wieder von seinem letzten Checkpoint an
6.2	Spieler ist im Level	1. Der Spieler fällt runter	2. Der Spieler die Hitbox des Levels und stirbt 3. Der Spieler fängt wieder von seinem letzten Checkpoint an
6.3	Spieler ist im Bossfight	2. Der Spieler verliert all seine Lebenspunkte	1. Der Gegner greift an

			3. Der Spieler fängt den Bossfight wieder von Anfang an
7.1	Spieler ist im Level Steht vor einem Gegner	2. Drückt Maus-Rechts-Klick im genauen Moment wo der Gegner den Spieler Trifft	1. Der Gegner greift an
8.1	Spieler ist im Level	1. Drückt Shift + Tasten [A] [D] oder [Pfeil-Tasten]	2. Der Spieler rollt in der ausgewählten Richtung
8.2	Spieler ist im Level Steht vor einem Gegner	2. Drückt Shift + Tasten [A] [D] oder [Pfeil-Tasten]	1. Der Gegner greift an 3. Spieler kriegt keinen Schaden, wenn er im richtigen Moment rollt
9.1	Spieler ist im Level	1. Der Spieler springt auf eine Plattform	2. Der Spieler fällt nicht durch
9.2	Spieler ist im Level	1. Der Spieler springt auf eine tödliche Plattform	2. Spieler stirbt
10.1	Im Spiel	1. Spieler startet Level	2. Spieler sieht Level und kann sich drin bewegen
11.1	Spieler ist im Level	1. Spieler bewegt im Level	2. Der Spieler findet Gegner und greift ihn an.
12.1	Spieler ist im Level Steht vor einem Gegner	1. Gegner greift an	2. Spieler kriegt Schaden
13.1	Spieler steht am Ende des Levels	1. Der Spieler beendet das Level	2. Eine neue Szene wird geladen und der Boss tritt auf
14.1	Spieler ist im Level Spieler ist bei einem Checkpoint	1. Spieler drückt auf "Level hochsteigen Knopf"	2. Spielt, steigt um einen Level

15.1	Spieler ist im Level Steht vor einem Gegner	1. Drückt Maus-Links-Klick und greift somit Gegner an	2. Gegner verliert all sein Leben, stirbt und lässt Geld fallen
16.1	Spieler ist im Level Spieler steht vor Händler	1. Drückt Maus-Links-Klick auf den "Kaufen" Knopf	2. Shop Fenster geht auf und man kann Sachen kaufen
17.1	Spieler ist im Level Spieler steht vor Händler oder Checkpoint	1. Spieler drückt auf "Level hochsteigen" oder "Kaufen" Knopf	2. Spieler gibt Währung aus um ein Level hochsteigen oder Item zu kaufen
18.1	Spieler ist im Level Spieler ist bei einem Checkpoint	1. Spieler drückt auf "Schlafen" Knopf	2. Das Spiel wird gespeichert, Level wird resetet und Spieler wird geheilt
19.1	Spieler ist im Level	1. Spieler drückt "Esc"	2. Spiel pausiert und ein kleines Menü wird angezeigt
20.1	Im Spiel	1. Spieler kann auf "Spiel Starten" oder "Optionen" Knopf drücken	2. Man startet das Level, oder man kann ein paar Einstellungen verändern
22.1	Spieler ist im Level Spieler ist bei einem Checkpoint	1. Spieler drückt auf "Schlafen" Knopf	2. Das Spiel wird gespeichert
23.1	Spieler hat das Spiel offen	1. keine	2. Pixelart
24.1	Spieler startet Spiel	1. keine	2. Unity-Ladebildschirm

2. Planen

Nummer	Frist	Beschreibung	Zeit (geplant)	Zeit (effektiv)
1.1	16.3	Spieler Bewegungsfunktion	20min	40min
2.1	16.3	Spieler Sprungfunktion	10min	30min
3.1	16.3	Spieler hat ein Rigidbody und Plattformen haben einen Boxcollider	2min	15min
4.1	16.3	Schwert dem Spieler hinzufügen	40 min	20min
5.1	16.3	Schild dem Spieler hinzufügen	40min	
5.2	16.3	Während Maus-Rechts-Klick gedrückt wird kann der Spieler nicht angreifen	20min	
6.1	16.3	Spieler kann von Gegnern Schaden sterben	30min	45min
6.2	16.3	Spieler stirbt, wenn er aus der Map fällt	20min	30min
6.3	23.3	Spieler stirbt von Boss Schaden	25min	40min
7.1	30.3	Spieler pariert	30min	
8.1	16.3	Spieler rollt	30min	
8.2	23.3	Spieler kriegt kein Schaden während er rollt	15min	20min
9.1	23.3	Spieler kann auf Plattformen stehen	20min	20min
9.2	23.3	Spieler stirbt, wenn er auf tödlichen Plattformen steht	30min	40min

