

---

# **BACHELORARBEIT**

---

Herr  
**Franz Müller**

**Untersuchungen zur  
Machbarkeit der Entwicklung  
von Games-AAA-Titeln durch  
kleine Teams**

**2017**

# **BACHELORARBEIT**

---

## **Untersuchungen zur Machbarkeit der Entwicklung von Games-AAA-Titeln durch kleine Teams**

Autor:  
**Herr Franz Müller**

Studiengang:  
**Medieninformatik und Interaktives  
Entertainment**

Seminargruppe:  
**MI14w3-B**

Erstprüfer:  
**Prof. Dr.-Ing. Wilfried Schubert**

Zweitprüfer:  
**M. Sc. Dipl.-Inf.(FH) Knut Alltroggen**

Einreichung:  
Mittweida, 18 Dezember 2017

# **BACHELOR THESIS**

---

## **Investigation on the feasibility of developing AAA Games by small teams**

author:

**Mr. Franz Müller**

course of studies:

**Mediainformatics and Interactive  
Entertainment**

seminar group:

**MI14w3-B**

first examiner:

**Prof. Dr.-Ing. Wilfried Schubert**

second examiner:

**M. Sc. Dipl.-Inf.(FH) Knut Alltroggen**

Submission:

Mittweida, December 18 2017

---

## **Bibliografische Angaben**

Nachname, Vorname: Müller, Franz

Thema der Bachelorarbeit: Untersuchungen zur Machbarkeit der Entwicklung von Games-AAA-Titeln durch kleine Teams

Topic of thesis: Investigation on the feasibility of developing AAA Games by small teams

75 Seiten, Hochschule Mittweida, University of Applied Sciences,  
Fakultät Angewandte Computer und Biowissenschaften, Bachelorarbeit, 2017

## **Abstract**

Die Gamesbranche erwirtschaftet jedes Jahr mit AAA Titeln Milliardenumsätze und wächst stetig weiter. Diese wissenschaftliche Arbeit trägt den Titel: Untersuchungen zur Machbarkeit der Entwicklung von Games-AAA-Titeln durch kleine Teams. In der Arbeit wird analysiert ob es möglich ist AAA-Games-Titel mit kleinen Teams umzusetzen. Dafür wird die Entwicklung der Branche analysiert um Ausgangslage und Anforderungen für AAA-Titel zu definieren. Mit diesen Daten werden neue Arbeitsweisen und Technologien untersucht und bewertet. Aus diesen wird dann ein optimales Modell abgeleitet mit dem AAA-Titel mit wenig Kapital und Entwicklern umgesetzt werden kann.

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis .....</b>	<b>V</b>
<b>Glossar.....</b>	<b>VII</b>
<b>Abkürzungsverzeichniss .....</b>	<b>X</b>
<b>Abbildungsverzeichnis .....</b>	<b>XI</b>
<b>1 Einleitung.....</b>	<b>1</b>
<b>2 Stand der Branche .....</b>	<b>3</b>
2.1 Handlungsträger und ihre Abhängigkeiten.....	3
2.2 Auswirkungen der Digitalisierung .....	6
<b>3 Indie und deren Chancen.....</b>	<b>12</b>
3.1 Charakteristik von Indie Studios .....	12
3.2 Bezahlmodelle .....	13
3.3 Lebenszyklen .....	17
3.4 Moderne Engines und deren Ansätze .....	19
3.4.1 Unity .....	20
3.4.2 Unreal Engine .....	21
3.4.3 Vergleich.....	22
<b>4 Prozedurale Inhalte.....</b>	<b>24</b>
4.1 Beispiele .....	24
4.1.1 No Mans Sky .....	24
4.1.2 Star Citizen .....	30
4.2 Vorteile.....	32
4.3 Nachteile.....	33
4.4 Optimale Nutzung für kleine Teams .....	34
<b>5 Neue Ansätze.....</b>	<b>36</b>
5.1 Programmierer und Designer .....	36
5.2 Modulare Systeme .....	37
5.3 Hellblade Independent AAA .....	38
5.3.1 Projektkonzept .....	38
5.3.2 Technische Lösung.....	40

---

<b>6</b>	<b>Das Optimale AAA Indie Modell...</b>	<b>45</b>
6.1	Zielgruppe und Vertrieb .....	45
6.2	Finanzierung .....	46
6.3	Technologie .....	48
6.4	Arbeitsweisen .....	49
6.5	Vermarktung und Transparenz .....	51
6.6	Erfolgsaussichten.....	54
6.7	Chancen für das Konzept.....	56
<b>7</b>	<b>Fazit.....</b>	<b>57</b>
7.1	Zusammenfassung.....	57
7.2	Ausblick.....	58
	<b>Literaturverzeichnis .....</b>	<b>XIII</b>
	<b>Eigenständigkeitserklärung .....</b>	<b>XVII</b>

---

## Glossar

AAA:	Inoffizieller Standard für High End Computerspiele
Addons:	Erweiterung für Computerspiele, wird auch für klassische Software benutzt
Assets:	Digitale Inhalte; in der Spieleentwicklung werden so oft 3D Modelle Texturen oder Codedokumente bezeichnet, aus denen das Spiel erstellt wird (Müller)
Balancing:	Austarieren der Stärken und Schwächen von Spielinhalten
Blue-Ray:	Optisches Datenspeichermedium in Diskformat
Casual Games:	Gelegenheitsspiele ohne größere Tiefe oder Anspruch.
Cartridges:	Speichermodule für Videospiele
Community:	Gemeinschaft
Crowdfunding:	Finanzieren eines Projektes durch Kapital einer großen Gruppe von Investoren, die jeweils kleine Beträge beisteuern
C++:	C++ ist eine objektorientierte Programmiersprache, die trotz des hohen Abstraktionsniveaus sehr maschinennah und effizient ist (Müller 2017)
C#:	Von Microsoft entwickelte objektorientierte Allzweck-Programmiersprache.
Dev Diary:	Ein Entwicklertagebuch
Direct3D:	3D API von Microsoft
Early Access:	Zugang zu einem Computerspiel in seinem vollen derzeitigen Umfang, während es sich noch in Alpha oder Betaphase befindet
Gameobject:	Instanz eines Objektes in einer Szene der Unity Engine
Games as a Service:	Spiele, die über eine Internetverbindung mit neuen Inhalten versorgt werden und die Spieler langfristig binden sollen.

---

InApp Käufe:	Einkäufe, die direkt aus der Anwendung getätigt werden wie Microtransaktionen
Indie:	Steht für Independent (unabhängig); wird oft benutzt, um die Unabhängigkeit von Publishern zu bezeichnen
Indie Entwickler:	Von einem Publisher unabhängige Entwickler
Java:	Eine von Sun Microsystems entwickelte objektorientierte Programmiersprache
Konsole:	Ein für das Spielen von Computerspielen dedizierter Computer
Lootbox:	Eine mit zufälligen Ingame Gegenständen gefüllte Box; kann als Microtransaktion erworben werden
Microtransaktions:	Digitale Güter, die über kleine Beträge gekauft werden
Nodes:	Knoten, mit denen im Visual Scripting programmiert wird; beinhalten Variablen, Funktionen und logische Operatoren
OpenGL:	Open Source 2D und 3D API
Patch:	Korrektur der Software
Permadeath:	Wenn der Spieler in einem Computerspiel scheitert, die Spielfigur zum Beispiel stirbt, ist das Spiel vorbei und kein Fortschritt wird gespeichert
Playstation:	Konsole vom japanischen Hersteller Sony
Polygon:	Vieleck, welches in Computerspielen zur Darstellung dreidimensionaler Objekte genutzt wird; meist dreieckig
Prefabs:	Sammlung vordefinierter verbundener Assets
Postprocessing:	Nachbearbeiten von Bildern (Müller 2017)
Publisher:	Ein Verleger für Computerspiele oder Software allgemein
Quellcode:	auch: Sourcecode; besteht aus Anweisungen in einer Programmiersprache, die für Menschen lesbar sind (Müller 2017)
Raspberry Pie:	Minicomputer



---

Rougelike:	Ein Computerrollenspiel in prozedural generierten Level, in dem der Tod der Spielfigur permanent ist
Season Pass:	Ein digitales Gut, das dem Inhaber über einen bestimmten Zeitraum Zugang zu allen Addons eines Computerspiels erlaubt
Seed:	Startwert für prozedurale Generatoren und oder Zufallsgeneratoren
Shared World:	Die persistente Online-Spielwelt ist für jeden Spieler die gleiche und wird von Spielern beeinflusst
Steam:	Onlineplattform für den Verkauf von PC Spielen von Valve
Strechgoals:	Weiterführende Ziele einer Crowdfunding Kampagne; oft mit weiteren Anforderungen an das Projekt verbunden
Trackingpoint:	Markierung, die eine Kamera oder Sensor verfolgen kann, um Daten zu sammeln
Visual Scripting:	Programmieren durch das Verknüpfen von visuellen Nodes
Voicactor:	Synchronsprecher.
Vollpreistitel:	Computerspiele mit einem Verkaufspreis von 40€ und mehr
Xbox:	Konsole vom US-amerikanischen Hersteller Microsoft

---

## Abkürzungsverzeichnis

AI:	Artificial Intelligence
API:	Application programming interface
BIU:	Bundesverband Interaktive Unterhaltungssoftware
CD:	Compact Disc
DLC:	Downloadable Content
DVD:	Digital Versatile Disc
IDE:	Integrated development environment
MMORPG:	Massively Multiplayer Online Role-Playing Game
PC:	Personal Computer oder Einzelplatzrechner
PR:	Public Relations
PVP:	Player versus Player
WOW:	Word of Warcraft
2D:	Zweidimensional
3D:	Dreidimensional

## Abbildungsverzeichnis

Abbildung 1 Einzelhandelsabsatz der meistverkauften Computer- und Videospiele weltweit im Jahr 2015 (VGChartz 2016) .....	5
Abbildung 2 Anteil der Marktsegmente an den Weltweiten Umsätzen im Gamesmarkt im Jahr 2013 und Prognosen bis 2017 (Horizont 2014) .....	6
Abbildung 3 Prognosen weltweite Umsätze im Markt für Digitale Games in den Jahren 2015 bis 2021(Statista 2017).....	7
Abbildung 4 Umsatz mit kostenpflichtigen virtuellen Zusatzinhalten für Computer und Videospiele in Deutschland von 2008 bis 2016 (BIU 2016) .....	9
Abbildung 5 Umsatz von Activision Blizzard in den Jahren 2005 bis 2016 (Activision Blizzard 2017) .....	10
Abbildung 6 Prognose der Umsätze Social/Casual Games in Deutschland von 2008 bis 2020(PwC 2017) .....	14
Abbildung 7 Anzahl der Abonnenten von World of Warcraft weltweit vom 1.Quartal 2005 bis zum 3. Quartal 2015 (Activision Blizzard 2015) .....	15
Abbildung 8 Gamestop A Aktie von September 2014 bis März 2017 (Börse Frankfurt 2017).....	16
Abbildung 9 Umsatz des Computerspielehändlers Gamestop in den Geschäftsjahren 2007 bis 2016 (Gamestop 2016) .....	17
Abbildung 10 Unterschiede des Early Access Modells .....	18
Abbildung 11 Engine Features .....	22
Abbildung 12 Gameplayschleife in No Mans Sky .....	25
Abbildung 13 No Mans Sky. Planet in niedrigster Detailstufe .....	26
Abbildung 14 No Mans Sky. Annäherung an Planeten .....	26
Abbildung 15 No Mans Sky. Eintritt in Planetenatmosphäre .....	27
Abbildung 16 No Mans Sky. Annäherung Planetenoberfläche.....	28
Abbildung 17 No Mans Sky. Landung auf der Planetenoberfläche .....	29
Abbildung 18 Star Citizen Stadtgenerator(Star Citizen 2017) .....	31

---

Abbildung 19 Star Citizen Generierte Stadt (Star Citizen 2017).....	32
Abbildung 20 Unreal Engine Visual Scripting.....	37
Abbildung 21 Hellblade Budget(Antoniades Tameem, 2014).....	39
Abbildung 22 Independent AAA Model (Antoniades Tameem Ltd 2014) .....	40
Abbildung 23 Hellblade Motioncapture mit Bonita Kamera (Ninja Theory Ltd. 2015a) .	41
Abbildung 24 Hellblade Performancecapture (Ninja Theory Ltd. 2015b).....	42
Abbildung 25 Hellblade Plant-Pot Scanner (Ninja Theory Ltd. 2015a) .....	43
Abbildung 26 Hellblade Nomralmap Scanns (Ninja Theory Ltd. 2015a).....	44
Abbildung 27 Weltweit durch Crowdfunding eingesammeltes Kapital von 2011 bis 2015 (CrowdExpert 2015) .....	47
Abbildung 28 BOUND Spielgrafik.....	50
Abbildung 29 Welche Social Media Plattform ist für ihr Unternehmen am wichtigsten.(Social Media Examiner (2017) .....	52
Abbildung 30 Hellblade Budget(Antoniades Tameem, 2014).....	54
Abbildung 31 Hellblade Sales(Ninja Theory Ltd. 2017a).....	55

# 1 Einleitung

Videospiele haben als Unterhaltungsmedium ein unerreichtes Potenzial, da sie viele andere Medien verbinden mit dem einen großen Unterschied, dass der Spieler tatsächlich in das Geschehen eingreifen kann. Videospiele sind heutzutage nicht mehr nur etwas für „Computer Freaks“. Sie sind in großen Teilen der Gesellschaft anerkannt und werden auch über die Verwendung zur bloßen Unterhaltung hinaus genutzt. Sie bieten einzigartige Chancen, wie etwa einen Planeten in der digitalen Welt zu finden, den vor seiner Entdeckung durch den Spieler niemand je zu Gesicht bekommen hat - nicht einmal die Entwickler selbst. Möglich ist das etwa im 2016 erschienenen „No Man`s Sky“ von Hello Games. Hier kann man 18 Trillionen prozedural generierte Planeten besuchen mit eigener ebenfalls prozedural generierten Fauna und Flora. Das ist in keinem anderen Medium möglich und macht Videospiele damit zu dem Medium mit dem größten Potenzial für die Zukunft. Erschaffen wurde dieses Universum von einem verhältnismäßig kleinen Entwicklerteam. Doch solch große Projekte von so kleinen Teams sind die Ausnahme, so wie im Fall von No Mans Sky, welches nur durch komplexe Mathematik von Seiten der Entwickler zu stemmen war. Und doch hatte auch No Mans Sky mit einigen Problemen von technischer Seite zu kämpfen, verstärkt durch die Erwartungen der Spieler. Denn die Branche wird eigentlich von riesigen Entwicklerteams mit noch größeren Projekten regiert. Für die Entwicklung solch großer Projekte werden normalerweise gigantische Geldmittel und hunderte Entwickler aufgebracht. (vgl. Antoniadou Tameem, 2014) Doch hier stellt sich die Frage, wie demnach ein kleines Team 18 Trillionen Planeten kann? Inwiefern und mit welchen Mitteln kann ein solches Studio digitale Spiele im AAA Standard realisieren? Diese Frage soll in dieser Arbeit behandelt werden, indem durch eine Auswertung relevanter Faktoren ein möglichst optimales Modell erstellt wird, mit dem kleine Teams große, hochqualitative Projekte umsetzen können.

Um diese Fragen zu beantworten, muss zuerst jedoch die derzeitige Situation der Branche betrachtet werden, welche als Handlungsumfeld für den Forschungsgegenstand fungiert. Die Entwicklungen, die zum aktuellen Verhältnis der agierenden Instanzen sowie ihren Vorgehensweisen geführt haben, sind ausschlaggebend für das Aufsetzen des Modells, da sie zunächst den Handlungsspielraum für kleinere Studios ermöglichten. Effektiv handlungsfähig werden sie jedoch erst durch die Wahl einer geeigneten Software, da sie die Hauptebene der eigentlichen Umsetzung darstellt. Aufgrund der fehlenden Kapazitäten kleinerer Studios kann bei der Generierung der Spielinhalte nicht auf herkömmliche Methoden wie die der großen Entwickler zurückgegriffen werden. Daher werden neuartige Verfahren, wie prozedurale Generierung, erforscht und bewertet, um das Modell weiter zu optimieren. Hierbei werden zwei unterschiedliche Ansätze anhand von „No Mans Sky“ und „Star Citizen“ zur Benutzung von prozeduralen Techniken analysiert,

um die optimale Herangehensweise zu finden. Es wird zudem die Entwicklung und Arbeitsweisen an dem Projekt „Hellblade“ von Ninja Theory analysiert, das bisher als einziges Studio das hier angestrebten Modell verfolgt und vollständig umgesetzt hat. Mit diesen Informationen kann dann ein optimales Modell erstellt werden, welches gleichzeitig als Leitfaden dienen soll, indem es Voraussetzungen und Ansätze zur einer vergleichsweise kostengünstigen Umsetzung durch wenige Entwickler bereitstellt. Zum Ende soll eine zusammenfassende Bewertung die Erfolgsaussichten in diesem neuen Modell zeigen. Es wird jedoch nicht die Entwicklung von Mobilegames analysiert, da diese dem AAA Standard nicht gerecht werden können. Primär wird die technische Umsetzung und nicht die wirtschaftliche Planung betrachtet, sodass in Folge dessen keine finanzielle Planung oder Businesspläne erwartet werden können.

## 2 Stand der Branche

### 2.1 Handlungsträger und ihre Abhängigkeiten

Um die Zusammenhänge, die zu der Möglichkeit geführt haben, Titel in AAA Qualität unabhängig umzusetzen, müssen die Handlungsträger analysiert werden. AAA steht für den höchsten erreichbaren Qualitätsstandard und Größe von modernen Videospielen. Für diese Spiele werden Millionen und riesige Teams aufgewendet, welche in den allermeisten Fällen von sogenannten Publishern oder von den Konsolenherstellern selbst finanziert werden. Publisher fungieren im klassischen Sinne als Verleger für Computerspiele, da für die Projekte ebenfalls immense Mittel für Werbung und Promotion aufgewendet werden. Bei den Konsolenherstellern entwickeln diese Spiele First- oder Third-Party Entwickler. First-Party Entwickler sind Entwicklerstudios, die den Konsolenherstellern selbst gehören und exklusiv für diese entwickeln. Third-Party Entwickler sind Studios, die nicht im Besitz der Konsolenhersteller sind, aber für diese Computerspiele entwickeln. Es wird oft auch der Begriff Second-Party Entwickler für Studios verwendet, die zwar exklusiv für die Konsolenhersteller Spiele entwickeln, aber nicht diesen gehören. Da diese Studios im Allgemeinen unter die Third-Party Entwickler fallen, ist der Begriff jedoch nicht gebräuchlich. Da die meisten Entwicklerstudios das Kapital für die Entwicklung nicht selbst aufbringen können, sind sie auf die Geldmittel der Publisher oder Konsolenhersteller angewiesen, da das Erstellen technisch moderner und anspruchsvoller Videospiele Beträge im dreistelligen Millionenbereich erfordert. Am Ende der Entwicklung sind es aber oft auch die Publisher, die den Großteil der Gewinne einstreichen. Den Studios wird im Normalfall ein fester Betrag ausgestellt, den sie für die Entwicklung des Spiels und die Bezahlung ihrer Angestellten aufwenden können. In den allermeisten Verträgen zwischen Entwickler und Publisher gibt es eine Royalty Klausel, die nicht die allgemeine Nutzung des geistigen Eigentums im klassischen Sinne bezeichnet, sondern die Beteiligung am Gewinn, sollte das Produkt mehr einspielen als erwartet. Dieses Geschäftsmodell war über viele Jahre hinweg der Normalfall.

*„What is unusual is just how long and steady the last few generations has been from PS1 to PS3. In that time, the retail model has dominated with a high fixed pricing model forcing games to compete on size and features becoming what is now known as AAA.“*  
(Antoniades Tameem, 2014)

Solch eine Stetigkeit in der Branche ist sehr ungewöhnlich, wie Tameem Antoniades schon 2014 anmerkte, und hat zum Beispiel zu dem hohen Preisniveau geführt, das wir heute für AAA „Vollpreistitel“ kennen. Daher haben sich bis heute auch einige Milliarden-schwere Publisher entwickelt, die den Markt beherrschen. Durch diese Stetigkeit und die starken Publisher ist es erst zu der jetzigen Situation gekommen, bei der viele Publisher

kaum kreative Risiken eingehen und neue Spiele oft nach den gängigsten Themen konzipiert. Solche Titel sind dabei oft nach einem ähnlichen Schema aufgebaut, um möglichst viel und vor allem sicheren Gewinn zu erwirtschaften, da sie sich bisher als lukrativ erwiesen.

*„In 2010 two movies came out that made the same amount of money. Black Swan and Clash of the Titans. One was creatively risky but low budget and made \$327m post budget and one was big and dumb and made \$368m. I asked one publisher outright which kind of bet they were more comfortable with. It wasn't Black Swan“ (Antoniades Tameem, 2014).*

Am Ende ist solch ein Publisher ein gewinnorientiertes Unternehmen, in dem die Aktionäre, die das Kapital verantworten, auch das letzte Wort haben. Viel entscheidender als der Spieltitel selbst sind vor allem Tabellen und Analysen der Verkaufszahlen, um abschätzen zu können, mit welcher Wahrscheinlichkeit und in welchem Maße sich ein Titel verkaufen lässt. Diese sicheren Gewinne sind angestrebt, weswegen Publisher meist eigene Studios vorweisen, die ausschließlich für sie entwickeln und beispielsweise für eine spezifische Marke des Publishers verantwortlich sind. Die großen Publisher besitzen meist AAA Marken, wie etwa Call of Duty oder Battlefield, die regelmäßig oder sogar jährlich neue Ableger bekommen.



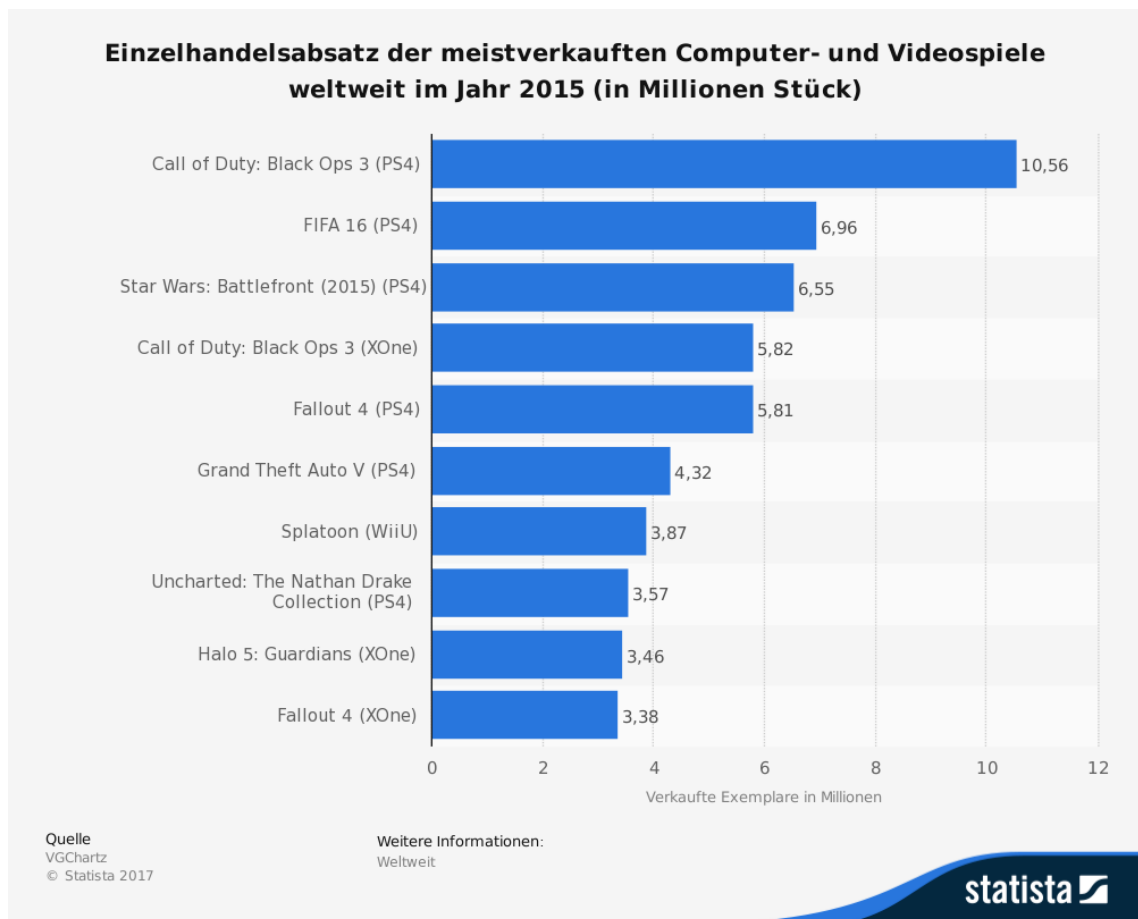


Abbildung 1 Einzelhandelsabsatz der meistverkauften Computer- und Videospiele weltweit im Jahr 2015 (VGChartz 2016)

Wie man in der Statistik von 2015 sieht, sind es auch oft diese jährlichen Titel, die die größten Gewinne erzielen. Doch der Jährliche Zyklus hat in den vergangenen Jahren starke Kritik erhalten, da in den Augen der Spieler keine Weiterentwicklung innerhalb der Spiele erfolgte. Activision Blizzard, einer der größten Publisher, hat daher die Entwicklung der Call of Duty Reihe auf Infinity Ward, Treyarch und Sledgehammer Games aufgeteilt, die sich mit den Teilen jedes Jahr abwechseln. Somit hat jedes Studio mehr Zeit für die Entwicklung ihrer einzelnen Titel. Diese Studios sind keine kleinen Studios; sie haben selbst hunderte Mitarbeiter und Entwickler. Für dieses Vorgehen wendet Activision Blizzard erfolgreich enormes Kapital auf. Diese Entwicklung ist auch darauf zurückzuführen, dass diese Jährlichen Titeln oft einen Online Modus bieten, in dem die Spieler gegeneinander antreten, welcher allerdings nur durch die zunehmende Digitalisierung und Vernetzung möglich ist. Diese sorgt aber wie im folgenden Kapitel beschrieben in ihrer weiteren Ausprägung für weitere tiefgreifende Veränderung.

