

Toon-Shading in Game und Animation – Unterscheidung und Definition der Filmästhetik in der stilisierten Charakterdarstellung

Susanne Teufelauer



MASTERARBEIT

eingereicht am
Fachhochschul-Masterstudiengang

Digital Arts

in Hagenberg

im November 2019

© Copyright 2019 Susanne Teufelauer

Alle Rechte vorbehalten

Erklärung

Ich erkläre eidesstattlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen entnommenen Stellen als solche gekennzeichnet habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Hagenberg, am 26. November 2019

Susanne Teufelauer

Inhaltsverzeichnis

Erklärung	iii
Vorwort	vi
Kurzfassung	vii
Abstract	viii
1 Einleitung	1
1.1 Zielsetzung	1
1.2 Aufbau der Arbeit	2
2 Ursprung und Entwicklung der Toonästhetik	3
2.1 Ursprung	3
2.2 Entwicklung im 3D Bereich	6
2.2.1 Animation	7
2.2.2 Games	8
2.3 Zusammenfassung	8
3 Toon-Shading in der Animation	10
3.1 <i>Love, Death & Robots</i>	10
3.1.1 Episode 4: <i>Suits</i>	10
3.1.2 Episode 15: <i>Blind Spot</i>	11
3.2 <i>Sonder</i>	12
3.3 <i>Spider-Man: Into the Spider-Verse</i>	13
3.4 <i>Feast</i>	17
3.5 Zusammenfassung	18
4 Toon-Shading in Games	19
4.1 <i>The Legend of Zelda: The Wind Waker</i>	19
4.2 <i>The Legend of Zelda: Breath of the Wild</i>	20
4.3 <i>The Wolf Among Us</i>	22
4.4 <i>Valkyria Chronicles 4</i>	23
4.5 <i>Ni no Kuni II: Revenant Kingdom</i>	25
4.6 Zusammenfassung	26

5	Gemeinsamkeiten und Unterschiede	28
5.1	Outlines	28
5.2	Detaillinien	29
5.3	Flache Farben	30
5.4	Harter Schatten	31
5.5	Weicher Schatten	32
5.6	Colorbleeding, Hatching und Lichttexturen	32
5.7	Mehrere Lichtquellen	33
5.8	Rimlight	34
5.9	Depth of Field	34
5.10	Zusammenfassung	35
6	Realisiertes Toon-Shading	36
6.1	Anforderungen	36
6.2	Umsetzung	37
6.2.1	Unity	38
6.2.2	Blender	40
6.3	Vergleich	43
6.4	Zusammenfassung	44
7	Zusammenfassung	46
A	Inhalt der CD-ROM/DVD	49
	Quellenverzeichnis	50
	Literatur	50
	Audiovisuelle Medien	51
	Software	51
	Online-Quellen	52

Vorwort

Zu aller erst möchte ich meinem Betreuer für das regelmäßige Feedback und die Betreuung dieser Masterarbeit danken. Das bereitgestellte Fachwissen und die strukturellen Vorschläge waren maßgeblich für den Verlauf und Abschluss dieser Arbeit verantwortlich.

Dank gebührt auch meinen Eltern, welche mir dieses Studium erst ermöglicht und an den Entscheidungen, die ich getroffen habe, nie gezweifelt haben. Als erste Akademikerin in der Familie macht mich das besonders stolz, da dies definitiv keine Selbstverständlichkeit ist und ich durch euch meine Wünsche und Träume erfüllen konnte.

Schlussendlich möchte ich noch meinen Mitbewohnern, Studienkollegen und Freund danken, da sie durch moralische Unterstützung und Bereitstellen von Rat und Tat meinen Seelenfrieden während des Studiums gewahrt haben.

Danke!

Kurzfassung

Das Ziel der vorliegenden Masterarbeit ist es, die Charakteristiken eines Toon-Shadings heraus zu arbeiten und einen Vergleich in den unterschiedlichen Anwendungen in Animationfilm und Game aufzustellen. Dazu wurden jeweils relevante Beispiele in den Bereichen Games und Animation gesucht und analysiert. Diese Analyse verdeutlicht zu aller erst, welche Merkmale einem Toon-Shading zuzuordnen sind und liefert zugleich eine Basis für die darauf folgende Gegenüberstellung. Dieser Vergleich stellt die Unterschiede und Gemeinsamkeiten der verschiedenen Toon-shadings in den zwei Medien dar. Es stellen sich einige Merkmale relevanter als andere heraus. So werden die Outlines eher im Bereich der Games und weiche Schatten oder Rimlights vermehrt in den Animationen verwendet. Auch gibt es gemeinsame Nenner, wie das Cel-Shading, also der harte Schatten, welcher in beiden fast gleichermaßen verwendet wird. Durch diesen Vergleich konnten Anforderungen an ein realisiertes Toon-Shading gestellt werden. Anhand dieser wurde er in Unity und auch Blender entwickelt. Es wurde bestätigt, dass ein filmisches Toon-Shading sehr wohl in einer Game-Engine entwickelt werden kann, jedoch Abzüge in der Qualität des Lichts und der Kantenglättung beinhaltet.

Abstract

The aim of this master thesis is to work out the characteristics of a toon shading and to compare the differences in animation movies and games. For this purpose, relevant examples in the areas of games and animation movies were sought and analysed. This analysis clarifies which features can be assigned to a toon shading and provides a basis for a comparison, which shows the differences and similarities of the different toon shadings in these two areas. Some features turn out to be more relevant than others. Outlines are more often used in the area of games, while soft shadows or rimlights are more often to be found in animation movies. There are also common denominators, such as cel shading which is used almost equally in both. Through this comparison the requirements for the produced toon shader could be determined. On this basis the shading was developed in Unity and Blender. It turns out that a cinematic toon shading can indeed be developed in a game engine, but contains some disbenefits in the quality of light and anti-aliasing.

Kapitel 1

Einleitung

Toon-Shadings und Non Photorealistic Renderings (NPR)¹ gehören schon seit der Entwicklung von computergenerierten Bildern zu einer beliebten Art der Darstellung. Sie können nicht nur sehr performant berechnet werden, sondern geben der Darstellung auch einen sehr individuellen und einzigartigen Stil. Doch im Zeitalter der möglichst realistischen Bildgenerierung wird gezielt das Toon-Shading oft als kindisch oder typischer Game-Stil abgestempelt. Welche Eigenschaften zeichnen diesen Stil aus und wie schaffen es aktuelle Filme und Serien ein Toon-Shading trotzdem als seriös und aufwändig darstellen zu können? Diese Masterarbeit beschäftigt sich genau mit dieser Frage und zeigt aktuelle und historisch relevante Beispiele zu deren Analyse auf. Zudem werden die gewonnenen Erkenntnisse an einem eigenen erstellten Shader angewandt, der die filmische Darstellung eines Charakters und seiner Umgebung in der Game-Engine Unity 2019.2.1 ermöglicht. In dieser Arbeit wird auf Grund einer besseren Lesbarkeit auf eine geschlechtsspezifische Formulierung verzichtet. Die gewählte männliche Form bezieht sich immer zugleich auf weibliche und männliche Personen, soweit es für die Aussage erforderlich ist.

1.1 Zielsetzung

Ziel dieser Arbeit ist es, die Eigenschaften eines Toon-Shadings zu definieren und die Unterschiede angewandter Beispiele in Animationen und Games näher zu beleuchten. Auf diesen aufbauend, wird ein Toon-Shading in Unity und Blender realisiert und verglichen. Dieses Toon-Shading soll sich stilistisch von den Merkmalen der Toon-Shadings in Games unterscheiden und somit die Eigenschaften der Toon-Shadings der Animationen beinhalten. Es soll somit aufgezeigt werden, dass eine filmische Darstellung in einer Game-Engine möglich ist und sich diese in der gezielten Gegenüberstellung zu einem in Blender gerenderten Toon-Shadings nur minimal unterscheidet. Dazu werden verschiedene Beispiele analysiert und verglichen. Diese Analyse soll die Gemeinsamkeiten und Unterschiede hervorheben und somit die Grundlage für die Findung der Anforderung liefern. Dafür werden gezielt dominante Merkmale eines Toon-Shadings betrachtet und auch die Art der Verwendung in den unterschiedlichen Beispielen hervorgehoben. So ist

¹NPR: Non-photorealistic Rendering bezeichnet eine nicht fotorealistische, sondern stilisierte Darstellung und Berechnung von Computergrafiken.

es beispielsweise notwendig, nicht nur die Verwendung eines Rimlights zu betrachten, sondern auch das Zusammenspiel mit weiterer Beleuchtung oder Einstellungsmöglichkeiten welche Strichstärke, Farbe und dergleichen betreffen. Somit entsteht ein tiefgreifender Vergleich, welcher nicht nur unterschiedliche Anwendungsfälle beleuchtet, sondern auch gezielte stilistische Entscheidungen miteinbezieht.

Die daraus resultierende Forschungsfrage lautet: Wie unterscheidet sich Toon-Shading in der Animations und Games Branche und welche Eigenschaften muss ein Toon-Shading in einer Game-Engine aufweisen, um eine filmische Ästhetik erreichen zu können?

1.2 Aufbau der Arbeit

Um aufzuzeigen, was ein Toon-Shading eigentlich ist und aus welcher Bedeutung heraus er sich entwickelt hat, wird in Kapitel 2 zuerst auf die geschichtliche Evolution der Toonästhetik eingegangen. Hier wird nicht nur der eingegrenzte Bereich der Gamesentwicklung und Animationsproduktion betrachtet, sondern auch dessen Ursprung. Zusätzlich werden auch die verschiedenen Arten und Anwendungsmöglichkeiten von NPR Stilen aufgezeigt und welche Wichtigkeit diese in der Entwicklung von Toon-Shadings aufweisen, beleuchtet.

In Kapitel 3 und Kapitel 4 werden die verschiedensten und auffallendsten Merkmale eines Toon-Shadings an gezielten Beispielen aufgezeigt und beschrieben. Die analysierten Beispiele wurden aus unterschiedlichsten Genres ausgewählt. Es wurde jedoch darauf geachtet, dass der Großteil dieser Projekt in den letzten fünf Jahren produziert und veröffentlicht wurden, damit eine aktuelle Gegenüberstellung dieser möglich ist und somit keine Verfälschungen basierend auf früheren technischen Einschränkungen vorhanden ist. Es wird, zusätzlich zu den sichtbaren Eigenschaften, auch auf technische Besonderheiten und individuelle Produktionsmerkmale eingegangen.

In Kapitel 5 werden diese Beispiele und deren Merkmale miteinander verglichen. Die Unterschiede und Gemeinsamkeiten werden einzeln hervorgehoben und genauer beleuchtet. Dies ermöglicht es zu beurteilen, welche Merkmale einerseits für ein Toon-Shading in der Animation unabdingbar sind und andererseits die Ästhetik eines Games unterstreichen würden. Die herausgearbeiteten Eigenschaften dieses Kapitels liefern die Basis für die Anforderungen des entwickelten Toon-Shadings.

Diese Anforderungen bilden die Grundlage für Kapitel 6. Hier werden die erarbeiteten Merkmale und deren Anwendung genau eingegrenzt und beschrieben. In Folge darauf ist die realisierte Umsetzung des Shaders zu sehen. Zuerst wird dieser in Unity in dem programmeigenen Shadergraph erstellt. Der Shadergraph ermöglicht es in einem nodebasiertem System einen Shader zu erstellen und zeigt eine systematisch und nachvollziehbare Darstellungsart auf. Es wird einzeln auf die Merkmale, deren Umsetzung und eventuelle technische Schwierigkeiten eingegangen. Zusätzlich wird dieser Shader so exakt wie möglich in Blender erstellt. Blender weist einen ähnlichen Shader Editor auf und lässt somit eine ebenso bildliche Darstellung des Aufbaus zu. Hier wird ebenfalls auf die Eigenschaften, Realisierung und auftretenden Problematiken eingegangen. Die zwei ähnlichen, aber dennoch unterschiedlichen Shader, werden anschließend miteinander verglichen. Somit können die Unterschiede, welche rein durch das verwendete System und der Art des Renderns entstehen, aufgezeigt und verglichen werden.

Kapitel 2

Ursprung und Entwicklung der Toonästhetik

Toonshading ist eine weit verbreitete Ausprägung des NPR. Er findet seine Anwendung in der Animation und auch Spieleentwicklung. Dieses Kapitel gibt einen Überblick über die Entstehung und die Herkunft des Toon-Shadings, zeigt die Entwicklung im analogen wie auch im digitalen Bereich und bereitet eine Analyse dessen Eigenschaften auf.

2.1 Ursprung

Das erste kommerzielle Aufkommen des Toon-Shadings war mit der Anwendung im Computerspiel *Jet Set Radio* [36] im Jahr 2000 [10]. Doch die Verwendung der Merkmale dieses Stils können schon viel früher wahrgenommen werden. Gezielt werden hier dominante Eigenschaften wie etwa das Cel-Shading oder Character Outlines hervorgehoben. Diese Eigenschaften finden ihre Verwendung in vielen Comicbüchern. Der erste offizielle Comic wurde 1897 unter dem Namen *The Yellow Kid in McFadden's Flats* [13] veröffentlicht und gilt nur deshalb als erster Comic, weil diese Bezeichnung auch auf dem Rückumschlag des Buches stand. Es beinhaltet schwarz und weiß Nachdrucke von populären Zeitungs-Comicstrips, welche schon Jahre zuvor veröffentlicht wurden. Das erste originale Comicbuch *New Fun #1* [15] wurde 1935 von National Allied Publications gedruckt [7].

Genauer betrachtet, können comictypische Merkmale geschichtlich jedoch schon viel früher wahrgenommen werden. So wurden etwa bereits vor über 20.000 Jahren Bilder in den Höhlen von Lascaux [63] gemalt, welche dominante Außenlinien, einfarbige Füllungen und teilweise flache Schattierungen enthalten [4]. Dies bezieht sich auf die damaligen Möglichkeiten der Kunst, aber dennoch zeigt es auf, dass auch mit einfachen Wegen und Mitteln sehr ausdrucksstarke Bilder kreiert werden können. Und aus dieser Einfachheit heraus, trotz der begrenzten Strukturen, Farbinformationen und Unschärfen noch genügend Informationen für Tiefe, Perspektive und Details bereitzustellen, entstand die Kunst des Comics, aus etwas Komplexem die Simplizität herauszufiltern [11].

Ein Comic zeichnet sich jedoch nicht nur durch seine Art der Darstellung von Bildern aus. Auch die Erzählung einer Geschichte mit Hilfe von sogenannten Panels, Sprechblasen, unterstützenden Wörtern, welche Ton und Geräusche simulieren, aber auch Linien

(a) *The Amazing Spiderman* [3] (1963).(b) *Saga* [14] (2012).

Abbildung 2.1: Sichtliche Unterschiede zwischen einem typischen Comic und einem aktuellen Graphic Novel.

zur Hilfe der Orientierung von Bewegung, tragen wesentlich zur Eingrenzung des Begriffs Comics bei. Dabei zählt nicht nur die Gestaltung der Einzelbilder, sondern auch dessen Anordnung, Platzierung und dadurch erzeugte visuelle Narrative der Erzählform. Innerhalb dieser Panels wird eben diese Strukturierung der Geschichte ebenso aufgegriffen und mit Hilfe von den vorher genannten unterstützenden Elementen weiter gesponnen und eingebunden. Dadurch ergibt sich über den ganzen Comic eine Kohärenz, welche die Geschichte und Erzählung, die sich darin befindet, umschließt und eine eigene Realität aufzeigt. Diese Kohärenz spiegelt sich auch in der Gestaltung einzelner Elemente dar, wie zum Beispiel den Charakteren. So unterschiedlich die Realitäten der verschiedenen Comics und deren Geschichten sind, so sind es auch die Darstellungsarten der Figuren und deren Umgebung [2].

In Abbildung 2.1 wird ersichtlich, wie diese Unterschiede aussehen können. Abbildung 2.1a zeigt den ersten Spiderman-Comic *The Amazing Spiderman* [3] aus dem Jahre 1963 von Joe Benett gezeichnet. Charakteristisch sind hier die grellen Farben, tiefschwarzen Schatten, skizzierte Halbschatten und ein Auslaufen der Farbe über die Linien, welche durch einen mangelhaften Druck ausgelöst werden. Es ist ein sehr schlichter Stil und lässt auch vermuten, dass der Comic selbst auf sehr dünnem Papier gedruckt und sehr günstig produziert wurde.

In Abbildung 2.1b ist ein sogenannter Graphic Novel namens *Saga* [14] aus dem Jahr 2012 von Fiona Staples gezeichnet, zu sehen. Ein Graphic Novel bezeichnet ein Schriftstück, welches zwar Comicelemente enthält, aber als Buch veröffentlicht wird. Aufgezeigt wird dies speziell durch ein oft sehr aufwändig gezeichnetes Hardcover und sehr hochwertige Seiten, welche durch einen ebenfalls qualitativ sehenswerten Druck beeindrucken. Durch diese Produktion stehen den Künstlern ein breiteres Farbspektrum und eine größere Druckgenauigkeit zur Verfügung. Auffallend ist hierbei, dass es nicht mehr nur rein schwarze Schatten gibt, sondern auch Abstufungen in diesen und dass

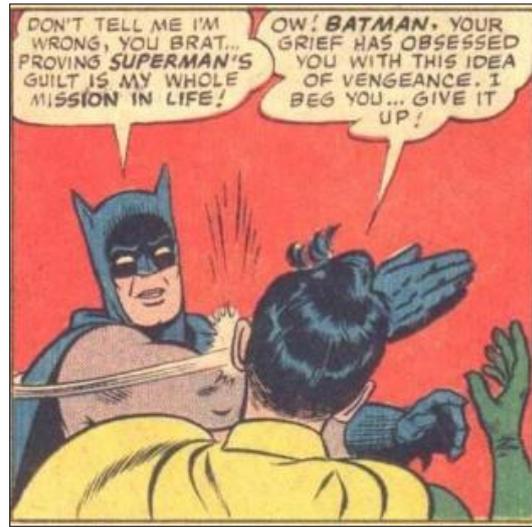
(a) Ben Day Dots in *Tarzan* [1] (1930–1933).(b) Hatching in *World's Finest* #153 [6] (1965).

Abbildung 2.2: Notwendige Herangehensweisen und Problemlösungen durch bedingte Limitierungen in der Produktion.

viele Hintergründe und Effekte mit Hilfe von Farbverläufe visualisiert werden [7].

Einige dieser Unterschiede in Abbildung 2.1 sind somit simpel der Qualität des Druckes und der Produktion im Allgemeinen verschuldet. Hervorzuheben ist hierbei die Art des Farbdruckes bei den auch sogenannten *Oldschool Comics*. Hauptproblematik ist eine Technik namens *four coloured dotprint*, oder auch als *Ben-Day Dots* bekannt. In den frühen Anfängen des Comics gab es noch keine Druckstandards oder Regelungen was Farbe, Farbstärke und Ähnliches betrifft. Somit mussten die Künstler selber versuchen, möglichen Druckfehlern bereits beim Erstellen des Comics entgegenzuwirken. Es standen vier Farben zur Verfügung: Cyan, Magenta, Gelb und Schwarz, auch als CMYK Druckfarben bekannt. Diese Farben können auch als Punkte überlappend gedruckt werden und somit ein größeres Farbspektrum für den Leser ermöglichen [66].

