

Duale Hochschule Baden-Württemberg
Mannheim

Seminararbeit

Star Greg
Das Unternehmensplanspiel

Studiengang Wirtschaftsinformatik

Vertiefungsrichtung Softwaremethodik

Verfasser: Britta Jochum
Julia Lakatos
Philipp Mail
Jan Schlenker
Marcel Steinleitner
Fredrik Teschke

Kurs: WWI 10 SWM A

Studiengangsleiter: Prof. Dr.-Ing. Jörg Baumgart

Modul: Umsetzung von Methoden der Wirtschaftsinformatik

Lehrveranstaltung: Fallstudie Systemanalyse

Dozent: Gregor Tielsch

Vorwort

Beginn: Britta

Wer hat nicht schon ein Mal den Wunsch verspürt, Vorstandschef eines Großunternehmens zu sein? Als Teilnehmer eines Unternehmensplanspiels, bei dem der Spieler mit einer realitätsnahen, simulierten Welt interagiert, ist dieser Wunsch gar nicht so weit hergeholt.

Schon seit Jahrhunderten ist man auf der Suche, neue Lernmethoden für die Weiterbildung zu erforschen und auszubauen. Dabei stehen neben theoretischen Methoden auch immer mehr die praktischen im Vordergrund. Eine spezielle Methode, um insbesondere komplexe Sachverhalte oder Systeme, wie beispielsweise die Funktionsweise eines Unternehmens, zu veranschaulichen, stellt hierbei das Planspiel dar. Man unterscheidet haptische (z.B. Brettspiele) und computergestützte Planspiele, die mittlerweile flächendeckend und in hoher Zahl am Markt erhältlich sind. Besonders populär ist das Unternehmensplanspiel TOPSIM der Firma *TATA Interactive Systems*, das an ca. 1500 Hochschulen, Akademien und Unternehmen verwendet wird.¹

Die zunehmende Bedeutung von Planspielen als Lernmethode ist gewiss nicht nur auf deren Anwendung beschränkt. Eine weitere Lernkomponente ergibt sich auf Ebene der Anforderungsanalyse und Entwicklung eines computergestützten Planspiels, was insbesondere für Studierende der Wirtschaftsinformatik große Lernchancen mit sich bringt. Da zu ihrem Bereichsfeld sowohl die betriebswirtschaftlichen als auch software- und programmiertechnischen Anforderungen an Analyse, Entwurf und Implementierung einer derartigen Software gehören, wird diese Aufgabe gerne in den Lehrplan übernommen.

Ende: Britta

¹ Vgl. *TATA Interactive Systems GmbH*, Planspiel-Methode, 2011.

Inhaltsverzeichnis

Verzeichnisse	vi
Abbildungsverzeichnis	vi
Tabellenverzeichnis	vii
Listingverzeichnis	vii
1 Einleitung	1
1.1 Ziel und Aufgabe	1
1.2 Was sind Planspiele?	2
2 Spielwelt	5
2.1 Einleitung	5
2.2 Szenario	5
2.2.1 Beschreibung der Spielwelt	5
2.2.2 Konjunkturverlauf	6
2.3 Unternehmen	9
2.4 Datenbasis	11
2.4.1 Einleitung	11
2.4.2 Raumschiffe	12
2.4.3 Personal	15
2.5 Spielablauf	18
2.5.1 Informationsphase	19
2.5.2 Steuerungsphase	19
2.5.3 Interaktionsphase	20
2.5.4 Simulationsphase	20
2.6 Spiellogik	21
2.6.1 Einleitung	21
2.6.2 Bauteilpreise	22
2.6.3 Absatzmengen	26
2.6.4 Fehlerkosten	29
2.6.5 Bewertung	32
3 Fachkonzept	34
3.1 Einleitung	34

3.2	UseCase Diagramm	34
3.2.1	Einleitung	34
3.2.2	Spielleiter	35
3.2.3	Spieler	36
3.3	Aktivitätsdiagramm	37
3.3.1	Übersicht	37
3.3.2	Personal verwalten	39
3.3.3	Bauteile einkaufen	40
3.3.4	Produktionsauftrag anlegen	40
3.3.5	Verkaufsangebot abgeben	41
3.4	Entwurfsentscheidungen	42
3.4.1	Einleitung	42
3.4.2	Die Kernelemente	43
3.4.3	Die Typen	44
3.4.4	Die Märkte	46
3.4.5	Die Abteilungen	47
3.4.6	Die Transaktionen	49
3.5	Implementierung	50
3.5.1	Einleitung	50
3.5.2	Erläuterung	52
3.5.3	Das Hilfspaket <i>util</i>	54
3.5.4	Umsetzung der Spiellogik	56
3.5.5	Entscheidungsfindung der Unternehmen	60
3.5.6	Spielablauf	61
4	UI Mockups	62
4.1	Einleitung	62
4.2	Startbildschirm	62
4.3	Spielübersicht	63
4.4	Finanzen	65
4.5	Personalwesen	67
4.6	Einkauf	69
4.7	Produktion	71
4.8	Verkauf	73
4.9	Endbewertung	74
5	jUnit Tests	77
5.1	Einleitung	77
5.2	Test der grundlegenden Spiellogik	77
5.2.1	Test der Klasse BauteilTyp	78
5.2.2	Test der Klasse BauteilMarkt	78
5.2.3	Test der Klasse RaumschiffTyp	79

5.2.4	Test der Klasse RaumschiffMarkt	80
5.3	Test des Spielablaufs	81
5.3.1	Unternehmens-Archetypen	82
5.3.2	Spielablauf	82
5.3.3	Spielergebnis	83
6	Fazit und Ausblick	85
6.1	Einleitung	85
6.2	Erweiterte Funktionen im Spielverlauf	85
6.2.1	Finanzwesen	85
6.2.2	Produktion	86
6.2.3	Zusatzaufträge	86
6.2.4	Beeinflussung der Abnehmer	86
6.3	Technische Erweiterungen	87
6.3.1	Benutzeroberfläche	87
6.3.2	Datenbankanbindung	87
6.4	Projektorganisation	87
A	Anhang	89
A.1	Klassendiagramm der Entwurfsphase	89
Literaturverzeichnis		93

Verzeichnisse

Abbildungsverzeichnis

1:	Konjunkturverlauf	8
2:	Unternehmensstruktur	9
3:	Spielablauf	18
4:	Die Beziehung zwischen Angebot und Nachfrage	22
5:	Legende zum Algorithmus	23
6:	Die Berechnung des Bauteilpreises	24
7:	Preis-Absatz-Funktion	27
8:	UseCase Diagramm	35
9:	Aktivitätsdiagramm	38
10:	Aktivitätsdiagramm zur Personalverwaltung	39
11:	Aktivitätsdiagramm zum Bauteileinkauf	40
12:	Aktivitätsdiagramm zur Produktion	41
13:	Aktivitätsdiagramm zum Verkauf	42
14:	Klassendiagramm - Kernelemente	44
15:	Klassendiagramm - Typen	45
16:	Klassendiagramm - Märkte	46
17:	Klassendiagramm - Abteilungen	47
18:	Klassendiagramm - Abteilungen	49
19:	Startbildschirm	63
20:	Übersichtsbildschirm	65
21:	Finanzen	67
22:	Personalwesen	69
23:	Container Triebwerk	70
24:	Einkauf	71

25:	Produktion	72
26:	Verkauf	74
27:	Auswertung pro Runde	75
28:	Gesamtauswertung	76

Tabellenverzeichnis

1:	Preise der Grundbauteile	12
2:	Preise der Sonderbauteile	13
3:	Bauteilmengen der Raumschiffe	14
4:	Lagerkosten der Raumschifftypen zu Beginn	14
5:	Zusatzkosten der Raumschifftypen zu Beginn	15
6:	Werbungskosten des Personals	17
7:	Laufende Kosten des Personals	17
8:	Beispielwerte für die Berechnung neuer Bauteilpreise	24
9:	Beispiel zur Berechnung des Absatzes - 1	28
10:	Beispiel zur Berechnung des Absatzes - 2	28
11:	Beispiel zur Berechnung des Absatzes - 3	29
12:	Beispiel zur Berechnung des Absatzes - 4	29
13:	Beispiel zur Berechnung der Fehlerkosten	31
14:	Speicherung von Daten in unterschiedlichen Klassen	55
15:	Rangfolge als Spielergebnis	83

Listingverzeichnis

1:	HashMap zur Lagerung von verschiedenen Produkttypen	52
2:	<i>einlagern()</i> der Klasse LagerAbteilung	52
3:	<i>kaufeEin()</i> der Klasse EinkaufsAbteilung	54
4:	<i>berechnePreis()</i> der Klasse BauteilTyp	57
5:	<i>berechnePreise()</i> der Klasse BauteilMarkt	58
6:	<i>berechneTypAbsatz()</i> der Klasse RaumschiffMarkt (1)	58
7:	<i>berechneTypAbsatz()</i> der Klasse RaumschiffMarkt (2)	58

8:	<i>berechneFehlerhafteMenge()</i> der Klasse ProduktionsAbteilug	59
9:	<i>berechneFehlerhafteMenge()</i> der Klasse ProduktionsAbteilug	60
10:	<i>testBerechnePreis()</i> der Klasse BauteilTypTest	78
11:	<i>testBerechnePreise()</i> der Klasse BauteilMarktTest	79
12:	<i>testGetKosten()</i> der Klasse RaumschiffTypTest	79
13:	<i>testGetLagerkosten()</i> der Klasse RaumschiffTypTest	80
14:	<i>testGetKosten()</i> der Klasse RaumschiffTypTest	81
15:	<i>testGetLagerkosten()</i> der Klasse RaumschiffTypTest	81

1 Einleitung

Beginn: Britta

1.1 Ziel und Aufgabe

Die vorliegende Arbeit stellt die Studie einer Gruppe von fünf Studierenden der Wirtschaftsinformatik (Schwerpunkt Softwaremethodik) an der DHBW Mannheim vor. Im Rahmen der Vorlesung Systemanalyse wurde über zehn Wochen an der Konzeption des Unternehmensplanspiels 'Star Greg' gearbeitet. Vorgaben zur konkreten Aufgabe der Fallstudie beschränken sich inhaltlich auf ein Unternehmensplanspiel, das wie es der Name schon sagt, betriebswirtschaftliche Geschäftsprozesse eines Unternehmens abbildet und auf mehrere Spieler ausgelegt ist, sodass ein markttypisches Angebot- und Nachfrageverhalten durch den Oligopolmarkt gegeben ist. Die Wahl der Branche des Unternehmens und die Art der Produkte oder Dienstleistungen, die es anbietet, sowie die Welt darf frei gewählt werden.

Wie in der Aufgabenstellung gefordert, stellt diese Arbeit nach einer kurzen Einführung in die Thematik der Planspiele die simulierte Spielwelt vor. Im Unternehmensspiel Star Greg wird in Anlehnung an *Star Wars* eine Welt im Universum simuliert, in der ein Fertigungsunternehmen Raumschiffe produziert. Die Komponenten und Akteure dieser Welt, sowie ein Einblick in die Spiellogik und ein typischer Spielablauf, sollen hier die grundlegende Konzeption von Star Greg näher beleuchten.

Im nächsten Kapitel wird das Fachkonzept bzgl. statischer und dynamischer Modelle umfangreich vorgestellt. Anhand der Erörterung verschiedener Problemstellungen werden wichtige, grundlegende Entwurfsentscheidungen begründet.

Anhand einer Auswahl aussagekräftiger Screens des UI Mockups werden die theoretischen Grundlagen dieses Planspiels transparenter gemacht. Die verschiedenen Screens sollen v.a. eine Vorstellung davon geben, wie der Spieler am Ende interagieren kann.

Schließlich wird das Gesamtergebnis auf Implementierungsebene in einer jUnit Testfallbeschreibung zusammengefasst. Während der erste ausgewählte Test die Spiellogik behandelt, veranschaulicht der zweite jUnit Test, wie Star Greg bei einer Spieleranzahl von drei Teilnehmern, deren vordefinierte Eingaben auf Grundlage unterschiedliche Charaktere ermittelt wurden, verläuft.

1.2 Was sind Planspiele?

Planspiele definieren sich je nach Einsatzfeld unterschiedlich in Bezug auf Ablauf, Struktur und Komplexität. Allumfassend ist ihnen nach dem Planspiel-Forscher und Inhaber der Firma UCS Ulrich Creative Simulations GmbH, Dr. Markus Ulrich, folgendes gemein:

Ein Planspiel versetzt die Teilnehmer in eine fiktive Situation, die ein vereinfachtes Abbild der Realität ist. Während mehrerer Spielrunden machen sich die Teilnehmer mit der Situation vertraut, führen Verhandlungen und fällen konkrete Entscheidungen.²

Typisch dabei ist, dass ein Konflikt als Bestandteil sozialer Wirklichkeit beschrieben wird, den es zu erörtern und lösen gilt.

Planspiele finden ihre Ursprünge schon sehr früh. Laut TOPSIM fanden die ersten Planspiele etwa 1000 v.Chr. als Kampfspiele in Indien statt, in Persien 800 v.Chr. sogar auch im Rahmen von Schachspielen. Ebenso werden sie in den USA schon seit langer Zeit in die militärische Ausbildung integriert, da so Kriegszüge besser geplant und durchgeführt werden können, ohne dabei reale Ressourcen zu verbrauchen. Um 1950 wurde das erste computergestützte Unternehmensplanspiel durch die American Management Association entwickelt, das zugleich auch ein einfaches Unternehmensplanspiel darstellte. Bis heute schreitet die Entwicklung dieser Planspiele stark voran, sodass sie zunehmend in sehr vielen Bereichen Anwendung finden.³ So sind sie auch schon an Schulen und Hochschulen wie der Universität Tübingen, fester Bestandteil des Lehrplans. Hier nehmen Studenten ganz Aktuell im Rahmen eines Seminars an dem Planspiel 'National Model United Nations Conference' (NMUN) teil, was die größte Simulation der Arbeit der Vereinten Nationen darstellt und weltweit von Studenten praktiziert wird. Ziel dieses Planspiel ist es, den Teilnehmern

² LO Lehrer-Online GmbH, Planspiele zur Bildung für nachhaltige Entwicklung, 2011.

³ Vgl. TATA Interactive Systems GmbH, Planspiel-Methode, 2011.

eine Vorstellung von politischen Geschehnissen, Strategien und deren Auswirkungen zu vermitteln.⁴

Eine besondere Stellung unter der Vielfalt von Planspielen nehmen die Unternehmensplanspiele ein, zu denen auch das dieser Arbeit zu Grunde liegende Unternehmensplanspiel Star Greg, gehört. Bei dieser Art des Planspiels handelt es sich nach Gabler um eine modellhafte Simulation von Unternehmensprozessen, in der die Spieler jeweils in der Rolle eines Unternehmensexchefs agieren. Das heißt, sie Konkurrieren mit ihren Gegner, den anderen Unternehmen, auf einem Oligopolmarkt und sind zu Beginn des Spiels alle mit gleichen Startbedingungen, d.h. gleicher Betriebsgröße und Finanzstruktur, ausgestattet.⁵ Die Herausforderung stellt sich dem Spieler insofern, innerhalb einer simulierten, modellhaften Realität strategisch und betriebswirtschaftlich zu handeln, um das Unternehmen auf dem Konkurrenzmarkt aufrecht zu erhalten. Dabei unterliegt die simulierte Welt verschiedenen dynamischen Konzepten, wie dem Konjunkturverlauf oder dem Zusammenspiel von Angebot und Nachfrage. In der Regel sind Unternehmensplanspiele rundenbasiert aufgebaut, sodass bei jeder neuen Runde Pläne und Teilpläne sorgfältig aufgestellt und anhand derer mehrere Entscheidungen getroffen werden müssen. Die Auswirkungen dieser Entscheidungen werden durch Algorithmen ermittelt. Normalerweise dauert eine Spielrunde bis zu mehrere Stunden, das ganze Unternehmensplanspiel bis zu mehreren Tagen.⁶

Ziel eines Planspiels ist es, den Teilnehmern eine Vorstellung davon zu geben, wie ein Unternehmen als Ganzes funktioniert und wie Entscheidungen zu treffen sind, um im Wettbewerb erfolgreich bestehen zu können. Durch Zwischenbewertungen kann der Spieler sein Verhalten und die daraus resultierenden wirtschaftlichen Konsequenzen und Geschäftsprozesse analysieren und anhand dessen seine strategischen und operativen Entscheidungen optimieren. Wichtig ist hierbei nicht nur, dass der Spieler Situationen erkennt und analysiert, sondern insbesondere auch, dass er die Situation einschätzen kann und lernt, auf die richtig Art und Weise zu reagieren. Das Kriterium der Einschätzung von Entscheidungen ist eine herausfordernde Disziplin, die dann aber die Fähigkeit liefert, zwischen Alternativen richtig wählen zu können.⁷

Der große Vorteil der Unternehmensplanspiele liegt ganz klar in ihrer praktischen Orientierung. Durch Learning-by-doing wird ein interaktiver, handlungsorientierter und zeitgemäßer Lernprozess verwendet, der zu dem Kreativität und Selbstständigkeit zulässt. Durch die Simulation der Spielwelt kann der Teilnehmer - ähnlich wie

⁴ Vgl. Romann, Entstehung eines Roadshow-Planspiels, 2010.

⁵ Vgl. Gabler Wirtschaftslexikon, Unternehmensplanspiel, 2011.

⁶ Vgl. Romann, Entstehung eines Roadshow-Planspiels, 2010.

⁷ Vgl. Baum, LOGOS/QU, 1995, S. 12-14.

ein Pilot in einem Flugsimulator - seine Fähigkeiten ohne nennenswertes Risiko ausüben und auf die Probe stellen. Dabei ist der unterhaltsame Aspekt nicht Zweck, sondern "Verstärker" des Lernprozesses. Häufig treten die Spieler gruppenweise gegeneinander an, was auch noch Skills im Bereich Teamwork und sozialem Umfang fördert.⁸

Ende: Britta

⁸ Vgl. Fleßa, Das computergestützte Planspiel MOSHI, 1996, S. 54-55.

2 Spielwelt

2.1 Einleitung

Beginn: Julia

Dieses Kapitel dient dazu, einen ausführlichen Überblick über die entwickelte Spielwelt zu schaffen, mit welcher der Spieler konfrontiert wird. Hierzu zählt einerseits das Szenario und den Konjunkturverlauf, was sich auf die Spielumgebung bezieht. Zudem wird auch das Unternehmen beschrieben, welches der Spieler führen muss. Ein weiterer Punkt dieses Kapitels stellt die Datenbasis dar. Hierbei wird genauer auf die Kosten für Raumschiffe, Bauteile und Personal eingegangen. Im Spielablauf werden die einzelnen Phasen, die der Spieler durchläuft, erläutert. Dieser beinhaltet die Informationsphase, die Steuerungsphase, die Interaktionsphase und die Simulationsphase. Als letzter Punkt wird die Spiellogik erläutert. Hierzu zählt die Veränderung von Bauteilpreisen, das Zustandekommen von Absatzmengen, die Entstehung von Fehlerkosten und die Bewertung der Spieler.

Ende: Julia

2.2 Szenario

2.2.1 Beschreibung der Spielwelt

Beginn: Julia

Wir befinden uns im Jahre 2271, Sternzeit 7410,2. Automobile oder andere antike Fortbewegungsmittel sind kaum mehr anzutreffen. Das Volk bedient sich nun an Raumschiffen, welche für unterschiedliche Zwecke konzipiert wurden. Zu nennen sind dabei die Forschungs-, Transport- und Militärraumschiffe. Da in der Vergangenheit

zunehmend Probleme bei Raumschiffherstellern aufgetreten sind, welche man auf Entscheidungen des höheren Managements zurückführen konnte, ist es unumgänglich dieses mithilfe des rundenbasierten Planspiels Star Greg zu schulen. Die Commander der Raumschiffhersteller finden sich in folgender Situation wieder.

Die wenigen Unternehmen in dieser Galaxis, die Raumschiffe anbieten, sind mit den grundlegendsten Abteilungen des 21. Jahrhunderts ausgestattet. Diese umfassen derzeit das Finanzwesen, das Personalwesen, den Einkauf, die Produktion und den Vertrieb. Droiden haben sich seit geraumer Zeit als nützliche und zuverlässige Arbeitnehmer erwiesen und sind daher als Personal in einem solchen Unternehmen unumgänglich. Dank früherer Forschungsarbeiten sind diese in drei verschiedenen Stufen verfügbar wodurch die Personalstruktur allein den Entscheidungen des Commanders unterliegt. Dieser hat zudem die Aufgabe, das Produktionsprogramm der Raumschiffe aufzustellen und die dafür nötigen Bauteile auf dem intergalaktischen Bauteilmarkt zu erwerben. Hierbei ist vor allem die Zusammensetzung der Raumschiffe durch die verschiedenen Bauteile zu beachten. Produzierte Raumschiffe werden dann durch Angabe eines Preises zum Verkauf angeboten. Des Weiteren hat der Commander Einblick in die Finanzlage seines Unternehmens wodurch er seine Einnahmen und Ausgaben bis auf den kleinsten klingonischen Credit (C) überprüfen kann. Hat der Commander seine unternehmerischen Tätigkeiten für diese Runde abgeschlossen, bestätigt er diese und gelangt dadurch die nächste Runde. Hier wird er mit den Verkaufszahlen der vorherigen Runde konfrontiert und muss nun auf ein Neues Entscheidungen treffen und seine unternehmerischen Fähigkeiten auf die Probe stellen. Ist die vorgegebene Anzahl von zehn spannenden und ereignisreichen Runden beendet, so wird der erfolgreichste Commander gekürt und ausgezeichnet.

Ende: Julia

2.2.2 Konjunkturverlauf

Beginn: Marcel, Julia

Um das Spiel dynamisch zu gestalten, wird ein Konjunkturverlauf, welcher auf Abbildung 1 auf S. 8 zu sehen ist, über die angedachten zehn Spielrunden fest vorgegeben. Dies soll zum einen das Spiel für die Teilnehmer spannend gestalten und diese durch eine vorgegebene Storyline motivieren. Zum anderen soll der Konjunkturverlauf die Gesamtnachfrage nach jedem Raumschiffstyp sowie die Personalkosten steuern. Die Storyline zieht sich über die kompletten zehn Runden und wird als Ereignisse den Teilnehmern mitgeteilt. Diese Ereignisse beeinflussen sowohl die Nachfrage nach

jedem einzelnen Raumschiffotyp als auch die Kosten für das Personal. Folgende Ereignisse wurden für die Storyline festgelegt:

- intergalaktischer Streik der Droiden
- Entdeckung des neuen Planeten Alpha Centauri VI
- Kriegserklärung der Klingonen

Zu Beginn des Planspiels, also in Periode 1, wird folgende Nachfragesituation angenommen. Die Kriegsraumschiffe werden mit 60 Einheiten nachgefragt, Handelsraumschiffe dagegen nur mit 30 Einheiten und Forschungsraumschiffe sogar nur mit 20 Einheiten. In der zweiten Periode steigt die Nachfrage nach Kriegs- und Forschungsraumschiffen, die Nachfrage nach Handelsraumschiffen bleibt gleich. Dabei bleiben die Personalkosten konstant bei 100% der festgelegten Werte. Das erste Ereignis wird in der dritten Periode ausgelöst. Der intergalaktische Streik der Droiden führt zu steigenden Personalkosten für die Unternehmen. Während diesem Streik steigen die Personalkosten auf 150% der Anfangswerte.

Von Periode 4 bis Periode 6 steigt die Nachfrage nach allen drei Raumschifftypen. Hierbei ist die gestiegene Nachfrage nach Forschungsraumschiffen besonders zu vermerken. In der vierten Periode sind bei den Personalkosten noch die letzten Ausläufer des Streiks der vorhergehenden Periode zu verzeichnen. Sie belaufen sich auf 120% der Anfangswerte. In Periode 5 und 6 haben sich dagegen die Kosten für das Personal wieder vollkommen erholt, sodass sich die Personalkosten wieder bei 100% eingependelt haben.

Das zweite Ereignis wird in Periode 7 mit der Entdeckung des neuen Planeten Alpha Centauri VI ausgelöst. Dies ist auf vermehrte Forschungsaktivitäten der letzten Perioden zurück zu führen. Die Nachfrage nach Kriegs- und Handelsraumschiffen ist in dieser Periode, im Gegensatz zu den Handelsraumschiffen, noch einmal stark gestiegen. Durch die Entdeckung des neuen Planeten herrscht ein erhöhtes Angebot auf dem Personalmarkt wodurch die Personalkosten auf 80% der Anfangswerte sinken. In der folgenden Periode stagniert die Nachfrage nach allen drei Raumschifftypen. In dieser Periode erholen sich die Personalkosten zu Ungunsten der Unternehmen und steigen auf 90% der Anfangswerte.

Das letzte Ereignis findet in Periode 9 statt. Zu diesem Zeitpunkt haben die Klingonen den Kriegszustand ausgerufen wodurch die Nachfrage nach Kriegsraumschiffen erheblich angestiegen ist. Die Nachfrage nach Handels- und Forschungsraumschiffen ist jedoch stark zurück gegangen, da diese Bereiche unter diesen Umständen vernachlässigt werden. Der starke Anstieg der produzierten Raumschiffe führt zu einem erhöhten Personalbedarf. Dieser fällt zu Ungunsten der Unternehmen aus, da

ein Anreiz in dem Unternehmen anzufangen geschaffen werden muss. Dies bedeutet für die Unternehmen, dass die Personalkosten auf 120% der Anfangswerte steigen.

In der letzten Periode geht die Nachfrage nach Kriegsraumschiffen etwas zurück da ein Waffenstillstandsabkommen getroffen wurde. Forschungsraumschiffe erfahrene in der letzten Periode noch einen kurzen Aufschwung, die Nachfrage nach Handelsraumschiffen bleibt jedoch gleich. Während der letzten Periode belaufen sich die Personalkosten wieder auf die gleiche Höhe wie in der ersten Periode des Unternehmensplanspiels.

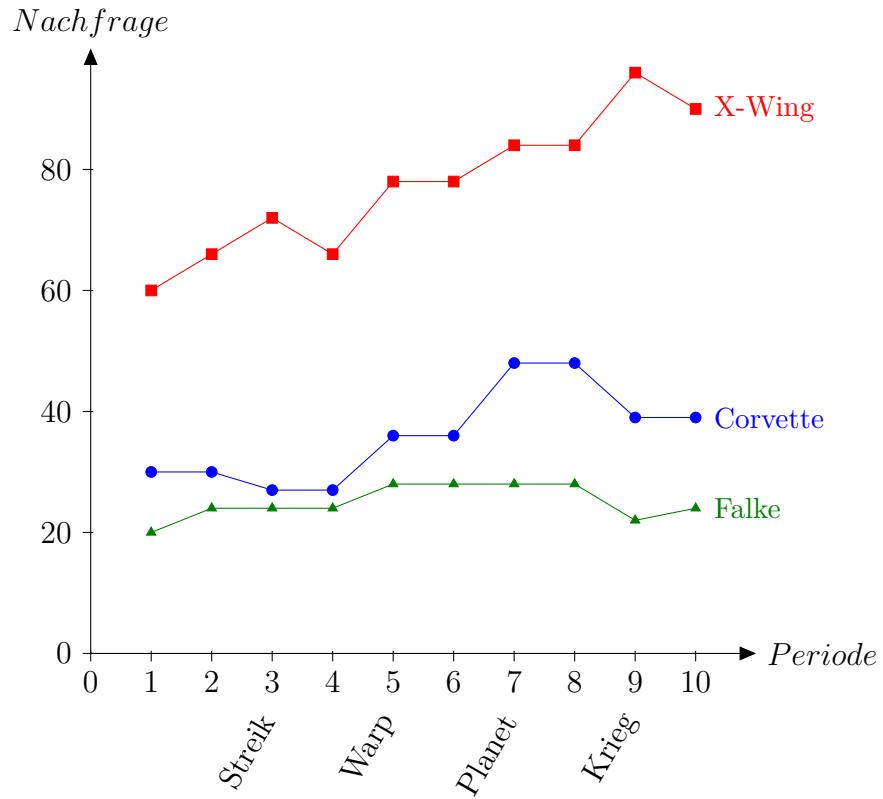


Abbildung 1: Konjunkturverlauf

Ende: Marcel, Julia

2.3 Unternehmen

Beginn: Britta

Ein Unternehmen innerhalb von Star Greg ist in die Bereiche Einkauf, Produktion, Verkauf, Lagerhaltung, Finanzen und Personalwesen eingeteilt (vgl. Abbildung 2).

Im Einkauf werden die zur Produktion benötigten Bauteile zu den aktuellen Bauteilpreisen eingekauft und auf Lager gelegt, sodass dann anhand des Lagerbestandes in der Produktion festgelegt werden kann, wie viele Raumschiffe von den einzelnen Typen hergestellt werden. Es liegen keine Mindest- und Höchstmengen vor, sodass der Spieler hier beliebig variieren kann. Zu beachten sei, dass die gelagerte Menge an Raumschiffen gleich der im Verkauf angebotenen Menge ist. Lediglich auf den Stückpreis kann im Verkauf Einfluss genommen werden. Ob das Angebot zu dem festgelegten Preis voll abgesetzt werden kann, bestimmt die Nachfrage auf dem Markt und das Verhalten der Konkurrenzunternehmen. Diejenigen Raumschiffe, die während der aktuellen Runde nicht abgesetzt werden konnten, verbleiben im Lager.

