



Hochschule für Angewandte Wissenschaften Hamburg

Fakultät Design, Medien und Information

Department Medientechnik

Studiengang Media Systems

Mediengestaltung 3

WiSe 2014/15

Christopher Giese	2114039
Lars Krafft	2126500
Laura Gohl	2115703
Robin Ladiges	1978865

Dozent

Prof. Gunther Rehfeld

Inhalt

1. Konzeption.....	4
1.1 Spielziel.....	4
1.2 Setting.....	4
1.3 Zielgruppe.....	4
2. Recherche.....	6
2.1 Marktanalyse.....	6
2.2 Markenrechte.....	7
2.3 Plattform.....	8
3. Game Design.....	10
3.1 Game Loop.....	10
3.2 Türme.....	10
3.3 Gegner.....	12
4. Gestaltung.....	14
4.1 Benutzerinterface.....	14
4.2 Türme und Raumschiffe.....	18
4.3 Sounds und Musik.....	19
4.4 Wort-Bild-Marke.....	20
5. Teamarbeit.....	23
5.1 Arbeitsaufteilung.....	23
5.2 Aufwandseinschätzung.....	24
5.3 Projektplan.....	25
5.4 Projektverlauf.....	25
5.5 Einzelleistungen.....	27
6. Anhang.....	29
6.1 Quellenverzeichnis.....	29
6.2 Spielen.....	30

1. Konzeption

1.1 Spielziel

In Deathstar PD werden die Gebäude des Spielers von vielen Gegnern gleichzeitig angegriffen. Der Spieler muss alle Gegner, welche in Wellen angreifen, besiegen, um zu verhindern, dass seine Primärgebäude vernichtet werden. Sollte eines der Primärgebäude zerstört werden, verliert der Spieler. Gelingt es dem Spieler alle Gegnerwellen abzuwehren, hat er gewonnen.

1.2 Setting

Das Spiel ist ein Tower Defense Spiel, also ein Strategiespiel, bei dem der Spieler Türme baut, welche für ihn gegen die Gegner kämpfen.

Vom Setting her orientiert sich Deathstar PD an den ersten Star Wars Film [1], bei dem der Todesstern (engl: Death Star), eine riesige runde Raumstation im Weltraum, von Rebellen in kleinen Raumschiffen angegriffen und in die Luft gesprengt wird. Der Spieler übernimmt die Kontrolle über den Todesstern und muss diesen gegen die angreifenden Rebellen verteidigen, indem er auf dessen Oberfläche Türme um kritische Angriffsziele herum baut. Die Primärziele der Rebellen sind die Lüftungsschächte, welche zum Hauptgenerator führen, und die Kommandozentrale, in der sich der Spieler, der Kommandant der Raumstation, physikalisch befindet.

Anders als bei den meisten anderen Tower Defense Spielen werden Türme in Deathstar PD nicht auf einer zweidimensionalen Ebene gebaut, sondern auf einer Kugeloberfläche im dreidimensionalen Raum - es entsteht ein neues Subgenre. Deswegen endet der Projekttitel auf PD für Planetary Defense, als Kontrast zur gebräuchlichen Abkürzung TD für Tower Defense.

1.3 Zielgruppe

Deathstar PD soll ein herausforderndes Spiel sein, das sich an Core- und Hardcore-Spieler richtet. Es ist also kein kurzweiliges Tower Defense Spiel, das man nach fünf bis zehn Minuten Spielzeit beendet hat, sondern eines, für das man, mit 20 bis 30 Minuten für einen Durchlauf, etwas mehr Zeit benötigt.

Diese Zielgruppen fordern herausfordernde Spiele mit einem entsprechend hohen Komplexitätsgrad. Dieser entsteht einerseits dadurch, dass es mehrere Primärgebäude, statt eines Einzelnen, gibt, auf die der Spieler ständig achten muss - was nicht ganz so einfach ist, wenn man aufgrund des sphärischen Planeten nicht alles gleichzeitig im Blick behalten kann. Andererseits unterscheiden sich die Spielmechaniken von Turm zu Turm und von Gegner zu Gegner, wodurch in Kombination gegenseitig komplexere Abhängigkeiten und Rückkopplungen entstehen.

Auch in der Ressourcenausgestaltung steckt Komplexität, statt einem einzigen Rohstofftyp zu haben, der zum Bauen benötigt wird, existieren in Deathstar PD zwei Rohstofftypen, die sich in ihrer Mechanik unterscheiden. Geld erhält der Spieler durch das Besiegen von Gegnern. Der begrenzte Energievorrat kann durch das Bauen von Kraftwerken erhöht werden. Entscheidet sich der Spieler einen Turm abzureißen, erhält er 100% der in das Gebäude gesteckten Energie zurück, aber nur einen Bruchteil des investierten Geldes.

Hat ein (Hardcore-)Spieler erfolgreich alle Wellen überlebt, das Spiel also gewonnen, besteht der Reiz, das Spiel erneut zu spielen, darin, das Spiel besser, schneller und effektiver zu überleben. Das Wiederspielpotenzial ist dadurch gegeben, dass die Primärgebäude und Gegner bei jedem neuen Spiel zufällig positioniert werden - jede Spielrunde wird dadurch einzigartig.

2. Recherche

2.1 Marktanalyse

Wenn man etwas Neues schaffen will, sollte man sich erst einmal anschauen, was es schon gibt, um zu sehen was man besser oder anders machen will. Dabei orientiert man sich an Dingen, die so ähnlich sind, wie das, was man machen möchte. Zunächst betrachtet man solche Dinge, die man bereits kennt, bevor man neue Dinge recherchiert. Die Spiele, die dem ähnlich sind was Deathstar PD sein soll, und die dessen Entwicklern bereits bekannt waren, sind Spore und Planetary Annihilation.

Die Weltraumphase von Spore [2] hat zu großen Teilen zur Entstehung der Idee des Bauens auf einer dreidimensionalen Kugel beigetragen. In Spore terraformt und kolonisiert man unter anderem kleine bunte Planeten. Hin und wieder greifen zufällig Piraten die Kolonien des Spielers an, die wiederum von den Verteidigungstürmen der Kolonien zurückgeschlagen werden. Innerhalb einer Weltraumkolonie können Gebäude und Verteidigungstürme in Spore jedoch nur auf fest vorgegebenen Baupositionen errichtet werden - eine Restriktion, die so in Deathstar PD nicht vorkommen soll.

Es ist nicht so trivial wie man zunächst annimmt eine Kugeloberfläche komplett mit einem Bauraster zu überziehen, ohne dass sich das Bauraster unschön verzerrt. Beispielsweise bei einem rechteckigen Raster, wie es auf der Erdoberfläche durch Längen- und Breitengrade üblich ist, verzerrt sich das Raster zu den Polen hin immer stärker. Auch andere regelmäßige Kugelparkettierungen mit höheren Vielecken verzerren sich zwangsläufig, wenn man nicht platonische oder archimedische Körper verwenden will. Deshalb verwendet Deathstar PD gar kein Bauraster, sondern erlaubt eine freie Platzierung der Gebäude und Türme, lediglich mit der Restriktion eines Mindestabstandes.