In Abbildung 2.2a ist die Anwendung der Ben-Day Dots zu sehen. Auch das gefürchtete *Colorbleeding*, bei dem die Farbe über die eigentliche Platzierung innerhalb der Outlines gedruckt wird, ist in diesem Bild ersichtlich [55]. Abbildung 2.2b zeigt die Herangehensweise für die Schattierungen. Es werden dünne, parallel zueinander stehende, schwarze Linien gesetzt, welche, je dunkler der Schatten werden soll, immer enger gezeichnet werden. Dies wird *Hatching* genannt. An Batmans linker Hand wird zudem noch eine Abwandlung, das *Cross-Hatching*, sichtbar. Hierbei werden die Linien nicht nur parallel, sondern auch normal zueinander gezeichnet. Die Kombination aus normalem Hatching und Cross-Hatching soll eine gewisse Tiefe vermitteln und wird somit oft für Details, wie etwa Muskeldefinitionen, oder harte Kontraste verwendet. Diese Limitierungen, denn es sind nichts anderes als produktionsbedingte Einschränkungen, haben somit den Stil des Oldschool Comics geprägt und definiert [49].

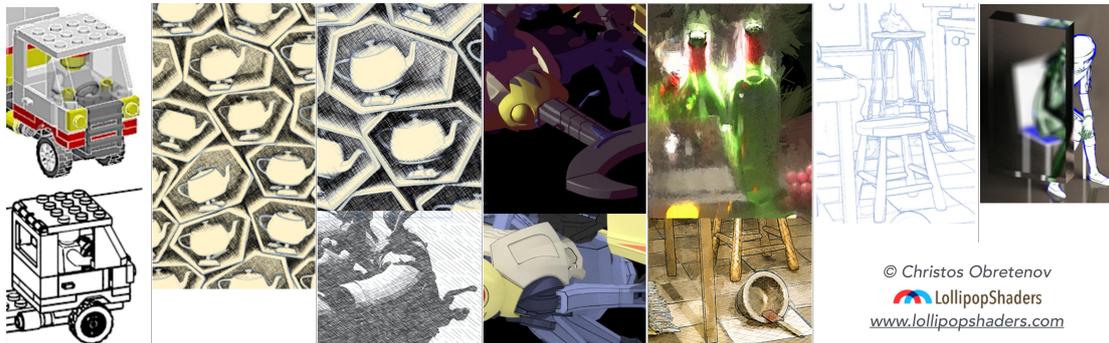


Abbildung 2.3: Verschiedene Non-Photorealistic Rendering Stile erstellt von Christos Obretenov von *LollipopShaders* [60].

2.2 Entwicklung im 3D Bereich

Nach der Digitalisierung des Comicbuches blieb auch der Übergang in den 3D Bereich nicht aus. Toon-Shading und Cartoons im Allgemeinen dominieren die Kinderfilmindustrie und auch in der Spieleentwicklung ist der Comiclook nicht mehr wegzudenken. In der Computergrafik wird diesbezüglich von NPRs gesprochen. Der Begriff NPR umfasst aber nicht nur das Toon-Shading per se, sondern beinhaltet alle Shader und Stile, welche nicht versuchen ein realistisches Abbild der realen Gegebenheiten zu liefern [58].

In Abbildung 2.3 sind sieben verschiedene Arten von NPRs ersichtlich, welche in Verbindung mit dem Begriff Toon-Shading auftreten können. Der erste Shader zeigt pure Outlines und sonst keine Information der 3D Modelle. Das Ergebnis wirkt simpel, aber trotzdem klar lesbar. Der zweite Shader verwendet Hatching als Schattenstilisierung und wendet, bis auf das Darstellen von diesem Schatten, keine weiteren Techniken an. Es ist auffallend, dass nur durch diese Struktur und Textur im Schatten bereits ein starkes malerisches Bild entsteht. Das dritte Rendering kombiniert eben diese Schatten mit den gleichen Outlines von zuvor. Zwar werden diese nicht so stark angewandt, aber das Bild wirkt dennoch leichter lesbar und viel strukturierter. Der vierte Shader wendet auch Farbe auf das Modell an und ist nicht mehr nur auf die Lichtgebung und Kanten angewiesen. Der Schatten hierbei weist ebenso eine leichte Stilisierung auf, da keine realistische Dämpfung vorhanden ist, sondern eine stufenweise Graduierung dieser. Das fünfte Rendering zeigt, wie auch mit Farbe und Texturierung stilisiert umgegangen wird. Es wurde ein Brush, eine Art Pinselstempel, auf die Farbe angewandt und das ganze auch versetzt von der aktuellen Position der Farbe des Modelles weitergeführt. Es entsteht sofort ein sehr malerisches Bild, welchem wieder mit Hilfe von Outlines eine klarere Lesbarkeit mitgegeben wird. Das Verschieben von Bildinformationen wird auch bei dem nächsten Shader angewandt. Hier sind wieder nur Kanteninformationen dargestellt, welche jedoch leicht abseits von deren wirklichen Position erscheinen. In der Animation sind diese durch ihre wackelnde Bewegungen besonders auffallend. Der letzte Shader zeigt die Kombination aus einem Toonsketch Shader in einem *physical-based environment*, also einer Renderumgebung, welche auf physikalische und realistische Formeln zurückgreift. Hierbei werden unrealistische Grunddarstellungen mit realistischen Lichtern, Materialien oder sogar Umgebungseffekten zusammengeführt. Das Ergebnis

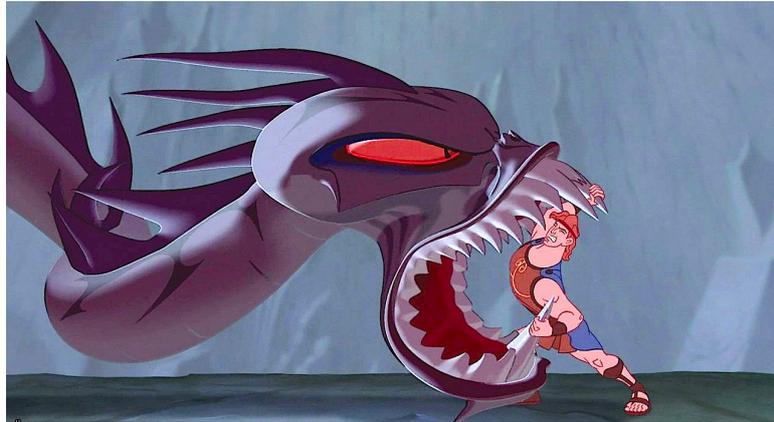


Abbildung 2.4: Die mit Hilfe von Computergrafik erstellte Hydra im sonst komplett handgemalten Film *Hercules* [20].

weist öfters eine gewisse Surrealität oder sogar etwas Träumerisches auf.

2.2.1 Animation

In der Animationsindustrie ist der Film *Schneewittchen und die sieben Zwerge* [22], 1937 von Disney veröffentlicht, die erste in Spielfilmlänge produzierte Animation, welche auf *Cel-Animation* basiert. Der Begriff *Cel-Animation* bezieht sich hierbei jedoch nicht auf eine stilistische Entscheidung, sondern rein auf die Art der Produktion. Ein *Cel*, welches von Celluloid abgeleitet wird, ist der englische Fachbegriff für eine durchsichtige Folie, auf der Objekte für die traditionelle Animation handgezeichnet werden. Diese verschiedenen Folien werden dann übereinandergelegt und ergeben somit das finale Bild. Celanimation hat somit den Vorteil, dass eine Folie öfter verwendet werden konnte und die Arbeitsgebiete auch leichter aufteilbar waren [9]. Der Übergang und die Digitalisierung von Cartoons in den 3D Bereich fand sehr schleichend statt. Als erstes fand die 3D Animation mit Hilfe der Hybrid Animation seinen Platz in der Branche. Schwierige Stellen oder Charaktere wurden als 3D Modell animiert und mit Hilfe von digitalen und schlichten Toon-Shading an den 2D Stil angepasst. So wurde zum Beispiel die Hydra in Disney's *Hercules* [20] komplett in 3D realisiert und noch zusätzlich mit 2D Partikeleffekten versehen. Die Hydra wurde mit Flat-Shading, Schatten und Highlights gerendert. Zusätzlich wurden mit Hilfe von Renderman auch noch Toon Lines generiert. Als Hercules der Hydra ihre Köpfe abschlägt, tritt Schleim aus der Wunde aus, welcher zu Anfangs ebenso mit 3D realisiert werden sollte. Es passte jedoch nicht in den flachen Stil, also wurde versucht, diese Eruption wieder mit 2D zu lösen. Der Unterschied zwischen 2D und 3D Animation bleibt in diesem Fall nicht unbemerkt. In Abbildung 2.4 wird ersichtlich, dass dieser Unterschied sehr positiv zu der Stimmung und des Gefühls des Übernatürlichen und der Gefahr, welche von der Hydra ausgeht, beiträgt [12].

Der erste komplett in 3D erstellte Animationsfilm in Spielfilmlänge ist *Toy Story* [25], welcher 1994 ebenfalls von Disney veröffentlicht wurde. Er glänzte mit einem sehr hohen Produktionswert, der durch realistische Lichtsetzung, Reflexionen und Schattenwürfe begeisterte.

(a) *Jet Set Radio* [36] (2000).(b) *XIII* [33] (2003).

Abbildung 2.5: Die ersten, als solche festgelegten, stilistischen Toon-Shadings in Videospielen.

2.2.2 Games

Videospiele, welche auf computergenerierte Grafik angewiesen sind, sind der Balance zwischen Grafikqualität und Performance schon seit Anbeginn ausgeliefert. Können Animationsfilme die Dauer der Berechnung stellenweise etwas in den Hintergrund rücken lassen, um der Qualität mehr Freiraum zu geben, so sind den Videospielen hier die Hände gebunden. Aufgrund dessen ist das Toon-Shading schon seit Längerem in diesem Gebiet im Einsatz, da dieses doch sehr stilisiert und performant funktioniert. Das erstes Game, welches offiziell mit einem Toon-Shading entwickelt wurde, ist *Jet Set Radio* [36], welches im Jahr 2000 erschienen ist. In Abbildung 2.5a werden klare schwarze Outlines, ein simples Cel-Shading und knallige Farben ersichtlich, welche sich dem Cartoon Stil von japanischem Anime und Manga annähern sollten [59]. Das erste Game, welches sich stilistisch an den westlichen Comics orientiert hat, ist *XIII* [33], welches 2003 erschienen ist. In Abbildung 2.5b ist ein Standbild aus diesem Game zu sehen. Auffallend ist das mehrstufige Cel-Shading, klare Outlines und comictypische Wortunterstützungen, welche Ton und Bild weiter untermalen sollen. Nicht zu sehen ist, wie das Game stellenweise auf die Verwendung von Panels zurückgreift, um auch mit nur einem Bildausschnitt, mehrere Perspektiven und verschiedene Informationen an den Spieler vermitteln zu können.

2.3 Zusammenfassung

Der Ursprung des Toon-Shadings hängt vom jeweiligen Standpunkt des Betrachters ab. Dieser kann sich auf die erste digitale oder analoge Veröffentlichung von stilisierten Games oder Comics beziehen, oder sich rein auf der Anwendung der grundlegenden Eigenschaften des Stils stützen. Das Toon-Shading selbst teilt seine stilistischen Merkmale mit den Comicbüchern, welche 1897 erstmalig auch unter diesem Namen veröffentlicht wurden. Die erste digitale Verwendung dieser Eigenschaften in einem Game unter dem Namen Toon-Shading passierte jedoch erst im Jahre 2000. Prägnante Eigenschaften, wie etwa das Cel-Shading oder dominante Outlines, können nicht nur in den ersten Co-

mics wahrgenommen werden, sondern sind bereits in Höhlenmalereien, welche vor über 20.000 Jahren entstanden sind, zu sehen. Somit sind auch die wichtigsten Merkmale eines Toon-Shadings nicht einer genauen Herkunft verschuldet, sondern dem Herunterbrechen von Charakteristiken per se. Die Grundlagen sind somit ein Mittel zum Zweck, um etwas auch mit wenigen Ressourcen darstellen und festhalten zu können. Diese Art der Darstellung hat sich über die Zeit weiterentwickelt und an die jeweiligen Medien angepasst. So war auch der Comic zu anfangs aufgrund seiner Einschränkungen simplifiziert. Weitere Stilelemente, wie die Detaillinien oder das sogenannte Hatching, sind jedoch erst seit der Entstehung des Comics dokumentiert. Einige dieser Eigenschaften, wie der Colorbleed oder die Ben Day Dots, sind der Qualität der Produktion oder des Druckes verschuldet und somit eher ungewollt entstanden.

In der Entwicklung im 3D Bereich wird auch von NPR gesprochen. Dieser Begriff beinhaltet alle Shader und Stile, welche versuchen eine stilisierte Darstellung von realen Gegebenheiten zu erschaffen. Einige der Eigenschaften des Toon-Shadings sind hier stark vertreten. So sind die Outlines oder das Hatching Merkmale, welche immer wieder zur Anwendung kommen und sofort dazu beisteuern, ein Bild weniger realistisch wirken zu lassen. Das Ergebnis wirkt oft simpel, aber trotzdem klar lesbar.

In der Animation gab Disney den Startschuss für die Verwendung von Toon-Shading in den Animationen in Spielfilmlänge. Jedoch oft auch aus der Motivation heraus, schwierige Stellen oder Charaktere als 3D Modelle zu animieren und den Shader an den Zeichenstil anzupassen, um ihn direkt in den Film übernehmen zu können.

In der Videospieleindustrie ist die Balance zwischen Grafikqualität und Performance seit Anfang an ein wichtiges Thema. Das Toon-Shading ist einer der am performantesten und doch stilisierten Shader, welcher regelmäßig zum Einsatz kommt. Doch auch hier wird manchmal getrickst oder versucht sich so nah wie möglich am Comicstil zu halten, damit dieser zur Hauptästhetik des Games wird und nicht als Notwendigkeit der Performance zu sehen ist.

Kapitel 3

Toon-Shading in der Animation

Animationen wollen dem Zuseher eine Geschichte erzählen und diese durch eine visuelle Ästhetik unterstützen. Sie versuchen den Stil der zu erzählenden Reise anzupassen. Toon-Shading wird oft durch die visuelle Einfachheit Kinderfilmen und -Serien zugeordnet. Dieses Unterkapitel soll aufzeigen, dass Toon-Shading keines Falls einfach und dementsprechend nicht nur dem jüngeren Publikum zuzuordnen ist [8].

3.1 *Love, Death & Robots*

Love, Death & Robots [21] ist eine amerikanische Serie, welche am 15. März 2019 auf *Netflix* publiziert wurde. Das Besondere an dieser Serie ist, dass unterschiedlichste Produktionsteams involviert sind und diese auch keine Einschränkungen im Bezug auf deren Stil unterliegen. Dadurch sammelte sich eine breite Menge an verschiedenst produzierten und ästhetisch unterschiedlichen Animationen an [77].

3.1.1 Episode 4: *Suits*

Die vierte Folge *Suits* wurde von *Blur Studio* [43] erstellt. Das Spannende an dieser Folge ist, dass *Blur Studios* hierbei selbst sagt, dass der non-photorealistische Stil für sie eine Herausforderung war. So hat dieses Team bereits Visual Effects für Filme wie *Avatar* [17] oder *Deadpool* [18] produziert, welche jedoch alle realistische Anforderungen



Abbildung 3.1: Verschiedene Einstellungen aus *Suits* [21].



Abbildung 3.2: Verschiedene Einstellungen aus *Blind Spot* [21].

beinhalteten und für sie somit die illustrierte Ästhetik eine neue Erfahrung bereitstellte. So mussten sie etwa die Realität des Lichtes, also die physikalisch korrekte Herangehensweise der Beleuchtung eines Objektes, brechen. Ein Beispiel ist hier das sehr dominante Rimlight, welches maßgebend zu dem Stil beiträgt. Diese Rimlight hat in vielen Einstellungen keinen realistischen Ursprung und musste somit im Prozess des Renderns noch gesondert behandelt werden, damit es nicht von anderen Schatten mit beeinflusst wird [61].

In Abbildung 3.1 sind vier Einstellungen der Folge zu sehen. Das zuvor angesprochene Rimlight ist in jeder einzelnen davon sehr prominent und trägt stark zur Formgebung, dem Fokus und der Unterscheidung zwischen Vorder- und Hintergrund bei. Interessant ist hier auch, dass zu fast jedem rötlichen Rimlight, sich außerdem ein bläuliches auf der anderen Seite befindet, auch wenn dies logisch gesehen nicht immer Sinn macht. So gibt es in der letzten Einstellung in der Abbildung keinen Ursprung für das bläuliche Licht auf der rechten Seite. Auch werden nur die Charaktere, welche sich gerade im Fokus befinden mit diesem Licht ausgestattet, was den Blick des Zusehers leichter lenken lässt und auch einen guten Kontrast zum Hintergrund bietet. Zusätzlich dazu wird ebenso sehr stark mit Depth of Field gearbeitet. Dies hilft nicht nur den Charakter besser abzugrenzen, sondern gibt dem Hintergrund auch noch eine malerische Ästhetik, welche mit den verwendeten handpainted Textures noch mehr gestützt wird. Ebenfalls wirken die Splattereffekte und Explosionen als wären sie selbst in einem Comicstil gezeichnet und hinzugefügt und helfen der Animation, mehr in eine stilistische Toonanimation eingeordnet zu werden. Die Schatten in Suits sind einerseits bereits in die Texturen eingezeichnet und andererseits werden stellenweise typische harte Toonshadowen beziehungsweise Cel-Shading oder auch leicht auslaufende, weiche Schatten verwendet.

3.1.2 Episode 15: *Blind Spot*

Die fünfzehnte Folge von *Love, Death & Robots* trägt den Titel *Blind Spot* und wurde von einem sechsköpfigen Kernteam erstellt [73].

In Abbildung 3.2 werden vier Einstellungen der Folge ersichtlich. Das erste Bild zeigt die Roboterfrau Kali. Hauptmerkmal sind die sehr groben und harten Schatten, welche sich auf ihrem Gesicht abzeichnen. Auch besitzt sie dünne Linien, welche ihre Form definieren und sie mit einer Outline umrahmen. Ihre Nase wird zusätzlich durch eine

Kantenzzeichnung noch hervorgehoben. Hier wurde eine dickere Linie verwendet, welche auch gleich als Schatten fungiert und somit formgebend ist.

Im zweiten Abschnitt ist Hawk zu sehen, wie er sich vor den Scheinwerfern eines Fahrzeuges befindet. Das Licht wirft ein starkes Rimlight auf ihn. Dieses Licht ist im Gegensatz zu dem Cel-Shading nicht einfarbig, sondern beinhaltet einen Bloom und ein Strahlen, welches den Stil von dem herkömmlichen Comiclook abhebt. Nichts desto trotz wird auch dieses Rimlight auf eine eher größere Fläche angewandt und wirkt im Standbild schon fast wie eine Aura. Dem wird durch eine gewisse Unregelmäßigkeit entgegengewirkt.

Die dritte Einstellung zeigt einen Roboter, welcher von einer Explosion wegfährt. Der Roboter ist genauso ein Charakter wie Kali und Hawk, aber genau in dieser Einstellung wird er in einem ganz anderen Stil dargestellt. Dies ist ganz alleine dem strahlenden Licht zu verschulden, welches ihm einen sehr weichen Farbverlauf verpasst. Die Outlines sind fast nicht mehr wahrnehmbar und auch die Formgebung und Definierung passiert hier über das weiche rote Rimlight und dem starken Kontrast. Die Explosion selbst erinnert an eine typisch 2D gezeichnete Comicexplosion, welche sich selbst nicht erhellt und die verschiedenfarbige Bereiche sehr klar voneinander unterscheidbar und abgrenzbar darstellt. Einzig allein an den Outlines der Explosion ist der Bloom ersichtlich.