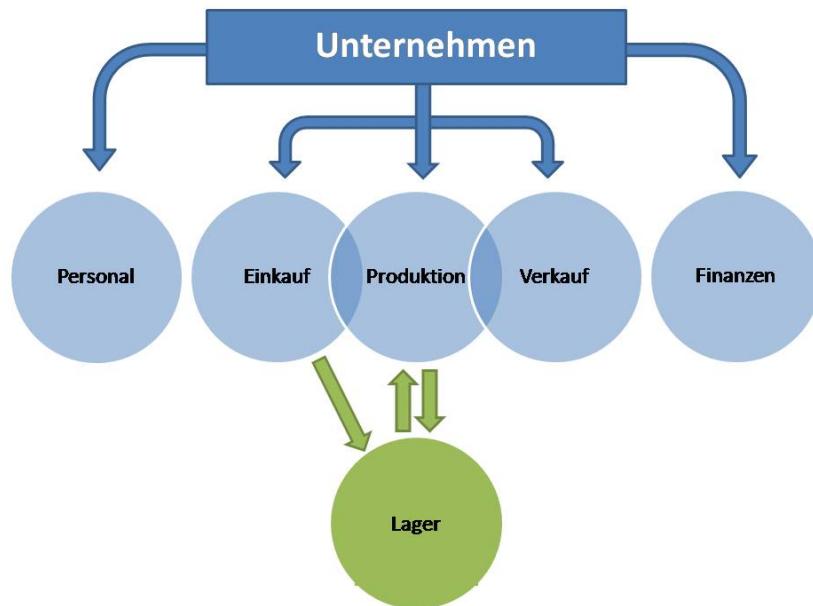


Abbildung 2: Unternehmensstruktur

Für die Organisation und Verwaltung des Lagers wurden zwei Varianten analysiert. Die erste Variante beschreibt im Wesentlichen ein Lager, wie man es in der Realität auch vorfinden würde. Die Lagergröße und damit die Kapazität, Bauteile und

Raumschiffe zu lagern und zu produzieren, sind beschränkt. Pro Runde fallen fixe Kosten an, unabhängig davon, wie stark das Lager ausgelastet ist. Bei einem mittelgroßen Lager und geringer Auslastung, z.B. in der Anfangszeit des Unternehmens, stellt es eine enorme finanzielle Belastung dar. Wählt man das Lager von Anfang an eher klein, so sind die anfänglichen Fixkosten zwar geringer, bei einer Produktionserhöhung fallen aber schnell - schneller als bei einem anfänglichen großen Lager - Sprungfixkosten durch die Zusatzinvestitionen des Anbaus oder Kaufs einer neuen Lagerhalle an.

Variante B, die in Star Greg realisiert wurde, beschreibt eine Art dynamisches Lager. Die Problematik der Fixkosten wird hier umgangen, in dem allein variable Kosten pro eingelagertem Bauteil oder Raumschiff anfallen (vgl. Kapitel 2.4). Die Größe des Lagers kennt keine Kapazitätsgrenze, sodass jederzeit Objekte eingelagert werden können. Konkret heißt das, dass bei einem leeren Lager keine Kosten für das Unternehmen anfallen. Geplant, aber zu diesem Zeitpunkt noch nicht umgesetzt, war außerdem, eine Obergrenze für die Kapazitätsgrenze einzuführen, die durch Zusatzinvestitionen erweitert werden kann, um einen zusätzlichen Parameter zu gewinnen, der die Ausbringungsmenge kontrolliert.

Eine weitere Besonderheit kann sich in der Produktion durch niedrig qualifiziertes Personal ergeben. Je geringer die Qualitätsstufe einer Arbeitskraft ist, d.h. je höher der Anteil der R2D2 und Kampfdroiden im Verhältnis zum Anteil der Droidekas sind, desto größer ist die Wahrscheinlichkeit, dass bei der Produktion eines Raumschiffs Ausschuss entsteht (vgl. Kapitel 2.4.3). In diesem Falle kann auf zwei Arten reagiert werden. Zum einen kann die Personalabteilung ihre Droiden aufrüsten, um vorbeugend in eine zuverlässige Fertigung der Raumschiffe zu investieren. Zum anderen ist es dem Unternehmen auch möglich, bei angefallenem Ausschuss für die mangelhaften Raumschiffe Zusatzinvestitionen zu leisten, um diese doch noch marktfähig zu machen und schließlich zu verkaufen. Die hohe Bedeutung dieses Punktes wird klarer, wenn man bedenkt, dass ein Unternehmen gerade in seiner Anfangsphase, bedingt durch wenige zur Verfügung stehende Ressourcen, nur eine vergleichsweise geringe Zahl an Raumschiffen produzieren und absetzen kann. An dieser Stelle muss es Möglichkeiten geben, die eventuell anfallenden, mangelhaften Raumschiffe durch einen in Relation stehenden Aufwand an den Markt zu bringen. Ist diese Option nicht gegeben, so hätte das Unternehmen keine Chance, sich eine Stellung auf dem Oligopolmarkt zu verschaffen und würde an seinen hohen Anfangseinbußen durch den Totalverlust des Ausschusses zu Grunde gehen.

In diesem Zusammenhang wird auch die Notwendigkeit gewiss, das Konto des Unternehmens, das über die Finanzabteilung verwaltet wird, zu einem festen Zinssatz überziehen zu können. Hier war anfänglich die Überlegung, genau diese Funktion nicht aufzunehmen. Ein Unternehmen könnte so nur diejenigen finanziellen Tätig-

keiten durchführen, für die genügend Kapital vorhanden ist, also möglicherweise die oben angesprochenen Zusatzinvestitionen nicht leisten. Aus diesem Grund wurde im Laufe der Bearbeitung hier eine Änderung vorgenommen. Neben diesem Argument, spricht auch die Tatsache, dass hierdurch risikoreiche Entscheidungen und Aktionen des Unternehmens durchführbar sind, für die Kontoüberziehung.

Ende: Britta

2.4 Datenbasis

2.4.1 Einleitung

Beginn: Jan

Notwendigkeit der Datenbasis

Die Datenbasis des Unternehmensplanspiels Star Greg ist die zentrale Anlaufstelle der Projektbeteiligten zur Beschaffung der vordefinierten Standardwerte für die Bauteilpreise, die Lagerkosten und die Peronaltykosten. Die Motivation zur Erstellung der Datenbasis lag vor allem darin, dass das Bild der Spielwelt transparenter wird und einheitliche Werte für die Implementierung der JUnit Test zur Verfügung stehen. Durch den gemeinsamen Einsatz des Programms Git zur verteilten Versionsverwaltung von Dateien konnte die Anforderung der Einheitlichkeit erreicht werden.

Flexibilität der Datenbasis

Da die Auswirkungen der Standardwerte auf den gesamten Spielverlauf im Vorhinein eines ersten Spieltests kaum absehbar sind, muss gewährleistet sein, dass die Werte innerhalb der Projektdurchführung angepasst werden können. Auch verworfene oder neue Ideen bringen oft eine Veränderung der Werte mit sich, weshalb auch hier die Notwendigkeit der Flexibilität gegeben ist.

2.4.2 Raumschiffe

Raumschiffbauteile

Die Raumschiffbauteile werden im Unternehmensplanspiel Star Greg auf dem Bauteilmarkt gekauft und bilden für die beteiligten Unternehmen die Grundlage für die Produktion der verschiedenen Raumschifftypen. Sie lassen sich in Grund- und Sonderbauteile gliedern. Da die JUnit Tests nachvollziehbar bleiben sollten und die Komplexität des Spiels nicht zu hoch werden durfte, fiel die Wahl auf drei verschiedene Grund- und drei verschiedene Sonderbauteile.

Zunächst wollen wir die Grundbauteile näher betrachten. Zu ihnen gehören das Rumpfbauteil, das Hitzeschild und das Triebwerk, wobei jedes der Bauteile einen Grund-, einen Minimal- und einen Maximalpreis hat. Des Weiteren beansprucht jedes Bauteil einen bestimmten Platz im Lager, sofern es nicht verbaut wurde bzw. sofern das Raumschiff, in dem es verbaut ist, nicht abgesetzt wurde. Die Staffelung der Werte ist in Tabelle 1 dargestellt.

Bauteiltyp	Grundpreis	Minimalpreis	Maximalpreis	Lagereinheiten
Rumpfbauteil	100,00€	60,00€	140,00€	1
Hitzeschild	200,00€	120,00€	280,00€	2
Triebwerk	500,00€	300,00€	700,00€	5

Tabelle 1: Preise der Grundbauteile

Der Gedanke hinter der Preisstaffelung der Grundpreise liegt darin, dass das billigste Bauteil mengenmäßig am meisten und das teuerste Bauteil mengenmäßig am wenigsten für die spätere Produktion der verschiedenen Raumschifftypen benötigt wird. Zudem sollte ein Unternehmensplanspiel betriebswirtschaftliche Aspekte möglichst realitätsnahe in einem Modell wiederspiegeln und eine Preisgleichheit von Produkten wäre eher realitätsfern.

Aus der Tabelle geht hervor, dass sich der Minimal- und der Maximalpreis aus dem Grundpreis wie folgt ableiten:

$$\text{Minimalpreis} = \text{Grundpreis} - 0,4 \cdot \text{Grundpreis} \quad (1)$$

$$\text{Maximalpreis} = \text{Grundpreis} + 0,4 \cdot \text{Grundpreis} \quad (2)$$

Die Grenzen sind für jedes Teil separat gewählt, da das teure Triebwerk beispielswei-

se nie so günstig werden soll wie das oft verwendete Rumpfbauteil. Die Begrenzung selbst ist zudem wichtig für die Berechnung der neuen Bauteilpreise (siehe Kapitel 2.6.2).

Die benötigten Lagerplatzeinheiten für die Bauteiltypen ergeben sich direkt aus dem Grundpreis und werden nicht vom Konjunkturverlauf beeinflusst. Sie errechnen sich, wenn man den jeweiligen Grundpreis des Bauteils durch 100 teilt. Die Lagerplatzkosten je Lagerplatzeinheit sind ebenfalls konstant und haben den Faktor 10, was bedeutet, dass die Lagerung eines Rumpfbauteils über eine Runde beispielsweise 1 Lagerplatzeinheit * 10 €/Lagerplatzeinheit kostet.

Als nächstes betrachten wir die Sonderbauteile. Auch sie besitzen Grund-, Minimal- und Maximalpreis und benötigen jeweilige Lagerplatzeinheiten bei der Lagerung. Die Werte sind in Tabelle 2 festgehalten.

Sonderbauteiltyp	Grundpreis	Minimalpreis	Maximalpreis	Lagereinheiten
Rumpfbauteil	1000,00€	600,00€	1400,00€	10
Geschütz	2000,00€	1200,00€	2800,00€	20
Forschungsausrüstung	3000,00€	1800,00€	4200,00€	30

Tabelle 2: Preise der Sonderbauteile

Ein Sonderbauteiltyp gehört jeweils zu einem bestimmten Raumschifftyp. Die Staffelung der Preise ist eine Andere als bei den normalen Grundbauteilen, da die Sonderbauteile pro jeweiligen Raumschifftyp nur einmal verbaut werden und sie der Staffelung der Raumschiffpreise zu Beginn des Spiels entsprechen soll (siehe Kapitel 2.4.2). Auch bei den Sonderbauteilen werden die Preise pro Runde neu berechnet, wobei dies separat zu den Grundbauteilen geschieht, da immer weniger Sonderbauteile als Grundbauteile benötigt werden.

Zusammensetzung der Raumschiffe

Aus den eingekauften Bauteilen können die Unternehmen Raumschiffe fertigen lassen, wobei jedes Raumschiff dabei eine fest vorgeschriebene Anzahl von Bauteilen hat. Aufgrund der schwankenden Bauteilpreise schwanken auch die Kosten für die Produktion der Raumschiffe, sofern man keine Bauteile besitzt, die in der letzten Runde eingelagert wurden. In Tabelle 3 ist die sind die Werte für die einzelnen Typen aufgelistet.

Die Preise zur für die Raumschifftypen am Beginn eines Spiels gemessen an den Grundpreisen der Bauteile berechnen sich daher wie folgt:

Raumschiffotyp	Rumpfbauteil	Hitzeschild	Triebwerk	Sonderbauteiltyp
X-Wing	18	6	4	Geschütz
Corellian Corvette	38	16	6	Transportkapsel
Millenium Falke	40	30	10	Forschungsausrüstung

Tabelle 3: Bauteilmengen der Raumschiffe

$$\begin{aligned} X - \text{Wing} &= 18 * 100,00\text{€} + 6 * 200,00\text{€} + 4 * 500,00\text{€} + 1000,00\text{€} \\ &= 6000,00\text{€} \end{aligned}$$

$$\begin{aligned} \text{Corellian Corvette} &= 38 * 100,00\text{€} + 16 * 200,00\text{€} + 6 * 500,00\text{€} + 2000,00\text{€} \\ &= 12000,00\text{€} \end{aligned}$$

$$\begin{aligned} \text{Millenium Falke} &= 40 * 100,00\text{€} + 30 * 200,00\text{€} + 10 * 500,00\text{€} + 3000,00\text{€} \\ &= 18000,00\text{€} \end{aligned}$$

An der Rechnung lässt sich erkennen, dass die jeweiligen Raumschifftypen zu Beginn des Spiels eine Staffelung in ihrem Gesamtpreis aufweisen. Wichtig war es ein Schema zu finden, bei dem jedes der Raumschiffe ein bestimmten Grundbauteiltyp hat, für den der Spieler bei dem jeweiligen Raumschiff am meisten investieren muss. So bezahlt man beim X-Wing anfangs am meisten für Triebwerke (2000,00€), bei der Corellian Corvette für die Rumpfbauteile (3800,00€) und beim Millenium Falken für die Hitzeschilder (6000,00€). Somit ist gewährleistet, dass alle Bauteile preislich gesehen steigen und schwanken können (vgl. Kapitel 2.6.2).

Die Lagerkosten für die Raumschiffe ergeben sich aus den Lagerkosten der jeweils verwendeten Bauteile multipliziert mit deren Menge. Da die Lagerplatzpreise/Lagerplatzeinheit am Anfang des Spiels einem Zehntel des jeweiligen Grundbauteils entsprechen, sind die Lagerpreise für die Raumschiffe hier auch noch gestaffelt wie die Raumschiffpreise (siehe Tabelle 4). Ein Spieler muss also sehr genau abwägen, ob er seine Raumschiffe wirklich am Raumschiffmarkt verkaufen kann, da die anfallenden Lagerkosten sein Kapital bei Nichtverkauf erheblich schmälern können.

Raumschiffotyp	Startpreis	Lagerkosten zu Beginn
X-Wing	6000,00€	600,00€
Corellian Corvette	12000,00€	1200,00€
Millenium Falke	18000,00€	1800,00€

Tabelle 4: Lagerkosten der Raumschifftypen zu Beginn

Desweiteren können in der Raumschiffproduktion durch den Einsatz von qualitativ minderwertigem Personal Zusatzkosten für die Produktion entstehen (siehe Kapitel 2.4.3). Diese Zusatzkosten errechnen sich ebenfalls aus den Werten der derzeitigen Bauteilpreise. Dabei wird jedes Bauteil mit dem festen Wert von 50% verrechnet, sodass die Zusatzkosten eines Bauteils meist der Hälfte des Einkaufspreises des Bauteils selbst entsprechen (siehe Tabelle 5). Auch hier muss der Spieler seine Entscheidungen genau durchdenken, da die Einstellung von minderwertigem Personal zu sehr hohen Zusatzkosten führen kann.

Raumschifftyp	Startpreis	Zusatzkosten zu Beginn
X-Wing	6000,00€	3000,00€
Corellian Corvette	12000,00€	6000,00€
Millenium Falke	18000,00€	9000,00€

Tabelle 5: Zusatzkosten der Raumschifftypen zu Beginn

Anfallende Zusatzkosten sollten von den Spielern tendenziell eher bezahlt als nicht bezahlt werden, weil ansonsten die Gesamtkosten der verwendeten Bauteile und der Personalaufwand für das entsprechende Raumschiff keine Einnahmen generiert haben. Aufgrund dieser Tatsache fiel auch die Entscheidung, dass die Konten in Star Greg von dem jeweiligen Spieler überzogen werden können. So hat ein Spieler, der durch seine Produktion bereits an seiner Kapitalgrenze angelangt ist, trotzdem noch die Möglichkeit, seine reparaturbedürftigen Raumschiffe zum Verkauf bereit zu machen und sie abzusetzen. Der Zinssatz bei Überziehung des Kontos von 25% ist dabei über die verschiedenen Spielrunden immer der Gleiche.

Ende: Jan

2.4.3 Personal

Beginn: Marcel

Einleitung

In dem Unternehmensplanspiel Star Greg werden zur Produktion von Raumschiffen sogenannte Droiden benötigt. Abhängig von der laufenden Produktion werden somit unterschiedlich viele Droiden benötigt, die selbst wiederum in 3 verschiedene Qualitätsstufen untergliedert sind.

Qualität und Quantität des Personals

Die drei unterschiedlichen Qualitätsstufen werden in Form verschiedener Droiden dargestellt. Es handelt sich hierbei um den einfachen R2D2, den höherwertigen Kampfdroiden und den Premiumdroiden Droideka.

Der Droide R2D2 besitzt die niedrigste Qualitätsstufe mit einer Qualität von 69 %. Dies bedeutet, dass die in die Produktion gegebenen Raumschiffe mit einer Wahrscheinlichkeit von 69 % fehlerfrei produziert werden können. In diesem Beispiel wird jedoch vorausgesetzt, dass in der Produktion ausschließlich R2D2 Droiden tätig sind.

Eine Qualitätsstufe höher ist der Kampfdroide angesiedelt. Ein Kampfdroide produziert mit einer 84 prozentigen Wahrscheinlichkeit fehlerfreie Raumschiffe. In dem Unternehmensplanspiel Star Greg können Droiden unterschiedlicher Qualitätsstufen zur selben Zeit in der Produktion aktiv sein. Die Qualitätsstufe des gesamten Personals wird in diesem Fall anhand eines Algorithmus berechnet, der in dem Kapitel „Fehlerkosten“ beschrieben wird.

Der Droide mit der höchsten Qualitätsstufe heißt Droideka. Er zeichnet sich durch seinen hohen fehlerfreien Durchsatz in der Produktion aus. Ein Raumschiff, das ausschließlich von Droidekas produziert wurde ist zu 99 % fehlerfrei.

Durch umfassende Umbaumaßnahmen eines Droiden, die mit Schulungen vergleichbar sind, können Droiden ihre Qualitätsstufe verbessern. Der Droide R2D2 kann somit durch zwei Umbaumaßnahmen bis hin zu einem Droideka aufsteigen. Kampfdroiden benötigen lediglich eine weitere Umbaumaßnahme, bis sie zu einem Droideka werden. Der Droideka besitzt bereits die höchste Qualitätsstufe und kann nicht umgebaut werden.

Unabhängig von der Qualität der verschiedenen Droiden haben die Qualitätsstufen keinen Einfluss auf die maximale Produktionsmenge. Dies bedeutet, dass die Kapazitätsgrenze in der Produktion lediglich über die Anzahl der verfügbaren Droiden gesteuert werden kann und nicht durch Umbaumaßnahmen zur Erhöhung der Qualitätsstufe. Man geht davon aus, dass die drei verschiedenen Droiden für die gleiche Aufgabe gleich viel Zeit benötigen. Der einzige Unterschied besteht darin, dass die unterschiedlichen Droiden in dieser Zeitspanne mit einer unterschiedlichen Sorgfalt ihre Arbeit erledigen.

Wie bereits in den vorherigen Kapitel näher erläutert, können in dem Unternehmensplanspiel drei unterschiedliche Raumschiffe produziert werden. Diese unterscheiden

sich in Ihrer Größe (der Anzahl der benötigten Bauteile). Das hat zur Folge, dass pro Raumschiff unterschiedlich viele Droiden benötigt werden. Zur Produktion eines X-Wings werden fünf Droiden benötigt. Eine Correlian Corvette kann von 10 Droiden und ein Millenium Falke von 15 Droiden produziert werden. Aus der Mindestmenge der Raumschiffe, die in der ersten Runde produziert werden müssen und der Zuordnung der Droiden pro Raumschiff ergibt sich so eine Mindestmenge von insgesamt 900 Droiden, die zu Beginn des Unternehmensplanspiels benötigt werden. ($5 \text{ Droiden} * 60 \text{ X-Wings} + 10 \text{ Droiden} * 30 \text{ C.Corvettes} + 15 \text{ Droiden} * 20 \text{ M.Falken} = 900 \text{ Droiden}$)

Personalkosten

Bei der Einstellung eines neuen Droiden fallen Werbungskosten an. Diese richten sich nach der jeweiligen Qualitätsstufe und sind wie in Tabelle 6 abgestuft.

Droide	Werbungskosten
R2D2	400,00€
Kampfdroide	600,00€
Droideka	800,00€

Tabelle 6: Werbungskosten des Personals

Zusätzlich zu den Werbungskosten fallen pro Spielrunde für die Droiden laufende Kosten an. Diese sind mit einem Jahresgehalt vergleichbar und ebenso wie die Werbungskosten abgestuft. (Tabelle 7)

Droide	Laufende Kosten
R2D2	100,00€
Kampfdroide	110,00€
Droideka	120,00€

Tabelle 7: Laufende Kosten des Personals

Wie bereits in dem Kapitel „Qualität und Quantität des Personals“ beschrieben, können Droiden durch Umbaumaßnahmen zu einer höheren Qualitätsstufe aufsteigen. Hierbei wurde beachtet, dass ein Droide der mit einer niedrigen Qualitätsstufe eingestellt wird und durch spätere Umbaumaßnahmen umgebaut wird mehr kostet, als ein Droide der direkt mit einer höheren Qualitätsstufe eingestellt wird. Die Kosten für den Umbau eines R2D2's zu einem Kampfdroiden belaufen sich auf 300 €. Der Umbau eines Kampfdroiden zu einem Droideka kostet den Unternehmer ebenfalls 300 €. Stellt man demnach einen R2D2 Droiden ein und baut ihn zu einem

Kampfdroiden um, so belaufen sich die Gesamtkosten auf 700€ (400 € Werbungskosten + 300 € Umbaumaßnahmen). Stellt man einen Kampfdroiden direkt ein, so kostet dies insgesamt 600 € (600 € Werbungskosten).

Ende: Marcel

2.5 Spielablauf

Beginn: Britta

Der idealtypische Verlauf von Star Greg lässt in vier Phasen unterteilen (vgl. Abbildung 3). Anhand der Beschreibung eines beispielhaften Rundenverlaufs wird dargestellt, was in den einzelnen Phasen geschieht und was der Spieler entscheiden kann.

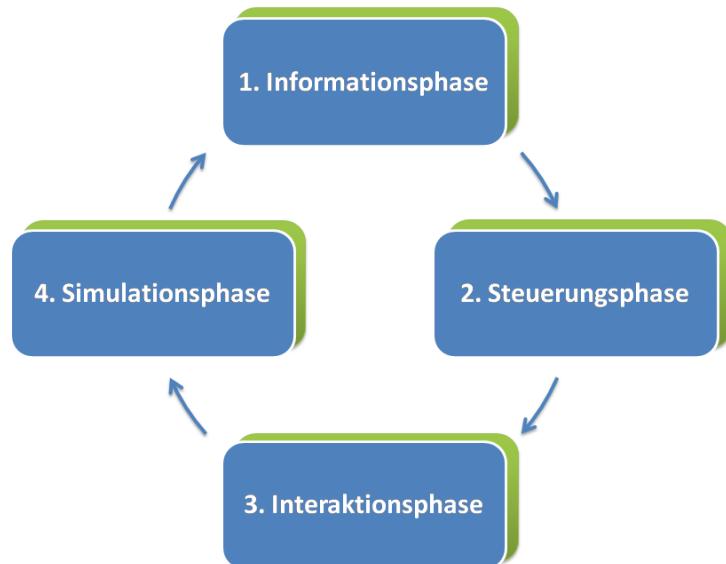


Abbildung 3: Spielablauf

2.5.1 Informationsphase

Spieleinführung

Der Spieler startet Star Greg und bekommt zunächst eine Spieleinführung, in der die Welt als solche deren Akteure und schließlich die von ihm einzunehmende Rolle innerhalb dieser Welt beschrieben wird. Aufgabe des Spielers ist es, im Weltall ein Unternehmen zu führen, das Raumschiffe produziert, und neben seinen Spielgegner auf dem Oligopolmarkt zu existieren oder gar die Marktmacht zu erringen. Die Entscheidungsmöglichkeiten, sowie die rundenbasierte Struktur des Spiels, werden in den Spielregeln offengelegt. Die Spieleinführung ist für alle Teilnehmer gleich. So werden mögliche Verständnisfragen geklärt, was wichtig ist, um allen Spielern die optimalen und insbesondere gleichen Voraussetzungen zu gewährleisten.

Start des Spiels

Beginn legt der Spieler seinen Unternehmensnamen und das Unternehmenslogo fest, unter dem er im Verlauf des Spiels identifiziert wird. Das Startkapital liegt für jeden Teilnehmer bei einer Million Euro. Falls außerordentliche Vorkommnisse vorliegen, wird der Spieler noch bevor er im Spiel aktiv wird, darüber informiert. Ebenso kann er sich an dieser Stelle, sofern er bereits eine Runde gespielt hat, einen Überblick über die Daten und Fakten aus seiner letzten Runde verschaffen. Diese Informationen wurden bewusst auf ein Minimum reduziert, um dem Spieler einerseits ein Gefühl zu geben, wie der Markt auf seine Entscheidungen reagiert, aber andererseits nicht zu viele Informationen über den Erfolg anderer Teilnehmer zu offenbaren:

- 'Star der letzten Runde': Das Raumschiff, das den höchsten Umsatz erzielt hat
- Anzahl der verkauften Raumschiffe
- Marktanteil im Verhältnis zur Branche

In der ersten Runde liegen diese Daten allerdings noch nicht vor!

2.5.2 Steuerungsphase

Nun muss der Spieler aktiv werden, Handlungspotentiale und Möglichkeiten erkennen und abwägen und schließlich Entscheidungen treffen. Dazu werden von ihm

zunächst Bauteile eingekauft. Beim Kauf orientiert er sich an drei Größen:

- momentane Bauteilpreise
- Vorstellung über die Produktionsmenge der einzelnen Raumschifftypen
- Lagerkosten

Demnach sollten also die Bauteile so eingekauft werden, dass eine gewisse Menge des Raumschiffstyps produziert werden kann, der momentan am günstigsten liegt. Zu beachten sei außerdem, dass die im nächsten Schritt angegebene Produktionsmenge, auch gleichzeitig die am Markt angebotene Menge ist, da stets der gesamt Bestand an fertigen Raumschiffen hierbei relevant ist. So fallen in der nächsten Runde im schlimmsten Fall, also wenn kein einziges Raumschiff verkauft werden konnte, Lagerkosten für alle produzierten Raumschiffe an.

Im letzten Schritt dieser Phase müssen die Preise festgelegt werden. Da der Spieler den Marktpreis nicht kennt, muss er auf Grundlage seiner Herstellkosten und seiner Gewinnerzielungsabsicht einen hierfür geeigneten Preis abwägen. An dieser Stelle wird die als optionale Erweiterung festgelegte und bislang nicht umgesetzte Marktforschung relevant. Sie soll es ermöglichen, Informationen über den Markt und die Konkurrenz gegen zusätzlichen Aufwand zu erhalten. So könnte sich der Spieler gerade auch die Preisfindung erleichtern.

2.5.3 Interaktionsphase

Nachdem der Spieler in der Steuerungsphase seine Strategien analysiert und durchdacht hat, gibt er in der Interaktionsphase nun seine endgültigen Entscheidungen ab. Er setzt eine Produktionsmenge und einen zugehörigen Preis fest, und gibt über das Einchecken dieser Daten ein verbindliches Angebot am Markt ab.

2.5.4 Simulationsphase

In dieser Phase werden die Entscheidungen aller Teilnehmer simuliert. Der essenzielle Bestandteil dessen, ist die Umsetzung aller dynamischen Größen über die in Kapitel 2.8 näher erläuterten Algorithmen. Damit wird zum einen die Auswertung der jetzigen Runde eingeleitet und zum anderen vorbereitend die entscheidenden Größen für die nächste Runde ermittelt.

Das Programm kann nun anhand der Summe der nachgefragten Bauteile aller Spieler

in der jetzigen Runde die Bauteilpreise der kommenden Runde berechnen. Ebenso werden der Ausschuss und damit die Fehlerkosten berechnet, sowie die tatsächlichen Absatzmengen für jedes Unternehmen auf dem Raumschiffmarkt bestimmt. So stehen alle nötigen Werte fest, um die Zwischeninformationen („*Star der letzten Runde*“, verkaufte Raumschiffe und Marktanteil) zum Start jeder Runde an den jeweiligen Spieler zu melden.

Ende: Britta

2.6 Spiellogik

2.6.1 Einleitung

Beginn: Fredrik

Bereits sehr früh in der Planungsphase des Projekts kam der Wunsch auf, das zu erstellende Planspiel durch den Einsatz flexibler Spielteile möglichst realitätsnah zu gestalten. Hierbei sollten nicht alle Werte wie z.B. Preise für die Dauer des Spiels fest vorgegeben sein, sondern anhand von dynamischen Einflüssen zur Spielzeit schwanken. Dies macht das Spiel für den Spieler einerseits interessanter, da er die Auswirkungen seiner Entscheidungen und derer seiner Mitspieler auf die Spielwelt beobachten kann. Andererseits erhöht dies auch den Lerneffekt, da Spieler die möglichen Auswirkungen mit in ihre Entscheidungsfindung einbauen und somit verantwortungsvoller handeln müssen.

Um das Planspiel in der vorgegebenen Zeit realisieren zu können, war es nötig, einige Bereiche für diese dynamische Umsetzung auszuwählen und andere außen vor zu lassen. Dabei wurde die Entscheidung gefällt, sowohl die Bauteilpreise, die Absatzmengen auf dem Raumschiffmarkt und die Fehlerkosten flexibel zu gestalten. Nicht in die Betrachtung einbezogen wurden beispielsweise die Personalkosten und die Grundnachfrage auf dem Raumschiffmarkt, die sich stattdessen abhängig von der Spielgeschichte entwickeln, um so die Spannung für die Spieler zu erhöhen.

Die folgenden Abschnitte geben einen Überblick über die Berechnungsvorschriften und Abläufe bei der flexiblen Berechnung der oben erwähnten Werte, woran sich zudem die Beschreibung der Bewertung der Unternehmen am Ende des Spiels anschließt.