Planetary Annihilation [3] ist ein Strategiespiel, bei dem der Spieler in einem Sonnensystem einen Planeten nach dem anderen erobert. Auch wenn einige Verteidigungstürme existieren, werden diese nur selten eingesetzt, weil Boden-, Luft- und Marineeinheiten aufgrund ihrer Mobilität vielfältiger einsetzbar sind. Auf metallenen Planeten besteht die Möglichkeit eine gigantische Laserkanone zu bauen, mit der es möglich ist, wie in Star Wars mit dem Todesstern, andere Planeten zu vernichten. Vor allem optisch ist Planetary Annihilation ein hoch gegriffenes Vorbild für die zukünftig mögliche Planetengestaltung.

Spore und Planetary Annihilation sind beides keine Tower Defense Spiele, weshalb mittels der Google-Bildersuche explizit nach solchen gesucht wurde, die auf einer Kugeloberfläche spielen

(Suchbegriff: "3D Tower Defense Sphere"). Gefunden wurden die beiden Spiele Star Defense und Colony Defense.

Das iOS exklusive Star Defense [4] spielt auf kubischen und sphärischen Oberflächen, bietet aber, abgesehen von dem dadurch eingeschränkten Blickfeld, keine Neuerungen im Vergleich zu üblichen Tower Defense Spielen, weil die Gegner, die in Wellen angreifen, sich auf einem fest vorgegebenen Weg, den sie ablaufen, beschränken.

Auch Colony Defense [5] für Xbox 360 und Windows setzt auf einen kugelförmigen Planeten mit vorgegebenen Laufwegen, erweitert das Konzept aber um mehrere Primärziele, Spawnpunkte und Weggabelungen. Auch wenn das Spiel für Windows erhältlich ist, merkt man der PC-Demo [6] deutlich an, dass es Primär für die Xbox 360 und damit für eine Controllersteuerung entwickelt wurde, weil die Bedienung mit Maus und Tastatur katastrophal bis gar nicht umgesetzt wurde.

Feste Laufwege der Gegnerwellen sind etwas, das vielen Tower Defense Spielen eigen ist. Die üblichste Abweichung von diesem Konzept besteht bei solchen Spielen, bei denen man Türme in den Laufweg der Gegner platzieren kann, wodurch diese dann einen anderen Weg bestreiten oder, falls keiner vorhanden ist, die Türme des Spielers angreifen, bevor sie zum Primärziel vorrücken. Deathstar PD löst sich komplett von dem Gedanken der Laufwege, weil es auf einer dreidimensionalen Kugeloberfläche im Vergleich zur zweidimensionalen Ebene wenig Neues bietet. Insbesondere im Science Fiction Setting mit Raumschiffen im Weltraum macht es überhaupt keinen Sinn, dass sich die Gegner an vorgegebene Wege halten.

2.2 Markenrechte

Bei einem Projekttitel wie Deathstar PD, der sich deutlich auf einen bestehenden Begriff bezieht, ist eine Markenrecherche unumgänglich, um die Verletzung eventuell bestehender Markenrechte Dritter auszuschließen.

Laut dem Deutschen Patent- und Markenamt (DPMA) existiert sowohl eine Wortmarke als auch eine Wort-/Bildmarke zu dem Begriff "Death Star" [7]. Glücklicherweise beschränkt sich der Schutz der beiden Marken des taiwanesischen Inhabers Micro-Star International (MSI), der Computerhardware herstellt, bezogen auf Computersoftware lediglich auf solche Software, welche aus Betriebssystemprogrammen besteht, was für Deathstar PD, als Computerspiel, auszuschließen ist.

Überraschenderweise bezieht sich laut DPMA keine der 277 eingetragenen Marken der Inhaber Lucasfilm Ltd. LLC und Lucasfilm Entertainment Company Ltd. LLC auf den Begriff "Death Star".

Der dänische Spielwarenhersteller LEGO A/S vertreibt ein Produkt mit der Bezeichnung Death Star™ [8]. Mit ™ werden im angloamerikanischen Markenrecht unregistrierte Warenmarken gekennzeichnet, die im Gegensatz zu registrierten Warenmarken (die mit ® gekennzeichnet werden) noch nicht amtlich registriert sind und deshalb einen geringeren Rechtsschutz besitzen. Für den deutschen Rechtsraum gewährt die Kennzeichnung mit ™ keinen rechtlichen Vorteil.

Um kein unnötiges Prozessrisiko einzugehen, empfiehlt es sich, vor einer kommerziellen Nutzung von Deathstar PD, eine Wortmarke für Computerspiele beim DPMA einzutragen.

2.3 Plattform

Die erste Idee war es, das Spiel für WebGL kompatible Browser mit der JavaScript 3D Library Three.js umzusetzen, um Spielern einen einfachen Zugang zum Spiel zu ermöglichen, der nicht durch benötigte Addons oder das Installieren einer Anwendung erschwert wird. Nachdem der erste Prototyp noch recht einfach zu realisieren war, wurde schnell klar, dass ein richtiger Editor sehr viel geeigneter ist, um visuell ansprechende Spiele zu entwickeln als reiner Programmcode und so fiel die Entscheidung Unity® für die Erstellung von Deathstar PD zu nutzen.

Begünstigt wurde diese Entscheidung durch die Anpassung der Zielgruppe, während zuerst Arbeitnehmer in ihrer Mittagspause als Hauptgruppe anvisiert wurden, änderte sich dies auf Grund unserer umfangreichen und komplexen Spielmechaniken zu einer Ausrichtung auf "Core-Gamer", also Menschen mit großer Erfahrung mit PC-Spielen. Da erfahrene Spieler meist auch besser und erfahrener mit ihrem PC umgehen können, ist die Installation eines Browser-Addons oder der Download einer Datei kein so großes Hindernis mehr. Außerdem nutzen "Core-Gamer" zum Spielen oft ihren eigenen Computer und keine von ihrem Arbeitgeber gestellten Rechner. Sie besitzen deshalb alle notwendigen Benutzerrechte, um Deathstar PD installieren zu können.

Ein weiterer wichtiger Punkt für Unity waren die Partikeleffekte. Unity bietet eine wirklich einfache Benutzeroberfläche um Blitz- und Explosionseffekte zu erstellen. Mit reinem JavaScript-Code wären solche optischen Highlights nur schwer bis gar nicht umzusetzen gewesen.