## 2.2 Auswirkungen der Digitalisierung

Die zunehmende Globalisierung und Vernetzung verändert auch die Gamesbranche. Man benötigt keine Konsole oder leistungsstarken Computer mehr um Videospiele zu spielen. Jeder hat die Möglichkeit dazu schon in seiner Tasche. Moderne Smartphones sind in der Lage, komplexe 3D Umgebungen in Echtzeit darzustellen.

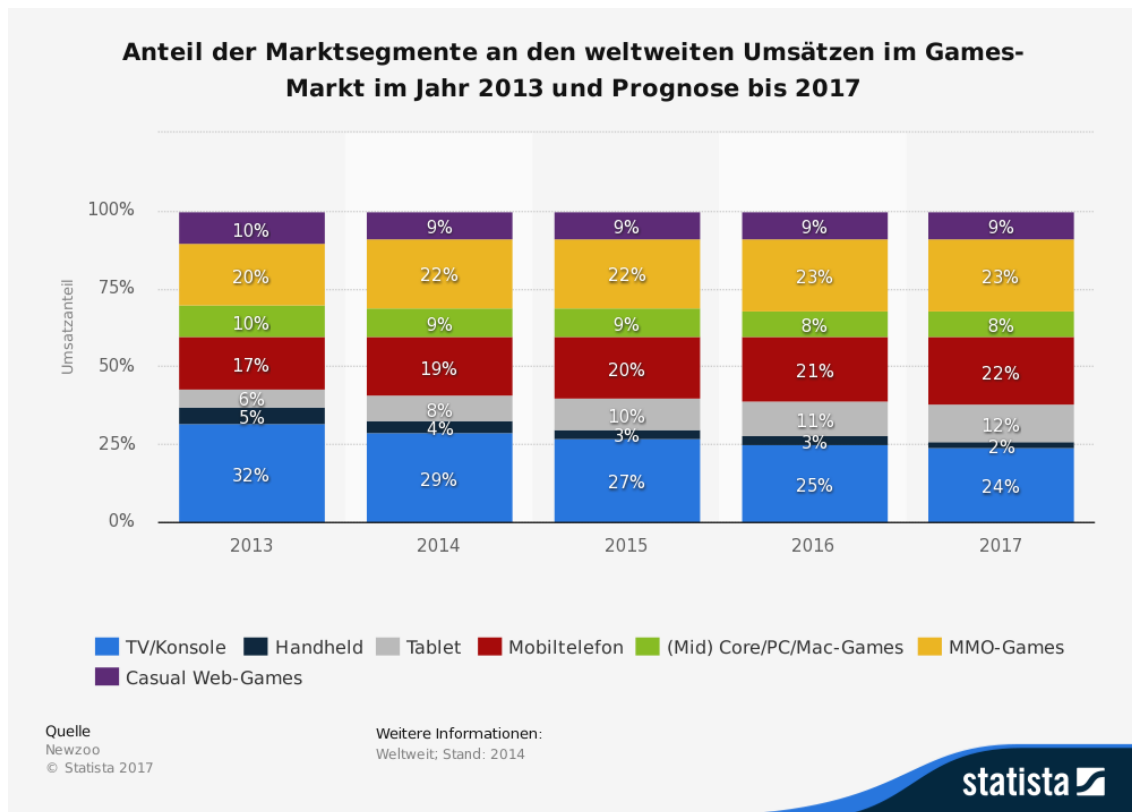


Abbildung 2 Anteil der Marktsegmente an den Weltweiten Umsätzen im Gamesmarkt im Jahr 2013 und Prognosen bis 2017 (Horizont 2014)

Das Marktsegment für Mobile Spiele auf Tablets sowie Smartphones vergrößert sich, wie in der Grafik zu erkennen ist, jedes Jahr auf Kosten der klassischen Plattformen. Die klassischen Plattformen umfassen die Heimkonsolen in Ihrer derzeitigen Iteration, mit Playstation 4, Xbox One, und der PC Plattform. Diese Entwicklung wird mit den immer stärkeren mobilen Endgeräten weiter andauern. In dieser Arbeit soll jedoch es vorrangig um Spiele für die letztere gehen. Die Anteile der klassischen Plattformen werden zwar im Verhältnis kleiner, wachsen jedoch in totalen Zahlen. Die Entwicklung hin zur Vernetzung ist somit nicht zwangsläufig schädlich für klassischen Plattformen. Im Gegenteil hat die zunehmende Vernetzung auch dazu geführt, dass neue Arten des Vertriebs von Videospiele entstehen konnten. Zuvor wurden Videospiele entwickelt, abgeschlossen und dann auf sogenannten „Cartridges“ gespeichert, welche dann in einem Geschäft an

den Spieler verkauft wurden. Zu dieser Zeit konnten Spiele nicht über eine Internetverbindung „gepoched“ werden, was so viel bedeutet, wie das Korrigieren von verbleibenden Fehlern. Somit verblieb auch jeder Fehler des Spiels auf dem Speichermedium. Später dann wurden sie auf CDs, DVDs und letztendlich auf Blue-Rays gespeichert. Jedes dieser Medien bot immer mehr Speicherplatz und Lesegeschwindigkeit. Aber nicht nur die Medien wurden größer; auch die Spiele selbst setzten immer größeren Speicherplatz voraus. Mit der Internetfähigkeit von Konsolen und PCs besteht jetzt eine weitere Methode, welche auf dem Vormarsch ist - das digitale Kaufen und Beziehen von Videospielen.

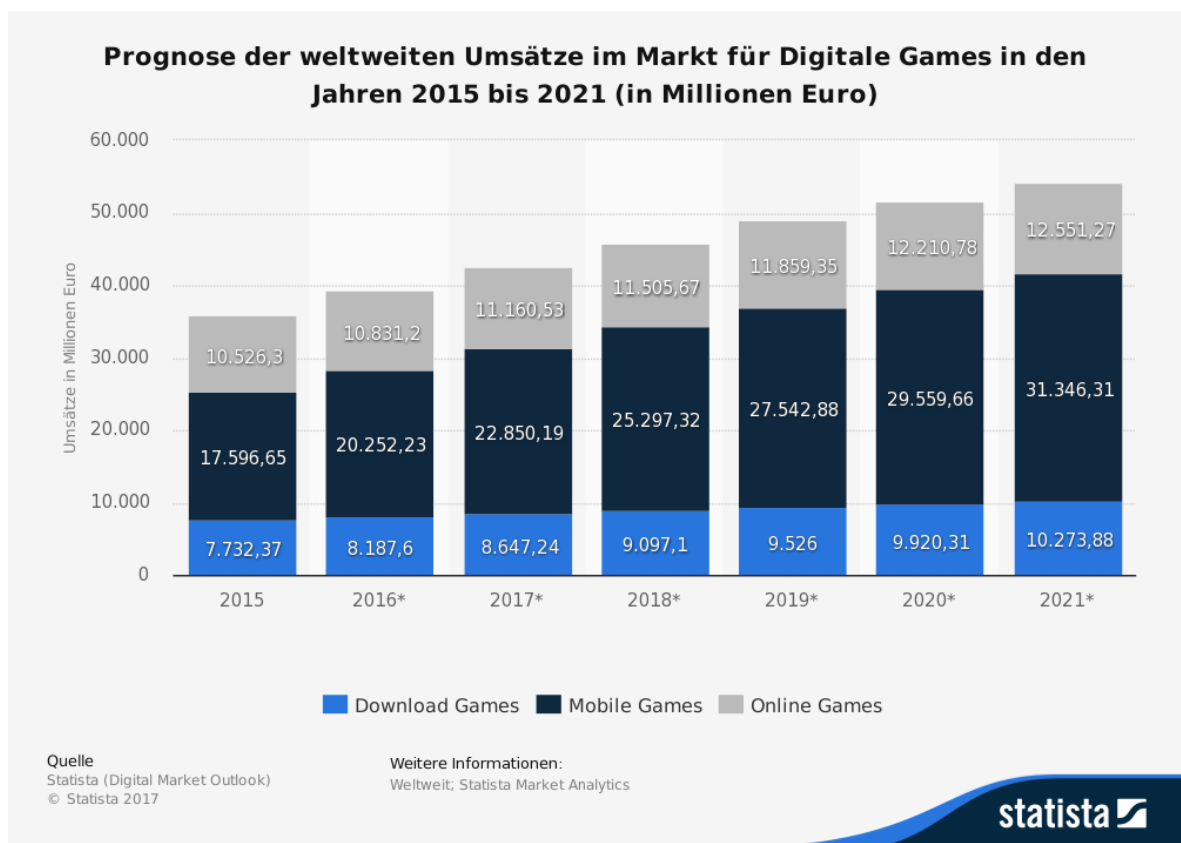


Abbildung 3 Prognosen weltweite Umsätze im Markt für Digitale Games in den Jahren 2015 bis 2021 (Statista 2017)

Auch wenn mobile Spiele ein ungemeines Potenzial aufzeigen, stehen Spiele, die durch den PS Store oder die Plattform Steam erwerblich sind, hoch im Kurs. Diese sogenannten Downloadgames stehen dem Spieler nach dem Kauf über ihr Konto des jeweiligen Anbieters zum Download bereit. Derartig erworbenen Spiele können allerdings nicht so einfach wie ein Spiel auf einer Disk getauscht oder wiederverkauft werden, da sie an das Benutzerkonto gebunden sind. Die Bequemlichkeit, die das Online-Kaufen und -Beziehen der Spiele mit sich bringt, überzeugt jedoch immer mehr Spieler, wie in der Grafik

zu sehen ist. Dieser digitale Umschwung wird enorme Auswirkungen auf den Gamesmarkt haben, auch wenn es in den nächsten Jahren noch Widerstand gegen das „alles digital“-Konzept geben wird. Und in dieser neuen Situation, in der keine Disks mehr gedruckt und in die Läden gebracht werden müssen, müssen auch die Publisher ihre neue Rolle finden. Denn durch diese Veränderung verlieren sie ihre ursprüngliche Daseinsberechtigung als Produktions- und Vertriebskette. Die neuen Aufgaben der Publisher wären in dieser Form des Marktes nicht weniger wichtig, besonders hinsichtlich des Bereitstellens des Kapitals und der Online-Infrastruktur wie Server und PR sowie der direkte Kontakt zu den Spielern. Das digitale Verkaufen von Spielen sorgt auch für ein anderes Preisniveau; die Anbieter der großen Onlineplattformen locken regelmäßig mit großen Rabattaktionen. So ist für viele PC Spieler der jährlich mehrfach auftretende Steam Sale die Hauptbezugsquelle für Spiele. Zu diesem Zeitpunkt können viele Spiele zu sehr niedrigen Preisen digital gekauft werden. Herbeigeführt wurde diese Entwicklung mitunter durch den Umstand, dass sich auf dem PC Markt Blue-Rays nicht durchsetzen konnten. Das bedeutet, dass moderne PC Spiele im Einzelhandel entweder mit bis zu 8 DVDs aufgrund des großen Speicherbedarfs vertrieben wird oder aber nur durch einen Code, den der Käufer dann bei Steam einlöst. Für die Entwicklung durch die voranschreitende Vernetzung sind weiterhin Online Games wichtig. Damit sind in dieser Statistik Spiele gemeint, in denen man für den Spielbetrieb eine stetige Internetverbindung benötigt. Dazu zählen zum Beispiel MMORPGs, wie der wohl bekannteste Vertreter World of Warcraft (kurz WoW) von Blizzard. MMORPGs sind große Massen Online Rollenspiele, in denen oft auch tausende Spieler in einer persistenten Online-Welt zusammenspielen. Diese Art Spiele gehören derzeit zum Games as a Service Modell. Zu dieser Gruppe können auch Shooter wie etwa Rainbow Six Siege von Ubisoft gehören, welches seit seiner Erscheinung ständig Verbesserungen und neue Inhalte erhält. Dieses Verfahren beinhaltet somit Games as a Service. Die Spiele werden nicht einfach nur veröffentlicht und dann wendet sich das Team einem neuen Projekt zu. Die Spiele werden über Jahre hinweg betreut und mit neuen Inhalten über die Internetverbindung versorgt. Hier stellt sich jedoch die Frage, warum ist der Trend, welcher in der Statistik zu erkennen ist, überhaupt präsent? Die Lösung für diese Frage bietet eine Statistik zu den Einnahmen mit so genannten „Microtransactions“. Denn diese sind es, die ein Spiel im Games as a Service Modell oft finanzieren.



Abbildung 4 Umsatz mit kostenpflichtigen virtuellen Zusatzinhalten für Computer und Videospiele in Deutschland von 2008 bis 2016 (BIU 2016)

Wie die Statistik zeigt, sind die Spieler immer mehr dazu bereit für virtuelle Zusatzinhalte zu bezahlen. Diese Entwicklung bleibt auch den Publishern hinter den größten Projekten nicht verborgen. Sie können jetzt entweder eine Einzelspielerfahrung in Auftrag geben, für die sie nur zum eigentlichen Kauf Geld verdienen oder sie finanzieren ein Games as a Service Spiel, welches über Jahre hinweg Geld abwirft. Für die Publisher ist es also die lukrativste Lösung, auf solche Projekte zu setzen, in denen sie nicht nur den Platz als primärer Geldgeber einnehmen, sondern auch die digitale Infrastruktur bereitstellen. So sichern sie sich trotz des Verlustes der Rolle als Produzent und Vertreibers der physischen Datenträger die stärkste Position im Projekt.



Abbildung 5 Umsatz von Activision Blizzard in den Jahren 2005 bis 2016 (Activision Blizzard 2017)

Sich des Verkaufserfolgs solcher Spiele bewusst und aufgrund der rückläufigen Zahlen ab 2012 ging Activision zum Ende des Jahres 2014 mit einem Paradebeispiel eines Games as a Service Spiels an den Markt. Destiny sollte ein reines Online Spiel sein, für das man eine stetige Internetverbindung benötigt und über diese mit neuen Inhalten versorgt wird. Der Erfolg dieses „Shared World“ Shooters bewies, wie erfolgreich das Modell sein kann. Sodass 2016 Overwatch folgte, welches das Geschäftsmodell teilt, jedoch von der Struktur ein anderes Spiel ist. Overwatch ist ein sogenannter „Hero Shooter“. Bei diesen Spielen dreht sich alles um die Spieler-gegen-Spieler Gefechte, die mit den namensgebenden Helden ausgetragen werden. Hier wird Geld erwirtschaftet, in dem Spieler Boxen mit neuen kosmetischen Inhalten für ihre Spielfiguren erwerben. Diese „Lootboxen“ sind äußerst erfolgreich und spielen viel Geld ein (Abb. 4). Sie werden mit realem Geld gekauft und enthalten dann einen zufälligen Spielgegenstand. Aber auch hier müssen immer neue Inhalte nachgereicht werden, um die Spieler am Spiel zu halten. Destiny 2 erscheint noch 2017 und hat sich sogar schon vor dem eigentlichen Erscheinen Millionenfach per Vorbestellung verkauft. Diesem Modell folgen nach diesem Erfolg natürlich auch andere Publisher wie Ubisoft und Square Enix, welche in ihren letzten Geschäftsberichten (Stand August 2017) angekündigt haben, sich ebenfalls stärker auf das Games as a Service Modell zu konzentrieren oder wie im Fall von Ubisoft, alle großen neuen Marken darauf auszurichten. Das bedeutet im Umkehrschluss jedoch auch, dass die

Publisher weniger Kapital für mittlere und kleine oder etwa Einzelspieler Spiele ausgeben werden. Das sorgt für ein großes Vakuum im Markt, welches nützlich für andere sein kann - die Indie Entwickler. Die für die Indie Entwickler entstandenen Chancen werden im nächsten Kapitel anhand der bestehenden Bezahlmodelle und Lebenszyklen genauer untersucht.

## 3 Indie und deren Chancen

### 3.1 Charakteristik von Indie Studios

Genau diese Abwesenheit mittelgroßer Spiele auf dem Markt könnte die große Chance für Indie Entwickler sein, da sie ebenfalls in der Lage sind, Titel solchen Formats umzusetzen. Das Indie steht jedoch für „independent“, also unabhängig. Das bedeutet jeder Entwickler, der sich selbst finanziert, ist Indie, unabhängig davon, wie groß er ist oder das Projekt, an dem er arbeitet. Dies gilt jedoch nicht für Publisher, die eigene Entwicklerstudios unterhalten. Indie Entwicklungen wurden erst durch die digitale Verfügbarkeit von Spielen ermöglicht, wodurch die Abhängigkeit von den Publishern und deren Produktionsketten wegfällt. Die zwei einzigen verbleibenden Hürden für die selbstständige Entwicklung von Videospiele sind die Finanzierung und die eigenen Fähigkeiten. Die finanzielle Unabhängigkeit führt zum einen dazu, dass die umgesetzten Projekte eher klein bleiben. Gleichzeitig kann jedoch auch eine kreative Freiheit freigesetzt werden, die den meisten Studios aufgrund der Machtposition der Publisher verwehrt bleibt und hat zu einigen herausragenden Indie Titeln geführt. No Mans Sky stellt so ein Beispiel dar, welches als ungewöhnliches Konzept jedem Publisher zu risikoreich wäre, für ein Independent Studio jedoch genau die richtige Ausgangsposition für die Etablierung in der Branche sein kann. Der Zugang zum erfolgreichen Vertrieb wird dabei durch das Aufkommen des Early Access (3.3) Modells gewährleistet. In diesem Modell erhalten die Spieler einen Vorabzugang zum vollständigen derzeitigen Stand des in der Entwicklung befindlichen Spiels. Sie behandeln mitunter Themen abseits von großen Blockbustern und können die Nischen abdecken, welche die großen AAA Titel auslassen. Doch die Entwicklung im Indie Stil ist mit Risiken verbunden, da das Spiel ein Erfolg sein muss, um die Produktionskosten zu decken, da es keinen Publisher gibt, der eventuell Geld nachreichen könnte. Wie sieht nunmehr die Monetisierung in den verschiedenen Modellen aus? Welche Chancen werden von den Indie Entwicklern genutzt? Einen Überblick über die häufigsten Bezahlmodelle gibt der folgende Absatz.

### 3.2 Bezahlmodelle

Für Videospiele haben sich über die Jahre viele verschiedene Bezahlmodelle etabliert. Ein Exkurs zu den derzeit geläufigsten soll aufzeigen, welches System für die Entwicklung eines Indie Titels geeignet wäre.

Zunächst bezeichnet „Free to Play“ ein Spielkonzept, das kostenlos gespielt werden kann. Durch die eigentliche Nutzung des Grundspiels entstehen keine Kosten, denn vielmehr wird der Gewinn über das Verkaufen von digitalen Zusatzinhalten oder über das



Platzieren von Werbung im Spiel oder darum erzielt. Auch der Verkauf von Nutzerdaten wie etwa Kaufverhalten oder Spielweise kann zur Erwirtschaftung von Gewinnen beitragen. „Free to Play“-Spiele erliden in der Spielergemeinde allerdings einen eher schlechten Ruf; zum einen werden sie als „Casual Games“, also simple Spiele für Gelegenheitsspieler, verschrien. Dies ist darin begründet, dass der Schwierigkeitsgrad des Spiels im Verlauf des Spielgeschehens so enorm ansteigt, dass Spieler dazu gedrängt wird, für ein leichteres Vorankommen Echtgeld auszugeben. In einigen Spielen lassen sich nicht nur kosmetische Spielinhalte erwerben, sondern auch Objekte die in das Spielgeschehen eingreifen. Wenn man in Spielen Objekte mit daraus resultierenden Spielvorteil käuflich erwerben kann, wird das als „Pay to Win“ bezeichnet. Dennoch erfreuen sich „Free to Play“-Spiele großer Beliebtheit, da die Hürde für den Einstieg aufgrund häufig geringer Systemvoraussetzungen sehr niedrig ist. Diese sorgen letztendlich dafür, dass möglichst viele Kunden die Chance haben, das Spiel zu spielen. Auch gibt es Mischformen mit „Pay to Play“-Spiele, in denen das Grundspiel kostenlos ist und man eine Premiummitgliedschaft erwerben kann, die dem Spieler Vorteile bietet. Ein Beispiel für das Konzept der „Free to Play“-Spiele stellt Candy Crush von King Digital Entertainment dar, wobei das Spiel an sich frei verfügbar ist. Hat man in den schwierigeren Leveln seine begrenzten Züge verbraucht, kann man per InApp-Käufe neue Züge kaufen, um das Level vielleicht doch noch zu schaffen. Die Spieler, die Ihre Freunde zum Beispiel per Facebook auf das Spiel aufmerksam machen, bekommen ebenfalls neue Leben. So sorgt der Entwickler dafür, dass sich das Spiel weiterverbreitet. Das Entwicklerstudio gehört seit Februar 2016 zu Activision Blizzard und damit wieder zu einem der größten Publisher. Diese haben der Wert dieser Spiele erkannt und versuchen eine gute Position in diesem Sektor zu etablieren. Dieses Konzept erfreut sich trotz des schlechten Rufs großer Beliebtheit und verzeichnet teils große Gewinne.