Das letzte Bild zeigt eine gezielte Gegenüberstellung zwischen bewegtem Charakter und einem statischen Umgebungsobjekt. Das statische Objekt besitzt keine wirklichen Outlines mehr, sondern an die Farbe angepasste Linien, welche die Farbareale abgrenzen. Es besitzt einerseits einen sehr weichen Schatten, aber andererseits wird der harte Schatten an den Zähnen zusätzlich aufgenommen. Auch wird das Licht, welches leicht von hinten nach vorne strahlt, an dem Objekt sichtbar. Der Charakter hingegen besitzt einen einfarbigen Körper, beziehungsweise Stellen, welchen nur durch den sehr harten Schatten Tiefe gegeben wird. Die Outlines, welche seinen Körper und die Figur definieren, sind ebenso klar ersichtlich. Das Licht, welches aus dem Tunnel kommt, wird hierbei überhaupt nicht berücksichtigt.

3.2 *Sonder*

Die Animation *Sonder* [23] wurde von *Soba Productions* produziert und in Unity 2017 [38] gerendert. Obwohl die Game-Engine als Rendertool verwendet wurde, fällt *Sonder* nicht unter realtime Animation, da nur die Einzelbilder abgespeichert und in Nuke weiter für die Postproduction eingeschleust wurden. Der Film selbst lässt sich in Unity nicht als fertige realtime Animation abspielen [78].

Nichtsdestotrotz ist *Sonder* ein interessantes Beispiel für die Produktion in Unity, da einerseits stark auf für die Game-Engine angepasste Modells eingesetzt und viele Eigenschaften und Stilisierungen in eben dieser eingebunden werden mussten. In Abbildung 3.3 wird zum Beispiel der in Unity entwickelte Shader gezeigt. Er basiert auf einem typischen Toon-Shading, welches jedoch einen leichten Verlauf erhielt, um nicht zu comichaft zu wirken und sich von anderen Unity Projekten zu unterscheiden [41].

Das Interessante an diesem Shader ist jedoch, dass er in unterschiedlichen Shots auch unterschiedlich eingesetzt wird. Der erste Shot in Abbildung 3.3 zeigt, wie das Toon-Shading fast die ganze Animation lang sichtbar ist. Es beinhaltet einen sehr großräumigen Schatten, der an den Kanten leicht ausläuft. Der Schatten, welcher von dem



Abbildung 3.3: Verschiedene Handhabungsarten des Schattens in *Sonder* [23].

Laub des Baumes auf den Protagonisten geworfen wird, ist im Gegensatz dazu ein sehr weicher Schatten.

Im zweiten Bild dominiert hier ganz stark ein harter Schatten. Die geworfenen Abdunkelungen der eigenen Hände, des Holzes und des Gegenstandes im rechts unteren Eck beinhalten gleichermaßen eine harte Kante, wie der Lambertschatten den er selbst auf seiner Kleidung durch die Lichtsetzung enthält. Die einzige Ausnahme macht seine Haut, welche immer noch einen leicht auslaufenden Schatten wirft.

Die dritte Einstellung zeigt, wie er mit starker Sonnenbestrahlung im Rücken gebückt über einem Fluss steht. Das Cel-Shading auf seiner Kleidung hat sich hierbei erneut verändert. Es ist nun kein weich auslaufender und auch kein purer harter Schatten mehr, sondern ein zweistufiges Shading, welches einen Verlauf über eine harte Abstufung darstellt.

In dem letzten Bild wird ersichtlich, dass auch der weich auslaufende Schatten auf die Kleidung angewandt wird und somit nicht immer eine Differenzierung zwischen dieser und seiner Haut stattfindet. Der Shader ist somit nicht nur vom Licht abhängig, sondern auch von der Stimmung die er vermitteln soll und wird subtil von Shot zu Shot angepasst. Auch gibt es einen Unterschied des Schattenwurfes von Objekten. Der Protagonist wirft ausschließlich harte Schatten auf den Boden, wohingegen die Umgebung nur weiche Schatten generiert. Rimlight wird bis auf wenige Ausnahmen sehr spärlich bis gar nicht verwendet. Auch gibt es bewusst keine Outlines, um beabsichtigt keinen Gamelook zu erzeugen. Die Animation beinhaltet auch keine Spezialeffekte, welche unter 2D oder Flatdesign einzuordnen sind.

Die Abgrenzung zur Umgebung erfolgt stark über die unterschiedlichen Farben. Der Hintergrund ist meist sehr gedeckt gehalten, wobei der Protagonist ein helleres rötliches Hemd und eine blaue Jeans trägt. Eine sehr dominante Tiefenunschärfe trägt zusätzlich zur leichteren Unterscheidung bei und die gezielte Lichtsetzung rückt den Protagonisten zur Gänze in den Mittelpunkt.

3.3 *Spider-Man: Into the Spider-Verse*

Der Animationsfilm *Spider-Man: Into the Spider-Verse* [24] war 2018 erstmalig im Kino zu sehen und gewann zahlreiche Preise als Best Animated Feature Film. Grund dafür ist unter anderem der herausstechende Stil. Production Designer Justin Thompson sagte



Abbildung 3.4: Standbildausschnitte aus dem Film *Spider-Man: Into the Spider-Verse* [24].

in einem Interview mit *IndieWire* [46], dass er bei Filmen, welche aus Comics entstanden, immer das Gefühl hatte, etwas sei in der Übersetzung verloren gegangen. Darum dachte er, es wäre grandios einen Film zu erstellen, welcher quasi aus dem Leben eines Comicbuches erzählt wird, inklusive Ben-Day Dots, Screen Tones, und Line Work.

In einem Video, veröffentlicht von *The New York Times* [76], sprechen die Regisseure Bob Persichetti, Peter Ramsey und Rodney Rothman über die visuellen Besonderheiten von Miles Unviversums und auf welche Eigenschaften gezielt Wert gelegt wurde. So erwähnten sie die sogenannten *Kirby Dots*¹, welche in den 80ern und 90ern Jahren in Comics sehr beliebt waren und im Film für Spezialeffekte verwendet werden. Auch wird auf das vollkommene Fehlen der typischen Tiefenunschärfe hingewiesen. Statt eines normalen Blurs wurde eine chromatische Aberration verwendet. Dieses Stilmittel hat seinen Ursprung aus dem mangelhaften Comicbuchdruck, welche eine Illusion von Verschwommenheit vermittelt.

In Abbildung 3.4 sind vier verschiedene Ausschnitte aus dem Film zu sehen. In dem ersten Ausschnitt wird der allgemeine Stil des Filmes sehr gut ersichtlich. Der Charakter per se ist eigentlich nicht mit einem Toon-Shading ausgestattet, aber der Film selbst vermittelt das allgemeine Gefühl eines Comics. Miles ist nicht flach schattiert und texturiert, sondern besitzt einen eher handgezeichneten Stil. Dieser resultiert aus handpainted Textures, welche mehr an Wasserfarben erinnern. Die Lichter und Schatten fallen eher realistisch, sprich sie laufen sehr weich aus und definieren den Charakter auch mit dezenten Highlights. Spannend ist hierbei, wie zusätzlich mit Schatten und Licht umgegangen wird. Besonders gut ist dies an der Lichtquelle sichtbar, welche aus dem Hintergrund von dem Polizeiauto aus nach vorne strahlt. Das Licht selbst besitzt eine Textur. Der gesamte Film – bis auf eine charakterliche Ausnahme – wird von Halftones und generellem Toning des Schattens und des Lichts dominiert. Das heißt, es werden eben diesen Texturen zugewiesen, welche die zu beeinflussenden Stellen schraffieren, oder mit Punkten versehen. Es wurde hier genau darauf geachtet die comictypischen Schattierungsweisen und Möglichkeiten anzuwenden. So wird alles, was von Licht getroffen wird, nicht weich erhellt, sondern mit hellen Punkten versehen, welche, je weiter

¹Kirby Dots, oder auch Kirby Krackle genannt, ist eine stilisierte Darstellungsart für Spezialeffekte wie etwa Explosionen, kosmische Energie oder Ähnliches. Dazu wird hauptsächlich mit Punkten und Kugeln gearbeitet, welche in unterschiedlichem Größen und Abstand voneinander platziert werden.

die Lichtquelle entfernt ist, langsam verschwinden [47]. Josh Beveridge, zuständig als Head of Character Animation, sagte in einem Interview mit *Awn* [68], dass kein Toon-Shading verwendet wurde, aber es definitiv Parallelen zu einem gibt. Es gibt auch nicht den einen Shader der verwendet wurde, es waren vielmehr mehrere komplexe, welche für verschiedene Lösungsansätze angepasst und ständig verändert wurden. In diesem Bild wird auch sehr gut sichtbar, dass ebenso mit komplementären Rimlights gearbeitet wurde. Zwar treten diese nicht so häufig auf wie in *Suits*, sind aber dennoch in stimmungsreichen Szenen vorhanden.

Die zweite Einstellung zeigt eine sehr malerische Verfolgungsjagd, welche durch harte und genauso präzise weichere Schatten und ein sehr dominantes Rimlight geprägt ist. Der Schatten wurde hier anders gehandhabt als in den anderen Szenen. Er besitzt die sonst verwendete Textur nicht so stark wie in anderen Einstellungen, hat aber dafür stellenweise ein grobes Halftoning im Übergang zwischen Schatten und Licht.

In dem dritten Bild werden die Linien, welche als Stilelement und Unterstützung der Ausdrücke verwendet werden, gut dargestellt. Das Spezielle an diesen Linien ist, dass keine herkömmliche Edge Detection zur Anwendung kommt, sondern diese durch eine Kombination aus händischer Platzierung und Machine Learning entstehen. Einerseits wurde ein System mit geriggten 3D Linien aufgebaut, um den Künstlern ein gezieltes Eingreifen in das Bild zu ermöglichen, andererseits wurde zusätzlich dieses durch künstliche Intelligenz erweitert und somit versucht den Prozess und die Produktivität der Künstler zu beschleunigen. Dafür wurden Trainingsdaten für jeden Charakter, in Form von verschiedenst positionierten Turntables, eingeschleust und somit das ganze System adjustiert. Die zweiten Trainingsdaten verwendeten das Ergebnis des ersten Durchlaufes, welche händisch angepasst wurden. Das finale Modell des Machine Learnings baute nur auf diesen zwei Trainingsdaten auf und wurde Basis für den gesamten Film. Nichts desto trotz platzierten die Artists händisch noch weitere 3D Linien, um den Charakteren noch mehr Ausdrucksstärke zu verleihen oder Bewegungen weiter zu unterstützen [71].

Im letzten Ausschnitt von Abbildung 3.4 werden die erwähnten Linien noch stärker sichtbar. Es zeigt auf, wie es sich stellenweise nicht nur um normale Linien handelt, sondern sogar um mehrere Linien übereinander, welche einen handgezeichneten und sogar skizzierten Eindruck vermitteln. Auch ist das bewusste Einsetzen von dem sogenannten Color Bleed zu sehen. Die Outline der Iris im Auge entspricht hierbei nicht der richtigen Outline der Textur. Vielmehr geht die Farbe der Iris noch über diese Linie hinaus und zeigt hiermit eine bewusste Entscheidung für eine unsaubere Darstellung auf.

In Abbildung 3.5 wird ersichtlich, wie dieser Wunsch, einen Comic in das Filmformat zu übersetzen, nicht nur auf eine Art des Comics, sondern auf verschiedene Stile angewandt wurde. Jeder dieser Charaktere hat seinen Ursprung in einer anderen Parallelwelt und somit auch eine andere Art der Darstellung. Von links nach rechts sind Gwen Stacy, Peni Parker, Miles Morales, Peter B. Parker, Peter Parker als Spider-Man Noir und Peter Porker oder Spider-Ham zu sehen. Gwen Stacy, Miles und Peter B. Parker haben einen ähnlich gewählten Stil, welcher von den technischen Aspekten kaum Unterschiede aufweist. Die anderen drei jedoch um so mehr. So kommt Peni Parker aus einem Anime-Universum, welches sich klar an ihrem Cel-Shading und auch an den prominenten arealen Reflexionen bemerkbar macht. Die Textur, welche ihr hierbei im Schatten fehlt, wird jedoch im Detail ihrer Kleidung wiedergegeben. Diese ist somit ein subtiles Bindeglied zwischen der Hauptästhetik und ihrer eigenen Darstellung. In einem



Abbildung 3.5: Alle Spider-Charaktere in *Spider-Man: Into the Spider-Verse* [24].

Interview mit *IndieWire* [46] wird außerdem hervorgehoben, dass Peni Parker aus einer Kombination von CG und 2D Elementen besteht. Auch ist Peni sehr flach geshadet und besitzt keine sichtbare Normalmap, welche ihr mehr Tiefe und Details gibt. Sie besitzt auch keine fixen Outlines, bis auf die Momente, in denen sich ihre Hauttöne überlagern. Dies betrifft zum Beispiel Szenen, bei denen sie die Arme abwinkelt und somit eine Differenzierung zwischen Oberarm und Unterarm stattfinden muss, oder einer fast ständig sichtbaren Linie, welche Kinn und Hals abgrenzt.

Das Universum von Spider-Man Noir spielt in dem Jahre 1933. Allein durch die Bildsprache und Ästhetik wird diese Zeitepoche sehr gut sichtbar. Spider-Man Noir sticht durch sein dunkles Erscheinungsbild und die starken Kontraste heraus. Im Gegensatz zu den Anderen ist nicht sein Schatten texturiert, sondern sein komplettes Aussehen. Dieses weist die drucktypischen Ben-Day Dots auf. Teile seiner Gewandung sind tief-schwarz und das Licht hellt die betroffenen Stellen bis zu einem Reinweiß auf. Auch seine chromatische Aberration besteht nur aus schwarz und weiß und keinen Farbtönen. Die Reflexionen seiner Fliegerbrille sind schräge Rechtecke, wie sie oft in minimalistischeren Zeichnungen gefunden werden. Ein sehr dominantes Rimlight definiert stetig seine Silhouette. Auch Details seiner Kleidung weisen stellenweise helle Highlights auf, um ihn leichter lesbar zu machen. Der Schatten selbst läuft sehr weich aus. In Einstellungen, in denen Spider-Man Noir nahe an der Kamera ist und er durch die Lichtsetzung wenig aufgehellt wird, wirkt er dank seinem Dot-Muster, den starken Kontrasten und großflächig einfärbigen Arealen, als wäre er als 2D-Figur hinzugefügt worden.

Spider-Ham hat seinen Ursprung aus einem 36 Jahre alten Scherz, welcher kurz darauf als Comic realisiert wurde. Im Bezug auf sein Erscheinungsbild waren die größte Inspiration die *Looney Tunes* Cartoons selbst [51]. Peter Porker ist wie Peni Parker sehr flach coloriert. Er besitzt kein typisches Cel-Shading, da sein Schatten sehr weich ausläuft. Jedoch enthalten seine Highlights und belichteten Areale hart Kanten, welche an dem Übergang der Hauptreflexion mit einem sehr groben Halftoning versehen sind. An diesem Halftoning wird sehr gut sichtbar, dass die Textur nicht an den Charakter gebunden ist, sondern eher über die Kamera oder Position selbst auf den Charakter projiziert wird. Das heißt, wenn Peter Porker seinen Kopf dreht, dreht sich die Textur nicht mit, sondern bleibt genauso erhalten. Auch besitzt er ein sehr präsenten Rimlight, welches seine Silhouette und Form definiert.



Abbildung 3.6: Verschiedene Szenen aus dem Kurzfilm *Feast* [19].

3.4 *Feast*

Feast [19] ist ein Kurzanimationsfilm aus dem Hause Disney. Er wurde 2015 mit einem Oscar als bester animierter Kurzfilm ausgezeichnet.

Die erste Einstellung aus Abbildung 3.6 zeigt den Hauptcharakter Winston. Ziel der Animation war, den Charme eines handgezeichneten 2D Films einzufangen und darzustellen. Dazu wurde ein komplett flaches Shading gewählt. Der Schatten ist einstufig und besitzt, egal in welcher Belichtungsart, immer nur einen hellen und einen dunklen Bereich. Diese Kombination war im Bezug auf Bewegungen, im speziellen Rotationen, problematisch, da hier gewisse Formen und Veränderungen der Position nicht gut zu lesen sind. Abhilfe schaffte hier die Entscheidung, einen zweifarbigen Hund zu wählen und die restlichen unübersichtlichen Stellen mit händisch platzierten Linien zu stützen. So sind etwa die Vorderbeine des Hundes von den Hinterbeinen klar durch eine dünne schwarze Linie getrennt. Hervorzuheben ist auch der gewählte Lichtbereich, welcher klar Vorder- und Hintergrund trennt und auch die Grenze für die Tiefenunschärfe vorgibt. [57]

Das zweite Bild ist dem ersten sehr ähnlich. Auch hier besitzt Winston eine einstufige Schattierung. Zusätzlich erhielt er auch noch Detaillinien am inneren Hals und Oberkörper und erhellte Schnurrhaare. Interessanter ist jedoch, wie der Futternapf behandelt wurde. Er weist nicht die typische flache Schattierung auf, denn er enthält eine Grundfarbe, welche jedoch im Inneren weitere Abstufung hat. Zusätzlich wurde eine realistischere Lichtreflexion an dessen linker Seite gewählt, um dem Napf seine typische Rundung zu verpassen. Auch ist hier die Textur, welche an jeder Kante angewandt wird, sehr gut sichtbar. Diese Textur, welche Kanten und Übergänge gröber auflöst und körniger gestaltet, soll das Gefühl einer Papierzeichnung vermitteln und den Stil von einem herkömmlichen Toon-Shading abheben [57].

Die letzte Einstellung aus Abbildung 3.6 zeigt noch eine weitere, gesonderte Variante der Schattierung. Die Fleischbällchen der Spaghetti im Vordergrund besitzen zusätzlich zu dem typischen zweistufigen Schatten noch eine eigene Verdunkelung an der Gegenseite des Lichtes. Auch werden die unscharfen Nudeln mit Highlights betont, obwohl sich diese bereits im Licht, also dem hellen Bereich des Shaders, befinden. Die handgesetzten schwarzen Linien an Winstons Stirn, wurden für den Ausdruck seiner Stimmung verwendet. Das Licht, welches in diesem Blickwinkel als Rimlight fungiert, wird auch

nicht als Abstufungen der Farben angewandt, sondern hellt die beleuchtete Silhouette gleichermaßen auf. Die simulierten Staubkörner im Hintergrund in Verbindung mit der Tiefenunschärfe geben dem Zuseher noch zusätzlich ein Gefühl für die Räumlichkeiten [57].

3.5 Zusammenfassung

Toon-Shading, welches in Animationen zur Anwendung kommt, ist nicht grundsätzlich Low-Budget oder eine sehr simple Darstellung. Alle genannten Beispiele zeigen, dass auch, wenn ein gemeinsamer Nenner vorhanden ist, das Ergebnis sehr weit voneinander abweichen kann. Eines der gemeinsamen Merkmale ist eine sehr präzise und durchdachte Ausleuchtung und Lichtsetzung. Durch diese gezielt angewandten Lichter, ergeben sich komplett neue Möglichkeiten ein Toon-Shading anzupassen oder zu individualisieren. Stellenweise wurde das Licht, wie etwa in der *Love, Death & Robots* Folge *Suits*, sogar in den Shader eingearbeitet. Dies ist besonders auffallend an den Rimlights zu sehen, welche die Konturen der Figuren besser lesbar machen und durch eine gezielte Farbgebung auch Stimmung und Ästhetik maßgebend beeinflussen können. Auch gibt es starke Abweichung bei den Schattierungen. Das typische Cel-Shading als einzelnes Schattenareal oder mit Abstufungen ist sehr wohl vertreten, aber auch eine Adaptierung mit Hilfe von Rampmaps oder Texturen wie etwa Hatching oder Dotting werden angewandt. Ebenso werden Outlines verschiedenst interpretiert und eingesetzt. Outlines oder schwarze Linien, welche die Details der Charaktere weiter hervorheben, sind zwar nicht so häufig vertreten, haben aber definitiv einen Platz in der Stilfindung von preisgekrönten Animationen verdient. *Spider-Man: Into the Spider-Verse* geht sogar so weit, dass noch weitere comictypische Aspekte in den Stil eingearbeitet wurden. So finden sich auch wackelige Linien und ein Colorbleeding in der Texturierung in diesem Animationfilm wieder.

