

Rollenspiel „Village Defender" (Project C)



Team: T. Lamping, I. Kraft, M. Bösling

Inhaltsverzeichnis

- Grundidee
- Konzept
- Umsetzung und verwendete Software
- Playthrough
- Grafiken
- Spielwelt
- Spielablauf
- Scripte
- Fazit
- Zeitaufwand

Grundidee

Unsere Grundidee war die Entwicklung eines klassischen Rollenspiels im Stil alter Zelda-Spiele. Ebenso wollten wir alle Grafiken selbst erstellen, um auch in diesem Bereich Erfahrungen zu sammeln.

Die Motivation im Rollenspiel liegt unter anderem im Erkunden und Sammeln von Erfahrung und Gegenständen. Dies bedeutet einen großen Aufwand im Bereich des Grafikdesigns. Um diesen Aufwand etwas geringer zu halten und Spieler trotzdem zu motivieren, mussten wir uns noch mit weiteren Motivationskonzepten auseinandersetzen.

Als Hauptmotivation sollte uns „die Verhinderung eines andern Zieles“ dienen. Das bedeutete in unserem Fall, dass der Endboss das Dorf überfallen will und wir ihn aufhalten müssen. Damit sollen die Spieler dazu bewegt werden, dass Spiel bis zum Schluss zu spielen, denn ohne sie wird das Dorf vernichtet.

Das Dorf fungiert als Basis, von der aus Missionen gestartet werden. Diese Missionen führen den Spieler zu unterschiedlichen Herausforderungen, wie z.B. dem Lösen von Rätseln.

Konzept

Genre: Rollenspiel

Darstellung: 2D, Top-Down-Ansicht

Zielgruppe: Gelegenheitsspieler

Spielwelt: Fantasy, frühes Mittelalter

Gameplay / Motivation: Erkundung, Kampf, Dorf retten

- Grundkonzept: Beschütze Bewohner des Dorfes vor dem Plünderer.
- Am Ende gibt es einen Bosskampf.
- Der Bosskampf wird leichter, je mehr Spieler sich darauf vorbereiten (Schwertquest).
- Vorbereiten erfolgt durch Sammeln von Gegenständen (Schwert sammeln).
- Es gibt Missionen in verschiedenen Welten (Schwertinsel).

Umsetzung und verwendete Software

Es gab Überlegungen auf den *RPG-Maker* zurückzugreifen, oder in *Unity* mithilfe von C# selber zu programmieren. Wir haben uns für den Kompromiss zwischen Arbeitsaufwand und Freiheit der Gestaltung für *Unity* entschieden. Des Weiteren haben wir zur Erstellung der Grafiken *Inkscape* benutzt. Zur Versionskontrolle haben wir *Github* verwendet.

Hier einmal eine vollständige Auflistung der verwendeten Software:

- Adobe Photoshop
- Gimp
- Inkscape
- Unity
- Github
- Discord
- Spriter
- Audacity
- Tiled
- Tiles2Unity
- Scribbus
- Magix Music Maker

Playthrough

Hier ein kompletter Playthrough des Spieles.

Grafiken

Ein Großteil der Arbeit entfiel auf die Erstellung der Grafiken. Mit *Inkscape*, einem Programm zur Erstellung von Vektorgrafiken, wurde versucht, Figuren nach Inspirationen aus dem Internet oder anderen Computerspielen zu erstellen. Ein Vorteil der Vektorgrafik war, dass wir uns nicht auf eine spezielle Auflösung bei der Erstellung festlegen mussten.