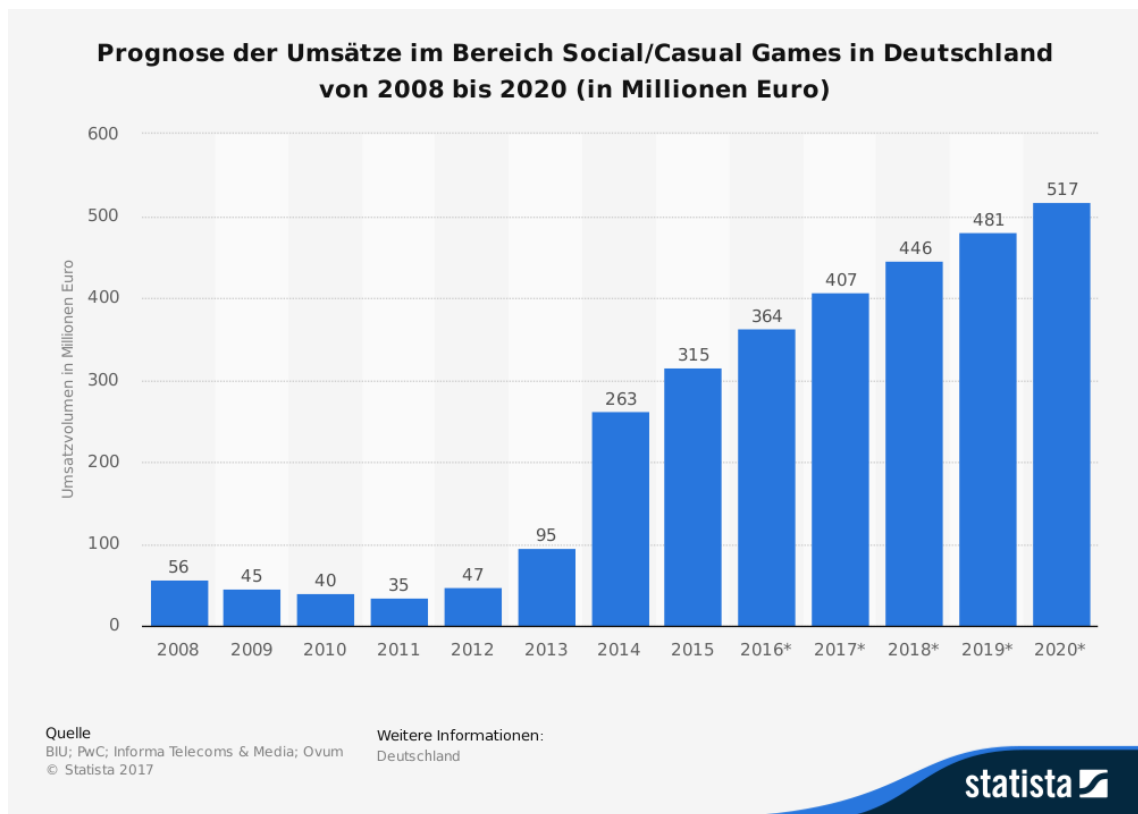


Abbildung 6 Prognose der Umsätze Social/Casual Games in Deutschland von 2008 bis 2020(PwC 2017)

Wie die Prognose zeigt, wird dieses Feld der Videospiele in den nächsten Jahren weiter stark wachsen und große Gewinne erwirtschaften.

„Pay To Play“ ist ein Geschäftsmodell, in dem der Spieler die Zeit kauft, in der er tatsächlich aktiv spielen kann. Das Konzept taucht allerdings weniger auf dem Markt auf, da es fast ausschließlich für Online-Spiele angewendet wird. Dies schränkt die Anwendungsfälle zwar stark ein, jedoch gibt es ausgewählte, enorm große Titel, die auf das Konzept setzen und sich damit teilweise schon seit mehr als 10 Jahren am Markt halten, was für die Videospielebranche ein langer Zeitraum ist. Als die zwei wichtigsten Vertreter wären dabei „World of Warcraft“ von Activision Blizzard und „EVE Online“ von CCP zu nennen. Hierbei muss der Spieler monatlich Abonnements abschließen. Bei „EVE Online“ beeinflusst dieser monatliche Beitrag sogar die hochkomplexe Ingame-Wirtschaft, da hier mit Spielzeit gehandelt werden kann. So können sich versierte Spieler den Beitrag zum Spielen im Spiel selbst erwirtschaften.

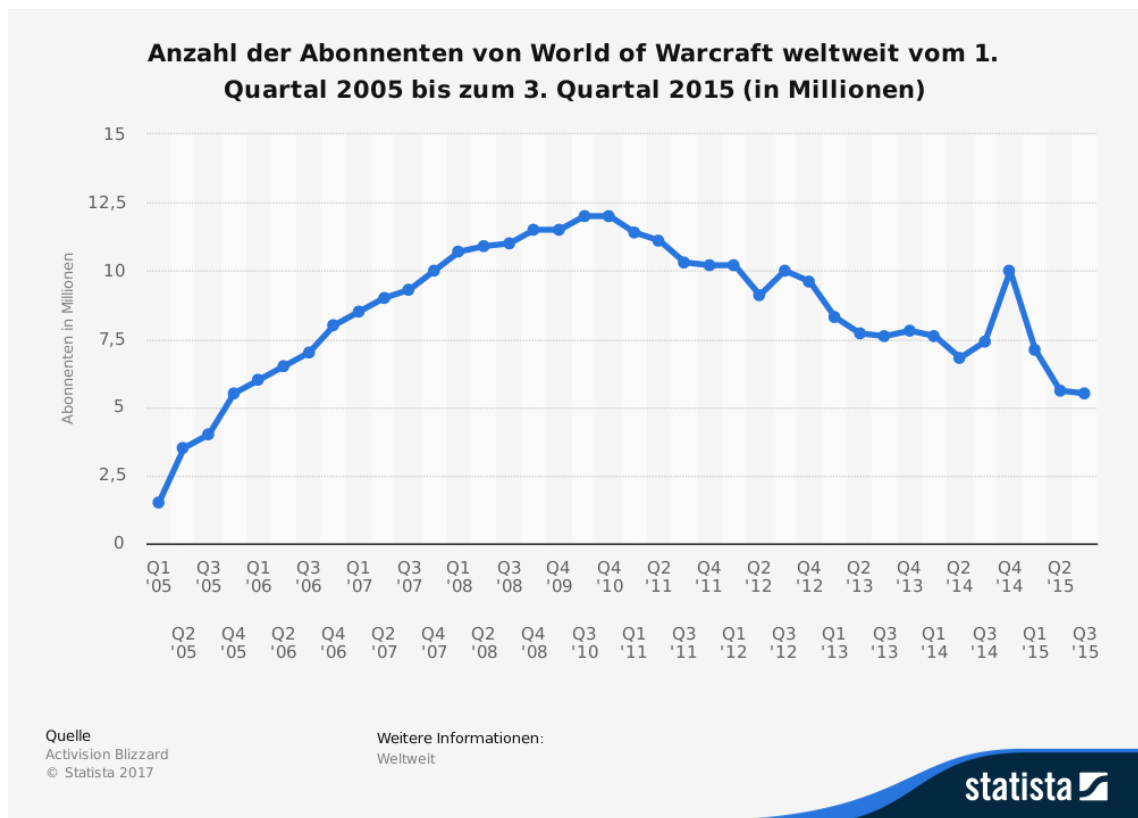


Abbildung 7 Anzahl der Abonnenten von World of Warcraft weltweit vom 1. Quartal 2005 bis zum 3. Quartal 2015 (Activision Blizzard 2015)

Wie in der Grafik zu sehen ist, verzeichnet „World of Warcraft“ seit seiner ersten Veröffentlichung Ende 2004 jeden Monat eine enorme Anzahl zahlende Spieler. Da „WoW“ stetig weiter mit Inhalten versorgt wird, ist auch kein Ende dieser Erfolgsgeschichte in Sicht.

„Buy to Play“ ist der klassische Ansatz im Vertrieb von Videospiele, wobei das Spiel entweder auf einem physischen Datenträger oder als digitale Kopie einmalig vom Spieler gekauft und dann beliebig lang und oft verwendet werden kann. Anwendung findet dieses Konzept in großem Maße sowohl bei Indie Spielen als auch bei AAA Titeln. Bei physischen Datenträgern besteht auch die Möglichkeit, dass Spiel gebraucht zu verkaufen, was allerdings für die Publisher eher schädlich ausfällt. Sie sehen von den wiederverkauften Spielen keine Einnahmen. Zu dieser Form können zusätzlich noch Microtransaktionen bestehen oder zusätzliche Spielinhalte, Addons oder DLC's, nachträglich erworben werden, welche das Spiel im Nachhinein erweitern. Oft wird für ein Spiel ein sogenannter „Season Pass“ angeboten, der zusätzlich zum Hauptspiel erworben wird und der alle Erweiterungen und zusätzlichen Spielinhalte beinhaltet. Dieser Season Pass vermindert den Wert einer gebrauchten Kopie, da das Grundspiel zwar enthalten ist, nicht jedoch die Zusatzinhalte. In den letzten Jahren haben die Anleger

jedoch das Vertrauen in das Geschäftskonzept mit gebrauchten Spielen verloren. Nachdem digital gekaufte Spiele auf dem Vormarsch sind und viele Publisher auf das Konzept von Games as a Service setzten, wird es schwieriger mit gebrauchten Spielen Gewinne zu erzielen.



Abbildung 8 Gamestop A Aktie von September 2014 bis März 2017 (Börse Frankfurt 2017)

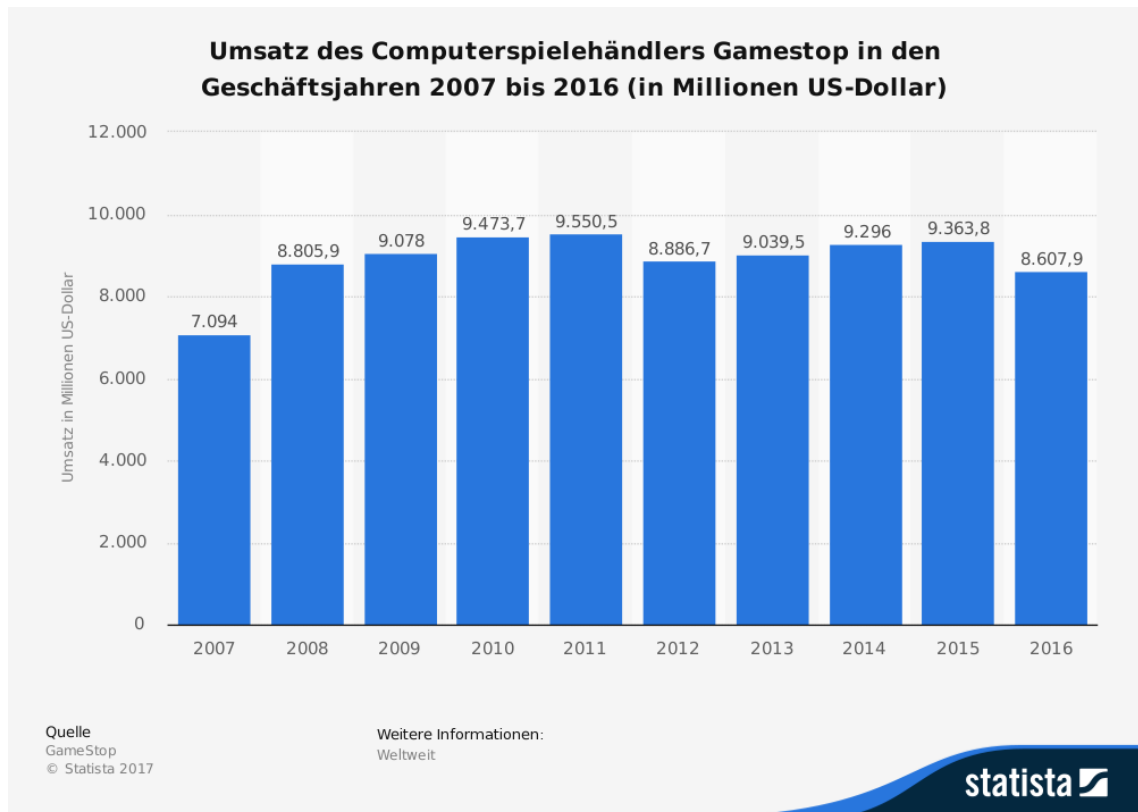


Abbildung 9 Umsatz des Computerspielehändlers Gamestop in den Geschäftsjahren 2007 bis 2016 (Gamestop 2016)

Vom Aktienkurs lässt sich ableiten, dass das Konzept eines Gebrauchtspielhändlers wie Gamestop nicht mehr genügend Potenzial besitzt. Auch wenn die Gamestop Corporation noch stabile Umsätze erwirtschaftet, haben die Anleger kein Vertrauen mehr in das Konzept. Mit den Bezahlmodellen haben sich jedoch auch die Lebenszyklen verändert, welche im Folgenden beschrieben werden.

### 3.3 Lebenszyklen

Die Lebenszyklen moderner Spiele unterscheiden sich stark und sind von Genre, Plattform und Geschäftsmodell abhängig. Hier unterscheidet man in Content-basierte Spiele - also Titel, die durch ihre erzählte Geschichte hervorragen - und denen, die mit einem komplexen Regelwerk aufwarten. Letzteres befasst sich im Kern mit den Mechaniken des Spiels, welche der Spieler lernen und sie meistern muss. Im Fall dieser Arbeit jedoch, ist für die spätere Erstellung eines optimalen Modells wichtiger, wie der technische Lebenszyklus eines Spiels aussieht. Im klassischen Modell wird eine Konzeptidee von einem Designer aus einem Entwicklerstudio erdacht, welcher dem Publisher die Idee unterbreitet. Möglich ist ebenfalls ein Plan zur Fortsetzung einer bereits bestehenden Reihe, bei welchem der Publisher ein eigenes Entwicklerstudio beauftragt oder nach

einem externen Studio suchen würde. Nach den Verhandlungen über Zeitraum und Budget würde die Aufgabenstellung anschließend vom Entwicklerteam analysiert werden und in einen Plan für deren Umsetzung einfließen. Das Designteam legt in dieser Phase schon wichtige Eckpunkte und Designentscheidungen fest. Ab diesem Zeitpunkt beginnt die eigentliche Entwicklung mit der technischen und künstlerischen Umsetzung. Sobald die Entwicklung einen bestimmten Stand erreicht hat und die grundlegenden Funktionen des Spiels implementiert sind, kann es zum Alphatest kommen. Diese Testphase wird gegen Ende des Entwicklungsprozesses meist von internen Testern durchgeführt. Die Alphaphase kann sehr lange andauern, da hier die größten Fehler gefunden und gelöst werden sollen. So soll die Bereitschaft für den folgenden Betatest erreicht werden, welcher kurz vor dem Release des Produkts in der mit authentischen Kunden durchgeführt wird. In den letzten Jahren jedoch hat sich besonders bei der Entwicklung von Videospielen das Konzept des Early Access verbreitet.

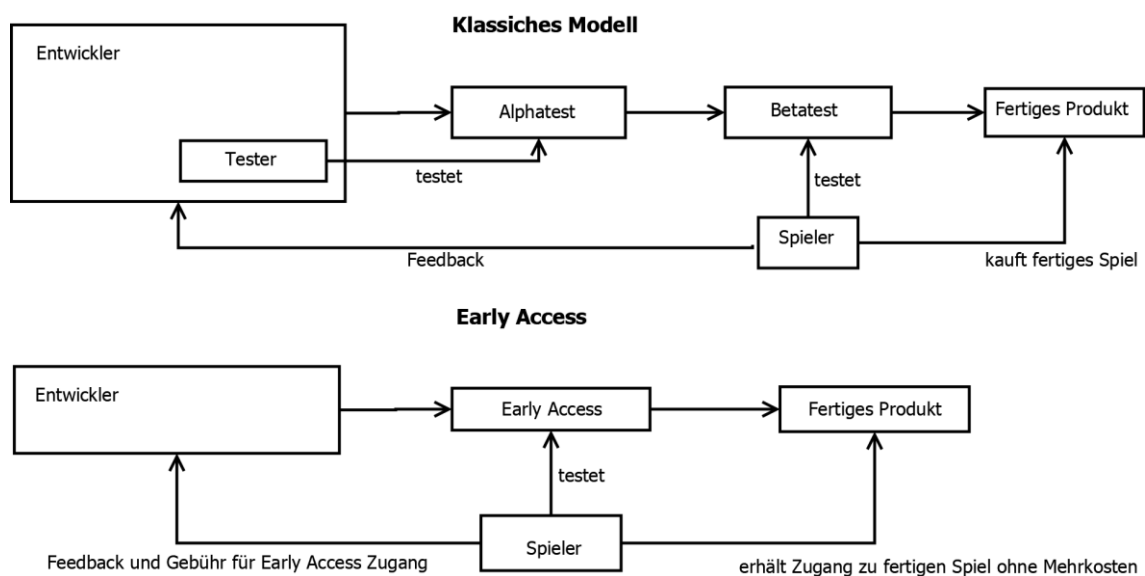


Abbildung 10 Unterschiede des Early Access Modells

Hier wird den späteren Kunden die Möglichkeit gegeben, die Spiele schon in einem sehr frühen Stadium zu erwerben und zu spielen. Der Vorteil für die Spieler ist dabei, dass sie das Spiel günstiger als zum Release erwerben und maßgeblich mit am Entwicklungsprozess beteiligt sein können. Sie geben den Entwicklern in der frühen Phase Feedback und Einnahmen, was für eine sicherere Finanzierung sorgt und teure Tester erspart. Außerdem kann der Entwickler so besser auf die Wünsche und Vorstellungen der Spieler eingehen. Das Konzept hat jedoch viel Kritik erhalten da viele der Projekte nach dem sie oft im Early Access verkauft worden sind, sich nicht mehr oder kaum noch weiterentwickeln. Daher sind viele Kunden bei diesem Konzept verunsichert, mit der Befürchtung, dass der frühzeitige Kauf sich im Nachhinein nicht rentiert, da Inhalte eventuell nicht wie

erwünscht erscheinen. Nach der Veröffentlichung wird in der modernen Spieleentwicklung oft noch Monatelang an Patches gearbeitet. Bei Content-basierten Singleplayer Spielen wird sogar dann noch an Erweiterungen gearbeitet, die den Spielumfang weiter erhöhen. Bei Multiplayer Spielen muss nach der Veröffentlichung mehr Aufwand betrieben werden, da zusätzlich die Online-Infrastruktur aufrechterhalten werden muss. Besonders PvP Spiele Bedarfen weitere Anpassung und so genanntes Balancing der Spielinhalte. Hier wird die Stärke von Spielinhalten so angepasst, dass der Spieler erst nach einer gewissen Zeit und Aufwand den High-End-Content erhält. Damit soll gewährleistet werden, dass kein Spieler nur durch die Konfiguration seiner Ausrüstung einen Vorteil erhält. Diese Spiele haben auch einen weit längeren Lebenszyklus als klassische Singleplayer Spiele, bei welchen sich das Entwicklerstudio nach dem Release oft schon dem nächsten Spielprojekt widmet. Diese Spiele werden in der modernen Entwicklung jedoch nicht jedes Mal von Grund auf neu programmiert, da es selbst von großen Entwicklern bei grafisch und spielerisch anspruchsvollen Titeln nicht zu bewältigen wäre. Es werden sogenannte Game Engines, die den Kern der modernen Produktion von Computerspielen darstellen, genutzt. Sie bilden das Framework, mit dem das Computerspiel erstellt wird. Untereinander unterscheiden sich die Engines jedoch stark, weswegen konkret abgeschätzt werden muss, welche die richtige für das ausgewählte Spiel darstellt.

### **3.4 Moderne Engines und deren Ansätze**

In den Anfängen der Spieleentwicklung standen fest verdrahtete Spieleautomaten, die nur durch die Verwendung von unausgereiften Programmiersprachen generiert werden konnten und zudem nicht portierbar waren. Nachdem die Spiele immer komplexer wurden, schien es kaum noch möglich diese in solchen Sprachen wie beispielsweise Assembler umzusetzen. Daher gelten sogenannte Entwicklungsumgebungen als bahnbrechend, da die Spieleprogrammierung stark von der Darstellung von Echtzeitgrafik abhängt. Diese nennen sich Engines und sind der wichtigste Bestandteil der modernen Spieleentwicklung. Sie bieten eine grafische Nutzeroberfläche und beinhalten meist schon die Möglichkeit 3D Objekte in Echtzeit darzustellen und oft auch schon das Wirken von physikalischen Kräften auf diese. In diesen Engines können die Spielwelten umgesetzt werden und arbeiten daher oft in Verbindung mit einer klassischen IDE zu Erstellung des Quellcodes. Bei den Engines wird vorrangig zwischen zwei Kategorien unterschieden; Third Party Engines und Custom Engines, die auch InHouse Engines genannt werden. InHouse Engines sind Engines, die vom Entwicklerstudio selbst mit ganz bestimmten Anforderungen entwickelt worden sind. Für das Modell zur effektiven Erstellung von Videospiele sind die Custom Engines von geringerer Bedeutung, da die Entwicklung dieser Kosten- und Zeitaufwendig ist. Eine Third Party Engine sorgt durch ihre schon bestehende Funktionalität dafür, dass schneller mit der eigentlichen Entwicklung begonnen werden kann. Auf dem Markt existieren mehrere dieser Engines. Hier

werden die zwei bekanntesten und am weitesten verbreiteten Engines analysiert und bewertet - die Unity Engine und die Unreal Engine.

### 3.4.1 Unity

Unity ist eine 3D Engine, entwickelt von Unity Technologies in den USA. Die Engine ist besonders beliebt bei Einsteigern der Branche, da sie einfach zu handhaben ist und als Scriptsprache einsteigerfreundliche Sprachen wie C# und Unity Script unterstützt. Große Popularität erlangte die Engine aufgrund ihrer hervorragenden Eigenschaften, nicht nur Bausteine zur Erstellung von Videospielen, sondern vor allem von Apps, bereitzustellen, beispielsweise für architektonische Visualisierungen. Denn Unity bietet umfassende Möglichkeiten, 2D Spiele zu entwickeln, besonders in Hinsicht auf mobile Endgeräte. Sie gilt als die am weitesten verbreitete Spieleengine. Dieser Erfolg ist auch der enormen Portierbarkeit der Engine zu verdanken. Anwendungen aus der Engine können für Spielekonsolen über Smartphones bis hin zu Kühlschränken entwickelt werden. Je nach Zielplattform nutzt Unity OpenGL oder Direct3D. Das Layout der Engine ist an 3D Animationsprogramme angelehnt, wobei es jedoch ebenfalls beliebig vom Entwickler angepasst werden kann. Es ist dem Entwickler auch möglich, selbst Fenster für den Unity Editor zu schreiben und diese im Entwicklungsprozess zu nutzen. Die Unity Versionen, die auf dem Standardweg über die Website des Entwicklers erworben werden kann, enthält keinen Zugriff auf den vollen Quellcode. Es kann jedoch, indem Unity Technologies direkt kontaktiert wird, eine Sourcecode-Lizenz erworben werden. Das bedeutet, dass nicht nur Fenster im Editor angepasst werden können, sondern auch der Quellcode des Editors an sich. Unity bietet weiterhin die Möglichkeit, sogenannte Prefabs anzufertigen. Ein Prefab ist ein in Unity erstellbares Gameobjekt, sogenannte einzelne Objekte in der virtuellen Szene. Diese Prefabs lassen sich direkt in der Szene konfigurieren und anschließend speichern. Sollten mehrere Instanzen des Prefabs in einer oder mehreren Szenen existieren, muss nur eine Instanz angepasst werden, um alle anderen Instanzen ebenfalls anzupassen. Im Zusammenspiel der Möglichkeit, eigene Editorfenster zu schreiben und dem Nutzen von Prefabs können auch spezielle Fenster für diese Prefabs und deren Anpassung erstellt werden. So kann der Arbeitsprozess stark spezialisiert und die Effektivität gesteigert werden, vorausgesetzt, es wird vorher die Zeit aufgewendet um diese Fenster und Prefabs durch Programmierer zu erstellen. Dies soll es vereinfachen, wiederkehrende Objekte zu verwalten und zu modifizieren. Unity hat in seiner Standardkonfiguration jedoch keine „Visual Scripting“ Funktion implementiert. Die Engine muss, wenn sie professionell genutzt werden will, von Unity Technologies für jeden Arbeitsplatz gekauft werden. Der Preis der Lizenzen richtet sich hierbei nach den Einnahmen des Entwicklerstudios, wobei mit einer Pro Lizenz der Gewinn unbegrenzt ist. Aus dem nicht mit dem Gewinn proportional ansteigenden Abgaben an Unity Technolo-



gies ist abzuleiten, dass der Entwickler der Engine auf weiteren Wegen Gewinne erwirtschaftet. Dies geschieht über sogenannte Unity Ads, welche das Einbinden von Werbung bezeichnen. Wenn der Entwickler des Spiels etwa Werbung in sein Spiel einbaut, erhält er dafür Geld vom Werbetreibenden. Da Unity Technologies die Technologie dafür bereitstellen, geht ein kleiner Teil der Gewinne an sie. Werbung kann so in Videospielen platziert werden, dass sie wie ein Teil der Spielwelt wirkt oder aber so, dass der Spieler die Werbung nicht sehen muss, dadurch aber einen kleinen Vorteil gewinnt. Ein Beispiel dafür wäre, dass der Spieler für das Ansehen eines Werbespots einen zusätzlichen Versuch für ein Level erhält. So muss der Spieler, wenn er nicht möchte, den Spot nicht sehen, erhält aber, wenn er den Spot sieht, automatisch eine gute Erfahrung, da er danach weiterspielen kann. So steht die Werbung in einem positiven Kontext. Besonders für diese eignen sich solche Werbekonzepte. Die Schlussfolgerung daraus ist, dass sich Unity für das angestrebte Entwicklungsmodell gut eignet, der Fokus der Engine jedoch mehr auf Mobile Games und einfacher Zugänglichkeit als auf Spielen mit AAA Qualitäten liegt.

