



RISE OF A SAMURAI
VON ALEXANDER BAUR

EIN ERFAHRUNGSBERICHT: Konzeption und Umsetzung eines komplexen Animationsfilms mithilfe von KI-Tools und modernen Workflows

BACHELORARBEIT VON ALEXANDER BAUR

HOCHSCHULE **RAVENSBURG-WEINGARTEN-UNIVERSITY**
STUDIENGANG **MEDIENDESIGN UND DIGITALE GESTALTUNG**
ABSCHLUSS **BACHELOR OF SCIENCE**

GUTACHTER I **PROF. MARKUS LAUTERBACH**
GUTACHTER II **MAGNUS PFLEGHAR**

ZEITRAUM **18. MÄRZ 2024 - 17. SEPTEMBER 2024**
ABGABETERMIN **10. SEPTEMBER 2024**

ABSTRACT

Die vorliegende Bachelorarbeit "Ein Erfahrungsbericht: Konzeption und Umsetzung eines komplexen Animationsfilms mithilfe von KI-Tools und modernen Workflows" beschäftigt sich mit der Untersuchung von modernen und etablierten Methoden zur Entwicklung eines Animationsfilms, die durch den Einsatz von Künstlicher Intelligenz bereichert wurden.

Für die Umsetzung des Projektes wurden alle grundlegenden Schritte im Animationsprozess berücksichtigt, die eine breite Palette an verschiedenen 3D-Bereichen und Werkzeugen abdecken.

Die Themenbereiche lassen sich allgemein in Recherche, Konzeption, Animation, Character- und Environmentdesign, VFX, Rendering und Postproduction unterteilen.

Ziel war es, alle genannten Bereiche möglichst durch den Einsatz von Künstlicher Intelligenz umzusetzen und herauszufinden, inwiefern KI-Tools komplexe Arbeitsschritte im 3D-Prozess abnehmen können oder dabei nur eine Assistenten-Rolle einnehmen.

Ein weiterer zentraler Bestandteil der Arbeit war die Nutzung der Unreal Engine, die als Dreh- und Angelpunkt sowie als verbindendes Element für alle Arbeitsschritte diente. Durch die vielseitigen Features und die nahtlose Integration der verschiedenen genutzten Tools deckte die Unreal Engine alle erforderlichen Teile der Animations-Pipeline in Echtzeit ab.

Das finale Produkt ist ein mehr als zweiminütiger Animationsfilm, der den Weg eines Samurai im Feudalen Japan in Form einer Heldengeschichte erzählt.

SCHLAGWÖRTER

- Animationsfilm
- Künstliche Intelligenz
- Moderne Workflows
- Unreal Engine
- 3D-Pipeline

01 EINLEITUNG 08

- 1.1 MOTIVATION 10
- 1.2 ZIELSETZUNG 11
- 1.3 IDEENFINDUNG 12
- 1.4 UMSETZUNG 14

02 RECHERCHE 20

- 2.1 HISTORISCHE EINORDNUNG 22
- 2.2 SAMURAI 26
- 2.3 KI-TOOLS 28

03 KONZEPTION 32

- 3.1 MOODBOARD 34
- 3.2 STORY 36
- 3.3 STORYBOARD 38
- 3.4 ANIMATIC 38

04 UMSETZUNG 40

4.1 ENVIRONMENT 42

- 4.1.1 GEOMETRIE 42
- 4.1.2 LANDSCAPE 50
- 4.1.3 FOLIAGE 54
- 4.1.4 PCG 58
- 4.1.5 NANITE 62
- 4.1.6 LIGHTING 64

4.2 CHARACTER 68

- 4.2.1 DESIGN 68
- 4.2.2 METAHUMANS 70
- 4.2.3 AUBAU EINES CHARACTERS 72
- 4.2.4 CLOTH 74

4.3 ANIMATION 78

- 4.3.1 GRUNDLAGEN 78
- 4.3.2 MOTION CAPTURING 80

4.4 VFX 86

- 4.4.1 EMBERGEN 86
- 4.4.2 NIAGARA 88

4.5 RENDERING 90

- 4.5.1 RENDERER 90
- 4.5.2 RENDERINGTOOLS 90

4.6 AUDIO 92

- 4.6.1 ELEVENLABS 92
- 4.6.2 UDIO 93

05 ERGEBNIS 94

06 FAZIT 102

- 6.1 HERAUSFORDERUNGEN 104
- 6.2 RESÜMEE 106
- 6.3 AUSBLICK 107

07 ZUSATZ 108

7.1 LITERATURVERZEICHNIS	110
7.2 ABBILDUNGSVERZEICHNIS	116
7.3 ABKÜRZUNGSVERZEICHNIS	120
7.4 GENDERVERMERK	121
7.5 DEEPL DISCLAIMER	121
7.6 EIDESSTATTLICHE VERSICHERUNG	122

01

EINLEITUNG



1.1 MOTIVATION

Die Fähigkeit, aufwendige Animationsfilme zu erstellen, war früher eine Aufgabe, die für große Studios mit umfangreichen Ressourcen und spezialisierten Teams vorbehalten war. [1] Doch in den letzten Jahren können durch technologischen Fortschritt und die Erschwinglichkeit von Hardware und Software, die zum Teil auch kostenlos für jeden zugänglich ist (z.B. Blender und Unreal Engine) auch einzelne Personen von zuhause aus ihre Ideen zum Leben erwecken [2].

Auch die Unreal Engine spielt in diesem Kontext eine zentrale Rolle. Als vielseitige Realtime-Engine, die sich nicht nur in der Spieleindustrie bewährt [3][4][5], sondern auch in der Film- und Serienproduktion immer mehr an Bedeutung gewinnt, hat sie sich mittlerweile bei vielen Studios als unverzichtbares Werkzeug für große Produktionen etabliert. [6][7][8] Da Epic Games den kompletten Funktionsumfang kostenlos zur Verfügung stellt, existieren zahlreiche Beispiele für visuell beeindruckende Privat- und Indieproduktionen. [9][10]

Darüber hinaus wird durch den aktuellen Trend der künstlichen Intelligenz immer deutlicher, welchen Einfluss diese Technologie auf unser Leben hat und auch in der Film- und Spieleindustrie auf wachsendes Interesse stößt. [11] Viele Studios sehen das Potenzial von KI und wollen diese immer stärker in den Entwicklungsprozess einbinden, um Projekte schneller und kostengünstiger produzieren zu können. [12]

Es war noch nie so einfach, hochqualitative Ideen als Einzelperson umzusetzen. Deswegen wurden in der vorliegenden Arbeit diese aktuellen Trends untersucht, um eine vielschichtige 3D-Pipeline innerhalb eines Animationsfilm abzubilden.



Abb. 1: Avatar - Behind The Scenes



Abb. 2: Vierzehnjähriger - Across the Spider-Verse



Abb. 3: Black Myth: Wukong - Unreal Engine 5



Abb. 4: Mandalorian - Virtual Production



Abb. 5: MoveAI - Mocap durch KI

1.2 ZIELSETZUNG

Die vorliegende Bachelorthesis verfolgte das Ziel, den grundlegenden 3D-Animations-Prozess mithilfe von modernen Werkzeugen und Künstlicher Intelligenz aufzuzeigen. Zudem wurden die Erfahrungen, die während der Entwicklung dieses Prozesses gewonnen wurden, in Form eines Erfahrungsberichts zusammengefasst.

Um den beschriebenen Prozess umzusetzen, das Ergebnis zu visualisieren und nach dessen Wirksamkeit sowie etwaigen Problemen zu testen, wurde ein Animationsfilm mit einer Laufzeit von über zwei Minuten entwickelt, der hinsichtlich der visuellen Qualität einer AAA-Produktion entsprechen sollte.

Der Schwerpunkt wurde auf die Animationen der Charaktere gelegt, die so einfach, schnell und flexibel wie möglich aufgenommen und verarbeitet werden sollten. Der Aufbau

des Setups zur Aufnahme wurde so kompakt wie möglich angesetzt und sollte ortsunabhängig möglich sein, ohne dabei die Qualität der Animationen zu beeinträchtigen. Die Aufnahme der Körper- und Gesichtsbewegungen wurde strikt voneinander getrennt, sollte jedoch mit der gleichen Hardware möglich sein, um Kosten zu vermeiden. Darüber hinaus wurde auf die Verwendung von fertigen Animation-Bibliotheken (z. B. Mixamo) möglichst verzichtet.

Bei der Erstellung der 3D-Umgebung und der visuellen Effekte wurde auf die Anwendung bereits erprobter Industriestandards zurückgegriffen. Zudem wurde bei der Umsetzung darauf geachtet, aufkommende Workflows und Tools zu priorisieren, um einen zukunftsweisenden Blick in die Industrie zu gewährleisten. Darunter fallen vor allem prozedurale und sogenannte "Non-Destructive" Workflows.

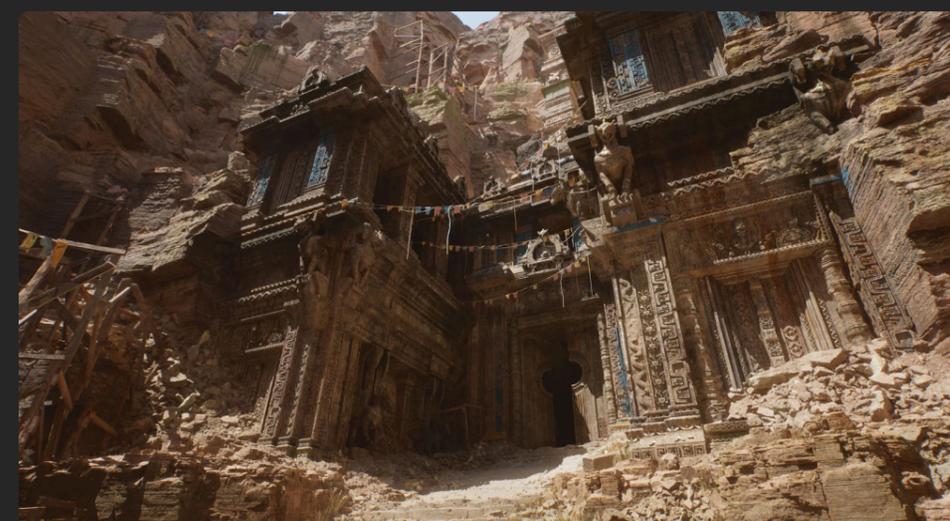


Abb. 6: Unreal Engine 5 | Next-Gen Demo

1.3 IDEENFINDUNG

Im Rahmen meines Studiums hatte ich die Möglichkeit, mich mit verschiedenen Aspekten des 3D-Bereichs vertraut zu machen. Zunächst befasste ich mich mit "Cyberpunk Projekt" mit den grundlegenden 3D-Prinzipien, welche das 3D-Modelling, Texturing, Lighting, Rendering und Compositing umfassten. Der Fokus lag zunächst auf der Software Blender mit dem Offline-Renderer Cycles. Bereits während des Renderprozesses wurde deutlich, dass das Offline-Rendering einen entscheidenden Nachteil aufweist, nämlich der Verlust an Zeit. Dieser Aspekt ist insbesondere für 3D-Projekte von essenzieller Bedeutung. Dies resultierte in der Suche nach alternativen Lösungen, welche letztlich zur Unreal Engine führte.

