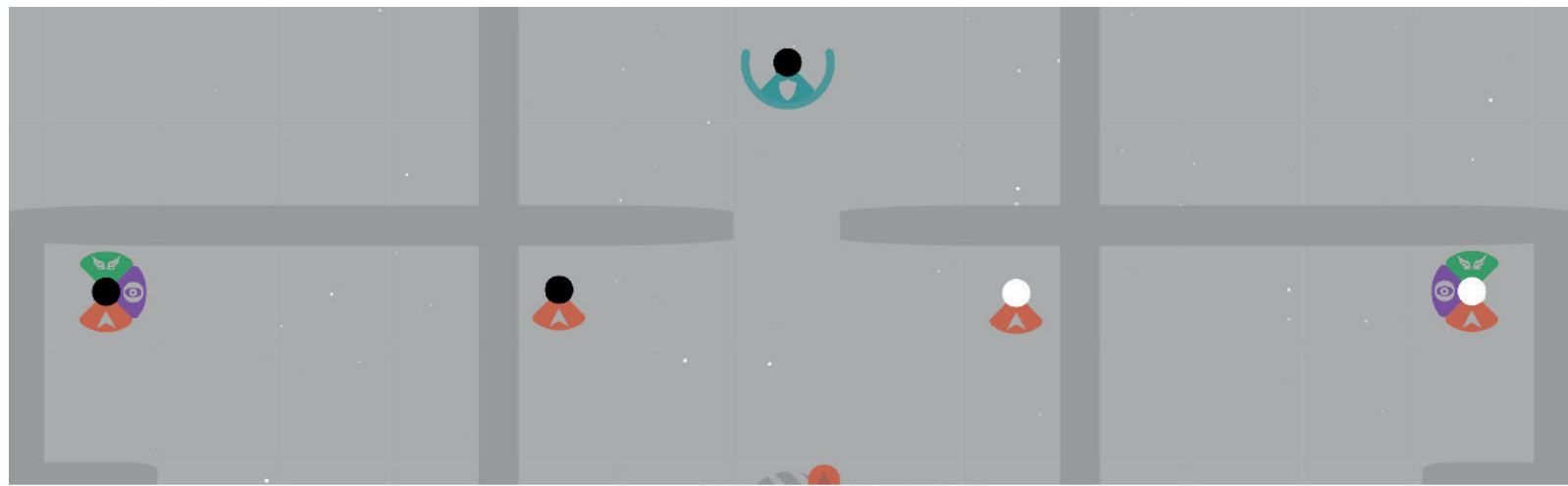
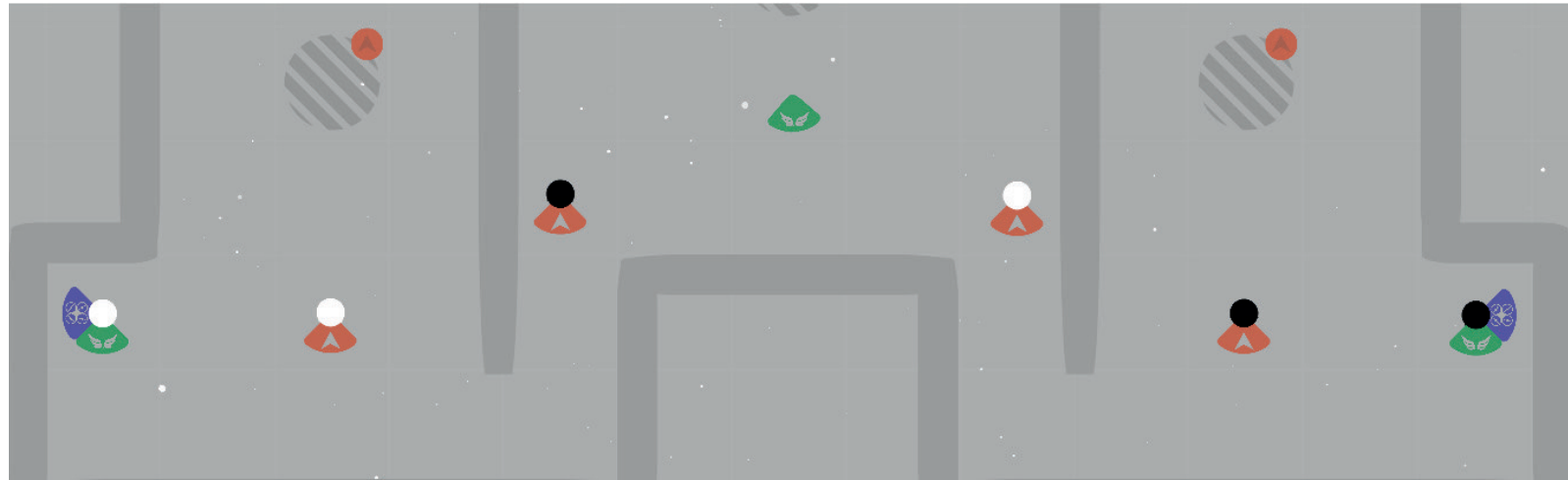




PLAYER I



Bachelorarbeit

Entwicklung eines digitalen Prototypen für ein rundenbasiertes Strategiespiel

Markus Sockel

Matrikelnummer: 530613

11.02.2014

HTW Berlin

Fachbereich 5

Game Design

Erstgutachter: Prof. Thomas Bremer

Zweitgutachter: Prof. Claus Wohlgemuth

Danksagung

Zunächst möchte ich mich an dieser Stelle bei all denjenigen bedanken, die mich während der Anfertigung dieser Bachelor-Arbeit unterstützt und motiviert haben.

Ganz besonders gilt dieser Dank Herrn Prof. Bremer, der meine Arbeit und somit auch mich betreut hat. Nicht nur, dass er viele wertvolle Hinweise gab, auch seine moralische Unterstützung und Motivation waren unschlagbar.

Daneben gilt mein Dank Max Heyder, Johannes Roth, Semjon Leinweber und meinem Bruder Matthias Sockel, die jeweils mehrmals ausführliches Feedback zu dem entstandenen Spiel gegeben haben. Sie wiesen auf Schwächen hin und konnten immer wieder zeigen, wo noch Schwierigkeiten bestanden.

Nicht zuletzt gebührt meinen Eltern Dank, die mich während des Studiums so sehr unterstützten.

Inhaltsverzeichnis

Danksagung	1
Abbildungsverzeichnis	5
1. Zielstellung	7
2. Ergebnis	11
3. Entwickeln der Spielregeln	15
3.1. Grundidee	15
3.2. Erster, analoger Prototyp	16
3.3. Verbessern der Spielübersichtlichkeit durch Regelveränderungen.....	18
3.4. Spawnen von Modulen	20
3.5. Module.....	22
4. Usability	25
4.1. UI.....	25
4.2. Kamera	26
4.3. Steuerung.....	27
5. Gestaltung	31
5.1. Schiffskern.....	31
5.2. Module	31
5.3. Outposts.....	32
5.4. Angriff	32
5.5. Hintergrund.....	33
5.6. Musik und Sounds	33
5.7. Wahl des Namens	33
6. Tutorial	35
7. Level Design	37
8. Verwendete Tools.....	41
Literaturverzeichnis	43
Eidesstaatliche Erklärung	45

Abbildungsverzeichnis

Alle hier aufgelisteten Bilder sowie das Titelbild dieser Bachelorarbeit wurden von mir angefertigt. Es sind Screenshots verschiedener Build-Versionen des Spiels. Alle Builds wurden mit der kostenlosen Version von Unity 4.3.3 erstellt.

Abb. 1	12
Abb. 2	12
Abb. 3	12
Abb. 4	12
Abb. 5	12
Abb. 6	13
Abb. 7	15
Abb. 8	15
Abb. 9	18
Abb. 11	19
Abb. 10	19
Abb. 12	20
Abb. 13	20
Abb. 14	21
Abb. 16	22
Abb. 15	22
Abb. 17	22
Abb. 18	23
Abb. 19	23
Abb. 20	25
Abb. 23	25
Abb. 21	25
Abb. 22	25
Abb. 25	26
Abb. 24	26
Abb. 26	27
Abb. 27	27
Abb. 28	27
Abb. 30	28
Abb. 29	28
Abb. 31	29
Abb. 33	31
Abb. 32	31
Abb. 34	32
Abb. 36	32
Abb. 35	32
Abb. 37	37

Abb. 38		38
Abb. 39		38
Abb. 40		39
Abb. 41		39

1. Zielstellung

Ziel der Bachelorarbeit ist es einen digitalen Prototyp eines rundenbasierten Strategiespiels zu erstellen.

Definition „Rundenbasiertes Strategiespiel“

Dafür soll zunächst geklärt werden was ein Strategiespiel ist, und was rundenbasiert bedeutet.

Eine gute Definition des Begriffs Strategiespiel findet sich auf MobyGames.com (Genre Definitions, k.A.):

„In its broad sense, the strategy genre encompasses games that emphasize problem-solving. Thinking and planning are necessary components of strategy; they can be used for such diverse purposes as preparing and positioning troops to gain advantage in a war, or figure out the principle of a puzzle. While adventure games may include puzzles, they focus on following a linear story while occasionally dealing with obstacles rather than being entirely dedicated to problem-solving. Puzzle, chess, and card games fall under the strategy category. [...] Strategy games can be either turn-based or real-time.“

Auf dieser Seite wird auch erklärt, was in Spielen unter dem Begriff „rundenbasiert“ (engl.: turn-based) zu verstehen ist:

„In turn-based games the flow of play is well-defined (entirely or partially) into „turns“. A player has time to decide what his/her next move will be. The computer AI opponent (or the next human player) then gets time to make their move(s).“ (ebd.)

Der am Ende entstehende Prototyp sollte also unter diese beiden Definitionen fallen.

Konkretisierung der Zielstellung

Da der Fokus meiner Arbeit im Bereich Game Design liegt, möchte ich nicht einfach eines der bekannten rundenbasierten Strategiespiele „kopieren“, sondern im Gegenteil ein Spiel mit möglichst innovativer Spielmechanik entwickeln.

Dafür habe ich im Vorfeld verschiedene rundenbasierte Spiele gespielt, um auf Schwächen aufmerksam zu werden, Raum für Verbesserungen zu entdecken und allgemein inspiriert zu werden. Dazu gehörten klassische rundenbasierte Strategiespiele wie *Civilization V* (Firaxis Games, 2010), *XCom: Enemy Unknown* (Firaxis Games, 2012) und *Battle World: Kronos* (KING Art Games, 2013), Sammelkartenspiele wie *Magic 2014 – Duels of the Planeswalkers* (Stainless Games, 2013) oder *Hearthstone: Heroes of Warcraft* (Blizzard Entertainment, 2013) und ungewöhnliche Ansätze oder Verschmelzungen der beiden Genres wie *Skulls of the Shogun* (17-Bit, 2013), *Scrolls* (Mojang, 2013), *Neuroshima Hex* (Big Daddy's Creations, 2013) und *FTL: Faster Than Light* (Subset Games, 2012).

Bei Spielen mit Sammelkartenspielelementen (Magic 2014, Hearthstone, Scrolls und Neuroshima Hex) haben mir vor allem die schnellen und häufigen Spielwendungen sowie die Unvorhersehbarkeit durch Zauber und Eigenschaftsveränderungen gefallen, die durch das Ziehen von Karten in jeder Runde zustande kamen. Dieses „Cards“-Pattern (Bjork & Holopainen, 2005, S. 93) mit der daraus entstehenden Spieldynamik sollte in meinem Prototyp ebenfalls vorhanden sein.