### 3.4.2 Unreal Engine

Die Unreal Engine ist eine voll auf C++ setzende Engine von Epic Games. Die Engine erschien zuerst 1998 und zur gleichen Zeit zusätzlich das Spiel Unreal, in dem die zu dieser Zeit erstaunliche Grafik der Unreal Engine zu sehen war. Die Unreal Engine ist für ihren Einsatz in vielen großen AAA Titeln bekannt. Die Besonderheiten der Engine sind vor allem, dass sie in ihrem vollen Umfang und mit kompletten Sourcecode Zugriff für jeden Nutzer kostenlos zum Download bereitsteht. Erst wenn ein Spiel Einnahmen erzielt, erhält Epic Games 5% von diesen Einnahmen. Das bedeutet auch, dass unbegrenzt viele Entwickler gleichzeitig an einem Projekt arbeiten können und nicht für einzelne Lizenzen bezahlt werden muss. Wenn also für das Spiel vor der eigentlichen Produktion auf Basis der Unreal Engine entwickelt wird, fallen dafür keine Kosten an, wobei die Engine beliebig oft und lange verwendet werden kann. Unreal hat ähnlich wie Unity mit den Prefabs ein System, um mehrere Instanzen einer Art von Objekt zu verwalten. Diese nennen sich in Unreal Blueprints und ermöglichen es nicht nur, wie bei den Prefabs, alle Instanzen eines Objekts simultan zu bearbeiten, sondern auch das visuelle Skripten von Verhalten über sogenannte Nodes. So soll es Designern ohne Programmierkenntnissen auch möglich sein, Objekte in der Spielwelt zu erstellen und diese dann rudimentär zu skripten. Den Designern können dann von den Programmierern neue Funktionen in den Blueprints zur Verfügung gestellt werden. Durch diese Methodik können sich die Programmierer stärker auf ihre Kernaufgaben konzentrieren und die Designer ihre Vorstellungen direkt ohne einen Programmierer umsetzen. Die Unreal Engine bietet zudem auch Support für alle gängigen Plattformen.

### 3.4.3 Vergleich

	Unity	Unreal
Voller Sourcecode Zugriff	beschränkt	Ja
Bezahlmodell	Monatlich bezahlte Lizenzen abhängig von der Anzahl der Entwickler	5% des Gewinnes des Gesamten Projektes
Visual Scripting	Nein	Ja
Support für Zielplattform (PC, PS4, Xbox One)	Ja	Ja
Scriptsprache	C#	C++

Abbildung 11 Engine Features

Für die Entwicklung im hier angestrebten Modell wird eine möglichst flexible, für alle Beteiligten einfach zu nutzende und kostengünstige Engine benötigt, um maximale Effektivität zu erreichen. Beide Engines bieten große Flexibilität. Unity bietet zwar Support für mehr Geräte als Unreal, jedoch ist für dieses Modell die Unreal Engine als bessere Option in Punkto Flexibilität zu betrachten. Denn sie unterstützt jede Plattform, die für ein Spiel in AAA Qualität nutzbar wäre. Zudem bietet sie von Haus aus vollen Zugriff auf den Sourcecode und eine Vielzahl von mitgelieferten Komponenten, wie einen Editor für Zwischensequenzen oder Materialeditoren und Animationstools. Auch in der Benutzerfreundlichkeit hat die Unreal Engine durch ihr „Visual Scripting“ System einen Vorteil gegenüber der Unity Engine. Hier können die Designer selber an dem Verhalten der Objekte arbeiten, was besonders den Programmierern, die bei einer Indie Entwicklung meist am häufigsten vertreten sein müssen, hilft, sich auf ihre Kernaufgaben zu konzentrieren. Der letzte zu betrachtende Punkt ist der Preis. Dieser spielt bei dem angestrebten Modell eine große Rolle, da bei einer Eigenfinanzierung das Kapital sehr begrenzt ist. Das Finanzierungsmodell der Unreal Engine ist als das bessere für diese Prämisse zu betrachten, da im Gegensatz zu Unity keine Kosten während der Entwicklung anfallen. So kann, indem viel der Entwicklungsarbeit in Unreal vorgenommen wird, die Kosten für Software und Lizenzen im Projekt minimiert werden. Dies ist durch den Umfang der mitgelieferten Funktionen möglich. Die Kosten für die Unreal Engine richten sich auch nach dem Release nach dem Erfolg des Spiels. Werden keine Einnahmen erzielt, müssen auch keine Abgaben an Epic gemacht werden. So wird das finanzielle Risiko für den Indie Entwickler weiter miniert. Die Unreal Engine erweist sich als die optimale Plattform für die Entwicklung in diesem Modell. Doch nicht jede Arbeitsweise ist nur von der Engine

abhängig. Viele Entscheidungen sind aufgrund der jeweiligen Anforderung durch das Projekt gegeben, wie beispielsweise die Erstellung einer sehr großen Spielwelt oder vieler Assets, die nicht durch ein so kleines Team manuell erstellt werden könnten. Der Schlüssel um diese Aufgaben bewältigen zu können, stellen, wie im folgenden Kapitel beschrieben, prozedurale Inhalte dar.

## 4 Prozedurale Inhalte

Moderne Spiele sollen möglichst viel Inhalt bieten. Diesen zu erstellen ist aber zeit- und kostenaufwendig. Um mehr Inhalte zu erstellen, werden oft mehr Mitarbeiter benötigt, da die Entwicklung nicht unbegrenzt lange andauern darf. Um ein AAA Spiel mit wenigen Entwicklern und einem verhältnismäßig geringen Budget umzusetzen, ist es nicht möglich, die Inhalte von vielen Entwicklern parallel erstellen zu lassen. Viele Spiele setzen daher auf prozedurale Inhalte, um diese Kapazitätenlücke zu komprimieren. Prozedurale Inhalte bezeichnen laut Derek Yu Elemente, die durch einen Algorithmus generiert werden (vgl. Short und Adams 2017: S.XVII). Durch einen Algorithmus geformte und erstellte Inhalte. Auch sehr frühe Spiele wie das schon 1980 veröffentlichte Rouge bedienten sich dieser Alternative und schufen somit sogar die Grundlage, aus der später große Titel wie etwa Diablo von Blizzard entstehen sollten. Die Verwendung von prozeduralen Inhalten muss jedoch im Vorfeld durchdacht und geplant werden, denn nicht alle Spielkonzepte eignen sich für prozedurale Inhalte. Sollte damit eine Kernspielmechanik umgesetzt werden, ist es vielleicht nötig, das ganze Spiel um diese automatisch generierten Inhalte zu konstruieren. Wenn nicht für die Kernsysteme kann die Generierung von prozeduralen Inhalten auch nur für einzelne Teile verwendet werden, zum Beispiel für ein Musiksystem, das seine Musik automatisiert und an das Geschehen im Spiel anpasst generiert. In diesem Kapitel wird ebenfalls die optimale Verwendung von prozedural generierten Inhalten beschrieben. Um zu bewerten, wie nützlich prozedurale Inhalte für das hier behandelte Konzept sind und wie sie verwendet werden sollten, müssen zunächst Vor- und Nachteile analysiert werden. Die beiden folgenden Beispiele sollen dabei analysiert und hinsichtlich dieser Aspekte untersucht werden, da sie unterschiedliche Herangehensweisen der Nutzung von prozeduralen Inhalten bieten. Aus beiden und können Probleme und Vorteile der prozeduralen Inhalte abgeleitet werden. Doch erst die Unterschiede der Herangehensweisen und die daraus resultierenden Ergebnisse sorgen für die Erkenntnisse, welche für ein optimales Modell wichtig sind. Zuerst wird hierbei ein Projekt betrachtet, dass jedes Element des Spiels prozedural generiert - No Mans Sky.

### 4.1 Beispiele

#### 4.1.1 No Mans Sky

Im 2016 erschienen No Mans Sky von Hello Games wird in jeder seiner Stufen prozedural generiert, woraus eine ganze Galaxie, von den Sternsystemen über die Planeten bis hin zur Fauna und Flora, entsteht. Auch Flotten von Handelsschiffen oder Weltraumpiraten werden durch einen Algorithmus in der Spielwelt platziert. Das Ziel des Spiels ist

es, das Zentrum der Galaxie zu erreichen. In einem Interview mit dem Playstation.Blog erklärte Shaun Murray von Hello Games, wie viele Planeten in No Mans Sky theoretisch möglich sind: „Wir arbeiten mit einem 64-bit-System, das heißt  $2^{64}$  ... oder 18.446.744.073.709.551.616 mögliche Planeten.“ (Murray Sean 2014) Im Weiteren erläutert er, dass bei No Man Sky jeder Planet eine spezifische Nummer erhält. Diese einzigartige Nummer, auch Seed genannt, identifiziert den Planeten nicht nur eindeutig, sondern ist auch der Basiswert für den Generator. Von diesem hängen alle Objekte auf dem Planeten ab; jede Erhebung, Pflanze oder Tier auf dem Planeten basiert auf diesem Wert. (vgl. Murray Sean 2014) Solch ein System ermöglicht es, diesen Seed zu notieren und den Planeten mit diesem Seed wieder zu besuchen. Man muss bedenken, dass die meisten Taten auf den Planeten nicht gespeichert werden, ausgenommen der Name, der dem Planeten oder entdeckten Arten gegeben wurde. Würde alles gespeichert werden, würde das zu einem nicht bearbeitbaren Maße an Daten führen. Für den jeweiligen Spieler wird immer nur das Sonnensystem generiert, in dem er sich derzeit aufhält. Die restlichen Systeme und Planeten sind, bis auf ihren Seed, nicht wirklich vorhanden, bis man sie besucht und sie generiert werden. Erst dann werden zum Seed gespeicherte Informationen, wie der Entdecker abgerufen. Das ermöglicht dem Spiel, auch selbst sehr wenig Speicherplatz zu benötigen. Da der Generator jedes Spielers auf der Welt auf die gleiche Art und Weise funktioniert, ist es sogar möglich, mit dem gleichen Seed die gleichen Planeten zu betreten. Da die Planeten in der Größe echten Planeten sehr nahekommen, ist es jedoch nicht möglich, die Planeten vollständig beim ersten Sichtkontakt durch den Spieler zu generieren, da sonst die Speicherlimits moderner Systeme überlastet werden würden. Um das Problem zu lösen werden, die Planeten in verschiedenen Detailstufen entlang einer Gameplay-Schleife generiert und gerendert.

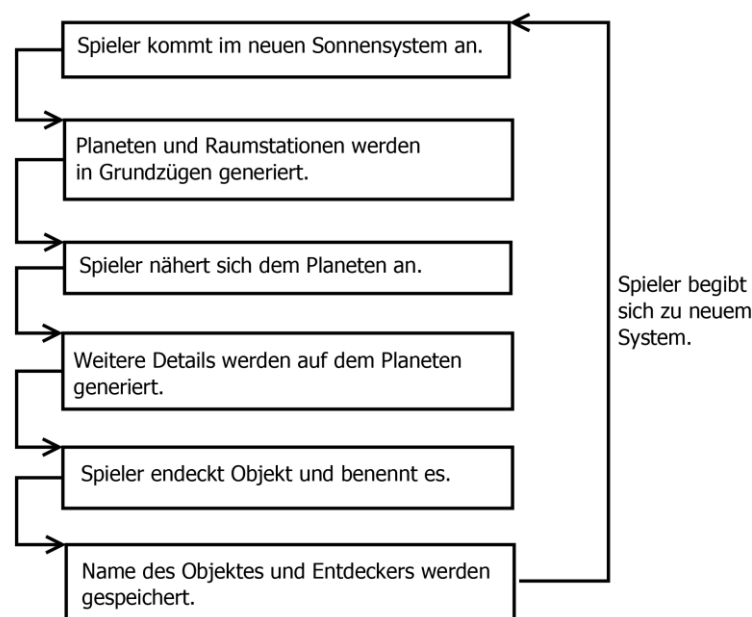


Abbildung 12 Gameplayschleife in No Mans Sky



Abbildung 13 No Mans Sky. Planet in niedrigster Detailstufe

Auf Abbildung 13 ist ein Planet aus No Mans Sky in seiner niedrigsten Detailstufe zu sehen. Der Spieler bewegt sich zwischen den Planetensystemen durch Warpsprünge. Diese Phasen nutzt das Spiel um das nächste Planetensystem zu so generieren, dass der Spieler, wenn er im System ankommt, alle Planeten und Raumstationen im System sehen kann. Erst wenn der Spieler sich dem Planeten nähert, wird für diesen auf Basis des Seeds mehr Detail generiert und gerendert.

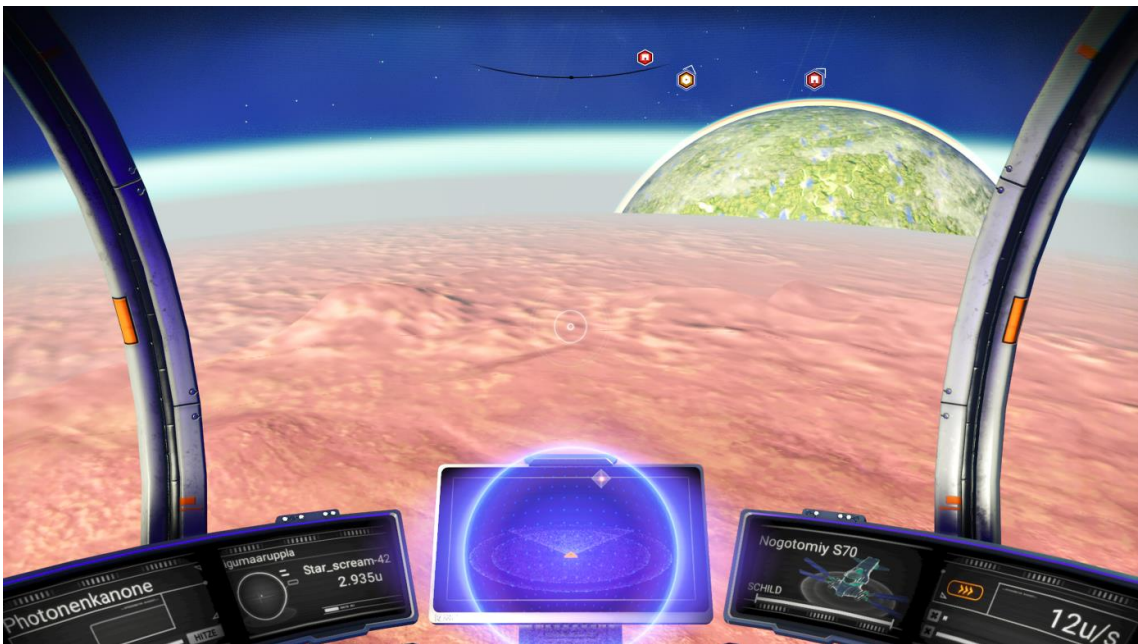


Abbildung 14 No Mans Sky. Annäherung an Planeten

Auf Abbildung 14 ist erkennbar, wie für den Planeten bei Eintreffen des Spielers eine feinere Oberflächenstruktur generiert wird. In dieser Abbildung anhand des Berges auf der linken Seite gut zu erkennen ist, dass die Entwickler nicht den ganzen Planeten in der gleichen Detailstufe generieren, sondern die Rückseite des Berges abgeschnitten wird, da der Berg sich auf der Grenze zwischen zwei Rastern des Planeten befindet. Nur das Raster, in dem sich der Spieler befindet, wird mit vielen Details dargestellt. Der Unterschied zwischen den beiden Feldern ist daher so stark zu erkennen, da für das vordere die Geometrie des Planeten schon verändert wird, um Erhebungen darzustellen, währenddessen im dahinterliegenden Feld jedoch nur die Illusion von Erhebungen durch moderne Rendertechniken erzeugt wird. Dies soll Rechenleistung und Speicher sparen. An diesem Punkt wird allerdings ein Problem der prozeduralen Inhalte sichtbar; es ist nicht auszuschließen, dass der Spieler solche Vorgänge erblickt. Der Entwickler kann nicht vorhersagen, dass der Planet diese konkrete Größe besitzt, auf dem sich die Raster auf diese Weise anordnen und genau auf der Grenze zwischen zwei Feldern, ein solch enormer Berg generiert werden würde. Ein automatisierter Testalgorithmus kann solche Probleme nur schwer erkennen. Selbst wenn das Problem erkannt werden würde, könnte wegen des einen Berges auf dem Planeten nicht auf die diese Funktionsweisen zur Performanceoptimierung verzichtet werden.



Abbildung 15 No Mans Sky. Eintritt in Planetenatmosphäre

In Abbildung 15 ist das Aufteilen der Planetenoberfläche in rechteckige Felder zu erkennen; in der Mitte scheint ein rechteckiger Bereich mit im Vergleich zum Rest höher aufgelösten Texturen. Die Entwickler verschleiern mit Hilfe der Visuellen Effekte wie des Schweißs um das Cockpit, der sich beim Eintritt in die Atmosphäre bildet, die niedriger



aufgelösten Bereiche. In dieser Phase sind schon erste Merkmale der Landschaft auszumachen, da nicht alle Objekte erst bei einer Annäherung generiert werden. Der Spieler hat somit schon aus dem Orbit die Möglichkeit nach besonderen Orten zu scannen. Diese werden mit dem Scantvorgang generiert, da es schon aus dem Orbit möglich sein muss, zu diesen Orten zu navigieren. Wenn also prozedurale Systeme genutzt werden, muss nicht nur bedacht werden, wie diese Inhalte generiert werden, sondern auch wann und in welcher Reihenfolge. Auch werden zum Beispiel Daten über das Terrain hinter dem Spieler viel schneller wieder vom Renderer verworfen, als die Standorte, der zuvor erwähnten besonderen Orte. So muss genau geplant werden, welche Inhalte noch relevant sind und welche nicht. Wenn mit so großen Objekten wie mit Pflanzen gearbeitet wird, kann nie der ganze Planet gleichzeitig im Speicher gehalten werden. Es muss immer sichergestellt werden, dass kein Planet oder eine Kombination bestimmter Planeten zu viel Speicher benötigt. Bei den prozedural generierten Inhalten ist dies sehr aufwendig und muss ebenfalls getestet werden.

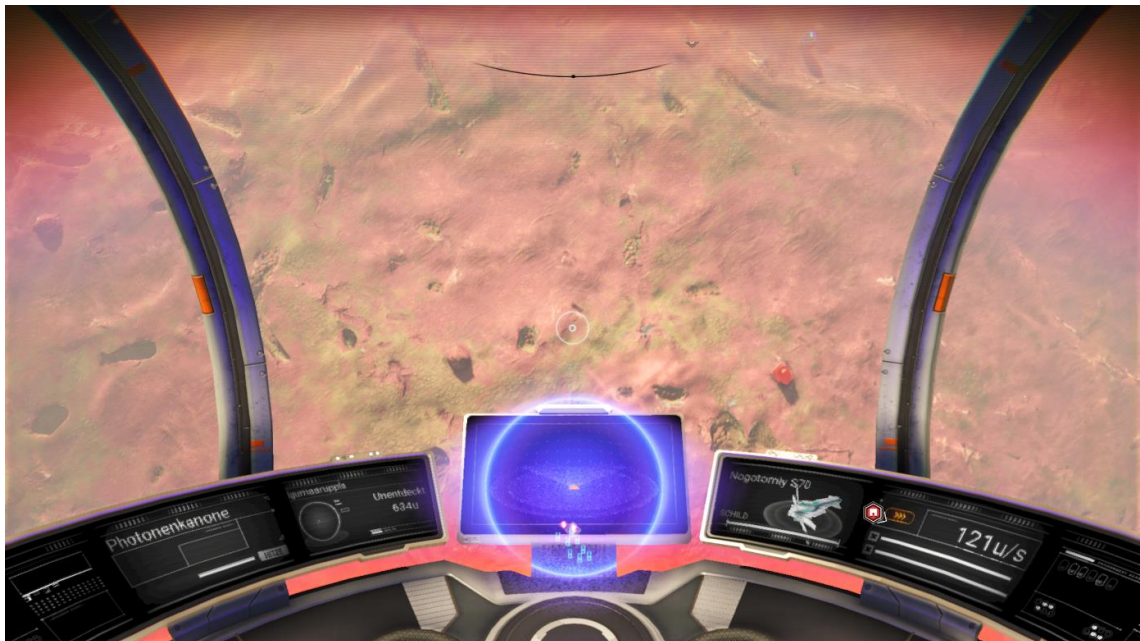


Abbildung 16 No Mans Sky. Annäherung Planetenoberfläche

In dieser Annäherungsphase werden der Spielwelt Texturen und Reccourcen hinzugefügt. Wie in Abbildung 16 zu sehen, wird hier endgültig festgelegt, welche Farben und Texturen der Boden haben wird; letztere werden ebenfalls prozedural generiert.





Abbildung 17 No Mans Sky. Landung auf der Planetenoberfläche

In der letzten Phase werden der Oberfläche noch Tiere und Pflanzen hinzugefügt. Am Horizont auf Abbildung 17 ist der Berg zu erkennen, welcher schon aus dem Orbit sichtbar war und auf der Grenze der Felder lag. Wird in dieser Höhe diese Grenze jedoch passiert, ist kein Unterschied mehr zu erkennen, da das Spiel das nächste Feld schon im Voraus generiert.

Auch die Tiere werden von einem Generator erstellt. Das bedeutet, dass die eigentlichen Geometrieinformationen, die in den Spieldateien hinterlegt sind, keine kompletten Tiere speichern, sondern nur als Container für die Teile der Tiere fungieren. Diese Dateien tragen die Endung `.GEOMETRY.MBIN` und werden durch `.SCENE.MBIN` Dateien in Relation gesetzt. Hier werden die Objekte aus den Geometriedaten in einem Szenengraphen angeordnet, welche wiederum auf die Geometriedateien referenzieren. Die Informationen über deren Zusammensetzung und die hierarchische Anordnung werden in einer weiteren Datei mit der Endung `.DESCRIPTOR.MBIN` gespeichert. Aus dieser Datei weiß der Generator, welche Teile er wie zusammensetzen kann. (vgl. 3D Gamedevblog 2016) Hier haben alle Objekte einer Kategorie, wie zum Beispiel Köpfe, welche durch einen Unterstrich abgetrennte Bezeichnung wie `_Head` aufweisen. Aus dieser Struktur lässt sich ableiten, dass wenn eine Kreatur generiert wird, das Programm durch eine Baumstruktur in Form der Scene läuft und aus jeder Kategorie auf Basis der Vorgaben der vorangegangenen Teile und des Planetenseeds weitere Teile auswählt. Um prozedural Inhalte zu generieren, wird also auch eine sehr strikt definierte Datenhaltung benötigt. Prozedurale Inhalte werden jedoch auch separat oder nur für Teile der Entwicklung so wie in Star Citizen genutzt.