Da es sich bei Unreal um eine Realtime-Engine handelt, die das Rendern in Echtzeit ermöglicht und darauf ausgelegt ist, mit möglichst geringem Aufwand visuell beeindruckende Ergebnisse zu erzielen, ist dieses kostenlose Programm für diejenigen, die darauf Wert legen, unschlagbar. Darüber hinaus etabliert sich Unreal immer mehr als Industriestandard, was ausschlaggebend für die Entscheidung war, alle weiteren Projekte auf dieser Software aufzubauen.

Zunächst wurden mit "Amethyst Island" und "Modern House Interior" reine visuelle Projekte entwickelt, die mit "Alien VR" und "Runa" die interaktiven Möglichkeiten der Unreal Engine bedienten und weitere Software etablierten.

Aus den genannten Projekten lassen sich drei für mich interessante Themengebiete ableiten, nämlich Realtime, Environment Design und Workflow-Optimierung.



Abb. 7: Cyberpunk Project



Abb. 8: Amethyst Island



Abb. 9: Modern House Interior



Abb. 10: Alien VR



Abb. 11: Runa

Ein weiterer wesentlicher Aspekt bei der Projektfindung stellte mein Praxissemester als Unreal Artist im 3D-Studio „Faber Courtial“ in Darmstadt dar. Das Studio hat sich auf historische Rekonstruktionen spezialisiert und bot mir die Möglichkeit, meine Kenntnisse in diesem Bereich zu vertiefen. Aus ähnlichen Gründen entschieden sie sich zum Wechsel auf die Unreal Engine und entwickeln fortlaufend frische Workflows, Tools und Automatisierungen. In diesen Bereichen durfte ich den Großteil meiner Arbeit beitragen.

Zusätzlich fand das Thema Künstliche Intelligenz immer größeren Stellenwert in der Entwicklung. Zum Ende meines Praxissemesters kam es zum Gespräch mit dem Creative Director über das Thema Künstliche Intelligenz. Im Verlauf des Gesprächs wurde ersichtlich, dass KI nicht nur als Spielerei, sondern als integraler Bestandteil zukünftiger Entwicklungen betrachtet werden sollte. Dabei wurde deutlich, dass die Grenzen zwischen Kreativität und Technik zunehmend verschwimmen. Das Gespräch inspirierte mich derart, dass ich für meine anstehende Bachelorarbeit KI als zentralen Bestandteil wählen wollte.

Somit stellten sich die Themen Künstliche Intelligenz, Prozedurale Tools und Historik als weitere Aspekte für meine Bachelorthesis heraus.



Abb. 14: Welten Saga II - Cusco



Abb. 12: Welten Saga II - Olypmia



Abb. 13: Welten Saga II - Olypmia

Im letzten Abschnitt meiner Ideenfindung kamen meine persönlichen Präferenzen ins Spiel, die das letztendliche Endprodukt der Arbeit bestimmten.

Zum damaligen Zeitpunkt verfügte ich lediglich über begrenztes Wissen im Bereich der Animation sowie des Visual Storytelling. Aus den genannten Gründen wurde beschlossen, diese Themen zum kreativen Grundbaustein der Arbeit zu machen. Als Setting wurde das feudale Japan zur Zeit der ersten Mongoleninvasion gewählt. Erzählerisch wurde sich auf die Samurai und deren Werte festgelegt. Die Geschichte wurde als Heldenreise konzipiert.

So entstand schließlich die Idee, den Animationsfilm "Rise of a Samurai" zu entwickeln.

1.4 UMSETZUNG

VORWORT

Im Kontext dieser Bachelorarbeit wurde eine Vielzahl von Tools verwendet, die in verschiedenen Phasen der Arbeit zum Einsatz kamen. Einige Programme lassen sich

mehreren Kategorien zuordnen. Daher möchte ich in diesem Abschnitt einen Überblick über die verwendete Software geben.

UNREAL ENGINE

Die Unreal Engine stellt eine Game-Engine dar, die erstmals 1998 durch das Unternehmen Epic Games im Rahmen der Entwicklung der Ego-Shooter Spielereihe Unreal vorgestellt wurde. Epic Games (ehemals Epic MegaGames) wurde 1991 von Tim Sweeney gegründet, der das Unternehmen bis heute als CEO leitet und mit Epic Games ein offenes Ökosystem für Nutzende auf allen Plattformen verfolgt und dafür sogar einen Rechtsstreit mit Apple austrägt. [13][14]

Der Umsatz von Epic betrug im Jahr 2023 einen Umsatz von etwa 5,6 Milliarden USD und wird im Jahr 2026 voraussichtlich einen Spitzenwert von 6,2 Mrd. USD erreichen. [15]

Zusammen mit Unity (38% Marktanteil) gehört die Unreal Engine (15% Marktanteil) zu den meist verwendeten Spiele-Engines weltweit [16]. Die hohe Popularität lässt sich insbesondere auf die beeindruckenden Rendering-Fähigkeiten sowie die Vielzahl an Features zurückführen. Zudem überzeugt das attraktive Preismodell Privatpersonen sowie große AAA-Produktionen. [17][18]

Zu den bekanntesten Spielen, die mit der Unreal Engine entwickelt wurden, zählen unter anderem Hogwarts Legacy, Mass Effect, Bioshock oder auch Fortnite, welches zu den erfolgreichsten Free2Play Spielen zählt.

Auch in der Film- und Serienindustrie wird vermehrt auf die Realtime-Engine gesetzt. Bekannte Beispiele dafür sind The Mandalorian, Love Death and Robots oder auch die kürzlich veröffentlichte Fallout-Serie. Hierbei werden vor allem die Fähigkeiten der Engine in der Virtual Production genutzt, um virtuelle Grafiken in Echtzeit mit der realen Welt zu verbinden. Ein Beispiel dafür sind LED-Volumes, die Echtzeit-Umgebungen in einer umfassenden Form darstellen können und als Ersatz zu traditionellen Green-Screen Setups dienen. [19]



Abb. 16: Matrix Awakens - Unreal Engine Technical Demo

Mittlerweile befindet sich die Unreal Engine in der fünften Iteration, die die Engine erneut revolutioniert hat. Das neue dynamische Beleuchtungs- und Reflexionssystem Lumen ermöglicht die Erzeugung realistischer Lichtszenarien ohne Baking in Echtzeit. Des Weiteren steht mit Nanite ein virtualisiertes Geometriesystem zur Verfügung, welches den Detailgrad von Meshes in der Szene automatisch optimiert. Dadurch ist beispielsweise die Echtzeit-Darstellung von hunderten von ZBrush-Sculpts mit Millionen von Polygonen möglich. [20]

Darüber hinaus wurde mit PCG / Procedural Content Generation ein System zur prozeduralen Verteilung von Meshes und Foliage in die Toolbox aufgenommen. [21]

Mit der Übernahme von Quixel im Jahr 2019 wurde eine umfangreiche Bibliothek von photogescannten Objekten, Pflanzen und Materialien in das Portfolio aufgenommen und nahtlos in die Engine integriert. [22]

Mit der Einführung der Metahumans, die über einen webbasierten Charaktereditor erstellt werden können, bietet Epic Games zudem fotorealistische virtuelle Menschen an, die mit den Animationsmöglichkeiten der Engine Hand in Hand gehen. [23]

Alle genannten Systeme und Bibliotheken nehmen in der Bachelorarbeit einen großen Anteil ein und zeigen die Stärken der Unreal Engine besonders gut auf. Ausschlaggebend für die Wahl von Unreal für die vorliegende Arbeit war neben den genannten Echtzeit-Features und Game-Engine-Standards, die breite Unterstützung von Software-Schnittstellen, die es ermöglichten, Daten aus externer Software nach Unreal sicherzustellen. Das komplette Paket, das die Unreal Engine bietet, war notwendig, um die Bachelorarbeit in diesem Umfang zu realisieren.



UNREAL ENGINE

Abb. 15

BLENDER

Blender ist eine Open Source Software, die von der Blender Foundation entwickelt wurde. Ähnlich wie Maya oder 3ds-Max deckt Blender die komplette 3D-Pipeline ab und ist seit der Version 2.8 gefragter denn je. Nicht nur bei Privatanwendern, sondern auch in der Industrie gewinnt Blender zunehmend an Beliebtheit und etabliert sich als Branchenstandard.[24]

Das Wachstum ist vor allem auf den kostenlosen Open-Source-Aspekt, die aktive Community, die zahlreichen Addons, die stetig wachsende Roadmap, der Plattformunterstützung (Windows, Linux, macOS) und natürlich die zahlreichen Möglichkeiten zurückzuführen. [25]

Zu den Möglichkeiten zählen unter anderem Modeling, Texturing, Rigging, Animation, Simulation, Compositing oder auch die Unterstützung zahlreicher bekannter Renderers (z.B. Cycles, Renderman, etc.). In den neueren Versionen wird vermehrt auf

Features gesetzt, die prozedurale Systeme mithilfe von "Non-Destructive" Workflows kombinieren. Dazu zählen die Geometry-Nodes oder auch Simulation-Nodes, die Ähnlichkeiten zum Workflow in Houdini vorweisen.

Die Möglichkeiten von Blender werden von Zeit zu Zeit durch ein internes "Blender Animation Studio" demonstriert, das durch Abonnements und Spenden finanziert wird und bekannte Veröffentlichungen wie Spring (2019), Agent 327 (2017) oder Spirit Fright (2021) hervorbrachte. [26]

Aufgrund meiner bisherigen Erfahrung mit Blender sowie der erfolgreichen Umsetzung mehrerer Projekte innerhalb der Software fiel mir die Entscheidung leicht, das 3D-Modelling, UV-Unwrapping und Sculpting damit zu realisieren. Die Wahl ist jedoch nur eine persönliche Präferenz. Die gleichen Ergebnisse lassen sich auch mit Lösungen wie Maya oder 3ds Max erzielen.