2.6.2 Bauteilpreise

Die Bauteile dienen den Unternehmen in dem Planspiel zur Herstellung von Raumschiffen. Für die Preisfindung der Unternehmen sind vor allem die variablen Stückkosten maßgeblich, die sich in der vereinfachten Darstellung des Planspiels lediglich aus der Summe der Bauteilpreise zusammensetzen. Hierbei schwanken die Bauteilpreise, was zunächst der Dynamik des Spiels zu Gute kommt, weil erreicht wird, dass Spieler auf die geänderten Bedingungen reagieren müssen, indem sie erkennen, dass gewissen Bauteilpreise gesunken sind.

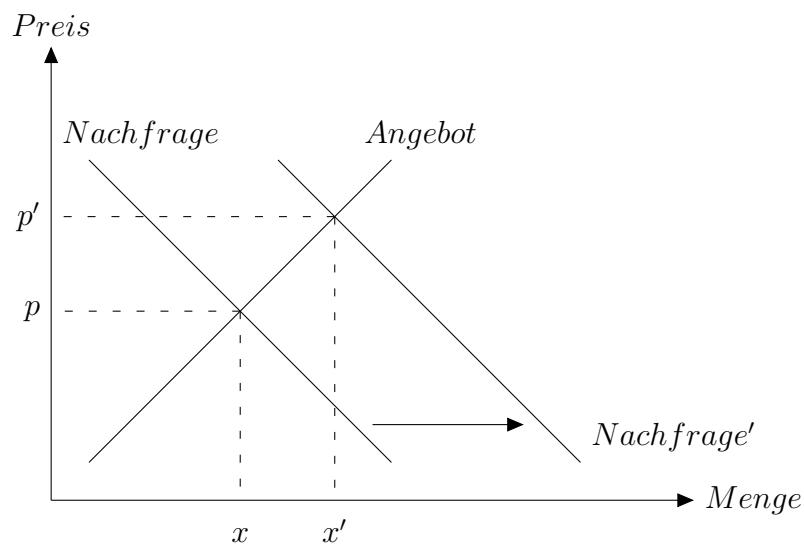


Abbildung 4: Die Beziehung zwischen Angebot und Nachfrage

Die Grundidee zur Veränderung der Bauteilpreise stammt aus der Betrachtung von Angebot und Nachfrage in der Makroökonomie. Wie in Abbildung 4 zu erkennen ist, führt die Erhöhung der Nachfrage bei gleich bleibendem Angebot zu einem neuen Marktgleichgewicht bei höheren Preisen. Ausgehend hiervon wurde der Algorithmus entwickelt, der durch Gleichung 3, Gleichung 4 und Gleichung 5 beschrieben wird. Die Bedeutung der Bezeichner ist Abbildung 5 zu entnehmen.

$$\bar{N} = \frac{1}{n} \cdot \sum_{i=1}^n x_i \cdot p_i \quad (3)$$

Mit Gleichung 3 wird für jede einzelne Bauteilgruppe (Standard- und Sonderbau-teile) zunächst die durchschnittliche Nachfrage in der zurückliegenden Periode berechnet. Die einzelnen Bauteilgruppen müssen hierbei getrennt betrachtet werden,

- \bar{N} : durchschnittliche Nachfrage
 x_i : von Bauteil i in zurückliegender Runde abgesetzte Menge
 p_i : Preis des Bauteils i in der zurückliegenden Runde
 ΔN_{rel} : relative Nachfrageabweichung eines Produkts von \bar{N}
 p_{neu} : neuer Preis des Bauteils
 p_{basis} : Grundpreis des Bauteils
 Δp_{max} : maximal erlaubte Abweichung vom Grundpreis

Abbildung 5: Legende zum Algorithmus

da die Sonderbauteile nur einen relativ geringen Anteil am Wert eines Raumschiffes haben, und die durchschnittliche Nachfrage zwischen den Gruppen somit nicht vergleichbar ist. Die Nachfrage wird am Umsatz gemessen, da beispielsweise von den teuren Triebwerken nur wenig, von den günstigen Rumpfbauteilen aber viele für ein Raumschiff benötigt werden.

$$\Delta N_{rel} = \frac{x_i \cdot p_i - \bar{N}}{\bar{N}} \quad (4)$$

In einem zweiten Schritt wird dann für jedes einzelne Bauteil der betrachteten Gruppe mit Gleichung 4 die relative Abweichung der Nachfrage für dieses Bauteil von der durchschnittlichen Nachfrage gebildet. Diese Abweichung kommt zu Stande, da sich die Mengen der benötigten Bauteile auf die verschiedenen Raumschifftypen in unterschiedlichen Verhältnissen verteilen. So führt eine überdurchschnittlich hohe Produktion von Millennium Falken beispielsweise zu einer überdurchschnittlich hohen Nachfrage nach Hitzeschilden, wie im Kapitel 2.4 ersichtlich wird.

$$p_{neu} = p_{basis} - \Delta p_{max} + \frac{2 \cdot \Delta p_{max}^2}{\Delta p_{max} + \frac{\Delta p_{max}}{(2 \cdot \Delta p_{max} \cdot \Delta N_{rel}) \left(\frac{2}{\Delta p_{max}} + 1 \right)}} \quad (5)$$

Zu guter Letzt wird dann, ausgehend von der errechneten relative Abweichung zur durchschnittlichen Nachfrage, der neue Preis des Bauteils anhand von Gleichung 5 berechnet. Lag eine überdurchschnittlich hohe Nachfrage vor, ist also die relative Abweichung positiv, so steigt der Bauteilpreis; entsprechendes gilt auch umgekehrt.

Wie sich der Preis genau verhält, ist abhängig von dem Grundpreis des Bauteils und der maximal erlaubten Abweichung von diesem Preis nach oben und unten. Entwickelt wurde diese kompliziert anmutende Funktionsvorschrift aus der Funktion des logistischen Wachstums, die dann dahingehend parametisiert wurde, dass der Schnittpunkt mit der x -Achse als Basispreis und die maximale Preisabweichung frei gewählt werden können. Hierdurch entsteht für den einfachen Sachverhalt eine sehr verschachtelte Funktion, die jedoch nur eine Abwandlung einer einfachen Funktion ist.

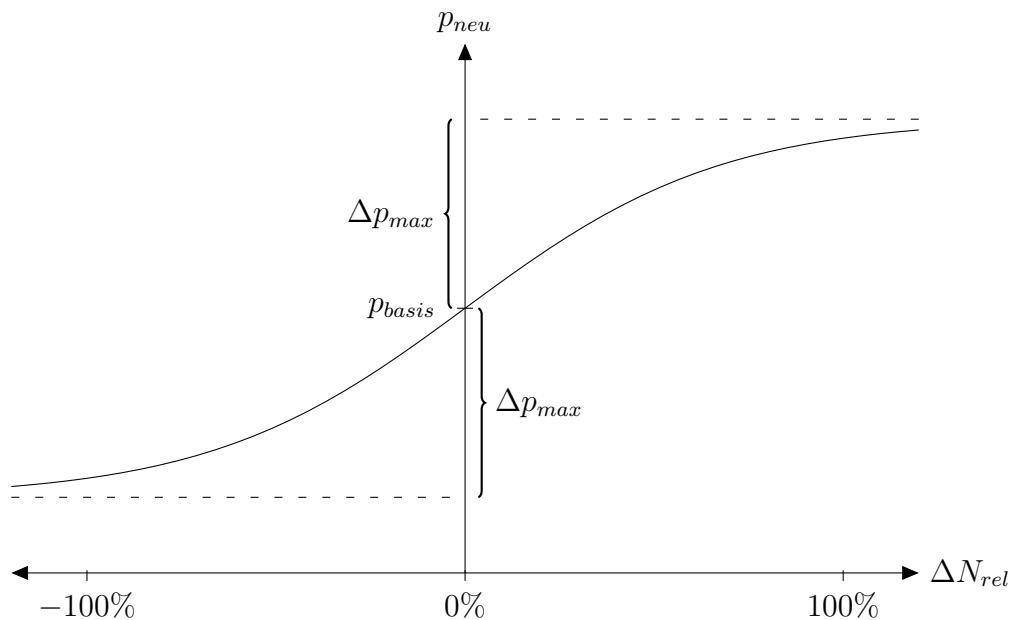


Abbildung 6: Die Berechnung des Bauteilpreises

Zur Verdeutlichung der Berechnung des letzten Schritts ist in Abbildung 6 der Funktionsgraph der Gleichung dargestellt. In diesem wird ersichtlich, wie sich der Preis eines Bauteils bei einer gegebenen Abweichung von der durchschnittlichen Nachfrage verhält. Zu erkennen sind ebenfalls die oben erwähnten Parameter, nämlich der Grundpreis p_{basis} und die maximale Abweichung Δp_{max} .

Bauteiltyp	Absatzmenge	Preis
Rumpfbauteil	130 Stück	100,00€
Hitzeschild	60 Stück	200,00€
Triebwerk	40 Stück	500,00€

Tabelle 8: Beispielwerte für die Berechnung neuer Bauteilpreise

Anhand eines Beispiels soll die Berechnung der Bauteilpreise abschließend betrachtet werden. Zu diesem Zweck nehme man die Werte in Tabelle 8 als die Werte

für die zurückliegende Spielrunde an. Folgt man nun der Berechnungsvorschrift, so muss zunächst mit Hilfe von Gleichung 3 die durchschnittliche Nachfrage berechnet werden.

$$\bar{N} = \frac{1}{3} \cdot (130 \cdot 100\text{C} + 60 \cdot 200\text{C} + 40 \cdot 500\text{C}) = 15000\text{C}$$

Im Durchschnitt wurde in der vergangenen Spielrunde also für jedes Standardbauteil insgesamt 15000C ausgegeben. Diese durchschnittliche Nachfrage, gemessen am Umsatz, wird dann verglichen mit der Nachfrage nach einem bestimmten Bauteil, beispielsweise dem Hitzeschild. Hierzu wird Gleichung 4 angewendet.

$$\Delta N_{rel} = \frac{60 \cdot 200\text{C} - 15000\text{C}}{15000\text{C}} = -0.2 \hat{=} -20\%$$

Die negative relative Abweichung sagt aus, dass im Vergleich zu den anderen Bauteilen, weniger Hitzeschilder eingekauft wurden, die Nachfrage nach diesen war also vergleichsweise gering. Wie sich dies auf den Preis für Hitzeschilder in der nächsten Runde auswirkt, ergibt sich aus Gleichung 5. Dazu müssen der Datenbasis in Kapitel 2.4 noch die Werte für den Basispreis $p_{basis} = 200,00\text{C}$ und die maximal erlaubte Abweichung von diesem $\Delta p_{max} = 80,00\text{C}$ entnommen werden.

$$p_{neu} = 200\text{C} - 80\text{C} + \frac{2 \cdot 80\text{C}^2}{80\text{C} + \frac{2 \cdot 80\text{C} \cdot (-0.2)}{\left(\frac{2}{80\text{C}} + 1\right)}} \approx 169.94\text{C}$$

Aufgrund der vergleichsweise niedrigen Nachfrage ergibt sich also in der nächsten Spielrunde ein niedrigerer Preis für Hitzeschilder, nämlich statt 200,00C nur noch 169,94C, was die Simulation des Zusammenhangs zwischen Angebot und Nachfrage in Abbildung 4 ausdrückt. Geschickte Spieler können so auf besonders niedrige Bauteilpreise reagieren, indem sie diese in Massen auf Vorrat einkaufen und so einerseits selbst Geld sparen, andererseits den Preis der Bauteile in der Folgerunde für die anderen Spieler in die Höhe treiben.

Ende: Fredrik

2.6.3 Absatzmengen

Beginn: Philipp

Nachdem alle Unternehmen die aktuelle Spielrunde eingecheckt haben, wird das Spiel simuliert. In der Simulationsphase wird unter anderem der Absatz berechnet, der jedem Unternehmen zusteht. Derer günstigste Anbieter soll eine höhere Anzahl an Raumschiffen absetzen können, als der mit dem höchsten Preis.

Um diese Vorgabe umzusetzen gibt es verschiedene Herangehensweisen. So könnte man einerseits die Angebote der Höhe nach anordnen und den einzelnen Rängen konstante Prozentsätze zuordnen. Beispielsweise würde man dem günstigsten Anbieter einen Absatz von 40% seiner produzierten Raumschiffe zusichern, dem Zweitgünstigsten 25%, dem Drittgrüngünstigsten 17,5%, dem Viertgünstigsten 12,5%, und so weiter. Diese Methode bringt jedoch fast nur Nachteile mit sich. So würde der günstigste Anbieter verglichen mit dem Zweitgünstigsten in jedem Fall 15% mehr Raumschiffe absetzen können, auch wenn sich der Preis nur geringfügig unterscheidet. Angenommen man würde dem teuersten Anbieter einen Absatz von 2% zusichern, dann wär es möglich den Verkaufspreis viel zu hoch anzusetzen und trotzdem einen Absatz und somit einen hohen Gewinn zu erzielen. Aus diesen Gründen haben wir uns dafür entschieden, den Absatz anhand einer Potenzfunktion, wie in Abbildung 7 abgebildet, darzustellen.

$$a_i = \frac{1}{p_i^3} \quad (6)$$

Da die einzelnen Raumschifftypen verschiedene Nachfrager ansprechen und das Unternehmen verschieden hohe Preise für die verschiedenen Raumschifftypen festlegen kann, wird dies für jeden der 3 Raumschifftypen abgebildet.

Diese Potenzfunktion zeigt welche Unternehmen wie viele Raumschiffe zu welchem Preis abgesetzen können. Der asymptotische Verlauf der Kurve sorgt dafür, dass Anbietern mit niedrigen Preisen mehr Absatz zusteht, als Anbietern mit hohen Preisen.

Um für jeden Anbieter den prozentualen Absatz zu bestimmen, wird der individuelle Absatz relativ zum Gesamtabsatz berechnet:

$$a_{rel.i} = \frac{a_i}{\sum a_i} \quad (7)$$

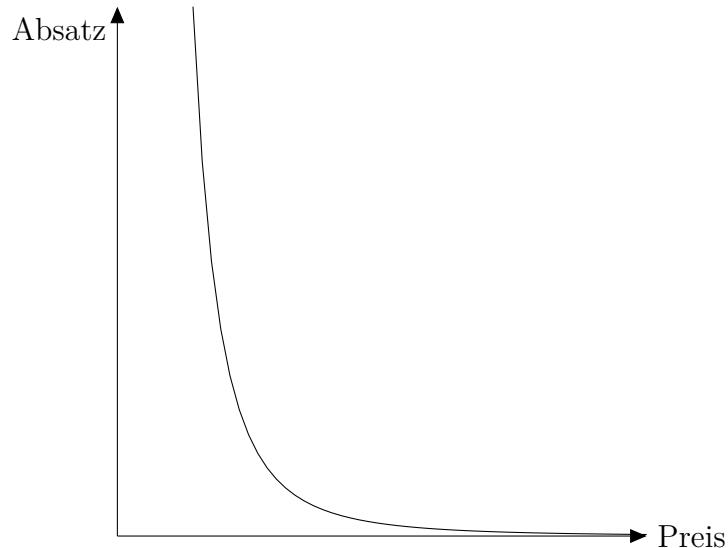


Abbildung 7: Preis-Absatz-Funktion

Dadurch, dass man durch die Berechnung des relativen Absatzes die verschiedenen Preise der Unternehmen mit einbezieht, hat diese Variante den Vorteil, dass geringe Preisunterschiede auch nur geringe Abweichungen beim Absatz zulassen. Außerdem hat ein Unternehmen, das einen unverhältnismäßig hohen Preis festgelegt hat, keine Chance Raumschiffe abzusetzen. Daher spiegelt diese Variante einen sehr realistischen Absatzmarkt wider.

Die Nachfrage bestimmt sich wie folgt:

$$N_{neu} = N_{alt} \cdot \left(1 - \left(\frac{p_{guenstigster}}{Kosten \cdot 3,5}\right)^4\right) \quad (8)$$

Über diese Formel wird unter Berücksichtigung der Bauteilkosten die Nachfrage bestimmt. Auch diese Funktion wurde implementiert um eine möglichst realistische Situation herzustellen. Würden in der Realität die Preise aller Anbieter eine bestimmte Grenze überschreiten, so wäre niemand mehr bereit ein Raumschiff zu kaufen. Diese Grenze wurde im Planspiel auf das 3,5-fache der Kosten gesetzt. Das heißt, dass ein Raumschiff, das mit Kosten von 6000€ produziert wurde, ab einem Preis von 21000€ keine Nachfrage mehr erzielt.

Mit Hilfe dieser Funktionen wurde der Absatzalgorithmus implementiert. Er kümmert sich darum, dass der Spieler mit dem niedrigsten Preis die Anzahl an Raumschiffen bis zu seinem errechneten relativen Absatz verkaufen darf. Nach aufsteigen-

dem Preis wird auf diese Weise weiter vorgegangen. Hat ein Unternehmen jedoch nicht ausreichend Raumschiffe produziert, kann es also dem ihm eigentlich zustehenden Absatz nicht nachkommen, so geht die übrige nicht befriedigte Nachfrage auf den nächst günstigeren Anbieter über.

Anhand eines Beispiels soll die Funktionsweise des Absatzalgorithmus verdeutlicht werden:

Es gibt 3 Unternehmen A, B und C, die ihre Raumschiffe auf dem Raumschiffmarkt anbieten. In Tabelle 9 sieht man die aktuellen Preise der einzelnen Unternehmen.

Unternehmen	Raumschifftyp	Preis
A	X-Wing	12000C
	Corellian Corvette	24000C
	Millenium Falke	36000C
B	X-Wing	10000C
	Corellian Corvette	22000C
	Millenium Falke	47000C
C	X-Wing	23000C
	Corellian Corvette	38000C
	Millenium Falke	33000C

Tabelle 9: Beispiel zur Berechnung des Absatzes - 1

Im Folgenden wird zunächst der Absatz anhand Gleichung 6 berechnet:

Unternehmen	Raumschifftyp	Absatz
A	X-Wing	$\frac{1}{12000^3} = 57,8 \cdot 10^{-14}$
	Corellian Corvette	$\frac{1}{24000^3} = 7,23 \cdot 10^{-14}$
	Millenium Falke	$\frac{1}{36000^3} = 2,14 \cdot 10^{-14}$
B	X-Wing	$\frac{1}{10000^3} = 75,1 \cdot 10^{-14}$
	Corellian Corvette	$\frac{1}{22000^3} = 9,39 \cdot 10^{-14}$
	Millenium Falke	$\frac{1}{47000^3} = 0,963 \cdot 10^{-14}$
C	X-Wing	$\frac{1}{23000^3} = 8,22 \cdot 10^{-14}$
	Corellian Corvette	$\frac{1}{38000^3} = 1,82 \cdot 10^{-14}$
	Millenium Falke	$\frac{1}{33000^3} = 2,78 \cdot 10^{-14}$

Tabelle 10: Beispiel zur Berechnung des Absatzes - 2

Hieraus ergeben sich folgende Summen:

X-Wing	$141, 12 \cdot 10^{-14}$
Corellian Corvette	$18, 44 \cdot 10^{-14}$
Millenium Falke	$5, 88 \cdot 10^{-14}$

Tabelle 11: Beispiel zur Berechnung des Absatzes - 3

Anhand der erhaltenen Werte kann man nun den relativen Absatz für jedes Unternehmen und jeden Raumschiffotyp berechnen.

Raumschifftyp	Unternehmen	rel. Absatz
X-Wing	A	40,94%
	B	53,22%
	C	5,82%
Corellian Corvette	A	39,21%
	B	50,92%
	C	9,87%
Millenium Falke	A	36,39%
	B	16,38%
	C	47,28%

Tabelle 12: Beispiel zur Berechnung des Absatzes - 4

Die Werte aus Tabelle 12 geben an, wie viel Prozent der nachgefragten Raumschiffe die einzelnen Unternehmen absetzen dürfen. Vor allem bei den Raumschiffen X-Wing und Corellian Corvette des Unternehmens C wird deutlich, dass ein zu hoher Preis kaum Nachfrage schafft.

2.6.4 Fehlerkosten

Da es, wie in den vorherigen Kapitel genauer beschrieben, mehrere Personaltypen gibt, die sich in ihrer Qualität unterscheiden und je nach Qualitätsstufe unterschiedliche Kosten bei der Produktion verursachen, musste ein Algorithmus definiert werden, der diese Mehrkosten in jeder Spielrunde ermittelt.

Zur Ermittlung der Mehrkosten werden Reparaturkosten, die für die Reparatur eines fehlerhaften Raumschiffs anfallen, herangezogen. Das heißt also, wenn ein Unternehmen ein fehlerhaftes Raumschiff produziert, wird das Konto des Unternehmens mit Mehrkosten, die für die Reparatur oder Behebung des Fehlers anfallen, belastet. Diese Kosten belaufen sich auf 50% der Produktionskosten.

Andere Überlegungen waren beispielsweise der Ausschuss ganzer Raumschiffe oder der Ausfall einzelner Bauteile. Vergleicht man den Ausschuss bei der Raumschiffproduktion beispielsweise mit der Automobilproduktion, so stellt man fest, dass es beim Zusammenbau einzelner Fertigbauteile nicht zum Ausschuss eines kompletten Endprodukts kommen kann. Daher erschien uns dieser Ansatz zum einen äußerst unrealistisch und zum anderen würde der Ausschuss eines ganzen Raumschiffs aufgrund des hohen Produktionspreises einen zu hohen wirtschaftlichen Nachteil für das jeweilige Unternehmen darstellen. Um das Planspiel nicht zu sehr von Zufallsergebnissen abhängig zu machen wurde daher auf diese Variante verzichtet.

Auch wenn der Ausfall einzelner Bauteile sehr realistisch erscheint und auch finanziell für die Unternehmen zu verkraften wäre, würden sich bei dieser Variante andere Probleme ergeben. So ist es einem Unternehmen möglich Fertigbauteile einzukaufen um daraus Raumschiffe zu produzieren. Fällt nun ein Teil aus, sind möglicherweise nicht genügend Bauteile auf Lager, um das Raumschiff trotz des Ausfalls eines Bauteils produzieren zu können. In der Realität hätte das Unternehmen die Chance, Teile kurzfristig dazuzukaufen, wohingegen beim Planspiel das Unternehmen nach dem Einchecken der Spielrunde keine Möglichkeit hat etwas am Einkauf oder der Produktion für die jeweilige Runde zu ändern. Aus diesem Grund wurde auch diese Variante als mangelhaft eingestuft und man hat sich auf die am besten ins Planspiel eingliederbare und im nachfolgenden Abschnitt beschriebene Variante der Reparaturkosten entschieden.

Aufgrund der Tatsache, dass sich das Personal eines Unternehmens aus allen drei Personaltypen zusammensetzen kann, ist zunächst die durchschnittliche Personalqualität, des im Unternehmen eingesetzten Personals zu bestimmen:

$$\bar{q} = \frac{1}{n} \cdot \sum_{i=1}^n q_i \cdot n_i \quad (9)$$

Mit dieser Formel wird die Durchschnittsqualität des Personals als gewichtetes arithmetisches Mittel berechnet. Die daraus erhaltenen Werte liegen in jedem Fall im Bereich 0,69 - 0,99.

Mit Hilfe dieser Information ist man nun in der Lage, die durch das Personal verursachte fehlerhaft produzierte Menge zu ermitteln. Zur Berechnung wird die in Java vorgefertigte *Math.random()*-Funktion der Klasse random verwendet. Diese Funktion generiert für jedes in Produktion gegebene Raumschiff eine Zufallszahl zwischen 0,0 und 1,0. Sollte die erzeugte Zufallszahl den Wert der Durchschnittsqualität des Personals überbieten, so wird dieses Raumschiff als fehlerhaft eingestuft. Ist die

Qualität des Personals also höher, so erhöht sich die Wahrscheinlichkeit fehlerfreie Raumschiffe zu produzieren. Wiederholt man diesen Vorgang für jedes produzierte Raumschiff, so lassen sich auf diese Weise die anfallenden Mehrkosten errechnen.

An einem vereinfachten Beispiel veranschaulicht, könnte dies wie folgt aussehen:

Das Unternehmen “Weedman Ships” beschäftigt 13 R2D2, 18 Kampfdroiden und 9 Droiden des Typs Droideka. Mit diesem Personal will das Unternehmen einen X-Wing , zwei Corellian Corvettes und ein Raumschiff des Typs Millenium Falke produzieren.

Mit diesen Angaben lässt sich zunächst die durchschnittliche Personalqualität des Unternehmens ermitteln:

$$\frac{13 \cdot 0,69 + 18 \cdot 0,84 + 9 \cdot 0,99}{40} \approx 0,78$$

Wie in Tabelle 13 dargestellt, werden nun Zufallszahlen generiert um festzustellen, ob bei der Produktion Fehler auftraten.

Raumschiff	Zufallszahl	Fehlerhaft?
X-Wing	0,3	Nein
Corellian Corvette	0,9	Ja
Corellian Corvette	0,7	Nein
Millenium Falke	0,1	Nein

Tabelle 13: Beispiel zur Berechnung der Fehlerkosten

Beim ersten Raumschiff erzeugt die Funktion Math.random() die Zufallszahl 0,3. Da diese die Personalqualität von 0,78 nicht überbietet, liegt kein Fehler vor. Dagegen weist eine der Corellian Corvettes einen Fehler auf. Die zweite Corellian Corvette und der Millenium Falke wurden wieder fehlerfrei produziert. Aufgrund der nur mittelmäßigen Qualität des Personals liegen bei einem von vier Raumschiffen Fehler vor. Hätte das Unternehmen verstärkt in ihr Personal investiert und in Form von Aufrüstung oder Neueinstellungen die Qualität verbessert, so hätte dieser Fehler möglicherweise vermieden werden können.

Um in diesem Beispiel den Fehler bei der Corellian Corvette zu beheben, würden bei Produktionskosten von 12000€ Reparaturkosten von 6000€ anfallen.

2.6.5 Bewertung

Eine Bewertung der einzelnen Spieler findet am Ende des Planspiels statt. Sie errechnet sich zu 70% aus dem ROI der Unternehmen und zu 30% aus dem Marktanteil. Der ROI war für uns hierbei die wichtigste Kennzahl um die erfolgreiche Geschäftsführung und damit den Erfolg der Unternehmen zu messen. Da es für ein Unternehmen allerdings auch wichtig ist, sich am Markt zu etablieren und ein gutes Image aufzubauen, haben wir uns dafür entschieden den Marktanteil mit in die Endauswertung einfließen zu lassen.

Der ROI ist eine Gegenüberstellung von Gewinn und eingesetztem Kapital:

$$ROI = \frac{\text{Gewinn}}{K_{eingesetzt}} \quad (10)$$

Der Gewinn berechnet sich wie folgt:

$$\text{Gewinn} = K_{End} - K_{Start} + FE_{Lager} \cdot 0,75 + Bauteile_{Lager} \cdot 0,5 \quad (11)$$

Um das Ergebnis nicht zu verzerren, werden Raumschiffe und Bauteile im Lager mit berücksichtigt und dem Gewinn hinzugaddiert. Produzierte Raumschiffe, die sich im Lager befinden, werden zu 75% und gekaufte Bauteile zu 50% ihres Wertes mit eingerechnet. Das eingesetzte Kapitel ist im Planspiel in Form des Startkapitals fest vorgegeben und für jeden Spieler gleich.

Jedoch ist zu Berücksichtigen, dass nur mit positiven ROI-Werten gerechnet werden darf. Aus diesem Grund ist es notwendig beim Auftreten einer oder mehrerer negativen ROI-Werten alle ROI's in soweit zu erhöhen, dass sich der niedrigste ROI bei 0 befindet. Auf diese Weise lässt sich nun der ROI anteilig berechnen:

$$ROI_{rel.} = \frac{ROI_{korrigiert}}{\sum ROI_{korrigiert}} \quad (12)$$

Beim Marktanteil handelt es sich um eine Gegenüberstellung des Umsatzes eines Unternehmens und der Umsätze aller Unternehmen:

$$\text{Marktanteil} = \frac{Umsatz}{\sum Umsatz} \quad (13)$$

Mit Hilfe des anteiligen ROI's und des Marktanteils können nun Punkte berechnet werden:

$$\text{Punkte} = \text{ROI}_{\text{rel.}} \cdot 70 + \text{Marktanteil} \cdot 30 \quad (14)$$

Anhand dieser Punkte wird eine Rangfolge der Unternehmen aufgestellt. Das Unternehmen mit den meisten Punkten gewinnt das Spiel.

Für die Bewertung hätte es noch andere Möglichkeiten gegeben. So wurde in der Gruppe zu Beginn des Projekts diskutiert, ob man die Endauswertung nicht anhand von Medaillen umsetzt, die die Spieler in jeder Spielrunde "gewinnen" können. Auf diese Weise hätte es beispielsweise eine Medaille für den größten Marktanteil oder den besten ROI der Runde gegeben. Mit dieser Variante hätte man die Spieler anhand von Zwischenbewertungen mit zusätzlichen Informationen zur momentanen Unternehmenslage versorgen können.

Ende: Philipp

3 Fachkonzept

3.1 Einleitung

Beginn: Julia

In diesem Kapitel wird die technische Seite des Planspiels betrachtet. Anfangs wird auf die Erstellten UML-Diagramme eingangen. Hierzu zählt das UseCase Diagramm und das Aktivitätsdiagramm. Anschließend werden Entwurfsentscheidungen, wie beispielsweise die unterschiedlichen Typen von Bauteilen und Raumschiffen, die verschiedenen Märkte, das Unternehmen, die Spielrunden und die Transaktionen näher beschrieben. Im Anschluss daran folgt ein Abschnitt zur Implementierung. Dabei wird auch die Entscheidungsfindung der Unternehmen sowie der Spielablauf erläutert.

3.2 UseCase Diagramm

3.2.1 Einleitung

Um einen Überblick über die Funktionen zu bekommen, die das Planspiel abdecken sollen, wurde das folgende UseCase Diagramm auf Abbildung 8 auf S. 35 erstellt. In diesem Diagramm gibt es zwei Akteure, der Spielleiter und der Spieler. Die einzelnen Funktionen der beiden Akteure werden in den beiden folgenden Kapiteln näher erläutert.