Die Probleme die mir aufgefallen sind, sind nicht wirklich neu: Strategiespiele wirken auf den ersten Blick aufgrund ihrer anfänglichen Undurchschaubarkeit alles andere als zugänglich. Daran sind z. B. oftmals vorhandene „Untersysteme“ (wie z.B. Diplomatie- und Religionssysteme in *Civilization V*, Basismanagement in *XCom: Enemy Unknown*) Schuld.

Noch bedeutender ist aber vermutlich die Unübersichtlichkeit der verschiedenen Kampf-Variablen (Angriffswert, Verteidigungswert, HP, Angriffsbonus gegenüber bestimmten Einheiten, Terrain-Bonus usw.). Info-Bildschirme mit Zahlenauflistungen sind eine Möglichkeit dem Spieler die Werte nachschauen zu lassen, stören den Spielfluss und -spaß aber enorm. „Outcome Indicators“ (Bjork & Holopainen, 2005, S. 142) wie Vorhersagen am Gesundheitsbalken sind eine weitere Möglichkeit, aber es benötigt stundenlanges Spielen bis man intuitiv weiß wie ein Kampf verlaufen würde.

Damit verbunden ist auch eine Unübersichtlichkeit durch das „Paper-Rock-Scissors“-Pattern (ebd., S. 398), wenn man sich merken muss welche Einheiten gegen welche Einheiten des Gegners wie effektiv ist.

Zu lange Spieldauer, lange Wartezeiten zwischen Zügen („Downtime“, ebd., S. 162) und zu wenig Überraschungen sind weitere Probleme, die rundenbasierte Strategiespiele zu einem Nischenprodukt machen. Und nicht zuletzt: wenn man mal einen Monat lang nicht mehr spielt, findet man oft nur schwer wieder in das Spiel hinein.

In meinem Prototyp sollen all diese negativen Faktoren möglichst vermieden werden.

Außerdem möchte ich, dass das Spiel für mehrere menschliche Spieler („Multiplayer Game“, ebd., S. 345-346) ausgelegt ist, da die Programmierung der KI zu aufwändig für den Rahmen dieser Arbeit wäre. Die Spieler sollten an einem Gerät direkt gegeneinander spielen können, damit das Spiel möglichst einfach getestet, entwickelt und anderen gezeigt werden kann. Dadurch fallen einige übliche Patterns rundenbasierter Strategiespiele wie „Fog of war“ (ebd., S. 133), „Exploration“ (ebd., S. 306) oder „Asymmetric Information“ (ebd., S. 135) weg. Denn diese funktionieren nur, wenn jeder Spieler nur einen Teil der Gesamtinformationen zur Verfügung hat. Das lässt sich mit nur einem Gerät aber nicht realisieren.

Somit ergibt sich für mich eine genauere Zielstellung:

Ich möchte ein rundenbasiertes Strategiespiel mit innovativen Gameplay-Elementen für 2 Spieler entwickeln, das nur auf einem Gerät spielbar ist, das schnell zu erlernen, übersichtlich und zugänglich ist, einen gewissen Arcade- und Sammelkartenspielfaktor hat (schnelle aber kurze und intensive Spielrunden mit sehr geringer Downtime). Strategie- und Taktikmöglichkeiten sollten darunter aber nicht leiden. Das Spiel soll in der mir zur Verfügung stehenden Zeit von mir allein umgesetzt werden können. Eine Übertragung auf ein mobiles Gerät sollte zudem im Hinterkopf behalten werden.

2. Ergebnis

Bevor ich zum Entstehungsprozess komme, möchte ich gleich die Regeln des finalen Spiels vorstellen, damit man bei den Regeländerungen während des Entstehungsprozesses besser die Übersicht behält.

Überblick:

- Das Spiel ist ein rundenbasiertes Strategiespiel in abstraktem Weltraum-Setting für 2 Spieler. Diese spielen gegeneinander.
- Beide Spieler kontrollieren dabei eine bestimmte Anzahl an Raumschiffen.

Spielfeld:

- Das Spielfeld ist ein quadratisches Raster mit Wänden, durch die sich Raumschiffe nicht hindurch bewegen können.

Raumschiffe:

- Raumschiffe bestehen aus dem Schiffskern und bis zu 4 Modulen, welche die Fähigkeiten des Schiffskerns erweitern.
- Der Schiffskern an sich kann sich nur innerhalb des festgelegten Bewegungsradius bewegen, und ist nach einem Angriff zerstört. Er selbst kann NICHT angreifen! Dafür ist ein Weapon-Modul notwendig.

Angriff:

- Hat ein Raumschiff ein oder mehrere Weapon-Module, so kann es gegnerische Raumschiffe, die sich auf Feldern innerhalb des Angriffsradius befinden, angreifen.
- Der Angriffsradius ist dabei standardmäßig so groß, dass der Spieler die 4 direkt angrenzenden Felder angreifen kann (also nicht diagonal).
- Ein Schiff kann nur ein Mal pro Runde angreifen.
- Nach einem Angriff steht dem Schiff nur noch der verbleibende Bewegungsradius zur Verfügung.

Schusslinie:

- Für den Angriff ist die Schusslinie entscheidend! Die Schusslinie beginnt beim Schiffskern des angreifenden Raumschiff und geht zum gegnerischen Schiffskern.
- Befindet sich ein Modul des gegnerischen Raumschiffs in der Schusslinie, so wird dieses getroffen.
- Wenn nicht, wird der gegnerische Schiffskern getroffen. Das gegnerische Schiff verliert dadurch die angebrachten Module, so dass andere Schiffe diese aufnehmen können.

Anbringen von Modulen:

- Befindet sich der Spieler nahe genug an einem Modul, so kann er dieses an einem freien Slot anbringen. Sind alle 4 Slots bereits belegt, kann er es nicht anbringen.

Rotieren der angebrachten Module:

- Der Spieler kann den „Ring“ mit angebrachten Modulen um den Schiffskern herum rotieren.
- So kann er den Schiffskern schützen, in dem er bei Vermutung eines gegnerischen Angriffs in der nächsten Runde ein Modul zum Abfangen des Schusses in die gegnerische Schusslinie platziert.
- Außerdem kann er so auch ein Modul schützen, in dem er bei Vermutung eines gegnerischen Angriffs auf das Modul in der nächsten Runde ein anderes Modul in die Schusslinie des Gegners platziert.

Outposts:

- Outposts sind Spielfelder, die der Spieler einnehmen muss, um zu Beginn seiner Runde eine Outpost-Aktion durchführen zu können.
- Outposts können neue Module erscheinen lassen oder ein zufälliges Teil (Schiffskern oder Modul) eines zufälligen gegnerischen Schiffs auf den 8 angrenzenden Feldern des Outposts angreifen.
- Um einen Outpost einzunehmen, platziert ein Spieler ein Raumschiff bis zum Ende seiner Runde auf ein Outpost-Feld.
- Outposts benötigen für das Durchführen einer Aktion Energie.
- Wenn der Spieler einen Outpost einnimmt, wird die Energie dieses und aller anderen Outposts des Spielers aufgeladen.
- Führt der Spieler eine Outpost-Aktion zu Beginn seiner Runde durch, wird die Energie dieses und aller anderen Outposts des Spielers aufgebraucht.
- Der Spieler kann nur einen Outpost pro Runde eine Aktion durchführen lassen, und muss diesen auswählen.
- Mit jeder Runde regeneriert sich die Energie jedes Outposts des Spielers zur Hälfte wieder. Nach 2 Runden kann der Spieler also wieder eine Outpost-Aktion durchführen.
- Wenn der Spieler zu Beginn seiner Runde keine Outpost-Aktion durchführen möchte, kann er diese auch auf nächste Runde verschieben. Die Energie der Outposts wird dabei nicht aufgebraucht.

Spielende:

- Wenn einer der beiden Spieler alle Raumschiffe des gegnerischen Spielers zerstört hat oder ein Spieler alle Outposts im Spiel eingenommen hat, so hat dieser gewonnen.

Module:



Abb. 1 Weapon: ermöglicht Angriff.



Abb. 2 Eagle Eye: erhöht den Angriffsradius.



Abb. 3 Wings: erhöht den Bewegungsradius.

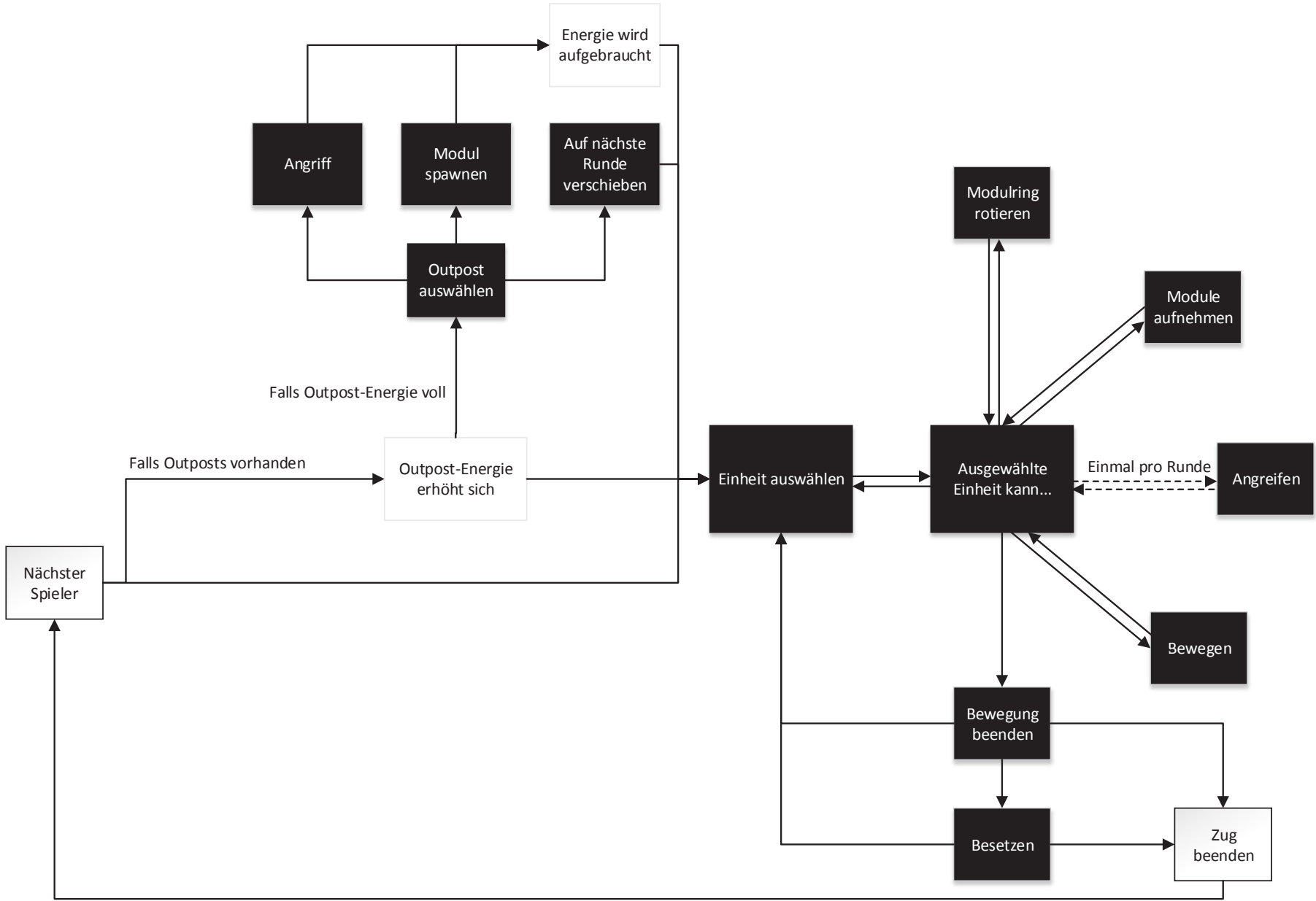


Abb. 4 Shield: bildet einen großen Schild, der einen Schuss abfängt.



