Maturitätsarbeit 2023
Documentation

Laneclash

2022

Version: 1.0
Date: 2022-10-18 19:06:03+02:00

Team: Marc Honegger (C6a)
Elia Schürpf (C6a)

Begleitperson: Martin Hunziker

Gymnasium
KZO - Kantonsschule Zürcher Oberland

Inhaltsverzeichnis

Teil I

Ziel

Kapitel 1

Vision

Ein funktionsfähiges Videospiel, welches jede vergleichbare Maturitätsarbeit in den Schatten stellt. Die Ideen sind nicht neu, sondern waren schon vor Jahren in Marcs Kopf. Jedoch war für die Umsetzung ein Team notwendig. Uns war dennoch von Anfang an bewusst, dass wir auch zu zweit nicht alle Funktionen hinkriegen werden, dennoch haben wir uns ein hohes Ziel gesetzt und für die Betreuung uns für Martin Hunziker entschieden. Vieles konnten wir schlussendlich auch umsetzen. Das grösste Wagnis war der Multiplayer und das haben wir dann auch an eigenem Leib erfahren. Alleine für den Multiplayer wurden über 100h investiert. Alles in allem kam ein funktionsfähiges und lustiges Spiel heraus. Einige Kommentare von Testern:

'sick game' - Marc Honegger
'absolut krass' - Elia Schürpf
'Masterpiece' - Martin Hunziker

Wie das Spiel jetzt aussieht, wie es zu diesem Punkt kommen konnte, aber auch wie es in Zukunft damit weiter geht, ist in dieser Arbeit beschrieben.

Kapitel 2

Anforderungen

2.1 KZO

Die Anforderungen der Schule halten sich in Grenzen. Zum einen gibt es fast keine genauen Anforderungen und die Anforderungen, die es gibt, sind relativ tief gehalten. Die drei Anforderungen der KZO sind:

1. **Zeit**
   Geplant ist ungefähr eine Lektion pro Woche, denn diese wir dem Stundenplan für das Semester abgezogen. Jedoch ist diese Anforderung bei uns nicht relevant, denn unser Projekt ist eine riesige Maturitätsarbeit. Wir werden ohne Probleme diese Zeitanforderung erreichen und auch bei Weitem überschreiten.

2. **Begleitperson**
   Jede Maturitätsarbeit braucht eine Begleitperson. Also eine Lehrperson, welche die Arbeit bewertet und einen unterstützt. Letzteres, also Unterstützung, war bei uns sehr schwierig, denn keine Lehrperson kennt sich mit Unity und C# genug gut aus. Wir sind bereits früh auf Herrn Kern, ein Informatiklehrer, zugegangen. Er hat uns an Martin Hunziker, Leiter der IT, weitergeleitet. Dieser nahm uns sehr gerne auf, mit der Anmerkung, dass unsere Arbeit sehr schwierig sein wird, umzusetzen.

3. **Plagiat**
   Arbeiten dürfen nicht einfach kopiert werden. Wir haben uns zwar bei gewissen Problemen online inspirieren lassen, jedoch sind diese keine Plagiate. Der grösste Teil unseres Codes und alles unserer schriftlichen Arbeit sind von uns geschrieben.

2.2 Unsere eigenen Anforderungen

Gewisse Features sind für das Funktionieren des Spiels wichtiger als andere. Zum Beispiel sind Einstellungen, Truppen und Multiplayer notwendiger als die Monetarisierung, zumindest nach unserer Ansicht. Deshalb ist dieser Abschnitt in 4 Gruppen unterteilt:

1. **Notwendig:** Ohne diese Features läuft das Spiel nicht oder ist sehr mangelhaft, voller Fehler und unspielbar. Ohne diese Features ist Idee des Spieles nicht erkennbar.

2. **Anvisiert:** Diese Features sind unser Ziel. Sie wurden Herr Hunziker, also der Begleitperson, angekündigt und sind alle in einem Plan festgehalten. Diese Ziele sollen dazu führen, dass das Spiel gut spielbar ist und es Spass macht. Sie sind auch dafür essenziell, das Spiel nicht zu monoton zu gestalten und sollten alle bis zum Abgabetermin der schriftlichen Arbeit vollendet sein.

3. **Möglichkeiten:** Möglichkeiten das Spiel noch auf die Spitze zu bringen. Keine dieser Anforderungen ist nötig für das Spielen, jedoch können sie das Spielerlebnis spannender und ausgereifter machen. Diese Ziele sind entweder in den Sternen geschrieben oder zu erreichen bis zur mündlichen Präsentation. Vereinzelt können diese, wie die anvisierten Anforderungen, noch bis zur schriftlichen Abgabe erreicht werden.

4. **Verworfen:** Weitere Ideen, welche wir hatten, jedoch als unmöglich oder nicht sinnvoll erachten.

Notwendig

- **System**
  Unser Spiel sollte auf den meisten modernen Geräten funktionieren. Es soll auf macOS und Windows fliessend gespielt werden können. Also mit mindestens 60FPS.

- **Benutzeroberfläche**
  Eine Startseite mit einem "Spielen"-Knopf, einem "Einstellungen"-Knopf, einem "Credits"-Knopf und einem "Verlassen"-Knopf. Sie soll intuitiv sein und einigermassen schön aussehen. Die Benutzeroberfläche soll mit Touch bedienbar sein, andere Eingabemöglichkeiten werden nicht unterstützt.

- **Lokaler Multiplayer**
  Das Spiel soll im lokalen Netz gespielt werden können. So zum Beispiel mit Freunden oder Familie. Es sollte mithilfe der IP-Adresse innerhalb des gleichen Netzes zusammengespielt werden können.