Ende: Julia

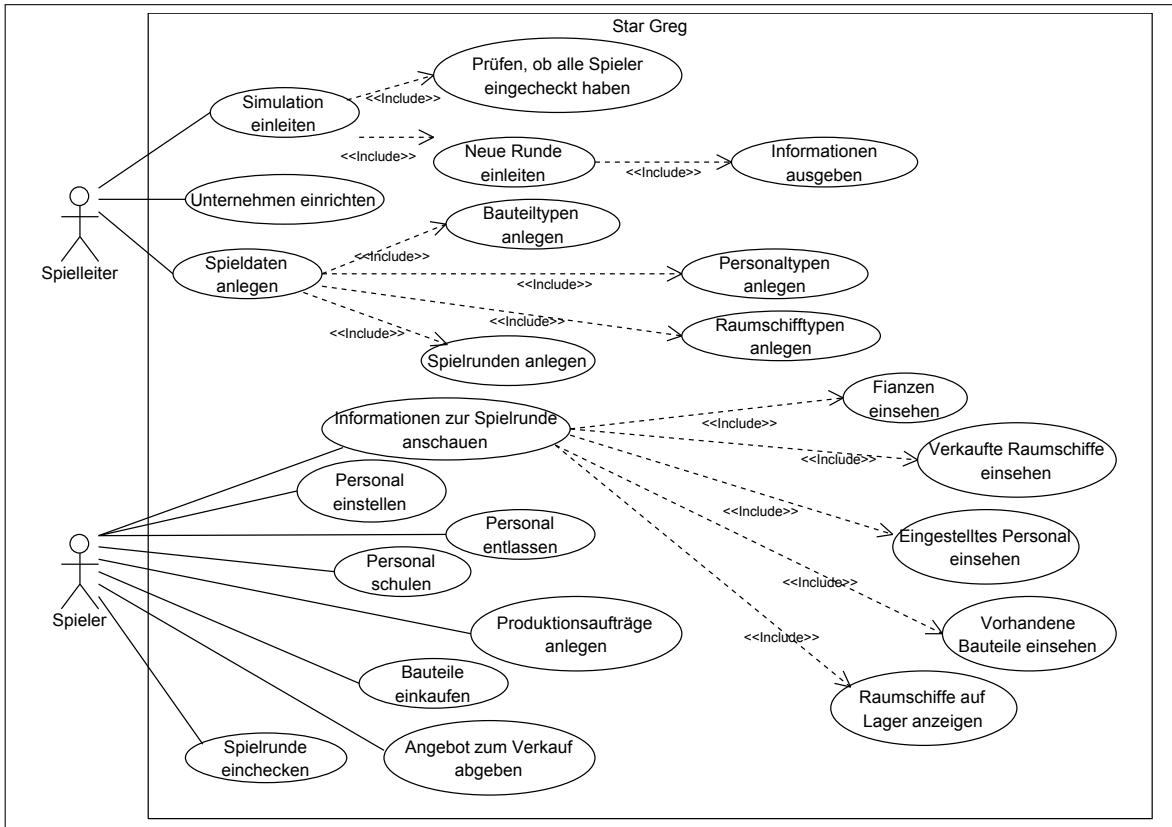


Abbildung 8: UseCase Diagramm

3.2.2 Spielleiter

Beginn: Marcel

Zentraler Akteur neben dem Spieler ist in dem UseCase Diagramm der Spielleiter. Er ist im Hintergrund tätig und leitet das gesamte Unternehmensplanspiel. Zur aktuellen Implementierungsphase ist es vorgesehen, dass der Spielleiter manuell von einer Person bedient wird. Es ist jedoch vorgesehen, dass die Funktionen des Spielleiters zu einem späteren Zeitpunkt vollständig durch den Computer übernommen werden, sodass für die Aufgaben des Spielleiters keine weitere Person benötigt wird.

Zu Beginn des Unternehmensplanspiels richtet der Spielleiter die Unternehmen ein. Er bekommt dazu von allen Mitspielern den Unternehmensnamen mitgeteilt und richtet auf Basis dieser Daten pro Mitspieler ein Unternehmen ein.

Neben dieser Aufgabe muss der Spielleiter alle Spieldaten anlegen, die für das Un-

ternehmensplanspiel notwendig sind. Dazu müssen von ihm alle Bauteiltypen, Personaltypen und Raumschifftypen angelegt werden. Nach dem Anlegen dieser Daten legt der Spielleiter die Spielrunden an.

Eine weitere zentrale Aufgabe des Spielleiters ist es die Simulation der Spielrunden einzuleiten. Spielrunden können aber erst eingeleitet werden, wenn der Spielleiter überprüft hat, dass alle Spieler die Spielrunden eingecheckt haben. Jeder Spieler muss dazu aller relevanten Tätigkeiten pro Spielrunde ausführen und diese daraufhin bestätigen. Ebenso wird eine neue Spielrunde eingeleitet. Mit dem Einleiten einer neuen Spielrunde gibt der Spielleiter ebenso Informationen zur neuen Spielrunde aus.

Die Unternehmen zu bewerten, liegt ebenfalls in der Hand des Spielleiters. Dies geschieht anhand einer festgelegten Formel, die zu einem späteren Zeitpunkt dieser Seminararbeit erläutert wird.

Ende: Marcel

3.2.3 Spieler

Beginn: Julia

Der zweite Akteur in diesem UseCase Diagramm stellt der Spieler dar. Er steuert ein komplettes Unternehmen und übernimmt somit auch alle im Unternehmen anfallenden Aufgaben. Um das Spiel zu starten übermittelt der Spieler den selbst ausgewählten Unternehmensnamen an den Spielleiter welcher wie oben beschrieben das Spiel einrichtet.

Ist dies geschehen, so hat der Spieler die Möglichkeit jegliche Informationen über das eigene Unternehmen und alle vorhandenen Abteilungen abrufen. Im Planspiel sind folgende Abteilungen integriert: Finanzwesen, Einkauf, Produktion, Verkauf und Personalwesen. Somit kann der Spieler die aktuelle Finanzlage des Unternehmens, die verkauften Raumschiffe, die gekauften Bauteile, die eingestellten Mitarbeiter sowie die eingelagerten Raumschiffe einsehen. Dadurch ist ein optimaler Überblick über das gesamte Unternehmen gewährleistet.

Dem Spieler ist es darauf hin ermöglicht, weiteres Personal einzustellen, falls er dies aus Kapazitäts- oder Qualitätsgründen benötigt. Bereits eingestelltes Personal kann weitergebildet werden wodurch die Fähigkeiten der einzelnen Mitarbeiter gesteigert

werden. Sind im Unternehmen zu viele Mitarbeiter angestellt, so hat der Spieler ebenfalls die Möglichkeit, Personal zu entlassen.

Im Einkaufsbereich kann der Spieler verschiedene Bauteile einkaufen, welche zur Produktion von Raumschiffen benötigt werden. Als nächster Schritt können Produktionsaufträge angelegt werden.

Anschließend kann der Spieler Verkaufsangebote für die produzierten Raumschiffe abgeben. Als letzten Schritt wird die Spielrunde vom Spieler eingechockt und somit beendet.

Details zu den einzelnen Vorgänge, wie zum Beispiel das Einkaufen von Bauteilen, die Personalverwaltung oder die Produktion, werden im Laufe der Seminararbeit genauer erläutert.

Ende: Julia

3.3 Aktivitätsdiagramm

Beginn: Fredrik, Julia

3.3.1 Übersicht

Das Aktivitätsdiagramm auf Abbildung 9 auf S. 38 soll den Spielablauf aus Sicht des Spielers verdeutlichen.

Um das Spiel zu starten muss der Spieler einen Namen für sein Unternehmen festlegen. Dies stellt die erste Aktion dar. Danach analysiert der Spieler die ihm zur Verfügung stehenden Informationen. Ist dies abgeschlossen, so gelangt er zu einem Entscheidungsknoten. Hierbei kann sich der Spieler zwischen folgenden Aktivitäten entscheiden: Personal verwalten, Einkäufe tätigen, Produktionsaufträge anlegen oder Verkaufsangebot abgeben. Diese einzelnen Vorgänge werden im Laufe dieses Kapitels genauer erläutert. Er kann sich aber auch dazu entscheiden, keine Transaktion zu tätigen.

Diese verschiedenen Aktivitäten werden in einem Entscheidungsknoten zusammengeführt. Hat der Spieler weiteren Informationsbedarf, so gelangt er zur Aktivität

“Auswirkungen analysieren” und kann sich die Veränderungen anschauen, die seine Transaktion mit sich geführt hat. Besteht kein Informationsbedarf, kann er diese Aktivität überspringen.

Möchte der Spieler nun weitere Transaktionen tätigen, so kann er wieder zum Entscheidungsknoten nach oben springen und hat wieder die Wahl zwischen Personal verwalten, Einkäufe tätigen, Produktionsaufträge anlegen oder ein Verkaufsangebot abgeben. Entcheidet sich der Spieler gegen eine weitere Transaktion, so folgt die Aktivität “Runde einchecken”. Anschließend folgt wieder ein Entscheidungsknoten. Existiert noch eine weitere Spielrunde, so gelangt der Spieler wieder zur Aktivität “Informationen analysieren”. War dies die letzte Spielrunde, so wird dem Spieler die Endbewertung angezeigt. Danach ist das Spiel beendet.

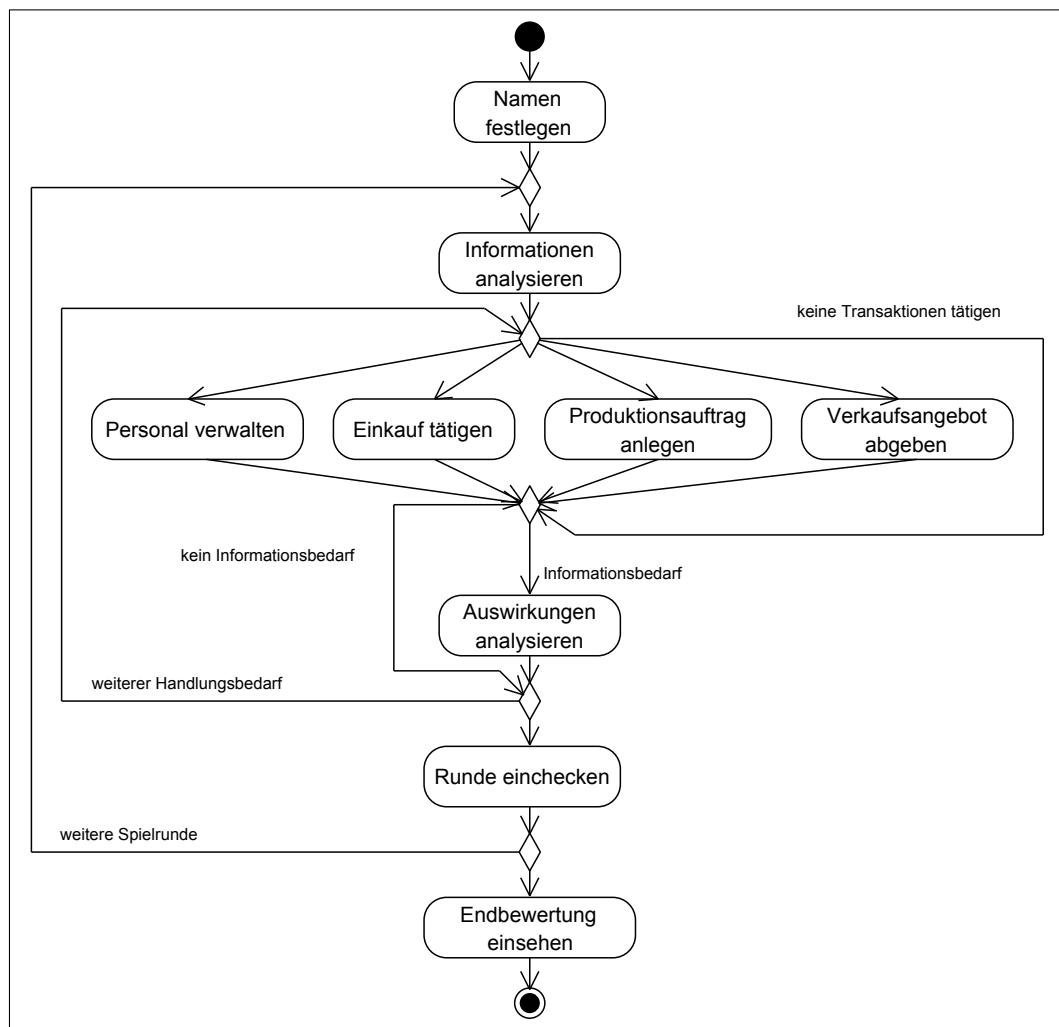


Abbildung 9: Aktivitätsdiagramm

3.3.2 Personal verwalten

Zur genaueren Betrachtung der Aktion “Personal verwalten” dient folgendes Aktivitätsdiagramm auf Abbildung 10 auf S. 39. Entscheidet sich der Spieler für diese Aktion, so hat er die Möglichkeit Personal einzustellen, aufzurüsten oder zu entlassen.

Um neues Personal einzustellen, muss der Spieler zuerst den Personaltypr auswählen, welchen er einstellen möchte. Danach folgt die Aktivität “Anzahl festlegen”. Anschließend trifft der Spieler auf einen Entscheidungsknoten. Ist genügend Geld vorhanden, muss der Spieler die Auswahl bestätigen. Reicht das verfügbare Geld jedoch nicht aus, gelangt der Spieler wieder zur Aktivität “einzustellender Typ auswählen” und durchläuft den Prozess noch ein mal.

Möchte der Spieler sein vorhandenes Personal aufrüsten, muss er zunächst den Personaltypr auswählen. Anschließend gelangt er zur Aktivität “Anzahl festlegen”. Ähnlich wie beim Einstellen von neuem Personal wird auch nun geprüft, ob genügend Geld vorhanden ist. Ist dies der Fall, wird die Auswahl bestätigt. Fehlt Geld, befindet sich der Spieler wieder bei der Aktivität “aufzurüstender Typ auswählen”.

Zum Entlassen von Personal ist ebenfalls der Personaltypr und die Anzahl festzulegen. Anschließend wird die Auswahl bestätigt und die Transaktion ist abgeschlossen.

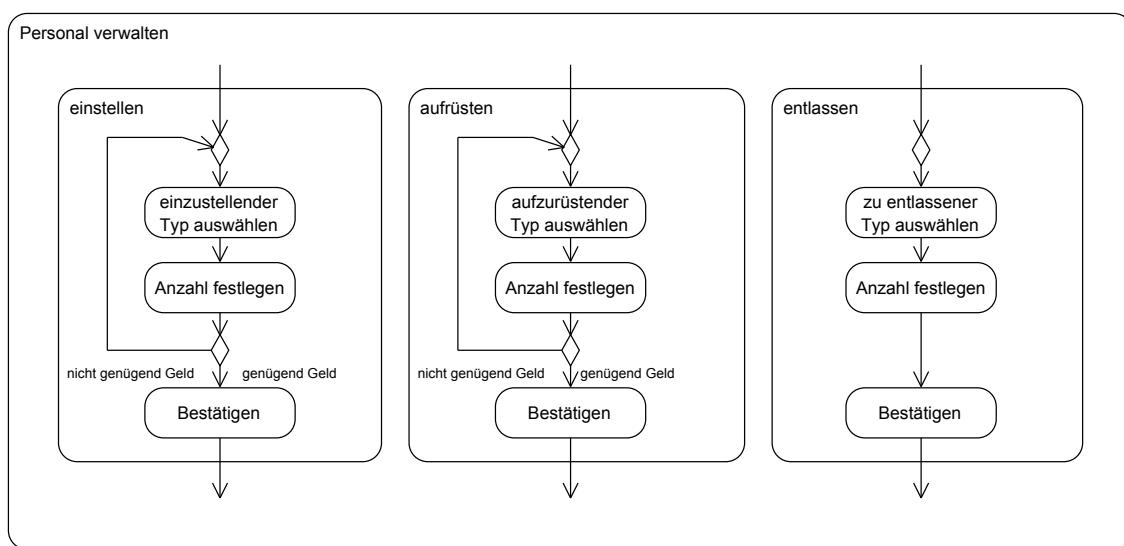


Abbildung 10: Aktivitätsdiagramm zur Personalverwaltung

3.3.3 Bauteile einkaufen

Entscheidet sich der Spieler für das Einkaufen von Bauteilen, muss er zunächst den Bauteiltyp wählen und anschließend die einzukaufende Anzahl festlegen. Im Anschluss daran wird wieder die Liquidität des Spielers geprüft. Ist das Geld ausreichend, muss die Auswahl bestätigt werden. Fällt die Prüfung negativ aus, gelangt der Spieler wieder zur Aktivität “Bauteiltyp auswählen”. Dies wird in Abbildung 11 auf S. 40 verdeutlicht:

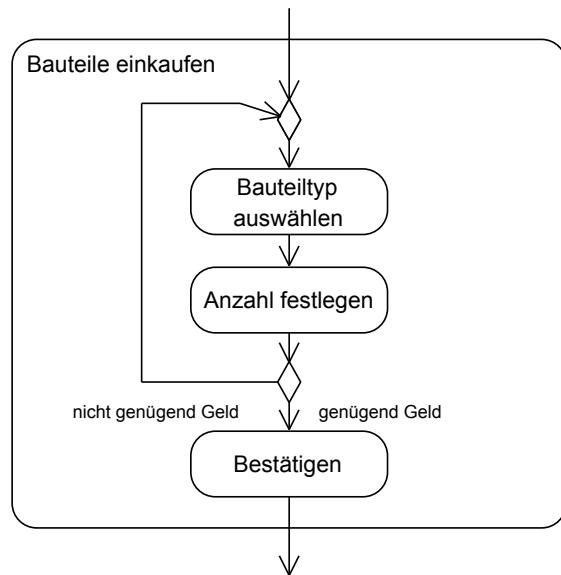


Abbildung 11: Aktivitätsdiagramm zum Bauteileinkauf

3.3.4 Produktionsauftrag anlegen

Beim Anlegen eines Produktionsauftrages, wie in Abbildung 12 auf S. 41 dargestellt, ist zuerst das zu produzierende Raumschiff auszuwählen. Anschließend legt der Spieler die Anzahl fest. Nun muss geprüft werden, ob zum einen genügend Bauteile für die Produktion vorhanden sind und zum anderen ob das eingestellte Personal die Anzahl an Raumschiffen in einer Periode produzieren kann. Fällt die Prüfung positiv aus, so muss der Spieler seine Auswahl bestätigen. Sind die Kapazitäten jedoch nicht ausreichend, befindet sich der Spieler wieder bei der Aktivität “Raumschifftyp auswählen”.

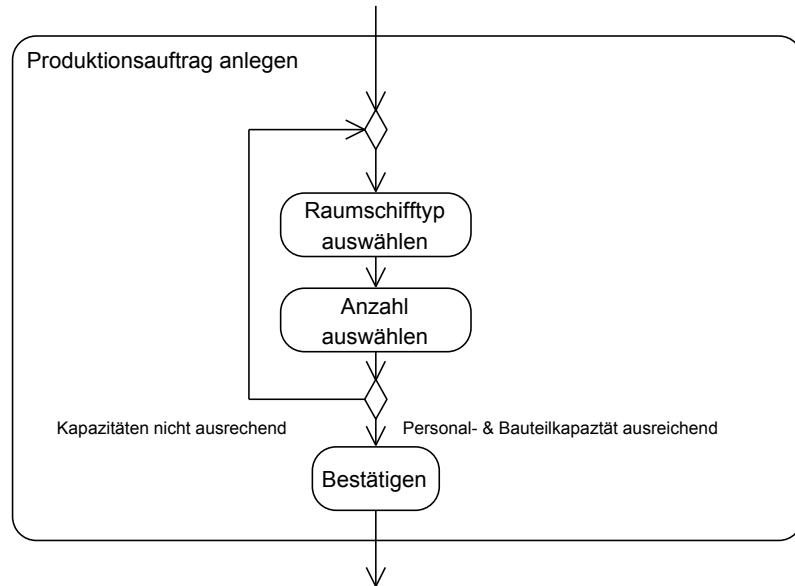


Abbildung 12: Aktivitätsdiagramm zur Produktion

3.3.5 Verkaufsangebot abgeben

Die letzte Transaktion kann im Bereich Verkauf getätigert werden. Dies ist auf Abbildung 13 auf S. 42 zu sehen. Zuerst wird der zu verkaufende Raumschifftyp ausgewählt. Danach entscheidet sich der Spieler für einen Preis, zu dem er die Raumschiffe verkaufen möchte. War bereits zuvor ein Angebot vorhanden, so wird dieses hiermit zurück genommen und danach bestätigt. War dies das erste Angebot, so gelangt der Spieler direkt zur Aktivität "Bestätigen".

Ende: Fredrik, Julia

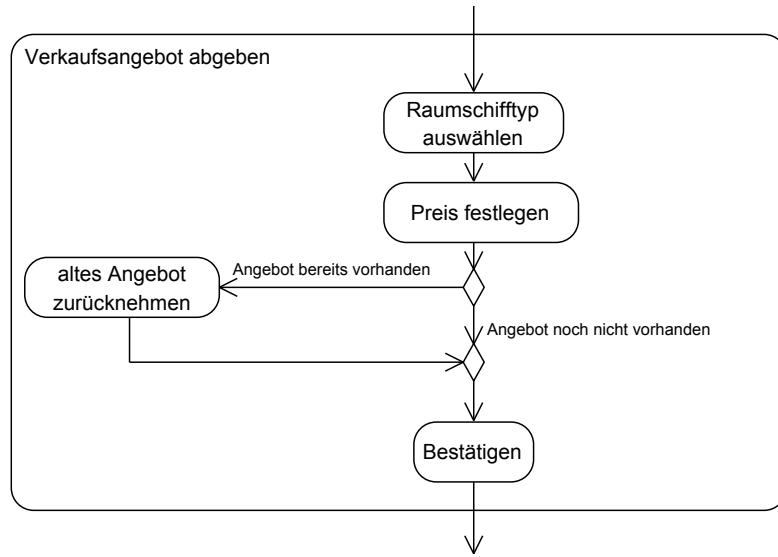


Abbildung 13: Aktivitätsdiagramm zum Verkauf

3.4 Entwurfsentscheidungen

Beginn: Fredrik

3.4.1 Einleitung

Zur Erstellung eines Unternehmensplanspiels bedarf es - wie in jedem Prozess der Software-Erstellung - nach einer anfänglichen Analyse der benötigten Funktionalität, deren Ergebnisse in Kapitel 3.2 anhand eines Use Case Diagramms verkürzt dargestellt wurden, Entscheidungen zum grundlegenden Entwurf der Software. Diese Phase grenzt sich klar ab von den Entscheidungen zur Implementierung des Projekts, die in Kapitel 3.5 erläutert werden, da sie sich auf einem höheren Abstraktionsniveau bewegt. Es geht hierbei nicht darum, die Lösung auf eine bestimmte Umsetzung hin zu optimieren, sondern die erkannten Anforderungen weitestgehend zu ermöglichen. Diese Entscheidungen sollen strukturiert werden mit Hilfe der Notation des UML Klassendiagramms.

Konkret bedeutet der höhere Abstraktionsgrad, dass Eigenheiten wie die System-Architektur (z.B. Client-Server-Architektur), die Persistierung von Daten oder die Benutzeroberfläche an dieser Stelle weder behandelt noch berücksichtigt werden sollen. Im Vordergrund stehen stattdessen die Spielwelt und alle Elemente, die den Ablauf des Spiels ermöglichen. Um eine gewisse Übersichtlichkeit zu gewährleisten,

wurde auf die Aufführung von *get()*- und *set()*-Methoden verzichtet, solange diese als trivial einzustufen sind. Ebenso gilt für alle Assoziationen die Annahme, dass diese mit der Kardinalität 1:1 zu verstehen sind, außer es sind abweichende Kardinalitäten eingetragen.

Aufgrund der Größe des Klassendiagramms ist dies erst in Kapitel A.1 auf S. 89 vollständig dargestellt. Im Folgenden sollen einige Bereiche aus diesem Modell getrennt voneinander betrachtet werden. Nach einer Vorstellung des Modellkerns werden zunächst die sogenannten Typen, die Beschreibungen von konkreten Gegenständen der Spielwelt wie z.B. den Raumschifftypen darstellen, betrachtet. Daran schließt sich eine Erläuterung der Märkte an, gefolgt von einer Betrachtung der Abteilungen. Zuletzt werden die zur Erfassung der Vorgänge in der Spielwelt und deren Speicherung dienenden Transaktions-Klassen erörtert.

3.4.2 Die Kernelemente

In Abbildung 14 sind die Kernelemente des Entwurfsmodells zu sehen. Hieran lässt sich gut erkennen, dass *Spiel* die zentrale Klasse darstellt. Sie steht in Verbindung mit der *SpielWelt*, die für die eigentliche Simulation der wirtschaftlichen Vorgänge verantwortlich ist und daher auch mit den *Märkten* und *Unternehmen* verbunden ist. Ein Unternehmen besteht dabei aus mehreren *Abteilungen*. Unabhängig von der Ebene der SpielWelt muss eine Speicherung der Vorgänge über das gesamte Spiel hinweg erfolgen, realisiert durch die Klasse *SpielRunde*. Diese dient zu Beginn des Spiels zur Aufnahme der Daten für den Spielverlauf wie Nachfrage-Entwicklung und Benachrichtigungen für den Spieler; im Verlauf des Spiels nimmt sie die Daten der abgeschlossenen Periode in sich auf um diese für die abschließende Bewertung verfügbar zu machen. Möglich wird die Speicherung aller erdenklichen Vorgänge durch die Klasse *Transaktion*, die eine Handlung eines Spielers repräsentiert. Bisher ohne Verbindung zu den anderen Klassen steht *Typ*, welches die Oberklasse für alle speziellen Typen (*BauteilTyp* etc.) darstellt. Die Verbindungen ergeben sich erst bei näherer Betrachtung der einzelnen Sektionen.

An dieser Stelle soll hervorgehoben werden, dass die Ablauflogik des Spiels von der Methode *simuliere()* in der Klasse *Spiel* ausgeht, die einerseits die Simulation der SpielWelt vorantreibt, in der dann Absatzmengen und Bauteilpreise berechnet werden, und andererseits alle alten Transaktionen abspeichert um eine neue SpielRunde einzuleiten, wobei sich Auswirkungen auf die SpielWelt ergeben.

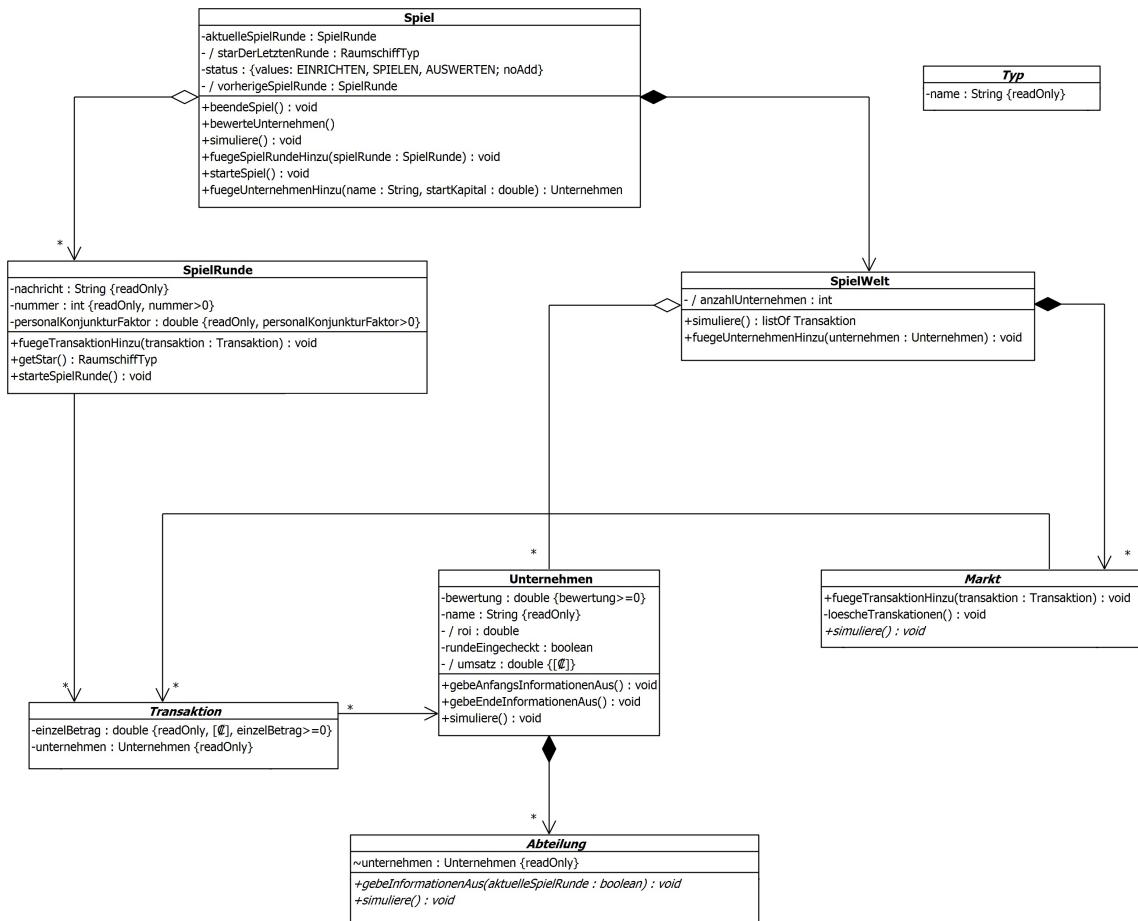


Abbildung 14: Klassendiagramm - Kernelemente

3.4.3 Die Typen

Der Begriff *Typen* bezeichnet, wie oben bereits erwähnt, eine Beschreibung einer Gruppe von Gegenständen der Spielwelt. Es wurde bewusst auf eine Erzeugung jedes einzelnen Objekts in der Spielwelt verzichtet, um die Anzahl der benötigten Objekte nicht Überhand nehmen zu lassen. Obgleich diese Entscheidung vielleicht wie ein Vorgriff auf den Abstraktionsgrad der Implementierung wirkt, wurde hierdurch versucht, die Schere zwischen dem Modell der Entwurfs- und dem der Implementierungsphase möglichst gering zu halten, vor allem da sich die gewählte Möglichkeit nicht negativ auf die Funktionalität auswirkt. Die Gemeinsamkeit, die allen Typen innewohnt, ist der Name, der daher in einer Oberklasse definiert wurde.