Abb. 5 Decoy Drone: Fängt einen Schuss auf den Schiffskern ab.

Abb. 6 Ablauf des Spiels



3. Entwickeln der Spielregeln

Im folgenden möchte ich aufzeigen, wie ich zu diesen Spielregeln gekommen bin:

3.1. Grundidee

Aus der anfänglichen Analyse bestehender rundenbasierter Strategiespiele ergab sich folgende Grundidee:

Das Spiel soll genretypisch Kampf als Schwerpunkt haben („Combat“-Pattern: Bjork & Holopainen, 2005, S. 145). Es geht also darum die „Units“ (ebd., S. 79) des Gegners zu zerstören („Overcome“-Pattern: ebd., S. 280-281) und um das zu erreichen möglichst viele der eigenen Units zu erhalten („Survive“- ebd., S. 298). „Conflict“ (ebd., S. 239) und „Competition“ (ebd., S. 238) motivieren also beide Spieler.

Alle Spieler haben jeweils dieselben Schiffe mit denselben Eigenschaften, können diese durch angebrachte Module („Pick-Ups“- ebd., S. 87) aber erweitern („Gain Competence“-Pattern: ebd., S.304). Diese Module können in die Schusslinie zu einem gegnerischen Schiff positioniert werden, um Schüsse des Gegners abzublocken und das eigene Schiff zu schützen. Zu diesem Zweck können die Module um das Schiff herum gedreht werden („Alignment“-Pattern: ebd., S. 172).



Abb. 7 Legosteine und Spielkarten helfen die Spielidee analog zu testen.



Abb. 8 Bücher bilden Wände. Mit einer Schnur kann der Angriffs- und Bewegungsradius eingehalten werden. Das Spielfeld wurde in ein 6x6 Raster aufgeteilt, so dass durch zweimaliges Würfeln die zufällige Spawn-Position eines Moduls ermittelt werden kann.

Dadurch müssten Spieler nicht mehrere Schiffstypen und deren Werte erlernen, was die Erlernbarkeit des Spiels vereinfacht und das Risiko senkt, dass man ein Spiel nach einem Monat nicht mehr spielen kann, weil man Werte und Eigenschaften der einzelnen Einheiten wieder vergessen hat.

Stattdessen können Eigenschaften der Schiffe durch Erweiterung mit Modulen nachvollziehbar und jederzeit ohne Mühe einsehbar verändert werden („Direct Information“-Pattern: ebd., S. 133) und sie dadurch in die gewünschte Richtung „spezialisiert“ werden.

Außerdem können Spieler ihre Schiffe beschützen, sie müssen es nur mit genügend Modulen ausgerüstet haben und entscheiden, auf welches Modul und dessen erweiternde Eigenschaft sie verzichten wollen („Ability Losses“, ebd. S. 175-176).

Diese Grundidee hat sich während des Entwicklungsprozesses nicht verändert.

3.2. Erster, analoger Prototyp

Die Spielidee wurde anfangs in einem analogen Prototyp getestet, um schnell grundlegende Regeln testen und verändern zu können, und so eine Basis für die Entwicklung des digitalen Prototyps zu formen.

Die Regeln der ersten Version (zusätzlich zur oben beschriebenen Grundidee):

- Der Spieler kann sich frei (ungebunden an ein Raster) mit dem Schiff innerhalb eines Bewegungsradius bewegen („Movement“-Pattern: Bjork & Holopainen, 2005, S. 147 sowie „Movement Limitations“-Pattern: ebd., S. 154). Denn die Raumschiffe sind die „Focus Loci“ (ebd., S. 169), durch sie führt der Spieler Aktionen durch. Ohne Limitierung der Bewegung würde dann einfach nur der Spieler mit den meisten Schiffen (der also zuerst alle Schiffe des Gegners abschießen kann) gewinnen.
- Die Ungebundtheit an ein Raster hatte ich so entschieden, weil Raster als „langweilig“ und einengend empfunden werden, und der Verzicht auf ein Raster bei *Skulls of the Shogun* (17-Bit, 2013) sehr gut funktioniert hat.
- Der Spieler kann die Module als Ring um sich in jedem beliebigen Winkel ausrichten.
- Der Angriffsradius ist doppelt so groß wie der Bewegungsradius (denn Raumschiffe können schießen, man erwartet also einen hohen Angriffsradius).
- Ein Schuss verursacht 1 Schaden („Damage“-Pattern: Bjork & Holopainen, 2005, S. 184), um die Werte möglichst einfach und nachvollziehbar zu halten.
- Der Schiffskern hat 2 HP, so dass man sich zumindest 1 mal pro Schiff gegen das Opfern eines Moduls entscheiden kann, ohne dass der Schiffskern gleich zerstört wird.
- Module haben 1 HP, so dass der Spieler immer wieder die „einfache“ Entscheidung treffen muss ob er eine Modul opfern will oder nicht. Es gibt kein „halbes Opfern“.
- Der Schiffskern hat eine Waffe, so dass das Schiff immer schießen kann – später hat sich das geändert.

Aktionen:

- Jedes Schiff darf nur eine Aktion pro Runde durchführen („Budgeted Action Points“-Pattern: ebd., S. 184).

- Als Aktion zählt Angreifen (Modul oder Schiffskern) oder das Ausspielen einer Aktionskarte.
- Jedes Schiff darf aber vor und/oder nach einer Aktion beliebig viele Module aufnehmen.
- Nach einer Aktion kann sich das Schiff nur noch innerhalb des verbleibenden Bewegungsradius (dem noch nicht vor der Aktion verbrauchten Bewegungsradius) bewegen.

Module:

- Der Spieler weiß erst was ein aufgenommenes Modul bewirkt, wenn er es aufgenommen hat („Randomness“-Pattern: ebd., S. 373 sowie „Irreversible Actions“-Pattern: ebd., S. 180).
- Zu Beginn des Spiels liegen 4 Module an zufälligen Positionen im Spielfeld.
- Wird ein Modul aufgenommen, erscheint ein neues, zufälliges Modul an einer zufälligen Position im Spielfeld. Dadurch ergeben sich „Asymmetric Abilities“ (ebd., S. 158-159) der einzelnen Raumschiffe.
- Der Spieler kann maximal 4 Module anbringen. Dadurch soll zu vielen Eigenschafts-Veränderungen und der damit verbundenen Unausgewogenheit und Unübersichtlichkeit vorgebeugt werden.

Aktionskarten:

- In dieser ersten Version waren noch Aktionskarten („Cards“-Pattern: ebd., S. 93) vorgesehen, um die besondere Spieldynamik von Kartenspielen (hohes Maß an Überraschungen und Wendungen) zu gewähren.
- Sie haben meist Eigenschaftsveränderungen zu Folge, die nicht an Module gebunden sind, oder stattdessen ein bestimmtes Schiff mit einem Modul aus.
- Aktionskarten werden zu Beginn jeder Runde gezogen
- Aktionskarten können verschiedene Aktionen durchführen, brauchen aber eine bestimmte Anzahl an gesammelten Energie-Punkten („Resources“-Pattern, ebd., S. 107-109).
- Energie-Punkte erhält man durch das Ausspielen von Energie-Aktionskarten

Module

- Schutzmodule (machen nichts außer Schüsse abzufangen):
 - mit 1 HP
 - mit 2 HP
 - mit 3 HP
- Schiff-HP-Erhöher:
 - Schiff HP +1
 - Schiff HP +2
- Einmalig: Rakete (zerstört ganzes Schiff inkl. Module)
- Einmalig: Teleporter (Teleport an beliebigen Ort)
- Doppelschuss (2 Angriffe in einer Runde)

Aktionskarten

- Energie
- Schiff-HP-Erhöher +1
- Modul-HP-Erhöher +1
- Doppelschuss (2 Angriffe in einer Runde)
- Neues Schiff
- Schutzmodule:
 - mit 1 HP
 - mit 2 HP
- Beliebiges Gegnermodul zerstören
- Teleporter (Teleport an beliebigen Ort)
- Beliebiges Gegnermodul stehlen und an eigenes Schiff anbringen
- Laser (Ziel verliert 1 HP)

Einmalig = Modul verschwindet nachdem es verwendet wurde.

Nach dem ersten Testspielen stellte sich das Spielprinzip als sehr interessant heraus, hatte etwas Einzigartiges und Potenzial für viele taktische Möglichkeiten, so dass ich es weiterverfolgte. Allerdings ließ die Spielübersichtlichkeit noch zu wünschen übrig.

3.3. Verbessern der Spielübersichtlichkeit durch Regelveränderungen

Entfernen der Aktionskarten

Um die Spielübersichtlichkeit (Übersichtlichkeit der Regeln und des Spielgeschehens) zu verbessern habe ich zuerst die Aktionskarten aus dem Spiel entfernt, denn Module sollten deren Funktion übernehmen. Außerdem trugen Module bereits zu Genüge zu der Sammelkartenspiel-typischen Spieldynamik bei, das Spiel war mit all den Zufallselementen eher zu unberechenbar.

Vereinfachen des Spielablaufs

Außerdem sollte das Aufnehmen von Modulen auch als Aktion zählen, so dass man sich entscheiden musste, ob man wirklich ein Modul aufnehmen oder einen Angriff durchführen will („Budgeted Action Points“-Pattern, Bjork & Holopainen, 2005, S. 184). Das sollte „Stimulated Planning“ (ebd., S. 384) fördern.

Allerdings hat es sich später als den Spielfluss zu lähmend herausgestellt. Deshalb habe ich diese Änderung in einer späteren Version wieder rückgängig gemacht.

Erster, digitaler Prototyp

Ich begann außerdem mit der Arbeit am digitalen Prototyp, um die bisherigen Regeln des analogen Prototyps umzusetzen, vor allem auch um zu sehen, wie sich das Spiel in den angepeilten Größenverhältnissen und mit UI spielen würde. Im digitalen Prototyp spielte sich das Spiel weniger übersichtlich, vermutlich eben wegen den Größenverhältnissen (die Legoraumschiffe waren deutlich größer im Vergleich zu ihrem Angriffs- und Bewegungsradius), weil das sporadische Text-UI zu viel Informationen gleichzeitig anzeigte und weil ich beim analogen Prototyp unbewusst mehr zurechtgeschoben habe, als man es beim digitalen Prototyp konnte.