### 4.1.2 Star Citizen

Star Citizen ist teilweise seit 2013 spielbar, befindet sich jedoch zum Zeitpunkt der Bearbeitung dieser Arbeit noch immer in Entwicklung. Entwickelt wird das Spiel von Robert Space Industries, das vor allem für seine immense Crowdfunding Kampagne bekannt ist. Über das Crowdfunding wurden schon mehr als 65 Millionen US Dollar eingenommen. (Robert Space Industries 2017) In diesem Projekt werden ebenfalls prozedural generierte Planeten verwendet, jedoch sind die weiteren Spielinhalte manuell erstellt. Auch ist die Zahl an Planeten viel geringer als bei No Mans Sky. Der Grund aus dem Prozedural generierte Planeten gewählt wurden, unterscheidet sich jedoch im Kern von No Mans Sky. Bei No Mans Sky war das Ziel eine enorme Menge an Planeten zu erschaffen, die die Spieler dann entdecken können. Bei der enormen Anzahl der Planeten war es nicht möglich, diese zuvor zu definieren. So wurde ein prozeduraler Ansatz gewählt.

"It's absolutely artist driven. We're using the procedural techniques to author planets on a global scale very quickly, but to give it a real sense of character. There are quite a few other games using procedural tech to do planets, and those achievements are impressive and cool, but I want to be able to say: I want to build this planet that looks like Hoth. I want to build this planet that looks like Tatooine." (Roberts Chris 2016)

Chris Roberts hebt hier in seiner Präsentation auf der CitizencCon 2016 hervor, dass bei Star Citizen die Planeten primär prozedural generiert werden, um Speicher zu sparen und Planeten in realistischer Größe abzubilden, sie aber dennoch von Designern erstellt werden. Dieser Ansatz ermöglicht es im Weiteren, mit dem Wissen, wie die Planeten aufgebaut sind, manuell Objekte und Missionen auf der Oberfläche zu platzieren. So können die Vorteile der prozeduralen Techniken und der von Designern erstellten Inhalte genutzt werden. Ein Beispiel dafür ist die Erstellung der Städte in Star Citizen.

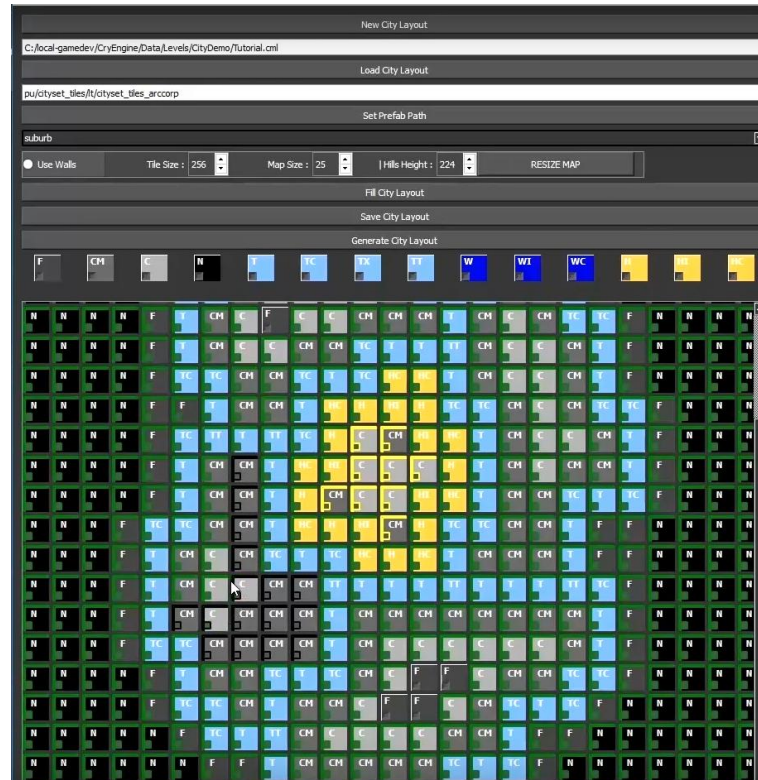


Abbildung 18 Star Citizen Stadtgenerator(Star Citizen 2017)

Für die Generierung der Städte in Star Citizen stehen den Designern besondere Tools zur Verfügung, wie beispielsweise der in der Abbildung 18 zu sehende Generator für die Städte, in dem die einzelnen Bezirke definiert werden. Die Bezirke wurden vorher von Designern und Programmierern entworfen und dem Generator zur Verfügung gestellt. Der Designer wählt an dieser Stelle die Höhe und die Position der einzelnen Bezirke und kann so das Erscheinungsbild der Stadt festlegen. Die einzelnen Bezirke müssen daher nicht einzeln erstellt werden, sondern werden vom Generator mit Inhalten und Objekten gefüllt und dann passend miteinander verbunden.



Abbildung 19 Star Citizen Generierte Stadt (Star Citizen 2017)

Das Ergebnis des Generators ist eine Stadt, welche ein immensen Detailgrad aufweist und binnen kürzester Zeit von einem Designer nach einem intelligenten Design konstruiert wurde. Von jeder Art Grundbaustein der Stadt müssen mehrere Versionen vorhanden sein. Würde jedes Mal der gleiche verwendet werden, würde dem Spieler auffallen, dass die Stadt von Code generiert wurde. Indem schon wenige Variationen eines Bausteins in variabler Skalierung und Rotation verwendet werden, erscheint die Stadt viel individueller und authentischer. So können die Entwickler binnen kürzester Zeit sehr viele Inhalte produzieren.

## 4.2 Vorteile

Durch prozedural generierte Inhalte entstehen viele Vorteile; ein fehlerfrei funktionierendes System kann, wie am Beispiel von No Mans Sky zu sehen, fast endlos viele neue Spielinhalte erzeugen. Ein wichtiger Punkt solche Inhalte zu benutzen ist vor allem die Zeitersparnis. Mit solch einem System können binnen kürzester Zeit enorme Mengen an Inhalten generiert werden. So nutze etwa das von 2011 von Bethesda veröffentlichte The Elder Scrolls: Skyrim solch ein System um die riesige Landmasse zu generieren. Das System generierte das Terrain, das dann händisch mit Inhalten gefüllt wurde. Viele oft auftretende Objekte wie Gras oder Bäume wurden dann durch die Designer mit prozeduralen Tools platziert. Mit diesen Tools kann unter vorgegebenen Rahmenbedingungen wie Dichte, Größe und Radius die einzelnen Objekte zufällig auf dem Terrain

platziert werden. So kann der Designer bestimmen, welche Flächen mit welcher Vegetation bewachsen sind, muss jedoch nicht jede Pflanze einzeln platzieren. Daraus lässt sich ableiten, dass auch wenn das Kernspiel nicht prozedural aufgebaut ist, solche Systeme bei der Erstellung großer Mengen an Inhalten sehr hilfreich sein können. Ein großer Vorteil der Systeme ist weiterhin, dass sie oft modular aufgebaut sind. Das bedeutet, wenn ein Teil manuell verbessert wird, sich diese Arbeit um ein Vielfaches multipliziert auf die verschiedenen Instanzen im System auswirkt. Prozedurale Generatoren sorgen auch für einen immensen Wiederspielwert, wie das schon 2006 in seiner Urform erschienene Dwarf Fortress von Tarn und Zach Adams beweist. Hier wird bei jedem Spielstart eine neue Fantasywelt generiert. Eine andere Vorgehensweise wäre, dass das Spiel so angelegt wird, dass der Spieler ein festes Level wieder und wieder spielt. Der Generator füllt es aber bei jedem Anlauf mit neuen Objekten und Gegnern an immer neuen Stellen und neuen Kombinationen. Besonders Rouge Like Spiele nutzen dieses Verfahren sehr oft, um einen stärkeren Wiederspielwert zu erzeugen. (vgl. Short und Adams 2017: S.9) Durch solche Systeme können Spiele entstehen, die nie durch das manuelle erstellen der Inhalte aufgrund des immensen Aufwandes möglich wären. Auch bieten solche Systeme technische Vorteile; so benötigen Spiele, die solche Generatoren benutzen, viel weniger Speicherplatz. Das kann sehr hilfreich sein, wenn zum Beispiel Planeten mit der realen Größe von Planeten, wie in No Man's Sky oder Star Citizen, erzeugt werden sollen. Solch großen Objekte benötigen nicht nur eine riesige Datenmenge allein für die Speicherung der Form des Körpers, vielmehr stellen die speicherintensiven Texturen das größte Problem dar. Um die Texturen für solche eine große Fläche statisch zu speichern, würden Unmengen an Speicherplatz benötigt, viel mehr, als eine moderne Blue Ray zum Beispiel zur Verfügung hätte. Nur durch das Generieren solcher Texturen und Formen ist es möglich, solch große Körper in einem Spiel in Echtzeit abzubilden. Star Citizen zeigt, wie die Kombination von prozeduralen Inhalten und manueller Bearbeitung zu qualitativen und glaubwürdigen Ergebnissen führt. Doch die Anwendung von prozeduralen Inhalten bietet im Gegensatz zum klassischen Ansatz nicht nur Vorteile.

### 4.3 Nachteile

Das Benutzen von prozeduralen Inhalten sorgt nicht immer für die gewünschte Zeiterparnis. Es kann mitunter sehr kompliziert sein, das System so anzupassen, dass es die gewünschten Inhalte erstellt. So kann es unter Umständen für den Entwickler einfacher sein, die Elemente selbst einzeln zu erstellen. Auch ist der enorme Testaufwand bei solchen Systemen zu beachten. Ein System, das eine große Menge an Inhalten produziert, erzeugt genauso viele Testfälle. Es ist davon auszugehen, dass wenn für die Erstellung der Inhalte ein prozedurales System verwendet wird, nicht die Zeit zur Verfügung steht, diese Inhalte einzeln zu erstellen. Das bedeutet auch, dass die Inhalte nicht alle einzeln getestet werden können. Hier heraus ergibt sich das Problem, dass die

Inhalte durch den Zeitmangel auch automatisiert getestet werden müssen. Die Erstellung eines automatisierten Testprogramms ist eine komplexe und zeitaufwendige Aufgabe. Zudem sind automatisierte Systeme nicht zu kreativem Denken befähigt. Demzufolge können sie die Inhalte nicht so wie ein menschlicher Tester überprüfen. Dieser könnte etwa, wenn er ein Performanceproblem erkennt, sofort versuchen dieses weiter auszuloten. Die Benutzung von prozedural generierten Inhalten bringt viele Punkte mit sich, die bei der Planung beachtet werden müssen. Dass die benötigten Spieldaten so klein ausfallen, darf dabei jedoch nicht nur als Vorteil betrachtet werden. Durch den besonderen Aufbau der Daten, den der Generator benötigt, sind diese sehr komplex und voneinander abhängig. Dies ist am Aufbau der Daten, die für die Erstellung von Kreaturen in No Mans Sky nötig ist, zu sehen. So muss nicht nur genau definiert werden, was generiert wird, sondern auch wann. Der Generator zum Beispiel muss wissen, in welcher Reihenfolge er die Inhalte generieren soll. So wird zum Beispiel die Welt direkt um den Spieler generiert. Der Spieler muss zu der Zeit aber schon von Missionszielen oder interessanten Orten in größerer Entfernung wissen. Auch muss alles generiert werden, was der Spieler möglicherweise erblicken könnte, denn es ist nicht vorauszusagen, wie der Spieler sich verhalten wird. Ein weiterer wichtiger Aspekt ist die Erhaltung der Daten im Speicher des Systems. So muss ständig entschieden werden, welche Inhalte nicht mehr relevant sind und welche noch einmal Verwendung finden. Dies muss erfolgen, da nicht unbegrenzt Speicher zur Verfügung steht. Bei den Tests muss neben der richtigen Funktion und Darstellung also auch auf die Speichernutzung der generierten Inhalte geachtet werden. In keinem Fall darf eine Kombination oder Variation der Inhalte das zur Verfügung stehende Speicherkontingent übersteigen. Aus den zuvor genannten Vor- und Nachteilen gilt es jetzt die Herangehensweise zu finden, in der möglichst viele Nachteile minimiert und dabei möglichst viele Vorteile genutzt werden. Diese Herangehensweise wird im folgenden Abschnitt beschrieben.

## 4.4 Optimale Nutzung für kleine Teams

Die Kernmechaniken an ein prozedurales System zu koppeln, kann sehr riskant sein, wie es am Beispiel von No Man's Sky zu erkennen ist. Bei einer Eigenfinanzierung ist es mitunter nicht möglich oder ratsam solche Risiken einzugehen. Bei einer länger andauernden Entwicklung gibt es, abgesehen von Krediten, kaum eine Möglichkeit mehr Kapital zu beschaffen. Es ist zu bedenken, dass durch die Verwendung prozeduraler Generatoren sowie automatisierter Testprogramme viel anspruchsvolle Arbeit für die Programmierer entsteht. Dies muss unbedingt bedacht werden und dementsprechend das Team zusammengestellt werden. Ein solches System sollte nur für die Kernmechaniken verwendet werden, wenn der Generator klar definiert werden kann, es eine enorme Anzahl an Levels oder Inhalten geben soll und dadurch für einen starken Wiederspielwert gesorgt wird. Sollte eine prozedural erstellte Kernmechanik jedoch unabdingbar für

das jeweilige Spiel sein, muss das restliche Spiel um diese Mechanik herum erstellt werden. Da sich mit der Entwicklung das Verhalten des Generators ändert, müssen auch alle Systeme um den Generator wiederholt angepasst werden. Ratsamer ist es demzufolge, prozedurale Inhalte mit manuell erstellten zu vermischen.

*„Bolstering a procedurally generated play environment with handcrafted content can add variety and depth to your players' experience while at the same time guiding them to play and explore in a method that uses the best part of your game system“* (Short und Adams 2017: S.63)

Jim Shepard weist hier im Werk von Short und Adams daraufhin, dass das manuelle Platzieren von Inhalten in einer prozedural erstellten Welt ungemein zu deren Abwechslungsreichtum und Tiefe beiträgt. Das Vermischen von prozeduralen und manuell erstellten Inhalten erweist sich als hervorragende Methode, wie das Beispiel Star Citizen beweist. So kann das Risiko durch die prozeduralen Generatoren minimiert werden und trotzdem schnell große Mengen Inhalte erstellt werden. Zudem können sehr große Objekte wie Planeten abgebildet werden. Hierfür werden die Inhalte einmal in der Entwicklung generiert, anstatt jedes Mal direkt im System des Spielers. Besonders für kleine Teams bieten solche Systeme große Vorteile. So können etwa Unreal Blueprints ideal in ein solches System eingebunden werden. Die Veränderung kann direkt im Blueprint vorgenommen werden und wird anschließend in jeder vom Generator erzeugten Instanz verwendet. Prozedurale Inhalte, die in Echtzeit auf dem System des Spielers generiert werden, funktionieren am besten in abgeschlossenen Systemen, die klar abgegrenzt und definiert sind. So wie etwa prozedural generierte Musik. Diese Abgrenzung minimiert Risiken und erleichtert es, Testfälle zu überprüfen. Auch ist so die Speichernutzung einfacher zu skalieren und zu verwalten. So weiß der Code zum Beispiel zu jeder Zeit die Beschaffenheit der zuvor definierten Landschaft und in welcher Reihenfolge sie geladen werden muss. Abgeschlossenen Systemen wie der Musik wird ein Fensteranteil des Speichers zugewiesen. Dies ist möglich, da hier sehr genau vorausgesagt werden kann, wie viel Speicher zu welcher Zeit benötigt wird. So kann sehr effektiv bei geringem Risiko für das Projekt gearbeitet werden. Dennoch können die Vorteile von prozeduralen Inhalten genutzt werden. Resultierend lässt sich sagen, dass prozedurale Inhalte Designer und Programmierer nicht ersetzen können. In der Entwicklung sollte demnach nicht ausschließlich auf prozedurale Inhalte gesetzt werden. Es liegt hauptsächlich an den Entwicklern, möglichst effektiv zu arbeiten. Hierfür sind einige Punkte entscheidend für den Erfolg des Projektes.



## 5 Neue Ansätze

### 5.1 Programmierer und Designer

Ein Team, das Videospiele entwickelt, besteht aus einer großen Zahl an unterschiedlichen Mitgliedern, wie Programmierern, Artists, Designern, Animatoren und Produzenten. In großen Produktionen, in denen hunderte von Entwicklern mitarbeiten, sind die einzelnen Mitglieder konkret auf eine Aufgabe spezifiziert. Auch gibt es für eine spezifische Aufgabe mitunter mehrere Entwickler. So gibt es zum Beispiel nicht nur Programmierer, sondern AI-, Animation-, Gameplay-, Grafik- und Netzwerkprogrammierer. In kleineren Projekten, wie im hier angestrebten AAA Indie Modell, stehen nur kleinere Teams zu Verfügung. Daher ist es besonders wichtig, die Aufgabenverteilung und das Zusammenspiel zwischen den Entwicklern abzustimmen. In der klassischen Entwicklung war für die Implementation von etwa Missionen ein Programmierer notwendig, der den Code geschrieben hat, nach dem dann die Mission abgelaufen ist. Dies ist sehr aufwendig, da der Programmierer nicht nur die Kernmechaniken des Spiels erstellen, sondern auch den eigentlichen Ablauf skripten muss. Wenn der Designer also etwas im Ablauf der Mission ändern will, benötigt er dafür die Hilfe des Programmierers. Daher haben sich andere Arbeitsweisen herausgebildet; für die meisten Entwickler ist es angestrebt, dass die Programmierer die Werkzeuge programmieren, mit denen die Designer dann die Spielinhalte erstellen können. Ein Beispiel dafür ist der Städtegenerator aus Star Citizen. (4.1.2) Mit diesem Konzept kann der Programmierer sich stärker auf die Implementation der Mechaniken konzentrieren, die dann von den Werkzeugen, die die Designer nutzen, verwendet werden. Dieses Konzept findet in der klassischen Softwaretechnik bereits Anklang und kann weiterhin durch das Nutzen von Visual Scripting enorm verstärkt werden.



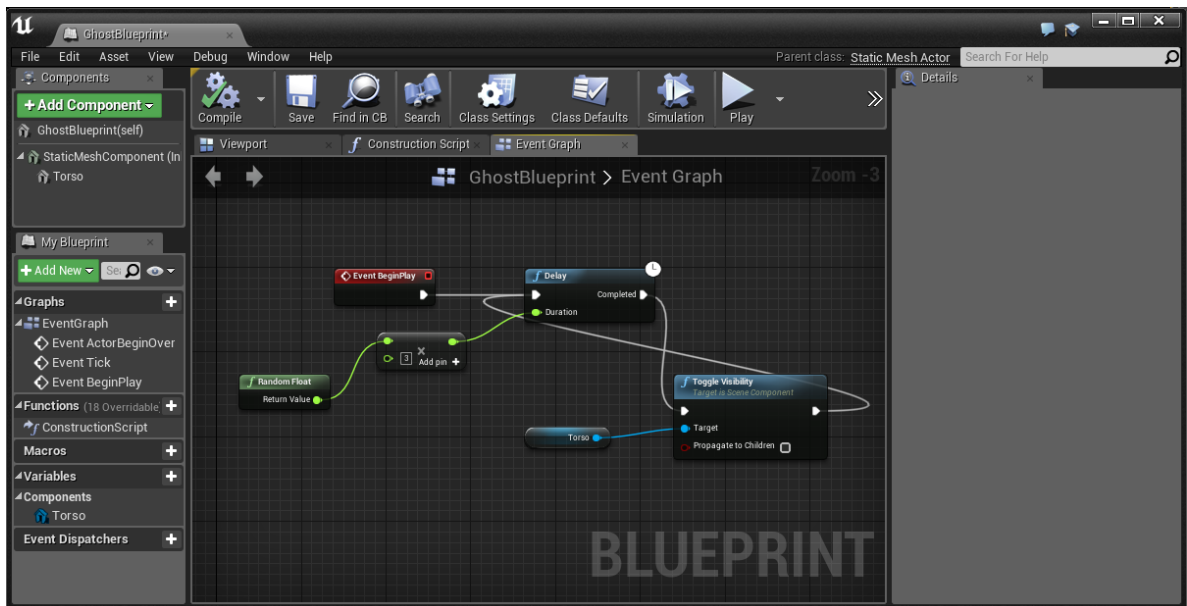


Abbildung 20 Unreal Engine Visual Scripting

In der Abbildung ist das Visual Scripting Fenster für das Blueprint System der Unreal Engine zu sehen. Mit diesem System kann über das Verbinden der Nodes, die Objekte im zentralen Fenster, programmiert werden. Dies soll es Designern, die keine Erfahrung mit Programmiersprachen haben, ermöglichen, Abläufe zu skripten. Die Unreal Engine ermöglicht es den Programmierern, Methoden in C++ zu schreiben, die dann für die Designer im Visual Scripting verfügbar sind. Diese Arbeitsweise ermöglicht es dem Team, sehr effizient und unabhängig voneinander zu arbeiten. So kann die Effektivität des Designers und des Programmierers gesteigert werden.

## 5.2 Modulare Systeme

In der Entwicklung mit sehr wenigen Entwicklern ist es nötig, auf Prinzipien der klassischen Softwaretechnik zurückzugreifen wie das Entwurfsprinzip der Modularität, bei der die entwickelten Systeme so modular und unabhängig wie möglich aufzubauen sind. So sollte, wenn ein System für Postprocessing erstellt wird, darauf geachtet werden, dass das System in jeder Szene des Spiels verwendet werden kann und nicht nur von einer bestimmten Szene abhängig ist. Wenn die Systeme auf diese Weise erstellt werden, benötigen die Entwickler für die eigentliche Erstellung länger, können aber, wenn sie flexibel genug konstruiert sind, an vielen Stellen wiederverwendet werden. Ein Steuerskript für einen Gegner kann auf diese Weise für mehrere Gegner verwendet werden. Dafür müsste man nur im Script eine Variable festlegen, wie die Geschwindigkeit der Angriffe, die ausschlaggebend für die Stärke eines Angriffs ist und somit von außen beeinflusst werden kann. Durch diese Variable wäre es möglich, große, langsame Gegner zu steuern und kleine schnelle. Auf diese Weise werden Systeme zum Beispiel nur

während der Implementation der ersten Level konstruiert. Alle weiteren Level entstehen durch eine Modifikation bereits bestehender, kombinierbarer Bausteine. Dieses Prinzip lässt sich gut mit dem Ansatz zwischen Programmierern und Designern verbinden. Wenn die Systeme möglichst unabhängig voneinander arbeiten, können sie schneller und einfacher implementiert und verlustfreier von den Designern genutzt werden. Für die Designer selbst ist der modulare Ansatz daher ebenfalls sehr hilfreich, da hierbei vergleichsweise wenige Teile der Spielwelt mit viel Aufwand auf ein AAA Niveau gebracht werden. Diese Arbeitsweise ermöglicht es, mit wenigen Designern große Level und Spielwelten zu erstellen, um die Spielwelt durch Detail glaubhaft zu gestalten. Damit die Objekte nicht nur kopiert und das Level zu gleichförmig aussieht, müssen die Objekte von einem erfahrenen Designer positioniert, rotiert und verformt werden. Auch kann so ein Objekt, indem es in einer anderen Größe wiederverwendet wird, als ein neues Objekt wahrgenommen werden. Dies kann auch durch verschiedenes Texturieren gehandhabt werden. Objekte erscheinen mit anderen Materialien als vollkommen neue Objekte. So kann durch eine intelligente Kombination aus einem Modell in verschiedenen Positionen, Größen und Rotationen sowie verschiedenen Materialien eine Vielzahl für den Spieler unterschiedlicher Objekte erzeugt werden. Diese Arbeitsweisen ermöglichen eine starke Steigerung der Effektivität einzelner Entwickler. Ein Titel hat diese Form der Aufgabenbewältigung schon umgesetzt - Hellblade. Hellblade ist daher so wichtig für das hier angestrebte Konzept, da es der einzige Titel ist, der schon eine komplette Entwicklung in dieser Art durchlaufen hat. Aus diesem Grund können aus den Vorgehensweisen und Techniken in diesem Projekt wichtige Informationen für das Konzept gewonnen werden.