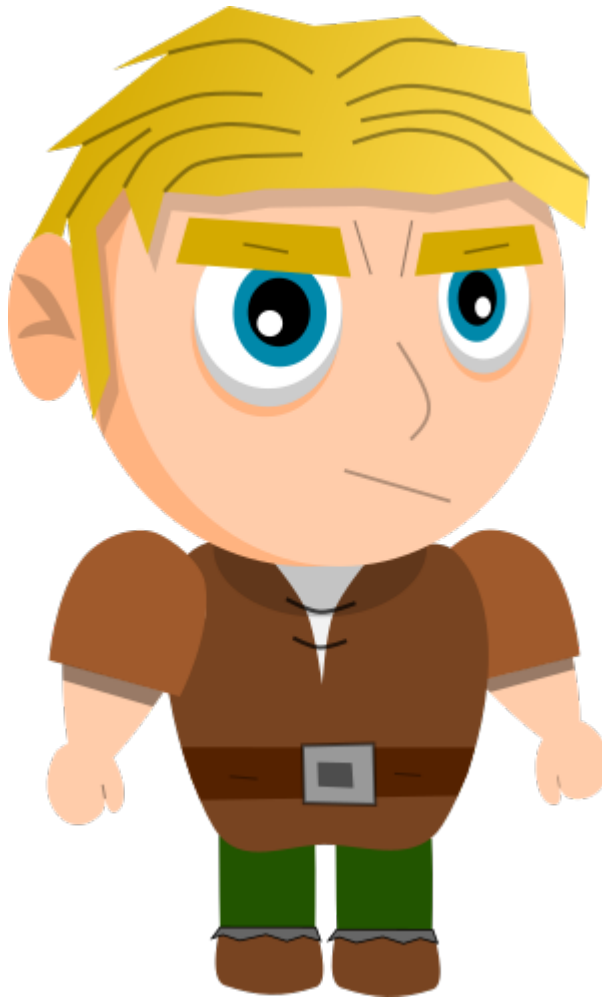
Es wurden 10 animierte Figuren erstellt. Diese hatten jeweils die Bewegungs- und Idle- (Leerlauf-)Animationen. Die Hauptfigur bekam noch 3 weitere Animationen mit unterschiedlichen Waffen: die Axt zum Baumfällen, das Holzsword als Anfangswaffe und das Stahlsword.

Die 12er-Sprite-Sheets wurden dabei direkt in *Inkscape* erstellt und die Größeren mit *Spriter* als Bone-Animation umgesetzt.

Des Weiteren hatten wir beschlossen, uns alle einmal an den Grafiken zu versuchen, was an den leicht unterschiedlichen Stilen zu erkennen ist. Im Prototyp sahen wir dies als vertretbar an. Für ein fertiges Endprodukt würden wir uns genauer an einen Stil halten.

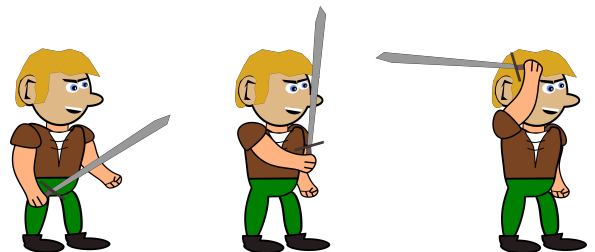
Im Folgenden einige Beispiele.

Der Protagonist von Village Defender.



Für die Animation mussten für alle 4 Richtungen Sprite Sheets erstellt werden. Genauso für den Wechsel der Ausrüstung. Hier hat sich der Vorteil von **Spriter** als Bone-Animation-Programm gezeigt, da man hier Gegenstände leichter ersetzen kann und die selbe Animation beibehält.

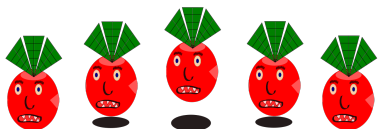
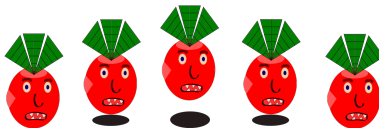
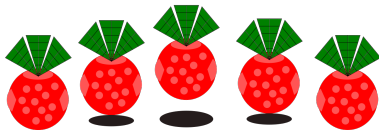
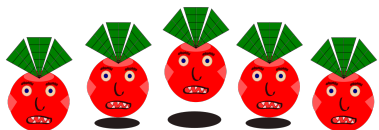
Zu Beginn des Projektes sah der Held noch etwas anders aus – die Farben aber sind geblieben.