- **Einstellungen**
  Auflösung, Vollbild und Lautstärke müssen eingestellt werden können.

- **Kamera**
  Die Kamera ist beweglich und das ganze Schlachtfeld ist sichtbar.

- **Truppen**
  Es braucht Truppen, die für einen kämpfen können. Diese sind notwendig für den Verlauf des Spieles und ohne sie hat das Spiel keinen Sinn.

- **Gewinnmöglichkeit**
  Eine Chance das Spiel zu gewinnen oder verlieren und es somit zu beenden. Dies kann sehr simpel mit einer Linie vollendet werden, welche beim Überschreiten das Spiel beendet.
- **Lanes**
  Unser Spiel ist zwar 2.5D, aber hat dennoch eine Tiefe.
- **Synchronisation**
  Unser Spiel soll in Echtzeit spielbar sein, deswegen müssen die beiden Spieler Synchronisiert das Gleiche sehen. Eine De-Synchronisation wäre verheerend.
- **Crossplay** Die Möglichkeit sich mit einem Windows Rechner und einem macOS Rechner zu verbinden und zusammenzuspielen.

Anvisiert

- **Helden**
  Beide Spieler haben eine Art Truppe, welche ihre Siegeslinie beschützt. Sie haben einen Racheeffekt, welcher die Lane ausradiert, damit das Spiel nicht sofort vorbei ist.
- **Design**
  Das Spiel sollte vom Aussehen her was hergeben. Es sollte nicht wie ein Prototyp, sondern wie ein vollendetes Spiel aussehen.
- **Deck**
  Zu Beginn Spieler könne ihr eigenes Deck aus einer Auswahl von Karten zusammenstellen. Im Spiel werden davon per Zufall Karten gezogen.
- **Bot**
  Ein sehr simpler Algorithmus um alleine gegen den Computer zu spielen auf den Schwierigkeitsstufen *'Einfach', 'Mittel', 'Schwierig'*. Es sollen rein zufällig Truppen geschickt werden, bei höherer Schwierigkeit mehr.
- **Truppen**
  - **Nahkampf** Eine Truppe mit wenig Reichweite.
  - **Fernkampf** Eine Truppe die auf Reichweite angreift. Sie scheisst Projektive, zum Beispiel ein Bogenschütze, der mit Pfeilen schiesst.
  - **Suizid** Eine Truppe die bei Berührung mit einem Gegner stirbt und einen Effekt auslöst.
- **Effekte**
  - **Gift:** Zeitlich limitierter und wiederholender Schadenseffekt. Eine visuelle Markierung soll vorhanden sein und das gleiche Gift, also der gleichen Truppe, soll nur einmal auf jemanden sein.

- **Dornen:** Truppen, welche Charaktere mit Dornen attackieren, erhalten schaden. Soll nach Auswahl auf Nahkämpfer, Fernkämpfer oder beides wirken.
- **Rache:** Truppen mit Rache haben einen Effekt nach ihrem Tod. Zum Beispiel eine Truppe beschwören oder Schaden verursachen.

Möglichkeiten

- **Zauber**
  Als Alternative sollen nicht alle Karten einfach eine Truppe herbeirufen. Gewisse Karten sollten nur einen Effekt auslösen. Beispiele für Zauber: "Heile deine Helden um 5 lebenGib deinen Helden 5 RüstungVerursache allen gegnerischen Truppen 5 SchadenZiehe 3 Karten"

- **Monetarisierung**
  Hierfür sind uns drei Möglichkeiten eingefallen, hier jeweils die Vor- und Nachteile:

  1. **Kaufbare Gegenstände**
     + Vielseitige und Nachhaltige Monetarisierung. Neue Features, bedeuten auch neue Geldmöglichkeit. Die Inhalte sollten auch erspielbar sein, somit kann bezahlen zwar zu Vorteilen führen, jedoch mit viel Spielen trotzdem erreichbar sein.
     - Pay-to-Win
  2. **Skins**
     + Weit verbreitet und sehr beliebt bei Spielern. Gibt keinem Spieler Vorteile
     - Wenig Nutzen und sehr aufwändig. Neue Designs zu erstellen, wäre bei uns nicht so sinnvoll. Wir haben noch sehr viel Features zu implementieren und sind nicht Designer. Für uns braucht es sehr viel Zeit eine brauchbare Darstellung zu erstellen und der Mehrwert, welcher dem Spiel beigetragen wird, ist marginal.
  3. **Kostenpflichtiges Spiel**
     + Einmalige Bezahlung, was für viele Nutzer lukrativer ist. Jedoch gilt dies vor allem bei Singleplayer Spielen.
     - Wir nehmen an, dass wenige Leute bereit wären, Geld für unser Spiel zu bezahlen. Dafür wird es nicht genug ausgereift sein. Auch ist diese Methodik nicht nachhaltig und führt nur zu einer einmaligen Geldspritze. Viele grössere Spiele führen deshalb später DLCs ein, um das Spiel zu erweitern. Jedoch ist dies bei einem Multiplayer Spiel Pay-to-Win. Abschliessen müssten wir höchstwahrscheinlich selbst Geld vorauswerfen, um unser Spiel anbieten zu können, z.B. auf Steam.

- **Helden**
  Eine Auswahl von Helden, mit unterschiedlichem Schaden, Leben und Effekten, würde das Spiel nochmals spannender machen. Auch sollten gewisse Truppen auf bestimmte Helden limitiert sein.