Kapitel 4

Toon-Shading in Games

Toon-Shading gehört zu den zeitlosesten Darstellungsarten in der Spieleindustrie [40]. Obwohl das Toon-Shading von der Eigenschaft des Cel-Shadings dominiert wird, gibt es viele andere Arten und Weisen diesen noch weiter zu definieren und zu differenzieren. Dieses Unterkapitel zeigt verschiedene Games, welche in ihren Genren berühmt für den gewählten Artstil sind, auf und analysiert deren visuelle Besonderheiten.

4.1 *The Legend of Zelda: The Wind Waker*

The Legend of Zelda: The Wind Waker [31] ist ein Computerspiel, welches 2001 für den Nintendo Gamecube erschienen ist. Es ist das erste Game der Zelda-Reihe welches mit dem herkömmlichen Stil gebrochen und sich für einen Cartoon-Stil entschieden hat. Diese Veränderung macht den Titel auch zu einen der meist diskutieren Games von Nintendo. Es wurde auf Grund seines Stils als Kinderspiel bezeichnet, gilt aber heute als Klassiker der Serie. [56]

In Abbildung 4.1 werden verschiedene besondere Eigenschaften des Shaders in *The Legend of Zelda: The Wind Waker* ersichtlich. Im ersten Bildabschnitt ist zu sehen, wie das Toon-Shading unter normalen Lichtbedingungen Link darstellt. Link besitzt sehr flache und kräftige Farben, wobei kleine Details, wie etwa der braune Gürtel, bereits in die Textur eingearbeitet sind. Der Schatten ist ein sehr comictypisches Cel-Shading,



Abbildung 4.1: Verschiedene Spielausschnitte aus *The Legend of Zelda: The Wind Waker* [31] und *The Legend of Zelda: The Wind Waker HD* [32].

welches einen sehr harten Übergang besitzt. Der Hintergrund selbst steht im starken Kontrast mit dem Charakter, da er hier auch größere Details, wie etwa der Türhenkel, in die Textur eingearbeitet sind und selbst eher flach wirkt. Der Schatten welcher von Link geworfen wird, ist ebenfalls ein harter Schatten.

Die zweite Einstellung zeigt eine Nahaufnahme von Links Gesicht, wodurch eine sehr spezielle Eigenschaft seiner Erscheinung sichtbar wird. Seine Augen und die Augenbrauen werden über den Haaren gerendert und somit nie von eben diesen verdeckt. Grund dafür ist einerseits die Anlehnung an den in Japan typischen Stil der Mangas, welche oft Augen trotz überlappende Haarsträhnen durchgängig und kräftig darstellen und andererseits der Motivation Link in diesem Game mehr mit der Umwelt interagieren zu lassen. Diese Interaktion drückt sich durch die Fixierung seiner Augen auf Objekte oder möglicherweise interessante Stellen in der Umgebung aus. Somit kann die Spielfigur, subtil und ohne die Immersion zu brechen, dem Spieler Hinweise geben. Auch gibt er durch seine Mimik Eindrücke in seine Gefühlswelt, welche durch die Wahl dieser Darstellungsart vielmehr in den Vordergrund rücken [50].

Der dritte Bildabschnitt zeigt wie *The Legend of Zelda: The Wind Waker* mit mehreren Lichtern umgeht. Links und rechts von dem Charakter befinden sich zwei Fackeln, welche das Hauptlicht – abgesehen von der Grundhelligkeit – in dem Raum ausmachen. Der Charakter per se wird aber nur von einem einzigen Licht beeinflusst, in diesem Fall jenes welches rechts von ihm steht. Geht Link ein paar Schritte in die entgegengesetzte Richtung, so wird das Hauptlicht auf das linke gewechselt und so auch Schattierung und Highlights angepasst. Der Übergang ist eine weiche Translation des Lichtes, welche erst bei genauerem Hinsehen auffällt und der Immersion des Games keinen Abbruch tut [72].

Das vierte Bild zeigt Link auf seinem Transportmittel namens Roter Leuenkönig. Hier wird ersichtlich wie nicht nur Link sich von Hintergrundobjekten, sondern auch von weiteren Charakteren abhebt. Der Rote Leuenkönig besitzt viel mehr Details in seinen Texturen und die Tiefe der Plankenzwischenräumen wird durch eine gezeichnete Schattierung vermittelt. Auch an den Stellen, welche einfärbig gelöst werden hätten können, besitzt er eine Struktur oder zumindest eine zweite Farbe, welche die Hauptfarbe umrandet oder mit Details versieht.

Der letzte Bildabschnitt zeigt Link in der Neuauflage *The Legend of Zelda: The Wind Waker HD* [32], welche 2013 für die Wii U erschienen ist. Hierbei wurden nicht nur Texturen verfeinert, sondern auch verschiedenste Beleuchtungsmöglichkeiten mehr genutzt. So wird hier ersichtlich, wie der Charakter erhellt wird, die Textur verschwimmt und zusätzlich ein Rim Light erhält. Der Bloom sorgt dafür dass Objekte nicht mehr so stark hervorgehoben werden und die Farben somit auch an Intensität verlieren. Diese Änderungen stießen bei manchen Fans erneut auf Kritik, da die Neuerungen ihrer Meinung nach zu realistisch für den toonhaften Stil wirken [62].

4.2 *The Legend of Zelda: Breath of the Wild*

The Legend of Zelda: Breath of the Wild [30] ist ein Game, welches 2017 auf der Nintendo Switch veröffentlicht wurde. Es gilt immer noch als eines der besten Games für diese Konsole und führt die verschiedensten Rankings an [42]. Es werden Stil und Mechaniken gleichermaßen gelobt und auch das Zusammenspiel dieser hervorgehoben.



Abbildung 4.2: Verschiedene Spielausschnitte aus *The Legend of Zelda: Breath of the Wild* [30].

In Abbildung 4.2 sind vier verschiedene Einstellungen aus *The Legend of Zelda: Breath of the Wild* zu sehen. Im ersten Ausschnitt wird die Grundlage des Shaders ersichtlich. Der Schatten auf Link's Haut ist einstufig und ein typisches Cel-Shading. Der Schatten auf seinen Haaren besitzt jedoch eine Zwischenfarbe und ist somit zweistufig. Auch Highlights werden als harte Areale angewandt. Es kommt zusätzlich noch ein sehr helles Rimlight auf der Seite des Lichtes zum Tragen, welches nicht nur seine Silhouette hervorhebt, sondern auch Details wie etwa die Form seiner Lippen definiert.

Im zweiten Bild wird ein zusätzliches Detail des Shaders sichtbar. Manche Highlights werden hier nicht als flaches Cel-Shading angewandt, sondern zusätzlich noch durch eine Textur geschleust. So besitzt etwa Link's Schulter, seine Waffe oder sogar seine Schuhe eine Schraffierung, die seiner Erscheinung einen gezeichneteren Look verleihen. Auch besitzt die Waffe im Gegensatz zur Kleidung nicht nur eine flach colorierte Oberfläche. Die handpainted Textures besitzen zusätzlich zu den typischen verschwommenen Farbvariationen noch eingearbeitete Unebenheiten und Schattierungen, welche Form und Struktur weiter definieren. Das Rimlight wird hier ebenso angewandt. Dieses Rimlight wird auch in der spielbaren Umgebung eingesetzt und kommt immer dann zum Tragen, wenn Charaktere oder Objekte gegen das direktionale Hauptlicht betrachtet werden.

Der dritte Abschnitt zeigt einen Nebencharakter, welcher vor einem Lagerfeuer steht. Diese zusätzliche Lichtquelle wird vom Shader ebenso aufgenommen und mit einer farblich angepassten Erhellung der betroffenen Stellen vermittelt. Das Feuer selbst hellt seine Umgebung jedoch nur mit einem Bloom auf. Die Umgebung besitzt im Gegensatz zu den Charakteren einen eher realistischeren Ansatz. Es werden weiche Schatten verwendet und auch die Texturen sind mit vielen Details und realistischen Farben und Strukturen angereichert. So ergibt sich ein starker Kontrast zwischen Charakteren, Objekten und der Umgebung selbst. Dies besitzt die Grundlage, dass das Game *The Legend of Zelda: The Wind Waker* die Basis für die Stilfindung war. Das Hauptproblem, welches mit diesem Stil auftrat, war jedoch, dass ein Hauptelement das realistische Verhalten von physikalischen Berechnungen und in der echten Welt vorkommenden Geschehnissen ist. Dieser Realismus war mit dieser Toonästhetik nicht vereinbar, da der Spieler Erkenntnisse, welche er in der echten Welt erlangt hat, auch auf das Game übertragen können sollte und ein komplettes Game im Cartoonlook keine fühlbaren Regeln oder realistisches Verhalten vorweist. Dies ist auch der Grund wieso dieser Stil weiterentwickelt und mit einer realistischeren Umgebung kombiniert wurde [44, 45].



Abbildung 4.3: Verschiedene Spielausschnitte aus *The Wolf Among Us* [29].

Das vierte Bild zeigt Link in einer sehr diffus beleuchteten Umgebung. Es ist ein Gewitter aufgezogen und die Sonne und deren Stand am Himmel nicht mehr sichtbar. Selbst der Schatten auf dem Boden ist nun sehr weich dargestellt und nicht mehr richtig zuordenbar. Trotzdem besitzt der Shader von Link immer noch eine klare Richtung und die selben kräftigen Farben und Abstufungen. Auch ist noch das sehr helle Rimlight zu sehen, welches auf dem Stein etwas dezenter angewandt wird.

Der letzte Ausschnitt aus Abbildung 4.2 macht ersichtlich, wie die Umgebung auf Distanz funktioniert. Texturen werden immer verschwommener und ein starker Nebel gibt dem Spieler ein Gefühl für Distanz. Dieser Nebel lässt weiter entfernte Objekte immer unklarer erscheinen, bis nur mehr die Informationen für die Form wahrgenommen werden können. Diese Form wird weiterhin durch das Rimlight gestützt. Auch grenzen sich somit Charaktere in weit entfernten Distanzen vom Hintergrund ab und können vom Spieler leichter wahrgenommen werden.

4.3 *The Wolf Among Us*

The Wolf Among Us [29] ist ein Game, welches 2013 von Telltale veröffentlicht wurde. Es gilt als eine Fortsetzung der *Fables* [16] Comic Serie.

In Abbildung 4.3 sind vier verschiedene Bilder aus dem Game zu sehen. Deutlich wird, dass *The Wolf Among Us* ganz gezielt auf einen Comiclook abzielt. Klar abgegrenzte Bereiche, dunkle Schatten, hohe Kontraste und die typischen schwarzlinigen Details sind ein Indiz dafür. Der erste Bildausschnitt zeigt eine Nahaufnahme von Bigby Wolf, dem Hauptprotagonisten des Games. Auffallend sind hier der dreistufige harte Schatten, welcher die Grundform des Charakters definiert und die sehr dominanten, schwarzen Linien, welche die Details hervorheben. Diese Außenlinien sind nicht animiert und deswegen wahrscheinlich bereits in die Textur integriert. Zusätzlich ist Bigby auch mit einer Outline ausgerüstet, welche nicht durchgehend die gleiche Strichstärke aufweist, sondern ebenso variiert. Er ist außerdem nicht komplett flach coloriert, sondern weist eine Art Wasserfarbeneffekt auf, der ihn etwas unsauber wirken lässt. Auch sind stellenweise Belichtungen und Schatten des Körpers selbst in die Textur eingearbeitet. Dies ist zum Beispiel beim Hals und den Schlüsselbeinen der Fall. Der Hintergrund beinhaltet in diesem Fall keine Unterschiede. Er wirkt aufgrund des mangelnden Schattens und der nicht vorhandenen Tiefe sehr flach und hebt sich bis auf den Detailgrad

nicht vom Charakter ab.

In Bild zwei ist Bigby als Detektiv im Einsatz. Dies begründet den stellenweise stilistischen Wandel in eine Richtung zum Noir. Kontraste werden erhöht und auch das Licht spielt eine gezieltere und dominante Rolle. Der Schatten des Charakters beschränkt sich nur mehr auf einen leichten Bereich seiner rechten Körperhälfte. Das Licht ist hier ein sehr kräftiges, welches die Form des Gesichtes sehr gut wiedergibt und nur auf den Charakter scheint. Der Übergang zwischen Licht und seiner normal belichteten Haut, ist eine Farbe, welche ungefähr der des Schattens entspricht. Outlines und dunkle Details, wie etwa sein Bart, werden von dem Licht nicht beeinflusst. Der Hintergrund ist diesmal stark erhellt, weist aber dennoch das gleiche Farbschema wie das von Bigby auf. Der Kontrast ergibt sich durch die Beleuchtung.

Im dritten Abschnitt der Abbildung steht Bigby vor einer Deckenlampe, welche hinter ihm auf den Boden leuchtet. Interessant ist hier zu beobachten, wie das Licht als weiche Belichtung und auslaufende Kegelform den Hintergrund aufhellt. Es werden auch schwarze Linien und dunkle Details beeinflusst. Das Hauptlicht für Bigby kommt hierbei jedoch von der anderen Seite. Er weist kein Rimlight oder andere Belichtungsmerkmale auf, welche auf das Licht hinter ihm schließen lassen würden. Er wirkt auf diesem Bild wie ausgeschnitten und integriert sich stilistisch gesehen nicht in den Hintergrund, welcher von der Deckenlampe beeinflusst wird.

Der letzte Ausschnitt zeigt Bigby im direkten Kontrast mit dem Hintergrund. Er selbst wird von einem rötlichen Licht beeinflusst und wirft auf seiner rechten Gesichtshälfte einen kalten Schatten. Der Übergang zwischen hellster und dunkelster Stelle trägt eine Stufe. Der Hintergrund ist dunkelblau und beinhaltet vereinzelte erhellte Fenster. Die Straßenlaterne wirft im Gegensatz zu der Beleuchtung im vorherigen Bild ein Licht, welches zwar auch weich verläuft, aber eine harte Grenze ihres Leuchtkegels aufweist. Die Umstellung auf diese harte Grenze, lässt das Licht viel mehr wie ein Teil der Comicszenerie wirken und bricht auch nicht so stark mit den harten Kanten des Hintergrundes. Interessant sind hierbei noch die schwarzen Detaillinien in Bigby's Gesicht, welche sich leicht verändert haben. Sie sind nicht so dominant, dünner und öfter durchbrochen.

Zu diesen Szenen von *The Wolf Among Us* muss zusätzlich noch erwähnt werden, dass alle Teil von Cutscenes, beziehungsweise nicht beeinflussbaren Stellen des Games sind. Dies beeinflusst die Bildausschnitte dahingehend, dass Lichter, Objekte und auch die Kameraausschnitte viel gezielter gewählt und platziert werden können. Der Shader in den Teilen des Games in denen Bigsby gesteuert werden kann, unterscheidet sich jedoch nicht von dem hier gezeigten. Auch die Schatten werden gleich aufgelöst.

4.4 *Valkyria Chronicles 4*

Das Computerspiel *Valkyria Chronicles 4* [35] wurde 2018 von *Sega* veröffentlicht.

In Abbildung 4.4 werden vier Einstellungen aus dem Game visualisiert. Die erste davon zeigt Protagonisten Claude Wallace in einer Cutscene. Der Charakter besitzt eine sehr weiche Schattierung. Aufgrund der gleich bleibenden, dunkleren Schattenfarbe und einem regelmäßigen Übergang, wird hier von einem Cel-Shading mit darunter liegender Rampmap ausgegangen. Die Texturen auf seiner Kleidung weisen einen Watercoloreffekt auf, bei dem verschwommene Unebenheiten und Flecken bereits in die Albedomap ein-



Abbildung 4.4: Verschiedene Spielausschnitte aus *Valkyria Chronicles 4* [35].

gearbeitet sind. Ebenso sind auch schwarze Linien, welche Details hervorheben, bereits in die Textur eingearbeitet. Der Charakter besitzt außerdem eine allgemeine Outline, welche um die Silhouette generiert wird. Zusätzlich zu dieser Outline wird bei genauem Hinsehen auf seine Nase und das Gesicht allgemein ein Colorbleeding ersichtlich. Hier ist die Farbe etwas außerhalb der umschließenden Outline.

Das zweite Bild zeigt einen Ausschnitt aus dem richtigen Gameplay. Interessant ist hierbei das Hatch-Shading, welches durch den geworfenen Schatten einer Wolke erscheint. Dieses Hatch-Shading ist nicht auf den Schatten gemapped und befindet sich auch nicht auf den Weltkoordinaten der Darstellung, sondern bewegt sich mit der Kamera mit. Es ist also ein auf das Projektionsfeld statisches Hatching, welches durch die Eigenschaften des Schattens stellenweise stärker oder schwächer sichtbar wird. *Valkyria Chronicles 4* konzentriert sich im Bezug auf das Visuelle mehr auf die Charaktere und Maschinen als auf den Hintergrund [54]. Dies ist eine gängige Herangehensweise in japanischen Spielen und wird auch in der Abbildung sichtbar. Jedes Hintergrundelement besitzt wie die Charaktere eine Outline. Besonders auffallend sind hier die Berge, welche weit im Hintergrund einen hohen Kontrast zur dunklen Strichfarbe aufweisen. Ebenso wird jeder Baum, jede einzelne Blume und sogar Erhöhungen des Bodens und der Wege mit dieser Outline versehen. Im direkten Gegensatz dazu stehen die Wolken, welche einen malerischen Stil aufweisen und keine Outlines besitzen. Die Stilverbindung aller Elemente ist hiermit nur der Wasserfarbeneffekt, der auch auf den Texturen der Umgebungsobjekte vorhanden ist. Das Game versucht sich somit, genauso wie der erste Teil der Serie, eher einem Gemälde mit Skizzenstil anzunähern [52].

Die dritte Einstellung zeigt das präsenteste Stilmittel von *Valkyria Chronicles 4*. Das gesamte Game ist von einer sehr dominanten weißen Vignette umrahmt, welche der ganzen Darstellung noch zusätzlich das Gefühl gibt, auf einer Leinwand gezeichnet zu sein. Auf die Frage hin, welcher der herausforderndste Aspekt war, diesen Artstil in ein Game zu integrieren, antwortete der Chief Artist Hitoshi Fukazawa, dass es das Zusammenspiel der Leinwand und der Linienstärke war [69]. Sobald die Linienstärke verschwand, konnten Konturen und Unterschiede der Formen verschwommen werden. So war er damit beschäftigt, sich Gedanken darüber zu machen, wie die Grenzen und Differenzierungen transportiert werden konnte und es dennoch als der Stil von *Valkyria Chronicles* wahrgenommen werden würde. Diese Linien und Linienstärken werden besonders im Zusammenspiel mit der Papiervignette sichtbar. Die Vignette überblendet



Abbildung 4.5: Verschiedene Spielausschnitte aus *Ni no Kuni II: Revenant Kingdom* [28].

bis auf die Silhouettenoutlines alles. Dies betrifft nicht nur Charaktere und Maschinen, sondern auch die Umgebung, inklusive der Berge im entfernten Hintergrund. Der weiche Übergang der Vignette und deren dominanten Linien tragen wesentlich zum Gefühl der Skizzenzeichnung auf einer Leinwand bei. Ermöglicht wird dies durch die darunterliegende, extra für dieses Game entwickelte Engine, welche den passenden Namen *CANVAS Engine* [70] trägt.