## 5.3 Hellblade und Independent AAA

### 5.3.1 Projektkonzept

Ninja Theory entwickelte von 2014 bis 2017 Hellblade Senuas Sacrifice unter ihrem Independent AAA Modell und dokumentierten die Entwicklung in einer Videoreihe. Die Entwickler haben das Spiel selbst finanziert, entwickelt und gepublished. Für dieses Konzept waren einige Faktoren äußerst entscheidend. So ist das Spiel, welches am 8. August 2017 erschienen ist, nur digital erhältlich. Dieser Umstand ermöglicht eine Umsetzung ohne Publisher, da somit der Vertrieb entfällt. Das Projekt ist so konzipiert, dass es lediglich mit nur um die 15 bis 20 Entwicklern auskommen sollte (*Ninja Theory 2017b*) und demnach nur einen Bruchteil der klassischen Entwicklungskosten zu benötigt.

„By keeping the team small and the costs low. It means that we only needs to sell two-hundred or threehundredthousand copys of he game in ordert o make the development budged back.“ (Matthews Dominic 2014)

Bei einem Verkaufspreis von 30€ pro Kopie liegt das Budget für Ninja Theory bei circa 6 bis 9 Millionen Euro. Da das Spiel nur die Hälfte der Spielzeit eines Vollprestitels anpeilt, wird es auch nur für die Hälfte verkauft.

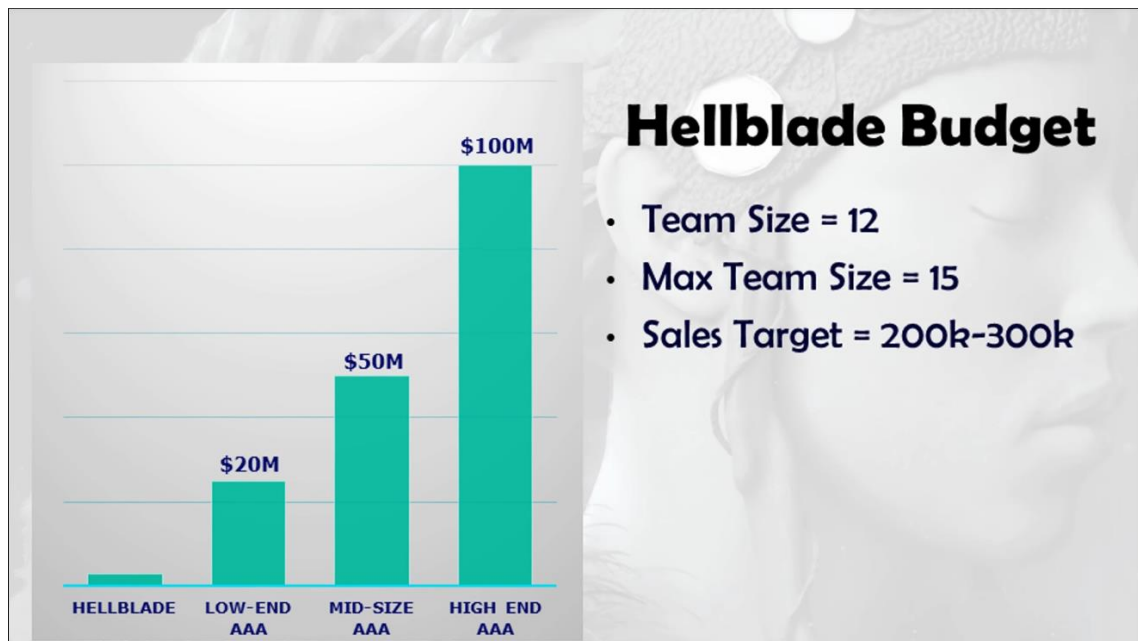


Abbildung 21 Hellblade Budget(Antoniades Tameem, 2014)

Das Spiel selbst soll jedoch die Qualitätsmerkmale einer AAA Produktion besitzen. Große Produktionen müssen viele Exemplare verkaufen, um die hohen Produktionskosten zu rechtfertigen, was wiederum bedeutet, dass die Spiele für eine möglichst große Allgemeinheit ansprechend sein müssen. Selbst für ein AAA Titel am Low-End werden laut dem Chief Creative Developer Tameem Antoniades um die 20 Millionen Dollar bis hin zu mehreren hundert Millionen benötigt. Aber durch die Notwendigkeit nur wenige Kopien verkaufen zu müssen, um die Produktionskosten wieder abzudecken, können Spielkonzepte besonders für kleine Zielgruppen umgesetzt werden. So ergeben sich Möglichkeiten, um Themen und Konzepte umzusetzen und zu behandeln, die von großen Produktionen aufgrund der geringeren Eignung für die Allgemeinheit nicht möglich wären.

Durch die strenge Limitation des Budgets sind für die einzelnen Bereiche der Entwicklung jeweils nur einzelne Personen verantwortlich, die kreative Lösungen finden müssen, um ihre Aufgaben in den vorgegebenen Rahmenbedingungen umzusetzen. Das hat Ninja Theory dazu gezwungen, mit neuartigen Herangehensweisen zu arbeiten. Das

Konzept von Ninja Theory beruht somit auf einer starken Verbindung zwischen den Entwicklern und den Spielern. Da das Entwicklerstudio keine finanzielle Sicherheit in Form eines Publishers vorweisen kann, ist es auf die Beziehung zu den Spielern angewiesen, da sie direkter Feedback geben und dadurch indirekt als Tester herangezogen werden können. Zudem können sie das Projekt zu einem gewissen Teil durch das Kaufen von Merchandise-Artikeln mitfinanzieren.

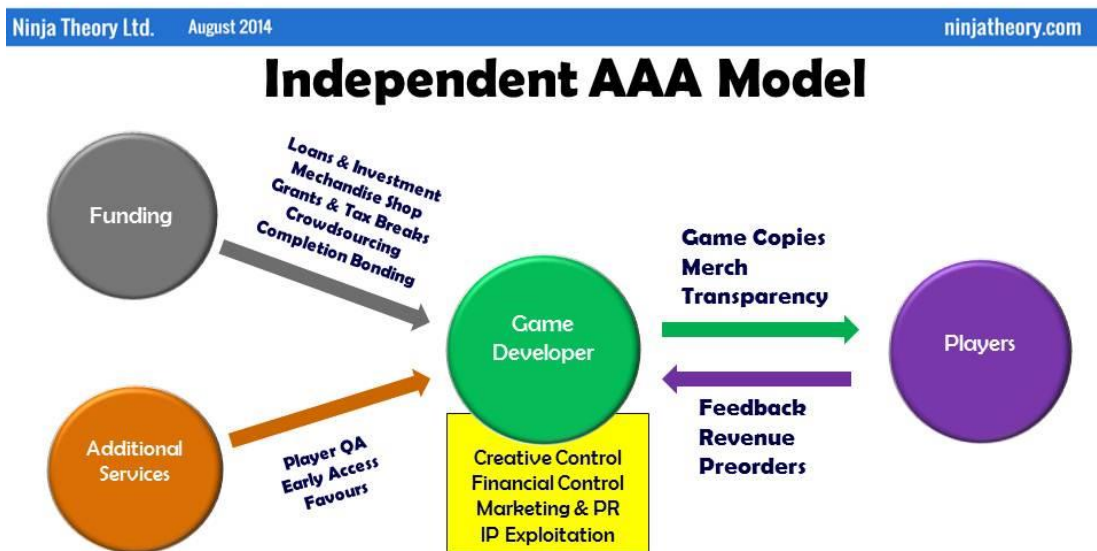


Abbildung 22 Independent AAA Model (Antoniades Tameem Ltd 2014)

Das Entwicklerteam bietet den Spielern nicht erst zum Ende der Entwicklung eine Kopie des Spiels, sondern sorgt mit der Videoreihe, die sich mit der Entwicklung beschäftigt, für Transparenz. Diese Transparenz sorgte während der Entwicklung für gute PR. Auch wurden viel der Testarbeiten an Spieler außerhalb von Ninja Theory ausgelagert.

### 5.3.2 Technische Lösungen

Um im Budget des Independent AAA Modells zu bleiben, musste Ninja Theory einige Aspekte der Technik selbst umsetzen. Bei klassischen AAA Produktionen wird der Dialog von einem Voice Actor gesprochen, die Motion Capture Daten von einem weiteren Schauspieler performt und Animator animieren dann das Gesicht der Spielfigur und die Bewegung der Kamera Frame für Frame für die Zwischensequenzen. Da dies extrem aufwendig zu synchronisieren ist, entwickelte sich daraus das Performancecapture. Hier werden Körper- und Gesichtsbewegungen sowie der gesprochene Text parallel aufgenommen und gleichzeitig von einem Schauspieler performt. Dies führt zu glaubwürdigeren Ergebnissen, da der Schauspieler während der Bewegungen sprechen muss, sodass Anstrengung in der Stimme nicht simuliert werden muss. Für diese Technik wird

im Normalfall ein speziell ausgestattetes Performance Capture Studio benötigt. Diese sind jedoch sehr teuer, da nicht nur die Technik benötigt wird, sondern auch viele Mitarbeiter, die die Aufnahmen betreuen und die Daten auswerten. Zudem kann es mitunter Monate dauern, bis die Daten wieder beim Entwickler angekommen sind. Sollte dann Unstimmigkeiten auftreten, ist es enorm kostspielig, neue Aufnahmen zu drehen, um etwaige Fehler zu korrigieren. Eine manuelle Anpassung der Daten wäre ebenfalls zu aufwendig. Da Hellblade bestrebt war, die Qualität von AAA Titeln halten zu wollen, war Performancecapture jedoch die einzige Methode, um schnell qualitative und realistische Zwischensequenzen zu erhalten.

Für das Motioncapturing wurden 12 Bonita Kameras von Vicon, einem Unternehmen, das Capture Hard- und Software vertreibt, benutzt. Diese werden mit einem Server verbunden, der die Daten sammelt. Um die Kosten weiter niedrig zu halten, wurden die Kameras nicht auf professionellen Stativen montiert, sondern auf einfachen Kleiderständern.

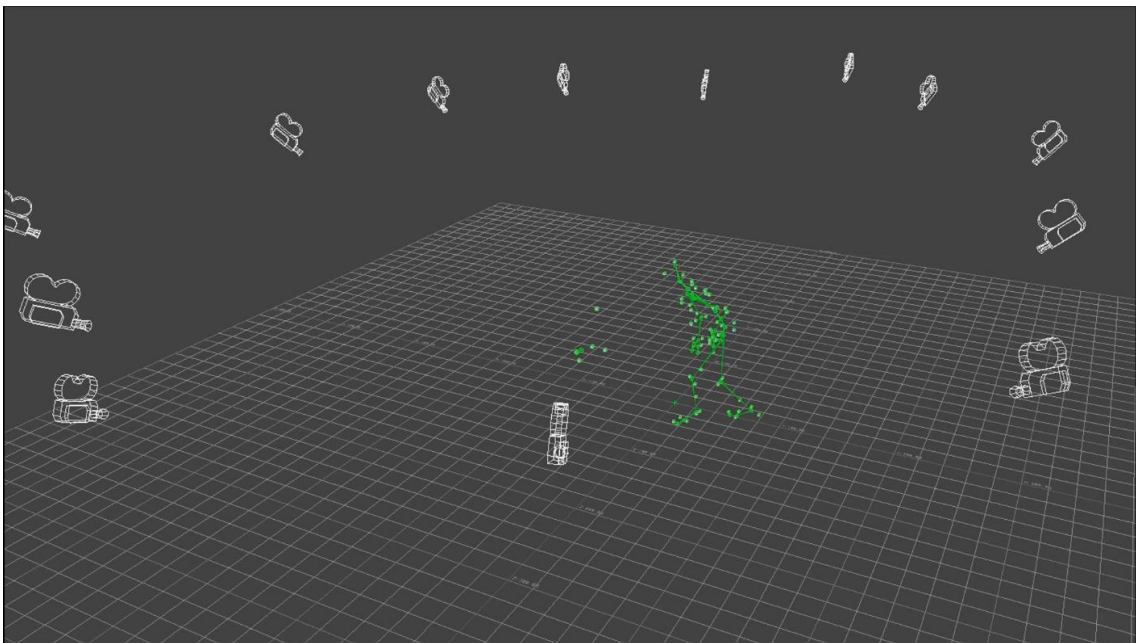


Abbildung 23 Hellblade Motioncapture mit Bonita Kamera (Ninja Theory Ltd. 2015a)

Wie in dem Bild klar zu erkennen, wird bei der Aufnahme der Bewegungsdaten nicht nur die Position der Spielfigur erfasst und aufgezeichnet, sondern auch die Position und Rotation der Kamera, die im fertigen Spiel das Geschehen in der Zwischensequenz rendern wird. So können nicht nur in einem Durchlauf mehr Daten gesammelt werden, sondern der Chief Creative Developer kann auf diese Art selbst genau die Bewegungen der Kamera bestimmen und gestalten.



Um die Gesichtsausdrücke aufzunehmen, sind gesonderte Kameras von Nöten, die immer direkt auf das Gesicht des Schauspielers gerichtet sind. Diese erfasst mit einem ähnlichen Verfahren wie die Motioncapture Kameras Trackingpoints, nur dass diese in viel höherer Dichte auf dem Gesicht der Schauspieler platziert werden. Die Trackingpoints beim Facecapture sind jedoch nur auf die Haut aufgemalte Markierungen und keine dreidimensionalen Objekte, wobei die Bewegung der Haut aufgezeichnet wird. Durch solch genauen Gesichtsausdrücke wird der Spielercharakter um ein vielfaches realistischer und glaubwürdiger. Für die Nutzung dieser Daten wird jedoch nicht nur der Kameraaufbau benötigt, sondern zudem eine Software. Im Falle der Gesichtsdaten wäre dies ein Facial Solver, der den Videofeed in 3D Daten interpretiert und wandelt. Im Gegensatz zum Motioncapture werden beim Aufnehmen von Gesichtsausdrücken nicht nur die Trackingpoints aufgenommen, da hier nicht nur die Position der Gelenke für die letztendliche Animation wichtig sind. Bei der Animation des Gesichts ist es auch wichtig, wie die Gesichtszüge und Haut sich bei verschiedenen Ausdrücken verhält und verändert, wofür primär die Trackingpoints als Referenzpunkte dienen.



Abbildung 24 Hellblade Performancecapture (Ninja Theory Ltd. 2015b)

In der Abbildung ist der Konferenzraum zu sehen, der von Ninja Theory zu einem Motioncapture Studio umgebaut wurde. Da gleichmäßiges Licht für die Aufnahme des Gesichts benötigt wird, wird dieses von einfachen LED Panels erzeugt, anstatt von Studiostrahlern. Da beim Performancecapture auch der Ton aufgenommen wird, musste auch die Schalldämpfung verbessert werden, an dieser Stelle realisiert durch einfache Sportmatten auf dem Boden. An diesem Beispiel ist gut zu erkennen, wie bei einer Entwicklung mit stark limitierten, finanziellen und personellen Mitteln schwierig und teuer erscheinenden Konzepte und Verfahren vereinfacht umgesetzt werden können.

„We also hope to show that a small Team with limited means is still able to capture high end karakter performances. And that is something that up until now has only be the preserve of big budget AAA Studios.“ (Antoniades Tameem 2014)

Für eine realistische Darstellung des Gesichts mit den Daten aus dem Performance Capture sind ebenfalls realistische Texturen notwendig, damit die Haut in verschiedenen Kontraktionen realistisch aussieht. Um Oberflächen realistisch zu rendern, benötigt eine Engine mehrere Texturen, auch Maps genannt. Neben einer Map, die die reinen Farbinformationen speichert, werden wieder andere genutzt, die zum Beispiel den Grad der Reflektion oder die Oberflächenbeschaffenheit beschreiben. Um zu berechnen, wie einfallendes Licht auf der Oberfläche dargestellt wird, wird oft eine Normalmap genutzt. In dieser sind die mathematischen Normalen für jeden Punkt der Oberfläche gespeichert. Mit diesen Informationen werden die Farbwerte der Grundtextur manipuliert, um die Illusion von Erhebungen durch einfallendes Licht zu erzeugen. Diese Technik wird nicht nur für Haut benutzt, sondern auch für andere Oberflächen. Ninja Theory hat diese Daten In House mit minimalem finanziellen mit einem Prototyp namens Plant-Pot Aufwand aufgenommen.

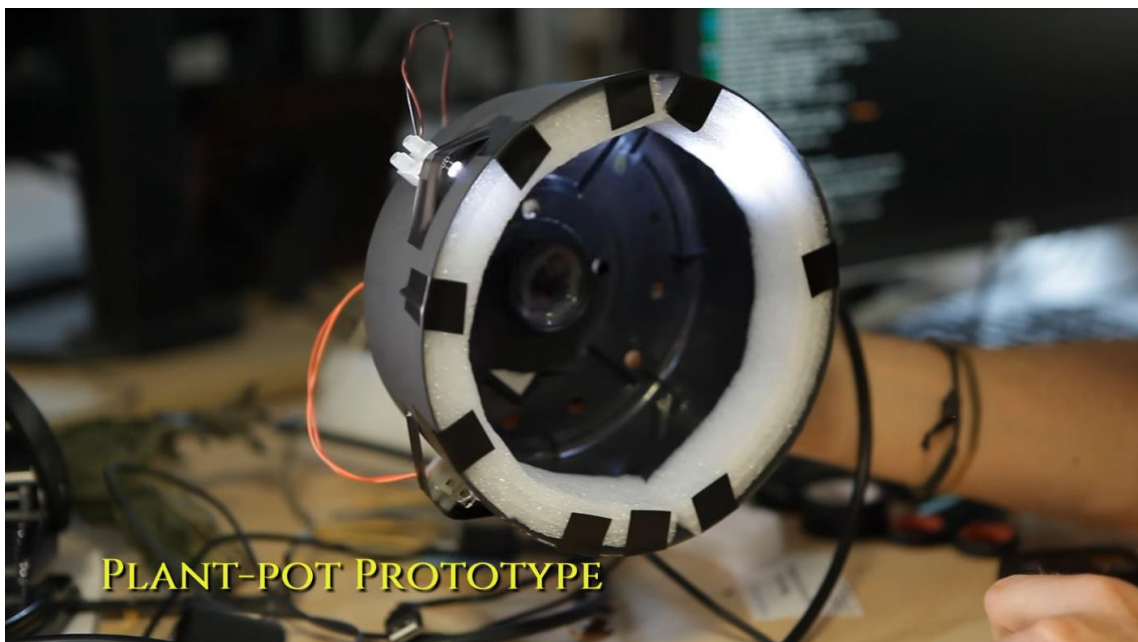


Abbildung 25 Hellbalde Plant-Pot Scanner (Ninja Theory Ltd. 2015a)

Das in der Abbildung zu sehende Gerät besteht aus einem Blumentopf - daher der Name - einer Kamera und vier LED's. Diese Bauteile sind mit einem Raspberry Pie Minicomputer verbunden, um die Lichter einzeln anzusteuern und die Daten aus der Kamera zu speichern. Die Kamera nimmt von jeder Oberfläche, auf die der Plantpot gesetzt wird, Fotos mit Licht aus verschiedenen Richtungen auf. Aus den Unterschieden in den Fotos kann dann errechnet werden, welche Stelle wann dem Licht ausgesetzt war und damit

in welcher Richtung die Normalen an dieser Stelle stehen. So kann eine hochauflösende Normalmap erstellt werden. Diese Funktionalität lässt sich jedoch nicht nur auf Haut anwenden, sondern auf jedes Material welches fotografiert werden kann. Das Erzeugen von Normalmaps mit Hilfe dieser Methodik spart einerseits Aufwand, der sonst von Designern manuell betrieben werden müsste und liefert im Resultat andererseits realistische Materialien.

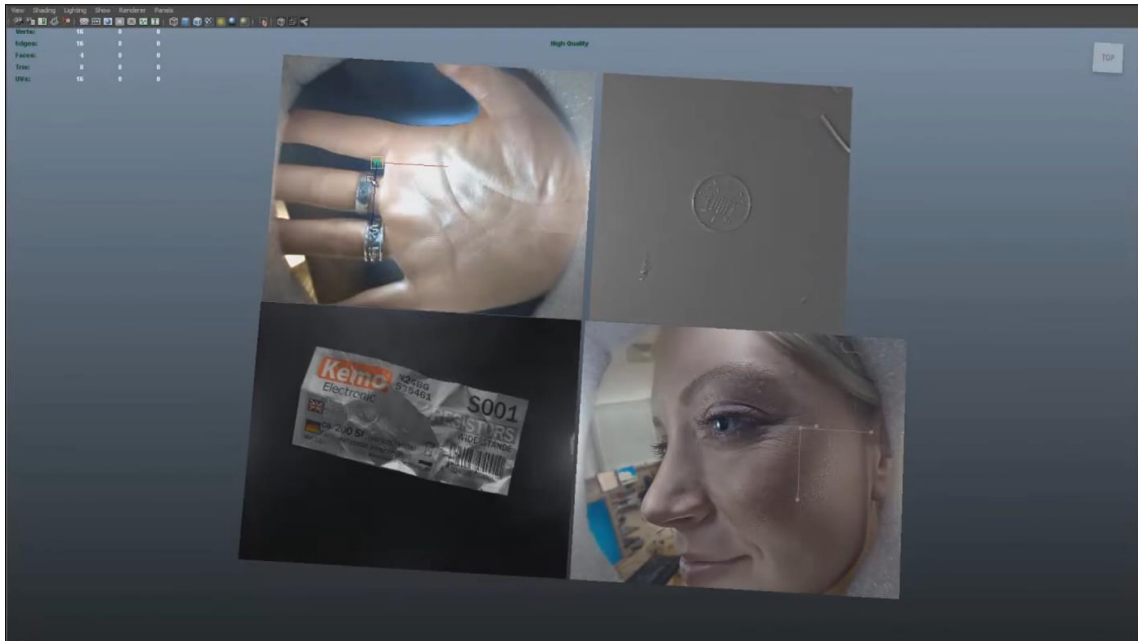


Abbildung 26 Hellblade Normalmap Scans (Ninja Theory Ltd. 2015a)

Die Abbildung zeigt verschiedene Objekte und wie sie unterschiedlich von den Lichtern im Plant-Pot beleuchtet werden. Die Abschattung in den nicht vom Licht erreichten Bereichen sind zu erkennen, aus welchen sich dann die Erhebungen errechnen lassen. Aus diesem Beispiel lässt sich ableiten, dass das Erstellen von High-End Inhalten nicht nur durch ein immenses Kapital und professioneller Hardware möglich ist. Die Entwicklung von Hellblade beweist, dass es unter diesen Umständen möglich ist, ein erfolgreiches Spiel mit AAA Qualitäten zu entwickeln und ausschließlich digital zu vertreiben. Nachdem die Techniken erforscht und definiert sind, muss betrachtet werden, wie das Konzept im Ganzen aufgebaut ist. Hierfür werden die gesammelten Informationen aus den vorangegangenen Kapiteln in Kapitel 6 zusammengetragen, um das angestrebte Konzept zu erstellen. Diese kritischen Faktoren werden ebenfalls in Kapitel 7 noch einmal zusammengefasst.



## 6 Das Optimale AAA Indie Modell

Die Entwicklung im AAA Indie Modell wird im Folgenden in den fünf wichtigsten Punkten anhand der abgeleiteten Informationen aus vorangegangenen Kapiteln erstellt. Das Ziel ist es, mit kreativer Freiheit ein Videospiel umzusetzen, das in seinen Mechaniken und optischen Präsentation die Qualitätsmerkmale einer AAA Entwicklung gleicht.

