

Hochschule der Medien Stuttgart
Fakultät Druck und Medien

Projekt Halte das Schloss

Projektbericht

Abgabedatum: 04. März 2019

Vorgelegt von: Alexander Allerdings,

Matrikelnummer: 34669

Studiengang: Medieninformatik

Betreuer: Prof. Walter Kriha

Olivia Odenthal,

34673

Niklas Werth

34518

1.1 Projektbeschreibung

Im Rahmen des Moduls „113400 Software-Projekt“ haben wir uns vorgenommen ein Spiel zu programmieren. Das Spiel sollte Elemente eines „Tower Defense“-Spiels mit Elementen eines Strategiespiels verbinden und online mit mehreren Spielern spielbar sein.

1.2 Verwendete Software

Als Game-Engine wurde Unity gewählt, die Grafiken wurden in Maya, Blender, Adobe-Photoshop, Adobe-Illustrator und Substance Painter erstellt.

1.3 Teammitglieder

Das Team stellte sich zusammen aus Alexander Allerdings (Programmierung), Olivia Odenthal (Programmierung und Grafik) und Niklas Werth (Grafik).

1.4 Ziel

In die verwendete Software musste sich komplett neu eingearbeitet werden, da nur grundlegende Kenntnisse in Unity, Maya, Blender und Photoshop vorhanden waren. Deshalb bot sich ein Ziel an, welches nicht zu hoch gesteckt, aber dennoch um weitere Elemente erweiterbar sein sollte.

Die Grundidee des Spiels sollte Elemente eines „Tower Defense“-Spiels mit Elementen eines Strategiespiels verbinden und online mit mehreren Spielern spielbar sein. Des Weiteren sollten alle Objekte des Spiels von uns selbst erstellt werden, um den maximalen Lernerfolg zu garantieren.

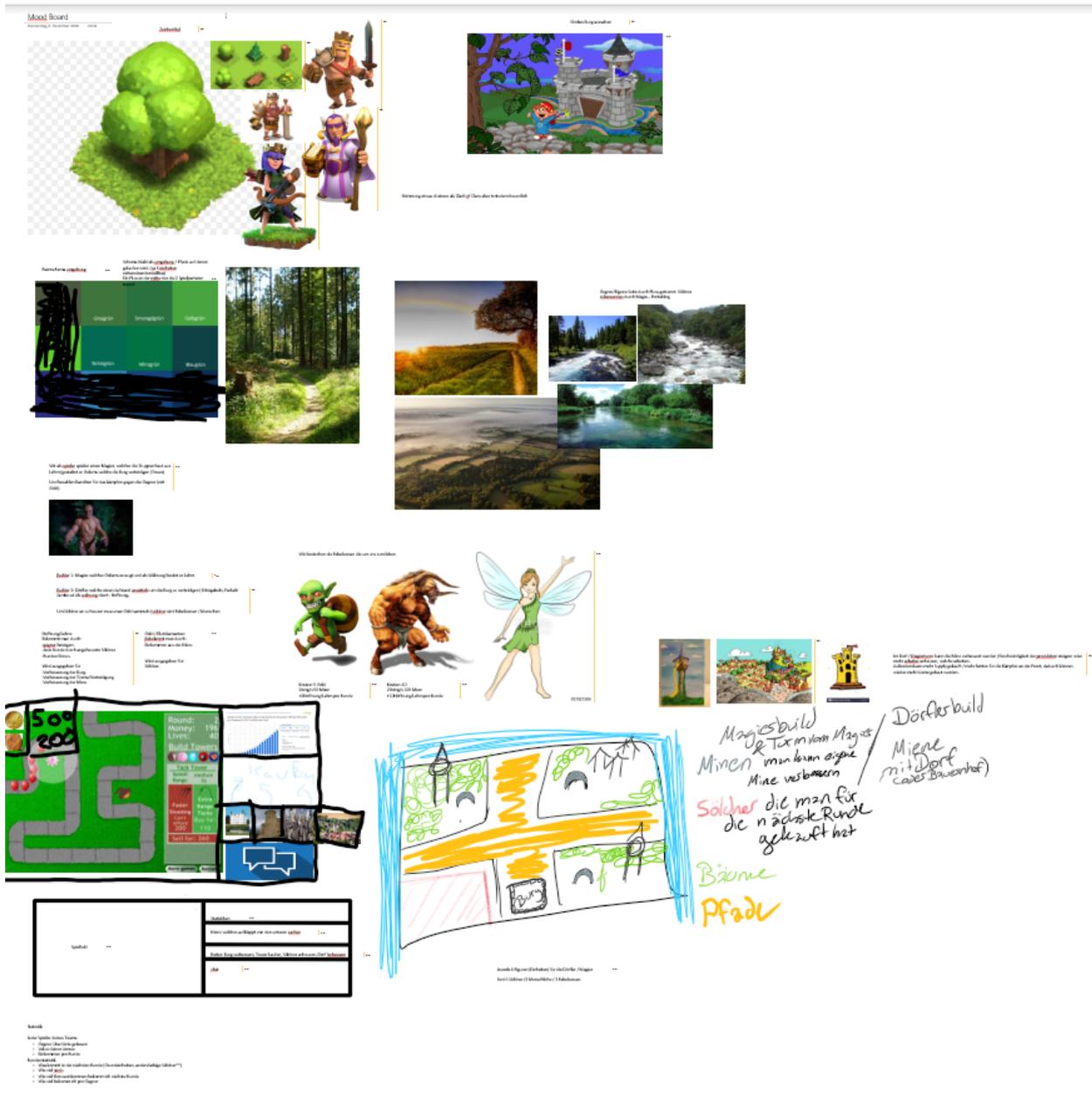
Außerdem sollten erste Erfahrungen in einem größeren Projekt gesammelt werden und alles was damit verbunden ist.