10.1	23.3	Level ist da	15min	25min
11.1	23.3	Gegner sind im Level	30min	30min
12.1	30.3	Spieler kriegt Schaden von Gegnern	15min	
13.1	30.3	Ende des Levels und Start des Bossfights	20min	5min
14.1	30.3	Level hochsteigen	20min	
15.1	30.3	Gegner sterben und lassen Geld fallen	25min	
16.1	30.3	UI Für den Laden	40min	
17.1	30.3	Spieler kann Währung mit Level hochsteigen und an Items verschwenden	20min	
18.1	30.3	Checkpoint speichert und heilt	30min	
19.1	30.3	Pausenmenü	20min	
20.1	30.3	Hauptmenü	5min	5min
21.1	30.3	Optionsmenü, worin man Lautstärke veränder kann	30min	
22.1	30.3	Checkpoint speichert	15min	
25.1	30.3	Easter Eggs	20min	
26.1	30.3	Mehr Level	5-45min	

* Die Nummer hat das Format N.m, wobei N die Nummer der Anforderung ist, zu der das Arbeitspaket gehört, und m von 1 an fortlaufend durchnummeriert wird.

** Teilen Sie diesmal Ihre Anforderungen in 45-Minuten-Arbeitspakete ein

3. Entscheiden

[Dokumentieren Sie hier allfällige Entscheidungen, die Sie getroffen haben. Sonst lassen Sie dieses Kapitel leer.]

4. Realisieren

Das Projekt befindet sich auf github unter [fügen Sie hier die URL Ihres *repository* ein]

5. Kontrollieren

5.1 Testprotokoll

[Beschreiben Sie hier, welche Fehler Sie gefunden haben.]

Nummer	Datum	Resultat	Durchgeführt
1.1.1	27.4.2022	Funktioniert	Egloff
2.1.1	27.4.2022	Funktioniert	Egloff
3.1.1	27.4.2022	Funktioniert	Egloff
4.1.1	27.4.2022	Funktioniert	Egloff
5.1.1	27.4.2022	Funktioniert nicht	Egloff
6.1.1	27.4.2022	Funktioniert	Egloff

6.2.1	27.4.2022	Funktioniert nicht	Egloff
6.3.1	27.4.2022	Funktioniert	Egloff
7.1.1	27.4.2022	Funktioniert nicht	Egloff
8.1.1	27.4.2022	Funktioniert nicht	Egloff
8.2.1	27.4.2022	Funktioniert nicht	Egloff
9.1.1	27.4.2022	Funktioniert	Egloff
9.2.1	27.4.2022	Funktioniert nicht	Egloff
10.1.1	27.4.2022	Funktioniert	Egloff
11.1 1	27.4.2022	Funktioniert	Egloff
12.1.1	27.4.2022	Funktioniert	Egloff
13.1.1	27.4.2022	Funktioniert nicht	Egloff
14.1.1	27.4.2022	Funktioniert nicht	Egloff
15.1.1	27.4.2022	Funktioniert	Egloff
16.1.1	27.4.2022	Funktioniert nicht	Egloff
17.1.1	27.4.2022	Funktioniert nicht	Pinto
18.1.1	27.4.2022	Funktioniert nicht	Pinto
19.1.1	27.4.2022	Funktioniert	Egloff
20.1	27.4.2022	Funktioniert	Pinto

22.1	27.4.2022	Funktioniert nicht	Pinto
23.1	27.4.2022	Funktioniert	Pinto
24.1	27.4.2022	Funktioniert	Pinto

Aus dem Grund, dass unser Team nach kurzer Zeit halbiert wurde, konnten wir nicht alle Anforderungen umsetzen. Wir haben trotzdem ein kleines Spiel programmieren können.

6. Auswerten

Das Arbeiten von zu Hause aus lief erstaunlich gut und hat sehr gut zu diesem Projekt gepasst.

Da zwei unser Teammitglied in die Kanti gewechselt sind, konnten wir nicht alles schaffen.