Erkannt und umgesetzt wurde dabei die folgenden Typen, die ihrerseits von der Klasse *Typ* erben:

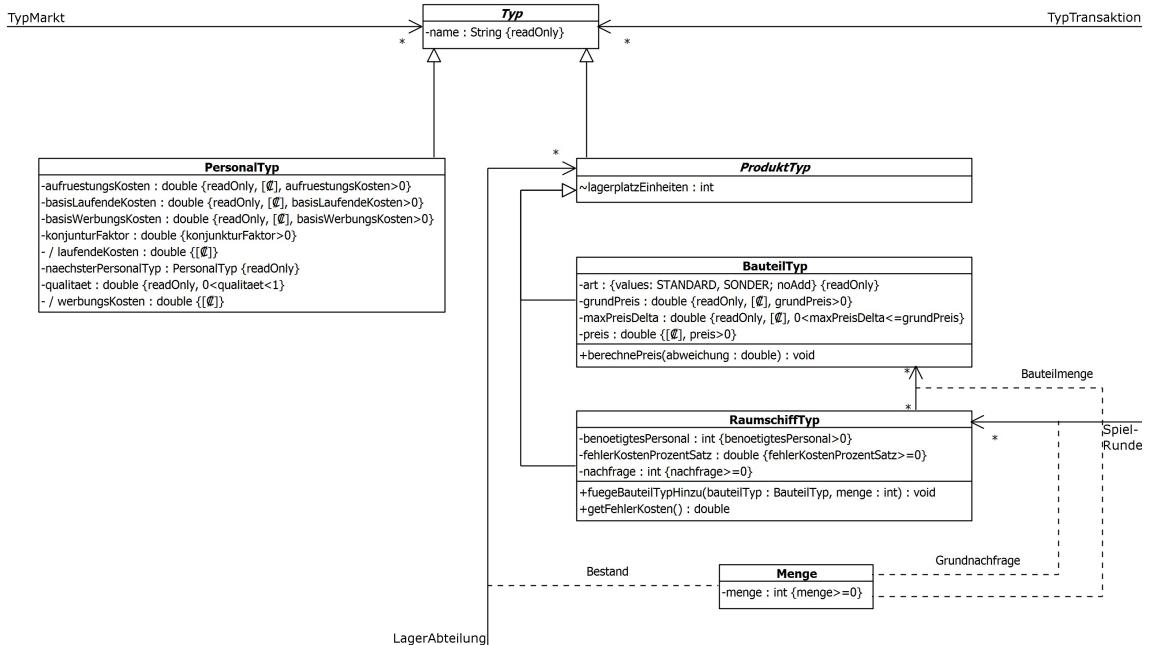


Abbildung 15: Klassendiagramm - Typen

- **PersonalTyp:** Die für das Spiel notwendigen Personaltyphen zeichnen sich einerseits durch ihren Qualitätsgrad, der die entstehenden Fehlerkosten beeinflusst, und andererseits durch die mit ihnen verbundenen Kosten aus. Hierzu gehören die Werbungskosten, die laufenden Kosten sowie die Aufrüstungskosten, wobei die ersten zwei abhängig von der Konjunktur auf dem Personalmarkt variieren können. Zusätzlich wird derjenige Personaltyph, auf den eine Aufrüstung erfolgen kann, gespeichert.
- **BauteilTyp:** In der Spielwelt des Planspiels können Bauteiltypen entweder die Rolle von Standard- oder Sonderbauteilen annehmen, wobei erstere für jeden Raumschiffotyp benötigt werden. Da dies in der Berechnung der neuen Bauteilpreise berücksichtigt werden muss, wird diese Eigenschaft abgespeichert. Zudem kann der Preis eines Bauteiltypen variieren, sodass neben dem Grundpreis auch die maximale Abweichung von diesem und der aktuelle Preis gespeichert werden müssen.
- **RaumschiffTyp:** Diese setzen sich aus verschiedenen Bauteilen zusammen, wobei Bauteile mehrfach benötigt werden können. Daher wurde die assoziative Klasse **Menge** eingeführt. Zudem benötigt ein Raumschifftyp eine gewisse Menge Personal in einer Runde zur Herstellung, und verursacht bei fehlerhaften Herstellung einen gewissen Prozentsatz an Fehlerkosten, gemessen an dem aktuellen Preis der Bauteile, aus denen er sich zusammensetzt.

Da sowohl Raumschifftypen als auch Bauteiltypen im Unternehmen gelagert werden können und für die Berechnung der hierdurch anfallenden Kosten der hierfür benötigte Platz Beachtung findet, erben beide von der abstrakten Klasse *ProduktTyp*. Ein Raumschiff verbraucht dabei so viel Platz wie die Summe seiner Bauteile.

3.4.4 Die Märkte

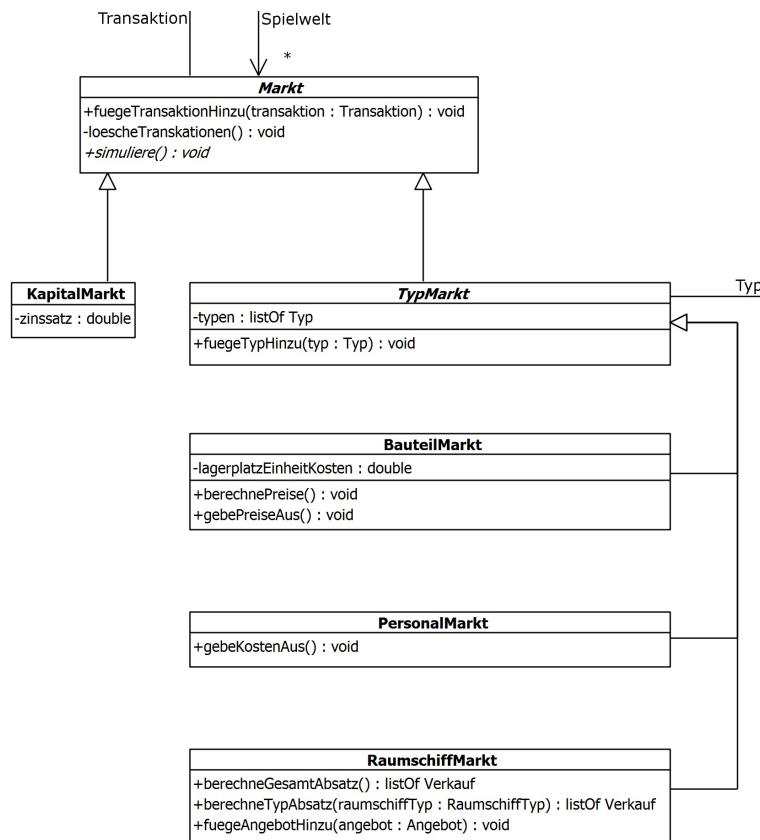


Abbildung 16: Klassendiagramm - Märkte

Im Planspiel gibt es verschiedene Märkte, die alle die zu ihnen gehörenden aktuellen Transaktionen verwalten. Betrachtet man diese näher, lässt sich feststellen, dass die meisten ebenfalls mit einer Art von *Typ* in Verbindung stehen, wobei sie diese Typen auch verwalten sollen. Aus dieser Überlegung heraus ergibt sich die abstrakte Klasse *TypMarkt*. Es folgt nun eine Erläuterung der einzelnen Märkte im Detail:

- **Raumschiffmarkt:** Dieser Markt ist vermutlich derjenige, der sich bei der Betrachtung von Raumschiffproduktions-Unternehmen in den Vordergrund drängt.

Er steht im Zusammenhang mit dem *RaumschiffTyp* und der Transaktion *Verkauf*. Es handelt sich für die Spieler um den Absatzmarkt, an dem sie Verkaufsangebote machen können, die sich durch eine Menge und einen Preis auszeichnen. Alle Angebote, die im Laufe Runde gemacht werden, müssen zwischengespeichert und zum Schluss der Runde ausgewertet werden, um die Absatzwerte berechnen zu können.

- **Bauteilmarkt:** Ähnlich gestaltet sich der BauteilMarkt, der für die Spieler den Beschaffungsmarkt darstellt. Hier werden *Bauteiltypen* und *Einkäufe* verwaltet, zusätzlich gibt es für alle Spieler einheitliche Kosten pro Lagerplatzeinheit, die sie über das Rundenende hinweg in ihrem Lager belegen. Damit sich Spieler informieren können, gibt es hier zudem die Methode *gebePreiseAus()*, die die aktuellen Bauteilpreise ausgibt.
- **Personalmarkt:** Dies ist der letzte *TypMarkt*, der *PersonalTypen* und alle *PersonalTransaktionen* verwaltet. Ebenfalls zur Information dient *gebeKostenAus()*, wodurch die aktuellen Werbungs-, laufenden und Aufrüstungskosten dargestellt werden.

3.4.5 Die Abteilungen

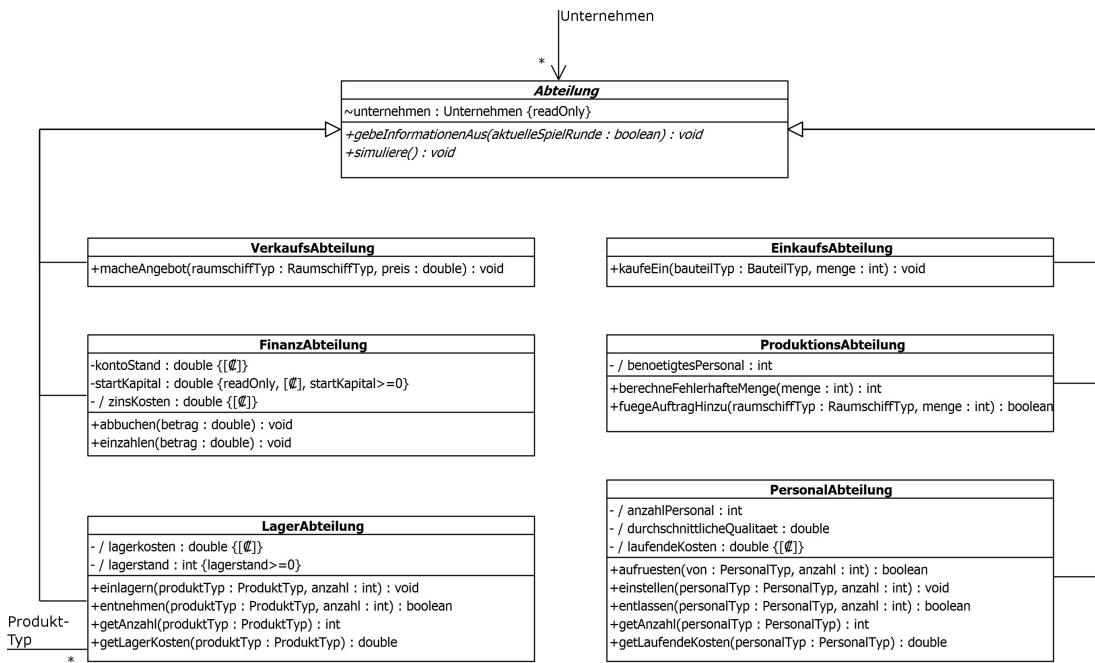


Abbildung 17: Klassendiagramm - Abteilungen

Ein produzierendes Unternehmen besteht aus verschiedenen Abteilungen, zu denen klassischerweise der Einkauf, die Produktion und der Verkauf zählen. Für das Planspiel wurden drei zusätzliche Abteilungen eingeführt: Finanzen, Lager und Personal. Dies folgt nicht in allen Punkten der Realität, dient aber auf der Ebene der Modellierung der einheitlichen Verwaltung aller Teile des Unternehmens. Jede *Abteilung* kann dabei durch *gebeInformationenAus()* Informationen zu dem sie betreffenden Kontext ausgeben. Mit Hilfe des Parameters *aktuelleSpielRunde* lässt sich steuern, ob sich die Informationen auf die aktuelle oder die zurückliegende Spielrunde beziehen sollen, was vor allem im Bereich des Verkaufs interessant ist.

- **EinkaufsAbteilung:** Diese Abteilung dient lediglich dem Einkauf von Bauteilen.
- **ProduktionsAbteilung:** Hier können Produktionsaufträge hinzugefügt werden, die im Laufe der SpielRunde abgearbeitet werden. Dabei muss ein Überblick über das noch zur Verfügung stehende Personal behalten werden. Bei der Simulation können zudem - abhängig von der Qualität des Personals - fehlerhafte Raumschiffe entstehen, die zusätzliche Kosten verursachen.
- **VerkaufsAbteilung:** Hier können Verkaufsangebote gemacht werden, die sich immer auf einen bestimmten Raumschifftypen beziehen. Dabei wird am Ende der Runde versucht, so viele der dann vorhandenen Raumschiffe dieses Typs - Lagerbestände sowie frisch produzierte - zum angegebenen Preis abzusetzen.
- **Finanzabteilung:** Das Unternehmen verwaltet hier seine Finanzen. Der Einfachheit halber beinhaltet dies die Abstraktion eines Kontos, das das Hauptgeschäfts konto des Unternehmens darstellt. Daher gibt es Methoden zum *einzahlen()* und *abbuchen()* von Geldbeträgen. Eine Unterdeckung des Kontos am Rundenende führt zu Zinskosten, die abhängig vom Zinssatz zur kurzfristigen Refinanzierung des Kapitalmarkts anfallen. Zur Berechnung des *ROI* am Ende muss auch das zu Beginn vorhandene Startkapital festgehalten werden.
- **PersonalAbteilung:** Diese verwaltet das angeworbene Personal, welches in verschiedenen Qualitätsstufen vorliegen kann. Die Gesamtanzahl des Personals beeinflusst dabei die Produktionskapazität, die durchschnittliche Qualität beeinflusst die Fehlerrate. Zusätzlich entstehen laufende Kosten, die bei der Simulation der Runde abgebucht werden. Spieler können Personal einstellen, aufrüsten oder je nach Situation auch entlassen.
- **LagerAbteilung:** Bauteile und Raumschiffe müssen nicht sofort verbaut oder verkauft werden, sie werden zunächst in das Lager des Unternehmens gelegt. Daher besteht eine Beziehung zur Klasse *ProduktTyp*, die ergänzt wird durch die assoziative Klasse *Menge*. Abhängig vom Lagerstand, der die Summe der benötigten Lagerplatzeinheiten aller gelagerten Einheiten darstellt, entstehen zum Ende einer Runde Lagerkosten.

3.4.6 Die Transaktionen

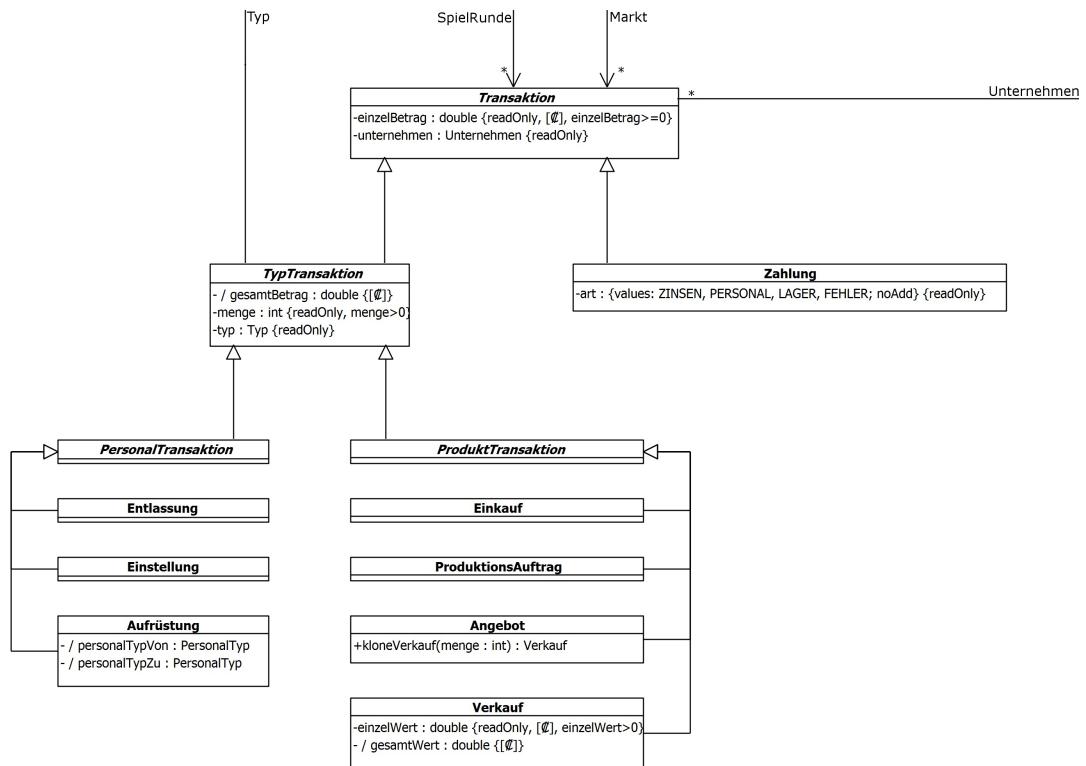


Abbildung 18: Klassendiagramm - Abteilungen

Wie bereits erwähnt müssen Transaktionen, die Spieler durchführen, dauerhaft gespeichert werden, um so die Unternehmen abschließend bewerten zu können. Hierzu sind zwar nicht alle Transaktionen, die auch tatsächlich anfallen, relevant, allerdings scheint eine Speicherung aller einerseits konsequent und ermöglicht andererseits in Zukunft relativ einfach das Hinzufügen einer Funktionalität zum Zurücknehmen bzw. Rückgängigmachen von Entscheidungen, die noch keine weiteren Auswirkungen hatten. Transaktionen stehen dabei fest mit einem Unternehmen in Beziehung und sind gekennzeichnet (mit Ausnahme der Entlassung) durch einen Betrag. Anlehnd an die Struktur der Märkte kann man hier jedoch wieder unterscheiden zwischen solchen Transaktionen, die mit Typen in Verbindung stehen, den *TypTransaktionen*, und der *Zahlung*. Die Typtransaktionen untergliedern sich dabei noch weiter in die *PersonalTransaktionen* und *ProduktTransaktionen*. Stehen sie mit einem Typ in Zusammenhang, wird auch eine Menge gespeichert (die assoziative Klasse wird aufgrund der 1:/* Beziehung nicht benötigt), wodurch sich auch ein Gesamtbetrag ergibt.

Es war dabei notwendig, einige weitere Unterklassen wie *Verkauf* abzuleiten, da an einigen Stellen durch die Spezialisierung weitere Daten abgespeichert werden müssen. Hier erschien es dann konsequent, auch die anderen möglichen Transaktionen in eigene Unterklassen auszulagern, auch wenn sie keine weitere Funktionalität bieten, und die Oberklassen abstrakt zu gestalten.

Es werden hier nur solche Transaktionen aufgeführt, die einer weiteren Erklärung bedürfen:

- **Angebot:** Da ein Angebot im Regelfall in einen fast identischen *Verkauf* übergeführt werden kann, der sich lediglich in der Menge unterscheiden kann, stellt diese Klasse die Methode *kloveVerkauf()* zur Verfügung.
- **Verkauf:** Für einen Verkauf soll auch der derzeitige Wert des Raumschiffotypen, gemessen an der Summe der Kosten für alle benötigten Bauteile, gespeichert werden. Dies dient der Berechnung des Marktanteils, der einerseits aufgrund der mangelnden Vergleichbarkeit nicht nur anhand von Stückzahlen berechnet werden kann, andererseits nicht am Umsatz gemessen werden soll, da dies Unternehmen mit hohen Preisen in der Bewertung zu sehr bevorteilt.

Ende: Fredrik

3.5 Implementierung

3.5.1 Einleitung

Beginn: Britta

Im Rahmen der objektorientierten Softwareentwicklungsmethodik sind wir nun im Teilkapitel der objektorientierten Implementierung angelangt. Bevor die Umsetzung der Entwurfsmodelle und -entscheidungen näher beschrieben wird, werden zwei grundlegende Fragen geklärt. Welche Programmiersprache wird verwendet? Mit Hilfe welches Versionsverwaltungsprogramms wird das Implementieren im Team realisiert?

Was die Programmiersprache angeht, so stand von Anfang an fest, dass es sich um Java handeln würde, da laut Aufgabenstellung jUnit Tests gefordert sind. Die sehr moderne und mächtige Programmiersprache bietet viele Vorteile. So stehen neben einer umfangreichen Funktionsbibliothek auch eine breite Palette an Funktionen und

Typen für den direkten Einsatz bereit, wie z.B. Hashmaps, Vektoren und Strings. Insbesondere Hashmaps, die in Java die Möglichkeit bieten Datenelemente mit Hilfe einer indexierten Tabelle zu speichern und so einen schnellen Zugriff gewährleisten, haben sich an vielen Stellen als sehr hilfreich erwiesen (vgl. Kapitel 3.5.2).

Nicht zuletzt ist Java das Paradebeispiel für eine objektorientierte Sprache. Um die Spielwelt Schritt für Schritt aufzubauen, bedarf es in der Objektorientierung zunächst einer ausführlichen, modellierenden Entwurfsphase, wie sie in Kapitel Kapitel 3 mit UML beschrieben wurde. Diese kann mit Java sehr gut übernommen und realisiert werden; notwendige Abweichungen werden zu einem späteren Zeitpunkt dieses Unterkapitels behandelt. Um im Team effizient arbeiten und vor allen Dingen gemeinsam implementieren zu können, müssen Informationen und Coding schnell, aktuell und jedem zugänglich ausgetauscht und parallel bearbeitet werden können. Zur Fehlervermeidung und schnellen -behebung, sowie zur Vereinfachung der Kommunikation unter den Projektmitgliedern ist es außerdem hilfreich, nachvollziehen zu können, von wem wann etwas geändert wurde. Außerdem stellen dabei Protokollierung, Wiederherstellung alter Dateizustände und die damit verbundene Archivierung der alten Zustände wichtige Anforderungen an ein Versionsverwaltungssystem dar.

Für das Projekt Star Greg wurde sich für das zentrale Versionsverwaltungssystem Git entschieden. Zentral bedeutet hierbei, dass es sich bei dem Aufbau von Git um ein Client-Server-System handelt. Der Zugriff auf das Repository erfolgt über Netzwerk. Der ausschlaggebende Grund für diese Wahl war die Tatsache, dass Git als Open Source Projekt kostenlos zur Verfügung steht und einfach für alle Teammitglieder installiert werden kann, da es sowohl auf UNIX- als auch auf Windowssystemen lauffähig ist. Sicherheit bietet die Rechteverwaltung, die es nur berechtigten Personen ermöglicht, neue Versionen hochzuladen.

Git unterscheidet sich in einigen Punkten von traditionellen Versionskontrollsyste- men. Der wesentliche Unterschied liegt in der Tatsache, dass kein zentraler Projekts- erver zur Verfügung steht, sondern dass sich jedes Teammitglied über den Befehl *pull* eine lokale Kopie des Repositories inklusiver History zieht. Wenn er diese bearbeitet hat und teilen möchte, findet lokal ein Merge statt und die geänderten Dateien werden per *commit* und *push* hochgeladen. So kann weitgehend offline gearbeitet werden.

Ende: Britta

3.5.2 Erläuterung

Beginn: Jan

Die Umsetzung der Entwurfsentscheidungen war ein iterativer Prozess, da viele Klassen des Spiels untereinander in Beziehung stehen und oft nachträgliche Anpassungen in den jeweiligen Klassen vorgenommen werden mussten. Zu den drei grundlegenden Punkten der Javaimplementierung in Star Greg gehören die Verwendung von Hashmaps, die Verwendung generischer Klassen und die Umsetzung der Datenhaltung. Auf sie wird im Folgenden näher eingegangen.

Hashmaps

In Star Greg kommen Hashmaps vor allem bei der Realisierung der assoziativen Klassen zum Einsatz. Beispielsweise wird in der Klasse LagerAbteilung eine Hashmap eingesetzt, um die Lagerung verschiedener Produkttypen mit ihren jeweiligen Mengen zu gewährleisten (siehe Listing 1).

```
private final IntegerHashMap<ProduktTyp> bestand = new
IntegerHashMap<ProduktTyp>();
```

Listing 1: Hashmap zur Lagerung von verschiedenen Produkttypen

Im Lager können zum einen Bauteile, zum anderen Raumschiffe eingelagert werden. Da die Klassen BauteilTyp und RaumschiffTyp beides Unterklassen von ProduktTyp sind, können von beiden Klassen auch die jeweiligen Objekte in der Hashmap unter Verwendung der Methode einlagern() (siehe Listing 2) gespeichert werden, welche in LagerAbteilung implementiert ist.

```
public void einlagern (ProduktTyp produktTyp, int anzahl)
{
    bestand.add(produktTyp, anzahl);
    this.lagerstand += produktTyp.getLagerplatzEinheiten()
        * anzahl;
}
```

Listing 2: *einlagern()* der Klasse LagerAbteilung

Generische Klassen

Die Verwendung generischer Klassen verkleinert die Anzahl der benötigten Klassen in Star Greg und gewährleistet Typsicherheit, da sie eine Typprüfung zur Kompilierzeit erzwingt. Somit können etwaige Implementierungsfehler schon vor der Laufzeit abgefangen werden.

Datenhaltung

Um die Datenhaltung im Spiel besser zu verstehen, wird nachfolgend auf die Klassen bzw. deren Objekte eingegangen, die an der Speicherung von Daten beteiligt sind.

Zuerst wird ein Spielobjekt angelegt, welchem dann die verschiedenen Märkte (Bauteilmarkt, Raumschiffmarkt, Personalmarkt und Kapitalmarkt) zugeordnet werden, in denen die Datenhaltung letztendlich realisiert wird. Danach werden dem Spiel die verschiedenen Unternehmensobjekte beigefügt, wobei zu jedem Objekt automatisch die zugehörigen Abteilungsobjekte erzeugt werden. Die Arten der Datenhaltung lassen sich bei Star Greg in zwei Kategorien einteilen - die Datenhaltung, die sich auf immer nur auf die derzeitigen Werte der Unternehmen beziehen, und die Datenhaltung, die mehrere Spielrunden betrifft. Erstere bezieht sich beispielsweise auf die Kontostände und die Lagerstände der Unternehmen. Diese Datenhaltung ist so realisiert, dass die einzelnen Unternehmensobjekte und deren Abteilungsobjekte Attribute besitzen (z.B. Kontostand in einer Finanzabteilung), die die aktuellen Werte für das Unternehmen beinhalten.

Die Umsetzung der zweiten Art der Datenhaltung wurde mit Hilfe der Transaktionen der Unternehmen umgesetzt. Transaktionen beeinflussen sowohl Größen der einzelnen Unternehmen, als auch Größen, die für die gesamte Spielwelt (also alle Unternehmen) eine Rolle spielen, wie die Bauteilpreise und die Nachfrage nach den Raumschiffen. Zudem ist mit den Transaktionen die Wertrückverfolgung von Daten gewährleistet. Um diese Art der Datenhaltung noch transparenter zu machen, wird an dieser Stelle auf die Beispiel einer Einkaufstransaktion von Bauteilen eingegangen. Ein Unternehmen tätigt seine Transaktionen mit Hilfe seiner Abteilungen, in unserem Beispiel also mit Hilfe seiner Einkaufsabteilung. Hier ist die Methode *kaufeEin()* realisiert (siehe Listing 3).

Bei dem Einkauf müssen der Bauteiltyp und die Menge des Bauteiltyps mitgegeben werden. In der Methode selbst wird zunächst ein Objekt der Klasse Einkauf erzeugt, welche eine Unterklasse von Transaktion ist. Jede Transaktion wiederum besitzt ein

```

public void kaufeEin(BauteilTyp bauteilTyp, int menge) {
    Einkauf einkauf = new Einkauf(bauteilTyp, unternehmen,
        menge, bauteilTyp.getPreis());
    double kosten = einkauf.getGesamtBetrag();
    unternehmen.getFinanzen().abbuchen(kosten);
    unternehmen.getLager().einlagern(bauteilTyp, menge);
    unternehmen.getSpielWelt().getBauteilMarkt().
        fuegeTransaktionHinzu(einkauf);
}

```

Listing 3: *kaufeEin()* der Klasse EinkaufsAbteilung

Attribut `unternehmen`, wodurch sie sich den verschiedenen Unternehmen eindeutig zuordnen lässt. Als nächstes folgt die entsprechende Kostenabbuchung im Finanzabteilungsobjekt und die Einlagerung im Lagerabteilungsobjekt des Unternehmens. Die letzte Zeile des Codes stellt den entscheidenden Schritt der Datenhaltung in unserem Beispiel dar. Da jedes Unternehmen einer Spielwelt zugeordnet ist, wird als erstes der Bezug auf die Spielwelt hergestellt. Mit Hilfe der Spielwelt wird dann der zugehörige Bauteilmarkt angesprochen, wobei jeder Markt besitzt einen Vektor, in denen Transaktionen verwaltet werden können. Der Bauteilmarkt sammelt somit alle Einkaufstransaktionen der Unternehmen, die am Ende einer Runde in dem jeweiligen Spielrundenobjekt abgespeichert werden. Da es in jeder Spielwelt über den gesamten Spielverlauf nur ein Objekt jedes Marktes geben soll, werden in der Simulationsphase die Transaktionen dann aus den jeweiligen Märkten entfernt, wodurch gewährleistet ist, dass keine redundanten Daten vorliegen und nur neue Transaktionen für die Berechnung schwankender Werte dienen.