Gegen Unity sprachen zuerst die Probleme mit der Versionskontrolle. Bei einer Teamarbeit mit mehreren Programmierern kann es ohne ein solches Verwaltungstool schnell zum ungewollten Überschreiben von Dateien kommen und ein effektives Arbeiten wäre nur schwer möglich gewe-

sen. Gelöst wurde dieses Problem durch das Einbinden eines Addons für Unity, das die einfache Verbindung der Unity-Entwicklungsumgebung mit SVN erlaubt. Eine gewisse Eingewöhnungszeit war zwar von Nöten und es lief auch mit diesem Addon nicht alles glatt, im Endeffekt stellte es aber eine erhebliche Erleichterung der Arbeit dar.

Nachdem einige Stunden investiert wurden, um sich mit Blender vertraut zu machen, wurde klar, dass mit diesem Programm keine schnellen Erfolge zu erzielen waren. Der Umstieg auf den UVG-Voxel Editor, einem selbstentwickelten einfachen Programm zum Erstellen von 3D-Modellen aus einzelnen Würfel-Blöcken, vereinfachte das Erstellen von Platzhalter-Modellen für Raumschiffe und Türme ungemein. Es war jedoch ein Problem diese Modelle in Unity zu importieren und dabei die Performance des Spiels nicht negativ zu beeinflussen.

Obwohl ursprünglich nur als Zwischenlösung geplant, gab es für die kantigen Platzhalter-Modelle überraschend positives Feedback, sodass diese Art Modelle nach der Zwischenpräsentation auch im finalen Spiel beibehalten wurden.

3. Game Design

3.1 Game Loop

Der grundlegende Game-Loop (Spielablauf, Handlungskreislauf) von Deathstar PD besteht darin, dass der Spieler Türme baut und Gegner in Wellen spawnen, welche wiederum automatisch mit den Gebäuden des Spielers kämpfen.

Das Bauen der Türme konkretisiert sich durch die dafür notwendigen Rohstoffe: Geld und Energie. Das Erlangen von Energie erfolgt durch das Bauen von Kraftwerken, die keine Energie zum Bauen benötigen, sondern lediglich Geld. Geld wird im Kampf mit den Gegnern, durch das Besiegen jener, erlangt. Damit der Spieler unter diesen Beschränkungen überhaupt an Rohstoffe gelangen kann, um Türme bauen zu können, die für ihn Gegner vernichten, um an Geld zu gelangen, benötigt er ein Startkapital. Verbaut sich der Spieler, oder möchte seine Turmauswahl nachträglich verändern, kann er Türme abreißen, um einen Anteil des investierten Geldes und der gesamten investierten Energie zurückzuerlangen.

Weil die Gegner immer stärker werden, die Anzahl Türme, durch die begrenzten Rohstoffe, begrenzt ist und die maximale Turmdichte durch den Bauradius bestimmt wird, kommt zusätzlich zum Bauen von Türmen die Option hinzu vorhandene Türme auszubauen, damit sie effektiver werden. Angriffsgeschwindigkeit, Reichweite, Schaden pro Treffer und Lebenspunkte erhöhen sich.

3.2 Türme

Um das Spiel interessant zu machen sind verschiedene Typen von Türmen unerlässlich. Diese unterscheiden sich in ihren Attributen und besonderen Fähigkeiten, sowie besonders den Kosten stark, wodurch der Spieler gezwungen wird zwischen den Vor- und Nachteilen zu entscheiden. Um es Kennern des Tower Defense Genres einfacher zu machen sich in das Spiel einzufinden, sind die Türme den genreüblich bekannten Varianten angepasst gestaltet.

MG-Turm

Um das Spiel interessant zu machen sind verschiedene Typen von Türmen unerlässlich. Diese unterscheiden sich in ihren Attributen und besonderen Fähigkeiten, sowie besonders den Kosten stark, wodurch der Spieler gezwungen wird zwischen den Vor- und Nachteilen zu entscheiden. Um es Kennern des Tower Defense Genres einfacher zu machen sich in das Spiel einzufinden, sind die Türme den genreüblich bekannten Varianten angepasst gestaltet.



Bild 3.2a MG-Turm

Laserturm

Der Laserturm ist teurer als der MG-Turm, dafür verursacht er mit seinem durchgehenden Strahl hohen Schaden. Dieser Turm ist dafür gedacht, direkt an den Primärgebäuden gebaut zu werden, sowie später, statt den MG-Türmen, das umliegende Gelände zu sichern. Um ihn in der Anfangsphase in ausreichender Stückzahl zu bauen ist er zu teuer.



Bild 3.2b Laserturm

Blitzturm

Der Blitzturm ist mit seiner hohen Feuerrate und der Spezialfähigkeit, mehrere Gegner gleichzeitig anzugreifen, ideal zur direkten Verteidigung der Primärziele. Allerdings ist er kostenintensiver als der Laserturm und hat eine geringere Reichweite, weshalb der Spieler ihn erst im späteren Verlauf des Spiels in größerer Anzahl bauen kann.



Bild 3.2c Blitzturm

Raketenturm

Der Raketenturm unterscheidet sich von den vorigen Türmen. Einerseits kostet er mehr Geld, benötigt doppelt so viel Energie und hat die geringste Feuerrate, andererseits verfolgen die Raketen ihr Ziel über die Feuerreichweite des Turms hinaus und verursachen pro Treffer den größten Schaden von allen Türmen. Da dieser Schaden dazu noch auf einen größeren Bereich um den getroffenen Gegner herum wirkt (Flächenschaden), ist der Raketenturm als äußere Verteidigung abseits der Primärgebäude gedacht. Der Spieler kann im späteren Verlauf des Spiels einige dieser Türme bauen, um die Gegner schon vor ihrem Eintreffen beim Primärziel zu schwächen.

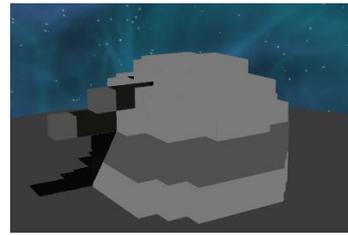


Bild 3.2d Raketenturm

Kraftwerk (Unterstützungsturm)

Das Kraftwerk unterscheidet sich grundlegend von den anderen Türmen und ist auch ansonsten eher untypisch für das Tower Defense Genre. Als teuerster Turm im Spiel kann der Spieler es sich erst spät leisten, dafür verstärken Kraftwerke andere Türme in ihrem Umkreis. Solche Unterstützungstürme gibt es zwar öfters in TD-Spielen, allerdings generieren Kraftwerke zusätzlich Energie, die zum Bau der anderen Türme benötigt wird. Damit ist der Spieler ab einem gewissen Punkt gezwungen, Kraftwerke zu bauen, wobei er bei der Positionswahl auch den Verstärkungseffekt bedenken muss.



Bild 3.2e Kraftwerk

3.3 Gegner

Fighter

Der Fighter ist der Angreifer, welcher am häufigsten vorkommt. Er ist quasi das "Standardmodell", auf den der Spieler direkt zu Beginn trifft und aus welchem sich die anderen angeifenden Raumschiffe entwickelt haben. Der Fighter stellt in den Punkten Geschwindigkeit, Feuerstärke und Lebenspunkte den Durchschnitt dar. Der Fighter greift nur die vom Spieler zu beschützenden Primärgebäude an.