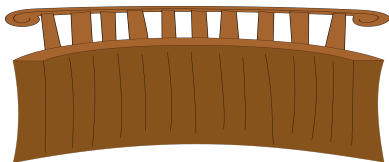
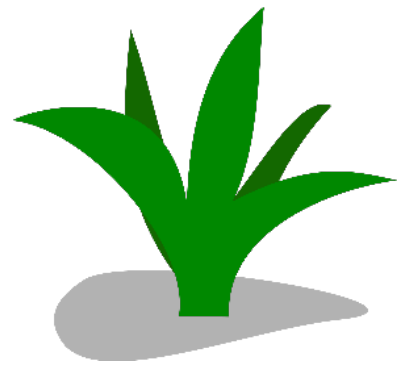
Die Dorfbewohner und Gegner

Die Dorfbewohner und Gegner haben weniger komplexe Sprite Sheets für ihre Animation bekommen, da sie nicht ständig im Fokus sind.



Sonstige Grafiken

Hier einige weitere Beispiele für von uns erstellte Grafiken.



Die Spielwelt

Neben der Erstellung der Figuren mussten noch weitere Grafiken für die Welt erstellt werden. Dabei wurde der Boden als sogenannte *Tilemap* erstellt, wofür ein *Tilesset* gebraucht wurde. Die anderen Objekte wie Häuser, Pflanzen, Schilder, Brunnen und Brücke wurden als Objekte eingesetzt. Insgesamt kommt man bei den Objekten auf etwa 50 Grafiken und ca. 10 Tilsets für Land, Wasser, Wüste, Steine usw.

Weiterhin gibt es kleinere Animationen für das Baumfällen, das Öffnen des Tores auf der Schwertinsel und für den Springbrunnen auf dem Marktplatz.

Das Dorf, in dem sich der Spieler frei bewegen kann.

Eine der 3 Hauptzonen. Hier auf dem Marktplatz findet der Endkampf statt.

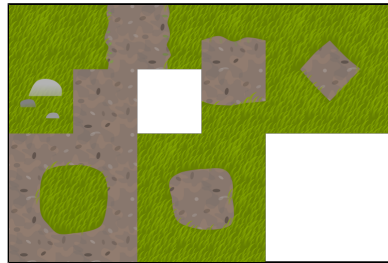
Das Waldgebiet um für die erste Aufgabe, Holz zu sammeln. Ein Ork patrouilliert und Gegner verstecken sich hinter den Bäumen.



Die Schwertinsel mit dem Rätsel.



Ein Tileset von 128 x 128 Pixeln für Tiled erstellt, um die Karte zu generieren. Jeder Grashalm ist dabei als Vektor gespeichert, was das System schon einmal auslasten kann, wenn man es nicht als png exportiert.

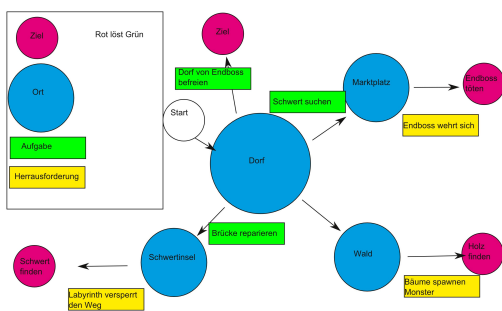


Ein Haus mit seinen Einzelteilen.



Der Spielablauf

Hier die schematische Darstellung des Spielablaufes. Um das Ziel (rot) zu erreichen, müssen wir erst alle Aufgaben (grün) lösen.



Am Anfang findet der Spieler sich im Dorf wieder. Dort erfährt er, dass er ein Schwert finden muss, um den Endboss besiegen zu können. Dieses befindet sich auf einer Insel, welche nur mit einer Brücke erreicht werden kann. Um die Brücke zu reparieren, wird Holz benötigt.