Die letzte Einstellung aus Abbildung 4.4 zeigt auf, wie eine starke Belichtung in *Valkyria Chronicles 4* verwendet wurde. Es macht den Anschein, dass hierbei eine eher realistischere Berechnung Anwendung gefunden hat, denn auch der Schatten an den Beinen weist nun einen größeren Dämpfungsbereich und feinere Abstufung auf. Das Licht ist nicht stilisiert und liefert einen großen Kontrast zu der Darstellungsart der Figur. Das Feuer im Hintergrund hingegen ist klar in sich aufgeteilt und abgegrenzt und wirkt wie 2D gezeichnet.

4.5 *Ni no Kuni II: Revenant Kingdom*

Das Computerspiel *Ni no Kuni II: Revenant Kingdom* [28] wurde 2018 für die Playstation 4 und Windows veröffentlicht.

In Abbildung 4.5 sind vier verschiedene Einstellungen aus dem Game ersichtlich. In dem ersten Bildausschnitt ist der Protagonist Evan Pettiwhisker Tildrum zu sehen. Er besitzt ein sehr klares Cel-Shading, welches nicht nur einen einstufigen Schatten, sondern außerdem einstufige Highlights aufweist. Diese sind besonders gut an seinen Haaren sichtbar, welche durch eine Normalmap ihre gezielte Form erhalten. Evan, so wie auch alle anderen Charaktere, ist mit einer einfach gehaltenen Outline und Edgedetection ausgestattet. Die Outline ist eine dünne schwarze Linie, welche in ihrer Linienstärke konstant bleibt und innerhalb eines Charakters keine unterschiedliche Struktur und Dicke aufweist. Er besitzt außerdem Detaillinien an Kleidung, Gesicht und Haaren. Diese lassen vermuten, dass sie über die Textur eingeschleust sind, doch werden sie in Echtzeit über Vertexinformationen berechnet [65]. Der Hintergrund ist im Gegensatz zu dem Charakter sehr malerisch texturiert und realistischer beleuchtet. Es werden handgezeichnete Texturen verwendet. Im Gegensatz zu dem ersten Teil der Serie, in dem das Licht noch in die Texturen eingearbeitet wurde, wurde in diesem gänzlich

darauf verzichtet und Schatten und Licht nur über die Engine berechnet. Das Licht trägt einen sehr wichtigen Teil dazu bei, das Game als eine 2D Animation aussehen zu lassen. Im ersten Teil war *Studio Ghibli* [75] maßgeblich an der Stilfindung involviert und diese Ästhetik sollte auch in diesem Game erneut erreicht werden [48, 64].

Das zweite Bild zeigt einen Dialog, bei dem der Spieler keinen Einfluss auf die Kamera hat. Besonders gut wird hier sichtbar, wie durch den fixen Blickwinkel der Hintergrund gesondert behandelt wird. Die malerischen Texturen sind hierbei die gleichen, doch wird mit einem Blur beziehungsweise einer Tiefenunschärfe die Umgebung verschwommen dargestellt. Dies differenziert den Charakter noch mehr von dem Hintergrund und hebt ihn stärker hervor. Interessant ist auch, wie die realistischeren Objekte mit den Charakteren harmonieren. Evan, der auf dem detailreicheren Stuhl sitzt, behält dank seiner kräftigen Farben und der Belichtung den Fokus. Es wird mit einer zusätzlichen Outline der Charakter von dem Stuhl abgegrenzt. Die schwarzen Linien werden nur dort gezogen, wo Evan auch auf die Grenzen des Objektes stößt und nicht innerhalb des Stuhles weitergeführt.

In Bild drei ist der inszenierte Auftritt des Gegners Brineskimmer zu sehen. Er ist eine Art Mischgestaltung aus den Hauptcharakteren und den Hintergrund, so vereint er Eigenschaften von beiden. Die Outlines und stellenweise arealen Schattierungen, lassen ihn definitiv als Charakter erscheinen. Die strukturierten Schatten, welche sogar eine Dämpfung statt einer klaren Linien aufweisen, die handgezeichnete Textur und gezielte Beleuchtung des Bauches geben dem Spieler jedoch die Ahnung, dass hier definitiv eine gezielte Differenzierung getroffen worden ist.

Der letzte Abschnitt aus Abbildung 4.5 zeigt den Gegner Black Knight. Er weist zu den typischen Eigenschaften in dieser Cutscene auch noch farbige Highlights einer Beleuchtung auf, welche seine Details und Struktur noch besser definiert. Dieser Einsatz von Farbe ist in *Ni no Kuni II: Revenant Kingdom* eher eine Ausnahme und wird nur in stimmungsvollen Cutscenes verwendet. Sobald sich das Game wieder im spielbaren Modus befindet, verschwindet dieses farbige Licht, beziehungsweise die Highlights des Charakters, und die herkömmliche Charakterdarstellung kommt wieder zum Tragen.

4.6 Zusammenfassung

Stilisierung von Games ist für Entwickler ein wichtiger Punkt in der Produktion. Hierbei geht es um die Aufmerksamkeit, welche von Außen auf das Game trifft und um das Game von anderen abzugrenzen oder sogar hervorzuheben [67]. Oftmals entscheiden sich Entwickler sogar dazu einen anderen Stil zu wählen als geplant, um dieses nicht als vermeintliche Kopie eines bereits existierenden, oder kürzliche veröffentlichten zu präsentieren. Auch kann es passieren, dass Folgetitel eines Games, welche ihre Darstellungsarten nicht anpassen oder weiterentwickeln, als nicht innovativ gelten und somit auf Kritik stoßen [5].

Es wurden verschiedene Genres gewählt, um auch zu zeigen, dass das Toon-Shading oder Cartoonstil nicht nur in Kinderspielen vorkommt, sondern sehr wohl ebenso an das erwachsene Publikum gerichtet ist. Diese Stile, wenn einmal festgelegt und in der Spielercommunity etabliert, werden bei Nachfolgetitel oft nur minimal verändert oder an die neue Hardware angepasst, um das Gefühl des gleichen Games zu vermitteln.

Outlines oder Detaillinien sind in den Games stark vertreten und auch ist die un-

terschiedliche Darstellung von Charakteren und dem Hintergrund nicht unüblich. *The Legend of Zelda: Breath of the Wild* geht sogar so weit, dass der Charakter im Toonlook dargestellt wird und der Hintergrund im direkten Gegensatz sogar realistische Elemente aufweist. Oft gibt es auch nur eine Hauptlichtquelle die den Charakter und dessen Schatten beeinflusst. Es werden aber dennoch stellenweise Aufhellungslichter oder Feuerstellen verwendet und berücksichtigt und somit die Bildstimmung beeinflusst. Das Toon-Shading hat nie keine Beeinflussung durch Licht, da sonst wichtige Tiefen- und Detailinformationen des Charakters verloren gehen. Dies fällt besonders gut bei *The Legend of Zelda: Breath of the Wild* auf, da auch bei Schlechtwetter und diffuser Sonnenbestrahlung der Shader eine klare Richtung aufweist. Rimlights werden, bis auf eine Ausnahme, eher selten im Gameplay selbst verwendet. Sie kommen mehr in den vordefinierten Cutscenes vor, in denen auch das Licht selbst viel gezielter gesetzt werden kann.

Kapitel 5

Gemeinsamkeiten und Unterschiede

In diesem Unterkapitel werden gezielt die Unterschiede und Gemeinsamkeiten hervor-gehoben und eventuelle Hintergründe für diese erläutert.

Auch wenn das Toon-Shading in Games und Animation das grundlegend gleiche Ziel, die Art der Darstellung zu stilisieren, individualisieren und abzugrenzen, verfolgt, so werden in Abbildung 5.1 doch klare Trends im Bezug auf Ähnlichkeiten und Unter-schiede sichtbar. In der Wahl der Projekte zur Analyse wurde versucht verschiedenste Games und Animationen in unterschiedlichen Genres und mit stilisierten Besonderhei-ten zu wählen. Es wurde Wert auf den Bekanntheitsgrad und Anerkennung der Artstils gelegt, um einen gemeinsamen Nenner zu schaffen und die Qualität des Endergebnisses trotz der Diversität zu gewährleisten.

5.1 Outlines

Der Einsatz von Outlines sind wie in Abbildung 5.1 zu sehen, eher den Games zuzu-ordnen. Diese Linien sind ein klares Element von den Comicbüchern, deren Stil manche Games so wie auch Animationen nacheifern. Aber sie sind nicht nur ein typisches Stilele-

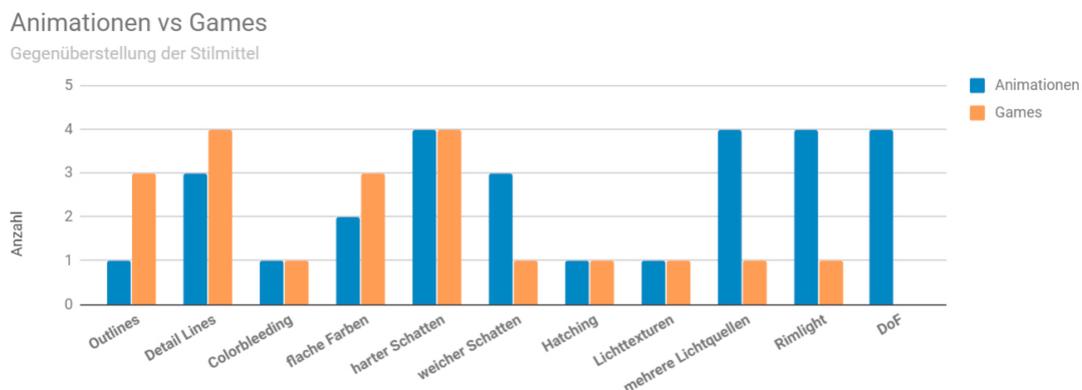


Abbildung 5.1: Eine Aufschlüsselung der verwendeten Eigenschaften in den Beispielen aus Kapitel 3 und Kapitel 4.



Abbildung 5.2: Gegenüberstellung verschiedener Anwendungen von Outlines in Animation und Game.

ment eines anderen Mediums, sondern auch eine sehr gute Möglichkeit den Charakter leichter zu definieren und vom Hintergrund abzugrenzen. So ist in Abbildung 5.2 ersichtlich, wie verschiedenst mit einer Outline gearbeitet wird. Das erste Bild zeigt die Animation *Love, Death & Robots Episode 15: Blind Spot*, in der die bewegten Charaktere und Objekte klar vom Hintergrund abgegrenzt werden. Hierzu trägt die Outline, welche die Charaktere umschließt und weitere Details hervorhebt, maßgeblich dazu bei. Gleiches gilt hier auch für das zweite Beispiel, das Game *Ni no Kuni II: Revenant Kingdom*. Auch hier ist der Hintergrund malerisch gehalten und die Charaktere werden klar durch eine Outline definiert. Eine andere Art Outlines zu verwenden, zeigt das dritte Bild aus dem Spiel *The Wolf Among Us*. Hier wird nicht nur der Charakter mit Outlines versehen, sondern alle 3D Objekte, auch jene, die im Hintergrund stehen und nicht die Aufmerksamkeit des Spielers auf sich ziehen wollen. Zusätzlich wird auch im Gegensatz zu den vorherigen Beispielen eine nicht konstante Strichstärke verwendet. Das letzte Bild zeigt ebenfalls ein Game, *Valkyria Chronicles 4*, welches die Outlines nur sehr subtil einsetzt. In den meisten Anwendungsfällen von Outlines, werden diese verwendet, um dem Charakter eine Grenze zu geben. Diese Grenze ist hier theoretisch auch gegeben, aber praktisch wurde hier bewusst ein Colorbleeding angewandt und die Linien sehr dünn gezeichnet. Bei genauerer Betrachtung wird ersichtlich, wie die Hautfarbe des Gesichts noch über die Outline hinausgeht.

5.2 Detaillinien

Detaillinien werden, wie in Abbildung 5.1 ersichtlich ist, beinahe gleichermaßen verwendet, sind aber in den betrachteten Projekten eher bei den Computerspielen zu finden. Es könnte vermutet werden, dass Detaillinien direkt mit der Verwendung von Outlines zusammenhängen, da sie doch nur eine Erweiterung der Silhouettenlinien auf Details innerhalb des Objektes sind. Doch Abbildung 5.3 zeigt, dass viel mehr Projekte Details mit schwarzen oder färbigen Linien hervorheben, als Outlines in diesen Games und Animationen zur Anwendung kommen. Outlines selbst sind auch kein Versprechen, dass Detaillinien verwendet werden, bei den hier gewählten Beispielen ist dies jedoch der Fall. Die Animationen *Love, Death & Robots Episode 15: Blind Spot*, *Spider-Man: Into the*



Abbildung 5.3: Gegenüberstellung verschiedener Anwendungen von Detaillinien in Animation und Game.

Spider-Verse und *Feast* verwenden diese Linien, um Stellen des Charakters abzugrenzen und Emotionen zu verdeutlichen. So wäre es bei letzteren Beispiel aufgrund der flachen Farbgebung oftmals nicht möglich die Pfoten des Hundes voneinander zu unterscheiden, wenn sie hintereinander zu sehen sind. In dem Game *The Legend of Zelda: Breath of the Wild* werden Detaillinien eher subtil eingesetzt. Den größten Anteil der Abgrenzung von zu ähnlich gefärbten Körperstellen übernimmt hier das Rimlight. Bei Objekten, wie zum Beispiel Link's Waffe, werden zusätzlich noch farbige Detaillinien verwendet, um dem Objekt mehr Tiefe zu geben. In *The Wolf Among Us* gehören die sehr dominanten Detaillinien zu dem Comicstil, welche die Entwickler anstrebten. Sie sind statisch in die Textur eingearbeitet und geben den Charakteren ein sehr comictypisches Aussehen. Das Game *Valkyria Chronicles 4* verwendet Detaillinien genauso subtil wie die Outlines. Sie werden angewandt, um der Textur mehr Informationen zu verleihen und sollen damit einen skizzierten Stil unterstützen. *Ni no Kuni II: Revenant Kingdom* benötigt diese Linien, um aus einer gleichfarbigen Fläche mehr Details hervorheben zu können. So würden ohne dieser Detaillinien die Falten in der Kleidung verloren gehen, oder aus seinen Haaren eine große einfarbige und bewegliche Fläche entstehen.

5.3 Flache Farben

Die Verwendung von Outlines und Detaillinien, um den Charakter leichter lesbar zu machen, hängt auch mit der Wahl der Farbgebung zusammen. In der Auswahl der Beispiele werden flache Farben fast gleichermaßen in Animationen und Games verwendet. Bis auf das Game *The Legend of Zelda: The Wind Waker*, welches stellenweise eine komplette flache Colorierung verwendet, besitzt jedes Projekt mit eben dieser Farbgebung Detaillinien, welche zur Abgrenzung helfen. In Abbildung 5.4 sind die Projekte mit einer flachen Colorierung sichtbar. Diese Art der Farbgebung ist in dieser Auswahl an Beispielen öfter in den Games zu finden als in den Animationen. Auffallend ist, dass keiner der Stile einem anderen gleich sieht, obwohl alle die gleiche Grundvoraussetzung haben. Am ähnlichsten sind sich noch die Animation *Love, Death & Robots Episode 15: Blind Spot* und das Game *Ni no Kuni II: Revenant Kingdom*. Diese Ähnlichkeit ist den Outlines und dem simplen Schatten verschuldet. Die anderen drei Beispiele verzichten auf

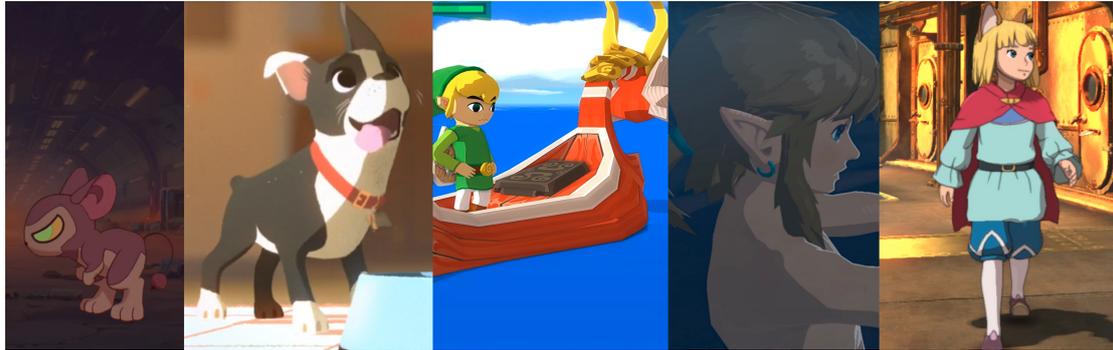


Abbildung 5.4: Gegenüberstellung verschiedener Anwendungen von flacher Colorierung in Animation und Game.



Abbildung 5.5: Gegenüberstellung verschiedener Anwendungen von harten Schattierungen in Animation und Game.

Outlines und verlassen sich auf Detaillinien, Schatten oder Rimlights um die Charaktere gut lesbar zu halten. Die restlichen Projekte besitzen handgezeichnete Texturen, welche entweder an Wasserfarben erinnern, eine Struktur aufweisen oder einfach als Schmutz zu identifizieren sind. *Spider-Man: Into the Spider-Verse* und *The Wolf Among Us* sind beides Projekte, welche trotz einer gezielt angewandten Textur, noch zusätzlich über Detaillinien verfügen.

5.4 Harter Schatten

In Abbildung 5.1 lässt sich ablesen, dass jeweils vier von fünf Projekten einen harten Schatten verwendet haben. Die Überschneidungen von den Animationen im Bezug auf weiche Schatten lassen sich auf die unterschiedlichen Anwendungen per Shot zurückführen. Es wird hier gezielt nur Schattierung des Charakters betrachtet und die der Umgebung außen vor gelassen. Allgemein wird ersichtlich, dass das Toon-Shading und auch der Cartoonlook eine starke Überschneidung mit der Anwendung von Cel-Shading beinhaltet und diese ebenso bewusst eingesetzt wird. In Abbildung 5.5 werden alle Arten der harten Schattierungen der Beispiele aufgezeigt. Manche davon verwenden stellen- oder shotweise eine unterschiedliche Art an Abstufungen der Schattierung, aber



Abbildung 5.6: Gegenüberstellung verschiedener Anwendungen von weichen Schattierungen in Animation und Game.

im Grunde haben sie alle etwas gemeinsam. Diese Schattierung bildet die Grundlage ihres Stils und wird zumeist sehr dominant zur Schau gestellt.

5.5 Weicher Schatten

In Abbildung 5.6 werden jene Projekte ersichtlich, welche eine weiche Schattierung vorweisen können. Auffallend ist hierbei, dass ebenso Projekte gezeigt werden, welche bereits einen harten Schatten in Verwendung haben. Dies kommt daher, dass Animationen ihre Darstellungsart per Shot oftmals anpassen, um gezielt auf einen Stimmungswechsel oder andere erwünschte Effekte eingehen zu können. In den hier analysierten Projekten werden bei *Sonder* und *Spider-Man: Into the Spider-Verse* die verschiedenen Arten der Schattierung verwendet. Bei *Sonder* sind dies oft Umgebungsschatten die auf den Charakter fallen oder das Cel-Shading erhält in einer idyllischen Szene einen weicherem Übergang. Bei *Spider-Man: Into the Spider-Verse* ist der gesamte Film mit Hilfe einer weichen Schattierung realisiert, bis auf einen Charakter, welcher gezielt die Toonoptik ansprechen soll. Bei den Animationen ist *Love, Death & Robots Episode 4: Suits* die einzige, welche nur eine weiche Schattierung einsetzt. Bei den Spielen ist dies *Valkyria Chronicles 4*. Diese zwei Projekte wollen einen malerischen Stil ansprechen und somit auf harte Abstufungen im Schatten verzichten. Die weiche Schattierung liegt hierbei stark auf der Seite der Animation.