Bild 3.3a Fighter

Mini-Fighter

Der Mini-Fighter ist eine abgeschwächte Version des normalen Fighters. Diese Einheit hat im Vergleich zum Fighter eine geringere Feuerstärke, weniger Lebenspunkte und eine geringere Größe. Stattdessen ist der Mini-Fighter schneller und tritt in größerer Anzahl auf, um dadurch eine Herausforderung für den Spieler darzustellen. Der Mini-Fighter greift nur die vom Spieler zu beschützenden Primärgebäude an.



Bild 3.3b Mini-Fighter

Bomber

Der Bomber ist langsamer und größer als Fighter und Mini-Fighter. Er hat stattdessen eine überdurchschnittlich große Feuerkraft und viele Lebenspunkte. Der Bomber ist der Gegner mit dem der Spieler als letztes konfrontiert wird und bietet eine neue Herausforderung für den Spieler, denn im Gegensatz zu den anderen Einheiten, greift der Bomber nicht nur die Primärgebäude des Spielers an, sondern auch alle vom Spieler gebauten Türme auf dessen Weg.

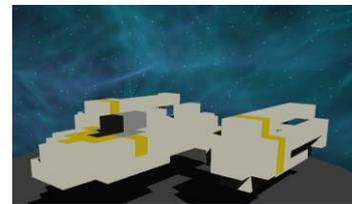


Bild 3.3c Bomber

4. Gestaltung

4.1 Benutzerinterface

Die GUI, das Benutzerinterface, ist ein ausschlaggebender Faktor für den ersten Eindruck auf den Spieler, die Spielbarkeit selber sowie das "Feeling", dem einheitlichen Stil des Spiels. Der erste Eindruck hängt hierbei nicht nur mit dem Design, sondern auch besonders mit der Anordnung und Komplexität der einzelnen GUI-Elemente zusammen.

Um diesen Punkt frühzeitig zu behandeln, ist das Design einigen Veränderungen vom ersten Entwurf bis zur finalen Version durchlaufen. Von Anfang an war jedoch klar, dass die Elemente keine "harten" 90° Ecken haben sollen. Im ersten Entwurf wurden diese durch zwei 45° Winkel ähnlich einer abgeschnittenen Ecke ersetzt, wodurch die Elemente weder rund noch allzu kantig wirkten. In diesem Stil gab es für die Anordnung der Elemente zwei Varianten.



Bild 4.1a: Erster Entwurf der GUI ähnlich zu Planetary Annihilation

Die erste Variante ist angelehnt an das Design aus Planetary Annihilation, wobei die Buttons zum Bauen von Türmen in einer Leiste an der Unterseite des Bildschirms angebracht sind. Ein Informationsfenster mit den Werten und der Beschreibung des ausgewählten Turms befinden sich in der linken unteren Ecke, ein Fenster mit einer Modellvorschau und Beschreibung in der rechten unteren Ecke. Ein Panel mit der Übersicht über die Ressourcen und Gegner sowie der Rückansicht befindet sich in der oberen rechten Ecke des Bildschirms.

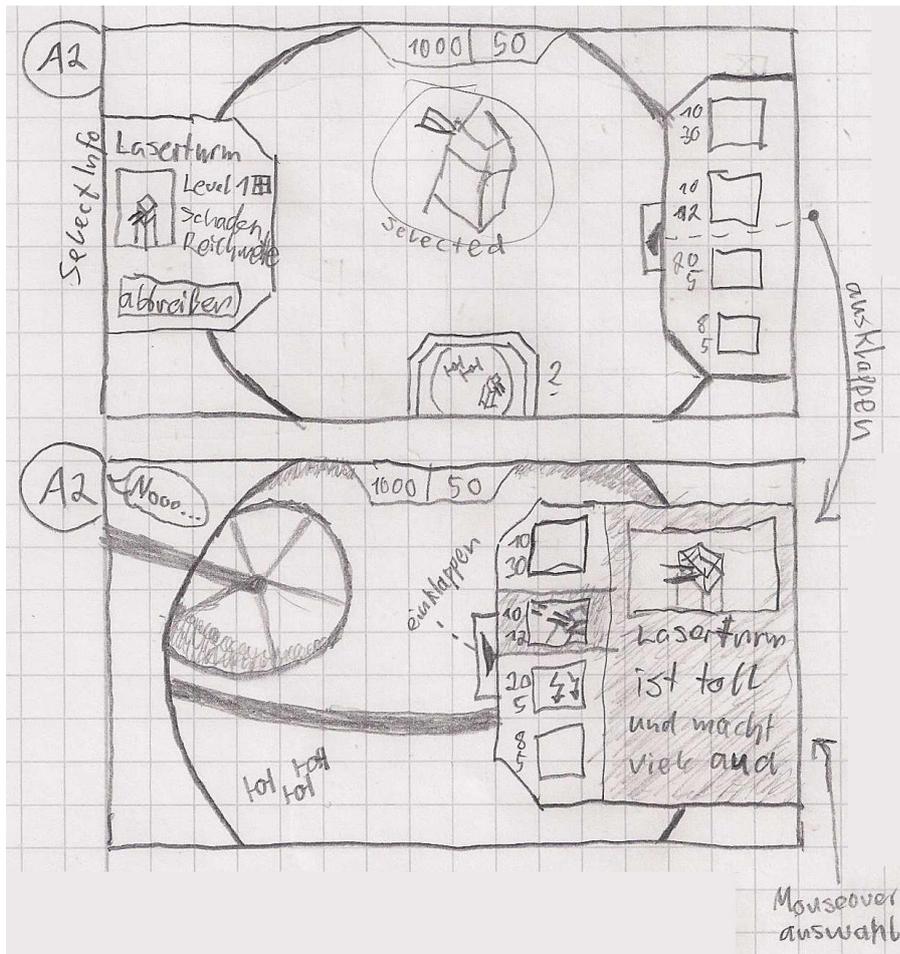


Bild 4.1b: Zweiter Entwurf der GUI im klassischeren Stil

Hierbei sind die Buttons ähnlich vieler TD-Spiele in einem großen Panel an der Seite des Bildschirms untergebracht. Um den Spieler nicht auf Dauer mit den Panels zu überladen, können diese mit einem kleinen Button ein- und wieder ausgefahren werden. Auf der linken Seite sind Informationen zum gerade ausgewählten Turret sowie die Upgrade- und Verkaufsmöglichkeit, auf der rechten Seite sind Buttons zum Bauen sowie Beschreibungstext und Vorschaubild. Das Panel zu Ressourcen und Gegnern ist am oberen Bildschirmrand mittig angebracht.