2. Projektablauf

2.1 Vorarbeit

In dieser Phase des Projekts wurde zuerst ein Moodboard erstellt, welches dafür da war, Elemente und Grafiken anderer Spiele, an welchen wir uns orientieren wollten, zusammen betrachten zu können. Anhand dieses Moodboards wurde die „must-haves“ des Spiels festgelegt und es wurde ein Grafikstil festgelegt. Es wurden außerdem Spielmechaniken aus bekannten Spielen wie Warcraft 3, Starcraft, Clash of Clans und einigen Browser Flash Games untersucht, um daraus Ideen für Spielmechaniken sowie Balancen in verschiedenen Spielen um diese Fair zu gestalten.

Daraufhin wurden die ersten Aufgaben verteilt, was dazu führte, dass Alexander und Olivia zu Beginn für das Programmieren eingeteilt waren, während Niklas mit den Grafiken beginnen durfte.



In dieser Phase bestand ausschließlich das Problem, einschätzen zu können, auf wie viele und welche Funktionen und Grafiken das Spiel beschränkt werden sollte. Aus diesem Grund wurde der Ansatz gewählt zuerst die wichtigsten Elemente zu erstellen und das Programm im Laufe der Zeit eventuell weiterentwickeln zu können, sofern es möglich ist. Dies wurde durch eine Abwandlung und Vereinfachung des Agilen Projektmanagements „Scrum“ gemacht. Nachdem wir so ein grundlegendes Spiel mit wenigen Funktionen hatten, konnten wir daraufhin Schritt für Schritt neue Funktionen implementieren. Dies unterstützte auch das Entwickeln mit 2 Entwicklern, da sich jeder einzelne Tasks herauspicken konnte die es zu bearbeiten galt.

2.2 Arbeitsphase

Da Unity vorher noch nie benutzt wurde, waren viele Funktionen der Game Engine für das gesamte Team unbekannt, weshalb sich jedes Teammitglied zuerst in Unity einarbeiten musste.

In der 2. Phase des Projekts widmeten sich Alexander und Olivia hauptsächlich dem Programmieren, Niklas widmete sich den Grafiken. In dieser Phase wurde sehr viel zusammen in einem Raum gearbeitet, sodass auch Gedanken der Programmierer in die Grafik miteinspielten und Gedanken des Grafikers in die Programmierung miteinfließen. Dadurch konnten einige Probleme gelöst werden, da für diese Probleme eine sehr simple Ursache herrschte, die durch einfache Fragen eines „unbeteiligten“ gelöst werden konnten und dadurch vor allem der „Tunnelblick“, welchen man über die Zeit an einem Problem entwickelt, beseitigt werden konnte.

Zunächst war es schwierig für Olivia sich in Unity einzuarbeiten. Es fehlte nämlich das grundlegende Verständnis für die Möglichkeiten die Unity bot. Erst nach einiger Zeit hatte sie richtig verinnerlicht, wie die Skripte aufzuteilen waren, wo man sie anhängen konnte und was alles Prefab werden sollte. Danach ging das Lernen deutlich schneller.