5.6 Colorbleeding, Hatching und Lichttexturen

Colorbleeding, Hatching und Lichttexturen sind spezielle Stilisierungsmittel, welche in dem Spektrum der Analyse in Abbildung 5.1 als gleich verteilt bezeichnet werden. Alle drei dieser besonderen Darstellungsarten werden in dem Film *Spider-Man: Into the Spider-Verse* angewandt. Im erste Bild von Abbildung 5.7 macht sich das Hatching im Schatten und das Colorbleeding bei der Abgrenzung der Iris bemerkbar. Das zweite Bild zeigt ganz klar die Textur, welche vom Licht des Polizeiautos ausgeht. Im Falle dieser Animation lässt sich dies so erklären, dass die Produzenten eine moderne Version des typisch alten Comicstils anstrebten. So wurden druckverschuldete Eigenschaften,



Abbildung 5.7: Gegenüberstellung verschiedener Anwendungen von Colorbleeding, Hatching und Lichttexturen in Animation und Game.

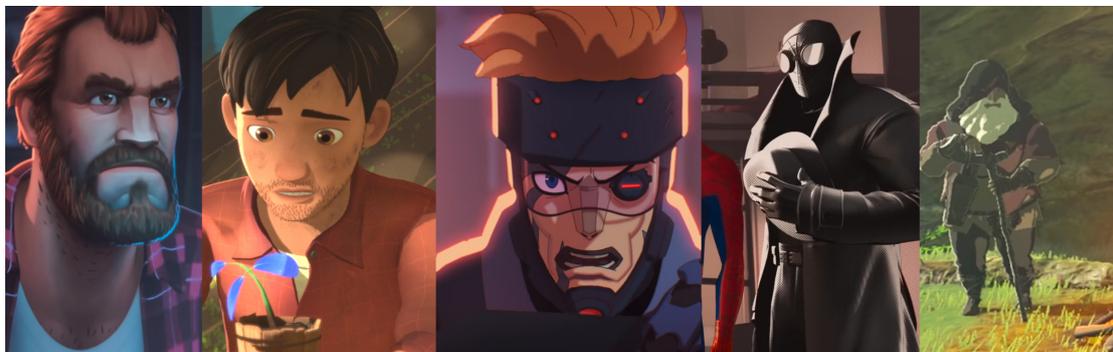


Abbildung 5.8: Gegenüberstellung verschiedener Anwendungen von mehreren Lichtquellen in Animation und Game.

wie etwa das Colorbleeding, und auch alternative Herangehensweisen für Halftoning und Schattierungen der typischen Comicdrucke in die Animation gezielt eingearbeitet. Auch *Valkyria Chronicles 4* versucht einen handgezeichneten Stil zu vermitteln und mit diesen Eigenschaften mehr in die Darstellung einer Skizze zu rücken. So ist ein feines, weitläufiges Hatching über den gesamten Bildschirm zu sehen, welche die Schatten der Wolken aufzeigen soll. Der letzte Ausschnitt der Abbildung 5.7 zeigt *The Legend of Zelda: Breath of the Wild*, welches die Texturen in den Lichthighlights nur sehr sparsam und nur an ausgewählten Stellen, wie etwa als Schraffierung an Waffen, Haaren oder einzelnen Arealen der Kleidung, verwendet. Dies macht sie zu einem sehr subtilen Stilelement.

5.7 Mehrere Lichtquellen

In Abbildung 5.1 wird ersichtlich, dass hauptsächlich Toon-Shading in Animationen von mehreren Lichtquellen beeinflusst wird. Die betreffenden Projekte sind in Abbildung 5.8 aufgereiht. Das einzige Game aus dem Analysepool, welches ebenso zum Beispiel ein nahes Lagerfeuer mit einbezieht und zu der allgemeinen Belichtung hinzurechnet, ist *The Legend of Zelda: Breath of the Wild*. Dennoch gibt es hier einen zusätzlich großen



Abbildung 5.9: Gegenüberstellung verschiedener Anwendungen von einem Rimlight in Animation und Game.

Unterschied. *The Legend of Zelda: Breath of the Wild* nimmt die Lichtquelle auch in Form des Cel-Shadings auf, das heißt es wird ein gleichfarbiges Areal aufgehellt. In den Animationen ist dies im Falle von *Love, Death & Robots Episode 4: Suits, Sonder* und *Spider-Man: Into the Spider-Verse* eine Beeinflussung mit realistischerem Licht. *Love, Death & Robots Episode 15: Blind Spot* verwendet für eine zusätzliche Aufhellung einen Bloom. Somit werden hier Toonelemente oder Ansätze von Toon-Shading mit realistischen oder realistischeren Gegebenheiten kombiniert.

5.8 Rimlight

In Abbildung 5.1 ist zu sehen, dass nicht nur die Anzahl der Lichtquellen stark auf der Seite der Animation liegt, sondern auch das Rimlight dort eine höhere Vertretung hat. Dies betrifft auch die exakt gleichen Animationen und das Game *The Legend of Zelda: Breath into the Wild*. Rimlight ist besonders wichtig, wenn keine Detaillinien verwendet werden, um die Form der Charaktere besser lesbar zu machen. In Abbildung 5.9 wird ersichtilch, wie in *Spider-Man: Into the Spider-Verse* und *Love, Death & Robots Episode 15: Blind Spot* beides verwendet wird. Die anderen Beispiele verzichten auf diese Kombination. In *Love, Death & Robots Episode 4: Suits* besitzt das Rimlight sogar eine besondere Stellung, da es maßgeblich zum Stil beiträgt. Es ist in fast jeder Einstellung vorhanden und wird beidseitig als komplementäre Beleuchtung eingesetzt. In den anderen Animationen findet diese Belichtung eine eher subtilere und nicht ganz so häufige Anwendung.

5.9 Depth of Field

Als letztes ist in Abbildung 5.1 die Tiefenunschärfe aufgelistet, welche in Abbildung 5.10 zu sehen ist. Die Tiefenunschärfe ist eigentlich keine Eigenschaft des Toon-Shadings per se, kann aber, wenn sie zur Anwendung kommt, die Stilfindung maßgeblich beeinflussen. Durch Tiefenunschärfe lassen sich so zum Beispiel Vordergrund und Hintergrund klar voneinander trennen und macht es somit leichter die wichtigen Charaktere lesbar zu machen. Dies findet eine starke Anwendung im Animationskurzfilm *Feast*. Eine besondere



Abbildung 5.10: Gegenüberstellung verschiedener Anwendungen einer Tiefenunschärfe in verschiedenen Animationsfilmen.

Art von Tiefenunschärfe wird von *Spider-Man: Into the Spider-Verse* verwendet. Sie wird hier nicht als typische Unschärfe dargestellt, sondern als chromatische Aberration angewandt.

5.10 Zusammenfassung

In diesem Kapitel wurden die jeweils fünf analysierten Games und Animationen verglichen. Diese Werke wurden so gewählt, dass eine große Bandbreite an Genres und hervorgehobenen Stilmitteln zur Verfügung steht. Die Gemeinsamkeiten dieser Beispiele sind speziell in den grundlegenden Stilisierung des Schattens und Lichtes vorhanden. So ist das comictypische Cel-Shading gleichermaßen in Animationen und Games zu sehen. Animationen verwenden jedoch weiche Schattierungen etwas häufiger. Auch gezielte Stilelemente wie etwa Hatching, Lichttexturen und Colorbleeding kommen in beiden selten, aber gleichermaßen vor. Auffällender sind hierbei die Unterschiede. So finden die typischen Outlines eher in Games ihren Platz. Auch Detaillinien werden knapp häufiger in den aufgezeigten Games verwendet, doch kommen sie auch in den Animationen vor. Es herrscht zudem eine scheinbar direkte Verbindung der Darstellungsart der Farben mit diesen Detaillinien. Wird eine flache Colorierung gewählt, so geht dies außerdem Hand in Hand mit den Detaillinien, welche den Charakter leichter lesbar machen. In den Animationen wird hier auch gerne ein Rimlight verwendet, welches in den Games, bis auf eine Ausnahme, nicht berücksichtigt wird. Dieses Rimlight trägt bei den Animationen stark zur Stimmung bei. Zusätzlich werden in den Animationen häufiger mehrere Lichtquellen verwendet, welche den Shader ebenso beeinflussen können. In Games wird hier hauptsächlich eine Lichtrichtung verwendet. Auch eine Tiefenunschärfe wird in Animationen oft eingesetzt. Sie hilft den Charakter zusätzlich vom Hintergrund zu differenzieren. In den gewählten Games wird diese überhaupt nicht angewandt. Hier muss die Unterscheidung mit bereits genannten Eigenschaften, kräftigeren Farben oder einer anderen Stilisierung gewährleistet werden.

Kapitel 6

Realisiertes Toon-Shading

Auf Basis der in Kapitel 5 erarbeiteten Gemeinsamkeiten und Unterschiede wird in der Game-Engine Unity 2019.2.1 ein Toon-Shading realisiert, welches die Ansprüche einer filmischen Darstellung erfüllen soll. Zusätzlich wird dieses noch in dem Programm Blender v2.80 [27] aufgebaut und die zwei Ergebnisse gegenübergestellt und verglichen. Auch wird auf die technischen Schwierigkeiten beider Programme eingegangen und welche Unterschiede diese auslösen.

6.1 Anforderungen

Abbildung 6.1 zeigt das verwendete 3D Modell. Dieser Charakter wurde im Zusammenspiel zwischen *Maya* [26] und *ZBrush* [34] erstellt und mit Hilfe der Software *Substance Designer* [37] texturiert. Es wurde gezielt darauf geachtet, den Charakter mit bewusst gesetzten Stilelementen, wie etwa dem geschwungenen und kreisförmigen Topologieverlauf und der mehrstufigen Rindentextur, zu versehen. Die Aufgabe des Shaders ist, diesen Stil zu unterstützen und nicht durch dominante Farbgebungen oder dergleichen zu verzerren. Um den malerischen Stil zu unterstreichen, wird hier auf die Outline verzichtet. Das Cel-Shading wird für das Hauptlicht eingearbeitet und verwendet. Diese harte Schatten liefern das Hauptmerkmal des Shaders und sollen das Toonelement unterstreichen. Da die geschwungenen Details bereits in Modell und Normalmap eingearbeitet sind, werden auch keine Detaillinien im Shader gesetzt. Die Rindentextur selbst besitzt ebenso bereits Linien, welche die Struktur des Holzes unterstreichen.

Der Shader muss aufgrund vieler verschiedener Lichter auch ein Zusammenspiel aus einem Hauptlicht und den Sekundärbeleuchtungen ermöglichen und diese sinnvoll und subtil einbinden können. Um keine zu starken Unterschiede zwischen beleuchteten und im Schatten liegenden Stellen zu gewährleisten, wird das Cel-Shading nur das Hauptlicht, also das gerichtete Licht der simulierten Sonne, einbeziehen. Auch wenn diese nicht sichtbar in der Szene verwendet wird, soll es die Richtung und die Stärke des Cel-Shadings bestimmen. Alle anderen Lichter werden als realistisches Licht gehandhabt, um einen sanften Verlauf der Farbgebung auf den Charakter zu projizieren. Dies ermöglicht eine kontrollierbare Stilisierung, welche in jeder Szene den gleichen Ausgangspunkt besitzt und weiche Übergänge in der Bewegung zwischen Licht und Schatten ermöglicht. Dies macht es nicht nur leichter, die unterschiedlichen Szenen besser einander anpassen



Abbildung 6.1: 3D Modell, welches zum Testen des Shaders verwendet wird. Ohne und mit Textur in Unity 2019.2.1 gerendert.

zu können, sondern setzt dem Charakter mit seinem Shader auch einer realistischen Belichtung aus, welches ihn aufgrund der Ergebnisse in Kapitel 5 mehr an eine filmische Darstellung anlehnt.

Spezielle Stilelemente wie Hatching, Colorbleeding oder Lichttexturen sollen nicht angewandt werden, da ein malerischer Stil angestrebt wird und diese die Ästhetik mehr in Richtung Comic oder Skizze rücken.

Der Shader soll in stark belichteten, eher dunklen Szenen und auch in Abschnitten, welche eine Kombination aus Schatten und mehreren Lichtquellen besitzen, funktionieren und den Charakter lesbar machen. Lesbar heißt in diesem Fall, dass er nicht nur seine eigenen Überschneidungen voneinander abgrenzen soll, sondern auch sich selbst als Charaktershader von den Umgebungsshadern der Bäume im Hinter- oder Vordergrund hervorhebt. Dazu soll ein Rimlight direkt in den Shader eingearbeitet werden. Dieses soll eine subtile Grenze darstellen, welche den Charakter weiter definiert. Die Umgebungsbäume werden diese Rimlight nicht erhalten und mit Hilfe von unterschiedlicher Farbgebung, einem gezielt eingesetztem Depth of Field in Kombination mit Nebel und gezielter Ausleuchtung von Hintergrund und Vordergrund, abgegrenzt werden.

6.2 Umsetzung

Die Umsetzung des Shaders erfolgt in der Realtime-Engine Unity in der Version 2019.2.1. Um auch einen Vergleich zu einer nicht Echtzeitumgebung aufzuzeigen, wird der Shader

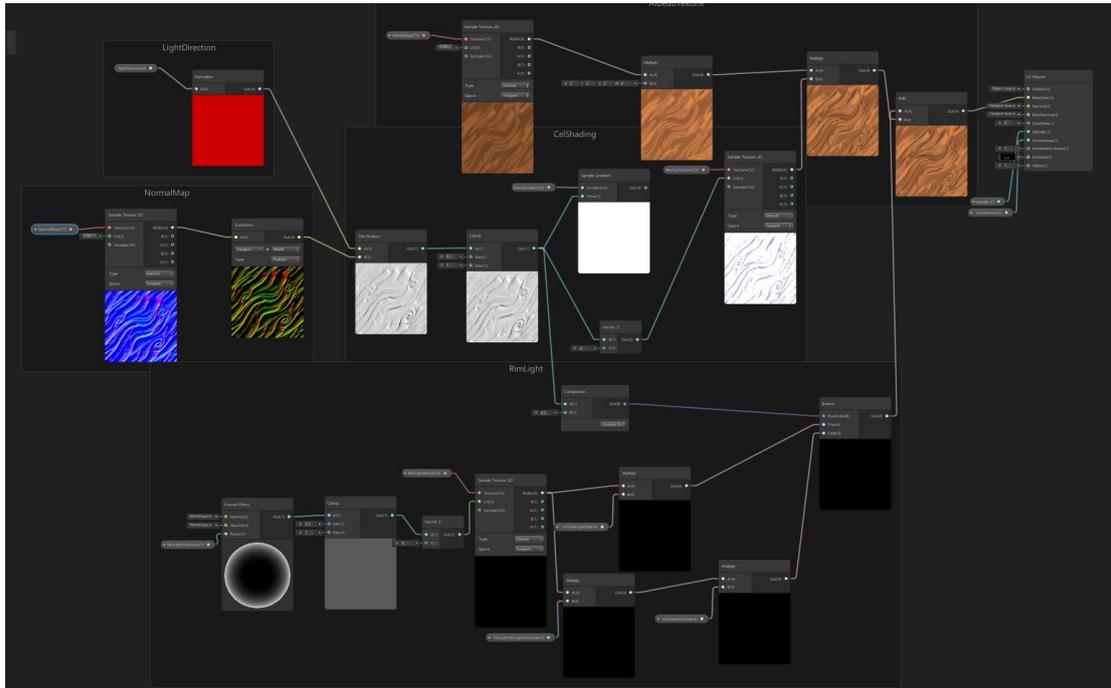


Abbildung 6.2: Screenshots des erstellten Shaders in Unity 2019.2.1 Shadergraph.

ähnlich in einer herkömmlichen Render-Engine aufgebaut. Da der Shadereditor dem von Unity ebenfalls ähnlich ist, ist die Wahl hierbei auf Blender in der Version 2.80 gefallen. Kleine Änderungen der Nodes aber auch eine umständliche Anpassung an fehlenden Einstellungen lassen den Shader nicht direkt ident auf die verschiedenen Programme übersetzen. Auf diese Änderungen wird im folgenden Abschnitt 6.2.1 und Abschnitt 6.2.2 eingegangen. Als Erstes wurde der Shader in Unity entwickelt und optimiert und danach in Blender möglichst getreu nachempfunden.

6.2.1 Unity

In Abbildung 6.2 wird ersichtlich, wie das Shadernetzwerk in Unity aufgebaut ist. Organisatorisch ermöglicht Unity es auch, einzelne Nodes in Gruppierungen einzuteilen und diese zu benennen. Zusätzlich können gewisse Nodes wie etwa Zahlen, Vektoren, Texturen oder Farben auch so angelegt werden, dass diese nicht im Shadergraph selber einzustellen sind, sondern sie der Nutzer auch außerhalb im eigentlichen Unity-Fenster im jeweiligen Material bearbeiten kann. Somit muss nicht jederzeit der Shadergraph geöffnet werden, wenn eine Änderung vorgenommen werden muss. In diesem Fall betrifft das Eigenschaften wie etwa die Normalmap, Texturen, Stärken des Rimlights und Farben des Rimlights.

Links oben ist die Gruppierung *LightDirection* zu sehen. Hier wird über ein externes, selbst geschriebenes Script, die Richtung des Hauptlichts, also das Directional Light, eingeschleust. Darunter befindet sich die Einbindung der Normalmap. Diese wurde zuvor aus Substancepainter generiert und hier über eine *Texture* Node eingeschleust. Diese

Texture Node wird auf *Type: Normal* eingestellt, um dem System sagen zu können, dass es sich hierbei um eine Normalmap handelt und somit die verschiedenen Farbchannels richtig interpretiert werden. Diese Normalmap wird in eine *Transform* Node geschleust, welche ihre Information aus dem Tangent-Space in den World-Space umwandelt, um auch auf äußere Umstände wie etwa das Licht reagieren zu können.

Licht und Normalmap werden nun mit Hilfe eines Skalarprodukts, also der *Dot Product* Node, auf einen einzelnen Wert zusammengefasst. Dieser Wert beschreibt nun Areale, die beleuchtet oder schattiert werden. Da diese Werte ein fehlerhaftes Resultat mit der Zahl 0 liefern, wird hiernach noch eine *Clamp* Node verwendet, um alles was unter 0,1 ist, als 0,1 zu werten. Danach kommt das Herzstück eines jeden Cel-Shadings. Es wird ein Gradient eingeschleust und mit Hilfe der generierten Werte die Zielfarbe ausgewählt. In diesem Falle wurde der Gradient zuvor als Textur abgespeichert und auch als Textur in den Shader eingeschleust. Über die UV Werte wird somit die Position des Wertes in der Textur abgebildet. Dafür muss der singuläre Wert noch in die X-Position eines neuen 2D Vektors gespeichert werden. Somit verschiebt sich, je nach Größe des Wertes, die Position des Zeigers immer weiter nach rechts. So ist die linke Seite der Textur die Beschreibung des Schattens und die rechte Seite die Beschreibung der erhellten Stellen. Es können so viele Abstufungen wie erwünscht in die Textur gearbeitet werden. Auch sind somit harte oder weiche Übergänge steuerbar. Die Textur, welche hier verwendet wurde, ist genau in der Mitte geteilt. Auf der linken Seite befindet sich ein dunkleres Graublau und die rechte Seite wurde bei einem reinweiß belassen. Um dies in der Echtzeitumgebung besser steuerbar zu machen, würde sich die *Gradient* Node sehr gut eignen. Aufgrund eines systeminternen Fehlers kann diese jedoch nicht ohne Weiteres auf einem deutsch eingestellten Betriebssystem verwendet werden und so musste ein anderer Lösungsweg gefunden werden [39]. Dieser generierte Schatten wird nun mit der Albedomap des Charakters multipliziert. Die Textur wurde zuvor noch etwas aufgehellt, um den Charakter mehr hervorzuheben.