Abb. 17



Abb. 18

SUBSTANCE 3D

Substance 3D ist eine Sammlung von Texturing-Tools, die den Industriestandard in diesem Bereich darstellen. [27] Im Jahr 2019 wurde Allegorithmic von Adobe aufgekauft, wobei die Summe nicht offiziell bekannt gegeben wurde. [28] Daraufhin wurde die Software in Adobes Abomodelle integriert. Bis heute können jedoch eigenständige Lizenzen über die Plattform Steam erworben werden, dabei ist zu berücksichtigen, dass diese nach einem Jahr keine Updates mehr erhalten. [29]

Substance Designer stellt einen Editor zur Erstellung von Materialien dar, welcher über einen Node-Graphen bedient wird. Die erstellten Materialien können innerhalb der Suite verwendet oder als einzelne Texture Maps exportiert werden.

Substance Sampler, ehemals unter dem Namen Alchemist bekannt, ermöglicht die Steuerung und Kombination von erstellten Substance-Materialien. Zudem bietet die Software die Option, aus Bildern von Oberflächen mithilfe von künstlicher Intelligenz und Algorithmen eigenständige Materialien zu generieren. In neueren Versionen wurde zudem die Möglichkeit der Photogrammetrie integriert, welche aus mehreren Bildern ein texturiertes Mesh erzeugt.

Substance Painter stellt hierbei das Herzstück der Substance Programme dar und wurde speziell für das Texturieren entwickelt. Painter basiert auf einem Ebenensystem mit Masken, welches mithilfe der mitgelieferten UV-Map des Meshes die gewünschten Materialien aufträgt. Ähnlich wie in gängigen Bildbearbeitungsprogrammen kann das Mesh mit Pinseln oder Alpha-Texturen bemalt werden. Zudem ist die Baking von Vertex-Daten in verschiedene Texturen möglich, um beispielsweise Abnutzungerscheinungen darzustellen. Darüber hinaus beinhaltet Painter eine umfangreiche Bibliothek mit nativen Smart-Material-Presets sowie eine Community-Library mit fertigen Materialien im Browser.

Im Rahmen dieser Arbeit wurde der Fokus auf Substance Painter gelegt, um die Hero Assets mit den gewünschten Materialien zu versehen. Das primäre Ziel bestand jedoch darin, das Texturing hauptsächlich in der Unreal Engine vorzunehmen, um eine möglichst hohe Performance zu gewährleisten.

MARVELOUS DESIGNER

Marvelous Designer ist eine Software, die speziell für das Entwerfen von virtueller Kleidung und Textilien entwickelt wurde und findet sowohl in der Modebranche als auch im 3D-Bereich ihren Einsatz. Marvelous wurde von der südkoreanischen Firma CLO Virtual Fashion Inc. entwickelt, die sich auf Softwarelösungen für die Modeindustrie spezialisiert hat und mit CLO 3D auch ein ähnliches Produkt auf dem Markt besitzt. Das Unternehmen verfolgt das Ziel, Modedesign und Textilproduktion zu revolutionieren. [30]

Die Software ermöglicht die 3D-Simulation von Kleidungsstücken, wobei physikalische Eigenschaften wie Stoffdichte, Elastizität und Faltenwurf realistisch dargestellt werden. Dazu müssen 2D-Schnittmuster entweder importiert oder direkt in der Software erstellt und anschließend zusammengenäht werden. Die 2D-Schnittmuster werden automatisch in 3D überführt. In Abhängigkeit von dem gewünschten Verhalten des

Kleidungsstücks sowie der Präzision der Simulation bietet Marvelous mit zahlreichen Parametern ein breites Spektrum an Möglichkeiten.

Die Entscheidung für die Verwendung der Software basierte auf den vielversprechenden Simulationsergebnissen der Kleidung sowie der einfachen Möglichkeit, die Kleidung nach Blender oder direkt nach Unreal zu exportieren. Ein weiterer Vorteil ist, dass Marvelous aus den 2D-Schnittmustern automatisch UV-Maps erstellt, was den Texturierungsprozess deutlich beschleunigt.



Abb. 19

CASCADEUR

Cascadeur ist eine auf Keyframe-Animationen spezialisierte Software, die sich auf die Erstellung von realistischen Charakterbewegungen und physikalisch korrekten Animationen mithilfe von künstlicher Intelligenz fokussiert. [31]

Das Tool wurde ursprünglich als In-House-Tool des Spielestudios "Nekki" für das Spiel "Shadow Fight" entwickelt und steht nun allen Interessierten als Abonnement zur Verfügung. [32] Der Slogan von Cascadeur lautet "The Easiest Way To Animate". Mithilfe des AutoPhysics-Tools können aus wenigen Keyframes realistische und natürliche Bewegungen erzeugt werden, wobei der Einsatz von KI eine wesentliche Rolle spielt.

Der KI-Aspekt sowie der Slogan waren maßgeblich für die Entscheidung, Cascadeur als Hauptwerkzeug für die Überarbeitung der Animationen zu verwenden. Zudem steht ein Template für den UE5-Manny zur Verfügung, welches mit dem MetaHuman-Skelett kompatibel ist. Somit wurde sichergestellt, dass die Pipeline nach Unreal problemlos funktionieren wird.



Abb. 20

SPEEDTREE

Speedtree hat sich auf die Erstellung von Vegetation, im Englischen als Foliage bekannt, mithilfe eines simplen Node-Graphen spezialisiert. Durch diverse Parameter lassen sich die Eigenschaften einer Node verändern, um die Foliage gemäß den Wünschen anzupassen. [33]

Speedtree wurde von Interactive Data Visualization, Inc. entwickelt und erstmals im Jahr 2002 veröffentlicht. Im Jahr 2021 erfolgte die Übernahme des Unternehmens durch Unity Technologies. [34]

Die Software gilt als Industriestandard zur Erstellung von Vegetation innerhalb der Spieleindustrie. Bekannte Beispiele sind unter anderem The Witcher 3, Elden Ring oder auch Uncharted 4.

Da Speedtree ein bereits erprobtes Tool innerhalb der Industrie darstellt und die Integration in die Unreal Engine problemfrei möglich ist, wurde die Software primär für die Erstellung der Bäume eingesetzt. In Kombination mit Pflanzenatlas aus den Megascans konnte eine glaubwürdige Vegetation generiert werden. Zusätzlich bietet die Pipeline nach Unreal die Möglichkeit, Windbewegungen automatisch über ein mitgeliefertes Material zu erzeugen.



Abb. 21



Abb. 22

EMBERGEN

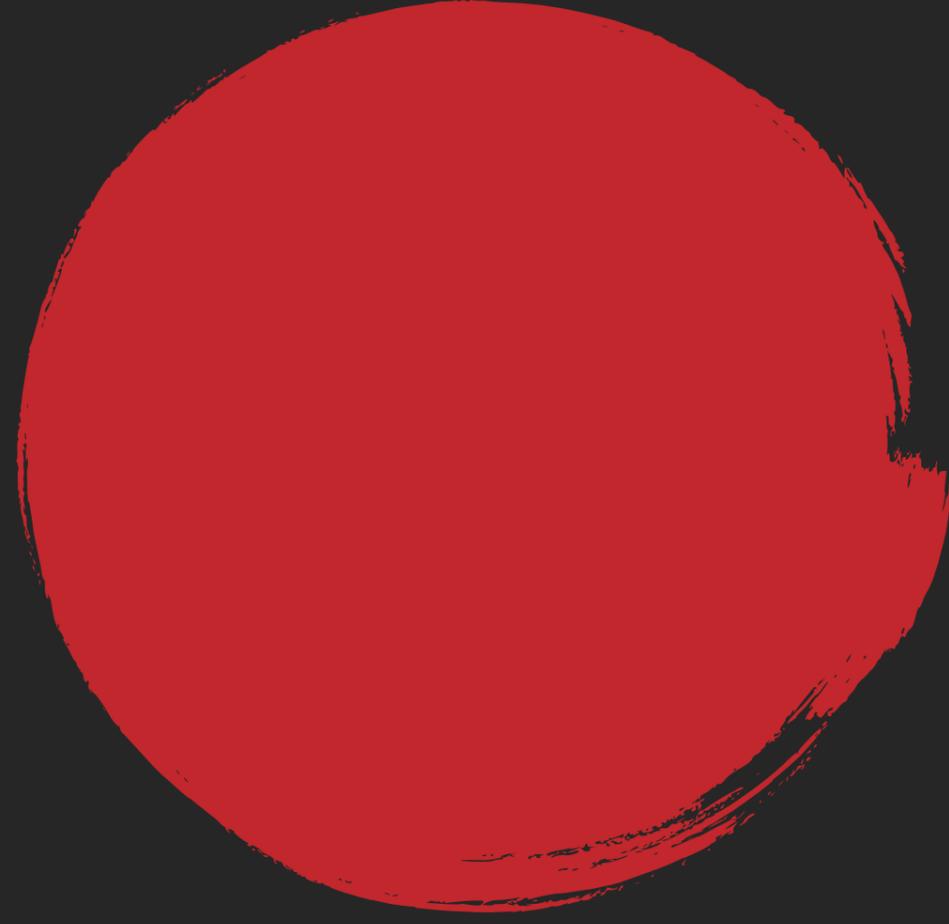
EmberGen ist eine Anwendung zur Echtzeit-Simulation von volumetrischen Effekten wie Feuer, Rauch, Explosionen und Nebel. Die Software ermöglicht komplexe Partikelsimulationen in Echtzeit, wodurch sich der Designprozess erheblich beschleunigt. Der Workflow basiert auf einem kompakten Node-Graphen, der eine durchgängige Node-Struktur aufweist und nur selten diesen Aufbau verlässt. Das Verändern der Simulationseigenschaften erfolgt durch Parameter innerhalb der Nodes.

Die Entwicklung von EmberGen erfolgte durch das US-amerikanische Unternehmen JangaFX, welches sich auf die Entwicklung von Echtzeit-Simulationssoftware spezialisiert hat. Des Weiteren umfasst das Produktportfolio von JangaFX die Tools LiquiGen für Wassersimulationen sowie GeoGen als Terraingenerator, welcher mit dem Programm Gaea vergleichbar ist.