- **Design**
  Wenn die Zeit vorhanden ist, kann das Design immer verbessert werden. Hier gehts es aber um den Feinschliff, z.B. mehr Hintergründe, und mehr Dinge selbst zu designen.

- **Tutorial**
  Eine Anleitung für Anfänger wäre sehr schön. Sie soll neuen Spielern das Anfangen erleichtern. Es sollte aber auch überspringbar sein, damit alte Spieler, es nicht nochmals spielen müssen. Es soll nicht lange sein und nicht schwierig. Dennoch soll es alle Mechaniken des Spiels erklären, am besten Anhand von Gameplay.

- **Truppen** Weitere Truppen können jederzeit erstellt werden. Hier ist kein Limit gesetzt. Leben, Schaden, Darstellung, Effekt und vieles mehr kann angepasst werden.

- Effekte
  - **Wiederbelebung:** Die Truppe wird wiederbelebt, sofort an Ort und Stelle oder mit einer Verzögerung am Startpunkt.
  - **Rüstung:** Die Truppe hat zusätzlichen zu den Leben auch Rüstung. Die Rüstung wird zuerst abgezogen und hat im Gegensatz zum Leben keine Limite.
  - **Aura:** Die Truppe fügt gegnerischen Truppen in einem gewissen Radius permanent Schaden zu.

- **Mehrsprachig**
  Das Spiel soll in Englisch, Deutsch und Französisch spielbar sein.

- **Ingame-Chatfenster**
  Ein Textfeld, in dem man sich mit dem Gegner unterhalten kann.

Bereits früh verworfen

- **Online Multiplayer**
  Besser als ein Multiplayer, der limitiert auf dasselbe Netz ist, ist ein Multiplayer, der weltweit verfügbar ist. Jedoch ist von einem Computer auf einen anderen zu verbinden dank der Firewall von Router und Computer, nahezu unmöglich. Es würden sich viele weitere Probleme ergeben, wie zum Beispiel Port-Forwarding. Dies ist einer der Gründe, weshalb ein Server sehr praktisch ist. Mit diesem wäre dieses Problem gelöst. Server sind aber teuer und aufwändig.

- **Shop**
  Nicht alle Karten sind von Beginn an freigeben, sondern müssen freigespielt werden. So kann zum Beispiel eine Ingame Währung erspielt werden und damit im Shop Karten oder sonstige Dinge gekauft werden. Der Shop kann mit Zahlungsmethoden ausgestattet werden und so zur eventuellen Monetarisierung beitragen.

- **Kampagne**
  Singleplayer gegen bestimmte vorprogrammierte Gegner. Sie sollen immer schwieriger werden und bestimmte Herausforderungen mit sich bringen. Bei Vollendung werden Belohnungen verteilt.

- **Anti-Cheat**
  Eine sehr komplexe Software um das Schummeln zu verhindern. Ist in allen grösseren Spielen vorhanden. Teilweise sogar auf Systemebene, oder sogar auf Kernel-Level, gespeichert und ausgeführt.

- **Errungenschaften**
  Eine Übersicht und die Möglichkeit die Errungenschaften zu erfüllen. Gegebenenfalls Belohnungen verteilen, wie zum Beispiel bestimmte Karten freischalten. Diese Erweiterung ist keines Falls notwendig und ein absolutes nice-to-have Feature.

Kapitel 3

Skizzen

nöd numme skizze bilder sondern villicht au churzi notize dezue wie mir es eus am afang vorgstellt hend

Kapitel 4

Risiko

Wir haben die Risiken eingeteilt in verschiedene Gefahrenstufen. Die Einstufung geschah basierend auf Wahrscheinlichkeit, wie schlimm ist der Worstcase und wie gut kann das Risiko reduziert werden. Jedes Risiko ist dann unterteilt in:

1. Risiko (Wahrscheinlichkeit und Schaden)
   (a) Reduktionsstrategie: Strategie um die Wahrscheinlichkeit und/oder Schaden zu reduzieren.
   (b) Reduziertes Risiko: Risiko nach der Umsetzung der Reduktionsstrategie.
   (c) Tatsächliches Outcome: Das tatsächliche Endresultat

4.1 Hoch

1. Vergange Maturitätsarbeiten, welche ein Videospiel als Ziel hatten, sind gescheitert (Wahrscheinlich und Kritisch)
   (a) Reduktionsstrategie: Sich des Risikos bewusst sein und bereit sein, viel Zeit und Arbeit zu investieren
   (b) Reduziertes Risiko: Das Risiko ist reduziert immer noch sehr hoch. Wir haben uns beide an Multiplayer fest gefressen und werden es höchst wahrscheinlich trotzdem umsetzen
   (c) Tatsächliches Outcome: Wie erwartet ist die Maturaarbeit nach 250h immer noch weit entfernt von vollständig. Dennoch ist das Ziel erreicht von einem funktionierendem Videospiel.
2. Beide Teammitglieder haben keine Ahnung von der Entwicklung einer Multiplayer-Funktion (Wahrscheinlich und Kritisch)
   (a) Reduktionsstrategie: Andere Spielideen erarbeiten, die ohne Multiplayer funktionieren.

(b) Reduziertes Risiko: Das Risiko ist reduziert immer noch sehr hoch. Wir haben uns beide an Multiplayer fest gefressen und werden es höchst wahrscheinlich trotzdem umsetzen

(c) Tatsächliches Outcome: Multiplayer ist mit Abstand der grösste Zeitfresser. Der Multiplayer alleine hat über 100h Arbeit, also fast ein Drittel unserer Arbeit ausgemacht.