In Tabelle 14 ist dargestellt, welche Daten in den jeweiligen Abteilungen und Märkten gespeichert werden, wobei nicht alle Abteilungen und Märkte aufgelistet sind, da nicht in allen die Speicherung von sich verändernden Daten vorkommt (z.B. Verkaufsabteilung und Kapitalmarkt).

Ende: Jan

3.5.3 Das Hilfspaket *util*

Beginn: Fredrik

Für die Realisierung des Planspiels wurde an einigen Stellen wiederkehrbare, abstrakte Funktionalität benötigt. Um diese Anforderungen mit möglichst geringem

Klasse	Speicherung von
Finanzabteilung	Kontostand
Lagerabteilung	Lagerkosten und Lagerstand
Personalabteilung	Anzahl des Personals, die durchschnittliche Qualität des Personals und die laufenden Kosten des Personals
Produktionsabteilung	Benötigtes Personal für die Raumschiffproduktion
Bauteilmarkt	Einkaufstransaktionen für Bauteile
Personalmarkt	Personaltransaktionen (z.B. Einstellung von Personal)
Raumschiffmarkt	Angebote und Verkäufe von Raumschiffen

Tabelle 14: Speicherung von Daten in unterschiedlichen Klassen

Aufwand zu lösen und die Möglichkeit der Wiederverwendbarkeit für andere Projekte aufzugreifen, wurden die hierfür entwickelten Elemente in das Paket *util* ausgelagert. Dazu zählen Methoden zum Umgang mit den Java Datentypen *Vector* und *HashMap*, sowie eine Klasse zur Ausgabe von Tabellen.

Umgang mit *Vector* und *HashMap*

Augrund der fehlenden Datenbank-Anbindung wurde die Speicherung von Daten vollständig über Klassen und Objekte realisiert. Allerdings stellt sich hier zuweilen der Zugriff auf einzelne Daten und das Aggregieren von Werten umständlicher da, als gewollt. Für diesen Zweck gibt es bereits mächtige Bibliotheken, wie beispielsweise die *Google Collections Library*. Auch diese hätte für den benötigten Zweck eingesetzt werden können, allerdings würde nur ein Bruchteil der Funktionalität benötigt, wozu aber die ganze Bibliothek integriert werden müsste.

Stattdessen wurden eigene generische Methoden implementiert, die die folgende Funktionalität bieten:

- **gruppierenVector:** Diese Funktion nimmt einen *Vector* sowie ein Objekt entgegen, dass das Interface *Gruppierung* implementiert, also eine Funktion *nach()* implementiert. Dies bewirkt, dass für jedes Element des Vectors die *nach()*-Funktion aufgerufen wird, um die Gruppe, zu der das Element gehören soll, zu bestimmen. Als Rückgabe wird eine *HashMap* geliefert, die jeder Gruppe einen *Vector* zuordnet, der die zu dieser Gruppe gehörenden Elemente enthält.
- **filtereVector:** Hier wird ein *Vector* nach einem bestimmten Kriterium gefiltert, und es werden die Elemente in einem *Vector* zurückgeliefert, die dem Filter entsprechen. Auch hierzu wurde ein Interface verwendet, das ähnlich aufgebaut ist

wie das oben erwähnte.

- **summriereVector:** Dies entspricht einer Aggregationsfunktion über einer Spalte in einer Datenbank. Implementiert wurde auch diese Funktionalität mit einem Interface, das eine Methode enthält, die für ein bestimmtes Element den aufzusummierenden Wert zurückliefert. Somit wird die Summe aller dieser Werte zurückgeliefert. In Kombination mit *gruppiereVector* ist es möglich, z.B. alle Verkäufe nach dem jeweiligen Unternehmen zu gruppieren und anschließend den Umsatz für jedes Unternehmen zu berechnen.

Aufbauend auf diesen allgemeinen Hilfsfunktionen wurden dann einige speziellere, häufig benötigte Funktionen wie *gruppiereVerkaeufeNachUnternehmen()* erstellt.

Die Klasse **TableBuilder**

Mit zunehmender Komplexität des Planspiels wurde es nötig, verschiedenste Daten für den Spieler möglichst übersichtlich über die Textausgabe darzustellen. Orientierung bot dabei die Standardklasse *StringBuilder*. So ist es möglich einen TableBuilder zu instanzieren um anschließend verschiedene Elemente, Spaltenenden, gleich ganze Zeilen oder zusätzliche horizontale Linien einzufügen. Abschließend kann der gesamte Inhalt mit Hilfe der *print()*-Funktion ausgegeben werden, wobei die erste Zeile wahlweise als Kopfzeile gekennzeichnet werden kann.

Diese Klasse wird durchgängig für die Ausgabe von Daten der Raumschiffmärkte und Abteilungen verwendet, und hat sich durch die zentrale Modifizierbarkeit der Darstellung der Ausgabe sowie der einfachen Verwendung zur Reduktion des benötigten Ausgabecodes bewährt.

Ende: Fredrik

3.5.4 Umsetzung der Spiellogik

Beginn: Jan

Die Umsetzung der Spiellogik (siehe Kapitel 2.6) konnte in Java direkt in den jeweiligen Klassen vollzogen werden, da die dahinter liegenden Algorithmen unabhängig von der gewählten Programmiersprache sind. Es mussten allerdings oft die Methoden der eigenen Klasse Util (siehe Kapitel 3.5.3) zu Hilfe gezogen werden, um Bei-

spielsweise eine Gruppierung der Bauteile zu realisieren. Nachfolgend wird auf die jeweiligen Programmteile der entsprechenden Klassen eingegangen.

Bauteilpreise

Die neuen Bauteilpreise werden in der Klasse BauteilTyp mit Hilfe der Methode *berechnePreis()* (siehe Listing 4) ermittelt, wobei die Logik genau der Formel Gleichung 5 aus dem Kapitel 2.6 entspricht.

```
public void berechnePreis(double abweichung) {
    preis = grundPreis - maxPreisDelta
        + 2 * Math.pow(maxPreisDelta, 2)
        / (maxPreisDelta
            + maxPreisDelta
            / (Math.pow(2 / maxPreisDelta + 1,
                2 * abweichung * maxPreisDelta)));
}
```

Listing 4: *berechnePreis()* der Klasse BauteilTyp

Die Methode *berechnePreis()* wird am Ende jeder Spielrunde in der Klasse BauteilMarkt in der Methode *berechnePreise()* (siehe Listing 5) aufgerufen, nachdem die Umsätze der einzelnen Bauteiltypen, der Durchschnittsumsatz und die Abweichung der Bauteiltypumsätze vom Durchschnittsumsatz berechnet wurden. Dies ist im Bauteilmarkt möglich, da hier die Einkaufstransaktionen der Unternehmen einer Spielrunde gespeichert sind.

Absatzmengen

Die Umsetzung der Nachfrage nach den Raumschiffen und der daraus resultierenden Absatzmengen ist in der Klasse RaumschiffMarkt realisiert, also in der Klasse, in der auch die Angebote der Unternehmen verwaltet werden. Der RaumschiffMarkt enthält die Methode *berechneTypAbsatz()* (siehe Listing 6), welche als Ergebnis die Verkäufe für die einzelnen Angebote bezogen auf einen RaumschiffTyp wiedergibt. Zunächst werden die Anteile nach der Gleichung 6 aus Kapitel 2.6.3 berechnet.

Hierbei wird auch der kleinste Preis der Angebote bestimmt. Als nächste folgt die Berechnung der neuen Nachfrage nach dem jeweiligen Raumschifftyp (siehe Listing 7), basierend auf der Gleichung 8 aus Kapitel 2.6.3.

```

private void berechnePreise() {
    [...]
    for(BauteilTyp bauteilTyp : typen) {
        [...]
        double abweichung = (umsaetze.get(bauteilTyp) -
            durchschnittsUmsatz) / durchschnittsUmsatz;
        [...]
        bauteilTyp.berechnePreis(abweichung);
        [...]
    }
    [...]
}

```

Listing 5: *berechnePreise()* der Klasse BauteilMarkt

```

public Vector<Verkauf> berechneTypAbsatz(RaumschiffTyp
    raumschiffTyp, Vector<Angebot> angebote) {
    [...]
    for (Angebot angebot : angebote) {
        double anteil = 1.0 / Math.pow(angebot.
            getEinzelBetrag(), 3);
        angebot.setAnteil(anteil);
        [...]
    }
    [...]
}

```

Listing 6: *berechneTypAbsatz()* der Klasse RaumschiffMarkt (1)

```

int nachfrage = raumschiffTyp.getNachfrage();
nachfrage = (int) Math.floor(
    nachfrage * (
        1 - Math.pow(
            niedrigsterPreis / (
                raumschiffTyp.getKosten() * 3.5),
            4)));

```

Listing 7: *berechneTypAbsatz()* der Klasse RaumschiffMarkt (2)

Zum Schluss folgt die letztliche Berechnung der Verkaufsmengen, wobei der relative Absatz aus der Gleichung 7 aus Kapitel 2.6.3 mit der berechneten Nachfrage multipliziert wird.

Fehlerkosten

In der Klasse ProduktionsAbteilung ist die Berechnung der Fehlerkosten implementiert. Hierfür dient die Methode *berechneFehlerhafteMenge()* (siehe Listing 9), worin zufällige Zahlen von 0 bis <1 mit der durchschnittlichen Qualität des Personals verglichen werden. Eine durchschnittliche Qualität von 80% entspricht dabei beispielsweise einem Wert von 0,8.

```
private int berechneFehlerhafteMenge(int menge) {
    int fehlerhafteMenge = 0;
    for (int i=0; i<menge; i++) {
        if (Math.random() > unternehmen.getPersonal().
            getDurchschnittlicheQualitaet()) {
            fehlerhafteMenge++;
        }
    }
    return fehlerhafteMenge;
}
```

Listing 8: *berechneFehlerhafteMenge()* der Klasse ProduktionsAbteilug

In der Methode *simuliere()* der Klasse ProduktionsAbteilung wird für jeden Produktionsauftrag die entsprechende fehlerhafte Menge mit den dazugehörigen Fehlerkosten verrechnet.

Bewertung

Die Bewertung der Unternehmen ist in den Unternehmenobjekten selbst realisiert. In der Klasse Unternehmen gibt es die Methode *getROI()*, die den relativen ROI für das Unternehmen gemäß Listing 9 von Kapitel 2.6.5.

Ende: Jan

```

public double getROI() {
    double rOI = finanzen.getKontostand();
    for (BauteilTyp bauteilTyp : spielWelt.getBauteilMarkt
        ().getTypen()) {
        rOI += lager.getAnzahl(bauteilTyp) * bauteilTyp.
            getPreis() * 0.5;
    }
    for (RaumschiffTyp raumschiffTyp : spielWelt.
        getRaumschiffMarkt().getTypen()) {
        rOI += lager.getAnzahl(raumschiffTyp) * raumschiffTyp
            .getKosten() * 0.75;
    }
    rOI = (rOI - finanzen.getStartKapital()) / finanzen.
        getStartKapital();
    return rOI;
}

```

Listing 9: *berechneFehlerhafteMenge()* der Klasse ProduktionsAbteilung

3.5.5 Entscheidungsfindung der Unternehmen

Beginn: Philipp

Zur Entscheidungsfindung der Unternehmen trägt die Informationsphase bei. Die Methode *gebeAnfangsInformationenAus()* der Klasse Unternehmen wird zu Beginn jeder Spielrunde ausgeführt und versorgt den Spieler auf einer Übersichtsseite mit Informationen bezüglich seines Absatzes, seiner Finanzen, der angefallenen Lagerkosten und des aktuellen Preisniveaus. Diese Informationen dienen Ihm für weitere Entscheidungen.

Wenn beispielsweise der Preis für ein Bauteil gerade sehr niedrig ist, so könnte er in dieser Spielrunde besonders viele Bauteile dieses Typs kaufen. Dies bringt zum einen Kosteneinsparungen mit sich, zum anderen steigen die Preise in der nächsten Spielrunde und die Konkurrenzunternehmen müssen die zuvor günstigen Bauteile zu hohen Preisen kaufen. Des Weiteren erscheint beim Eintreten eines Events, das fest in unserer Storyline vorgegeben ist, eine Nachricht. Wird in dieser Nachricht beispielsweise ein Nachfragerückgang bei X-Wings vorhergesagt, so sollte der Spieler die Preise für X-Wings senken um diese absetzen zu können.

Mit Hilfe der Methode *gebeEndInformationenAus()* kann der Spieler jederzeit einsehen welche Änderungen er im Laufe der Spielrunde vorgenommen hat und wie sich

diese auswirken.

3.5.6 Spielablauf

Nachdem ein Unternehmen alle gewünschten Änderungen vorgenommen hat, kann es die Runde einchecken. Dies geschieht mit Aufruf der Methode `rundeEinchecken()` in der Klasse Unternehmen. Erst wenn alle Unternehmen eingechoekt haben, kann die Runde simuliert werden.

Um die Runde zu simulieren muss der Spielleiter die Methode `simuliere()` der Klasse Spiel aufrufen. Diese Methode simuliert dann alle Transaktionen auf dem Bauteil-, Personal- und Absatzmarkt.

Ende: Philipp

4 UI Mockups

4.1 Einleitung

Beginn: Marcel

Die Benutzeroberfläche spielt bei einem Unternehmensplanspiel eine erhebliche Rolle. Sie soll einfach und ansprechend gestaltet sein, sodass sich der Spieler auf den Kern des Unternehmensplanspiels konzentrieren kann.

Zur Konzeption der Benutzeroberfläche wurden Mockups entwickelt, die den Grundaufbau des späteren Unternehmensplanspiels auf dem Bildschirm zeigen sollen. Die Mockups wurden mit dem Mockuptool Balsamiq erstellt. Dieses Tool bietet eine einfache und flexible Lösung zur Erstellung von Mockups. Trotz der eingeschränkten Funktionen des Tools ist es möglich, eine Navigation innerhalb der Mockups zu entwickeln, sodass der Spielablauf anhand der Benutzeroberfläche erkannt werden kann. In den folgenden Kapiteln werden die einzelnen Mockups näher erläutert.

4.2 Startbildschirm

Zu Beginn des Unternehmensplanspiels Star Greg öffnet sich ein Startbildschirm. Hierbei muss der Spieler, sein Unternehmen einrichten und somit ein neues Spiel generieren lassen. Zur Einrichtung des Unternehmens wird ein Unternehmensname erforderlich, der nach belieben festgelegt werden kann. Hierzu steht ein Textfeld oberhalb des sogenannten „CoverFlows“ bereit. Neben dem Unternehmensnamen muss sich der Spieler auch für ein Firmenlogo entscheiden. Hierfür steht ein Overflow bereit, das verschiedene Bilder enthält.

Nach getätigter Eingabe des Unternehmensnamen und der Auswahl des Firmenlogos bestätigt der Spieler seine Eingaben durch drücken des rot hinterlegten „Star Greg Starten“ Buttons. Durch diesen Button wird der UseCase „Unternehmen einrichten“

ausgelöst. Infolgedessen wird das Unternehmensplanspiel gestartet und der Spieler gelangt zur nächsten Benutzeroberfläche, der Spielübersicht(s. Abbildung 19).



Abbildung 19: Startbildschirm

Ende: Marcel

4.3 Spielübersicht

Beginn: Julia

Befindet sich der Spieler auf der Spielübersicht, so kann er sich schnell einen Überblick über das Unternehmen schaffen. Der ausgewählte Unternehmensnamen sowie das Unternehmenslogo ist im oberen Bereich des Bildschirmes zu sehen. Direkt darunter findet man eine Anzeige über die aktuelle Spielrunde und wie viele Runden noch ausstehen.

Desweiteren hat der Spieler eine Übersicht über die verkauften Raumschiffe der letzten Runden in Form eines Säulendiagramms. Die eigenen Daten sind hierbei dem

Branchendurchschnitt gegenübergestellt wodurch der Spieler die aktuelle Lage seines Unternehmens besser einschätzen kann. Rechts daneben befindet sich ein Kreis- oder Kuchendiagramm welches Aufschluss über den Marktanteil des Unternehmens gibt. Als dritte Informationsquelle auf der Spielübersicht dient der “Star der letzten Runde”. Dieser, im Mockup auf Abbildung 20 auf S. 65 ist es der Millenium-Falke, kürt das meistverkaufte Raumschiffmodell der letzten Runde.

Im unteren Bereich des Bildschirmes hat der Spieler unter anderem die Möglichkeit sich über den “Spielregeln” Button noch einmal selbige durchzulesen. Sind in der aktuellen Spielrunde Ereignisse wie zum Beispiel ein Streik aufgetreten, so kann sich der Spieler diese Informationen über den Button “Informationen zur aktuellen Runde” noch einmal anschauen. Um zur ausführlichen Bewertung der letzten Spielrunde zu gelangen, klickt der Spieler auf den Button “Bewertung einsehen”. Zudem findet der Spieler auf der Spielübersicht den grün hinterlegten Button “Spielrunde einchecken” den gleichnamigen UseCase zum abschließen einer Spielrunde auslöst.

Am linken Rand der Benutzeroberfläche befindet sich die Navigationsleiste. Über diese kann der Spieler zu den einzelnen Abteilungen wechseln und dort Informationen einsehen oder Transaktionen tätigen. Über die Navigationsleiste gelangt der Spieler beispielsweise zum Finanzwesen, welches in Kapitel 4.4 beschrieben wird.

Ende: Julia

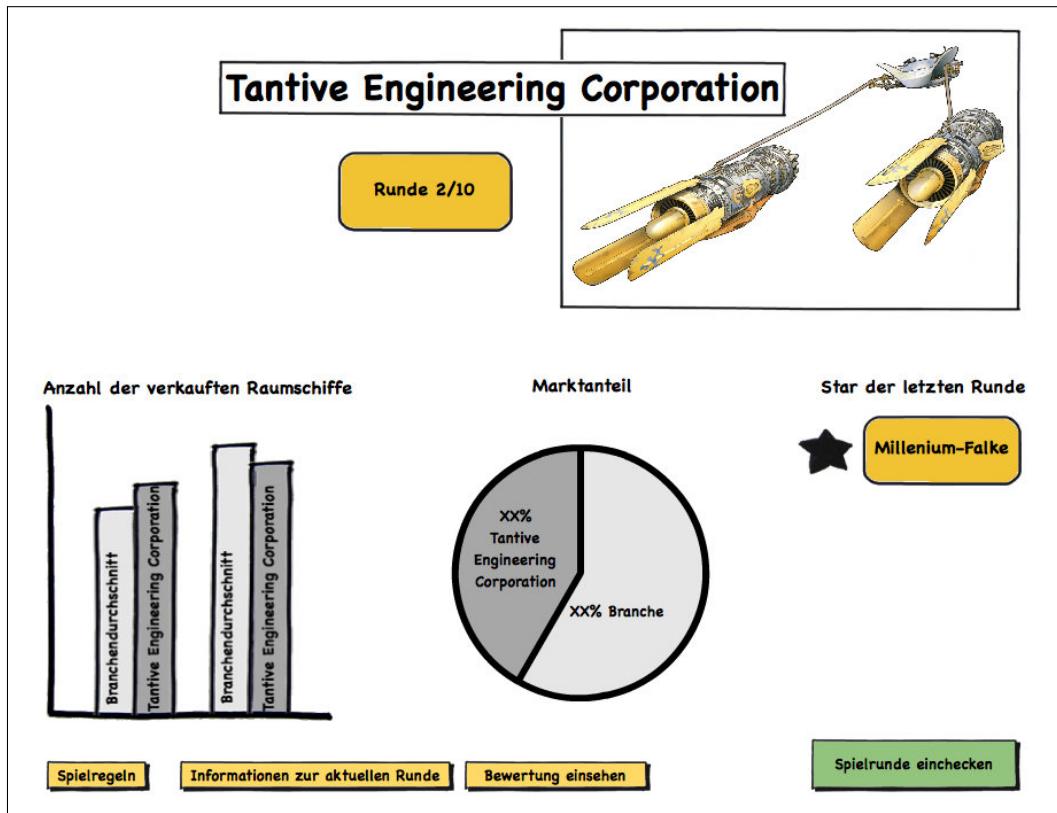


Abbildung 20: Übersichtsbildschirm

4.4 Finanzen

Beginn: Marcel

Eine zentrale Informationsquelle des Spielers zeigt die Benutzeroberfläche „Finanzen“. Hier erhält der Spieler eine Übersicht über alle im Unternehmen durchgeföhrten Geldflüsse. Die Benutzeroberfläche ist in drei zentrale Elemente untergliedert. Es handelt sich um die Einnahmen und Ausgaben pro Spielrunde sowie einen kurzen Überblick über die gesamte Geldsituation in dem Unternehmen.

Die Einnahmen pro Spielrunde werden den Ausgaben pro Spielrunde gegenübergestellt und befinden sich im linken Teil der Benutzeroberfläche. Das Unternehmensplanspiel Star Greg ist darauf ausgerichtet, ausschließlich Raumschiffe zu produzieren. Deshalb besteht der Umsatz des Unternehmens lediglich aus den Preisen des jeweiligen Raumschiffs multipliziert mit der verkauften Anzahl. Daraus folgt, dass es insgesamt drei Einnahmequellen gibt. Es handelt sich um den Umsatz des X Wings, der Correllian Corvette und den Umsatz des Millenium Falken. Diese drei verschie-

denen Umsätze werden in einer einfachen Tabelle dargestellt und aufsummiert. Die Gesamteinnahmen der Spielrunde werden dem Spieler in einem grün hinterlegten Textfeld unterhalb der drei Einnahmequellen angezeigt.

Neben den Einnahmen gibt es Ausgaben pro Spielrunde. Diese werden im rechten Teil der Benutzeroberfläche dargestellt. Die Ausgaben fallen in drei Bereichen des Unternehmens an. Es handelt sich um die Produktion, das Lager und das Personal. Die Ausgaben der Produktion setzen sich aus fünf Posten zusammen. Es sind Gesamtkosten für alle gekauften Triebwerke, Hitzeschilder, Rumpfbauteile und Sonderbauteile. Zusätzlich werden zu den Produktionskosten auch die durch den Ausschuss verursachten Zusatzkosten hinzugerechnet. Unterhalb der Produktionskosten werden die Personalkosten aufgelistet. Diese setzen sich aus den Werbungskosten, laufenden Kosten (Wartungskosten), und den Kosten der Umbaumaßnahmen zusammen. Das dritte Element der Ausgaben pro Spielrunde sind die gesamten Lagerkosten. Sie setzen sich aus den belegten Lagerplatzeinheiten multipliziert mit den Kosten pro Lagerplatzeinheit zusammen.

Die gesamten Ausgaben pro Spielrunde werden in einem rot hinterlegten Textfeld dargestellt. Durch das grün hinterlegte Textfeld der Einnahmen pro Spielrunde und das rot hinterlegte Textfeld der Ausgaben pro Spielrunde soll der Spieler schon bei kurzer Betrachtung des Bildschirms einen Überblick über die Geldflüsse im Unternehmen bekommen.

Im unteren Teil der Benutzeroberfläche befindet sich ein Überblick über die gesamte Finanzsituation im Unternehmen. In der ersten Zeile der Tabelle wird der Kontostand zu Beginn der Spielrunde angezeigt. Darunter befinden sich die Ein- und Ausgaben der Spielrunde sowie der Kontostand am Ende der Spielrunde(s. Abbildung 21).

Ende: Marcel

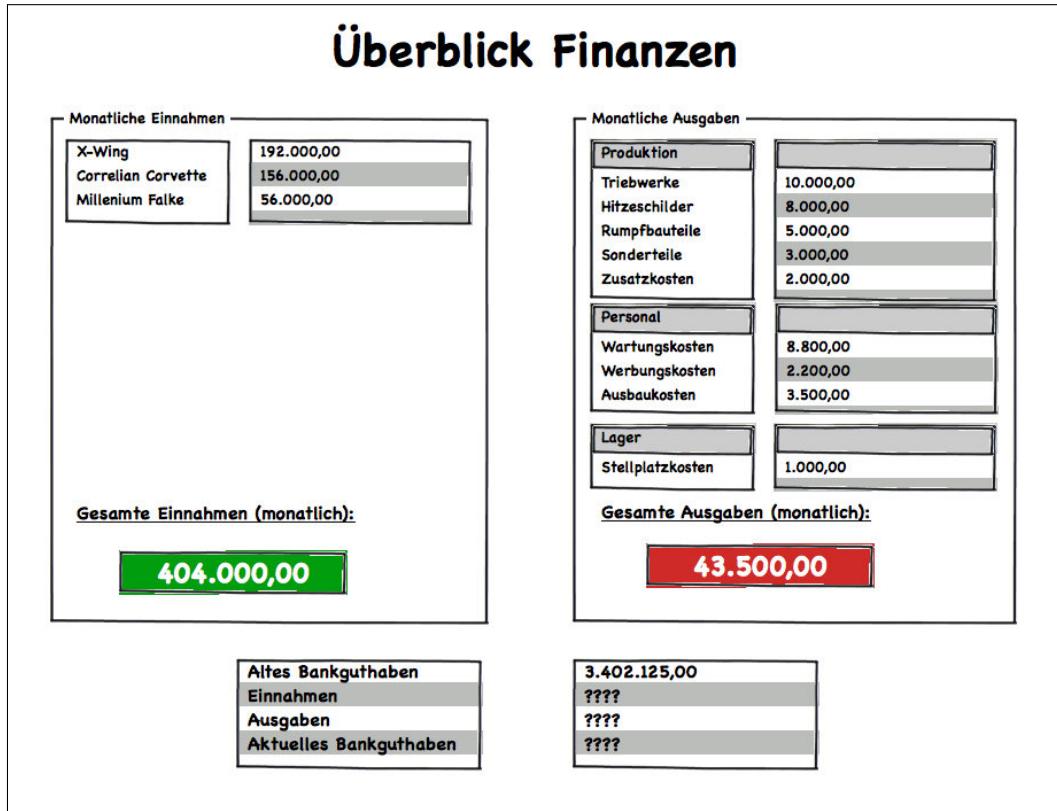


Abbildung 21: Finanzen

4.5 Personalwesen

Beginn: Julia

Um Raumschiffe produzieren zu können, benötigt der Spieler arbeitstüchtige Droiden, welche im Bildschirm “Personal” verwaltet werden. Dieser ist auf Abbildung Abbildung 22 auf S. 69 zu sehen.

In der oberen Hälfte des Bildschirmes sind drei verschiedene Droidentypen zu sehen, welche dem Spieler zur Verfügung stehen. Diese Typen sind in drei verschiedene Stufen unterteilt, welche jeweils in eigenen Containern dargestellt werden. Jeder Container enthält spezifische Daten zum jeweiligen Droidentyp. So wird beispielsweise der Droide Stufe eins im ersten Container beschrieben. Dabei ist ein Bild des Droiden zu sehen. Um nähere Informationen zu diesem Typ zu erhalten, steht dem Spieler ein Hilfebutton in Form eines Fragezeichens in der rechten unteren Ecke des Bildes zur Verfügung. Rot hinterlegt ist zum einen der Preis eines Droiden, welcher sich aus der später beschriebenen Datenbasis aus Kapitel 2.4.3 auf S. 15 ergibt.

Daneben kann der Spieler die Anzahl der anzuheuernden Droiden auswählen. Aus der Anzahl und dem Preis der Droiden ergeben sich dann die Gesamtkosten für diese Droidenstufe. Da ein Droide monatlich gewartet werden muss, findet man die dafür anfallenden Kosten ebenfalls in dem Container zur jeweiligen Droidenstufe. Sind alle anzuheuernden Droiden ausgewählt, so kann der Spieler auf der rechten Bildschirmseite die gesamten Anheuerkosten ablesen und durch den grünen Button “Ausgewählte Droiden anheuern” den Einkauf bestätigen welches dem UseCase “Personal einstellen” entspricht.

Auf der linken unteren Hälfte des Bildschirmes befindet sich zum einen ein Container zum Ausbauen von Droiden und zum anderen einen Cointainer zum Entlassen von Droiden. Im Bereich “Ausbauen” hat der Spieler die Möglichkeit, Droiden zur Stufe 2 beziehungsweise zur Stufe 3 auszubauen. Beim Ausbauen zur nächst höheren Stufe fallen Kosten an, welche den Rot hinterlegten Textfeldern zu entnehmen sind. Rechts daneben kann die gewünschte Anzahl für den Ausbau angegeben werden. Hier hat der Spieler die Möglichkeit sich über den Hilfebutton genauer über das Ausbauen von Droiden zu informieren. Mit dem Button “Ausbau starten” wird der UseCase “Personal weiterbilden” ausgelöst. Darunter befindet sich der Bereich “Entlassen” in dem zur jeweiligen Stufe die gewünschte Anzahl an Droiden auszuwählen ist, welche entlassen werden soll. Auch hier befindet sich ein Hilfebutton für genauere Informationen. Der Button “Droiden entlassen” bestätigt die Auswahl. Dies stellt die Realisierung des UseCase “Personal entlassen” dar.

Als weitere Informationsquelle für den Spieler dient das Liniendiagramm auf der rechten unteren Seite des Bildschirms. Hierbei werden die verfügbaren Droiden über die verschiedenen Spielrunden dargestellt. Eine Trennung der einzelnen Droidenarten wird durch unterschiedliche Linien getroffen.