Es ist in der Tat weniger bekannt, dass EmberGen von einer Vielzahl an großen und kleinen Studios genutzt wird. Dabei hat sich EmberGen als schnelle und unkomplizierte Alternative zu Houdini im Bereich der Feuersimulationen etabliert. Zu den Kunden von JangaFX zählen unter anderem Playstation, Bethesda sowie Weta Digital. [35]

Aus diesem Grund eignete sich EmberGen besonders gut für die Erzeugung der visuellen Effekte im Rahmen dieser Arbeit. Darüber hinaus ist EmberGen in der Lage, die durchgeführten Simulationen als VDBs (Volumetrische Daten) zu exportieren, welche von der Unreal Engine unterstützt werden.

02 RECHERCHE



2.1 HISTORISCHE EINORDNUNG

MONGOLISCHE INVASION

Die mongolische Invasion auf Japan im späten 13. Jahrhundert stellt einen wesentlichen Bestandteil der japanischen Geschichte dar. Sie beeinflusste die Kultur und Politik des

Landes grundlegend. Die Invasion stellt eine der bedeutendsten Schlachten Japans dar und zeigte die Tapferkeit und Stärke der Samurai im Kampf gegen einen überlegenen Gegner.

MONGOLISCHES REICH

Zu Beginn des 13. Jahrhunderts stiegen die Mongolen unter der Führung von Dschingis Khans zur Invasionsmacht auf und eroberten große Teile Ost- und Westasiens. Durch die Unterwerfung Nordchinas waren die Mongolen zudem gefürchteter denn je, denn sie hatten Zugriff zu neuartigen Waffentechnologien wie Schusswaffen und Explosivgeschosse.

Kublai Khan, der Enkel von Dschingis Khan und Gründer der Yuan-Dynastie (regierendes mongolisches Kaiserhaus) hatte den Plan, die Macht der Mongolen auf Japan auszuweiten.

Dadurch würde die Expansion des Mongolischen Reiches fortgesetzt werden. Zusätzlich ist Japan aufgrund der kostbaren territorialen Ressourcen ein wirtschaftlich interessantes Ziel. [36]



Abb. 23: Kublai Khan

RUHE VOR DEM STURM

Zunächst unternahm Kublai Khan im Jahr 1268 den Versuch, auf diplomatischem Wege ein "freundschaftliches" Verhältnis zum Shogunat von Kamakura herzustellen. Dazu wurde eine Gesandtschaft mit verschiedenen Forderungen entsandt. Sollte deren Erfüllung verweigert werden, wurde eine militärische Invasion angedroht. Das Shogunat reagierte gleichgültig und ignorierte alle Forderungen. Auch weitere Gesandtschaften wurden ohne eine entsprechende Antwort zurückgeschickt. Kublai Khan war erbost und wollte seine Drohungen wahr machen. [37]



Abb. 24: Brief an das Shogunat

DIE ERSTE INVASION "SCHLACHT VON BUN'EI" | 1274

Eine Streitmacht von etwa tausend Schiffen, bestehend aus mongolischen, chinesischen und koreanischen Truppen mit einer Gesamtstärke von zwanzig- bis vierzigtausend Mann, wurde nach Japan entsandt. Zuerst wurden die Inseln Tsushima und Iki angegriffen, die auf dem Weg zum japanischen Festland lagen. Auf Tsushima erwartete die Invasoren der Überlieferung nach eine Kavallerie von achtzig berittenen Samurai unter der Führung von Sō Sukekuni. Die Verteidigung der Samurai blieb erfolglos, sodass die beiden Inseln von den Invasoren erobert wurden. Daraufhin konnten die mongolischen Truppen in der Hakata-Bucht auf Kyūshū, dem japanischen Festland, landen, was zur sogenannten "Schlacht von Bun'ei" führte. Den Japanern gelang es, eine Streitmacht von mindestens vier- bis sechstausend Samurai und einfachen Soldaten zu mobilisieren, die sich den Mongolen entgegenstellten. Obwohl die Samurai gut ausgebildet waren, verfügten sie über wenig Kriegserfahrung. Zudem waren sie hinsichtlich ihrer Ausrüstung und ihrer zahlenmäßigen Stärke unterlegen. Außerdem waren die Samurai lediglich an den ehrenhaften Zweikampf gewöhnt, was sich deutlich von der kompromisslosen Kriegstaktik der Mongolen unterschied. Durch den Einsatz von Sprengkörpern und Schusswaffen kämpften die Japaner zudem in Angst und Schrecken. In der Folge drängten die Mongolen die Verteidiger ins Landesinnere, während sich die Japaner in der Festung "Mitzuki" verschanzten und auf weitere alliierte Truppen warteten. Die mongolischen Invasoren zogen sich jedoch vorerst zurück. Als Gründe für

den Rückzug der Mongolen sind Versorgungsprobleme und eigene Verluste anzunehmen. Der Rückzug führte zum Wendepunkt der Schlacht, denn nach der Einschiffung der Truppen in der Hakata-Bucht zog ein Taifun auf, der einen Großteil der Schiffe versenkte. Die siegreichen Japaner deuteten den Sturm als "Kamikaze – Göttlicher Wind" und dankten den Göttern und Geistern des Landes, den Kami, für ihre erhörten Gebete. [38][39]



Abb. 25: Göttliche Winde - Kamikaze | Kikuchi Yōsai

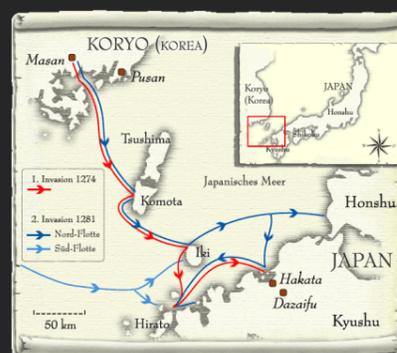


Abb. 26: Mongolen Invasion - Karte

VORBEREITUNG AUF WEITERE INVASIONEN

Den Japanern wurden Informationen zugetragen, denen zufolge eine zweite Invasion zu erwarten sei. Aus diesem Grund wurde die Entscheidung getroffen, die eigenen Streitkräfte zu verstärken und Maßnahmen zur Verteidigung Kyūshū zu ergreifen. Aufgrund der ständigen Bereitschaft aller kriegstauglichen Männer kam es zu einem Mangel an Arbeitskräften in der Landwirtschaft, was mehrere Hungersnöte zur Folge hatte. Zusätzlich verbreitete sich durch den letzten Sieg der Shinto-Glaube stark im Land. Deswegen wurden die Priester und Mönche gebeten, in den Schreinen und Tempeln für einen weiteren Sieg Japans zu beten. Im Mai 1275 wurde ein

mongolischer Gesandter zum Bakufu geschickt, welcher jedoch hingerichtet wurde, um die Position Japans als unnachgiebige und starke Macht zu demonstrieren. Zunächst wurde auf Seiten Japans ein direkter Angriff auf Korea geplant, jedoch aufgrund mangelnder finanzieller Mittel verworfen. Stattdessen wurde eine Verteidigungsmauer um die Hakata-Bucht errichtet. Kublai Khan war nicht gewillt, die letzte Niederlage und Provokationen Japans hinzunehmen und forderte den Ausbau der mongolischen Streitmacht. Demnach bestand das Heer der zweiten Invasion aus etwa hundert-siebzigtausend Soldaten und viertausend Schiffen. [36]



Abb. 27: Samurai entern mongolische Schiffe



Abb. 28: Schlacht an der Hakata-Bucht

DIE ZWEITE INVASION "SCHLACHT VON KŌAN" | 1281

Die zweite Invasion Japans durch die Mongolen lässt sich in zwei Phasen unterteilen. Grund dafür waren logistische Probleme, die durch einen Aufstand der Hafentarbeiter und den Tod eines hohen chinesischen Offiziers verursacht wurden. Ohne chinesische Unterstützung griffen die Koreaner im Frühjahr 1281 zunächst die Insel Tsushima an. Der erste Vorstoß konnte dank der besseren Verteidigung und einer Truppenstärke von über tausend Samurai abgewehrt werden. Beim zweiten Vorstoß mit Unterstützung des ersten Teils der chinesischen Truppen wurden zunächst Tsushima und Iki eingenommen. Danach rückten die Truppen in Richtung Kyūshū vor und landeten im Gebiet zwischen Munakata und der Hakata-Bucht.

Diesmal gingen die Samurai mit einer offensiveren Strategie vor und konnten die angreifenden Truppen zunächst zurückschlagen. Mit dem Eintreffen der restlichen chinesischen Truppen kam es zur Hauptkonfrontation mit den Verteidigern in der sogenannten "Schlacht von Kōan", welche sich über zwei Monate erstreckte. Am 15./16. August kam es erneut zu einem unerwarteten Taifun, der massive Schäden auf Seiten der Mongolen anrichtete. Etwa ein Drittel der koreanischen Flotte und mehr als die Hälfte der chinesischen Truppen wurden vernichtet. Daraufhin zogen sich die restlichen Truppen zurück, was einen weiteren Sieg Japans zur Folge hatte. [40]

2.2 SAMURAI

DER SAMURAI

Der Begriff "Samurai" hat seinen Ursprung in "Saburai", was so viel bedeutet wie "Diener". und geht der Bedeutung "Jemanden, der dem Herrn/Adeligen dient" nach. In der deutschen Sprache kann der Begriff mit "Krieger" übersetzt werden, was die Zugehörigkeit zum japanischen Kriegsadel erklärt. In Japan hingegen wird "Bushi" verwendet, der jemanden bezeichnet, der "die Kriegskunst beherrscht".

Aus der Begriffserklärung lässt sich ableiten, dass ein Samurai ein bewaffneter Krieger ist, der zum Schutz eines Adligen dient. Die Grundherren sind dabei der direkten Kontrolle des Tenno, also des Kaisers, unterstellt. Ihre Aufgaben umfassten die Herrschaft über ihr Gebiet sowie den Schutz der dort lebenden Menschen. Daher ist davon auszugehen, dass die Grundherren selbst ausgebildete Krieger sind.

Ein klassischer Samurai ist ein mit Pfeil, Bogen und Schwert bewaffneter Reiterkrieger, der über die Zeit selbst

zum Grundherren aufsteigen kann und dadurch die Möglichkeit erhält, andere für sich arbeiten zu lassen. Neben dem Kriegshandwerk bildeten sich die Samurai auch in Kunst, Wissenschaft, Religion und Philosophie aus. Der Alltag war auf ständige Kampfbereitschaft ausgerichtet. Jede Bewegung glich einem Ritual, das von Kindheit an einstudiert wurde.