3. Es könnten sich Bugs in der abgabebereiten Version verstecken (Wahrscheinlich und Kritisch)

   (a) Reduktionsstrategie: So viel testen wie nur möglich. Sich ausserdem auf die kritischen Bugs konzentrieren.

   (b) Reduziertes Risiko: Das Risiko ist selbst reduziert sehr hoch. Wir wollen ein perfektes Game abgeben.

   (c) Tatsächliches Outcome: Wir haben unser Möglichstes gegeben. Wir können nur die Daumen drücken, dass Herr Hunziker keine neuen Bugs findet.

4. Die Version für die Abgabe könnte aus unbekannten Gründen crashen (Unwahrscheinlich (hoffentlich) und Kritisch)

   (a) Reduktionsstrategie: So viel testen wie nur möglich.

   (b) Reduziertes Risiko: Das Risiko ist selbst reduziert sehr hoch. Man konnte bereits des Öfteren beobachten, wie sich eine schlechte erste Version auf eine Firma auswirken kann.

   (c) Tatsächliches Outcome: Wir haben unser Möglichstes gegeben. Wir könnten nur noch die Daumen drücken und hoffen, dass es keine fatalen Fehler gibt.

4.2 Mässig

1. Elia hat noch nie mit Unity gearbeitet (Wahrscheinlich und Unbedenklich)

   (a) Reduktionsstrategie: Elia befasst sich zu Beginn mit Unity

   (b) Reduziertes Risiko: Weiterhin kann angenommen werden, dass Elia auf gewisse Schwierigkeiten mit Unity stossen wird. Jedoch sollten diese nicht viel Zeit oder Mühe benötigen.

   (c) Tatsächliches Outcome: Elia konnte alles Nötige zu Beginn erlernen und bei allfälligen Fragen auf das Internet oder Marc zurückgreifen.

4.3 Tief

1. Wir werden selbst nicht mit unserem Spiel zufrieden sein. (Wahrscheinlich und Unbedenklich)

(a) Reduktionsstrategie: Sich bewusst sein, dass wir nur 2 Schüler sind, die keine bisherigen Erfahrungen hatten. Wir dürfen uns nicht an etwas festfressen".

(b) Reduziertes Risiko: Es stellt sich das Risiko, dass wir unsere Zeit verschwenden. Wir müssen teilweise einfach weitermachen.

(c) Tatsächliches Outcome: Man ist nie richtig fertig mit einem Spiel. Es gibt immer Dinge, die man Verbessern könnte. So könnten auch wir noch gefühlt tausende weitere Stunden investieren.

Teil II

Produkt

Aktuelles Produkt

4.4 Aktuelle Features

TL;DR: Wir konnten fast all unsere Anforderungen erfüllen.

4.4.1 KZO

1. **Zeit**
   Geplant war ungefähr eine Lektion pro Woche. Wir haben die geschätzten 50 Stunden um einen Faktor von sechs überschritten. Unser gemessener Zeitaufwand, während aktivem Programmieren, beträgt über 330 Stunden. Da sind die Stunden, in denen wir uns in der Freizeit, und mit Herrn Hunziker getroffen haben, gar nicht notiert.

2. **Begleitperson**
   Wir hatten zum Glück kein Problem eine Begleitperson zu finden. Martin Hunziker war gerne bereit unsere Arbeit im Blick zu behalten.

3. **Plagiat**
   Zwar sind einige Grafiken aus dem Internet, jedoch sind alle Nutzungsfrei. Auch sonst haben wir keinen Code geklaut und verwendet.

4.4.2 Unsere eigenen Anforderungen

Notwendig

- **System**
  Unser Spiel funktioniert sowohl auf macOS als auch auf Windows. Dies ausserdem mit einer FPS von 300, bis teilweise 500, bei den meisten modernen Geräten

- **Benutzeroberfläche**

  - **Startpage**

Unsere Startseite hat einen "Spielen"-Knopf, einen "Einstellungen"-Knopf, einen "Credits"-Knopf und einen "Verlassen"-Knopf. Unsere vereinzelten Tester bewerteten die Startseite als intuitiv verständlich. Bedienbar mit der Maus, hatten die Tester keine Probleme sich zurechtzufinden.

– **Credits**

Es gibt die Möglichkeit die Credits vorzeitig zu Verlassen. Dies geschieht durch das Drücken des Knopfes mit dem Haus als Icon. Die Tester fanden dies auch sehr intuitiv.

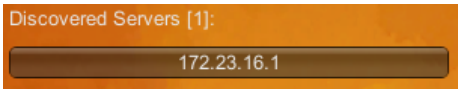
– **Settings**

Man kann hier die Auflösung ändern und auswählen, ob das Spiel im Vollbild dargestellt werden soll. Auch gibt es hier die Möglichkeit die Musik lauter, leiser zu stellen oder zu muten. Wenn man Einstellungen geändert hat, muss man dies bestätigen.

Wenn man die Auflösung ändert und dies bestätigt, werden die Änderungen angewendet und man hat zehn Sekunden Zeit erneut zu bestätigen. Sonst werden die Änderungen zurückgesetzt. Dies wurde implementiert, um eventuellen Problemen vorzubeugen.

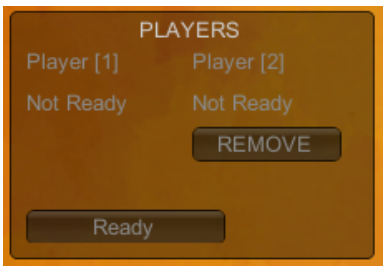
– **Server Management**

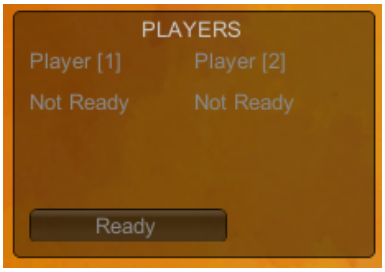
Nach Interaktion mit dem "Start"-Knopf gibt es die Möglichkeit einen Server zu Hosten oder einer bereits existierenden Lobby als Client beizutreten. Ausserdem kann man automatisch nach bereits existierenden Servern suchen.