So ist die erste Aufgabe des Spielers, durch Baumfällen genügend Holz im Wald zu sammeln. Dabei stellen sich ihm aber Gegner in dem Weg, die er mit seinem Holzschwert bekämpfen kann.

Ist die Brücke repariert, gelangt der Spieler auf die Insel, wo das Schwert zu finden ist. Das Schwert wiederum befindet sich hinter einem Tor, welches nur mit einem Juwel geöffnet werden kann.

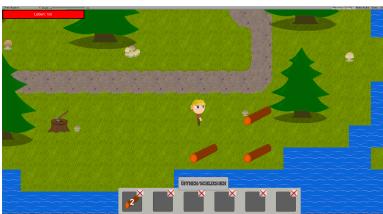
Um an das Juwel zu kommen, muss ein Rätsel gelöst werden. Dabei geht es darum, Steine zu verschieben und den richtigen Weg (Labyrinth) zum Juwel zu finden.

Sobald der Spieler das Schwert hat, geht es zurück zum Dorf, wo der Endboss auf dem Marktplatz wartet. Dieser muss nun besiegt werden, um das Spiel erfolgreich zu beenden.

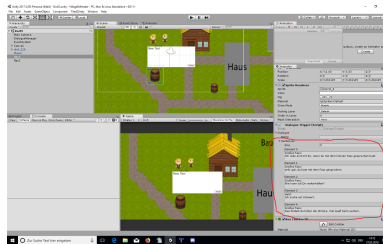
Scripte

Der andere Großteil der Arbeit war die Erstellung der Scripte. Wir haben insgesamt 35 Scripte geschrieben, welche jeweils eine bestimmte Funktion erfüllen. Darunter fallen z.B. das Inventar- oder Dialog-System und ein Trigger, welcher z.B. den Endboss unter bestimmten Bedingungen erscheinen lässt.

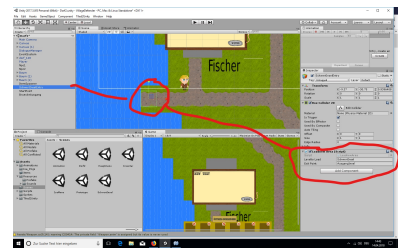
Das **Inventar-System** dient der Anzeige für Questaufgaben, wie z.B. dem Sammeln von Holz. Das Inventar kann man öffnen und schließen. Für zukünftige Versionen hat das Inventar größeren Platz bekommen, als es momentan im Spiel benötigt. Auch der x-Button ist noch nicht implementiert und dient nur als Platzhalter für spätere Funktionen, um das Inventar ggf. zu leeren.



Mit dem **Dialog-System** können jedem NPC (Non Player Character) eine Anzahl an Sätzen zugeordnet werden. Je nach Quest-Fortschritt werden unterschiedliche Antworten aus den gespeicherten Sätzen gegeben.



Ein **Teleportations-Script**, um zwischen verschiedenen Ein- und Ausgängen zu springen. Dies funktioniert zwischen zwei verschiedenen Szenen und auch innerhalb einer Szene.



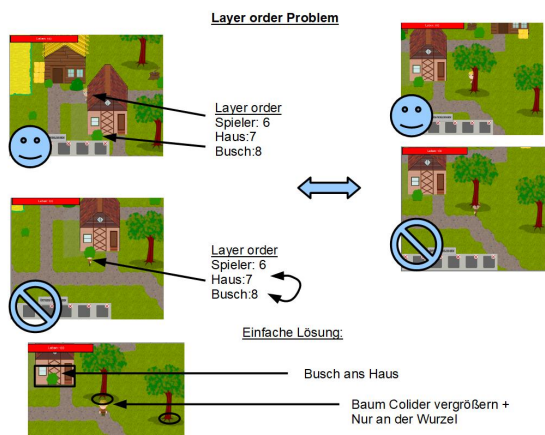
Fazit

Schwierigkeiten gab es bei der Größe des Projektes. Die Anzahl an Grafiken, Animationen und Scripten wuchs während der Entwicklung immer schneller an und bald war klar, dass wir unsere Vorstellungen nicht so umsetzen konnten, wie es geplant war.