Im unteren Teil von Abbildung 6.2 ist die Erstellung des Rimlights zu sehen. Basis dafür bildet die sogenannte *Fresnel* Node. Im Previewbild ist sehr gut ersichtlich, wie diese funktioniert. Es findet an den Kanten entlang des Objektes eine Aufhellung statt, welche gegen den Mittelpunkt hin wieder abnimmt. Die Stärke und Ausbreitung dieser Aufhellung kann durch einen Parameter gesteuert werden, welcher hier auch für den Nutzer freigelegt wurde. Diese wird nun aus dem bereits genannten Grund in eine *Clamp* Node geschleust, um ein fehlerhaftes Ergebnis vorzubeugen. Da die *Fresnel* Node einen natürlichen Verlauf besitzt wird auch hier erneut eine Rampmap verwendet, um einen harten Übergang zu schaffen. Danach wird zwischen zwei Rimlights unterschieden: Dem Rimlight, welches auf den beleuchteten Stellen liegt und jenem, welches auf den schattierten Arealen zur Anwendung kommt. Dafür wird das Resultat in zwei *Multiply* Nodes geschleust, welche jeweils eine eigene Farbe hinzurechnet. Zusätzlich wird für das Rimlight an den dunklen Stellen noch ein zweites *Multiply* angehängt, um somit auch noch die Stärke, also Helligkeit und somit Präsenz dieses Rimlights, einstellen zu können. Beide werden dann in eine *Branch* Node gesteuert, welche entscheidet, welches dieser beiden Rimlights zum Einsatz kommt. Die Logik hinter dieser Entscheidung kommt von der *Comparison* Node, welche das Resultat der *Clamp* Node von Normalmap und Lichtrichtung betrachtet und beurteilt. In diesem Fall wird geprüft, ob der eingehende Wert größer als 0,5 ist. Wenn ja befindet sich dieser Pixel im Licht und das



Abbildung 6.3: Rendering des finalen Shaders in Unity 2019.2.1.

erste Rimlight wird verwendet. Wenn nicht, dann das andere.

Das ausgewählte Rimlight wird nun mit Hilfe einer *Add* Node auf die Textur mit dem bereits eingearbeiteten Cel-Shading gelegt und dieses Ergebnis als fertige Textur dem engineeigenem Shader übergeben.

Normalerweise ist es üblich bei einem Toon-Shading einen *Unlit* Shader zu verwenden, da dieser jegliche Lichter in der Szene ignoriert. Da hier jedoch eine zusätzliche realistische Beleuchtung angestrebt wird, fiel die Entscheidung hierbei auf den *Lit* Shader.

In Abbildung 6.3 ist der finale Shader am Charaktere zu sehen. Als Rimlights wurden Komplementärfarben eingestellt, um den filmischen Charakter noch zu unterstreichen.

6.2.2 Blender

Blender löst das Echtzeitarbeiten an einem Shader etwas eleganter als Unity. Hier kann der Shader Editor ebenfalls parallel zur Scene View oder einer Object View geöffnet sein. Größter Unterschied ist jedoch, dass der Shader bei jeder Änderung nicht immer gespeichert werden muss und ein paar Sekunden neu kompiliert. In Blender funktioniert dies im Gegensatz zu Unity in Echtzeit und es gibt für das Testen in diesem Fall keinen Grund, Variablen und Eigenschaften außerhalb des Shader Editors verändern zu wollen.

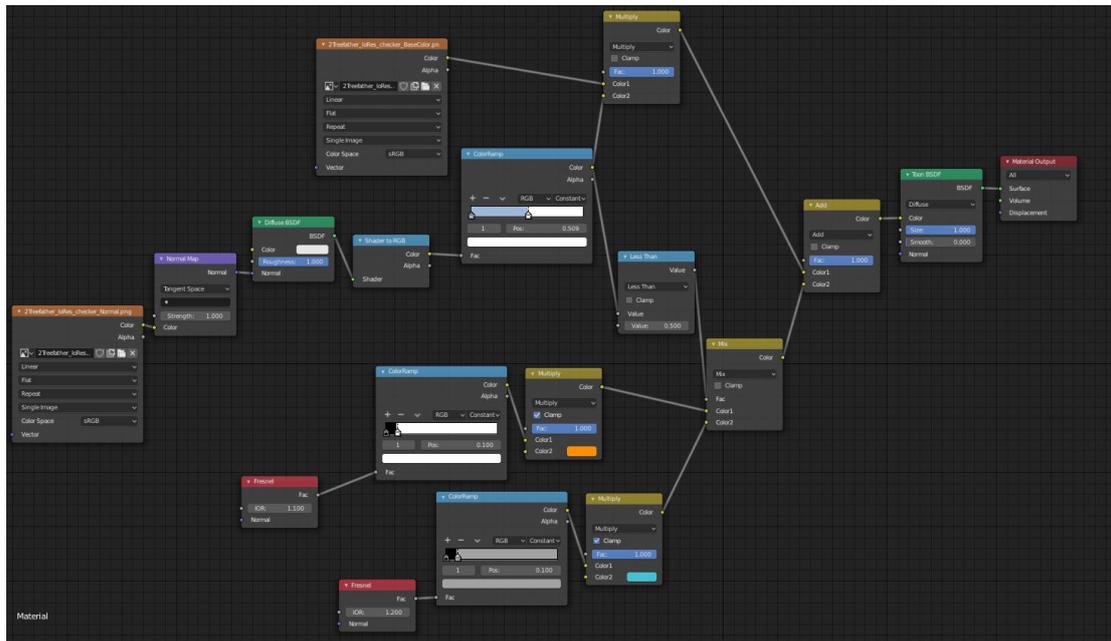


Abbildung 6.4: Screenshots des erstellten Shaders in Blender 2.80 Shader Editor.

Da Blender auch nur dazu verwendet wurde, den Shader nachzubauen, wurde auf dies gänzlich verzichtet und alle Einstellungen direkt im Editor getätigt.

Ein großer Unterschied gegenüber Unity ist hier die Basis des Cel-Shadings. In der Unityversion wird hier ein einzelnes Licht mit einbezogen, in Blender selber ist es mit den bereits existierenden Nodes nicht möglich im Shader angeben zu können, dass nur ein Licht verwendet werden soll. Hierfür gibt es stattdessen die *Diffuse BDF* Node. Diese Node ist eigentlich eine Standard Shader Node, welche eine Lambert Diffuse Berechnung anwendet. Diese bildet somit unsere Schattierungs- und Lichtareale ab, genauso wie es die *Dot Product* Node zwischen Lichtrichtung und Normalmap in Unity liefert. Da hier auch die Normalmap miteinbezogen werden soll, wird diese als Textur in eine *Texture* Node importiert, mit Hilfe einer *Normal* Node in eine Normalmap umgewandelt und dann in das Normalattribut der Shader Node angehängt.

Um mit dem daraus resultierenden Shaderergebnis weiterarbeiten zu können, ist es notwendig, die Shaderinformation in ein weniger komplexes Format zu verwandeln. Dies geschieht mit der *Shader to RGB* Node. Diese Farbinformation bildet somit die Grundlage für das Cel-Shading, welches über eine Ramp gesteuert wird. Anders als in Unity ist die interne Verwendung eines Gradients über eine *Gradient* Node auch für deutschsprachige Systeme möglich und kommt hier somit auch zur Anwendung. Hier wird direkt im Shader Editor der Verlauf eingestellt und über den Eingang *Frac*, als Index gesteuert. Das Cel-Shading ist somit abgeschlossen.

Um die Textur einfließen zu lassen, wird diese über eine *Texture* Node importiert und mit Hilfe einer *Multiply* Node eingebunden.

Das Rimlight funktioniert aus logischer Sicht genauso wie das in Unity. Der Ausgang der *Ramp* Node wird mit Hilfe einer *Less Than* Node geprüft, ob der aktuelle Wert im



Abbildung 6.5: Rendering des finalen Shaders in Blender 2.80.

Schatten oder im Licht liegt. Dementsprechend wird dann das jeweilige Rimlight über eine *Mix* Node ausgewählt. Hierbei muss jedoch beachtet werden, dass zwei verschiedene *Fresnel* Nodes zur Auswahl kommen, da das Rimlight im Schatten dünner ist, als jenes, welches beleuchtet wird. Diese Nodes fließen dann in ihre jeweils eigenen *Ramp* Nodes, bei denen bereits die Helligkeit angepasst wird. Dies ist notwendig, damit das Rimlight im Schatten nicht zu unrealistisch wirkt und dem Charakter eine helle Outline verpasst wird. Im Unity Shader wurde diese Problematik über eine extra Einstellungsmöglichkeit der Helligkeit für den Nutzer gelöst, da die Rampmaps dort als Textur eingeschleust werden müssen und somit keine leichte und schnelle Anpassung der Helligkeit möglich ist. Danach werden die jeweiligen harten Lichter mit einer Farbe in einer *Multiply* Node zusammengemischt, um auch hier der artistischen Gestaltung keine Grenzen zu setzen.

Das Ergebnis des Rimlights wird nach der Kombination aus Textur und Cel-Shading über eine *Add* Node hinzugefügt, in eine *Toon BSDF* Shader Node geschleust und dann mit Hilfe einer *Material Output* Node als finales Ergebnis definiert.

In Abbildung 6.5 ist das Resultat und der gerenderte Charakter in dem Programm Blender zu sehen.



Abbildung 6.6: Vergleich der Renderings von Unity (jeweils links) und Blender (jeweils rechts).

6.3 Vergleich

In Abbildung 6.6 ist der direkte Vergleich zwischen den erstellten Shadern ersichtlich. Zu aller erst ist die Standardbelichtung der beiden Programme sichtbar. In Unity wurde mit Hilfe eines Skydomes eine allgemeine Erhellung hinzugefügt. In Blender selber war dies im Bezug auf den Shader nicht wirklich möglich, da mit der zusätzlichen Aufhellungen einige Details des Cel-Shadings verloren gingen. Hier müsste im Shader gegen diese Aufhellung gearbeitet und die schattierten Stellen zusätzlich verdunkelt werden.

Beide Programme verwenden die OpenGL Orientierung der Normalmap, somit musste hier keine weitere Anpassung angewendet werden.

Bezüglich des Cel-Shadings, ist dieses auch im Shader von Unity viel stärker zu sehen. In Blender ist der harte Schatten schon fast subtil und fällt nur direkt ins Auge, wenn ein direkter Vergleich zwischen herkömmlichen Shading und Cel-Shading stattfindet. Dies rührt auch hauptsächlich von der Belichtung, da in der Unity Version der weiche Schatten, den das Objekt selbst auf sich wirft durch den Skydome zusätzlich abgeschwächt wird. In der Blenderversion wird hier nur mit Hilfe von weiteren Lichtsprüngen des directional Lights gearbeitet um eine Aufhellung zu ermöglichen. Dadurch wird der weiche Schatten nicht so sanft aufgehellt und überlappt quasi den harten Schatten, welcher somit keinen starken Kontrast mehr erzeugen kann.

Die Rampmap für das Cel-Shading funktioniert in beiden Programmen gleichermaßen gut, obwohl hierbei die Anwendung in Blender in den verwendeten Versionen um einiges komfortabler vonstatten ging.

Für die Lichtqualität selbst bedeutet der Vergleich, dass das Licht in der offline Renderumgebung viel besser in das Objekt eingearbeitet wird. So werden Bereiche, welche dem Licht ausgesetzt sind, auch viel genauer aufgehellt und auch die Grenze zwischen Licht und Schatten ist hierbei um einiges genauer und nicht so wackelig wie in Unity. Dies wird gezielt an dem Kringel auf seinem Brustkorb ersichtlich.

Unterschiede im Rimlight können gut an seinem Kopf erkannt werden. So war es von Anfang an schwer, die Stärke der Linien gleich einzustellen, da in Unity selbst der

Fresneffekt in Kombination mit einem Echtzeit Anti-Aliasing nicht konstant ist. So verschwinden dünne Linien im Prozess der Kantenglättung fast vollkommen, aber auch dickere Areale, wie etwa an seiner – von uns aus gesehenen – rechten Kieferseite, werden nicht als Fläche gezeichnet, sondern können Unebenheiten aufweisen. In Blender ist das Rimlight konstant zu sehen. Dünne Linien verschwinden nicht und es können somit viel feinere Details hervorgehoben werden. So ist der hintere Teil seiner Schulterplatte in Blender gut abgegrenzt, in Unity jedoch komplett mit einem Rimlight versehen, obwohl andere Stellen, wie etwa die Achselhöhle darunter, vermuten lassen würden, dass das Rimlight in Unity dünner eingestellt ist als in Blender.

Auch die Farbe des Rimlights weist einen dominanten Unterschied vor. Interessanterweise ist der gleiche RGB Wert beider Engines nicht die gleiche Farbe. Hierzu musste die hexadezimale Farbinformation verwendet werden, um das gleiche Ergebnis zu erzielen. Des Weiteren wird auch beobachtet, dass die Farben im Rimlight in Blender um einiges intensiver dargestellt werden als in Unity. Dies könnte durch eine weitere Einstellungsmöglichkeit, welche die Helligkeit oder die Transparenz der Farbe regelt, angepasst werden. In Unity wirkt das Rimlight viel subtiler und somit auch natürlicher.

6.4 Zusammenfassung

Zusammengefasst heißt das, dass beide Shader in den jeweiligen Programmen gut funktionieren und schnell stilisierte Ergebnisse liefern. Während der Erstellung der Shader wurde klar, dass beide Programme ihre Stärken und Schwächen haben. So ist es möglich, in Unity gezielt einzustellen, welche Lichter überhaupt zur Anwendung kommen sollen. Jedoch bewies sich die Einbindung der Rampmap in der Version 2019.1.2 und einem Betriebssystem mit Deutsch als eingestellter Sprache als Schwierigkeit, da hierbei über eine eingeschleuste Textur gearbeitet werden musste. Blender hingegen konnte dies mit einer simplen *Ramp* Node lösen, welche sich in Echtzeit einstellen lässt. Jedoch konnte mit den standardmäßig existierenden Nodes nicht ausgewählt werden, welches Licht für die Schattierung verwendet werden sollte. Interessanterweise zeigt somit auch Blender schneller sichtbare Ergebnisse, da die Erstellung und Änderungen des Shaders in Echtzeit in einem zweiten Fenster beobachtet werden kann. In Unity selbst muss der Shader bei jeder Änderung gespeichert und neu kompiliert werden, bevor sich das Modell in der Szene erst anpasst. Um dies zu beschleunigen, werden anpassbare Variablen als Parameter angelegt, um diese auch außerhalb des Shadergraphs ändern zu können.

Die Ergebnisse zeigen auch den Vorteil des Offlinerenderings. Lichtspezifische Eigenschaften werden in Blender viel genauer in das Objekt eingearbeitet und auch die Kantenglättung lässt hier ein besseres Ergebnis beobachten. Das Echtzeit-Antialiasing in Unity lässt subtil, aber bei genauer Beobachtung sichtbar dünne Linien des Rimlights verschwinden und auch größere Areale des Rimlights enthalten leichte Löcher. Auch die Abgrenzung zwischen beleuchteten Stellen und Schattierungen sind in Unity stellenweise als leicht wellige Linien zu sehen. In Blender sind dies sehr saubere und geradlinige Abgrenzungen. Für den allgemeinen Stil des Shaders liefern beide Programme ein ähnliches Ergebnis. Hier kommt es dann auf die noch zusätzliche Belichtung an, um ihn gut in Szene zu setzen. Im Allgemeinen bedeutet die Gegenüberstellung, dass es möglich ist einen stilisierten Echtzeitshader in Unity zu erstellen, welcher die gleichen Anforderungen einer offline Umgebung erfüllt. Jedoch müssen kleine Abzüge bezüglich

Details und der verarbeiteten Lichtqualität in Kauf genommen werden.

Kapitel 7

Zusammenfassung

Die behandelte Forschungsfrage dieser Masterarbeit – wie unterscheidet sich Toon-Shading in der Animation und Games Branche und welche Eigenschaften muss ein Toon-Shading in einer Game-Engine aufweisen, um eine filmische Ästhetik erreichen zu können – wird mittels der Analyse der gewählten Beispiele beantwortet. Toon-Shading in Animationen und Games unterscheidet sich primär über die Verwendung von lichtspezifischen und simplifizierenden Eigenschaften. So werden Outlines und flache Colorierungen verstärkt in Games und weiche Schatten, Einbeziehungen mehrerer Lichtquellen und Rimlight gehäuft in Animationen verwendet. Der charakteristische, harte Schatten und auch stilistische Details wie Hatching, Colorbleeding und Lichttexturen sind in beiden gleichermaßen zu finden. Somit erschließt sich daraus die Erkenntnis, dass ein Toon-Shading einen harten Eigenschatten, zusätzlichen weichen Umgebungsschatten, Einbezug mehrerer Lichtquellen und ein Rimlight benötigt, um den Stil einer Animation zu entsprechen.

Toon-Shading gehört zu dem Überbegriff der Non-photorealistic Renderings. Es ist nicht nur als Stilisierung für Cartooncharaktere beliebt, sondern teilt ihre Eigenschaften mit vielen anderen Darstellungsarten der NPRs. Der direkte Ursprung dieses Stils ist der Comic, welcher 1935 erstmalig auch unter diesem Namen veröffentlicht wurde. Die Art der Zeichnungen oder Stilisierung dieser Kunst, lässt sich aber bis zu den Höhlenmalereien zurückverfolgen. Daraus lässt sich schließen, dass das Toon-Shading eigentlich seinen Ursprung aus der Simplifizierung von Zeichnungen des Menschen hat, was schon seit den ersten dokumentierten Malereien der Fall ist.

Das Toon-Shading ist heutzutage in der Animations- wie auch in der Gameindustrie stark vertreten. Es weist schnell ersichtliche Eigenschaften auf, welche jedoch in der Anwendung stark voneinander abweichen können und somit ein großes Spektrum an stilistischen Entscheidungen und grafischen Möglichkeiten zur Verfügung stellen. Das Toon-Shading ist nicht mehr nur ein Mittel zur performanten und simplen Darstellung, sondern auch eine Möglichkeit Games artistisch und komplex zu visualisieren. Diese verschiedenen Darstellungsarten werden anhand von individuellen Beispielen aus unterschiedlichen Genres aufgezeigt. Es werden die dominanten, subtilen oder nicht vorhandenen Merkmale der Animation oder des Games untersucht und hervorgehoben und somit eine Basis der Gegenüberstellung geschaffen. Diese Gegenüberstellung der Eigenschaften bildet die Grundlage, um die Differenzierung des Toon-Shadings in Games und Animationen in Worte zu fassen und festlegen zu können. Auffallend ist hierbei, dass

sich die Grundlagen eines Toon-Shadings, wie die flache Colorierung, der harte Schatten und die Anwendung von Licht- und Schattentexturen und dem Colorbleeding, fast gleich verteilt in beiden Genres wiederfindet. Dennoch gibt es zwischen Games und Animationen auch deutliche Unterschiede. So sind die Outlines stark den Games zuzuordnen und die weichen Schattierungen, Verwendung von mehreren Lichtquellen und einem Rimlight, vermehrt in den Animationen zu finden. Einzig und allein die Tiefenunschärfe ist nur in den Animationen vertreten. Jedoch muss hierbei gesagt werden, dass dies keine Eigenschaft des Toon-Shadings selbst ist, sondern nur ein Hilfsmittel, um die Objekte von deren Hintergrund oder Vordergrund leichter abgrenzen zu können. Diese herausgearbeiteten Merkmale liefern die Grundlage für das realisierte Toon-Shading.