Den gewollten Server kann man mit einem einfachen Linksklick auswählen.

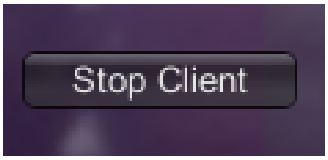
– **Lobby Management**





Player (1) ist dabei immer der Host, und Player (2) immer der Client. Das eigentliche Spiel beginnt erst, sobald beide Spieler bereit, also "ready", sind. Dieser Status "ready" wird dabei immer aktualisiert dargestellt. Der Host hat ausserdem die Möglichkeit den Spieler (2) aus der Lobby zu entfernen. Wir wollten zuerst eine Möglichkeit für einen "Schlüssel" hinzufügen, allerdings stellte sich dies als überflüssig heraus, da alle Spieler unter einem Dach sitzen.

– **InGame-UI**



Der Host hat im Spiel die Möglichkeit, den Server zurück zur Lobby zu schicken, und das Spiel somit neuzustarten. Er kann auch direkt aufhören zu hosten. Auch der Client hat durchgehend die Möglichkeit den Server zu verlassen.

Im Pausemenu kann man ausserdem auf die Schnelle die Musiklautstärke und die Stärke des Schneefalls ändern.

- **Lokaler Multiplayer**
  Das Spiel kann im lokalen Netz gespielt werden. Zum Testen auch auf demselben Gerät. Als Protokoll haben wir TCP gewählt. Dies, da TCP viel zuverlässiger ist. Folgende vier Punkte waren entscheidend für die Wahl von TCP über UDP:

  - **Zuverlässigkeit**
    Keine Probleme mit verloren gegangenen Paketen. Ging ein Paket verloren, sendet TCP es einfach erneut. Entweder werden alle Daten erfolgreich übermittelt, oder man bekommt einen Error.
  - **Sequenziert**
    TCP stellt sicher, dass jede Nachricht in der gleichen Reihenfolge eintrifft, in der sie gesendet wurde. Wenn man "a" dann "b" sendet, erhält man auf der anderen Seite garantiert auch zuerst "a" dann "b".
  - **Verbindungsorientiert**
    TCP hat sich auf das Konzept einer Verbindung konzentriert. Eine Verbindung bleibt so lange offen, bis entweder der Client oder der Server sich dazu entscheiden, sie zu schliessen. Anschliessend werden sowohl Client als auch Host benachrichtigt, dass die Verbindung beendet wurde.
  - **Überlastungskontrolle**
    Wenn ein Server überlastet wird, drosselt TCP selbstständig die Daten, um einen Zusammenbruch durch Überlastung zu verhindern.

- **Einstellungen**
  Auflösung, Vollbild, Lautstärke und Stärke des Schneefalls können eingestellt werden.

- **Kamera**
  Die Kamera ist beweglich und das ganze Schlachtfeld ist sichtbar. Dabei kann man auch zoomen. Kontrollierbar ist dies mit den Pfeiltasten.

- **Truppen**
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT

- **Gewinnmöglichkeit**
  Um zu Gewinnen, muss eine eigene Truppe die Portale des Gegners durchschreiten. Dies verhindern zuerst sogenannte Helden. Sterben jene Helden, erledigt ihr Racheeffekt die ganze Linie auf der sich der Held befand.

- **Lanes**
  Unser Spiel hat 4 sogenannten Linien ("Lanes"). Diese unterschiedlichen Linien wurden in der Tiefe verschoben.

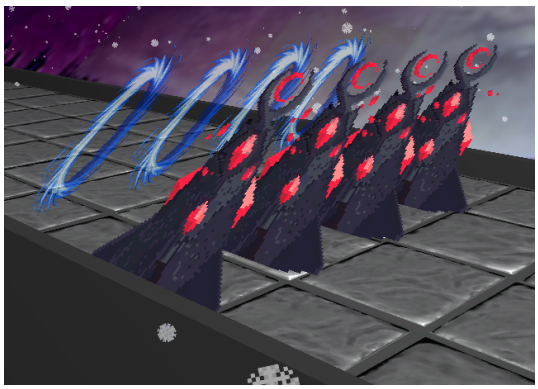
- **Synchronisation**
  Unser Spiel Synchronisiert alle Daten in Echtzeit. Dadurch sehen alle Spieler immer das Gleiche. Wird der Client aus einem unbekannten Grund disconnected kann er versuchen sich erneut zu verbinden und der Server wird dann direkt alle Daten synchronisieren. Auch hier hilft und das gewählte Protokoll TCP. Geht ein Paket verloren, wird es automatisch erneut gesendet. Dies hilft und dabei, die Synchronisation zu jeder Zeit aufrecht zu erhalten.

- **Crossplay** Es ist möglich sich mit einem Windows Rechner und einem macOS Rechner zu verbinden. Allerdings muss dem Spiel dazu die Kommunikation durch die Firewall teilweise erlaubt sein. Bei einer eher aggressiveren Firewall wird zudem teilweise der Server-Broadcast geblockt. Dies hätte zur Folge, dass man die Server nicht automatisch finden kann. Verbinden sollte man sich allerdings trotzdem noch können.