Das Schwert spielte für den Samurai eine wichtige Rolle und wurde daher auch als "Seele des Samurai" bezeichnet. Die Art und Weise, wie das Schwert geführt wird, nennt man "Kenjutsu". Ein Duell war meist ein ritueller Zweikampf gegen einen ebenbürtigen Gegner, bei dem es vor allem um die Ehre ging. Die Kämpfe wurden sehr überlegt geführt und waren nach wenigen Schlägen entschieden. Die Schlachten folgen diesem Ansatz und sind als abgesprochene Einzelduelle zu verstehen. War kein mutiger Krieger mehr bereit, sich dem Kampf zu stellen, galt die Schlacht als gewonnen. [41][42]



Abb. 29: Japanischer Samurai nimmt Tributzahlungen von Unterworfenen

BUSHIDŌ DER WEG DES KRIEGERS

Die Samurai handeln nach einem Verhaltenskodex, dem sogenannten "Bushidō", übersetzt der "Weg des Kriegers". Der Begriff setzt sich aus "bushi = Krieger" und "dō = Weg" zusammen. Das "Bushidō" ist eng mit dem Begriff "Budō" verbunden, welcher die praktische Disziplin der japanischen Kriegskunst darstellt. Der Ehrenkodex lässt sich drei Glaubensrichtungen zurückführen, dem Buddhismus (ein schicksalsbezogener Glaube, der die Angst bei Gefahren nimmt), Zen (lehrt die Meditation und Achtsamkeit, zur Stärkung des Geistes) und

Shintoismus (enthält die Werte und die spirituelle Verbindung mit den Ahnen und Geistern) zurückführen lassen. Parallel zum Aufstieg des Shogunats zur Zeit des 12. Jahrhunderts entwickelte sich der Kodex.

Für die Samurai ist er eine Lebenseinstellung, die fundamental für jeden Aspekt des Lebens, des Alltags und des Sterbens ist. Das Bushidō ist nicht niedergeschrieben, sondern liegt im Herzen der Samurai. Dabei symbolisieren die sieben Tugenden die Werte eines Samurai. [43]

DIE 7 TUGENDEN

義

Gi = Aufrichtigkeit, Gerechtigkeit und Rechtschaffenheit
Ein Samurai trifft seine Entscheidungen nach dem Prinzip der Gerechtigkeit und der Integrität. Er handelt korrekt, ohne Zweifel oder Zögern.

勇

Yu = Mut
Ein Samurai zeigt Mut, nicht nur körperlich, sondern auch moralisch. Er handelt nach seinen Überzeugungen und verteidigt das Richtige.

仁

Jin = Menschlichkeit und Güte
Ein Samurai ist mitfühlend und zeigt Fürsorge gegenüber anderen. Er nutzt seine Stärke, um die Schwachen zu schützen und hilft denen, die in Not sind.

礼

Rei = Einhaltung der Etikette, Höflichkeit
Ein Samurai verhält sich stets respektvoll und höflich gegenüber anderen, unabhängig von deren Status

誠

Makoto / Shin = Wahrheit, Wahrhaftigkeit und Unverfälschtheit
Ein Samurai zeichnet sich durch absolute Aufrichtigkeit aus. Seine Worte und Taten sind stets wahrhaftig und er handelt mit ehrlicher Absicht.

名譽

Meiyo = Ehre, Wertschätzung der Pflichten und Privilegien
Ein Samurai strebt stets danach, in Ehren zu leben und niemals Schande über sich oder seine Familie zu bringen.

忠義

Chūgi / Chū = Treue, Loyalität und Pflichtbewusstsein
Ein Samurai ist seinem Herrn, seiner Familie und seinen Prinzipien gegenüber absolut loyal. Seine Treue ist beständig und unerschütterlich.

2.3 KI-TOOLS

VORWORT

Ein weiterer Teil der Recherche umfasste die Suche nach KI-Tools, die den Entwicklungsprozess der Arbeit bereichern könnten. Aufgrund der breiten Anwendung von künstlicher Intelligenz in zahlreichen Bereichen und Werkzeugen war eine Priorisierung bei der Recherche erforderlich. Die für das Projekt

KONZEPT

ChatGPT ist ein von OpenAI entwickelter KI-Chatbot, welcher auf einem Large Language Model basiert. Das Sprachmodell wurde auf riesigen Datensätzen trainiert und ist in der Lage, textbasierte Antworten auf sowohl einfache als auch komplexe Fragestellungen zu liefern. Der Chatbot eignet sich insbesondere als Basis für Recherchezwecke sowie das Schreiben der Geschichte an. Da die Antworten häufig einen künstlichen Eindruck erwecken oder Fehler enthalten, ist eine Überprüfung und Überarbeitung durch Menschenhand sinnvoll. Zudem basieren einige andere Anwendungen auf ChatGPT, sodass eine Vertiefung mit dem Thema Prompting von Vorteil war. [44]

Für die Entwicklung des Storyboards und etwaigen Konzeptbildern erwiesen

AUDIO

Innerhalb des Audiobereichs lassen sich mit künstlicher Intelligenz vermutlich die beeindruckendsten Ergebnisse erzielen. Im Rahmen der Recherche stellte sich ElevenLabs als vielversprechendste Option zur Generierung von Soundeffekte und Vocals für die Sprecherstimme dar. Das Feature des Instant-Voice-Clonings ermöglichte die Erstellung einer gewünschten Stimme durch wenige

relevanten Tools lassen sich daher in verschiedene Kategorien einteilen, wobei speziell auf die nach meinen Tests besten Ergebnisse eingegangen wird. Es ist anzumerken, dass es nicht alle Arten von KI-Tools in die Umsetzung des Projektes geschafft haben.

sich sogenannte "Text-to-Image"-Anwendungen, wie MidJourney, DALL-E oder Stable Diffusion, als vielversprechend. Die beiden erstgenannten Tools erzielen insbesondere auch bei einfachen Prompts überzeugende Ergebnisse und erfreuen sich in diesem Bereich großer Beliebtheit. Der Fokus liegt auf Stable Diffusion, da nach einer kostenlosen Lösung gesucht wurde, die lokal auf dem Rechner läuft. Zusätzlich bietet die Software eine hohe Anpassbarkeit durch verschiedene Modelle und viele Erweiterungen. Ähnliche Ergebnisse können auch mit den anderen Tools erreicht werden. Da viele andere KI-Anwendungen auf Stable Diffusion aufbauen, ist es empfehlenswert, sich mit dieser Software vertraut zu machen. [45]

Vocalspuren. In Kombination mit der Plattform Udio, die auf die Generierung von Musik spezialisiert ist, bestand die Möglichkeit, alle erforderlichen Audioelemente für den Film abzudecken.



Abb. 30: Generierte 3D-Assets im Vergleich - Unbearbeitet mit Wireframe

ASSETS

Die Generierung von 3D-Assets könnte zu einer wesentlichen Entlastung bei der Gestaltung des Environments beitragen. Aus diesem Grund wurde dieser Bereich genauer untersucht. Insbesondere die Features "Text-to-3D" und "Image-to-3D" waren vielversprechend, da diese in Kombination mit ChatGPT und Stable Diffusion einen interessanten Workflow ermöglichen würden. In beiden Anwendungsfällen erfolgt die Generierung des Meshes sowie der Texturen durch künstliche Intelligenz.

Eine Vielzahl der getesteten Tools konnte mit Einschränkungen kostenfrei genutzt werden. Dabei sind insbesondere die Tools Tripo, Rodin, Meshy und LumaAI positiv hervorzuheben. Tripo konnte insbesondere durch das automatische Rigging-Feature sowie den besten Resultaten überzeugen. Da die Qualität der Assets in Nahaufnahme nicht genügte, fanden die Tools keine Verwendung während der Umsetzung.



Abb. 31: Generiertes Facemesh - Hyperhuman ChatAvatar
 Prompt: male, old samurai, weathered face,
 wise demeanor, calm demeanor

CHARACTER

Obgleich die zuvor genannten Tools ebenfalls zur Generierung von Charakteren verwendet werden können, wurde nach einer Möglichkeit gesucht, einen kompatiblen Workflow mit den Metahumans zu finden, da deren Animationsfeatures als Basis des Projektes dienen. Die Verwendung von Hyperhumans ChatAvatar hat sich als vorteilhaft erwiesen. Mithilfe eines Prompts kann ein gewünschtes texturiertes Facemesh mit hoher Qualität erzeugt werden. Anschließend besteht die Möglichkeit, den erstellten Kopf innerhalb der Unreal Engine in einen Metahuman umzuwandeln, wodurch dieser die vollständige Bandbreite an Funktionen für die Animationen erhält. Aus Zeitgründen habe wurde diese Kategorie jedoch nicht berücksichtigt.

ANIMATION

Die Erfassung der für die Animationen erforderlichen Bewegungsdaten kann auf unterschiedliche Weise erfolgen. Das Ziel der vorliegenden Arbeit bestand darin, ein unkompliziertes Setup zu finden, welches ein Gegenstück zur traditionellen Methode eines Motion Capturing Anzugs darstellt. Die Lösung für das dargestellte Problem stellte die Anwendung von AI Mocap dar. Zu den führenden Tools in diesem Bereich gehören zum Zeitpunkt der Umsetzung Rokoko Studio und MoveAI. Die Entscheidung fiel letztendlich auf MoveAI, da das Setup lediglich aus einem iPhone und einem Stativ bestand und somit einfacher als die Lösung von Rokoko war. Des Weiteren kann das Setup in jeder Umgebung flexibel aufgebaut werden.

Eine Überarbeitung der aufgenommenen Bewegungsdaten ist jedoch erforderlich. In diesem Kontext war der bereits in einem anderen Kapitel erwähnte Cascadeur von Relevanz.

Neben dem AI-Mocap waren "Text-to-Animation"-Anwendungen eine weitere Option zur Erstellung der Daten. So generieren Mootion und Saymotion anhand eines kurzen Prompts komplette Bewegungen. Gelegentlich entstehen dabei brauchbare Animationen, allerdings ist das Verfahren noch zu wenig ausgereift, um in dieser Arbeit berücksichtigt zu werden.