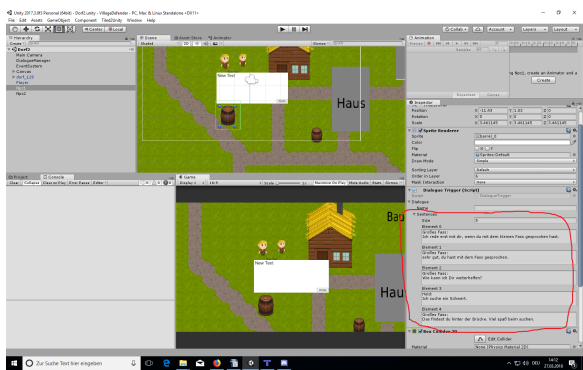
Wir haben uns deshalb daran orientiert einen Prototyp zu entwerfen, der in sich abgeschlossen sein soll. Für Unity gibt es viele Scriptbeispiele und Tutorials, doch mussten diese erst erlernt und auf unser Projekt angewendet werden.

Durch den Verzicht auf fertige Scripte und Grafiken konnten wir viele Erfahrungen sammeln, da alles selbst erstellt werden musste. Deshalb lag der Fokus eher auf dem Weg der Erstellung und nicht auf einem fertigen Produkt.

Im Folgenden zwei Beispiele von Problemen, die während des Projektes aufgetreten sind.



Probleme mit dem Dialog-System



Das Problem mit der *Layer Order* führte dazu, dass Objekte in falscher Reihenfolge dargestellt werden. Wenn der Spieler z.B. um einen Baum läuft, ist er mal vor dem Baum und dann dahinter. Hier haben wir uns für eine einfache Lösung entschieden, so dass der Spieler gar nicht mehr vor dem Baum gehen kann. Dazu wurde der Collider des Baums vergrößert.

Das Dialogsystem war eine größere Herausforderung als erwartet. Die erste Überlegung war es, eine XML-Datenbank zu erstellen. Es stellte sich aber heraus, dass für den Prototyp ein komplexer Baumstruktur mit Rede- und Antwortsystem zu aufwändig ist. Außerdem gibt es im Prototyp noch keine Situation, in der wir so ein System brauchen. Deshalb haben wir uns für eine einfachere Version entschieden und die Texte dem jeweiligen Objekt einzeln im Script zugewiesen.

So ist es möglich, dass je nach Weltereignissen (z.B. Quest-Fortschritt) die NPC unterschiedliche Antworten parat haben.

Zeitaufwand

Folgend eine Übersicht über die aufgewendeten Stunden für den jeweiligen Bereich.

Person	Themen	Gesamtzeit in Stunden
M. Bösling	Teambesprechungen und Organisation	20 + 20
	Grafik und Design	80
	Animationen	20

	Scripte	150
	Sound	10
I. Kraft	Teambesprechungen und Organisation	30
	Grafik und Design	180
	• Einarbeitung • Bewohner • Tiles • Häuser und Marktplatz • Natur • Nachträg. Änderungen	• 30 • 35 • 20 • 55 • 30 • 10
	Animationen	55
	Scripte	30
	Sound	5
T. Lamping	Teambesprechungen und Organisation	30
	Grafik und Design	30
	Animationen	10
	Scripte	200
	• Bugfixes • Schieberätsel • Gegner KI • Inventar • Kampfsystem • Quest für Schwertinsel • Erster Prototyp • Bäume fällen + Holz spawn	• 70 • 30 • 25 • 20 • 20 • 15 • 15 • 5
	Sound	30