Sie begann damit die Map zu gestalten indem sie das Terrain versuchte passend zu formen. Stellte allerdings später fest, dass es eine viel einfachere Variante gab die Map zu erstellen die auch weniger Rechenleistung beanspruchte. Somit entschied sie ein flaches Terrain zu nutzen und die Ecken mithilfe einfacher Quader zu bilden.

Olivia war vor Allem damit beschäftigt das Grundgerüst für das Spiel zu bauen. Darunter der Timer und das damit verbundene Spawnen. Hierbei war wichtig dass verschieden viele Einheiten verschiedener Arten gerecht auf die 3 Spawnpunkte (bzw auch eine variable Anzahl der Spawnpunkte) verteilt werden mussten.

Auch das Setzen der Einheiten und das Entwickeln des dazugehörenden Grids war Teil ihres Aufgabenfeldes. Dabei sollten Einheiten nur auf bestimmten Feldern gesetzt werden (welche durch das Grid angezeigt wurden). Zudem sollten die Einheiten nur auf den möglichen Positionen angezeigt werden.

Dabei hatte sie länger mit den Raycasts zu kämpfen. Das Problem war, dass die Einheiten nicht richtig der Maus folgten. Sie waren immer leicht versetzt.

Später erstellte sie auch noch die Healthbars und sorgte dafür dass Einheiten Schaden zugefügt werden konnte.

Beim Erstellen der GUI wurde darauf Wertgelegt eine Möglichst gute Struktur einzubauen. Die Buttonleisten wurde dabei durch Grid Layouts sauber angeordnet.

Das Erstellen der GUI brachte auch zwischenzeitlich Probleme mit sich. Die Canvas (auf welcher sich die GUI Elemente befanden) wurde nämlich nur zum Teil von der Camera erfasst. Es stellte sich später heraus dass nur ein falsch eingestellter Faktor das Vorschaubild verzerrte. Ein weiterer Aspekt der lange für Verwunderung sorgte war dass die verschiedenen Panels die Sicht auf das Spielfeld trübten. Auch das ließ sich aber mit dem Anpassen der Einstellungen beheben sodass die kräftigen Farben des Spieles wieder richtig wirkten.

Beim Einbinden von Sounds gab es anfangs auch einige Schwierigkeiten. Anfangs wurden die Töne nur direkt am Anfang des Spieles abgespielt. Danach blieb es still.

Der zweite Versuch führte zunächst zu einem komplett stillen Spiel. Als dann aber rausgefunden wurde wie man den minimalen und maximalen Radius der Geräuschquelle einstellen kann wurde das Problem gelöst.

Zudem wurden auch viele weitere kleine Aufgaben übernommen zu welchen es keine spezifischen Probleme gab.

Allerdings gab es ein großes Problem als plötzlich alle Prefabs nicht mehr funktionierten. Wir hatten aus versehen auf das Projekt in einer anderen Unityversion verwendet welche alle Prefabs unnutzbar gemacht hatte. Von da an stellten wir sicher, dass wir zu jeder Zeit die gleiche Version verwendeten.

Ein großes Problem für Alex war es sich auf die Unterstützung von Unity ein zu lassen. Da vorherige Kenntnisse nur durch komplett selbst erstellte Engines und dem Programmieren ohne Tools wie den Prefabs erlangt wurde fing er anfangs auch in diesem Projekt an Dinge wie eine Physic Engine zu schreiben und das Pathfinding über einen selbst geschriebenen A* Algorithmus. Dies war jedoch unnötig da die meisten solcher einfacheren Funktionen bereits in Unity implementiert sind und GameObjekten als Funktion (wie ein Script) mitgeben kann.

Nachdem sich also in C# und vor allem Unity eingefunden wurde, ging das entwickeln von neuen Dingen um einiges schneller von der Hand. Es konnten nun auch größere Abhängigkeiten zusammengebaut werden. Wie beispielsweise das Partikeleffekte wie das ausspucken von Goldmünzen aus der Miene sich anhand von bestimmten Werten wie dem Gold pro Sekunde Wert und dadurch bei jedem Spieler anders aussehen.

Ein großer Teil der Zeit wurde jedoch in das entwickeln der KI Agenten (die Tower und Gegner). Welche sich in unterschiedlichen Momenten unterschiedlich verhalten sollen. Hierfür wurden mehrere State Machines implementiert. Dies gestaltete sich vor allem im Hinblick auf Performance als schwierig. So haben wir zum testen der Performance immer einen alten Laptop welcher lediglich eine Onboard-Grafikkarte besitzt genutzt und diesem minimum 300 Einheiten gleichzeitig Simulieren lassen. Wenn dies mit einer annehmbaren Performance lief (25+ Fps) galt der jeweilige Algorithmus als annehmbar für den jeweiligen Fall.