Abb. 32: Beispiel für AI-Mocap mit MoveAI

03 KONZEPTION



3.1 MOODBOARD

Die Erstellung eines Moodboards wird im gestalterischen Kontext oft zur Findung der inhaltlichen und visuellen Stimmung des Projektes genutzt. Ähnlich einer Collage werden Inspirationen und Referenzen in Form von Bildern oder Zeichnungen zusammengetragen, um einen ersten Eindruck des angestrebten Werkes zu gewinnen. Dabei werden wesentliche Aspekte hinsichtlich des Stils wie die Farbgebung, Beleuchtung, Setting, Kameraeinstellungen, Szenerien, Dialoge und weitere Elemente dargestellt. In Abstimmung mit Außenstehenden erweist sich das Moodboard als besonders hilfreich, um die grobe Stimmung zu definieren und sich nicht in Details zu verlieren. [46]



Abb. 33

Auch im Rahmen der vorliegenden Arbeit wurde ein Moodboard erstellt, um die genannten Anhaltspunkte zu visualisieren und die grundlegende Idee zu konzipieren. Die übergeordneten Themen sind das feudale Japan zur Zeit der Mongoleninvasion sowie die Geschichte eines aufstrebenden Samurais. Als hauptsächliche Inspirationsquelle ist "Ghost of Tsushima" zu nennen, welches die Stimmung und den Inhalt des angestrebten Animationsfilms am besten einfängt.



Abb. 34



Abb. 35



Abb. 36

3.2 STORY

In diesem Teil der Konzeptionierung erfolgt die Definition einer zusammenhängenden Geschichte, welche auf der durchgeführten Recherche sowie dem angefertigten Moodboard basiert. Der Rahmen der Erzählung orientiert sich an der sogenannten Heldenreise nach Vogler, einer grundlegenden Struktur, die den Verlauf der Handlung in verschiedene Phasen unterteilt und die Entwicklung des Helden in diesem Prozess beschreibt. Um den Umfang der Arbeit zu reduzieren, wurde die Geschichte lediglich bis zur "Entscheidenden Prüfung" ausgelegt, sodass sie für den Zuschauer offen endet. Dennoch kann die Geschichte als abgeschlossen betrachtet werden, da sie den gesamten drei Akt umfasst. [47]

Der Animationsfilm thematisiert die Geschichte des Samurai Aoi, welcher den Namen seiner Familie aufgrund ihres tragischen Todes durch die Hand von mongolischen Invasoren

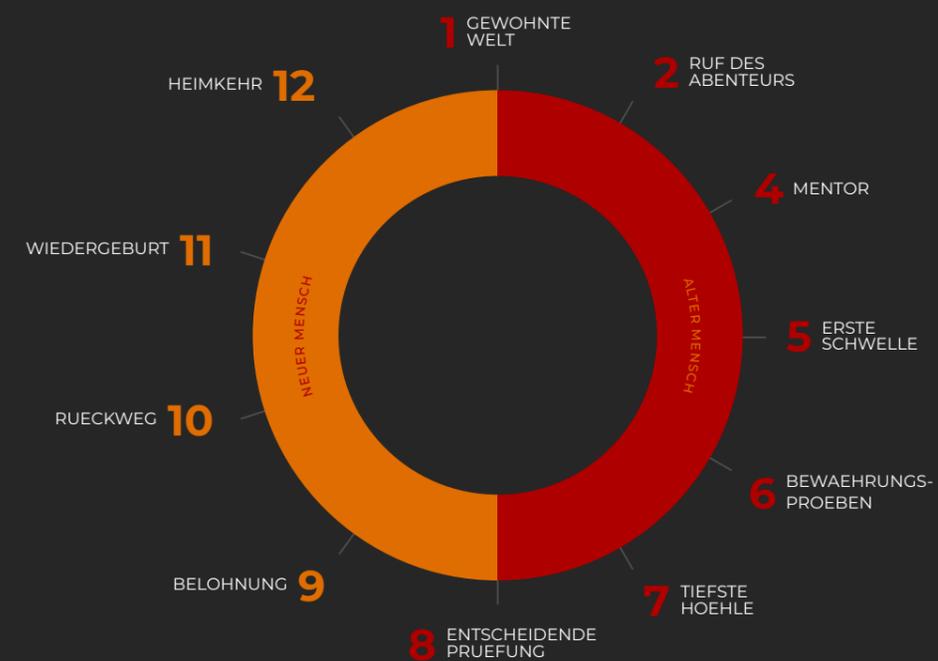
reinwaschen möchte. Um sich dem Gegner stellen zu können, muss er sich auf den Weg eines Samurais begeben, um die Tugenden, die einen solchen Krieger ausmachen, zu verinnerlichen.

Die Handlung ist im feudalen Japan zur Zeit der mongolischen Invasion zwischen den Jahren 1274 und 1281 angesiedelt. Aoi verkörpert einen herrenlosen Samurai, einen sogenannten "Ronin", der die Werte eines Samurai in Form von Tugenden repräsentiert. Die Sprecherstimme, welche den verstorbenen Vater darstellt, fungiert als Mentor des Protagonisten und begleitet ihn auf seinem Weg zum Aufstieg zu einem vollwertigen Samurai.



Abb. 37: Darstellung eines herrenlosen Samurai - Ronin

HELDENREISE NACH VOGLER



3.3 STORYBOARD

Das Storyboard stellt ein Hilfsmittel dar, welches die zuvor in Worten verfasste Geschichte in bildlicher Form wiedergibt. In den meisten Fällen wird eine skizzenhafte Darstellung gewählt, wobei jedoch die Freiheit besteht, den Stil des Storyboards nach Belieben zu wählen. Wichtig ist es, dass die Vision des Projekts klar kommuniziert wird. In diesem Kontext sind insbesondere Aspekte wie Perspektive, Blickwinkel und Kameraeinstellungen von entscheidender Bedeutung für die Veranschaulichung. [48]

Im Falle von "Rise of a Samurai" wurden mithilfe von Stable Diffusion diverse Konzeptbilder generiert, welche die Abfolge der Szenen veranschaulichen und deren Stimmung einfangen.

Dazu musste zunächst ein geeignetes Modell für Stable Diffusion ausgewählt werden. Ein Modell stellt in diesem Kontext ein neuronales Netz dar, welches mit spezifischen Daten trainiert wurde, um Bilder mit einer

bestimmten Ästhetik zu generieren. In diesem Fall wurde das Modell "ReV Animated" von der Plattform "CivitAI" ausgewählt, welche eine umfangreiche Community-Bibliothek von Modellen für Stable Diffusion bereitstellt. Des Weiteren werden mit dem Modell auch Bildbeispiele veranschaulicht, welche die getroffenen Einstellungen und Prompts beeinhalteten, die zu diesem Ergebnis geführt haben. Um ein optimales Ergebnis zu erzielen, ist es empfehlenswert, sich an diesen Vorgaben zu orientieren.

Zusätzlich wurde ein Bild als Ausgangspunkt verwendet, um genau zu bestimmen, wie stark sich die KI an diesem Input orientieren soll. Die verwendeten Bilder wurden so ausgewählt, dass die Perspektive, Formen und Strukturen des Bildes mit der Vorstellung für das letztendliche Concept Art übereinstimmen.

3.4 ANIMATIC

Im finalen Schritt der Konzeptionsphase erfolgt die Erstellung eines Animatics. Dabei handelt es sich um eine vereinfachte animierte Version des Storyboards, die eine Vorabversion des Films darstellt. Dadurch ist es möglich, einen ersten Eindruck vom Film zu gewinnen und entsprechende Änderungen am Script und Passing frühzeitig vorzunehmen. In den meisten Fällen genügt eine rudimentäre Version der Sounduntermalung, um die Wirkung des Films zu testen. Animatics spielen in sämtlichen Bereichen der Bewegtbildproduktion eine bedeutsame Rolle und werden vor den Drehs erstellt. Die Bewertung des

Entwurfs durch mehrere Instanzen erlaubt eine Prognose des Erfolgs der Produktion bereits in einer frühen Phase. Hierbei entscheidet sich ein möglicher Abbruch oder Stop der Filmproduktion. Ein solches Animatic wurde auch für dieses Projekt in einer unvertonten Version erstellt. [49]

Link: <https://youtu.be/chqaSeO3-k0>



Abb. 38: Szene 01 - Tiefer Wald



Abb. 39: Szene 02 - Ankunft am Friedhof



Abb. 40: Szene 03 - Rückblende: Tod des Baters



Abb. 41: Szene 04 - Aufbruch in das Abenteuer



Abb. 42: Szene 05 - Training: Schwertkampf



Abb. 43: Szene 06 - Geist stärken: Meditation



Abb. 44: Szene 07 - Traditionen: Tempel



Abb. 45: Szene 08 - Respekt: Wertschätzung zeigen



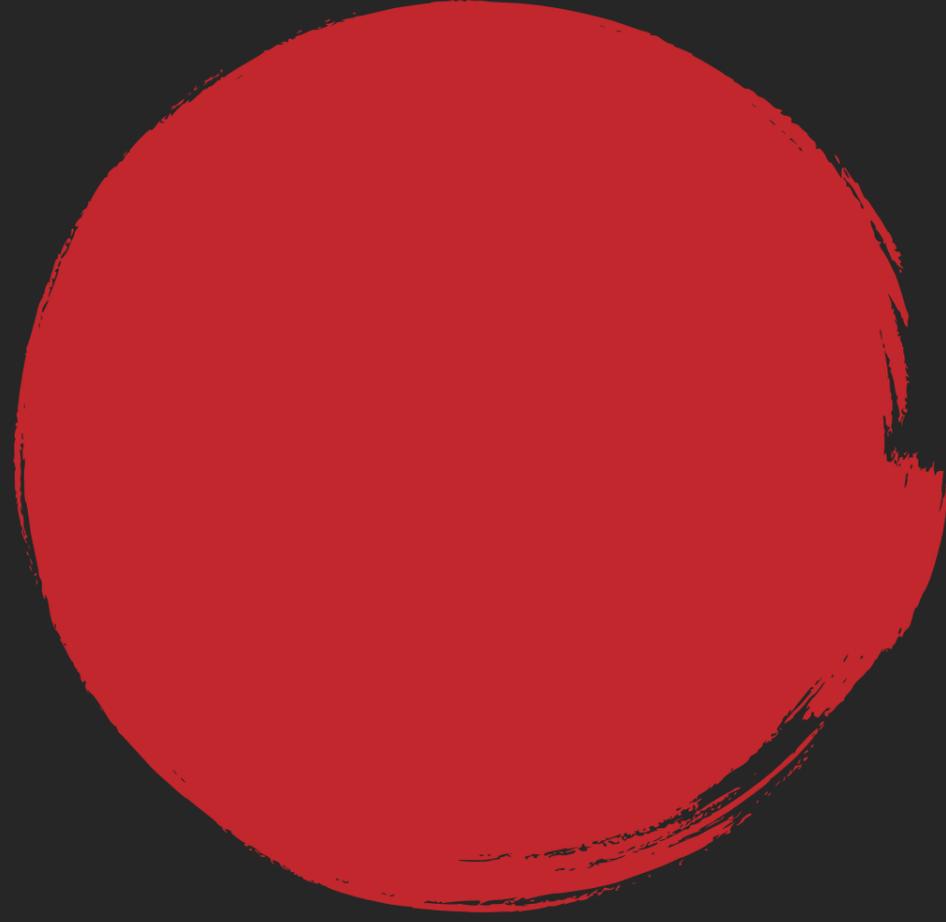
Abb. 46: Szene 09 - Ehre: Duellkampf



Abb. 47: Szene 10 - Mut: Stellt sich dem Gegner

UMSETZUNG

04



4.1 ENVIRONMENT | 4.1.1 GEOMETRIE

3D-MODELING

Die Erstellung von Geometrie, welche virtuelle, dreidimensionale Körper darstellt, erfordert den Prozess des 3D-Modelings. Auf diese Weise ist es möglich, jedes erdenkliche Objekt durch Verschieben, Hinzufügen und Manipulieren von Polygonen innerhalb eines Meshes in 3D zu visualisieren. In diesem Kontext findet zumeist der Begriff "Asset" Verwendung. Eine Szene kann demnach als eine Sammlung unterschiedlicher Assets definiert werden, die im dreidimensionalen Raum verteilt sind. [50]