Ende: Julia

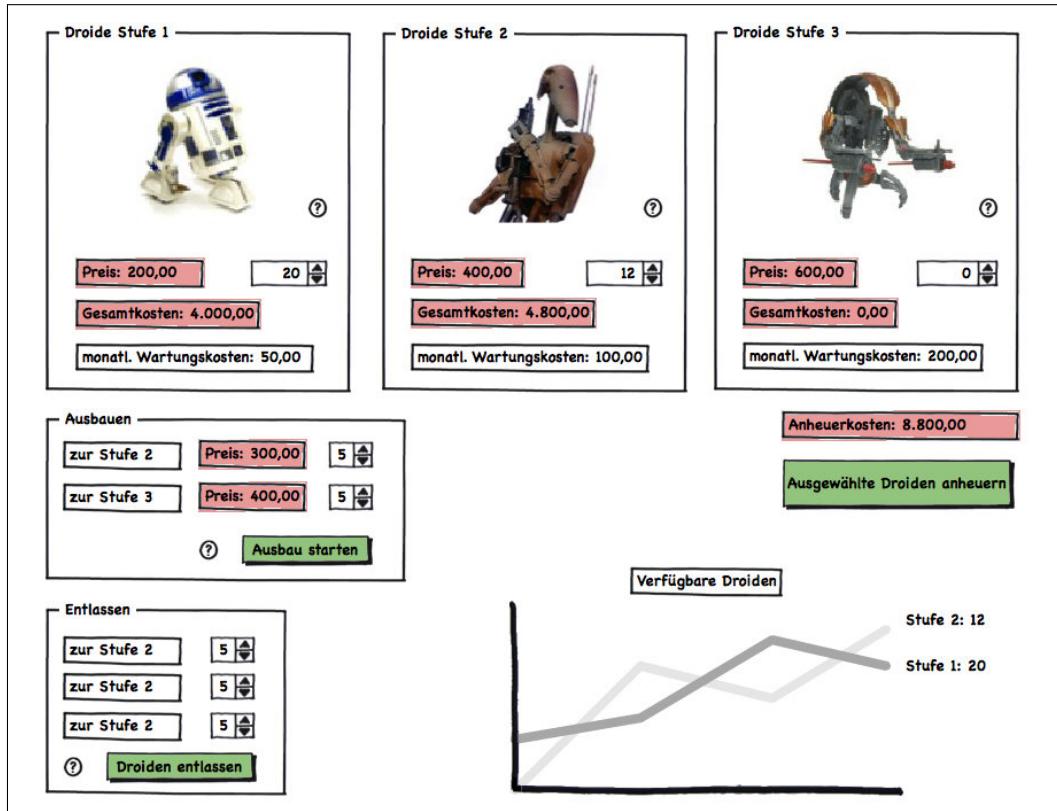


Abbildung 22: Personalwesen

4.6 Einkauf

Beginn: Marcel

Zur Produktion der Raumschiffe werden diverse Bauteile benötigt. Hierzu steht der Bildschirm „Einkauf“ dem Spieler zur Verfügung.

Im oberen Teil des Bildschirms befinden sich drei Container, die jeweils zu den Triebwerken, Rumpfbauteilen und Hitzeschildern alle notwendigen Daten enthalten. Ein solcher Container soll anhand des Containers „Triebwerke“ näher erläutert werden.

Der Container Triebwerk enthält ein Bild des zum Einkauf angebotenen Triebwerks. Innerhalb des Bildes befindet sich ein Hilfebutton, der in Form eines Fragezeichens dargestellt wird. Über diesen Button erscheint ein PopUp-Fenster, das dem Spieler umfassende Informationen zu dem Triebwerk und zur benötigten Anzahl der Triebwerke pro Raumschiff liefert. Rechts neben diesem Bild kann der Spieler die Anzahl

gewünschte Anzahl einzukaufender Triebwerke angeben.

Neben dem aktuellen Marktpreis des Triebwerks, der sich unterhalb der gewünschten Stückzahl befindet, kann der Spieler über einen weiteren Button die Preisentwicklung von Triebwerken auf dem Einkaufsmarkt anschauen. Bei Betätigung des Buttons, öffnet sich ein PopUp, das ein Liniendiagramm der aktuellen Preisentwicklung enthält. Im unteren Teil des Containers Triebwerk wird mittig der Gesamtpreis der Triebwerke bei gewünschter Anzahl angezeigt. Abbildung 23 zeigt den Container Triebwerk.

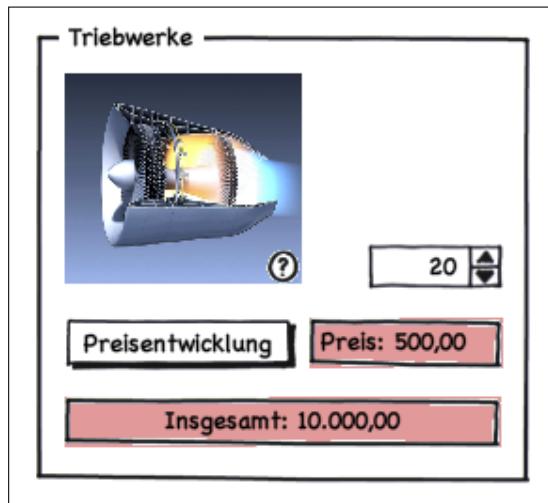


Abbildung 23: Container Triebwerk

In ähnlichen Containern werden auf dem Einkaufsbildschirm auch die Sonderbauteile des jeweiligen Raumschiffes angezeigt. In diesen hat der Spieler jedoch nur die Option, die gewünschte Anzahl der Bauteile anzugeben, sowie den Bauteilpreis einzusehen.

Der gesamte Einkaufspreis wird in roter Schrift rechts neben den Sonderbauteilen dargestellt. Darüber befindet sich ebenso das aktuelle Bankguthaben, sodass der Spieler mit Hinblick auf die aktuelle Finanzsituation des Unternehmens handelt. Hat der Spieler letztendlich die Anzahl aller benötigten Teile ausgewählt, so kann er seinen Einkauf durch einen grünen Button unterhalb des gesamten Einkaufspreises bestätigen. Der gesamte Einkaufsbildschirm wird in dem nachfolgenden Bild gezeigt (s. Abbildung 24).

Ende: Marcel

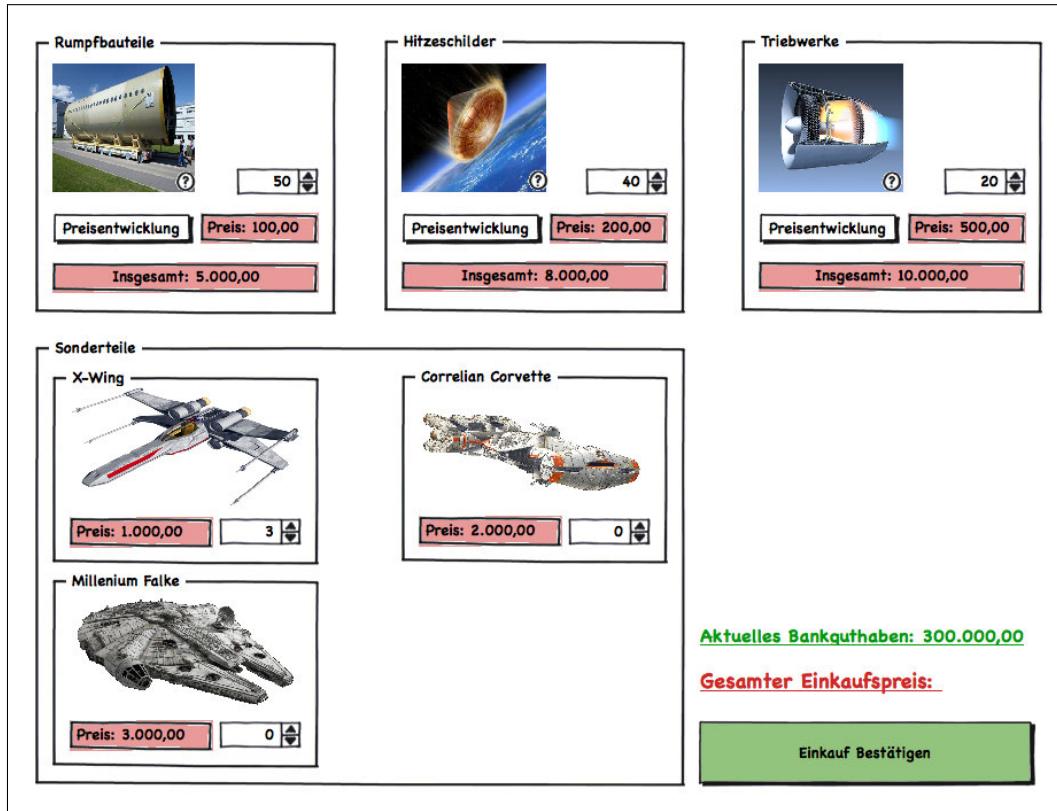


Abbildung 24: Einkauf

4.7 Produktion

Beginn: Julia

Das Produzieren der Raumschiffe wird im Bereich "Produktion" gesteuert. Das Mockup hierzu ist auf Abbildung 25 auf S. 72 zu sehen.

Auch dieser Bildschirm ist in Containern aufgeteilt. Jeder Container enthält Informationen zu einem der drei Raumschifftypen, X Wing, Corellian Corvette und Millenium Falke. Ein Container enthält eine grafische Darstellung des Raumschiff-typs sowie ein Hilfebutton zur näheren Erläuterung des Raumschiffes an der rechten unteren Ecke der Grafik. Betätigt man diesen Button erscheint ein PopUp-Fenster welches die weiteren Informationen enthält. Darunter befindet sich eine Tabelle welche die Anzahl der verschiedenen Bauteile enthält, die für die Produktion eines einzigen Raumschiffes von diesem Typ benötigt werden. Zudem lassen sich darunter die anfallenden Kosten pro Raumschiff ablesen. Über ein Auswahlfeld kann der Spieler die gewünschte Anzahl der zu produzierenden Raumschiffen bestimmen. Daraus

errechnen sich dann die Gesamtkosten für die Produktion der Raumschiffe dieses Typs. Diese Auswahl kann der Spieler für jeden Raumschifftyp treffen.

Auf der rechten Seite des Bildschirmes kann der Spieler die Kosten aller Produktionsaufträge ablesen. Darunter befindet sich der Button “Raumschiffe produzieren” wodurch die im UseCase Diagramm auf Abbildung 8 auf S. 35 genannten Produktionsaufträge angelegt werden. Durch das Betätigen dieses Buttons erscheint ein PopUp-Fenster welches den Vorgang bestätigt.

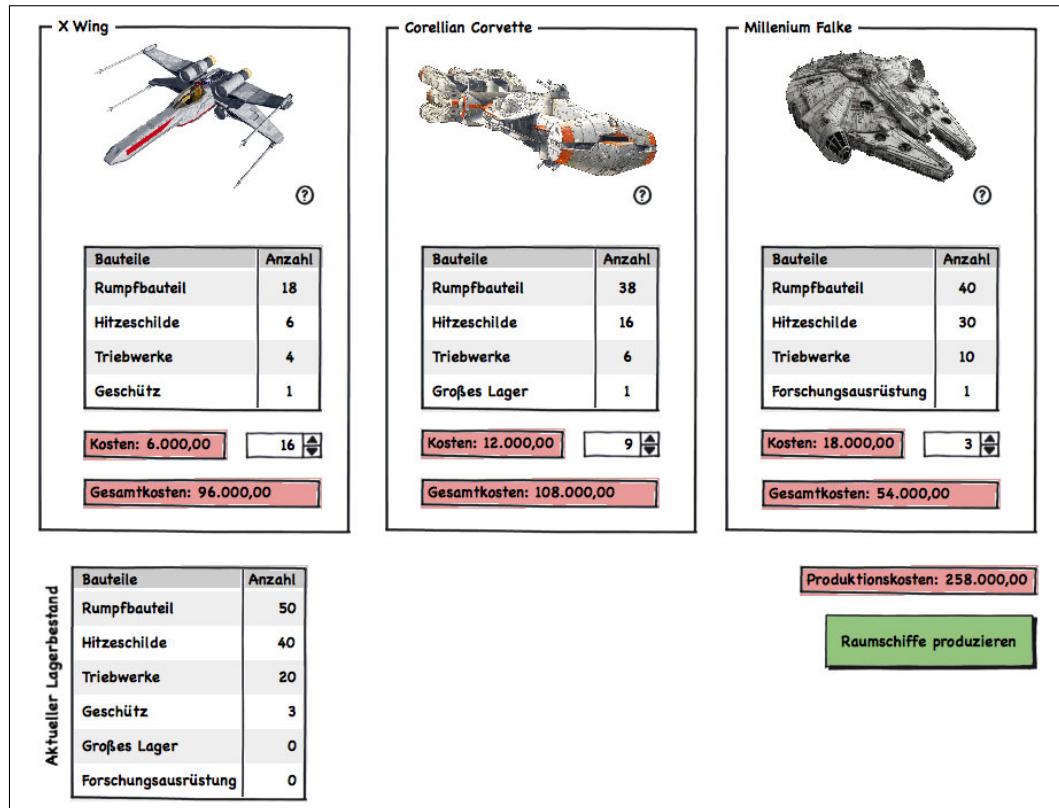


Abbildung 25: Produktion

Ende: Julia

4.8 Verkauf

Beginn: Marcel

Der Verkaufsbildschirm gibt dem Spieler die Möglichkeit seine produzierten Raumschiffe am Absatzmarkt anzubieten.

Der Bildschirm enthält die drei verschiedenen Raumschifftypen. Neben jedem Raumschiff wird dem Spieler die verfügbare Menge, das heißt die in der letzten Periode produzierten oder die im Lager verfügbaren fertigen Raumschiffe, angezeigt. Unterhalb der verfügbaren Menge, werden die gesamten Herstellungskosten des Raumschiffstyps dargestellt.

Aus dem Verkaufspreis, den der Spieler in einem gelb hinterlegten Textfeld eingeben kann und den gesamten Herstellungskosten ergibt sich ein Deckungsbeitrag, den der jeweilige Raumschiffstyp zum Unternehmenserfolg beiträgt. Dieser Deckungsbeitrag wird zwischen den Herstellungskosten und dem festgelegten Verkaufspreis angezeigt.

Hat der Spieler für jeden Raumschiffstyp einen Preis festgelegt, so wird ihm der gesamte Verkaufspreis in einem grün hinterlegten Textfeld angezeigt. Über den Button unterhalb des gesamten Verkaufspreises kann der Spieler seine Preisangebote abgeben und somit die produzierten Raumschiffe am Absatzmarkt anbieten. Nachfolgend wird der Verkaufsbildschirm dargestellt (s. Abbildung 26).

Ende: Marcel

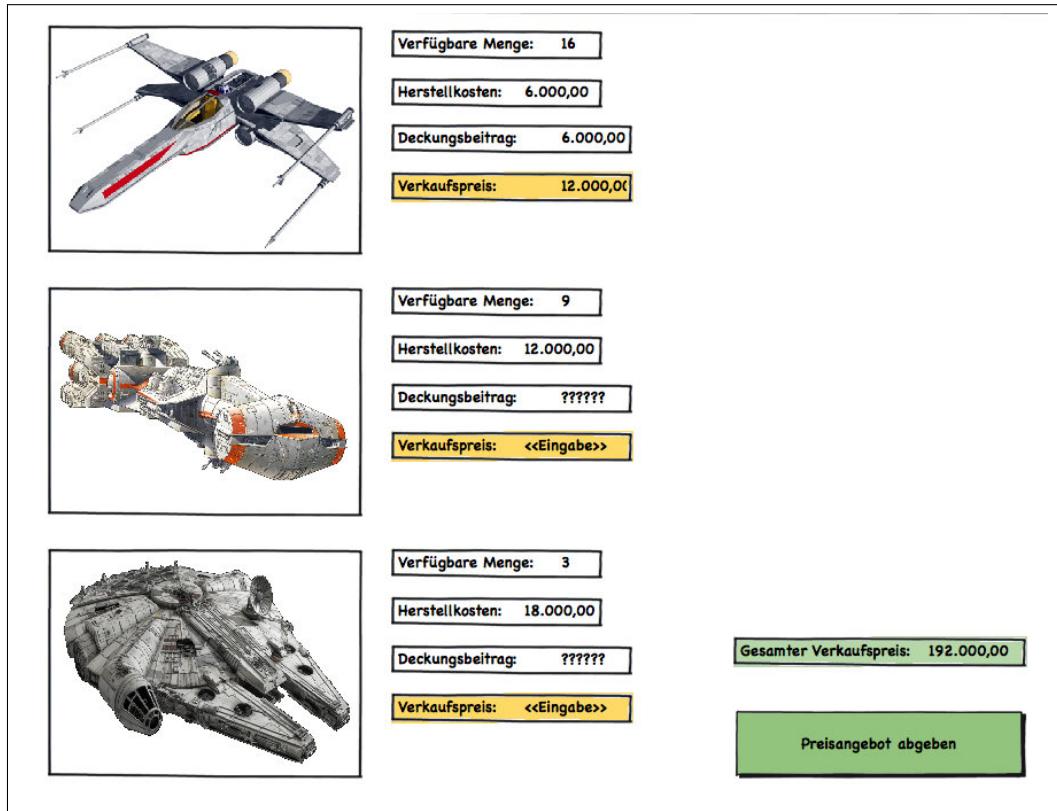


Abbildung 26: Verkauf

4.9 Endbewertung

Beginn: Marcel

Zur Auswertung gelangt der Spieler über einen Button auf der Spielübersicht. In der Auswertung, kann der Spieler Informationen der vergangenen Spielrunden sowie die Gesamtauswertung nach Ablauf der zehn Spielrunden einsehen.

Rundenweise werden die Finanzen des Unternehmens festgehalten. Dazu wird das Bankguthaben am Anfang der Runde in einer Tabelle dargestellt. Die Tabelle enthält ebenso die in der Spielrunde erhaltenen Einnahmen sowie getätigten Ausgaben. Aus diesen Informationen ergibt sich das Bankguthaben am Ende der Runde. Denkbar wäre hier bei einer Realisierung der Benutzeroberfläche, dass der Spieler zusätzlich ein Liniendiagramm, welches das Bankguthaben und die Einnahmen und Ausgaben der Spielrunde enthält, angezeigt bekommt. So kann die Finanzsituation der jeweiligen Runde besser optisch dargestellt werden. Nach jeder Spielrunde wird ebenso der Star der vergangenen Runde festgehalten. Dieser wird anhand eines Bildes zentral

dargestellt. (Abbildung 27)

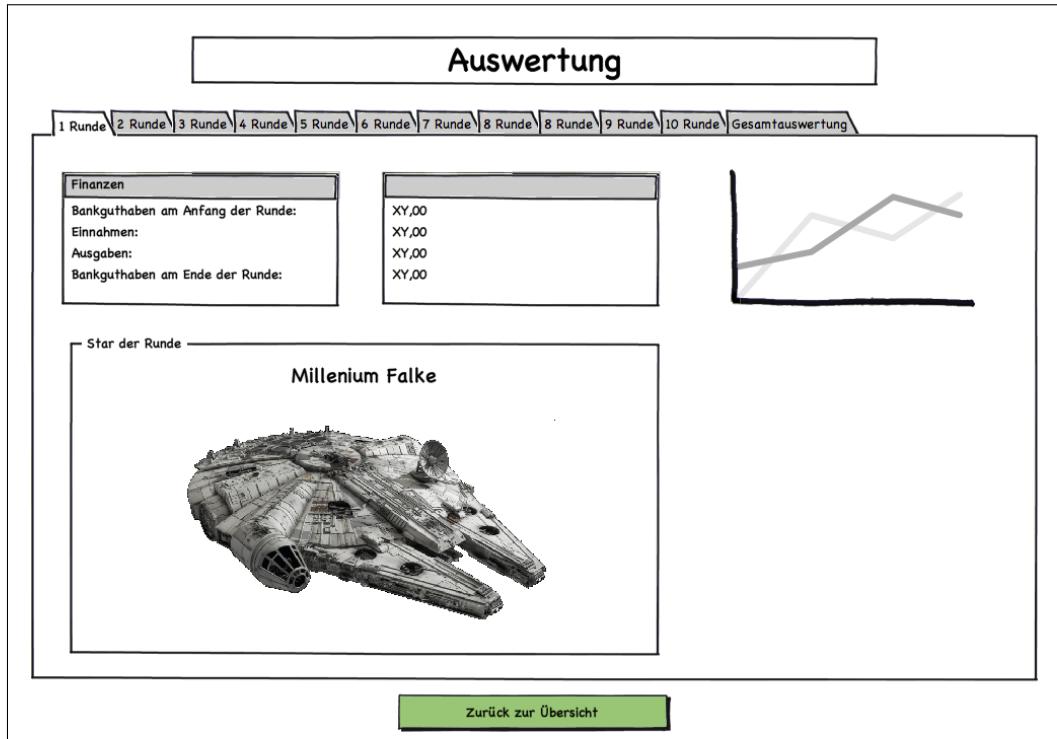


Abbildung 27: Auswertung pro Runde

Die Gesamtauswertung wird nach den zehn Spielrunden ausgegeben. Sie besteht aus einer Tabelle mit den drei Spalten Rang, Unternehmen und Punktzahl. Der Rang ist absteigend sortiert. Dies bedeutet, dass das Unternehmen auf Rang eins das Unternehmensplanspiel gewonnen hat. Die zweite Spalte enthält den Unternehmensnamen. In der letzten Spalte wird letztendlich die Punktzahl angezeigt, die der Spieler über die zehn Spielrunden erreicht hat. Damit die Berechnung der Punktzahl nachvollziehbar ist, enthält der Bildschirm zur Gesamtauswertung den Button „Berechnung der Punktzahl“. Über diesen Button gelangt der Spieler zu einem PopUp. In diesem PopUp wird dem Spieler ausführlich erläutert, aus welchen Faktoren sich die erreichte Punktzahl zusammensetzt (s. Abbildung 28).

Die Gesamtauswertung ist der letzte erscheinende Bildschirm in dem Unternehmensplanspiel Star Greg. Der Spieler hat zwar noch die Möglichkeit alle anderen Bildschirme einzusehen, kann aber keine Aktionen mehr ausführen. Das Spiel wird dann durch ein Schließen des Fensters beendet.

Ende: Marcel

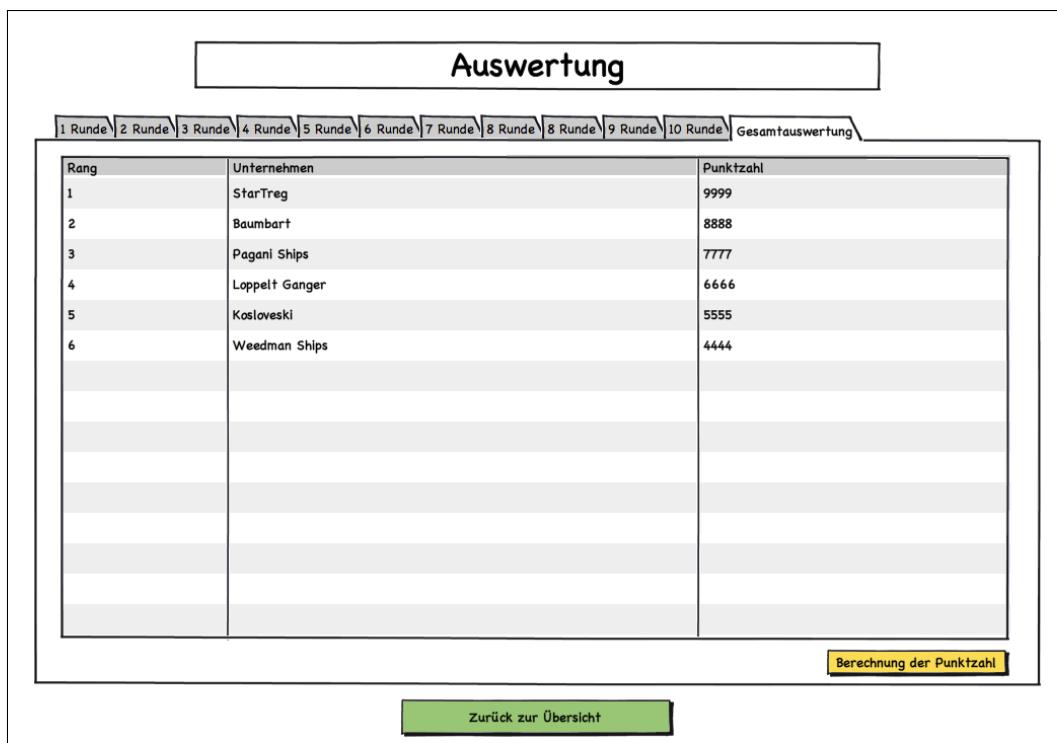


Abbildung 28: Gesamtauswertung

5 jUnit Tests

5.1 Einleitung

Beginn: Fredrik

Eine der zentralen Anforderungen an die Implementierung des Unternehmensplanspiels war die Verwendung von *jUnit* Tests. Diese sollen die Funktionalität des Codes verifizieren und für einen schnellen Überblick über die Funktionsweise dieses Spiels sorgen. Zusätzlich tragen existierende *jUnit* Tests zu einer effizienteren Programmierung bei, da anlehnend an die testgetriebene Entwicklung die Validität jeder Veränderung daran geprüft werden kann, ob festgelegte Endzustände nach wie vor erreicht werden.

Für Star Greg wurden einerseits *jUnit* Tests erstellt, die die kritischen Abschnitte der Spiel-Logik, die nicht als trivial außer Acht gelassen werden können, verifizieren, andererseits gibt ein Test des gesamten Spielablaufs die Möglichkeit nachzuvollziehen, wie ein Spielverlauf aussehen kann, und welche Funktionalitäten auf welche Art und Weise verwendet werden. Diese Testfälle wurden in einem separaten *test*-Paket angelegt und orientieren sich in der Namensgebung an den Klassen, die sie verifizieren sollen. In den folgenden zwei Abschnitten werden diese beiden Felder detaillierter dargestellt.

Ende: Fredrik

5.2 Test der grundlegenden Spiellogik

Beginn: Jan

5.2.1 Test der Klasse BauteilTyp

Beginn: Jan

Es gibt in der Klasse BauteilTyp muss die Methode *berechnePreis()* getestet werden, um sicherzustellen, dass das die richtige Preisberechnung nach einer Spielrunde auch gewährleistet ist.

In der Klasse BauteilTypTest wird hierfür zunächst der Bauteiltyp Rumpfbauteil deklariert, der anschließend in der Methode *setUpBeforeClass()* mit den Werten der Datenbasis initialisiert. In der Testmethode *testBerechnePreis()* wird dann auf dem Rumpfbauteil die zu testende Methode *berechnePreise()* ausgeführt. D die Methode *berechnePreis()* und ein Vergleich mit dem erwarteten Wert folgt (siehe Listing 10).

```
public void testBerechnePreis() {  
    rumpfbauteil.berechnePreis(.5);  
    Assert.assertTrue(Math.round(rumpfbauteil.getPreis())  
        == 130);  
}
```

Listing 10: *testBerechnePreis()* der Klasse BauteilTypTest

5.2.2 Test der Klasse BauteilMarkt

Mit Hilfe der Klasse BauteilMarkt können die Abweichungen der einzelnen Bauteiltypumsätze vom Durchschnittsumsatz errechnet werden. Dies ist notwendig, damit die BauteilTyp-Methode *berechnePreis()* die richtigen Werte nach einer Spielrunde erhält. Die Berechnung des Durchschnittsumsatzes, der Abweichung vom Durchschnittsumsatz und der Aufruf der Methode *berechnePreis()* für jeden Bauteiltyp findet in der Methode *berechnePreise()* statt, die in der Klasse BauteilMarktTest getestet wird.

Zunächst wird ein Objekt des Typs Bauteilmarkt deklariert. Anschließend werden Bauteiltypen deklariert und in der Methode *setUpBeforeClass()* initialisiert. Im Gegensatz zur Klasse BauteilTypTest werden hier allerdings drei verschiedene Bauteiltypen erzeugt, da sonst die oben genannte Abweichung 0 betragen würde. Dem Bauteilmarkt werden die Bauteile in seinem Vektor *typen* hinzugefügt.

In der Testmethode *testBerechnePreise()* werden nun Einkaufstransaktionen hinzugefügt. Anschließend wird die Methode *simuliere()* auf dem Bauteilmarkt ausgeführt, die wiederum selbst darauf die Methode *berechnePreise()* aufruft. Abschlie-

ßend werden die erwarteten Werte der neu berechneten Bauteilpreise mit den tatsächlichen Werten abgeglichen.

```
public void TestBerechnePreise() {
    [...]
    bauteilMarkt.simuliere();
    Assert.assertTrue(Math.round(rumpf.getPreis())==63.00);
    Assert.assertTrue(Math.round(hitzeschild.getPreis())
        ==170.00);
    Assert.assertTrue(Math.round(triebwerk.getPreis())
        ==693.00);
}
```

Listing 11: *testBerechnePreise()* der Klasse BauteilMarktTest

5.2.3 Test der Klasse RaumschiffTyp

In der Klasse RaumschiffTyp muss getestet werden, ob die Kosten und die Lagerplatzseinheiten eines Raumschiffs richtig berechnet werden, da diese sich aus den Kosten der einzelnen Bauteiltypmengen der Raumschiffe ergeben.

Für die Testmethoden werden verschiedene Bauteiltypen und Raumschifftypen deklariert. Anschließend folgt eine Initialisierung in der *setUpBeforeClass()* Methode für beide Typen nach den Werten der Datenbasis.

Die mengenmäßige Zuteilung der Bauteile erfolgt in den Testmethoden selbst. In der Testmethode *testGetKosten()* folgt nach der Zuteilung ein Vergleich der erwarteten und der tatsächlichen Kosten (siehe Listing 12).

```
public void testGetKosten(){
    [...]
    Assert.assertTrue(corellian_corvette.getKosten() == (
        rumpf.getPreis()*38 + hitzeschild.getPreis()*16 +
        triebwerk.getPreis()*6));
}
```

Listing 12: *testGetKosten()* der Klasse RaumschiffTypTest

In der zweiten Testmethode wird getestet, ob die Raumschiffkosten den summierten Lagerkosten der zugehörigen Bauteile entsprechen (siehe Listing 13).

```
public void testGetLagerplatzEinheiten() {
    [...]
    Assert.assertTrue(millenium_falke.
        getLagerplatzEinheiten() == (rumpf.
            getLagerplatzEinheiten()*40 + hitzeschild.
            getLagerplatzEinheiten()*30 + triebwerk.
            getLagerplatzEinheiten()*10));
}
```

Listing 13: *testGetLagerkosten()* der Klasse RaumschiffTypTest

5.2.4 Test der Klasse RaumschiffMarkt

In der Klasse RaumschiffMarkt muss getestet werden, ob die erwartete Nachfrage nach den Raumschiffen und die erwartete Absatzzuteilung den Istwerten nach einer Spielrunde entspricht.