Das größte Problem welches sich 3 Wochen vor der Medianight zeigte und das ganze Projekt zunächst zum scheitern verurteilte war das implementieren der Mehrspielerfunktion. Zunächst wurde das Projekt als Singleplayer Spiel entwickelt um dieses später durch Networking zu einem Multiplayer-Spiel um zu schreiben.

Hierfür wurde zunächst ein Dezentraler Server benötigt, welchen Unity jedoch testweise bereitstellte, nachdem man sich hierfür registrierte.

Da sich niemand mit Networking in Unity auskannte musste sich Alex ohne Vorwissen alles aneignen um diese Aufgabe zu erfüllen. Dies bereitete allen jedoch mehrere Stolpersteine, da vieles der Gamelogik umgeschrieben werden musste und zunächst das allgemeine Verständnis der verschiedenen Zustände eines Clienten/Servers begriffen werden musste. Ein Großteil der Zeit wurde daraufhin damit verbracht Schritt für Schritt den existierenden Code umzuschreiben und Teile welche auf dem Server bzw. auf dem Localen Client laufen aufzuspalten (liegen immer in dem selben Scriptfile und werden nur durch Erweiterungen von Funktionen [Command] auf verschiedenen Geräten ausgeführt). Nach 2 Wochen intensivster Recherche und mehr als 50 Stunden Arbeit um den Singleplayer in einen Multiplayer um zu schreiben war es jedoch geschafft und es konnte an den Feinschliff und das Balancing gehen.

Diese Erfahrung war jedoch sehr wichtig und hat Alex am meisten voran gebracht. Nicht nur dadurch, dass er nun weiß wichtige und kritische Dinge nicht nach hinten zu verschieben, sondern auch wie man sicher Netzwerke aufbaut um Spieler am Cheaten(Schummeln) zu hindern.

Um dem Spiel noch etwas mehr Feinschliff zu geben hab Alex sich eine Woche vor der Medianight noch an das Preprocessing herangewagt und dadurch Beleuchtung, Farben und ausschmücken der Szene übernommen hat. Dies wurde nach Möglichkeit durch

bereits gesammelte Erfahrungen in der Fotografie, einigen Tutorials, sowie ein wenig herumprobieren mit den verschiedenen Schieberegler bewerkstelligt.

Ein wichtiger Teil der Arbeit war auch das Dokumentieren im Quellcode. Da sich Alex meist 2-3 Tage am Stück für jeweils mehrere Stunden an das Projekt machte und dadurch manchmal den halben Quellcode umstrukturieren und umschreiben musste, war es für die anderen schwer nachzuvollziehen, was sich in den letzten 2-3 Tagen alles verändert hat.

Niklas war für die Grafiken verantwortlich und erstellte mehrere Charaktere. Diese stellten sich zusammen aus Menschen, Golems und Geistern. Außerdem wurden noch Umgebungselemente, wie eine Mühle, eine Höhle und einer Burg erstellt. Dafür mussten die Modelle zuerst in Maya erstellt werden. Hierbei war das große Problem, dass mit Grundkenntnissen kein anatomisch korrekter Körper zu modellieren war, weshalb einige Tutorials für das Erstellen der Modelle notwendig waren.

Nachdem diese Hürde überwunden war und nach einiger Zeit die Charaktere erstellt waren, konnten diese in Unity für die ersten Tests importiert werden.

Dadurch, dass die Grafiken zunächst deutlich mehr Zeit für die Erstellung benötigten, als gedacht, wurden zunächst Funktionen im Spiel implementiert, für welche Platzhalterobjekte für die Tests ausreichten.

Als die Modelle erstellt waren, mussten die Texturen angefertigt werden. Hierbei musste beachtet werden, dass alle Designs zueinander harmonisch sein mussten. Somit wurden zuerst in Photoshop Bilder zu den Charakteren gemalt, an welchen sich dann später für die Texturen für die 3D Modelle orientiert wurden. Hierbei wurden die Entscheidungen für das finale Design im Team getroffen. Für die Texturen wurde das Programm „Substance Painter“ verwendet. Dabei stellten sich einige Probleme: Substance Painter war vorher unbekannt und ohne gewisse Einstellungen gemacht zu haben, konnten keine UV-Maps sondern nur ein „gebaketes“ Objekt exportiert werden. Für Unity wurden jedoch die UV-Maps für verschiedene Kanäle des Modells benötigt.