Da dieses Design allerdings keinen guten Platz für die Rückansicht des Planeten bot, wurde es wiederum verworfen und das dritte Design entwickelt. Ab hier sollte die GUI möglichst aus runden Elementen bestehen oder abgerundete Ecken haben, um zu der Spielbesonderheit des runden Planeten zu passen.

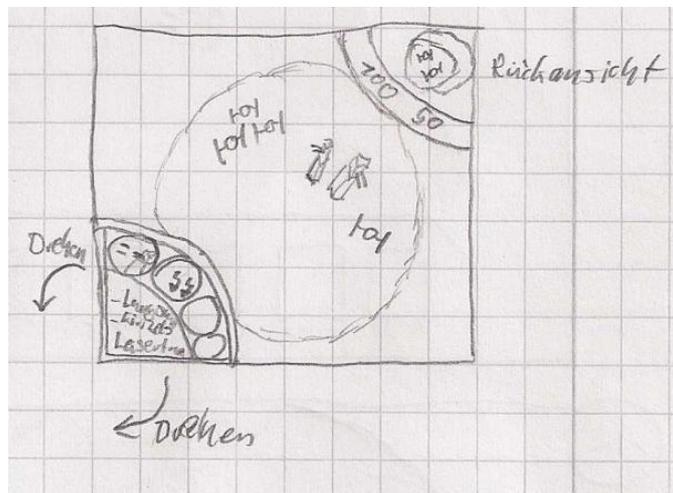


Bild 4.1c: Finaler Entwurf der GUI

Die Übersicht über die Ressourcen ist nun in die obere linke Bildschirmecke gesetzt, die Übersicht über die Gegner in die obere rechte Ecke. Die Rückansichtkamera ist als einzelnes Element in der unteren rechten Ecke platziert. Einzig die Bau- und Upgradeoptionen haben sich noch bis zum finalen Stand verändert. Der ersten Idee nach sollte das kreisrunde Baupanel in der linken unteren Ecke standardmäßig die Buttons zum Bauen der Türme sowie die Beschreibung anzeigen. Sobald ein Turm auf dem Spielfeld ausgewählt wird, rotiert das Panel um 90° und beinhaltet die Attributsinformationen sowie die Abriss- und Upgrade-Buttons. Da diese Drehung technisch problematisch ist, wurde die Idee verworfen und das Panel wird nur noch zum Bauen verwendet.



Bild 4.1d: Finales Interface im Spiel

Beim Überfahren der Buttons zum Bauen mit der Maus werden Baukosten und Beschreibung angezeigt. Wenn nicht genügend Ressourcen verfügbar sind, werden die Buttons ausgegraut. Bei der Auswahl eines Turms auf dem Planeten erscheint ein Overlay-Interface, welches die momentanen Werte des Turms anzeigt und die Möglichkeit bietet den Turm abzureißen oder aufzugraden.

Da kein Gruppenmitglied über große illustratorische Begabung verfügt, sind die GUI-Elemente zwar in einem einfachen, aber zueinander und dem Spiel passenden Stil gehalten.

4.2 Türme und Raumschiffe

Der Stil und die Assets sollten thematisch, wie der Name schon indiziert, in erster Linie an Star Wars orientiert sein.

Zunächst war die Idee, für die Modelle Blender zu verwenden. Da leider kein Mitglied der Gruppe über entsprechend ausführliches Vorwissen im Bezug auf dieses Programm hatte, musste sich dort zunächst eingearbeitet werden.

Damit trotzdem parallel auch an dem Game in Unity gearbeitet werden konnte, wurden Platzhalter-Modelle im UVG-Editor, ein von Lars Krafft in einem anderen Projekt selbst geschriebener Voxel-Editor, erstellt.

Der simple Stil der UVG-Modells überzeugte und daher wurde entschieden, diesen simplen Stil beizubehalten und nicht, wie vorgesehen, komplexere Modelle in Blender zu erstellen. Auch die Einfachheit im Umgang mit dem UVG-Editor, besonders im Vergleich zu Blender, spielten dabei eine Rolle.

Allerdings brachte die neue Software auch gewisse Probleme mit sich. Da der UVG-Editor "nur" ein Studentenprojekt und keine kommerzielle Software ist, ist er nicht besonders optimiert. Das bedeutet, es dürfen keine zu großen Modelle, also Modelle mit zu vielen Voxeln erstellt werden, da sonst das Arbeiten auf Grund der Performanzprobleme stark erschwert wird.

Die Modelle durften also nicht besonders groß werden. Dies schränkte natürlich auch den Detailreichtum ein und es war eine Herausforderung, trotzdem stilistisch erkennbar an Star Wars anzuknüpfen.

Zunächst war die Vermutung, dass es schwieriger sein würde, die angreifenden Raumschiffe zu designen, als die verteidigenden Türme, da Raumschiffe prinzipiell komplexer aufgebaut sind und man im Normalfall weniger Erfahrung im Aufbau von Raumschiffen als Türmen hat.

Es stellte sich allerdings heraus, dass aufgrund der Varietät der bereits im Star Wars-Universum vorhandenen Raumschiffdesigns, welche sich leicht als Inspiration nutzen ließen, das Designen zum Glück nicht ganz so schwierig war.

Tatsächlich musste beim Designen der Türme mehr Kreativität aufgebracht werden. Wo es bei den Raumschiffen das größte Problem war, ein glaubwürdiges Modell aus recht wenigen Voxeln zu erschaffen, welches erkennbar an Star Wars angelehnt ist, bestand bei den Türmen die Schwierigkeit darin, glaubwürdige Turmmodelle im Stil von Star Wars zu entwerfen, da das Universum in diesem Fall doch recht wenig abwechslungsreiche Vorlagen bietet.

Im Grunde gibt es auf dem "echten" Todesstern nur die von uns genannten MG-Türme und Laserrtürme zur Abwehr naher Feinde. Auf den großen populären Todesstrahl (eine riesige Satellitenschüssel) wurde verzichtet, weil ein solch großes Modell einen Großteil des Planeten einnehmen würde und das Planetenmodell grundlegend hätte verändert werden müssen.

Modelle, wie z.B. Unterstützungs- und Raketentürme mussten komplett neu erdacht und an den Stil angepasst werden.

4.3 Sounds und Musik

Der Sound und die Musik spielten für dieses Projekt eine eher nebensächliche Rolle. Natürlich bringen sie das Spiel auf eine ganz neue Ebene und können den Unterschied zwischen einem guten und einem schlechten Spiel ausmachen, da aber kein richtiger Profi auf dem Gebiet zur Verfügung stand, wurde sich stärker auf andere Bereiche konzentriert und sich beim Sound auch mit wenig zufrieden gegeben. Ein weiterer Grund für diese Herangehensweise ist die angesprochene Zielgruppe: Als „Core-Gamer“ ist die erste Aktion in einem kleinen Spiel wie unserem meist, die Musik auszuschalten und die Lautstärke der Sounds weit herunter zu regeln, um im Hintergrund eigene Musik abzuspielen - etwas, das leider viel zu häufig in kleinen Spielen nicht möglich ist.