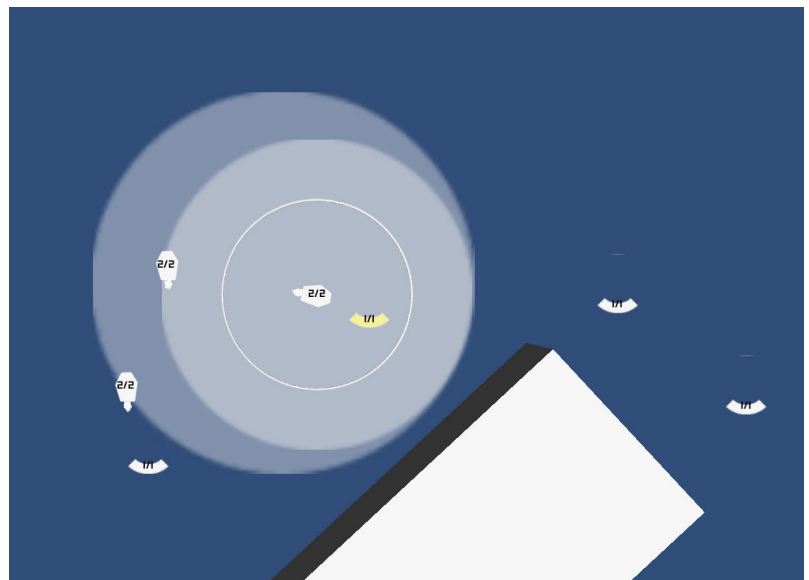


Abb. 9 Screenshot des ersten, digitalen Prototyps. Der Angriffsradius (weißer Ring) ist nur noch halb so groß wie der Bewegungsradius (transparenter Kreis).

Reduktion des Angriffsradius

Zuerst änderte ich deshalb den Angriffsradius (auf Abb. 9 und 10 angezeigt als weißer Ring um dem ausgewählten Raumschiff) und reduzierte ihn auf die Hälfte des Bewegungsradius. Dadurch wurde das Spiel übersichtlicher, da man auch nur noch einen kleineren Bereich um seine eigenen Raumschiffe beobachten musste, um sicherzugehen, dass sie in der nächsten Runde nicht angegriffen werden würden.

Umstieg auf quadratisches Raster

Außerdem hat sich als Problem bemerkbar gemacht, dass das Rotieren der Module in Schusslinie kaum funktioniert hat, da man einfach von hinten an das gegnerische Schiff heran fliegen und auf den Schiffskern schießen konnte. Deshalb entschloss ich mich doch für die Bewegung auf einem Raster.

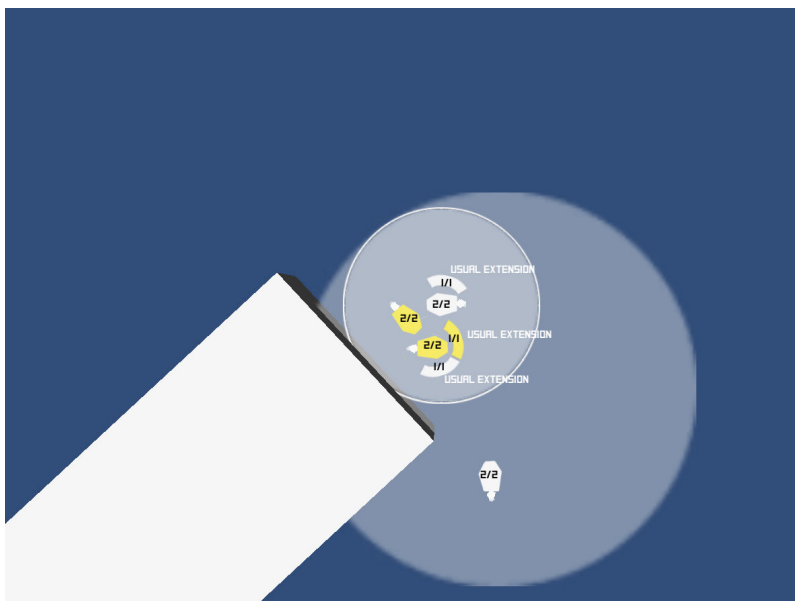


Abb. 10 Ohne Raster funktioniert das Blocken durch Module im digitalen Prototyp leider so gut wie gar nicht.

Da man sich bei einem hexagonalen Raster wieder einfach um das blockierende Modul herum bewegen könnte, entschied ich mich für ein quadratisches Raster.

Dabei dürfen andere Schiffe nicht auf ein bereits belegtes Feld ziehen, was taktisch verwendet werden kann um z.B. Wege des Gegners durch eigene Schiffe zu blockieren („Obstacles“-Pattern: Bjork & Holopainen, 2005, S. 75).

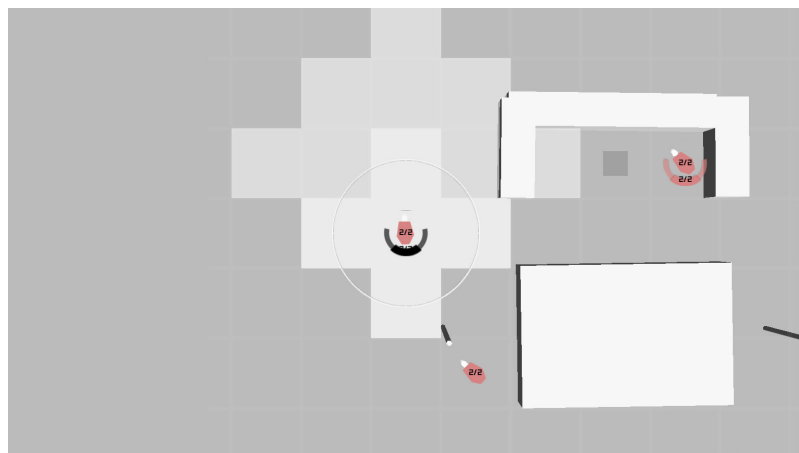


Abb. 11 Die transparenten weiß-grauen Felder zeigen den zur Verfügung stehenden Bewegungsradius, die weniger transparenten weiß-grauen Felder den nach einem Angriff verbleibenden Bewegungsradius.

Dadurch muss auch der Angriffsradius rasterartig angezeigt werden. Da man den Angriffsradius dank des Rasters schnell gelernt hat, muss dieser nicht ständig angezeigt werden. Der Angriffsradius beschränkt sich standardmäßig auf die 4 direkt anliegenden Felder. Das macht es sehr einfach vorherzusehen, wo eine Angriffssituation in der nächsten Runde möglich ist und wo nicht.



Abb. 12 Rasterartiger Angriffsradius

Weitere Vereinfachung der Kampfwerte

Außerdem habe ich später die Schiff-HP auf 1 reduziert, wodurch die Kämpfe intensiver wurden, die Bedeutung der Module weiter stieg, und eine HP-Anzeige überflüssig wurde.

3.4. Spawnen von Modulen

Wie sollten neue Module im Spiel erscheinen?

Das anfangs zufällige Spawnen von Modulen auf dem Spielfeld war zu unvorhersehbar und für den Spielausgang zu entscheidend, als dass es zu einem Strategiespiel passte.

Energiesäulen

Damit vor einem Spieler nicht zufälligerweise ein mächtiges Modul erscheint, das er in der selben Runde noch verwenden kann, habe ich zunächst „Energiesäulen“ eingebaut. Der Spieler muss sich erst mit neu aufgenommenen Modulen zu diesen Energiesäulen bewegen, damit die Module aktiviert werden - der Erweiterungseffekt also in Kraft tritt („Chargers“-Pattern: Bjork & Holopainen, 2005, S. 104). Ohne Aktivierung kann das Modul aber trotzdem Angriffe abwehren.

Die Energiesäulen hatten den schönen Nebeneffekt, dass sich die Raumschiffe mehr auf dem Spielfeld verteilt haben, weshalb ich sie noch lange im Spiel behielt, schließlich aber doch entfernte.



Abb. 13 Fliegt der Spieler mit einem noch nicht aktivierten Modul an eine Energiesäule, wird diese durch einen „Energiestrahl“ aktiviert, und die Erweiterungseigenschaft des Moduls hat Effekt.

Rotierendes Spawnen

Trotzdem sollten die Module an vorhersehbaren Stellen im Spiel erscheinen („Spawn Points“-Pattern, ebd., S. 105).

Deshalb probierte ich zunächst ein „rotierendes“ Spawnen, wo also Spawnpunkte festgelegt sind, von denen sich der aktive Spawnpunkt jede Runde in einem sich wiederholenden Muster ändert und ein neues Modul hervorbringt.

Das reduzierte den Zufalls-Faktor zwar enorm, teilweise musste man aber 4 Runden warten bis ein Modul erschien, während da wo keine Raumschiffe in der Nähe waren sich die Module häuften. Da man aber mit dem Kampf in direkter Nähe beschäftigt war und sich nicht 2-4 Runden wegbewegen konnte um ein neues Modul zu holen, hat man diese einfach nicht verwendet und das Spiel war eher langweilig.

Outposts

Deshalb habe ich mich schließlich für „Outposts“ (Stützpunkte) entschieden, die der Spieler einnehmen muss („Gain Ownership“-Pattern: ebd., S. 112 sowie „Strategic Locations“-Pattern, ebd., S. 100-101). Zu Beginn einer Runde kann der Spieler aus den zur Auswahl stehenden Outposts einen auswählen, an dessen Position ein neues Modul gespawnt werden soll. Dabei wird während der Auswahl angezeigt welches Modul gespawnt werden würde.

Um die Outposts noch nützlicher und relevanter zu machen, haben sie die Fähigkeit bekommen anstatt nur Module spawnen zu können auch feindliche Raumschiffe in den direkt angrenzenden 8 Feldern angreifen zu können („Deadly Traps“-Pattern, ebd., S. 74). Dabei muss der Spieler sich zu Beginn der Runde entscheiden ob er ein Modul aufnehmen oder den Gegner durch den Outpost angreifen will („Budgeted Action Points“: ebd., S. 184).



Abb. 14 Outposts können angreifen.

Um „Camping“ (ebd, S. 154) von Raumschiffen auf Outposts - bis die Raumschiffe möglichst viele Module aufgenommen haben - zu verhindern, habe ich festgelegt, dass Outposts Energie benötigen um eine Aktion durchführen zu können („Resources“-Pattern, ebd., S. 107-109). Diese wird pro Runde zur Hälfte regeneriert, so dass der Spieler aller 2 Runden eine Aktion durchführen kann.

Damit der Spieler so bald wie möglich von dem Outpost profitiert, wird die Energie aller Outposts des Spielers beim Einnehmen eines neuen Outposts vollständig regeneriert. So kann er gleich zu Beginn der nächsten Runde eine Aktion an diesem oder anderen Outposts durchführen.

Außerdem habe ich eingebaut, dass man das Spiel auch gewinnen kann, in dem man alle Outposts des Levels einnimmt. Dadurch können Spieler 2 grundverschiedene Ziele verfolgen (Gegner eliminieren oder alle Outposts besetzen), was das Spiel weniger vorhersehbar macht („Selectable Sets of Goals“-Pattern: ebd., S. 328 aber trotzdem „Symmetric Goals“: ebd., S. 333).

Da die Outposts bereits für genügend Verteilung der Raumschiffe auf dem Spielfeld sorgten, habe ich zuletzt die Energiesäulen entfernt, die durch das vorhersehbare Spawnen von Modulen an festgelegten Orten durch Outposts auch völlig überflüssig geworden sind.