Nach diesem Schritt stellte sich dann das Problem, dass Modelle mit Texturen in Unity auf einem anderen Weg eingebunden werden müssen, als Modelle ohne Texturen. Nach einigen YouTube Videos konnte jedoch auch dieses Problem behoben werden, sodass die fertigen Charaktere ins Spiel eingebunden werden konnten.

Daraufhin mussten die Umgebungselemente gestaltet werden. Diese waren sehr viel einfacher zu erstellen, da durch das Erstellen der Charaktere schon sehr viel gelernt wurde und sich der Workflow und die Kenntnisse im Programm dadurch sehr stark verbessert haben. Dadurch konnten dann auch alle Elemente fristgerecht zur Medianight fertiggestellt werden, obwohl während der Erstellung der Charaktere daran gezweifelt wurde. Hierbei wurde Niklas dann zusätzlich von Olivia unterstützt, da auch sie Erfahrungen im gestalterischen Bereich sammeln wollte.

Hierfür nutzte Olivia ihre Grundkenntnisse in Blender um einige GameObjekte zu erstellen (verschiedene Bäume, den Magierturm, die Flagge).

Beim Erstellen der Bäume hat sie eine ihr zuvor unbekannt Technik in Blender kennengelernt. Hierfür hat sie erstmal den Verlauf des Stammes und der Äste der Bäume durch Vertices und Edges vorgezeichnet. Danach kam der Skin Modifier zum Einsatz welcher dem Stamm eine Dicke gab. Nun konnte die Dicke an jedem Vertex angepasst werden. Danach sollten noch die Baumkrone erstellt werden. Dabei wurde im Sculptmode Spheres zu natürlich aussehenden Baumkronen geformt.

Eine Schwierigkeit bestand darin die Texturen so auf die Gameobjekte zu legen dass der

Übergang im Spiel nicht auffällt. Das heißt die Stellen wo die Ränder der Texturen sein würden sollten gut versteckt sein.

Alex missbrauchte dann auf eigene Faust noch die erstellten Bäume und veränderte diese im Spiel, wodurch es nun graue Bäume gab, welche als Steinklippen fungieren sollten. Dafür war jedoch kein Blender oder andere 3D Modellierung erforderlich, sondern eher ein versehentliches verschieben von Farben auf falsche Oberflächen, während er sich eigentlich um die Gestaltung, Ausarbeitung und der korrekten anzeige der GUI kümmerte, welche zuvor von Olivia bereits grob zusammengebaut wurde.

Des Weiteren hat Niklas das Poster für die Medianight in Photoshop und Illustrator gestaltet. Dabei gab es keine Probleme.



Zudem arbeiteten Niklas und Olivia zusammen an einem Bild welches als Titelbild für das Projekt im MediaNight Flyer dienen sollte.



2.3 Medianight

Das Finale des Projekts, war die Medianight. Hier wurde das Projekt der Öffentlichkeit vorgestellt. Bei der Medianight war das Spiel schon fertiggestellt, sodass es problemlos vorgestellt werden konnte.

Die Resonanz auf das Spiel war hierbei sehr positiv, sodass wir uns auch vorstellen können, an diesem Programm weiterzuarbeiten und es zu verbessern.

Da es einige Interessenten für das Spiel gab, wurde ein Downloadlink erzeugt und dieser dann verteilt. Wodurch das Spiel nun kostenfrei als Demo erhältlich ist.

3. Projektmanagement

Wie bereits im vorherigen Abschnitt erwähnt, stand das Team in permanentem Austausch, da auch sehr häufig im selben Raum gearbeitet wurde. Oft wurde auch einfach an einem Rechner gearbeitet, sodass alle zusammen versucht haben ein gemeinsames Problem zu lösen. Die zu Beginn gesteckten Ziele konnten erreicht werden und zusätzliche Elemente erstellt werden, wie z.B. ein kleines Balancing der Einheiten, der Erstellung von Partikeleffekten oder der Erstellung einer Mühle.