### 6.1 Zielgruppe und Vertrieb

Hierbei werden nicht die klassischen Größen in Budget und Marketing angestrebt. Von Free To Play ist in diesem Fall abzuraten, da in diesem Konzept Elemente aus dem Spiel entfernt werden müssten, um später nachgekauft zu werden. Oder das Spiel müsste so ausgelegt werden, dass zusätzlich eingekaufte Inhalte das Spiel erst dauerhaft spielbar machen, was jedoch mit den Arbeitsweisen, die benötigt werden, um entsprechende Inhalte im AAA Standard zu erstellen, kollidiert. Die Pay to Play Finanzierung ist vor allem für Online Games funktional, da der Spieler bei einem monatlichen Beitrag erwartet, dass das Spiel seine Fortschritte persistent speichert und er dieses auch für Monate spielen kann. Diese sind allerdings mit kleinem Budget schwer umzusetzen, besonders, weil die Menge an designten Spielinhalten schwer aufrecht zu erhalten ist. Auch müsste für Serverstrukturen gesorgt werden, was das finanzielle Risiko des Projektes weiter erhöhen würde. Als Bezahlmodell für das Spiel sollte daher eher ein Buy to Play Modell verwendet werden, da hier die Größe des Projektes sehr genau skaliert und vorausgesagt werden kann. Somit ist es einfacher, finanzielle Risiken abzuschätzen und das Projekt zu planen. Sollte ein kommerzieller Erfolg eintreten, kann durch Zusatzinhalte das Spiel erweitert werden. Folgt das Spiel einer Mechanik oder einer Narrative, umso linearer ist es und desto besser kann der Entwicklungsaufwand geschätzt werden. Dieses Spiel sollte über die schon vorhandenen digitalen Marktplätze für die drei derzeit wichtigsten Plattformen, PS4, Xbox One und PC vermarktet werden. Sich auf weitere Zielplattformen zu konzentrieren, würde den Arbeitsaufwand stark erhöhen. Diese drei können auf ähnliche Art und Weise erstellt werden und decken den größten Teil des Marktes ab. Die neueren Iterationen der Konsolen wie Xbox One X oder die PS4 Pro können auch ohne vom Entwickler getätigte Optimierung die Performance des Spiels verbessern. Zudem ist die Möglichkeit von Raubkopien auf den Konsolen, im Gegensatz zur PC Plattform, minimal. Da für das Projekt kein Publisher zur Verfügung steht, wird das Spiel nur digital vertrieben. Wenn für das Projekt weniger Kapital notwendig ist, müssen auch weniger Kopien verkauft werden, um die Entwicklungskosten zu decken und Gewinne zu erzielen. Das bedeutet, dass mit dem Spiel kleinere, spezialisierter Zielgruppen angesprochen werden können. Durch eine spezialisierte Zielgruppe kann auch der Inhalt oder das Gameplay des Spiels spezifischer sein. So ergeben sich Möglichkeiten,

um Marktlücken effektiv auszunutzen. Hellblade zeigt an dieser Stelle, wie es das konkrete Thema der Mentalen Krankheit behandelt, welches für die Allgemeinheit der Spieler eher ungewöhnlich ist. Für eine effektive Entwicklung eines Spiels unter diesem Modell ist jedoch einiges zu beachten.

## 6.2 Finanzierung

Bei einer Indie Entwicklung wird nicht auf externes Kapital verzichtet, denn ohne dieses wäre eine professionelle Entwicklung nicht möglich. Die Unabhängigkeit besteht darin, dass der Entwickler selbst die kreative Freiheit und die Entscheidungsgewalt im Projekt besitzt. Daher ist eine Indie Entwicklung unter Umständen wie bei den EA Originals auch unter einem Publisher möglich. Jedoch stützen sich Indie Entwickler im Normalfall auf andere Quellen. Die Finanzierung soll in diesem Konzept in zwei Schritten erfolgen. Der erste ist eine Konzeptphase, in dem minimale Mittel benötigt werden. Hier wird das Konzept für das Spiel erstellt, vorbereitet und gestartet, wobei das nötige Budget zunächst durch Eigenkapital oder einer Förderung beziehungsweise eines Micro Kredites erfolgt. Durch moderne Finanzierungsmethoden wie etwa eine Crowdfunding Kampagne können zusätzliche Geldmittel erreicht werden. Dieser Schritt soll als Testphase dienen. Im speziellen Fall der Entwicklung in Deutschland ist anzumerken, dass es keine bundesweite Förderung für Videospiele gibt, daher ist die Lage für Entwickler in Deutschland als schwierig zu beschreiben. Auch ist in den Förderprogrammen auf Landesebene nicht die Rede von Games, wenn, dann lediglich als ein Bestandteil der Filmförderung im jeweiligen Bundesland. Der Bundesverband Interaktive Unterhaltungssoftware ist ein wichtiger Ansprechpartner für die Förderungen und bietet Informationen zu den einzelnen Möglichkeiten. Lutz Anderie führt in seinem Buch „Games Industry Management“ einige Beispiele für Start Up Förderungen an: „Die BMH Beteiligungs-Managementgesellschaft Hessen schafft mit ihren vielfältigen Fonds in Form von stillen oder auch offenen Beteiligungen gute Voraussetzung für ein Dauerhaftes und Solides Unternehmenswachstum.“ „Der High Tech Gründerfonds der German Silicon Valley Accelerator (GSVA) finanzieren insbesondere junge Technologieunternehmen zu attraktiven Konditionen.“ (Anderie Lutz 2016: S207) Indem diese schon in diesem Schritt erstellt wird, kann hier die Crowdfunding Kampagne als Indikator für das Potenzial und den Bedarf auf dem Markt dienen. Das Crowdfunding soll einen großen Teil der Finanzierung darstellen und für weitere Investoren aufzeigen, dass die Nachfrage für ein solches Konzept auf dem Markt besteht. Hier können durch Stretchgoals weitere Einnahmen erzielt werden und somit ein Bestehen der Finanzierung garantieren. Die Stretchgoals bringen jedoch weitere Verpflichtungen, was die Inhalte des Spiels angeht, mit sich. Die reine Finanzierung durch eine Crowdfunding Kampagne ist nicht üblich, aber möglich.

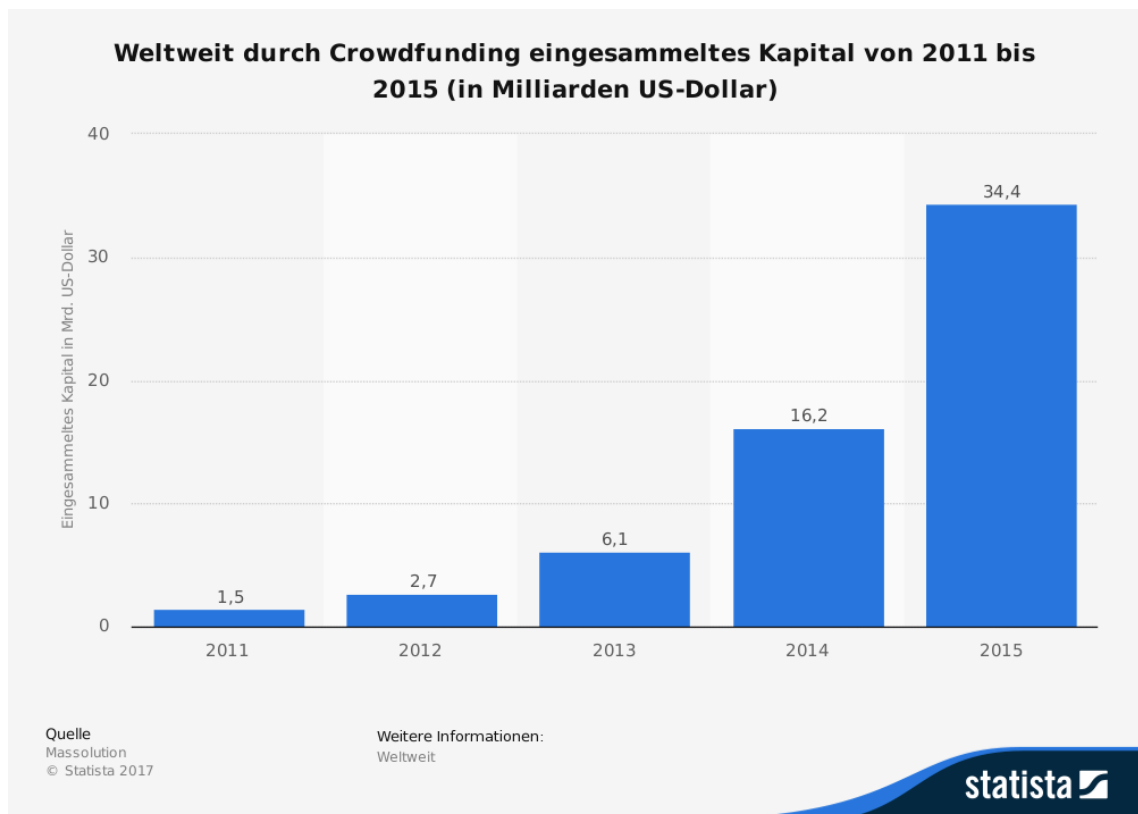


Abbildung 27 Weltweit durch Crowdfunding eingesammeltes Kapital von 2011 bis 2015 (CrowdExpert 2015)

Wie in der Statistik zu erkennen, hat sich die Bereitschaft der Menschen in eine Crowdfunding Projekt zu investieren in den Jahren 2011 bis 2015 enorm erhöht. Aus diesen Daten lässt sich ableiten, dass eine Crowdfunding Kampagne eine wichtige Einnahmequelle darstellen kann. Für die Entwicklung von Games wird aber oft mehr Geld benötigt, als eine Crowdfunding Kampagne bereitstellen kann. 2015 wollte King Art ihr Spiel „Die Zwerge“ mit einer Kickstarter Kampagne finanzieren: „260.000 US-Dollar erbittet sich das Team von der Community. Das stellt laut Informationen auf der Kickstarter Seite rund 15 Prozent des Gesamten Entwicklungsbudgets.“ (Anderie Lutz 2016: S205) Anhand dieses Beispiels ist zu erkennen, dass Entwickler ihre Projekte nur teilweise durch Crowdfunding finanzieren, wobei der Rest des Kapitals durch Investoren gedeckt werden muss.

Im zweiten Schritt der Finanzierung muss durch eine Mischung aus klassischen und modernen Finanzierungsmethoden das restliche Kapital bezogen werden. Als klassische Methoden sind Investoren und Kredite zu betrachten, die meistens nur erreicht werden können, wenn die Crowdfunding Kampagne bisher erfolgreich war. Mit diesem erreichten Ziel können Investoren und mögliche Kreditunternehmen von den Möglichkeiten des Projektes überzeugt werden. Die Produktion kann wie bei Hellblade durch weitere Wege

wie das Verkaufen von Fanartikeln finanziert werden. Wenn die Finanzierung nicht vollständig durch einen großen Investor wie einen Publisher erfolgt, ist es, wie die Beispiele aus selbstfinanzierten Projekten zeigen, ratsam, das Projekt aus vielen verschiedenen Quellen zu finanzieren. So besteht nicht nur weniger Abhängigkeit von einer Quelle, auch ist es einfacher, potenzielle Geldgeber von kleineren Investitionen zu überzeugen. Sollte das Projekt gegen Ende jedoch noch weitere Geldmittel benötigen, kann eine Early Access Phase in Betracht gezogen werden. In dieser könnten Spieler für einen geringeren Betrag als für das fertige Spiel, das Spiel testen und im Normalfall zu Release die Vollversion erhalten. Diese Praktik erzeugt nicht nur mehr Kapital, sondern lagert auch viel Testarbeit an die Spieler aus. Der Aufwand, der bei professionellen Spielen für das Testen betrieben wird, sollte folglich nicht unterschätzt werden. Dieser Aufwand würde dann von bezahlten Testern ausgeführt, da es den eigentlichen Entwicklern aufgrund strenger Zeitpläne nicht möglich, ist viel Zeit in das Testen zu investieren.

### 6.3 Technologie

Die Anwendung der richtigen Technologie ist für das Konzept entscheidend, da moderne Engines die Entwicklung des Projekts enorm beschleunigen können. Für dieses Projekt muss eine schon bestehende Engine genutzt werden, da die Entwicklung ansonsten mit den begrenzten Mitteln nicht zu schaffen wäre. Zudem kann so der besonders der grafische Anspruch an AAA Qualitäten so ebenfalls gemeistert werden. Mit einer professionellen Engine, wie der zuvor gewählten Unreal Engine, steht dem Team die gleiche Technologie zur Verfügung wie großen AAA Studios, die viele ihrer Projekte ebenfalls mit der Unreal Engine umsetzen. Mit dieser oder ähnlichen Engines kann das Spiel zudem für mehrere Zielplattformen erstellt werden. Die für das Konzept wichtigen Plattformen PC, Xbox One und PS4 können von Haus aus von der Engine abgedeckt werden. So muss bis zu einem bestimmten Stand der Entwicklung nur an einer Instanz des Projektes gearbeitet werden, bis jenes die Plattformspezifischen Anpassungen erfahren muss, wie zum Beispiel die Steuerung oder das Errungenschaften System der jeweiligen Plattformen und Netzwerke. Von modernen Spielen wird erwartet, dass sie die Errungenschaften wie Steam Archivements oder Playstation Trophys beinhalten. Indem erst mit einer einzelnen Instanz des Projekts gearbeitet wird, kann der Arbeitsaufwand weiter minimiert werden. Zudem bietet eine moderne Engine wie die Unreal Engine die Möglichkeiten, moderne Arbeitsweisen wie das besondere Verhältnis zwischen Programmierern und Designer (5.1) einfacher zu gestalten, indem Funktionen wie das Visual Scripting genutzt werden. Andere Features der Unreal Engine, wie den schon integrierten Editor für Zwischensequenzen, bieten zusätzliche Hilfe. Durch das Aufkommen von preiswerten Techniken für die Aufnahme von Bewegungs- und Performancedaten von Schauspielern, wie etwa von Ninja Theory genutzt, wird es möglich, hochqualitative Se-

quenzen zu erstellen. Für einen Großteil der benötigten Daten wird nunmehr kein professionelles Motioncapture Studio mehr benötigt, da kommerzielle Endbenutzer-Hardware diese Aufgaben ebenfalls erfüllen kann. Zudem kann eine eigene Lösung besser skaliert werden, um möglichst kosteneffizient zu arbeiten. Diese Hardware steht dem Entwickler dann, im Gegensatz zu einer gemieteten Session in einem Studio, für weitere Projekte wieder zur Verfügung. Da Ninja Theory direkt mit Epic, dem Entwickler der Unreal Engine, zusammenarbeitet, ist zu erwarten, dass die Lösungen die Ninja Theory, Epic und Vicon für Hellblade erarbeitet haben, auch in die kommerzielle Unreal Version einfließen.

## 6.4 Arbeitsweisen

Für die Umsetzung des Projektes sind erfahrene Entwickler notwendig, um Arbeitszeiten zu planen und mitunter schwierige Elemente umzusetzen. Die Planung der Arbeit und der Pipelines für die Produktion von Assets ist in diesem Modell essenziell, da hier diese Probleme nicht mit einem höheren personellen Aufwand behoben werden können. Es wird viel Erfahrung benötigt, um Zeiten für Assets und Systeme genau abzuschätzen. Daher ist es eher unwahrscheinlich, diese Art von Projekt mit einem völlig unerfahrenen Team erfolgreich umzusetzen. Für die Produktion von Spielinhalten gelten weiterhin einige Regeln; so sollte darauf geachtet werden, dass die produzierten Inhalte, sollten diese viel Zeit in Anspruch nehmen, an vielen Stellen des Spiels wieder Verwendung finden, um die hohe Produktionszeit zu rechtfertigen. Die Assets müssen ebenfalls einen bestimmten Qualitätsstandard erreichen, welcher jedoch an einen spezifischen grafischen Stil gekoppelt ist. So kann ein Spiel etwa nur einfache Farben als Texturen verwenden, anstatt komplexere mehrfarbige Texturen. Möglich wäre ebenfalls die Verwendung simpler geometrische Formen. Das bedeutet, dass ein abstrakterer Stil die Produktion vereinfachen kann, ohne dabei die Ästhetik des Spiels zu mindern.



Abbildung 28 BOUND Spielgrafik

Ein Beispiel für diese Art von grafischen Stil ist das 2016 erschienene BOUND von Plastic. Hier werden nur einfache geometrische Figuren mit einer minimalen Anzahl an Polygonen für die Spielwelt verwendet, wobei die Objekte ständig von Code beeinflusst und platziert werden, um eine möglichst lebendige Spielwelt zu erschaffen. Dies erzeugt dann die Ästhetik des Spiels. Dies bedeutet jedoch wieder mehr Arbeit für Programmierer und weniger für Designer. Die Zusammenstellung des Teams ist ebenfalls vom Projekt an sich abhängig und wird weiterhin durch das AAA Indie Konzept beeinflusst. Besonders wenn prozedurale Inhalte genutzt werden sollen, welche in diesem Konzept sehr hilfreich sind, ist ein höherer Anteil an technischen Entwicklern wie Programmierern notwendig, um die Generatoren zu erstellen und anzupassen. Auch für das Testen von prozeduralen Inhalten sind mitunter Programmierer notwendig, da die Inhalte möglicherweise durch ihre enorme Anzahl, von einem automatisierten Programm getestet werden müssen. Auch im allgemeinen Fall ist ein Projekt als AAA Indie Projekt stärker von Programmierern abhängig, da diese im Allgemeinfall die Werkzeuge bereitstellen, mit denen die wenigen Designer dann sehr effektiv arbeiten sollen. Im Allgemeinen ist zu beachten, dass wenn Arbeit für Designer eingespart werden soll, diese auf Seiten der Programmierer dazukommt. Dasselbe gilt auch für die Gegenrichtung; sollte ein eigenes System für das Motioncapturing verwendet werden, muss darauf geachtet werden, dass der jeweilige Animator Erfahrung mit Motioncapture Systemen und Daten besitzt. Damit es zwischen den einzelnen Entwicklern nicht zu unnötigen Wartezeiten kommt, sollten ihre Aufgaben möglichst unabhängig voneinander zu bewältigen sein. Weiterhin sollte es im Ablaufplan des Projektes möglichst keine kritischen Pfade geben, da eine unmittelbar voneinander abhängige Aufgabenverteilung das gesamte Projekt verlangsamen können. Sollte diese unvermeidbar sein, müssen für die einzelnen Entwickler Aufgaben

als Puffer bereitstehen, indem sie zum Beispiel weitere entkoppelte Assets erstellen. Wichtig für den Erfolg des Konzepts ist die effektive Nutzung der schon vorhandenen Mittel. So sollten mitgelieferte Funktionen der verwendeten Engine auch genutzt werden. Oft liefern Unreal und Unity Asset Pakete mit, auf denen das Projekt aufgebaut werden kann, wie beispielsweise Firstperson Controller. Diese beinhalten dann schon grundlegenden Funktionen sowie die Steuerung, die für ein Firstperson-Spiel genutzt wird. In den meisten Fällen ist es daher ratsam, diese Assets als Basis für die eigene Entwicklung zu nutzen. So spart das Team nicht nur die Zeit ein, die es benötigt hätte, die grundlegenden Funktionen zu implementieren. Oft sind diese Assets direkt von den Entwicklern der Engines erstellt und daher gut optimiert und robust. Auch bieten die Engines digitale Stores, in denen Assets eingekauft werden können. Diese können genutzt werden, sollte das Team Probleme bei der Erstellung bestimmter Funktionen oder Assets haben. So kann wenn das Team noch nie ein Tag Nacht System erstellt hat dieses dort einkaufen und kommerziell nutzen und anpassen. Es kann finanziell sinnvoller sein ein Asset zu kaufen, anstatt selbst viel Entwicklungszeit zu investieren. Jedoch stellt diese Option eher eine Notlösung dar, denn Assets die selbst erstellt werden, können viel besser an den Stil des Spiels angepasst und modifiziert werden.

## 6.5 Vermarktung und Transparenz

Das Spiel sollte mindestens auf den drei wichtigsten digitalen Marktplätzen angeboten werden. Für den Vertrieb auf der PC Plattform ist Steam von Valve der wichtigste Anbieter. Steam ist nicht der größte digitale Anbieter für Videospiele im Bereich PC - dies ist der chinesische Anbieter Tencent - aber dennoch der wichtigste in der westlichen Welt. Die beiden weiteren Anbieter sind der PS Store und Xbox Games Store. Mit diesen drei werden die wichtigsten derzeitigen Plattformen für AAA Spiele abgedeckt. Die drei Plattformen PS 4, Xbox One und PC die mit den Anbietern abgedeckt sind, sind am besten für das Konzept geeignet da sie die Entsprechende Rechenleistung bieten oder im Falle eines PC's bieten können. Da dem Projekt nur ein sehr geringes Budget für Promotion zur Verfügung steht muss jegliche Werbung über soziale Medien geschehen. Diese bieten eine immense Reichweite an, ohne dabei weitere Kosten für den Entwickler zu verursachen. Auch hier sollten die wichtigsten abgedeckt sein.

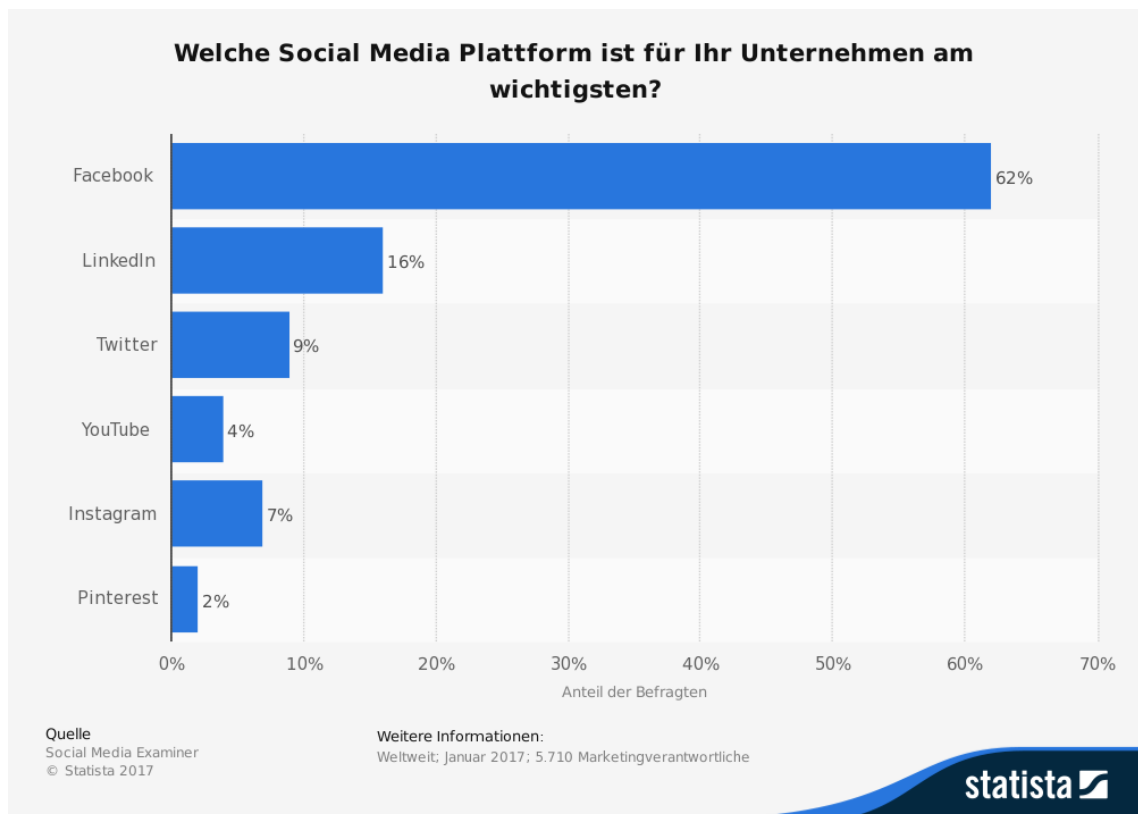


Abbildung 29 Welche Social Media Plattform ist für ihr Unternehmen am wichtigsten. (Social Media Examiner (2017))

Auf der Grafik ist zu sehen, dass für die meisten Unternehmen Facebook mit Abstand das wichtigste soziale Netzwerk ist. Facebook stellt mit der Möglichkeit, eine eigene Seite für das Spiel zu erstellen und direkt mit den Spielern in Kontakt zu bleiben, eine weitreichende Plattform für die PR-Arbeit dar. Als zweitwichtigste Instanz gilt LinkedIn, ein Netzwerk für berufliche Kontakte. Dieses kann besonders für ein Videospiel sehr von Vorteil sein, weil hier nicht unbedingt die Spieler, aber andere einflussreiche Entwickler erreicht werden können. Der stetige Kontakt zu den Spielern ist für ein solches Projekt sehr wichtig, da die Menge an laufend veröffentlichten Titeln sehr groß ist und daher der Spieler an das Spiel gebunden werden muss. Durch eine Crowdfunding Kampagne ist der Entwickler inoffiziell dazu verpflichtet, den Spielern, die schon Geld in das Projekt investiert haben, Auskunft über den Entwicklungsstand des Projektes zu geben. Die Hellballe Entwickler Ninja Theory vollführten beispielsweise eine ungewöhnliche Marketingstrategie, in der sie den Spielern durch eine Videoserie über die Entwicklung schon frühzeitig einen Einblick in das Projekt erlaubten. Diese sogenannten Development Diaries gaben den Spielern von Anfang an einen tiefen Einblick in die Entwicklung und führte letztendlich zu einem Anstieg der Popularität des Projektes bereits vor dem Release.