Anvisiert

- **Helden**
  Beide Spieler haben im Moment den gleichen Helden. Dies ist ein magischer Turm. Er hat keine Angriffsmöglichkeiten. Allerdings kann er sich passiv heilen und so als eine temporäre Grenze zwischen den gegnerischen Truppen und den eigenen Portalen agieren. Die eigenen Truppen werden durch diese Portale hinter den Helden gespawned.

- **Design**




Wir haben uns für ein 2.5d Design entschieden. Dabei ist der Vorteil, dass wir 2d Grafiken benutzen können. Die Sprites werden dabei um 45 Grad gedreht. Auch die Kamera schaut in einem 45 Grad Winkel auf das Geschehen. Dadurch ergibt sich die Illusion einer Perspektive, in der man von der Seite auf das Geschehen schaut.

- **Deck**
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT
  IMPORTANT

- **Truppen**
  - **Nahkampf** Eine Truppe mit wenig Reichweite.
  - **Fernkampf** Eine Truppe die auf Reichweite angreift. Sie scheisst Projektive, zum Beispiel ein Bogenschütze, der mit Pfeilen schiesst.
  - **Suizid** Eine Truppe die bei Berührung mit einem Gegner stirbt und einen Effekt auslöst.

- **Effekte**
  - **Gift:** Zeitlich limitierter und wiederholender Schadenseffekt. Eine visuelle Markierung soll vorhanden sein und das gleiche Gift, also der gleichen Truppe, soll nur einmal auf jemanden sein. (Hier: Pilz-Truppe)
  - **Rache:** Truppen mit Rache haben einen Effekt nach ihrem Tod. Zum Beispiel eine Truppe beschwören oder Schaden verursachen. (Hier: Helden mit Feuer)

Möglichkeiten

- **Monetarisierung**

  1. **Kostenpflichtiges Spiel**
     - Wir haben eventuell vor unser Spiel nach Vollendung der Dev Phase zu veröffentlichen. Allerdings auf keinen Fall für einen hohen Preis.

- **Design**
  Wir werden auch in Zukunft das Design noch verbessern. Hier gehts es aber um den Feinschliff, z.B. mehr Hintergründe, und mehr Dinge selbst zu designen.

- **Truppen** Weitere Truppen können jederzeit erstellt werden. Hier ist kein Limit gesetzt. Leben, Schaden, Darstellung, Effekt und vieles mehr kann angepasst werden. Wir haben alle Programme mit dem Hintergedanken geschrieben, dass es jederzeit ohne grössere Anpassungen möglich ist, neue Truppen hinzuzufügen.

Fürs erste verworfen

- **Online Multiplayer**
  Um sich mit Rechnern ausserhalb des eigenen Netzwerkes verbinden zu können, müssten wir das ganze Multiplayer System unseres Spieles anpassen, was einen kompletten Recode zur Folge hätte. Ausserhalb würde sich die Frage nach der Finanzierung stellen, falls wir einen externen Server kaufen / mieten würden.

- **Shop**
  Ein Shop ohne Anti-Cheat und/oder Server ist die perfekte Angriffsfläche für Cheater und hat somit nicht viel Sinn. Wir müssten dann ausserdem ein Ranking-System hinzufügen, damit wir die Spieler belohnen können.

- **Kampagne**
  Unser Spiel ist vor allem auf den Multiplayer ausgelegt. Eine Kampagne setzt ausserdem voraus, dass man einen offline Gegner ("Bot") hat. Wir haben uns bereits früh gegen eine Kampagne entschieden, da dies nach unserer Meinung nach in diesem Stadium leider eine komplette Zeitverschwendung wäre.

- **Anti-Cheat**
  Dies ist für uns ohne Erfahrung in diesem Bereich und einem lokal berechnetet Spiel ein Ding der Unmöglichkeit. Wir haben keine Erfahrung in diesem Bereich und es ist ein extrem komplexes Thema. Allerdings hat der Host sozusagen über fast alles die Macht, der Client kann also fast gar nicht kaputt machen. Wir hoffen ausserdem auf die Menschen, die unser Spiel spielen, und darauf dass sie nicht versuchen zu schummeln.

- **Errungenschaften**
  Belohnungen zu erhalten ist immer toll. Wenn es eine Währung wäre, bräuchten wir allerdings auch einen Shop. Man müsste ausserdem neue Dinge hinzufügen, die man zuerst erspielen muss. Es könnten auch Pokale sein, allerdings bräuchten wir dann bereits wieder ein Ranking-System.

- **Bot**
  Ein guter Bot passt sich der Spielweise des Gegners an. Einen so guten Bot zu programmieren ist eine eigene Herausforderung für sich. Wir hatten zwar die Idee, dass der Bot immer zufällige Truppen zu zufälligen Zeiten spielt. Allerdings würde dies auch eine Menge Balancing benötigen, wofür wir im Moment leider zu wenig Zeit haben.

- **Helden**
  Wir hatten als Ziel, mehrere verschiedene Helden zu gestalten und zu programmieren. Allerdings stellte sich relativ früh heraus, dass wir die nötigen Ressourcen lieber an einer anderen Stelle investieren. Wir haben nun einen Helden, allerdings würde sich das Ausbauen der Helden nach der Abgabe als spannend gestalten.

- **Ingame-Chatfenster**
  Wir hatten eine funktionierende Version, allerdings stellte sich dies als überflüssig heraus. Der Grund ist, dass man normalerweise neben der Person sitzt, mit der man im Moment spielt.