Bei der zeitlichen Einteilung der Teilziele wurde zu Beginn allerdings relativ schnell vom Plan abgewichen, da besonders die Einarbeitungszeit unterschätzt wurde. Da dadurch auch der Fortschritt zunächst nur sehr zäh war, wurden die Anfangsziele zunächst nicht erreicht. Allerdings wurde für spätere Ziele deutlich mehr Zeit eingeplant, als später benötigt wurde, sodass anders als nach ungefähr der Hälfte der Zeit erwartet, alle Ziele erreicht werden konnten, die zu Beginn vorgenommen wurden.

4. Lernerfolg

Als Gruppe haben wir gelernt, Aufgaben systematisch im Team anzugehen und diese mit einem bestimmten Gesamtziel auch umzusetzen. Sehr wichtig war hierfür immer der kontinuierliche Austausch, was den Stand des Fortschritts betrifft, um sich auch nach dem Fortschritt der anderen Aufgabenfelder richten zu können.

Alex Lernerfolg zeigt sich vor allem im Netzwerkbereich. Das erstellen von Netzwerkapplikationen sowie dem synchron von wichtigen Dingen über dieses Netzwerk. Des Weiteren wurde sich intensiv mit Unity und der Sprache C# beschäftigt. Dadurch können zukünftige Projekte nun um einiges schneller angegangen werden und umgesetzt werden. Weiteres wurde im Laufe des Projektes die Dokumentation im Quellcode um einiges verbessert, wodurch es für alle übersichtlicher blieb.

Olivia hat allgemein an Programmiererfahrung gewonnen indem sie zum einen neue Probleme gelöst hat und zum anderen das erste Mal mit C# gearbeitet hat. Zudem hat sie sich komplett neu in Unity einarbeiten müssen und somit in eine zuvor völlig fremde Arbeitsumgebung. Nun hat sie ein Verständnis vieler Funktionen von Unity. Des Weiteren hat sie erste praktische Erfahrung in der gemeinsamen Konzeptionierung und Durchführung eines größeren Projektes gesammelt. Außerdem hat sie ihre Kenntnisse in Blender erweitert und gelernt wie man Objekte richtig Texturen gibt. Nicht zuletzt hat sie gelernt den Überblick über Termine zu bewahren, sodass das Team immer rechtzeitig die notwendigen Materialien abgeben konnte.

Niklas hat sich vor Allem in Maya und Substance Painter sehr stark verbessert. Es fällt ihm

nur einfach, anatomisch korrekte Modelle zu erstellen und diese zu texturieren. Außerdem kann er nun in Unity diese Modelle korrekt einbinden, sodass die Programmierer mit den Modellen auch arbeiten können. Durch den Austausch mit den Programmieren, hat er allerdings auch viel über Unity an sich gelernt, wie eine Netzwerkverbindung grundlegend in Unity erstellt wird, oder wie eine Game Engine an sich funktioniert.

5. Mögliche Weiterarbeit

Dadurch, dass wir noch einige Ideen haben, wie das Spiel verbessert werden könnte, könnten wir und unter Vorbehalt vorstellen, das Spiel tatsächlich weiterzuführen. Mögliche weitere Funktionen wären zum Beispiel ein neuronales Netz für das Balancing, oder Animationen für die Einheiten im Spiel.

Außerdem wäre ein Ki- Mitspieler welcher Variabel in die Lobby hinzugefügt werden kann sehr von Vorteil. Diesen würden wir ebenfalls über neuronale Netze realisieren.

6. Fazit

Zusammenfassend hat das Projekt für alle Teammitglieder sowohl Fortschritte im graphischen, als auch im programmiertechnischen Bereich bedeutet, da nicht nur viel in der Einzelarbeit gelernt wurde, sondern durch die Erklärungen der Arbeitsabläufe innerhalb des Teams und die Zusammenführung in Unity musste auch jeder die Grundkenntnisse für beide Bereiche kennen.

Sowohl in diesen Bereichen, als auch im Bereich des Projektmanagements und des Zeitmanagements wurden Fortschritte erreicht, sodass zukünftige Projekte sehr viel einfacher zu absolvieren sein werden.

Besonders viel lernen konnten wir durch die aufgekommenen Probleme, die zum Ende des Projekts auch zum Großteil gelöst werden konnten.

Außerdem war die Medianight ein toller Abschluss, um das Projekt den anderen Studierenden vorstellen zu können, was unter anderem durch die sehr positive Resonanz ein sehr schöner (vorläufiger) Abschluss des Projekts war.

Bedanken wollen wir uns außerdem bei Herrn Prof. Kriha, der uns jederzeit für Frage und Anregungen zur Seite stand.