3.5. Module

Bei den Modulen war es mir wichtig, dass jedes Einzelne den Ausgang eines Kampfes beeinflussen konnten. Außerdem sollten sich Module untereinander nicht ähneln („Orthogonal Unit Differentiation“: Bjork & Holopainen, 2005, S. 79-81) und ihr Effekt sich sinnvoll addieren, wenn man mehrere von ihnen aufnimmt.

Weapon

Dieses Modul ist sehr wichtig. Ohne dem Weapon-Modul kann kein Schiff angreifen. Anfangs bedeutete der Besitz mehrerer Weapon-Module, dass ein Schiff öfter angreifen kann. Durch die Vereinfachung der HP-Werte (alles standardmäßig 1 HP) ist das aber so gut wie überflüssig, so dass der „Vorteil“ mehrerer Angriffs-Module jetzt einfach nur noch darin besteht, dass der Spieler ein anderes Weapon-Modul zur Not opfern kann.



Abb. 15

Eagle Eye



Abb. 16

Das Modul „Eagle Eye“ vergrößert sofort den Angriffsradius. Bei mehreren „Eagle Eye“-Modulen addiert sich der Effekt. Das Modul hat dabei die Farbe des Angriffsradius.

Wings

Das Modul „Wings“ vergrößert sofort den Bewegungsradius. Bei mehreren „Wings“-Modulen addiert sich der Effekt. Das Modul hat dabei die Farbe des Bewegungsradius.



Abb. 17

Shield



Abb. 18

Das Schild blockiert einen Schuss aus 3 möglichen Richtungen. Bei einem Treffer wird es zerstört.

Während der Entwicklung hat das Schild sich stark verändert. Zu Beginn führen 2 Schilder links und rechts aus dem Modul heraus, die jeweils separat 1 HP hatten. Um zu gewährleisten, dass man nicht einfach auf das Schildmodul schießen kann um das Modul und die Schilder mit einem Mal zu zerstören, erhielt dieses Modul 2 HP. Außerdem sollte sich ein getroffenes Schild zu Beginn einer neuen Runde wiederherstellen. Das war noch in der ersten Version, als sich die Schiffe noch nicht in Rastern bewegt haben. Als das Raster eingebaut wurde, habe ich das Regenerieren wieder herausgenommen.

Um das Modul deutlich zu vereinfachen und die 2-HP-Modul-Ausnahme zu vermeiden, erscheint in der finalen Version einfach ein einziges, großes Schild vor dem Modul. Dieses fängt einen Angriff auf das Shield-Modul oder eine der beiden benachbarten Seiten ab.

Decoy Drone

Dieses Modul fängt einmalig einen gegnerischen Angriff auf den Schiffskern ab, in dem sich das Modul-Symbol in Form einer Drohne opfert.

Im Grunde erhöht es also die HP des Schiffskerns ohne dass eine extra HP-Anzeige notwendig ist.



Abb. 19

4. Usability

Nach dem ich die Entstehung der Spielregeln beschrieben habe, sollen an dieser Stelle Maßnahmen zur Verbesserung der Usability aufgelistet werden:

4.1. UI

Aktionsmarkierungen

Um dem Spieler Feedback zu geben, bei welchen Objekten in seiner Nähe er welche Aktionen durchführen kann, werden Markierungen um Objekte angezeigt,



Abb. 20

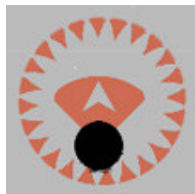


Abb. 21

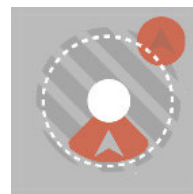


Abb. 22

die aufgenommen („Greifarm“),

angegriffen (ein Ring aus Angriffssymbolen)

und besetzt werden können (ein Ring aus Strichlinien).

Angriffs- und Bewegungsradius

Um die Informationsflut auf den Spieler möglichst gering zu halten, habe ich den anfangs permanent sichtbaren Angriffsradius geändert, so dass er nur noch angezeigt wird, wenn über dem aktuell ausgewählten Raumschiff die RT-Taste gedrückt gehalten wird, oder wenn der Spieler einen Gegner angreifen will.

Außerdem wird der Bewegungsradius jedes Raumschiffs automatisch angezeigt, wenn sich der Auswährling über einem Raumschiff befindet.



Abb. 23 Der Spieler kann auch den Bewegungsradius eines gegnerischen Schiffs sehen, in dem er den Auswährling über dessen Schiff positioniert.

Auswahl/Auswahlring

Um mit dem Auswahlring von Einheit zu Einheit zu wechseln oder bei einem Angriff oder Aufnehmen von Modulen von Ziel zu Ziel zu wechseln, bewegt der Spieler einfach den linken Controlstick in die Richtung des gewünschten Ziels.

Die Farbe des Auswahlrings zeigt an, welcher Spieler gerade dran ist („Handles“-Pattern: Bjork & Holopainen, 2005, S. 93).

Modul-Info-UI

Wenn der Spieler den Auswahlring über ein Schiff bewegt, werden die Namen der angebrachten Module angezeigt. Außerdem wird der Name eines Moduls angezeigt, nach dem es aufgenommen wurde.

Um jederzeit den Namen sowie die Beschreibung eines angebrachten Moduls ansehen zu können, kann der Spieler - wenn sich der Auswahlring über dem Schiff mit dem angebrachten Modul befindet - das digitale Steuerkreuz in Richtung des Moduls (ausgehend vom Schiffskern) gedrückt halten.

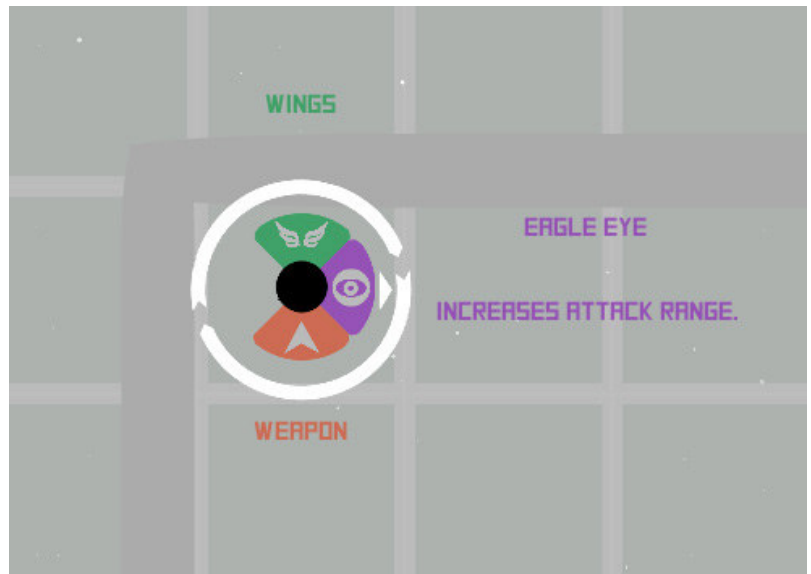


Abb. 24 Per Druck des digitalen Steuerkreuzes in Richtung des Moduls wird dessen Beschreibung angezeigt.

4.2. Kamera

Um dem Spieler den Spielfeldüberblick zu geben den er braucht oder gerade wichtige Spielgeschehen hervorzuheben gibt es 3 verschiedene Kamera-Zoom-Einstellungen:

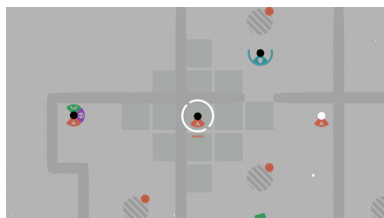


Abb. 25 Standard-Zoom

Meist spielt sich das Spiel in der Standard-Zoom-Stufe ab. In dieser plant der Spieler seine Züge und führt sie durch.

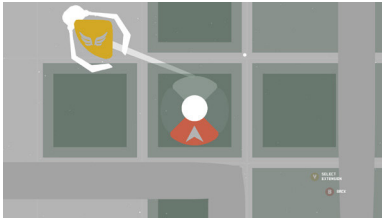


Abb. 26 Herangezoomt

Führt er eine Aktion durch (Angreifen oder Modul aufnehmen) zoomt die Kamera nah heran, so dass der Spieler genauer die um ihn liegenden Ziele sehen kann und gut sichtbar zwischen den zur Auswahl stehenden Zielen wechseln kann. Außerdem bringt der Zoom auch eine optische Veränderung zum Standard-Modus, so dass der Spieler intuitiv weiß, dass er jetzt mit dem Controller andere Aktionen bewirkt.



Abb. 27 Weggezoomt

Während der Runden- und Spielerankündigung sowie der Auswahl der Outpost-Aktion zoomt die Kamera heraus, so dass der Spieler das gesamte Spielfeld überblicken kann. Die markante optische Veränderung zum Standard-Zoom-Modus signalisiert dem Spieler, dass der nächste Spieler dran ist bzw. dass er eine Outpost-Aktion auswählen soll.

Um dem Spieler darüber hinaus mehr Übersicht über das Spielfeld zu geben, kann er die Kamera jederzeit mit Hilfe des rechten Controlsticks bewegen.

Zughilfe

Der Spieler kann seine Bewegung jederzeit wieder korrigieren. Hat er allerdings bereits ein Schiff angegriffen, kann er seine Bewegung nur innerhalb des verbleibenden Bewegungsradius korrigieren, denn ein Angriff ist eine „Irreversible Action“ (ebd., Bjork & Holopainen, 2005, S. 180).

Hat der Spieler alle seine Raumschiffe bewegt, beginnt automatisch die nächste Runde. Ich hatte an dieser Stelle ein PopUp-Fenster eingebaut, das nachfragt, ob man die Runde wirklich beenden will.

Da man aber nur in äußerst seltenen Fällen die Runde noch nicht beenden will, und das Pop Up den Spielfluss etwas unterbricht, habe ich mich entschieden es wieder zu entfernen.

4.3. Steuerung

Aktionsauswahl

Anfangs hatte ich auf dem Bildschirm unten rechts nur stehen was die Buttons A und B machen. Hatte man ein Schiff ausgewählt, so erschien ein Aktionsmenü, aus dem man mit dem linken Control Stick die zur Verfügung stehenden Aktionen auswählen konnte.

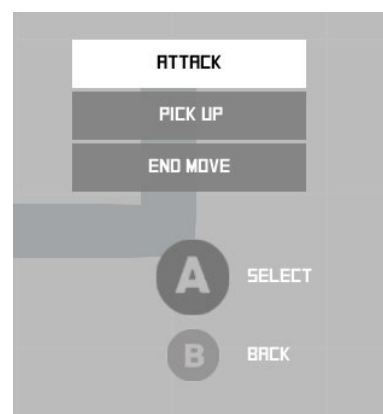


Abb. 28 Alte Aktionsauswahl

Um einen schnelleren Spielfluss zu erreichen, habe ich die Aktionsauswahl so geändert, dass man alle Aktionen mit den Tasten A, B, X, Y durchführen kann:

- A = Schiffbewegung bestätigen
- B = Position zurücksetzen / Abbrechen
- X = Angriff
- Y = Modul aufnehmen bzw. Modul spawnen



Abb. 29 Neue Aktionsauswahl

Outposts kann man einfach besetzen, in dem man das Schiff auf das Outpost-Feld bewegt.

Dadurch ist außerdem immer sichtbar welche Aktionen dem Spieler in der aktuellen Situation zur Verfügung stehen.