Die Sounds entstanden daher meist, wenn es beim Programmieren zeitweilige Probleme gab, bzw. man "grad nicht weiter kam" und ein Ausgleich zu der recht eintönigen Programmierarbeit geschaffen werden musste. Sie wurden mit einem einfachen Headset aufgenommen und mit Audacity soweit verfremdet, dass sie eine angemessene Untermalung für die im Spiel dargestellten Effekte darstellten.

Eine große Hilfe beim Erstellen der Effekte war der Unity-Editor, weil in ihm die Effekte auf Knopfdruck abgespielt werden können und die Sounds so schon in passender Länge aufgenommen werden können. Der charakteristische Star Wars Lasersound wurde leider als einziger nicht selber aufgenommen, da dazu entweder ein über 250 Meter hoher amerikanischer Funkturm oder eine mehrere Meter lange Stahlfeder benötigt wird [9]. Da dieser Sound aber trotzdem verwendet werden sollte, wurden diese Sounds von einer Website [10] mit vielen nachgemachten Sounds, die dort frei angeboten werden, verwendet.

Genau wie die Sounds entstand auch die Musik eher "nebenbei", orientiert sich allerdings nicht an den Originalen aus Star Wars, da die orchestrale Untermalung zum einen für uns nicht umsetzbar war und zum anderen nicht zu dem angestrebten Spielprinzip gepasst hätte. Stattdessen

fiel die Entscheidung auf elektronische Klänge mit viel Hall und sphärischen Tönen um das Science-Fiction-Setting zu unterstreichen.

Das sehr kurze Intro des Hauptmenüs passt durch seine Einfachheit sehr gut zu der Leere des Weltraums im Hintergrund und den simplen Buttons des Menüs, es hinterläßt beim Spieler ein Gefühl der Größe und Leere. Dadurch soll eine angespannte Erwartungshaltung beim Spieler verursacht werden. Theoretisch könnten sich jederzeit Gegner im großem Nichts des Weltraumes verbergen und könnten von überall heraus anrücken.

Die eigentliche Musik während des Spiels ist zwar von den Instrumenten sehr ähnlich gehalten, allerdings wesentlich treibender als die sehr langen Einzeltöne des Intros. Das Stück orientiert sich sehr stark an dem Song „What is Love“ [11], wechselt die Grundakkorde allerdings nur jeden zweiten Takt, um etwas dezenter zu bleiben als das Vorbild und dadurch nicht vom eigentlichen Spielgeschehen abzulenken.

4.4 Wort-Bild-Marke

Ursprünglich war es nicht geplant, eine Bildmarke für Deathstar PD zu entwerfen, aber aus dem Wunsch heraus die ausführbare Programmdatei mit einem eigenen Icon, statt dem Standard Unity-Icon, zu versehen, entwickelte sie sich.

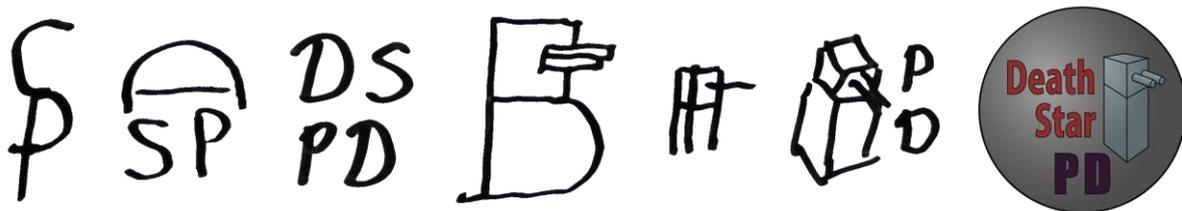


Bild 4.4a: Icon-Entwicklung ausgehend von Whiteboard-Skizzen

Begonnen wurde mit Skizzen am Whiteboard die versuchten mit den Initialen DSPD herumzuspielen, was nicht so gut funktionierte, unter anderem, weil die Initialen für sich selbst stehend nicht sehr einleuchtend sind, wenn man den Titel nicht bereits kennt. Die Kombination mit dem stilisierten Standard MG-Turm war die Folge. In Inkscape [12] wurde dann die erste Vektorgrafik des Icons erstellt, bei dem der volle Titel ausgeschrieben wurde, sowie der MG-Turm und der Kreis einen 3D-Effekt durch Beleuchtung bekamen. Die runde Form des Icons spiegelt die Besonderheit des Spieles wieder, dass auf einer Kugeloberfläche gebaut wird.

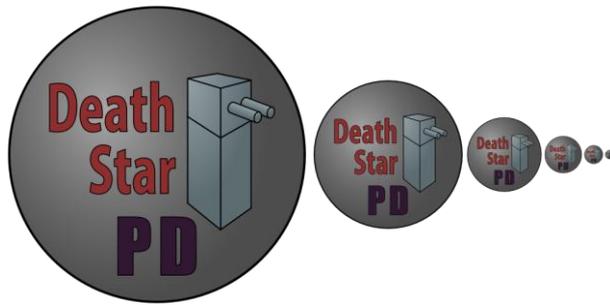


Bild 4.4b: Icon mit 512, 256, 128, 64, 32 und 16 Pixeln Seitenlänge

Insbesondere bei kleineren Auflösungen, wie es für Icons benötigt wird, kam dieses Logo, aufgrund der dünnen Umrisse des Turmes, der relativ zum Gesamtbild kleinen Schrift und der ähnlich grauen Farben, nicht gut rüber. In Ermangelung einer besseren Idee blieb es vorerst für längere Zeit bei diesem Icon.

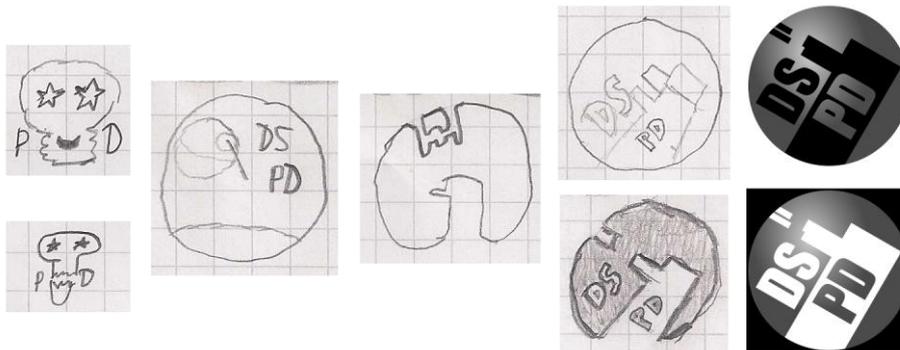


Bild 4.4c: Logo-Entwicklung ausgehend von Papier-Skizzen

Beim Blättern durch ein Buch, das eine Vielzahl verschiedener Logos abbildet [13], fiel der Blick auf runde einfarbige Logos, die dadurch entstehen, dass man vom schwarzen Grundkreis Details "herausstanzt" ("Differenz"-Funktion in Inkscape). Es entstand der Gedanke, den MG-Turm aus dem Planeten herauszustanzten.