- **Mehrsprachig**
  Das Spiel ist im Moment nur in Englisch verfügbar. Eventuell werden wir dies nach der Abgabe der schriftlichen Arbeit noch anpassen.

- Effekte
  - **Wiederbelebung:** Die Möglichkeit einer Wiederbelebung würde sich nur für spezielle Truppen anbieten. Der Effekt Wiederbelebung würde wahrscheinlich ein Vielfaches davon kosten, wenn man die Truppe erneut losschicken würde.
  - **Rüstung:** Uns schien dies nicht nötig zu sein. Dies da es sich erst lohnen würde, sobald wir mehr Truppen, Effekte und mehr hinzugefügt hätten.
  - **Aura:** Das Prinzip einer Aura hört sich ganz gut an. Allerdings müssten wir sehr wahrscheinlich einen Recode des Damage-Systems machen. Dies würde uns mehr Zeit kosten, als es im Moment wert wäre.

- **Tutorial**
  Ein Tutorial ist ein immenser Zeitaufwand. Es muss so gestalten sein, dass der Spieler gar nicht bemerkt, dass es ein Tutorial ist. Wir haben uns ausserdem darauf geachtet, das ganze Spiel so intuitiv wie möglich zu gestalten.

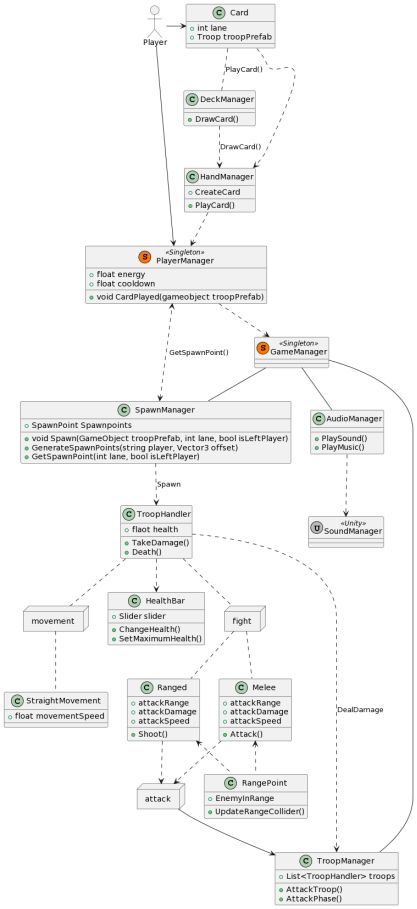
- **Monetarisierung**
  Ohne Anti-Cheat leider vollkommen nutzlos.
  1. **Kaufbare Gegenstände**
     - Neben Schummelmöglichkeiten, stellt sich hier ausserdem die Gefahr eines möglichen Pay-to-Win.
  2. **Skins**
     - Der Spieler bräuchte ein Inventar, das mit einem Server kontrolliert und synchronisiert wird.

- **Zauber**
  Wir hatten zu einem frühen Zeitpunkt eine einigermassen funktionierende Testversion. Der Zauber wurde dem Spieler auch auf einer Karte verteilt. Eine erneute Nutzung davon würde allerdings viele Änderungen des gesamten Spiels bedeuten. Dies hätte sehr wahrscheinlich auch einen Recode zur Folge.

– **Dornen:** Nachdem wir den Zauber "Gift" hinzugefügt haben, sahen wir den Grund nicht, nun auch noch Dornen hinzuzufügen.

Kapitel 5

Architektur



Kapitel 6

Feedback

Kapitel 7

Zukunft

Teil III

Organisation

Kapitel 8

Technologien

8.1 Unity

`https://unity.com/`
Eine Lauf- und Entwicklungsumgebung für Spiele. Sehr beliebt für Indie-Developer. Unity ist ein mächtiges und sehr hilfreiches Programm bei der Entwicklung von Videospielen. Einige Vorteile sind:

- Physik
  - Schwerkraft
  - Reibung
  - Beschleunigung und Bremsen
- Visuell
  - Schwerkraft
- Cross-Plattform

8.2 Github

`https://github.com/`

8.3 Jetbrains Rider

`https://www.jetbrains.com/rider/`

8.4 Jira

`https://www.atlassian.com/software/jira` Webbasiertes Programm für Projektmanagement und Zeiterfassung.