Rotierender Modus vs. Hot-Seat-Modus

Um das Spielen komfortabler zu machen habe ich einen rotierenden Modus (jeder Spieler hat sein eigenes Gamepad, das er nutzen kann wenn er dran ist) und einen Hot-Seat-Modus (die Spieler wechseln sich immer an einem Gamepad ab) eingebaut.

Auswahl der Steuerung

Wenn das erste Spiel gestartet wird erscheint der Steuerungs-Auswahl-Bildschirm. In diesem legen die Spieler in einer „Tabelle“ fest, mit welchem Eingabegerät sie steuern, und ob sie Spieler 1 oder 2 sein wollen. Außerdem kann man sich hier stattdessen für den Hot-Seat-Modus entscheiden.

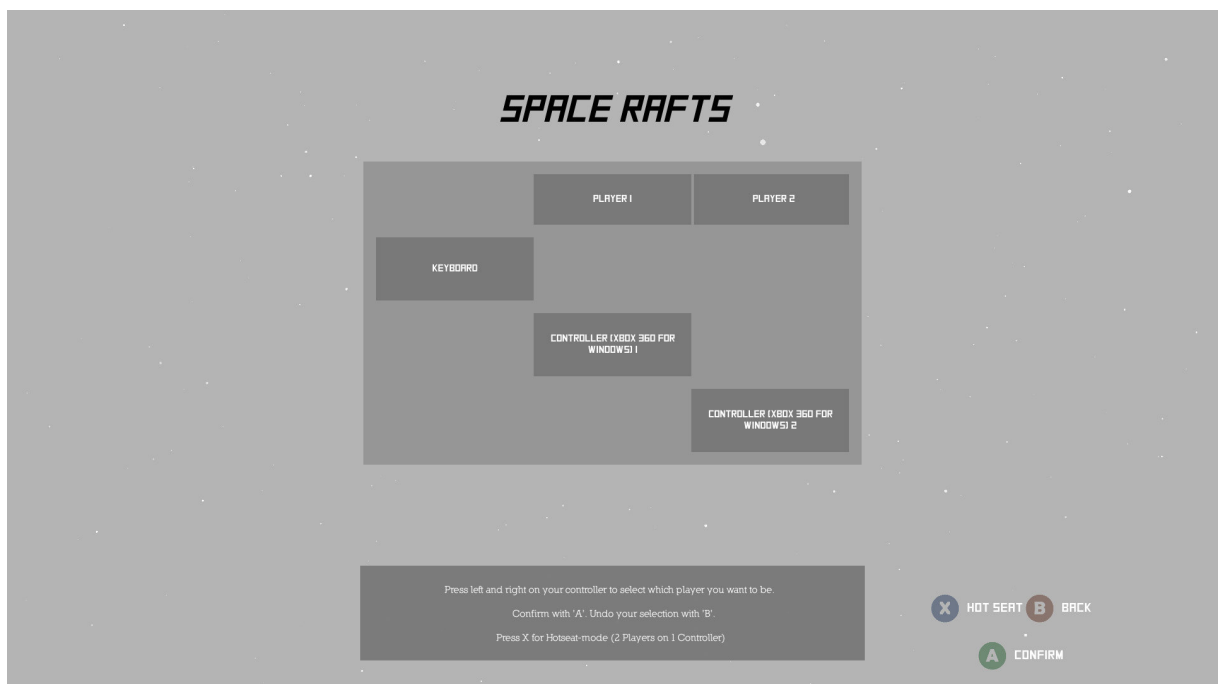


Abb. 30 Controller-Auswahl-Bildschirm

Der Steuerungs-Auswahl-Bildschirm ist auch im Hauptmenü unter Optionen verfügbar.

Modul aufnehmen

Um Module möglichst unkompliziert aufzunehmen, werden sie an den nächsten freiliegenden Slot angebracht, wenn der Spieler mit „Y“ bestätigt. Dabei kann er den Ring mit Modulen drehen, so dass er den nächsten freiliegenden Slot ändern kann.

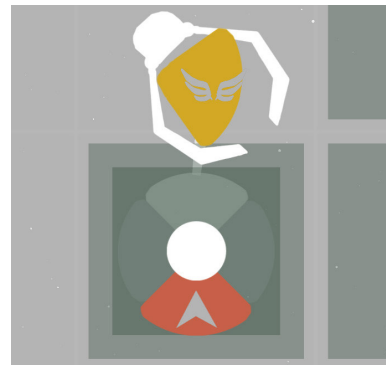


Abb. 31 Wenn man ein Modul anbringen will wird der nächste freie Slot hervorgehoben.

5. Gestaltung

Allgemein ist die Gestaltung minimalistisch gehalten um das Gameplay hervorzuheben und den Gestaltungsaufwand möglichst gering zu halten.

5.1. Schiffskern

Die Schiffskerne wurden auf das Nötigste reduziert: die runde Form symbolisiert die Fragilität des Kerns (Zerstörung mit einem Schuss), die Farbe entspricht der Spielerfarbe („Handles“-Pattern, Bjork & Holopainen, 2005, S. 93). Der Schiffskern und die daran befestigten Module werden ausgegraut angezeigt, wenn der Spieler das Raumschiff in einer Runde bewegt hat. So hat der Spieler eine Übersicht, bei welchen Raumschiffen er noch über eine Aktion entscheiden kann.

Damit die Raumschiffe etwas mehr wie Raumschiffe wirken, haben sie eine schwebende Bewegung.



Abb. 32 Schiffskern

Außerdem habe ich anfangs versucht die Gesundheit der Raumschiffe durch Drohnen anzuzeigen, die um das Schiff herumfliegen. Das wurde allerdings zu unübersichtlich. Schließlich hat das dazu geführt zu testen wie es ist, wenn Schiffskerne und Module mit nur einem Schuss zerstört werden können (so dass keine HP-Anzeige mehr notwendig ist). Das Spiel wurde deutlich intensiver und jeder Zug bedeutungsvoller, deshalb habe ich es so gelassen.

5.2. Module

Die Gestaltung der Module erwies sich als eine der größten Herausforderungen. Da ich das Spiel anfangs für bis zu 4 Spieler entwickeln wollte, habe ich die Schiffskerne mehrfarbig gehalten, und die Module dafür einfarbig. Ich hatte gehofft die Funktion der Module durch einfarbige Symbole mit Binnenstruktur deutlich genug anzeigen zu können. Das war aber leider nicht der Fall, deshalb habe ich mich entschieden das Spiel nur für 2 Spieler zu machen (schwarz und weiß), und dafür den Modulen unterschiedliche Farben zu geben. Das hilft enorm die Module und ihre Funktionen

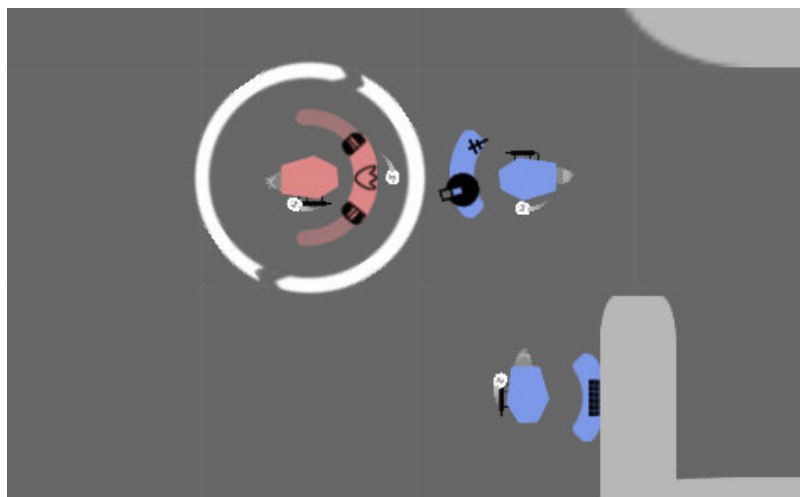


Abb. 33 Versuch, die Funktionen der Module nur durch Silhouetten und Binnenstruktur darzustellen.

voneinander zu unterscheiden. Die Symbole haben dabei dieselbe Farbe wie der Hintergrund. Wenn ein Modul aufgrund des Gebrauchs kurz- oder langfristig nicht mehr verfügbar ist, färbt sich das Modul-Symbol in eine dunklere Variante der Modulfarbe.

Außerdem drehen sich Symbole nicht mit, sondern haben immer die gleiche Ausrichtung, so dass sie sich schneller wiedererkennen lassen.



Abb. 34 Die fünf verschiedenen Module.



Abb. 35 Outpost

5.3. Outposts

Um zu zeigen wie weit ein Outpost aufgeladen ist, befindet sich als „Status Indicator“ (Bjork & Holopainen, 2005, S. 142) ein Ring um dem Outpost, der leer, halb voll oder voll sein kann. Dadurch sollte der Spieler schnell bemerken, dass der Outpost 2 Runden benötigt, um sich wieder voll aufzuladen.

Außerdem haben Outposts das gleiche Weapon-Modul-Symbol wie es Schiffe haben können, wodurch der Spieler an die Angriffs-Funktion des Outposts erinnert wird. Das Anzeigen des Angriffsradius, wenn der Spieler eine Outpost-Aktion durchführen kann, erfüllt den selben Zweck.

5.4. Angriff

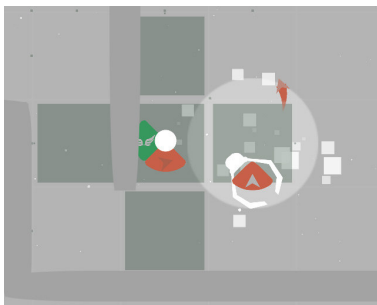


Abb. 36 Angriff

Anfangs habe ich den Angriff durch einen Laserstrahl dargestellt. Um aber das Design kohärent und möglichst einfach zu halten, fliegt jetzt das Weapon-Symbol zum Ziel, zerstört dieses und kehrt wieder zum Weapon-Modul zurück. Dadurch gibt es auch nicht mehr das Logikproblem, dass ein sehr weit schießender Laserstrahl standardmäßig nur die 4 direkt anliegenden Felder angreifen kann.

Nach einem Angriff werden übrigens die Symbole aller Weapon-Module am Schiff ausgegraut angezeigt, um dem Spieler mitzuteilen, dass er seinen Angriff mit diesem Schiff in dieser Runde bereits durchgeführt hat, und kein weiterer mehr zur Verfügung steht.

Anfangs wollte ich den Schiffskernen standardmäßig ein Weapon-Symbol geben, um die Kohärenz zu den Weapon-Modulen zu wahren (da zu dem Zeitpunkt auch noch der Schiffskern allein angreifen konnte). Dadurch kam ich auf die Idee, dass der Schiffskern standardmäßig nicht angreifen kann. Dadurch war kein zusätzliches Symbol auf dem Schiffskern notwendig, das Spiel wurde intensiver und die Taktikmöglichkeiten nahmen zu.

5.5. Hintergrund

Der Hintergrund ist grau mit weißen Sternen, um das Weltraum-Szenario aufzuzeigen und dem Spiel mehr Räumlichkeit zu verleihen. Schiffe und Module heben sich von dem grauen Hintergrund eindeutig ab.

Außerdem sieht man im Hintergrund ein klassisches, kaum sichtbares Raster.

5.6. Musik und Sounds

Die Musik wurde so ausgewählt, dass sie „treibend“ aber ruhig ist, um zum allgemeinen Spielgefühl zu passen.

