



Bericht zur Entwicklung des 2D Games

“HAI ICH BIN EIN COOLES GAME”

Michelle Pötsch

Gruppe : “404 - Groupname not found”

Berlin, 02.02.022

Dozent: Ibrahim Khaled Reguieg

Inhaltsverzeichnis

Organisation im Team	2
Spielkonzept	2
Aufgabenbereiche	2
Startmenü	2
Pausenmenü	3
Leveldesign	4
Zusätzliche UIs & Bugfix	4
Literaturverzeichnis	5

Organisation im Team

Zu Beginn des Projekts haben wir als Gruppe entschlossen regelmäßig Sonntags ein Meeting zu halten. Dort haben wir wöchentlich Aufgaben verteilt und unsere Arbeitsstände besprochen. Kommuniziert haben wir hauptsächlich über Discord. Für Meeting Notizen, Brainstorming und mehr, haben wir Miro gewählt. Unseren Code haben wir auf GitHub in einem Repository abgelegt. Ich habe mir einen eigenen Branch erstellt und diesen später gemerged.

Spielkonzept

Unser 2D-Spiel sollte ein Top-Down Pixel adventure sein. Wir entschieden uns für das Steuern eines Charakters in verschiedenen Leveln in denen verschiedene, teilweise versteckte, Gegner sind. Das Ziel sollte sein alle Gegenstände zu sammeln bevor die Zeit abgelaufen ist.

Aufgabenbereiche

Da ich gerne kreativ bin habe ich, nach Absprache, meinen Fokus der Aufgaben auf das UIs gelegt. Meine zwei großen Aufgabenbereiche umfassten das Design und die Entwicklung des Start- sowie Pausenmenüs. Im Verlauf des Projekts überschnitten sich einige Bereiche mit Aufgaben anderer Teammitglieder welches auch zu anderen Tätigkeiten führte. Als es zum Design der Level kam habe ich einige Design hierfür erstellt. Außerdem war eine begleitende Tätigkeit das andauernde Testen um Fehler zu finden und diese Frühzeitig korrigieren zu können. Zu allerletzt habe ich die Präsentation für die Vorstellung des Spiels vorbereitet. Nun gehe ich genauer auf einzelnen Aufgabenbereiche ein.

Startmenü

Beim Starten unseres Spiels gelangt der User zuerst auf das Startmenü welches meine Aufgabe zu Entwickeln war.

Anfangs habe ich einige Skizzen angefertigt und Brainstorming betrieben um Ideen für das Design zu sammeln. Folgedessen habe ich benötigte Button und Untermenüs zusammengestellt. Danach habe ich diese groben Ideen mit den anderen Teammitgliedern besprochen und angepasst. Im nächsten Schritt habe ich eine neue Szene namens "StartMenu" in unserem Unity Projekt erstellt. Hier habe ich ein Canvas erstellt das aus einem Hintergrund und mehreren Canvases für die jeweiligen Untermenüs

erstellt. Nachdem einige Button erstellt waren habe ich die Funktionalität mit C# in einem "MenuManager" Script programmiert. Beim zuweisen der GameObjects habe ich mit Tags gearbeitet. Die ineinander verschachtelte Hierarchie hat andernfalls zu Problemen in der Suche der Referenzierungen geführt. Durch das de- und aktivieren der UIs konnte ich die Funktionalität recht einfach umsetzen. Nachdem das Menü einwandfrei funktionierte habe ich nach Assets für das UI gesucht und davon eingebaut. Mit Photoshop konnte ich einige nachbearbeiten und für uns anpassen. Mit verschiedenen Fonts, Images und Farben ist das folgend zu sehende Design entstanden.



Design: Startmenü, Help Untermenü, Settings Untermenü

Hierbei habe ich darauf geachtet das 2D-Pixel Design und die bereits bestehenden Farben zu verwenden. So ist ein einfaches minimalistisches Menü entstanden.

Pausenmenü

In der 3 Woche habe ich mich dann an das Pausemenü des Spiels gesetzt. Um das Design konstant zu halten habe ich das Canvas des Startmenüs kopiert und hier als Overlay eingefügt. Da das Menü bis auf den "Resume" Button, um das Spiel fortzufahren, gleich blieb, schwankte ich zwischen einem zweiten Script das auf Vererbung zurückgreift oder einem instanzieren des Startmenüs im Pausenmenü. Nach einigen Tests und Refactoring Versuchen ist hier das Problem der Musikeinstellung aufgetaucht. Beim Ausschalten der Musik im Startmenü wurde dieser "Mute" Zustand nicht an die Game Scene weitergeleitet. Dies konnte ich dann durch das zugreifen der "musicIsOn" variable fixen und so konnten beide Klassen miteinander Sprechen. Beim späteren Betrachten ist mir aufgefallen das einfache Getter und Setter ebenfalls eine gute Lösung gewesen wäre.

Leveldesign

Nachdem wir im Team alle wichtigen Funktionalitäten gemerged hatten haben wir uns die Zeit genommen die Levels zu designen. Ich habe die folgend zu sehenden Welten Ice, Farm und Darkworld designt.



Level Design: Ice - Farm - Dark World

Ein anderes Teammitglied hat hierbei die gewählten Assets Vorlagen in das Projekt integriert mit den wir die Welten aufbauen konnten. Aus großen Bildern die eine Sammlung vieler Objekte war konnte nun mit dem umranden eines Teils des Bildes einzelne Objekte gewählt und als Tile auf das Level positioniert werden. Durch das Nutzen mehrerer Ebenen hierbei, wie Beispielsweise Boden oder Wasser, konnte sichergestellt werden wie das Verhalten des Spielers ist. Zum Schluss haben wir alle Level zusammen als eine große Map zusammen- gefügt.

Zusätzliche UIs & Bugfix

Zwei Wochen vor Abgabe haben wir alles gemerged und ich habe mich intensiv an Testen und Fehlersuche gesetzt. Alles das ich fand notierte ich in den Discord-Chat dass jeder immer aktuell ist. In einem unserer letzten Gruppen Gespräche ist uns aufgefallen dass es noch kein Feedback an den User gibt sobald dieser gestorben oder gewonnen hat. Demzufolge habe ich mich an die UIs für beide dieser Fälle gesetzt. Hierbei habe ich mich erstmals stark in die Codes der anderen Gruppenmitglieder einlesen müssen. Da es zwei Gründe für einen Tod gibt (Zeit abgelaufen und alle Leben verloren) musste ich hier die richtigen Stellen im Code finden um den jeweils richtigen Grund anzeigen lassen zu können. Die Schrift des "You Died" UIs habe ich in Rot gewählt und die bei einem gewonnenen Spiel dagegen in Grün.



UI Design: Verloren, Gewonnen

In den Tagen vor der Abgabe habe ich mich mit der gesamten Gruppe zusammengesetzt um alle restlichen Merges zusammen durchzuführen oder Fehler zu beheben. Anschließend habe ich die Powerpoint Präsentation für das Team vorbereitet und Screenshots des Spiels eingefügt. Zum Schluss habe ich mit dem Builden für MacOS beschäftigt indem ich ihn gebuildet und getestet habe.



coolesGame.app

Literaturverzeichnis

1. Font Assets

<https://www.dafont.com/de/bitmap.php>

2. Button Assets

<https://assetstore.unity.com/packages/2d/gui/puzzle-stage-settings-gui-pack-147389>

3. Hintergrundbild

<https://www.istockphoto.com/de/vektor/pixel-art-spiel-design-in-8-bit-stil-charakter-gm1196370362-341265463>

4. Powerpoint Präsentation

<https://docs.google.com/presentation>