Die Anforderungen des Toon-Shadings sind eine Kombination aus Merkmalen, welche dominant in den Animationsbeispielen vertreten sind und jenen, welche in Animation und Game gleichermaßen zur Anwendung kommen. Somit ist das Cel-Shading, also der harte Schatten, die Grundlage für das erstellte Toon-Shading. Zusätzlich werden aber auch weiche Lichter und Schatten der Umgebung berücksichtigt und ein steuerbares Rimlight grenzt den Charakter weiter ab. Die Outline wird komplett verworfen und eine malerische Textur aus Substance Designer liefert die Basis für die Farbgebung. Die Umsetzung erfolgte zuerst in Unity. Hierbei können alle Anforderungen eingearbeitet werden und es wird deutlich, dass die Realisation dieses filmischen Toon-Shadings in Unity's Shadergraph möglich ist. Um dies noch weiter zu unterstreichen, wurde dieser Shader so exakt wie möglich zusätzlich in einer herkömmlichen Render-Engine nachgebaut. Die Wahl fiel aufgrund des ähnlichen Shader Editors auf Blender. Auch hier lieferte die Realisierung, bis auf wenige Abweichungen, ein ähnliches Ergebnis. Die Resultate der beiden Renderings wurden schlussendlich verglichen und trotz des fast identen Aufbaus, wurden wenige Unterschiede offenbart. So zeigt Blender einen besseren Umgang mit Licht, welche die Aufhellung und Lichtreflektionen in einer besseren Qualität einarbeitet. Die Abgrenzung zwischen Licht und Schatten weist in Unity stellenweise wackelige Linien auf, welche in Blender jedoch gerade dargestellt werden. Auch ist die Kantenglättung in dem offline Rendering zuverlässiger als in Unity, denn in letzterem werden dünne Linien des Rimlights verschluckt und auch flächigere Stellen weisen unregelmäßige Einfärbung auf. In Unity kann jedoch das Cel-Shading leichter auf nur ein Hauptlicht angepasst werden. Zusätzlich muss gesagt werden, dass Blender den Umgang mit den Shader Editor besser implementiert hat, da hierbei schneller Ergebnisse sichtbar werden als in Unity.

Zusammengefasst heißt das, dass es möglich ist ein filmisches Toon-Shading in einer Game-Engine zu realisieren. Das fast identisch aufgebaute offline und realtime Rendering weist zwar Unterschiede auf, aber diese sind nur bei genauerer Betrachtung und Gegenüberstellung sichtbar und somit keine Probleme für die kohärente Ästhetik der Animation. Wichtig hierfür ist eine starke Einbindung von weichem und realistischem Licht, die Berücksichtigung von mehreren Lichtquellen und ein Rimlight, welches bei Bedarf auch mit Komplementärfarben in Licht- und Schattenbereichen verwendet werden kann. Um die Essenz des Toon-Shadings ebenso einzufangen, sollte nicht auf das typische Cel-Shading verzichtet werden.

Da sich offline und realtime Rendering aufgrund technischer Fortschritte immer mehr aneinander annähern, wird es in Zukunft immer leichter fallen, auch eine filmische Ästhetik in Game-Engines verwenden zu können. So wird Unity bereits für wenige Anima-

tionsproduktionen verwendet, besitzt jedoch den Vorteil, dass auch hier nicht realtime, sondern mit wenigen Sekunden Berechnungszeit, gerendert werden kann. Dies ermöglicht nicht nur eine flexiblere Produktion, sondern verkürzt bei richtiger Anwendung auch die Produktionszeit erheblich. Auch Blender hat ihren ersten Realtimerenderer Eevee veröffentlicht und setzt somit andere Renderprogramme unter Druck [53, 74].

Anhang A

Inhalt der CD-ROM/DVD

Pfad: /

img	Gesamte, verwendete Bildmaterial.
articles	Referenzierten Onlineartikel und Videos.
shader/unity	In Unity erstellen Shader, inklusive aller benötigten Komponenten.
shader/blender	In Blender erstellten Shader, inklusive aller benötigten Komponenten.
thesis	Masterarbeit in Form einer .pdf und des Latex Projektes.

Quellenverzeichnis

Literatur

- [1] George Carlin und Hal Foster. *Edgar Rice Burroughs' Tarzan: The Sunday Comics, 1931–1933*. Milwaukie: Dark Horse Books, 2013 (siehe S. 5).
- [2] Neil Cohn. *The Visual Language of Comics: Introduction to the Structure and Cognition of Sequential Images*. London: Bloomsbury Academic, 2013 (siehe S. 4).
- [3] Tom DeFalco und Joe Bennett. *The Amazing Spider-Man*. New York: Marvel, 1963 (siehe S. 4).
- [4] Chris Gavaler. *On the Origin of Superheroes: From the Big Bang to Action Comics No. 1*. Iowa City: University of Iowa Press, 2015 (siehe S. 3).
- [5] Ulf Hagen. „Where do Game Design Ideas Come From? Innovation and Recycling in Games Developed in Sweden“. In: *Breaking New Ground: Innovation in Games, Play, Practice and Theory: Proceedings of the 2009 Digital Games Research Association Conference*. London: Digital Games Research Association, 2009. URL: http://www.digra.org/dl/display_html?chid=09287.25072.pdf (siehe S. 26).
- [6] Edmund Hamilton u. a. *World's Finest #153*. New York: DC Comics, 1965 (siehe S. 5).
- [7] Robert C. Harvey. *The Art of the Comic Book: An Aesthetic History*. Jackson: University Press of Mississippi, 1996 (siehe S. 3, 5).
- [8] Eric Herhuth. *Pixar and the Aesthetic Imagination*. Oakland: University of California Press, 2017 (siehe S. 10).
- [9] Teresa de Lauretis und Stephen Heath. *The Cinematic Apparatus*. London: Palgrave Macmillan, 1980 (siehe S. 7).
- [10] Raul Reyes Luque. *The Cel Shading Technique*. Techn. Ber. CiteSeer, 2012. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.691.5831&rep=rep1&type=pdf> (siehe S. 3).
- [11] Scott McCloud. *Understanding Comics: The Invisible Art*. New York: HarperPerennial, 1994 (siehe S. 3).
- [12] Tina O'Hailey. *Hybrid animation: Integrating 2D and 3D Assets*. 2. Aufl. New York: Routledge, 2014 (siehe S. 7).
- [13] Edward Waterman Townsend. *The Yellow Kid in McFadden's Flats*. New York: G. W. Dillingham, 1986 (siehe S. 3).

- [14] Brian K. Vaughan und Fiona Staples. *Saga*. Portland: Image Comics, 2012 (siehe S. 4).
- [15] Malcolm Wheeler-Nicholson u. a. *New Fun Comics #1*. New York: National Allied Publications, 1935 (siehe S. 3).
- [16] Bill Willingham u. a. *Fables*. New York: Vertigo, 2002 (siehe S. 22).

Audiovisuelle Medien

- [17] *Avatar*. Film. Drehbuch/Regie: James Cameron, Produktion: James Cameron, Jon Landau. 2009 (siehe S. 10).
- [18] *Deadpool*. Film. Drehbuch: Rhett Reese, Paul Wernick, Regie: Tim Miller, Produktion: Simon Kinberg, Ryan Reynolds, Lauren Shuler Donner. 2009 (siehe S. 10).
- [19] *Feast*. Film. Drehbuch Regie: Patrick Osborne, Produktion: Kristina Reed. 2014 (siehe S. 17).
- [20] *Hercules*. Film. Drehbuch: Ron Clements, Barry Johnson, Don McEnery, Irene Mecchi, Jon Musker, Bob Shaw, Regie: John Musker und Ron Clements, Produktion: John Musker, Ron Clements, Alice Dewey, Noreen Tobin. 1997 (siehe S. 7).
- [21] *Love, Death + Robots*. Serie. Produktion: David Fincher und Tim Miller. 2019 (siehe S. 10, 11).
- [22] *Snow White and the Seven Dwarfs*. Film. Drehbuch: Ted Sears, Richard Creedon, Regie: David D. Hand, Produktion: Walt Disney. 1937 (siehe S. 7).
- [23] *Sonder*. Film. Drehbuch: Jessica Kitchens und Neth Nom, Produktion: Sara Sampson, Regie: Neth Nom. 2018. URL: <https://vimeo.com/253571683> (siehe S. 12, 13).
- [24] *Spider-Man: Into The Spider-Verse*. Film. Drehbuch: Phil Lord, Regie: Bob Persichetti, Peter Ramsey und Rodney Rothmann, Produktion: Avi Arad, Amy Pascal, Phil Lord, Christopher Miller und Christina Steinberg. 2018 (siehe S. 13, 14, 16).
- [25] *Toy Story*. Film. Drehbuch: Joss Whedon, Andrew Stanton, Joel Cohen, Alec Sokolow, Regie: John Lasseter, Produktion: Ralph Guggenheim, Bonnie Arnold. 1995 (siehe S. 7).

Software

- [26] Autodesk. *Maya*. URL: <https://www.autodesk.com/products/maya/overview> (siehe S. 36).
- [27] Blender. URL: <https://www.blender.org> (siehe S. 36).
- [28] Bandai Namco Entertainment. *Ni no Kuni II: Revenant Kingdom*. Playstation 4. 2018 (siehe S. 25).
- [29] Telltale Games. *The Wolf Among Us*. Windows. 2013 (siehe S. 22).

- [30] Nintendo. *The Legend of Zelda: Breath of the Wild*. Nintendo Switch. 2017 (siehe S. 20, 21).
- [31] Nintendo. *The Legend of Zelda: The Wind Waker*. GameCube. 2002 (siehe S. 19).
- [32] Nintendo. *The Legend of Zelda: The Wind Waker HD*. Wii U. 2013 (siehe S. 19, 20).
- [33] Southend Interactive Ubisoft Paris. *XIII*. Windows. 2003 (siehe S. 8).
- [34] Pixologic. *ZBrush*. URL: <https://pixologic.com> (siehe S. 36).
- [35] Sega. *Valkyria Chronicles 4*. Windows. 2018 (siehe S. 23, 24).
- [36] Smilebit. *Jet Set Radio*. Dreamcast. 2000 (siehe S. 3, 8).
- [37] *Substance Designer*. URL: <https://www.substance3d.com/products/substance-designer> (siehe S. 36).
- [38] *Unity 3D*. URL: <https://www.unity.com/> (siehe S. 12).

Online-Quellen

- [39] *[Shader Graph] Shader Doesn't Compile When Using a Gradient Property and a Regional Format With Comma Decimal Separator*. 2015. URL: <https://issuetracker.unity3d.com/issues/shader-graph-shader-doesnt-compile-when-using-a-gradient-property-and-a-regional-format-with-comma-decimal-separator-is-used> (siehe S. 39).
- [40] Emad Ahmed. *In praise of cel-shading*. URL: <https://www.eurogamer.net/articles/2018-12-05-in-praise-of-cel-shading> (siehe S. 19).
- [41] Ashif Ali. *Sonder: Animated Film Made With Unity*. 2018. URL: <https://cghow.com/sonder-animated-film-made-with-unity/> (siehe S. 12).
- [42] *Best-Reviewed Switch Games Of All Time*. URL: <https://www.metacritic.com/feature/25-best-nintendo-switch-games> (siehe S. 20).
- [43] *Blur Studios*. URL: <http://www.blur.com> (siehe S. 10).
- [44] Brian. *Zelda: Breath of the Wild art director on how The Wind Waker HD shaped the game's art style*. 2017. URL: <https://nintendoeverything.com/zelda-breath-of-the-wild-art-director-on-how-the-wind-waker-hd-shaped-the-games-art-style/> (siehe S. 21).
- [45] Jak Connor. *The Legend of Zelda: Breath of the Wild art-style explained*. 2017. URL: <https://www.tweaktown.com/news/63954/legend-zelda-breath-wild-art-style-explained/index.html> (siehe S. 21).
- [46] Bill Desowitz. *Spider-Man: Into the Spider-Verse: Breaking the Rules of Animation*. 2018. URL: <https://www.indiewire.com/2018/12/spider-man-into-the-spider-verse-breaking-the-visual-rules-of-animation-1202027410/> (siehe S. 14, 16).
- [47] Bill Desowitz. *Spider-Man: Into the Spider-Verse: Phil Lord & Chris Miller Introduce Game-Changer Miles Morales*. 2018. URL: <https://www.indiewire.com/2018/11/spider-man-into-the-spider-verse-phil-lord-chris-miller-miles-morales-animation-game-changer-1202020215/> (siehe S. 15).

- [48] Bandai Namco Entertainment Europe. *Ni no Kuni II: Revenant Kingdom – PS4/PC – Behind the scenes: Light and Shadow (Developer diaries)*. 2018. URL: <https://www.youtube.com/watch?v=EUT2qplBYQE> (siehe S. 26).
- [49] Matt Fussel. *Hatching and Cross Hatchin'*. URL: <https://thevirtualinstructor.com/hatchingcrosshatching.html/> (siehe S. 5).
- [50] Gamespot. *The Legend of Zelda: The Wind Waker Video Interview*. 2019. URL: <https://www.gamespot.com/videos/the-legend-of-zelda-the-wind-waker-video-interview/2300-2911976/> (siehe S. 20).
- [51] Todd Gilchrist. *Caught In A Ham Is The Spider-Verse Prequel You Never Knew You Needed*. 2019. URL: <https://nerdist.com/article/spider-ham-spider-man-into-the-spider-verse-prequel/> (siehe S. 16).
- [52] AJ Glasser. *Valkyria Chronicles Review: World War II Gets a Girlie Makeover*. URL: <https://kotaku.com/valkyria-chronicles-review-world-war-ii-gets-a-girlie-5110804> (siehe S. 24).
- [53] Blender Guru. *Money doesn't interest me – Creator of Blender talks about its future*. 2018. URL: https://www.youtube.com/watch?time_continue=2&v=qJEWOTZnFeg (siehe S. 48).
- [54] Jason Hidalgo. *Flowers and gunpowder: Valkyria Chronicles 4 review | Technobubble*. 2018. URL: <https://eu.rgj.com/story/life/2018/10/12/valkyria-chronicles-4-review-technobubble/1593926002/> (siehe S. 24).
- [55] John Hilgart. *In Defense of Dots: The lost art of comic books*. 2010. URL: <https://4cp.posthaven.com/in-defense-of-dots-the-lost-art-of-comic-book> (siehe S. 5).
- [56] Chris Hinton. *How The Wind Waker got better with age*. 2019. URL: <https://www.nintendoenthusiast.com/2019/01/08/wind-waker-got-better-age/> (siehe S. 19).
- [57] Jason. *Making of Disney Feast Short Production*. 2015. URL: <http://www.cgmeetup.net/home/making-of-disney-feast-short-production/> (siehe S. 17, 18).
- [58] Helen Leblanc. *Non-Photorealistic Rendering 101*. URL: <https://info.e-onsoftware.com/blog/non-photorealistic-rendering-101> (siehe S. 6).
- [59] Olivia Lennox. *Cel Shading: the Unsung Hero of Animation?* URL: <http://www.animatormag.com/computer/cel-shading-hero-animation/> (siehe S. 8).
- [60] *LollipopShaders*. URL: <http://www.lollipopshaders.com/> (siehe S. 6).
- [61] Raul Reyes Luque. *Love, Death + Robots with Blur Studio – Behind the Scenes*. 2019. URL: <https://cgsociety.org/news/article/4413/love-death-plus-robots-with-blur-studio-behind-the-scenes> (siehe S. 11).
- [62] Jeffrey Matulef. *Witness a comparison between The Wind Waker HD and the original GameCube version*. 2013. URL: <https://www.eurogamer.net/articles/2013-09-12-witness-a-comparison-video-of-the-wind-waker-hd-and-its-gamecube-counterpart> (siehe S. 20).
- [63] Luiz Oosterbeek. *Lascaux Cave Paintings – An Introduction*. URL: <http://www.bradshawfoundation.com/lascaux/> (siehe S. 3).

- [64] Luke Plunkett. *The Art Of Ni no Kuni 2*. 2018. URL: <https://kotaku.com/the-art-of-ni-no-kuni-2-1825485221> (siehe S. 26).
- [65] Thomas Poulet. *Ni No Kuni 2: frame analysis*. 2018. URL: <https://blog.thomaspoulet.fr/ninokuni2-frame/> (siehe S. 25).
- [66] *Printing Comic Books Through the Ages*. 2012. URL: <https://www.printindustry.com/blog/?p=4071> (siehe S. 5).
- [67] Rhianna. *Why Art Style Is More Important Than Graphics in Game Design*. 2017. URL: <https://maxlouiscreative.com/2017/11/24/art-style-vs-graphics-game-design/> (siehe S. 26).
- [68] Dan Sarto. *Creating A Stylized Universe for Sony's Spider-Man: Into the Spider-Verse*. 2018. URL: <https://www.awn.com/animationworld/creating-stylized-universes-sonys-spider-man-spider-verse> (siehe S. 15).
- [69] Sega. *Hitoshi Fukazawa, chief artist on Valkyria Chronicles 4 talks about the most challenging aspect of designing the feel of the game*. URL: https://twitter.com/SEGA_Europe/status/1059822717116051457 (siehe S. 24).
- [70] Sega. *The CANVAS Engine: Watercolor Paintings in Motion*. URL: <http://valkyria.sega.com/canvas.php> (siehe S. 25).
- [71] Mike Seymour. *Ink Lines and Machine Learning*. 2019. URL: <https://www.fxguide.com/fxfeatured/ink-lines-and-machine-learning/> (siehe S. 15).
- [72] Simon. *Zelda Wind Waker – Hyrule Travel Guide*. 2014. URL: <https://simonschreibt.de/gat/zelda-wind-waker-hyrule-travel-guide/> (siehe S. 20).
- [73] Yekaterina Sinelschikova. *People want the rough stuff: Russian animator on Netflix's hottest cartoon*. 2019. URL: <https://www.rbth.com/arts/330132-russian-animator-netflix-cartoon> (siehe S. 11).
- [74] Simon J. Smith. *Baymax Dreams*. 2018. URL: <https://unity.com/madewith/baymax-dreams> (siehe S. 48).
- [75] *Studio Ghibli*. URL: <https://www.studioghibli.com.au/> (siehe S. 26).
- [76] The New York Times. *Spider-Man: Into the Spider-Verse and How a Scene Crawled to Life | Anatomy of a Scene*. 2019. URL: <https://www.youtube.com/watch?v=BULa8qozTy0> (siehe S. 14).
- [77] tomPower. *No, Netflix's Love, Death & Robots Short The Witness Didn't Use Live-Action*. URL: <https://www.ign.com/articles/2019/04/02/how-netflixs-love-death-robots-created-that-eye-popping-animation> (siehe S. 10).
- [78] Unity. *Unite Berlin 2018 – Exclusive Premiere: Sonder*. 2018. URL: <https://www.youtube.com/watch?v=LJQWmHMVvJM&t=1347s> (siehe S. 12).