Musik und Sounds haben Synthesizer-Elemente, um gut zum abstrakten Weltraum-Szenario zu passen.

Musik und Sounds wurden nicht von mir erstellt, aber - wenn es notwendig war und die Lizenz es erlaubt hat - von mir minimal bearbeitet (Lautstärke verändert, Liedlänge gekürzt,...). Die Urheber, Lizenzen und ein Vermerk ob das Werk von mir bearbeitet wurde, sind im Menüpunkt „Credits“ im Spiel einsehbar.

5.7. Wahl des Namens

„Space Rafts“ soll das Spielgefühl widerspiegeln mit oftmals unbewaffneten, sich lediglich bewegen könnenden Raumschiffen ähnlich wie ein Einsiedler auf einem selbstgebauten Floß in einem gefährlichen Meer „ausgeliefert zu sein“, sich durch das Sammeln wertvoller Module - die man nach kurzer Zeit ja doch wieder verliert - möglichst länger als der Gegner am Leben zu halten und einfach ums Überleben zu kämpfen.

Außerdem erinnert der Name an den englischen Begriff „Spacecrafts“, also Raumschiffe.

6. Tutorial

Da das Spiel möglichst schnell erklärt werden muss, ohne den Spieler aufgrund der Komplexität abzuschrecken, habe ich versucht das Tutorial möglichst knapp zu halten, so dass der Spieler alle wichtigen Infos lernen kann, aber auch noch genug zum selbst entdecken hat.

Dafür habe ich zunächst gesammelt, was der Spieler unbedingt wissen muss, und in welcher Reihenfolge man es dem Spieler am Besten beibringen kann.

Zu Beginn werden in wenigen Bildern die wichtigsten Regeln zusammengefasst. Dabei müssen (wenn mit 2 Controllern gespielt wird) beide Spieler bestätigen, dass sie die Regel gelesen haben und bereit für die nächste sind. So wird sichergestellt, dass beide Spieler die Regeln in ihrem Tempo lesen können. Danach werden dem Spieler während des Spielens Hinweise eingeblendet, wenn er bestimmte Aktionen durchführt.

Da alles in einem Tutorial beizubringen zu viel gewesen wäre, habe ich es in 2 Tutorials aufgeteilt:

Tutorial 1

Im ersten Tutorial lernt der Spieler alles, um ein komplettes Spiel spielen zu können. Um den Einstieg möglichst einfach zu halten, gibt es in diesem Tutorial-Level nur Weapon-Module. Wenn eine von beiden Parteien keine Weapon-Module mehr hat, ist das Level beendet, da der Spieler ohne angreifen zu können keine Chance mehr hat zu gewinnen. Denn wie man neue Module erhält erfährt man erst im zweiten Tutorial. Außerdem ist das Level so klein, dass man die Kamera nicht bewegen muss (was man ebenfalls erst im zweiten Tutorial lernt).

Wenn der Spieler das erste Mal ein Schiff auswählt, wird ein Hinweis zur Funktionsweise des Bewegungsradius eingeblendet. Darauf folgen Hinweise zu Markern die anzeigen, ob ein Ziel angegriffen oder ein Modul angebracht werden können. Wenn der Spieler die Taste zum Aufnehmen eines Moduls drückt, erklärt ein Hinweisbildschirm, wie er die richtige Stelle zum Anbringen des Moduls auswählen kann.

Tutorial 2

Im zweiten Tutorial werden dem Spieler Outposts und zusätzliche Tipps beigebracht, so dass er schließlich alles weiß, was er über das Spiel wissen muss.

Wenn das Spiel beginnt, sieht der Spieler auf dem Spielfeld mehrere noch unbekannte Module (insgesamt sind in dem zweiten Tutoriallevel 3 weitere Module zum Spawnen möglich). Wählt er ein Schiff aus wird erklärt, wie er Infos über neue Module anzeigen lassen kann. Wählt er das Schiff mit installiertem „Eagle Eye“ Modul an (also dem Modul, das den Angriffsradius erhöht), erfährt der Spieler, mit welcher Taste er den Angriffsradius anzeigen lassen kann. Daraufhin werden noch Tipps eingeblendet wie man die Kamera bewegen kann und wie man die Runde vorzeitig (ohne alle Einheiten bewegt haben zu müssen) beenden kann.

7. Level Design

Die Levels lassen sich beeinflussen durch folgende Faktoren:

- Wände, die von Raumschiffen unpassierbar sind („Obstacles“-Pattern: Bjork & Holopainen, 2005, S. 75-76). Sie grenzen nicht nur das Spielfeld ab, sondern legen auch die Bewegungsmöglichkeiten der Spieler und mögliche Kampfareale fest. Enge „Gassen“ z.B. provozieren Kampfsituationen. Außerdem kann der Spieler sie taktisch verwenden um den Weg des Gegners zu blockieren.
- Platzierung der Schiffe.
- Anfängliche Ausstattung der Schiffe mit Modulen - welches Schiff hat welche Module?
- Platzierung bereits vorhandener Module auf dem Spielfeld.
- Platzierung der Outposts.

Um „Player Balance“ (Bjork & Holopainen, 2005, S. 399) zu gewährleisten, ist das Levellayout in allen Levels symmetrisch gehalten („Symmetry“-Pattern: ebd., S. 389).

Außerdem haben beide Spieler meist die gleiche Anzahl an Schiffen und Modulen. In „Level 1“ ist das jedoch nicht der Fall, um den Vorteil desjenigen, der zuerst dran ist, auszugleichen.

Es gibt 2 Tutoriallevels und 3 „normale“ Levels:

Tutoriallevel 1

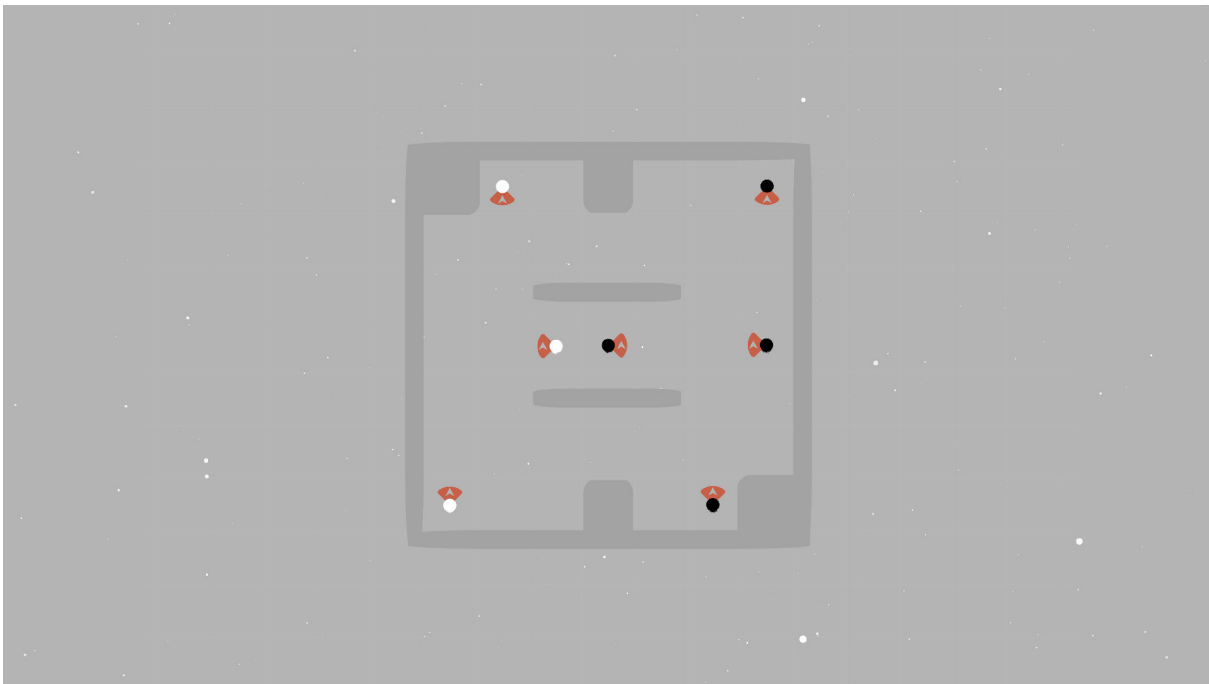


Abb. 37 In dem ersten Tutoriallevel gibt es keine Outposts. Jedes Schiff hat ein Weapon-Modul. Dadurch ist sichergestellt, dass der Spieler im ersten Tutorial nicht überfordert ist.

Tutoriallevel 2

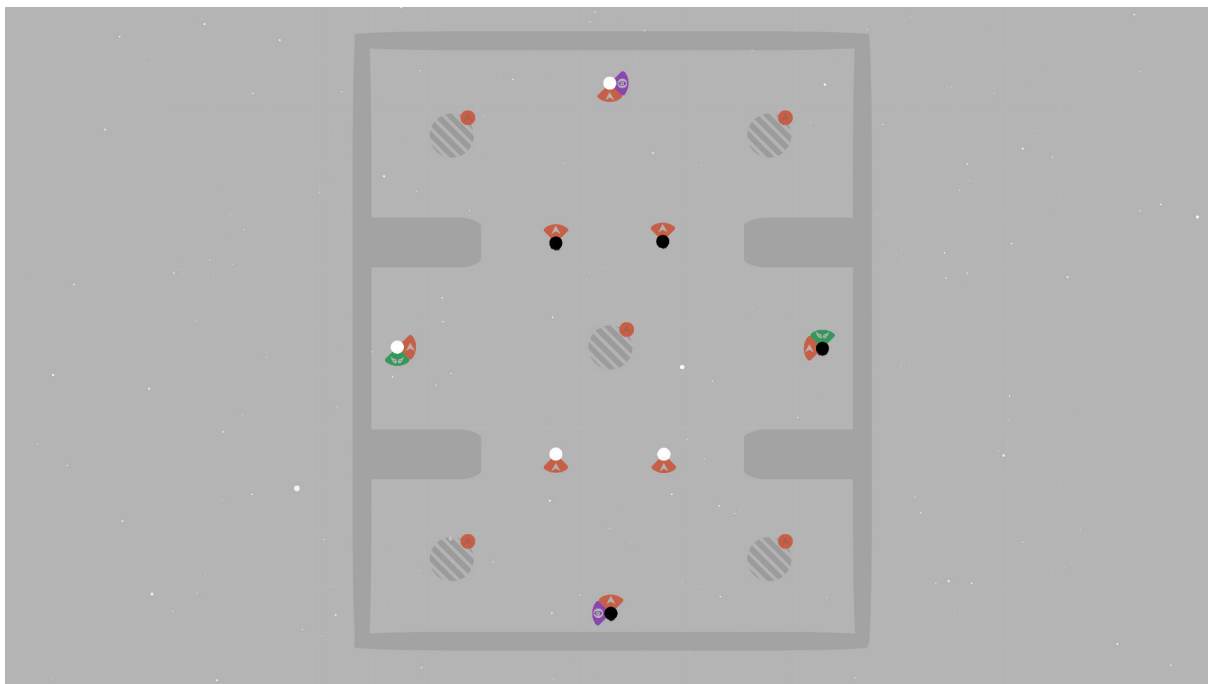


Abb. 38 Das zweite Tutoriallevel ist auf Kämpfe im Zentrum ausgelegt. In diesem Level gibt es abgesehen von dem Decoy Drone-Modul alle Module, damit der Spieler allmählich mit diesen vertraut wird.