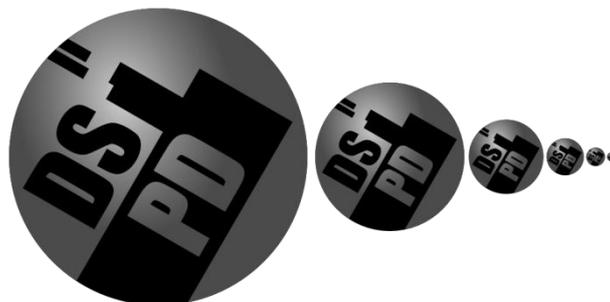


Bild 4.4d: Logo mit 512, 256, 128, 64, 32 und 16 Pixeln Seitenlänge

In kleineren Auflösungen ist das Logo besser erkennbar als das erste Icon. Die graue Hintergrundfarbe verliert durch den Beleuchtungseffekt die Einfarbigkeit, wodurch es sich insbesondere für verschiedenste Desktophintergründe, die vom Icon-Gestalter nicht beeinflusst werden können, besser eignet. Für Einsatzumgebungen, in denen der Hintergrund kontrollierbarer ist, wie z.B. das Hauptmenü, die Webseite oder diese Dokumentation, kommen einfarbige Varianten des Logos als Bildmarke besser zur Geltung.

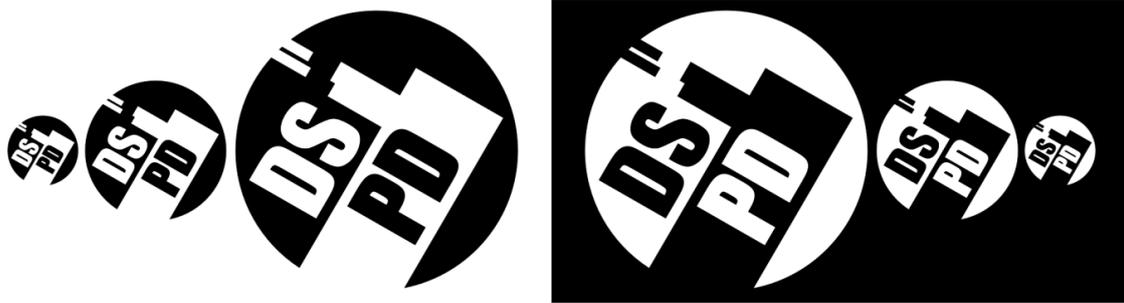


Bild 4.4e: Logo einfarbig in Weiß und Schwarz für dunkle bzw. helle Hintergründe

Hat man eine Bildmarke und wird sie gestalterisch mit dem Titel kombiniert, entsteht eine Wort-Bild-Marke. Durch das Hervorheben der Initialen im Titel erklärt sich deren Verwendung im Logo. Das PD in DeathStar PD erschließt sich uneingeweihten nicht von selbst, weshalb der Begriff Planetary Defense im Begleittext erwähnt werden sollte.



Bild 4.4f: Wort-Bild-Marke in Weiß und Schwarz für dunkle bzw. helle Hintergründe

5. Teamarbeit

Um die Teamarbeit reibungslos zu gestalten, sind, sofern terminbedingt möglich, bei jedem der regelmäßigen Meetings alle Mitglieder anwesend gewesen und hatten bei jedem Thema Mitspracherecht, um möglichst viele Ideen bedenken und ausarbeiten zu können. Zwischen den Meetings wurden E-Mails und Skype als Kommunikationsmittel verwendet, um allgemeine Fragen an das Team oder spezifische Fragen an Einzelne, bei Überschneidungen der einzelnen Arbeitsbereiche, zu klären. Dies ist besonders dann der Fall gewesen, wenn eine Veränderung der Dateien nicht funktionierte oder unerwartete Fehler an anderer Stelle verursachte.

Um das gemeinsame Arbeiten an den Dateien zu gewährleisten wurde SVN als Tool zur Versionsverwaltung genutzt, da es für Einsteiger, durch ein einzelnes Repository, relativ einfach zu bedienen ist und eine schnelle Synchronisation der Dateien bietet.

Zur gemeinsamen und gleichzeitigen Arbeit an der Dokumentation wurde ein Etherpad verwendet.

5.1 Arbeitsaufteilung

Die Arbeitsaufteilung im Team wurde nach relativen bisherigen Erfahrungen getroffen, da ein Projekt dieser Größe nur schwer realisierbar gewesen wäre, wenn jedes Mitglied vollkommen neue und unbekannte Aufgaben übernommen hätte.

Da Robin C. Ladiges am meisten Erfahrung als Programmierer hatte, übernahm er die Rolle des Hauptprogrammierers für Spielmechaniken und K.I. und beschäftigte sich weiterhin mit der Gestaltung des Logos. Er hatte die ursprüngliche Idee für das Spiel, die von allen gemeinsam konkretisiert wurde.

Lars Krafft hatte schon Erfahrung in den Bereichen Sounds, Grafikeffekte und Modellierung, weshalb er für diese Arbeitsbereiche und ihre zugehörige Programmierung verantwortlich war.

Laura Gohl war verantwortlich für das grafische Design inklusive Konzeptzeichnungen der verschiedenen Turm- und Raumschiffotypen sowie deren Modellierung.

Christopher Giese war verantwortlich für das Benutzerinterface und die Projektleitung.

Alle Mitglieder waren am allgemeinen Design des Spielprinzips, inklusive der Ziele, Türme, Gegner und Ressourcen beteiligt.