Zahlreiche Videos und Social Media Post konnten infolgedessen online aufgerufen werden, für dessen Produktion eine interne Mitarbeiterin verantwortlich war. So müssen keine extra Mittel für eine Agentur oder ein Filmteam ausgegeben, zudem eine interne Produktion besser kontrolliert werden kann, hinsichtlich der Veröffentlichung brisanter Informationen nach außen. Mit dieser Art von PR Arbeit wird jedoch weiterhin erreicht, dass sich die Spieler persönlicher angesprochen fühlen, als durch eine Pressemitteilung eines Medienvertreters. Das Entwicklerstudio erzeugt so ein herausragendes Selbstbild, indem es mit seiner Offenheit im starken Kontrast zu den großen Publishern steht, welche sich mit ihren Projekten in Geheimhaltung hüllen. Auch sorgt diese Offenheit dafür, dass es einfacher ist, den Release, wenn nötig zu verschieben, da die Spieler informiert sind, wo die Probleme liegen und weshalb es zu Verzögerungen kommt. Diese Transparenz ist für Indie Studios wie Ninja Theory nur durch ihre Unabhängigkeit von einem Publisher möglich. Letzterer muss als Unternehmen an der Börse mitunter aus strategischen oder rechtlichen Gründen anders mit den Informationen verfahren, da deren Weitergabe an einzelne andere Teilhaber benachteiligen könnte. Des Weiteren verbreiteten die Entwickler zum Release von Hellblade den Anschein, dass, wenn die Spielfigur zu viele digitale Tode stirbt, der gesamte Spielstand mit allen Fortschritten gelöscht wird. Die Andeutung eines Permadeath-Systems ermöglichte eine weiterhin gesteigerte Identifikation der Spieler mit der Protagonistin des Spiels. Diese Angst vor dem Scheitern stieß eine Diskussion in den sozialen Netzwerken an, die dem Studio zusätzliche Publicity einbrachte. Auf vielen Spielewebsites hat Hellblade daher neben dem eigentlichen Test zwei weitere Artikel erhalten; einen zum Permadeath und den zweiten in dem klar gestellt wurde, dass der Permadeath nur ein Bluff ist. Solche ungewöhnlichen Designentscheidungen auf dem Rücken der sozialen Medien auszutragen, kann mitunter zur Popularität des Titels beitragen. Besonders im AAA Indie Modell, in dem die kreative Freiheit sehr hoch ist, besteht nicht nur eine Basis, auf das Projekt aufmerksam zu machen, sondern auch Diskussionen anregen zu können. Aber wie realistisch ist es nun, dass all diese Ansätze zusammenkommen und ein erfolgreiches Produkt entsteht?

## 6.6 Erfolgsaussichten

Da diese Art der Entwicklung noch sehr neu ist, gibt es nur einen Vertreter, der das hier beschriebene AAA Indie Konzept bereits durchgehend umgesetzt hat; Hellblade von Ninja Theory.

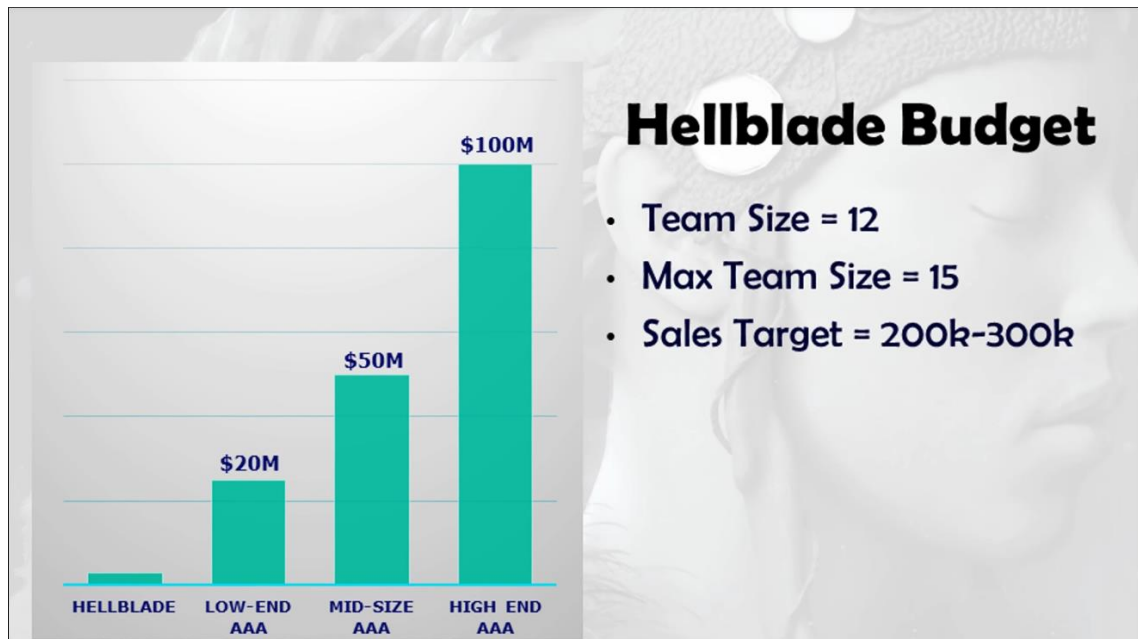


Abbildung 30 Hellblade Budget (Antoniades Tameem, 2014)

Laut der Präsentation von Tameem Antoniades 2014 ist das angestrebte Verkaufsziel, um die Produktionskosten zu decken, 300000 Kopien. Trotz dessen ist Hellblade nur auf zwei Plattformen verfügbar - Windows PC und die Playstation 4. Laut SteamSpy besitzen auf Steam 187006 +- 13094 Spieler Hellblade. (SteamSpy 2017) Eine Ungenauigkeit kommt daher zu Stande, dass die Daten nicht unmittelbar von Steam erhoben wurden. Hellblade hat für die Plattform PC sein Verkaufsziel bereits erreicht. Selbst im schlechtesten Fall hat Hellblade allein auf dem PC mehr als 170000 Kopien verkauft. Der Playstation.Blog berichtet sogar: „*Hellblade: Senua's Sacrifice was the bestselling game on Playstation Store in August*“. (Dutton Fred 2017). Ninja Theory hat mit ihrem letzten Dev Diary auch Zahlen über den Verkauf veröffentlicht. Diese Zahlen müssen normalerweise kritisch betrachtet werden, da ein Entwickler diese womöglich anders auslegen könnten. In diesem Fall gehen sie allerdings mit den anderen Quellen und Meldungen einher.



Abbildung 31 Hellblade Sales(Ninja Theory Ltd. 2017a)

In der Grafik ist zu erkennen, dass der Verkauf von Hellblade schon mit etwa 75000 Vorbestellungen begonnen hat. Auch flacht die Kurve langsamer ab, als erwartet, was auf ein höheres Interesse am Titel und auf eine wirksame Art und Weise der Vermarktung hinweist. Besonders durch die sehr transparente PR über den Entwicklungszeitraum und bevor der Veröffentlichung konnte Hellblade mit sehr vielen Vorbestellungen starten. Über den Kurznachrichtendienst Twitter gab Ninja Theory am 22 November 2017 folgendes bekannt:

*„#Hellblade: Senua's Sacrifice has reached 500k sales in 3 months, moving the game into profit ahead of our expectations!“*. (Ninja Theory 2017b)

Ein weiterer Zuspruch zum erfolgreichen Verkauf des Produkts kann dadurch getätigt werden, da Ninja Theory einen Teil ihrer Einnahmen sogar spenden konnte:

*„In Support of World Mental Health Day, which is observed each year on October 10, Ninja Theory is donating all proceeds from today's sales of its most recent release, Hellblade: Senua's Sacrifice, to charity.“* (Sirani Jordan 2017)

Dieses Statement und die zuvor dargelegten Zahlen beweisen, dass es möglich ist, ein Projekt im Raum zwischen AAA Produktionen und Indie Titeln zu platzieren und finanziell erfolgreich zu sein, wenn die zuvor aufgeführten Kriterien eingehalten wurden.

## 6.7 Chancen für das Konzept

Die zuvor analysierten Zahlen beweisen, dass auch kleine Teams, wenn sie die spezifischen Arbeitsweisen verfolgen und die ihnen zur Verfügung stehende Technik effizient nutzen, Spiele mit AAA Qualitäten umsetzen zu können. Anwenden lässt sich dieses Konzept jedoch nur auf gut skalierbare Titel wie Hellblade, da Spiele mit einer aufwendigen Online-Architektur dem Studio nur eine aufwendige Langzeitbetreuung aufbürden würden. Das Entwicklerteam muss je nach Plattform und Art des Spiels ebenfalls mit Spielern rechnen, die sich einen unfairen Vorteil durch diese Lücken verschaffen. Da diese Individuen, nachdem eine Sicherheitslücke geschlossen wurde, nach der nächsten lukrativen suchen, muss der Entwickler weiter an den Systemen und aufkommenden Problemen arbeiten. Selbst wenn zudem keine dedizierten Server bereitgestellt werden müssten, muss das Entwicklerteam weiter am Balancing und dem Reparieren von Bugs arbeiten. Diese Bugs werden zwar auch bei klassischen Singleplayer Spielen nachträglich per Patch behoben, bereiten allerdings keine Probleme der Art von Online-Spielen. Die Arbeitsweisen können auch genutzt werden um Projekte die nicht in das Konzept passen zu optimieren. Jedoch beschränkt sich der größte Nutzen des Konzepts auf Singleplayer Erfahrungen. Die mit diesem Modell erstellten Titel können sich jedoch nur schwer mit dem Umfang großer AAA Produktionen messen, da es selbst durch etwa prozedurale Generierung nur sehr schwer möglich ist, massive Mengen an Inhalten zu generieren. Vor allem bedarf es größerer Kapazitäten, diese konstant auf AAA Standard aufrechtzuhalten. Letztendlich bedarf es immer noch der Arbeit von menschlichen Entwicklern, um dieses Ziel zu erreichen.

## 7 Fazit

### 7.1 Zusammenfassung

Die Analyse klärt die Fragen nach den kritischen Erfolgsfaktoren, um ein digitales Spiel im AAA Standard mit wenigen Entwicklern umzusetzen. Durch die Analyse der Branche sowie Techniken und Möglichkeiten haben sich folgende Faktoren als kritisch erwiesen:

- Es werden erfahrene Entwickler benötigt, um das Projekt effektiv zu planen.
- Für den Verkauf muss das Buy to Play Modell genutzt werden.
- Eine breit aufgefächerte Finanzierung muss getätigt werden.
- Die wichtigsten Plattformen PC, PS4 und Xbox one müssen abgedeckt sein.
- Es muss eine vorgefertigte Engine genutzt werden, welche:
  - die angestrebten Plattformen unterstützt.
  - sich flexibel anpassen lässt.
  - zu Beginn mit geringen finanziellen Mitteln nutzbar ist.
- Es müssen Techniken zur Minimierung des Arbeitsaufwandes betrieben werden.
- Das Team muss, wenn prozedurale Verfahren verwendet werden, aus mehreren Programmierern bestehen.
- In den sozialen Medien muss genügend Reichweite generiert werden, um hervorzustechen.
- Es können nur Einzelspieler-Titel umgesetzt werden.

Wenn das Projekt diese Kriterien umsetzt, können kleine Teams Spiele im AAA Standard umsetzen.

## 7.2 Ausblick

Von den Daten aus Kapitel 2 lässt sich ableiten, dass, nachdem viele große Publisher ihre Projekte auf Games as a Service Projekte umstellten, diese Entwicklung weiter andauern wird. Das Aufkommen von immer schnelleren und flächendeckenderen Internetverbindungen begünstigt die Entwicklung zusehends. Das klassische Kaufen von Videospiele auf physischen Datenträgern wird sich noch einige Jahre halten, da es noch einen großen Teil Spieler gibt, die über keine Breitbandinternetverbindung verfügen. Jedoch kann davon ausgegangen werden, dass, da PC Spiele die nicht den Sprung auf Blue Rays geschafft haben, diese noch vor den Konsolenspielen vollständig auf physische Datenträger verzichten werden. Es ist nicht zu erwarten, dass die Industrie nachdem die Blue Ray schon seit 2006 mit der Einführung der PS3 in großen Massen verfügbar war, PC Spiele noch auf Blue Rays veröffentlichen wird. Für Indie-Entwickler wird es in Zukunft wieder schwieriger, da große Publisher ebenfalls diese Lücke im Markt erkennen. Mit den EA Originals führt EA ein Programm ein, um Indie Entwickler zu unterstützen. Dadurch droht den Indie Studios jedoch gleichzeitig Gefahr, da das Programm vielmehr dazu dient, die Stellung EA's in der Indie Szene stärken, sodass auch aus diesem Marktsegment Gewinne erzielt werden können. Durch die Etablierung der Independent Studios werden die Publisher unter Druck gesetzt; so stellt EA beispielsweise alle seine Titel auf die Frostbite Engine um, damit alle EA Studios Assets und Arbeit teilen können. Durch neue Technologien jedoch, wie etwa von Ninja Theory vorangetrieben, erhalten mehr und mehr Entwickler die Möglichkeit, hochqualitative Spiele umzusetzen und somit auch ein kleines Studio in der Lage ist, Spiele in AAA Qualität abzuliefern. Vor allem ausgehend von den Verkaufszahlen von Hellblade ist klar, dass die Nachfrage nach dieser Art von Spielen enorm ist. Mit diesem Wissen und weiteren Projekten wie Hellblade könnte sich die Branche in den nächsten Jahren stark wandeln. Die Entscheidung darüber wird der Kunde und Spieler mit seiner Kaufentscheidung fällen.

Für eine weiterführende Arbeit könnte, da dieses Modell nur auf Einzelspieler-Spiele ausgelegt ist, untersucht werden, welche Faktoren entscheidend wären, um Multiplayer-Spiele mit kleinen Teams und beschränkten Budget umzusetzen. Dahingehend könnte ebenfalls analysiert werden, welche Indie-Titel anhand ihres Genres, Stils und Gameplay erfolgreich waren. Aus diesen Ergebnissen könnten die kritischen Faktoren für die Entwicklung eines Multiplayer-Spiels auf Basis der Eigenschaften eines erfolgreichen Indie Titels erstellt werden.

---

## Literaturverzeichnis

### Bücher:

Anderie Lutz (2016): Games Industry Management, Springer-Verlag Berlin Heidelberg 2016

Short, Tanya X und Tarn Adams (2017): Procedural Generation in Game Design. Taylor & Francis Group Boca Raton(2017)

### Hochschulschriften:

Müller Franz (2017): Bericht zum Praxis-/Forschungsmodul. Hochschule Mittweida. Praktikumsbericht

### Internet:

Activision Blizzard. n.d. (2015) Anzahl der Abonnenten von World of Warcraft weltweit vom 1. Quartal 2005 bis zum 3. Quartal 2015 (in Millionen). Statista. Zugriff am 18. September 2017. Verfügbar unter <https://de.statista.com/statistik/daten/studie/208146/umfrage/anzahl-der-abonnenten-von-world-of-warcraft/>.

Activision Blizzard. n.d. (2016) Umsatz von Activision Blizzard\* weltweit in den Jahren 2005 bis 2016 (in Millionen US-Dollar). Statista. Zugriff am 14. August 2017. Verfügbar unter <https://de.statista.com/statistik/daten/studie/163582/umfrage/umsatz-von-activision-blizzard-seit-2005/>.

Antoniades Tameem (2014) The Independent AAA Proposition. Zugriff am 13. November 2017. Verfügbar unter: <http://www.hellblade.com/the-independent-aaa-proposition/>

Antoniades Tameem (2015) Hellblade Development Diary 10: Capturing Performance. Zugriff am 14. November 2017. Verfügbar unter: <https://youtu.be/YBprDIzcXw>

BIU. n.d. (2016) Umsatz mit kostenpflichtigen virtuellen Zusatzinhalten für Computer- und Videospiele (Mikrotransaktionen) in Deutschland von 2008 bis 2016 (in Millionen Euro). Statista. Zugriff am 7. August 2017. Verfügbar unter <https://de.statista.com/statistik/daten/studie/200040/umfrage/umsatz-mit-virtuellen-zusatzinhalten-fuer-video-spiele-in-deutschland-seit-2008/>.

Börse Frankfurt Gamesstop A Aktie(2017) Zugriff am 22. September 2017. Verfügbar unter <http://www.finanzen.net/aktien/Gamestop-Aktie>

CrowdExpert. n.d. (2015) *Weltweit durch Crowdfunding eingesammeltes Kapital von 2011 bis 2015 (in Milliarden US-Dollar)*. Statista. Zugriff am 13. November 2017. Verfügbar unter <https://de.statista.com/statistik/daten/studie/258199/umfrage/weltweit-durch-crowdfunding-eingesammeltes-kapital/>.

Dutton Fred Playstation.Blog. (2017) Hellblade: Senua's Sacrifice was the bestselling game on PlayStation Store in August. Zugriff am 19. November 2017. Verfügbar unter: <https://blog.eu.playstation.com/2017/09/08/hellblade-senuas-sacrifice-was-the-best-selling-game-on-playstation-store-in-august/>

Für-Gründer.de. n.d. (2017) *Crowd-Finanzierungsvolumen in Deutschland im Jahr 2016 nach Segmenten (in Millionen Euro)*. Statista. Zugriff am 6. Oktober 2017. Verfügbar unter <https://de.statista.com/statistik/daten/studie/631225/umfrage/crowdfinanzierungsvolumen-in-deutschland-nach-segmenten/>.

GameStop. n.d. (2016) Umsatz des Computerspielehändlers Gamestop in den Geschäftsjahren 2007 bis 2016 (in Millionen US-Dollar). Statista. Zugriff am 22. September 2017. Verfügbar unter <https://de.statista.com/statistik/daten/studie/253229/umfrage/umsatz-von-gamestop/>.

GDC(2017) Building Worlds in No Man's Sky Using Math(s). Zugriff am 22. September 2017. Verfügbar unter: <https://www.youtube.com/watch?v=C9RyEiEzMiU>

Horizont. n.d.(2014) Anteil der Marktsegmente an den weltweiten Umsätzen im Games-Markt im Jahr 2013 und Prognose bis 2017. Statista. Zugriff am 7. August 2017. Verfügbar unter <https://de.statista.com/statistik/daten/studie/317018/umfrage/prognostizierte-entwicklung-der-marktsegmenten-im-games-markt/>.

Matthews Dominic (2014) Hellblade Development Diary 5: Business of Creation. Zugriff am 14. November 2017. Verfügbar unter: [https://youtu.be/\\_f3NMMrQFEE](https://youtu.be/_f3NMMrQFEE)

Murray Sean Playstation.Blog. (2014) Erforscht die 18.446.744.073.709.551.616 Planeten von No Man's Sky. Zugriff am 20. Oktober 2017 Verfügbar unter: <https://blog.de.playstation.com/2014/08/26/erforscht-die-18-446-744-073-709-551-616-planeten-von-no-mans-sky/>

Ninja Theory Ltd. (2015a) Hellblade Development Diary 10: Capturing Performance. Zugriff am 14. November 2017. Verfügbar unter: <https://youtu.be/YBprDIzcXxw>



Ninja Theory Ltd. (2015b) Hellblade Development Diary 18: The Shoot Set Up. Zugriff am 14. November 2017. Verfügbar unter: <https://www.youtube.com/watch?v=FACTByOjgyQ>

Ninja Theory Ltd. (2017a) Hellblade Development Diary: The Launch. Zugriff am 27. November 2017. Verfügbar unter: <https://www.youtube.com/watch?v=sDvATt2VBxU>

Ninja Theory Ltd.(2017b) Ninja Theory Twitter Handle. Zugriff am 27. November 2017. Verfügbar unter: <https://twitter.com/NinjaTheory/status/933305851212943360>

PwC. n.d. Prognose der Umsätze im Bereich Social/Casual Games in Deutschland von 2008 bis 2020 (in Millionen Euro). Statista. Zugriff am 22. September 2017. Verfügbar unter <https://de.statista.com/statistik/daten/studie/239104/umfrage/umsaetze-im-markt-fuer-mobile-games-in-deutschland/>.

Roberts Chris(2016) Full CitizenCon 2016 Presentation Zugriff am 1.November 2017 Verfügbar unter: <https://youtu.be/XuDj5v81Nd0?t=1h22m22s>

Roberts Space Industries (2017) funding Goals. Zugriff am 31.Oktober 2017 Verfügbar unter: <https://robertsspaceindustries.com/funding-goals>

Sirani Jordan (2017) IGN All Hellblade Profits go toward charity Today for World Mental Health Day Sacrifice Zugriff am 19. November 2017. Verfügbar unter: <http://www.ign.com/articles/2017/10/10/all-hellblade-profits-go-toward-charity-today-for-world-mental-health-day>

Star Citizen (2017) CitizenCon 2947 – Complete Broadcast Zugriff am 1.November 2017 Verfügbar unter: <https://www.youtube.com/watch?v=eu71Dcpf3c0>

Statista. n.d. (2017) Prognose der weltweiten Umsätze im Markt für Digitale Games in den Jahren 2015 bis 2021 (in Millionen Euro). Statista. Zugriff am 7. August 2017. Verfügbar unter <https://de.statista.com/statistik/daten/studie/457425/umfrage/weltweite-umsaetze-im-markt-fuer-digitale-games/>.

Statista. n.d. (2017)*Weltweite Umsätze im Markt für Digitale Games im Jahr 2016 sowie eine Prognose bis 2022 (in Millionen Euro)*. Statista. Zugriff am 13. November 2017. Verfügbar unter <https://de.statista.com/statistik/daten/studie/457425/umfrage/weltweite-umsaetze-im-markt-fuer-digitale-games/>.

SteamSpy (2017) Hellblade: Senua's Sacrifice Zugriff am 19. November 2017. Verfügbar unter: <http://steamspy.com/app/414340>

---

Social Media Examiner. n.d. *Welche Social Media Plattform ist für Ihr Unternehmen am wichtigsten?*. Statista. Zugriff am 13. November 2017. Verfügbar unter <https://de.statista.com/statistik/daten/studie/463928/umfrage/wichtigste-social-media-plattformen-fuer-marketingverantwortliche/>.

VGChartz. n.d.(2016) Einzelhandelsabsatz der meistverkauften Computer- und Videospiele weltweit im Jahr 2015 (in Millionen Stück). Statista. Zugriff am 14. August 2017. Verfügbar unter <https://de.statista.com/statistik/daten/studie/517877/umfrage/absatz-der-meistverkauften-konsolenspiele/>.

3D Gamedevblog (2016). No Man's Sky – Procedural Content. Zugriff am 20.Oktober 2017 Verfügbar unter: <http://3dgamedevblog.com/wordpress/?p=836>

## Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe. Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

---

Ort, Datum

Vorname Nachname