Während des Modellierung-Prozess erfolgt der Aufbau eines Meshes mit Quads, wobei eine Fläche (Face) durch vier Eckpunkte (Vertices) definiert wird. [51] Für die Nutzung innerhalb von Realtime Engines ist jedoch die

Verwendung von Dreiecken (Tris) erforderlich. Dazu ist eine Triangulierung des Meshes erforderlich. Dabei werden Quads in zwei Tris und Ngons (n-Eckpunkte) in eine entsprechende Anzahl von Dreiecken unterteilt. Der zuvor beschriebene Prozess wird beim Importieren in die Unreal Engine jedoch automatisch übernommen. [52]

Für die Nutzung in Echtzeit ist es von entscheidender Bedeutung, dass die Meshes von 3D-Modellen möglichst effizient erstellt werden. Dabei sind eine niedrige Polygonanzahl, eine saubere Topologie, die richtige Skalierung, ein sinnvoller Pivot-Punkt und eine passende Ausrichtung der Normalen wesentliche Aspekte, die ein hochwertiges 3D-Modell auszeichnen.

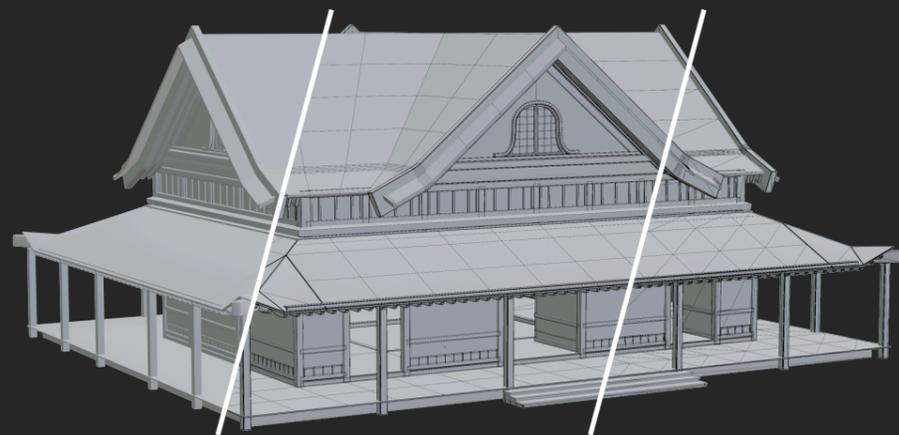


Abb. 48: Beispiel für die Geometrie eines Hauses | Links = Solid / Mitte = Quads / Rechts = Tris

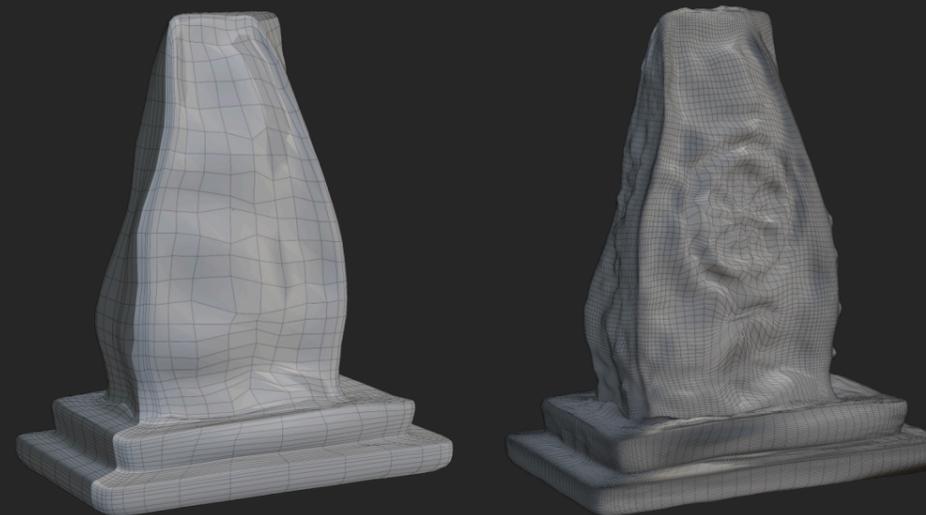


Abb. 49: Beispiel für das Sculpting eines Grabsteins | Links = Low Poly / Rechts = High Poly

SCULPTING

Das Sculpting kann mit dem Formen von physischem Ton und Lehm verglichen werden. Die Verformung des 3D-Mesh erfolgt durch den Einsatz verschiedener Pinsel und Werkzeuge. Dabei lässt sich "Auftragen" (Add), "Glätten" (Smooth), "Kratzen" (Scrape), "Ziehen" (Pull) und "Schieben" (Push) die gewünschte Form erzeugen. Dadurch können sehr detaillierte, komplexe und organische Formen erstellt werden, die mit dem 3D-Modeling schwierig zu realisieren wären. Der Vorteil dieser Methode besteht darin, schnell und einfach, ohne auf Topologie achten zu müssen, Geometrie zu erzeugen und den künstlerischen Aspekt in den Fokus zu rücken. Zusätzlich kann das Sculpting dazu verwendet werden, um Meshes zu überarbeiten, ohne dabei die Komplexität des Meshes zu verändern.

Aufgrund der hohen Rechenleistung, die für die Erstellung von

hochauflösender Geometrie (High-Poly) durch das Sculpting erforderlich ist, wird in der Regel zunächst eine reduzierte Version des Meshes (Low-Poly) erstellt. Der Prozess zur Erstellung eines Low-Poly-Modells wird als Retopology bezeichnet. Dabei kann ein manueller oder automatischer Vorgang gewählt werden. [53][54]

Im Fall dieser Arbeit wurde das Sculpting in erster Linie zur Überarbeitung dynamischer Objekte wie Kleidung verwendet. Dadurch wurden Unstimmigkeiten im Mesh bereinigt, sowie bessere Ergebnisse und Performance in Realtime-Prozessen, wie Cloth-Simulations erzielt. Bei statischen Objekten wurde der Retopology-Prozess sowie weitere Anpassungen am Mesh übersprungen, da Nanite innerhalb der Unreal Engine diese Aufgabe automatisch übernimmt.

UV-UNWRAPPING

Das UV-Unwrapping bezeichnet den Prozess, bei dem ein 3D-Mesh auf eine zweidimensionale Fläche, die sogenannte UV-Map, projiziert wird. Dadurch wird es möglich, Texturen auf ein Objekt anzuwenden. Dieser Vorgang lässt sich am Besten mit dem Auseinanderfalten eines Würfels vergleichen. Der Begriff "UV" steht in diesem Kontext für die Achsen des Texturkoordinatensystems, welche traditionell als "U" und "V" bezeichnet werden, um sie von den "XYZ"-Achsen des 3D-Raums zu unterscheiden. [55]

Für primitive Objekte findet in der Regel eine Projektion mittels einfacher Methoden statt, wobei hier die Cube-, Cylinder- und Sphere-Projection zu nennen sind. Diese Projektionsmethoden ermöglichen eine mathematische Lösung des Unwrappings.

Bei komplexen Objekten erfolgt eine Unterteilung der Geometrie in einzelne Flächen mithilfe von Nähten (Seams), welche anschließend als UV-Map unwrapped werden.

Ein wesentlicher Aspekt dieses Prozesses ist die sogenannte Texeldichte. Sie beschreibt das Verhältnis zwischen der Auflösung der Textur und dem 3D-Objekt im Raum.

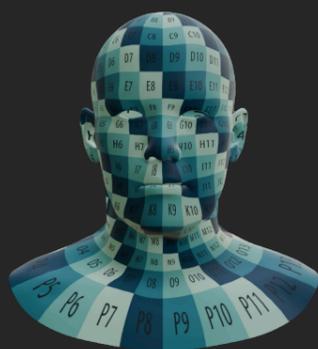


Abb. 51: UV-Mapping des Gesichts eines MetaHumans mit einem Checker-Pattern

Eine niedrige Texeldichte führt zu einer unscharfen Wirkung der angewandten Textur, während eine hohe Texeldichte eine scharfe Wirkung erzeugt. Eine unregelmäßige Texeldichte hingegen führt zu einer inkonsistenten Auflösung, was die Glaubwürdigkeit des Ergebnisses beeinträchtigt. [56]

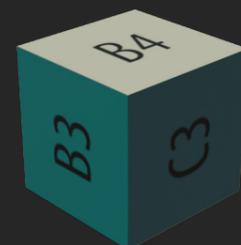
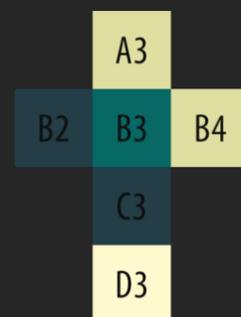


Abb. 50: UV's eines Würfels anhand eines Checker-Pattern

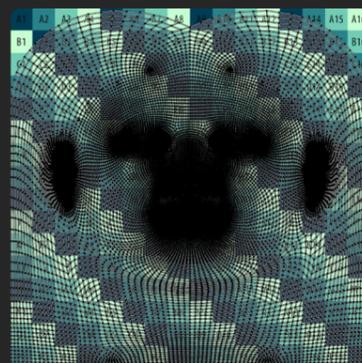
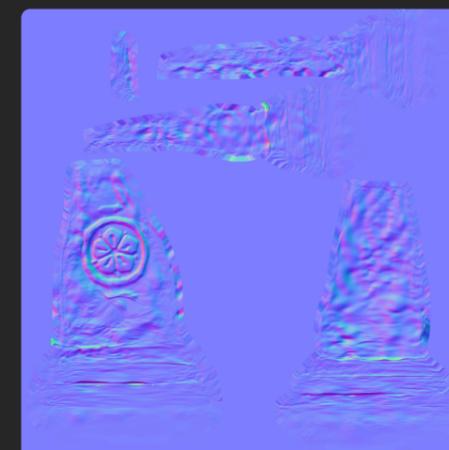


Abb. 52: Normal Map Baking am Beispiel eines Grabsteins



BAKING

Unter Baking wird die Speicherung von essenziellen Informationen eines 3D-Modells in diversen Texturen verstanden. Das Ziel besteht darin, durch diese Daten eine Erweiterung der Möglichkeiten innerhalb des Texturierungs- und Shading-Prozesses zu erhalten sowie komplexe Berechnungen in Echtzeit zu verhindern. Dabei können auch die Normal-Informationen eines hochauflösenden Meshes (High-Poly) auf eine niedrig aufgelöste Variante (Low-Poly) des gleichen Modells übertragen werden. Dieser Vorgang wird auch als Normal Map Baking oder Detail Baking bezeichnet.