8.5 Confluence

https://www.atlassian.com/de/software/confluence

8.6 LaTex

https://www.latex-project.org/

8.7 Stable-Diffusion

https://stability.ai/

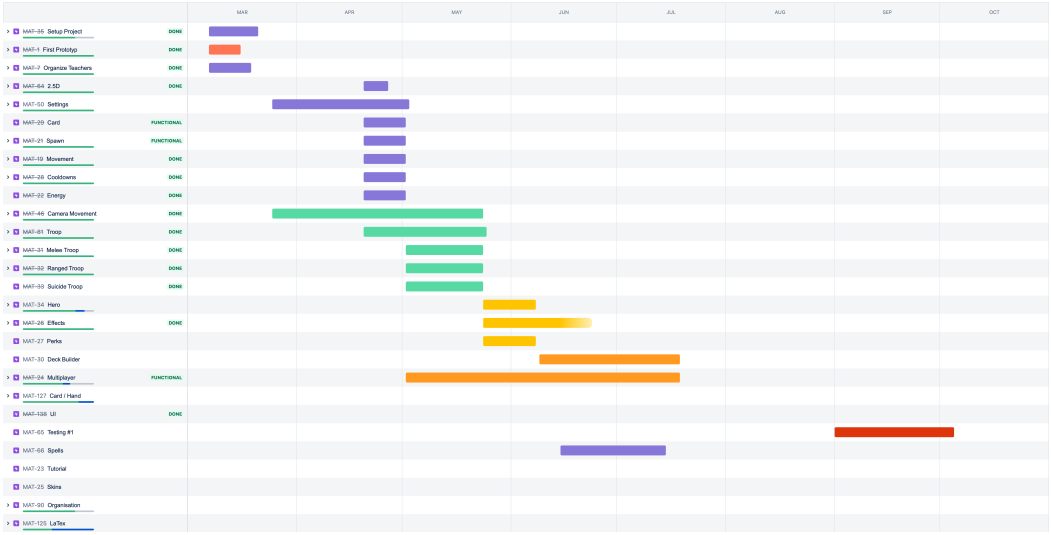
Kapitel 9

Team Management

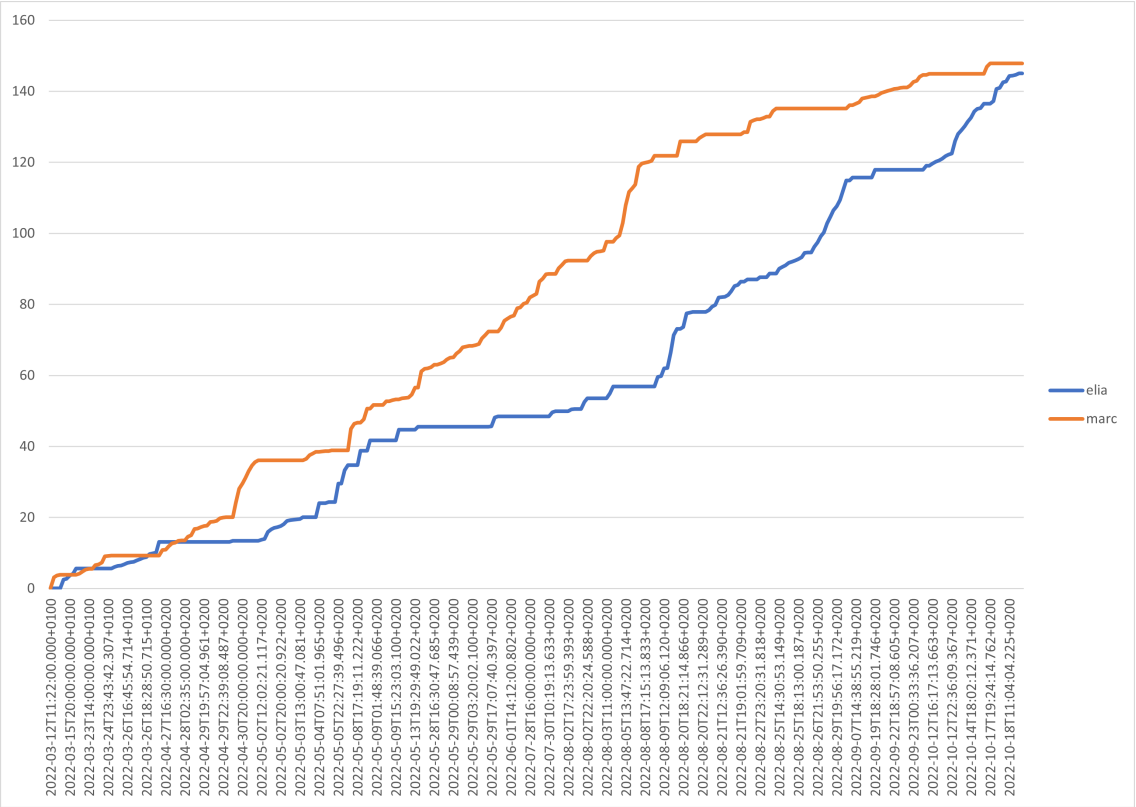
Kapitel 10

Time Management

10.1 Roadmap



10.2 Time-Tracking



HIER NOCH ENDZEITEN IN LISTE SCHREIBEN
treffen mit hunzi und jezek oder starbucks oder dihei....
07.03 treffe mit hunzi im inf büro 45m
08.03 treffe im starbucks vor em theater 3h
15.03 jezek und hunzi 40m im 1.03 oder so 45m
20.04 treffe bim marc vor allem roadmap 4h
04.05 treffe mit hunzi im besprechigszimmer 2.20 2h
15.6 treffen mit hunzi im 2.20(gruppenzimmer) nach morgenschuel 1h30m
01.09 treffe mit hunzi nach vormittagsschuel im 12i 1h 10m
23.09 treffe mit hunzi nach 16h 1h
16.10 starbucks vor bowle 2h

Teil IV

Reflexion

Kapitel 11

Team

Rückblick

Nächstes Mal

Kapitel 12

Elia

Rückblick

Nächstes Mal

Kapitel 13

Marc

Rückblick

Nächstes Mal

Glossar

This document is incomplete. The external file associated with the glossary 'main' (which should be called `main.gls`) hasn't been created.

Check the contents of the file `main.glo`. If it's empty, that means you haven't indexed any of your entries in this glossary (using commands like `\gls` or `\glsadd`) so this list can't be generated. If the file isn't empty, the document build process hasn't been completed.

You may need to rerun LaTeX. If you already have, it may be that TeX's shell escape doesn't allow you to run makeindex. Check the transcript file `main.log`. If the shell escape is disabled, try one of the following:

- Run the external (Lua) application:

  ```
  makeglossaries-lite "main"
  ```

- Run the external (Perl) application:

  ```
  makeglossaries "main"
  ```

Then rerun LaTeX on this document.
This message will be removed once the problem has been fixed.