Getestet werden müssen also die Methoden *berechneTypAbsatz()* und *berechneGesamtabsatz()*. Zunächst werden hierfür ein Raumschiffmarkt und mehrere Raumschiffe mit voreingestellten Nachfragewerten deklariert. Auch eine Unternehmensreferenz muss erzeugt werden, da sich der spätere Gesamtabsatz auf mehrere Unternehmen bezieht und die Referenz in der Methode *setUpBeforeClass()* mehreren Unternehmensobjekten zugeordnet wird. Hier wird des Weiteren die Methode *getKosten()* für die einzelnen Raumschiffe überschrieben, da eine Kostenschwankung bei dem Test des Raumschiffmarktes nicht notwendig ist. Es findet ein Hinzufügen der Raumschiffe zu dem Raumschiffmarkt statt. Vor jeder neuen Testmethode werden die Nachfragewerte auf die Anfangswerte gesetzt. Jede Testmethode für sich bezieht sich zudem nur auf die Angebote, die ihn ihr erzeugt werden, da am Ende jeder Testmethode die Methode *simuliere()* auf den Raumschiffmarkt angewendet wird.

Als erstes betrachten wir die Methode *testBerechneAbsatzTyp()* (siehe Listing 14). Hier werden Angebote für ein Raumschiff abgegeben, wonach dazugehörigen Verkäufe mit Hilfe der Methode *berechneTypAbsatz()* ermittelt werden.

Es wird getestet, ob keine Verkaufsmenge kleiner als 0 ist. Aufgrund der Tatsache, dass die Gesamtmenge der abgesetzten Raumschiffe nicht größer sein darf als die Nachfrage nach den Raumschiffen gemessen an der Unternehmensanzahl, findet am Schluss der Methode ein letzter Vergleich statt.

In der zweiten Testmethode *testBerechneGesamtAbsatz()* (siehe Listing 15) werden ebenfalls zuerst Angebotsobjekte erstellt. Im Gegensatz zu *testBerechneTypAbsatz()* geschieht dies allerdings für mehrere Raumschiffe, da hier nicht nur der Absatz eines Raumschiffes relevant ist. Die Angebotstransaktionen werden zum Raumschiff-

```

public void testBerechneTypAbsatz() {
    [...]
    Vector<Verkauf> verkaeufe = raumschiffMarkt.
        berechneTypAbsatz(xwing, angebote);
    for (Verkauf verkauf : verkaeufe) {
        int menge = verkauf.getMenge();
        Assert.assertTrue(menge >= 0);
        gesamtMenge += menge;
    }
    Assert.assertTrue(gesamtMenge <= nachfrageXwing *
        spielWelt.getAnzahlUnternehmen());
}

```

Listing 14: *testGetKosten()* der Klasse RaumschiffTypTest

markt hinzugefügt und die Methode *simuliere()* wird darauf aufgerufen, die selbst die Methode *berechneGesamtAbsatz()* aufruft. Eine Hashmap *map* bekommt zu jedem Raumschifftyp die entsprechenden Verkaufsvektoren mit den Raumschiffverkäufen zugeordnet. Am Ende kann somit überprüft werden, ob alle Raumschiffe in der Hashmap enthalten sind und mindestens einmal abgesetzt wurden.

```

public void testBerechneGesamtAbsatz(){
    [...]
    Assert.assertTrue(map.containsKey(xwing));
    Assert.assertTrue(map.get(xwing).size() > 0);
    Assert.assertTrue(map.containsKey(corvette));
    Assert.assertTrue(map.get(corvette).size() > 0);
}

```

Listing 15: *testGetLagerkosten()* der Klasse RaumschiffTypTest

Ende: Jan

5.3 Test des Spielablaufs

Beginn: Fredrik

Um beispielhaft einen vollständigen Spielablauf nachvollziehen zu können, wurde in der Klasse *SpielTest* die Methode *testGesamtAblauf()* angelegt. Zuerst werden

sämtliche Spieldaten mittels `setUpBeforeClass()` erzeugt, um die Basis für den Spielablauf zu schaffen. In der Test-Methode selbst lässt sich der Ablauf der Runden mitsamt der Phasen, die alle Spieler in diesen durchlaufen, erkennen. Abschließend wird die Bewertung der Unternehmen vorgenommen und ausgegeben.

5.3.1 Unternehmens-Archetypen

Um zu überprüfen, ob die Algorithmen zur Berechnung der Absatzmengen und Bauteilpreise, sowie die Bewertung der Unternehmen sich wie geplant auswirken, wurden verschiedene Archetypen von Unternehmen modelliert, um mit Hilfe dieser ein Testspiel durchzuführen:

- **Föderation:** Dieser Archetyp kennzeichnet sich dadurch aus, dass ihn eigentlich nichts besonders kennzeichnet. Er hält sich an die etablierten Marktstandards, die in der Planung der Datenbasis geschaffen wurden. Hierbei geht er keine besonderen Risiken ein, sondern wählt eine kontrollierte und wohl durchdachte Vorgehensweise. Hierzu zählen die graduelle Aufrüstung seines Personals, um Fehlerkosten zu minimieren und keine punktuelle Belastung durch Aufrüstungs- und Werbungskosten zu erfahren, und eine angemessene Produktion in allen Segmenten des Raumschiffmarkts.
- **Imperium:** Das Imperium hingegen verfolgt eine gänzlich andere Strategie. Durch eine Überproduktion und ein daraus resultierendes Überangebot sollen Konkurrenten vom Markt gedrängt werden. Die Verkaufspreise sollen minimiert werden, sodass andere Unternehmen mit höheren Preisen keinen Absatz mehr finden, während alle Ausgaben auf einem Minimum gehalten werden sollen. Problematisch gestaltet sich für diesen Archetypen vor allem die Wahl desjenigen Preises, der seine Dumping-Strategie optimal unterstützt, ihm aber dennoch ausreichend Gewinn beschert.
- **Rebellen:** Dieser dritte Archetyp macht wiederum alles anders. Statt den Markt zu überschwemmen, sollen tendenziell weniger Raumschiffe produziert werden, die zu einem hohen Preis verkauft werden. Dadurch soll auch eine frühe Personalaufbau finanziert werden, die zu minimalen Fehlerkosten beiträgt.

5.3.2 Spielablauf

Der bekannte Spielablauf zieht sich durch den gesamten Test. So wird zunächst das Spiel gestartet, wodurch indirekt die Daten der ersten Spielrunde geladen wer-

den. Nun dürfen die Unternehmen die ihnen zur Verfügung stehenden Informationen analysieren und Entscheidungen treffen, um abschließend die Runde einzuchecken. Haben dies alle Unternehmen erledigt, so kann die Spielrunde simuliert werden. Dabei stellen die drei oben definierten Unternehmenstypen die Grundlage für die Entscheidungsfindung in den konkreten Spielsituationen dar. Konfrontiert mit einer gegebenen Informationslage wurden aufbauend auf die Strategie des jeweiligen Unternehmens Entscheidungen getroffen. Diese Vorgehensweise wurde für sämtliche Spielrunden wiederholt.

5.3.3 Spielergebnis

Nachdem die zehnte Runde simuliert wurde, wird das Spielergebnis berechnet. Dies manifestiert sich in einer Rangfolge der Unternehmen, die auf einer Bewertung mit Punkten basiert. Dabei ergibt sich die in Tabelle 15 dargestellte Ergebnistabelle, wobei die genaue Anzahl an Punkten im Detail aufgrund der zufällig ausgewerteten Fehlerkosten variieren kann.

Rang	Unternehmen	Punkte
1.	Föderation	39
2.	Rebellen	33
3.	Imperium	27

Tabelle 15: Rangfolge als Spielergebnis

Hier ist klar zu erkennen, dass der Unternehmens-Archetyp, der durch das Unternehmen Föderation symbolisiert wird, mit einem Abstand in Führung liegt, gefolgt von den strategisch hochpreisig situierten Rebellen mit dem Imperium, welches versuchte durch Dumping-Preise zu gewinnen, auf dem letzten Rang. Nun ist es fraglich, ob diese Rangfolge auch dem gewünschten Ergebnis entspricht.

Hierbei ist es sicherlich ein gutes Zeichen, dass dasjenige Unternehmen, welches die für die Spielwelt durch uns empfohlene Strategie umsetzte, gewinnen kann, ohne dass dabei der Grundansatz der Simulation oder der Bewertung angepasst werden musste. Auch im Bezug zur Realität dürfte das Ergebnis als valide erscheinen. Fraglich bleibt lediglich, ob nicht doch ein Spieler mit einer innovativen, vielleicht risikofreudigen Strategie, die Möglichkeit haben sollte, das Feld anzuführen. Gewinnt nämlich lediglich das Unternehmen, welches sich strikt an die Empfehlungen hält, so ist der Lerneffekt des Planspiels als nicht besonders hoch zu bewerten.

Allerdings sollte, nur weil dies das Ergebnis bei den drei willkürlich gewählten Unter-

nehmenstypen ist, nicht sofort angenommen werden, dass eine alternative Strategie nicht auch zum Sieg führen kann. Dies müsste sich im realen Test mit mehreren Spielern zeigen, die wirklich intensiv ihre Strategie durchdenken können, und mit mehr Zeit zur Entscheidung an vielen Stellen sicherlich auch besser fundierte Entscheidungen fällen könnten als hier im Test geschehen. Somit verifiziert auch dieser Test grundsätzlich die Implementierung des Planspiels, ganz davon abgesehen, dass er sicherstellt, dass das gesamte Spiel überhaupt erst fehlerfrei abgeschlossen werden kann.

Ende: Fredrik

6 Fazit und Ausblick

Beginn: Julia, Philipp

6.1 Einleitung

Während der Entwicklung des Planspiels wurden einige Ideen und Möglichkeiten diskutiert, welche den vorgegebenen Zeitrahmen gesprengt hätten. Zum einen wurden erweiterte Funktionen im Spielverlauf besprochen, welche die Komplexität des Spieles gesteigert hätten. Zum anderen lässt ein Blick auf die technischen Aspekte des Planspiels weitere Punkte offen, welche noch realisiert werden könnten. Auf dies und auf die Organisation des Projekts wird im Folgenden genauer eingegangen.

6.2 Erweiterte Funktionen im Spielverlauf

6.2.1 Finanzwesen

Eine Überlegung bezüglich des Finanzwesens war es, anstelle des automatisch eingeräumten Dispositionskredits mit festem Zinssatz dem Spieler eine manuelle Kreditaufnahme anzubieten. Auf diese Weise wäre es dem Spieler möglich vorausschauend den Kreditrahmen zu bestimmen und diesen zu einem angegebenen Zinssatz aufzunehmen. Dieser Zinssatz würde dadurch von der Geldnachfrage abhängig sein und sich von Runde zu Runde verändern. Somit würde dem Spieler mehr Verantwortung und Entscheidungskraft übertragen werden.

6.2.2 Produktion

Mit der Einführung von zusätzlichen Maschinen, welche der Spieler während einer Spielrunde erwerben könnte, wäre das Planspiel realistischer aufgebaut. Durch diese Maschinen könnte die Produktionskapazität weiter erhöht, neue Funktionen in das Raumschiff eingebaut oder der Ausschuss verringert werden. Das zuvor aufgenommene Darlehen könnte beispielsweise zur Finanzierung einer neuen Maschine verwendet werden.

Im Spielverlauf könnte die revolutionäre Erfindung des Warp-Antriebs die Nachfrage erheblich beeinträchtigen. Dieser Antrieb ermöglicht den Raumschiffen mit Warp-Geschwindigkeit das Weltall zu durchqueren. Nach Einführung des Warp-Antriebs würde die Nachfrage nach Raumschiffen mit herkömmlichen Antrieben stark zurück gehen, was den Spieler dazu zwingen würde, die alten Modelle schnellstmöglich abzusetzen. Des Weiteren müsste der Spieler seinen Maschinenpark aufrüsten um der Nachfrage nach den modernen Raumschiffen gerecht zu werden.

6.2.3 Zusatzaufträge

Um noch mehr Dynamik in das Spiel zu bringen, könnte man in bestimmten Runden die Nachfrage nach Raumschiffen durch Zusatzaufträge erhöhen. Diese kann der Spieler nach Prüfung der Rentabilität annehmen oder ablehnen. Auf diese Weise könnte der Spieler freie Kapazitäten nutzen, seinen Gewinn steigern sowie das Image verbessern.

6.2.4 Beeinflussung der Abnehmer

Bemerkt der Spieler einen Rückgang seiner Absatzzahlen so wäre es ihm mit dem so genannten "Macht"-Button möglich, die Entscheidungen der Abnehmer zu seinen Gunsten zu manipulieren. Allerdings würde dann das Risiko bestehen, durch die galaktischen Ordnungshüter für eine Runde auf den Planeten Despayre verbannt zu werden. Während dem Aufenthalt auf Despayre wäre es dem Spieler nicht möglich im Namen seines Unternehmens zu handeln.

Die Einführung einer Marketingabteilung würde dem Spieler noch mehr Möglichkeiten bieten, seine Absatzzahlen zu beeinflussen. Durch Investitionen in die Marktforschung könnte er beispielsweise durch Befragungen Informationen über zukünftige

Nachfrageveränderungen erlangen und könnte somit sein Produktionsprogramm darauf abstimmen. Zudem könnte er durch Werbemaßnahmen seine Beliebtheit erhöhen und somit die Abnehmer legal beeinflussen. Dies könnte zu erheblichen Gewinnsteigerungen führen und würde zudem das Spiel noch dynamischer und realistischer machen.

6.3 Technische Erweiterungen

6.3.1 Benutzeroberfläche

Um das Planspiel benutzerfreundlich zu gestalten, wäre eine intuitive Benutzeroberfläche von Nöten. Diese könnte sich an den bereits vorgestellten Mockups orientieren. Darauf aufbauend könnte sich jeder Spieler selbstständig am System anmelden und zur Startseite gelangen. Würde man zudem die Funktionen des Spieleiters in Java-Code umsetzen, so würde das Spiel diese Aufgaben automatisch übernehmen. Auf diese Weise könnten auch unerfahrene Spieler das Planspiel eigenständig durchführen.

6.3.2 Datenbankanbindung

In der aktuellen Version des Planspiels werden die entstehenden Daten in Java mithilfe von Hashmaps und Vektoren realisiert. Um die Performance und die Übersichtlichkeit des Planspiels zu erhöhen wäre eine Datenbankanbindung sinnvoll. Die zentrale Speicherung auf die Daten würde zudem einen einfacheren Zugriff auf abgespeicherte Daten ermöglichen. Der Hauptvorteil einer Datenbankanbindung würde in der dauerhaften Speicherung der Daten liegen. Auf diese Weise wären die gespeicherten Informationen auch nach dem Schließen des Planspiels abrufbar.

6.4 Projektorganisation

Durch Brainstorming wurden zu Beginn des Projekts viele Ideen zum Planspiel gesammelt. Auf diese Weise wurde dann das Thema “Raumschiffe” ausgewählt. In weiteren Treffen wurde die Idee weiter ausgebaut und mögliche Funktionen diskutiert. Hierbei ging es vor allem um die Vor- und Nachteile, die eine realisierte

Funktion auf das Spiel und den Teilnehmer hätte. Diese wurden im Anschluss in die Kategorien “Pflicht”, “Optional” und “Nicht realisierbar” eingeteilt. Der Schritt war das Erstellen des UseCase beziehungsweise des Klassendiagramms.

Um die Produktivität zu steigern wurden nun Arbeitsgruppen gebildet. Eine Arbeitsgruppe hat sich um das Erstellen der Mockups anhand des UseCase Diagramms gekümmert. Die andere hat das Klassendiagramm in Java-Code umgesetzt. Um die erstellten Klassen zu Testen wurden jUnit Tests angelegt und implementiert. Somit war es möglich die Korrektheit des Spielsablaufs zu testen. Über diesen Zeitraum fanden ständige Treffen zwischen den Arbeitsgruppen statt, um den Fortschritt im Überblick zu behalten und gegebenenfalls Anpassungen vorzunehmen. Auf diese Weise wurde die Aufnahme von optionalen Funktionen besprochen.

Während des Projekts sind einige Dateien angefallen, welche für jedes Mitglied von Bedeutung war. Um den Datentransfer zu vereinfachen und einen ständigen Austausch zu gewährleisten wurde das Versionsverwaltungssystem Git genutzt. Auf diese Weise hatte jedes Mitglied die aktuellste Version der Daten zur Verfügung und konnte die neusten Änderungen direkt einsehen.

Das Verwenden des Textverarbeitungsprogramms L^AT_EX in Kombination mit Git vereinfachte das Erstellen der Seminararbeit. So war es jedem Mitglied möglich sein eigenes Kapitel zu verfassen und dieses in einer eigenen Datei zu speichern. Diese Dateien wurden am Ende mithilfe von L^AT_EX in ein Dokument zusammengeführt.

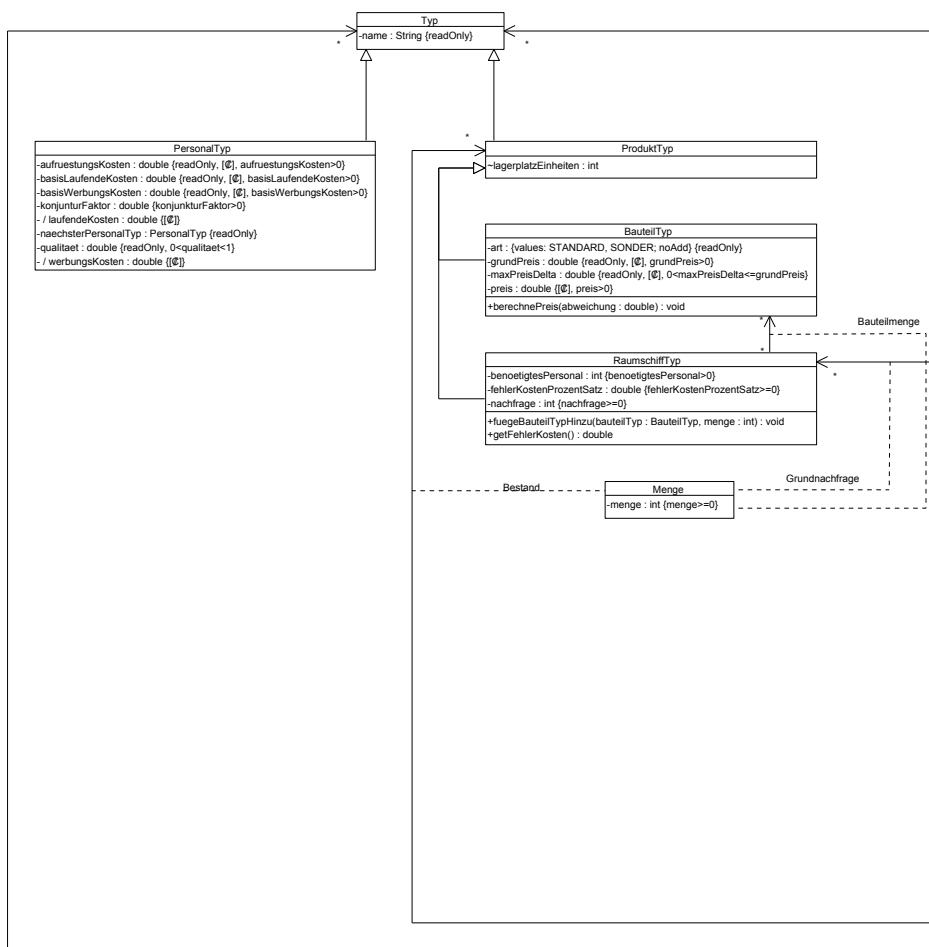
Abschließend ist zu sagen, dass sowohl die Planung als auch die Durchführung des Projektes reibungslos von statthen ging.

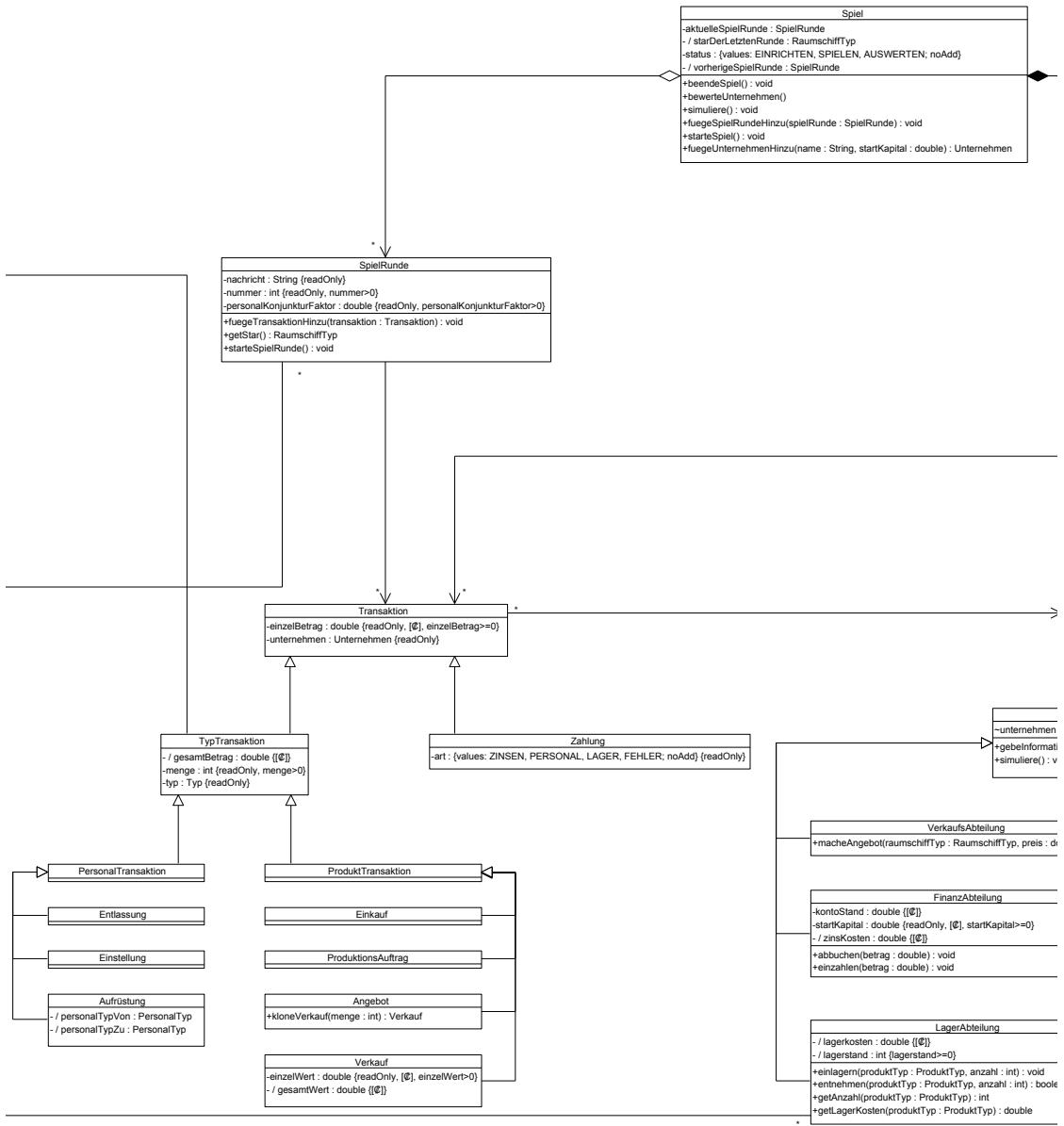
Ende: Julia, Philipp

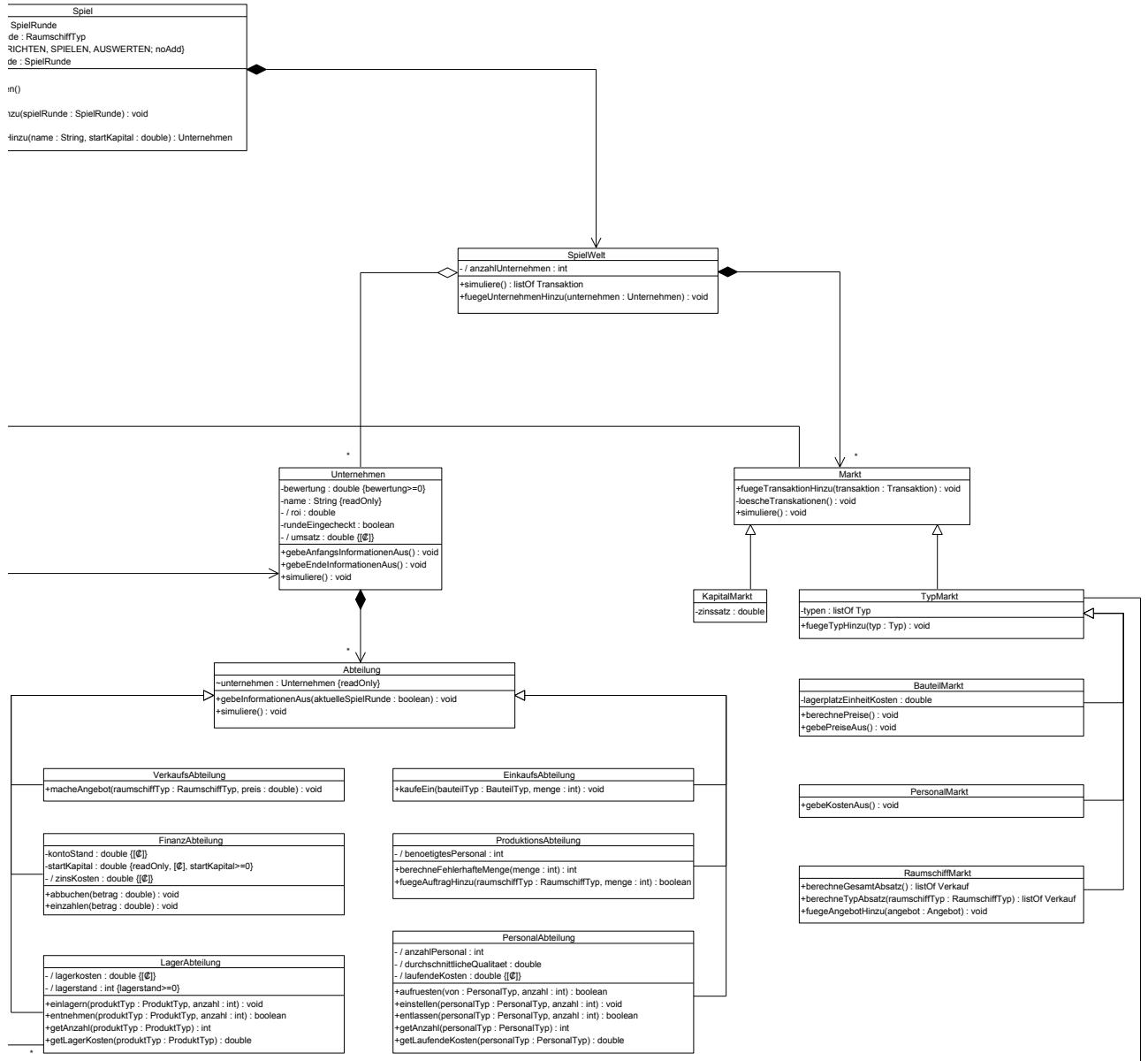
A Anhang

A.1 Klassendiagramm der Entwurfsphase

Aufgrund der Größe des Klassendiagramms wird dieses in drei Teilen dargestellt. Dabei sind diese in ihrer Reihenfolge entsprechend ihrer ursprünglichen Position im Gesamtmodell von links nach rechts zu betrachten.







Literaturverzeichnis

Baum, Ulrich [LOGOS/QU, 1995]: Das qualitäts- und umweltorientierte Logistik-Unternehmensplanspiel LOGOS/QU. 1. Auflage. Aachen: Verlag Shaker, 1995, ISBN 3-8265-5225-3

Fleßa, Steffen [Das computergestützte Planspiel MOSHI, 1996]: Das computergestützte Planspiel MOSHI (Management of Small Hospitals) als strategische Waffe im Kampf für eine effizientere Gesundheitsversorgung in Tansania. 1. Auflage. Nürnberg: Erlangen-Nürnberg: Forschungsgruppe Medizinökonomie am Lehrstuhl für Betriebswirtschaftslehre und Operations-Research der Universität, 1996, ISBN 3-923031-58-0

Gabler Wirtschaftslexikon [Unternehmensplanspiel, 2011]: Unternehmensplanspiel. 2011 <URL: <http://wirtschaftslexikon.gabler.de/Definition/unternehmensplanspiel.html>>

LO Lehrer-Online GmbH [Planspiele zur Bildung für nachhaltige Entwicklung, 2011]: Planspiele zur Bildung für nachhaltige Entwicklung. 2011 <URL: http://www.lehrer-online.de/planspiele-zur-bne.php?show_complete_article=1>

Romann, Daniel-Pierre [Entstehung eines Roadshow-Planspiels, 2010]: Eventmanagement - Entstehung eines Roadshow-Planspiels - Was ist ein Planspiel? Wie entsteht ein Planspiel? Welches sind die besten Einsatzfelder? 1. Auflage. Saarbrücken: VDM Verlag Dr. Müller GmbH & Co.KG, 2010, ISBN 978-3-639-30821-1

TATA Interactive Systems GmbH [Planspiel-Methode, 2011]: Planspiel-Methode. 2011 <URL: http://www.topsim.com/de/ueber_planspiele/methode/>

Ehrenwörtliche Erklärung

„Wir erklären hiermit ehrenwörtlich:

1. dass wir unsere Seminararbeit mit dem Thema

**Star Greg
Das Unternehmensplanspiel**

ohne fremde Hilfe angefertigt haben;

2. dass wir die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Seminararbeit gekennzeichnet haben;
3. dass wir unsere Seminararbeit bei keiner anderen Prüfung vorgelegt haben;
4. dass die eingereichte elektronische Fassung exakt mit der eingereichten schriftlichen Fassung übereinstimmt.

Wir sind uns bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.“

Ort, Datum