5.2 Aufwandseinschätzung

Person	Aufgabe	Aufwand
<hr/>		
Robin Ladiges		
	Programmierung (Spiel)	35h
	Programmierung (KI)	25h
	Modellierung	20h
	Kommunikation	30h
	Dokumentation	20h
	Recherche	5h
<hr/>		
Lars Krafft		
	Grafikeffekte	40h
	Programmierung	30h
	Modellierung	20h
	Kommunikation	30h
	Dokumentation	20h
	Recherche	5h
<hr/>		
Laura Gohl		
	Design	10h
	Konzeptgrafiken	35h
	Modellierung	50h
	Kommunikation	30h
	Dokumentation	20h
	Recherche	10h
<hr/>		
Christopher Giese		
	Design	20h
	GUI	35h
	Modellierung	20h
	Projektleitung	10h
	Kommunikation	30h
	Dokumentation	20h
	Recherche	10h

5.3 Projektplan

Datum	Leistung
17.09.	Konzeptmeeting, WebGL-Prototyp
24.09.	Unity-Prototyp
01.10.	Abstrakte Towerklasse
08.10.	Konzeptpräsentation
15.10.	Einarbeitung in Unity und Blender
22.10.	Planet, Grundlegende Programmierung, Modell-Dummies
29.10.	Grundlegende GUI, Spielprinzip/-Aufbau
05.11.	Fertige Programmierung, erste Modelle
12.11.	Weitere Modelle und erste Texturen
19.11.	Weitere Texturen und letzte Modelle
26.11.	Letzte Texturen und Sounds
01.12.	Balancing
10.12.	Finaler Status

5.4 Projektverlauf

Datum	Leistung
17.09.	Konzeptmeeting, WebGL-Prototyp
24.09.	Unity-Prototyp
01.10.	Abstrakte Towerklasse
08.10.	Konzeptpräsentation
15.10.	Meeting, Skybox
22.10.	Turmbau, 2 Turm-Dummies
29.10.	UVG-Importer, 1. Logo
05.11.	Meeting, 1. GUI, Hauptmenü, Pausenmenü, Konzeptgrafiken
12.11.	UVG-Editor, 1. Turm, Laser- und Explosionseffekte
19.11.	Grafikeinstellungen, 2. Turm, Icons, Zwischenpräsentation
26.11.	Meeting, 3. Turm, Effekte, Rückansicht, Sounds
01.12.	Soundeinstellungen, 4. Turm, Primärziele, Turm-Programmierung, 2. Logo
10.12.	Meeting, Showroom, Raumschiffe, Gegner-Programmierung, Ressourcen
17.12.	2. GUI, Balancing, 5. Turm, Abschlusspräsentation

Im Projektplan war vorgesehen, dass für Modelle Blender verwendet wird. Da diese Software eine hohen Einarbeitungszeit erfordert und es insgesamt länger dauert komplexere Modelle zu erstellen, wurde der Fokus zunächst darauf gelegt sich mit den grundlegenden Mechaniken des Programms vertraut zu machen. Durch den Wechsel auf UVG-Modelle war die bereits investierte Einarbeitung überflüssig, aber ein hohes Maß an weiterer notwendigerer Zeit wurde eingespart. Auch mussten keine Texturen für Türme und Gegner mehr erzeugt werden, weil die Färbung der Modelle bereits einfach im UVG-Editor erfolgte. Dadurch ist allerdings auch viel Zeitaufwand für die Programmierung entstanden, welche dringend nötig war.

Im Endeffekt ist das Spiel sogar funktional komplett, statt wie zu Anfang geplant nur ein ausgearbeiteter Prototyp mit guten Modellen und einem Spielprinzip. Einzig das Balancing konnte nicht in größerem Umfang getestet und genau eingestellt werden, was allerdings für diesen Projektumfang ausgereicht hat.

5.5 Einzelleistungen

Person	Aufgabe	Aufwand
<hr/>		
Robin Ladiges		
	Dokumentation	37h
	Kommunikation	27h
	Programmierung (Spiel)	17h
	Programmierung (KI)	15h
	Programmierung (Sonstiges)	15h
	GUI	15h
	Grafik	14h
	Logo	6h
	Recherche	5h
	Sounds	2h
	Gesamt	153h
<hr/>		
Lars Krafft		
	Dokumentation	13h
	Kommunikation	26h
	Sounds	27h
	Musik	5h
	Effekte	16h
	Effekte (Programmierung)	18h
	Models	13h
	Model Import	6h
	Gesamt	124h
<hr/>		
Laura Gohl		
	Dokumentation	5h
	Recherche	12h
	Kommunikation	25h
	Einarbeitung Blender	8h
	Einarbeitung Gimp	6h
	Setup Grafiktablet	3h
	Skizzen	11h
	Artwork	10h
	Modelle	26h
	Gesamt	106h
<hr/>		

Christopher Giese

Dokumentation	15h
Kommunikation	28h
Einarbeitung Paint.net	3h
GUI Entwürfe	12h
GUI Umsetzung	30h
Recherche	6h
Gesamt	94h

6. Anhang

6.1 Quellenverzeichnis

- [1] Film: Star Wars: Episode IV - A New Hope, Lucasfilm, 20th Century Fox, USA, 1977
- [2] Spiel: Spore, Maxis Software, Electronic Arts, 2008,
<http://www.mobygames.com/game/spore>
- [3] Spiel: Planetary Annihilation, Uber Entertainment, 2014, <http://www.uberent.com/pa/>
- [4] Spiel: Star Defense, Rough Cookie, ngmoco, 2009, <http://stardefense.ngmoco.com/>
- [5] Spiel: Colony Defense, Mana Bomb Games Studio, 2010,
<http://www.mobygames.com/game/colony-defense>
- [6] Demo: Colony Defense für Windows, 4Players,
http://www.4players.de/4players.php/download_start/Downloads/Download/54768.html
(Abgerufen am 23.10.2014)
- [7] Marke: Registernummern 012915476 und 012915641, Deutsches Patent- und Markenamt, München, <https://register.dpma.de/DPMAregister/marke/registerhabm?AKZ=012915476> und <https://register.dpma.de/DPMAregister/marke/registerhabm?AKZ=012915641> (Abgerufen am 14.01.2015)
- [8] Artikel: Death Star™, LEGO Shop, LEGO A/S, Artikelnummer 10188,
<http://shop.lego.com/en-US/Death-Star-10188> (Abgerufen am 14.01.2015)
- [9] Video: Slinkies and Star Wars Sound Effects, Adam Micolich, YouTube, 2009,
<https://www.youtube.com/watch?v=aqtqiuSMJqM> (Abgerufen am 16.12.2014)
- [10] Website: Star Wars Sounds, Tsing, The Sa-Matra, <http://www.sa-matra.net/sounds/starwars/> (Abgerufen am 16.12.2014)
- [11] Musik: What is love?, Haddaway, Coconut Records, 1992
- [12] Software: Inkscape, Software Freedom Conservancy, <https://inkscape.org/>
- [13] Buch: Dos Logos, Robert Klanten und Nicolas Bourquin, Die Gestalten Verlag, Berlin, 2004, Seite 155, ISBN 978-3-89955-055-9

6.2 Spielen

Das Spiel kann online mittels Unity® Web Player direkt im Browser gespielt werden oder für Windows und GNU/Linux heruntergeladen werden. Dazu wurde eine Webseite eingerichtet, die unter <https://games.blackpenguin.de/DeathstarPD/> erreichbar ist. Die etwa 20 bis 30 Stunden, die zur Gestaltung der Webseite nötig waren, werden nicht als Einzelleistung gelistet und werden nicht der Projektleistung angerechnet.

Der SHA1-Fingerabdruck des Zertifikates der Webseite lautet:

4B:BF:58:ED:DB:B8:7D:EB:DF:52:5D:9C:93:4D:D5:9A:F6:A3:05:EC