Level 1

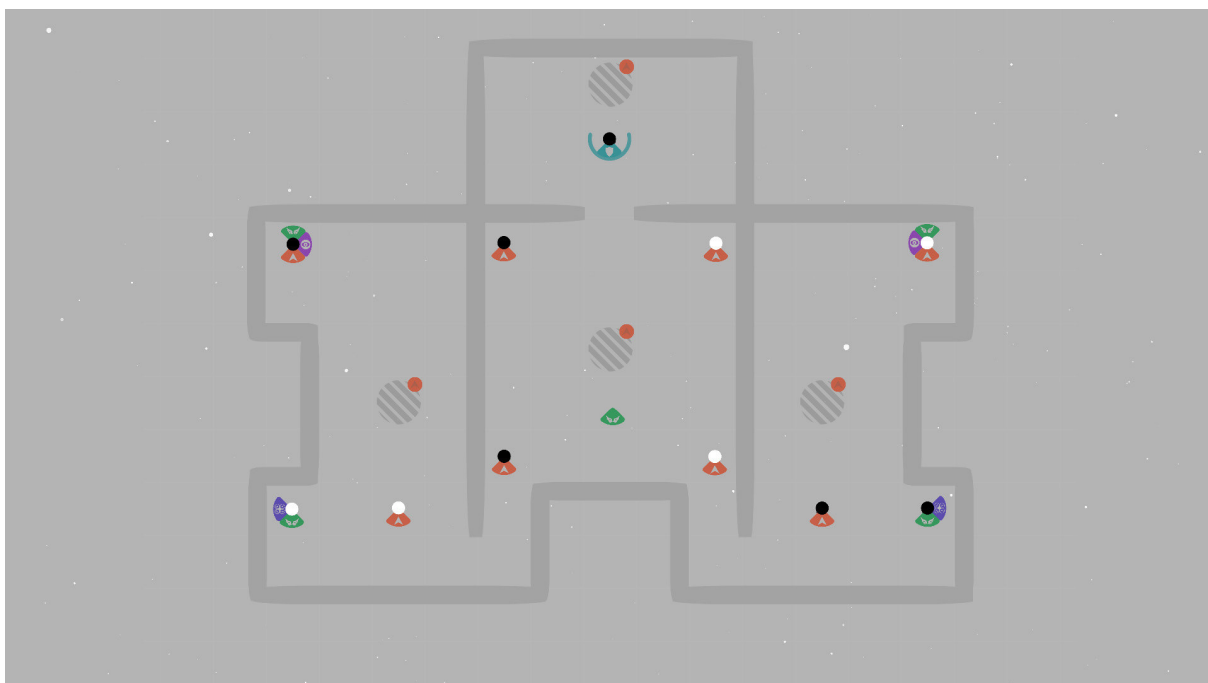


Abb. 39 Dies ist das größte Level. In 4 „Kammern“ kann es bis zu 4 parallele Kampfsituationen geben. Mit Glück kann man in diesem Level durch das Besetzen von Outposts schnell gewinnen.

Level 2

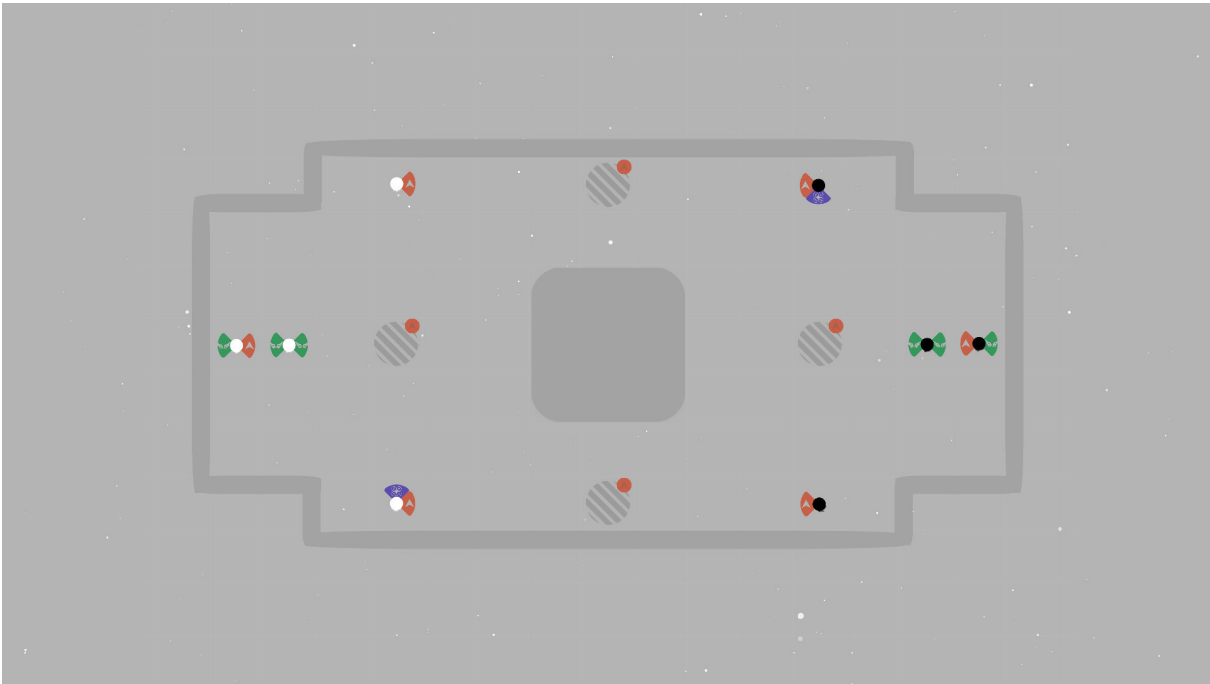


Abb. 40 In diesem Level finden Kämpfe auf keinem zentralen Platz statt, sondern am wahrscheinlichsten in der oberen oder unteren „Gasse“. Die Spieler müssen vorausplanen in welche Gasse sie welche Schiffe mit welchen Modulen schicken. Dabei besitzen die Spieler jeweils u. a. 2 Einheiten mit sehr hohem Bewegungsradius.

Level 3

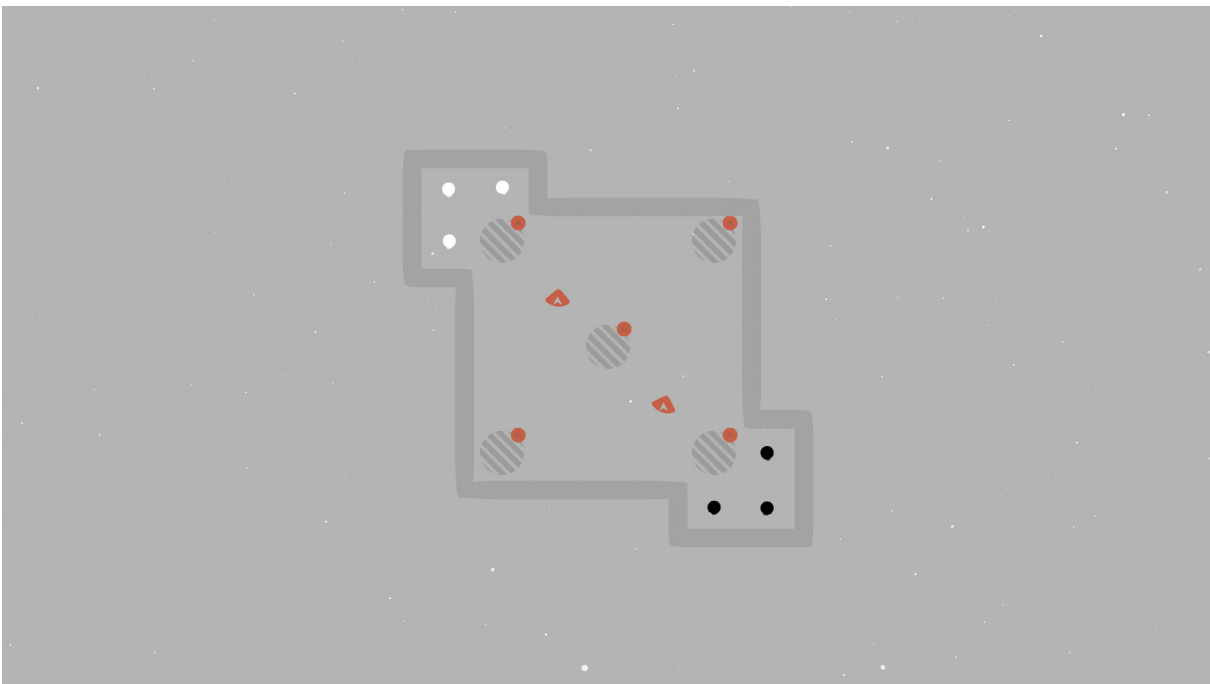


Abb. 41 Dies ist das kleinste Level. Beide Spieler fangen komplett ohne Module an. In dem Level kommt es vor allem darauf an, den Gegner geschickt zu blockieren und dabei aufzupassen, dass man sich nicht selbst im Weg steht.

8. Verwendete Tools

Zu Beginn habe ich mit der kostenlosen Version von Unity 4.2 gearbeitet und schließlich das Projekt auf Unity 4.3 upgegradet. Die letzte Version, mit der alle Screenshots in dieser Bachelorarbeit, der angehangene Projektorder und der angehangene Build entstanden sind, ist die kostenlose Version von Unity 4.3.3.

Zum Programmieren habe ich Microsoft Visual Studio 2013 Professional verwendet.

Alle Grafiken wurden in Adobe Photoshop CS 2 angefertigt.

Sounds und Musik habe ich - wenn es notwendig war und die Lizenz es zugelassen hat - mit Audacity nachbearbeitet.

Literaturverzeichnis

- 17-Bit. (2013). *Skulls of the Shogun* [iOS, PC, Windows Phone, Xbox 360]. Redmond, USA: Microsoft Studios.
- Big Daddy's Creations (2013). *Neuroshima Hex* [Android, Kindle Fire, iOS]. Gleiwitz, Polen: Portal Publishing.
- Bjork, S. & Holopainen, J. (2005). *Patterns in Game Design*. Hingham: Charles River Media.
- Blizzard Entertainment. (2013). *Hearthstone: Heroes of Warcraft* [Android, iOS, PC]. Santa Monica, USA: Activision Blizzard
- Firaxis Games. (2010). *Civilization V* [PC]. Novato, USA: 2K Games.
- Firaxis Games. (2012). *XCOM: Enemy Unknown* [iOS, PC, PS3, Xbox 360]. Novato, USA: 2K Games.
- KING Art Games. (2013). *Battle Worlds: Kronos* [PC]. Hamburg, Deutschland: Crimson Cow.
- Genre Definitions. (k.A.) Abgerufen am 01. Februar 2014, von <http://www.mobygames.com/glossary/genres>
- Mojang (2013). *Scrolls* [PC]. Stockholm, Schweden: Mojang.
- Stainless Games. (2013). *Magic 2014: Duels of the Planeswalkers* [PC, PS3, Xbox 360]. Renton, USA: Wizards of the Coast
- Subset Games (2012). *FTL: Faster Than Light* [PC]. Fremont, USA: Subset Games.

Eidesstaatliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Berlin, den 11. Februar 2014

Markus Sockel