Die Software Substance Painter unterstützt das Baking und kann beispielsweise folgende Informationen in Texturen speichern:

- Ambient Occlusion (lokale Umgebungsschatten)
- Normal (Oberflächendetails)
- Direction (Richtungen in die die Oberfläche zeigt)
- Curvature (Kanten und Vertiefungen)
- Position (Relative Position innerhalb eines normalisierten Würfels)

So können beispielsweise mit der Curvature-Map, die Kanten eines Metallwürfels abgenutzt erscheinen oder mit der Ambient Occlusion Map, Moos aus den Ritzen eines verwucherten Autos wachsen. [57]

TEXTURING

Nach der Erstellung des 3D-Objekts fehlt es der Oberfläche noch an Farbe und Struktur. Im Texturing geht es somit darum, die Geometrie mit ausgewählten Materialien zu versehen, wodurch das Erscheinungsbild des Objektes bestimmt wird. Dabei bestehen zwei Möglichkeiten, das Texturing auf prozeduraler oder texturbasierter Ebene.

Prozedurale Materialien werden mithilfe von mathematischen Algorithmen meist in Form von Nodes erzeugt, wodurch unbegrenztes Detail bezüglich der Auflösung ermöglicht wird. Zusätzlich besteht die Möglichkeit, die Eigenschaften durch Parameter zu modifizieren und somit eine individuelle Anpassung des jeweiligen Objektes vorzunehmen. Dies erweist sich insbesondere bei der Anwendung wiederkehrender Muster als vorteilhaft, wie Holz, Metall, Stein und Marmor. Ein wesentlicher Vorteil

besteht darin, dass UVs nicht zwingend erforderlich sind, sodass prozedurale Texturen auf jedes Objekt anwendbar sind.

Im Falle der texturbasierten Methode erfolgt die Zuweisung von Grafiken mit unterschiedlichen Materialeigenschaften (Albedo, Roughness, Normal, usw.) mittels der UV-Map. Dabei kann durch Texturpainting auf die Oberfläche des Objektes gezeichnet werden, um Details gezielt zu setzen. Damit ist diese Methode deutlich präziser, realistischer und performanter als die prozedurale Variante. Bei den importierten Bildern ist es wichtig zu beachten, dass diese wiederholbar (seamless) sind. Falls dies nicht gegeben ist, kommt es zu einem abrupten Übergang zwischen dem Ende und dem Anfang des Bildes, was eine unschöne Kante im Material zur Folge hat. [58]



Abb. 53: Texturierter Grabstein



Abb. 54: Texture 1 - Albedo



Abb. 55: Texture 2 - Packed



Abb. 56: Texture 3 - Normal

Zusammenhängend mit der texturbasierten Methode ist die Verwendung von Substance Painter zu erwähnen. Hierbei wurden bestimmte Hero-Assets texturiert, die spezielle Details durch Texturpainting und Generatoren (Abschnitt "Baking") aufweisen. Da der Vorgang spezifisch für ein 3D-Modell erfolgt, ist eine Nutzung der exportierten Texturen ausschließlich für dieses Objekt möglich. Substance Painter bietet darüber hinaus mit dem Channel Packing die Option, mehrere Materialeigenschaften in die einzelnen RGB(A)-Kanäle einer Textur zu hinterlegen. Dies führt zu einer deutlich besseren Speicherauslastung.

Für den Export der Texturen wurde folgende Form gewählt:

- Textur 1: Albedo (RGB) + Opacity (A)
- Textur 2: Ambient Occlusion (R) + Roughness (G) + Metallic (B)
- Textur 3: Normal (RGB)
- Textur 4: Emissive (RGB) | optional

Für das Texturing der vorliegenden Arbeit wurde das Texturing hauptsächlich mit texturbasierten Materialien umgesetzt. Dabei wurde stets der performanteste Ansatz gewählt, um eine möglichst effiziente Nutzung der Ressourcen innerhalb der Unreal Engine zu gewährleisten. [59]

MATERIALS

Im 3D-Kontext stellen Materialien eine Sammlung von Attributen und Eigenschaften dar, die das Aussehen und Verhalten einer Oberfläche definieren. Diese ähneln den aus der realen Welt bekannten Äquivalenten. Zu den Materialeigenschaften zählen unter anderem Farbe (Base Color), Spiegelung (Specular), Metallizität (Metalness), Rauheit (Roughness), Leuchtkraft (Emissive) und Transparenz (Opacity). Im Wesentlichen wird hierbei definiert, wie sich ein Material bei Licht- und Schatteneinwirkung verhält. In diesem Kontext wird zudem häufig der Begriff "Shading" verwendet. Da jedoch in Bezug auf die Unreal Engine zumeist der Begriff "Material" verwendet wird, wird in dieser Arbeit stets dieser Begriff verwendet.

In der Unreal Engine findet das sogenannte "Parent-Child"-Prinzip Anwendung auf die Materialien. Ein Master-Material fungiert in diesem Kontext als Parent, wobei das Node-Setup, also die Logik, integraler Bestandteil davon ist. Die daraus erstellten Instanzen, welche die Childs des Parents darstellen, steuern unabhängig voneinander die globalen Werte des Masters. Die Instanzen können folglich auf jedes Objekt angewendet werden. Zudem teilen

sich Instanzen desselben Master-Materials dieselben Ressourcen, was zu einer Verbesserung der Performance beiträgt. [60]

Für Rise of a Samurai wurden somit wenige verschiedene Master-Materials angelegt, welche sämtlichen benötigten Einsatzzwecke abdecken und als Instanzen auf die Assets vervielfältigt wurden. Der Aufbau eines solchen Materials lässt sich exemplarisch in verschiedene Schritte unterteilen.



Abb. 57: Schritt 1 - Grundlegende Textureslots

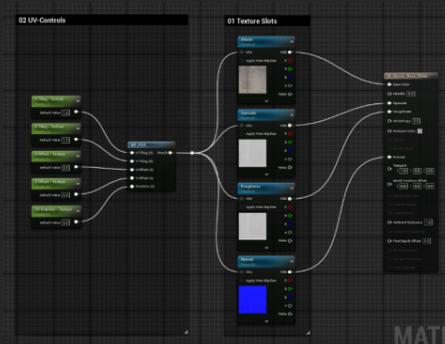


Abb. 58: Schritt 2 - UV-Einstellungen (Tiling, Rotation, Offset)



Abb. 59: Schritt 1 & 2 - Ergebnis

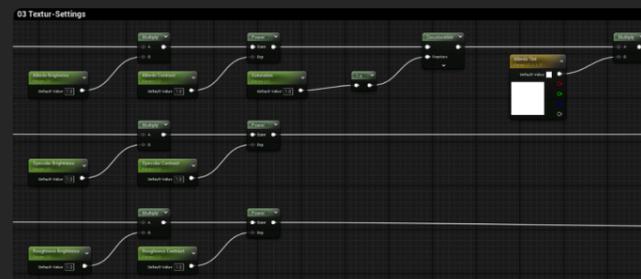


Abb. 60: Schritt 3 - Anpassungen der Textureigenschaften

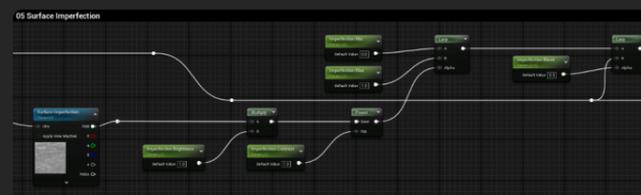
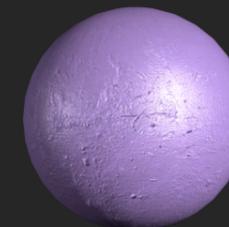


Abb. 61: Schritt 4 - Surface Imperfections

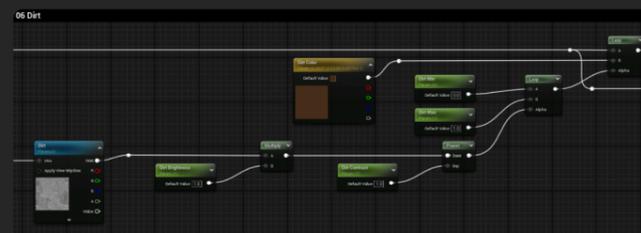
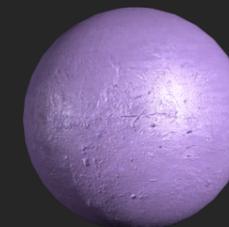
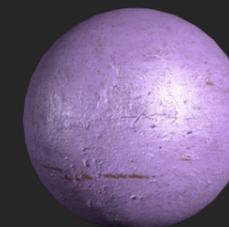


Abb. 62: Schritt 5 - Verschmutzungen (Dirt-Layer)



MEGASCANS



Abb. 63: Megascans in einer Szene

Aufgrund des hohen Arbeitsaufwands in anderen Bereichen der Umsetzung war es nicht möglich, alle benötigten Assets und Materialien selbst zu erstellen. Daher wurden Megascans, eine umfangreiche Bibliothek von hochauflösenden, fotorealistischen 3D-Scans und Texturen verwendet. Die Bibliothek von Quixel weist dabei eine nahtlose Integration innerhalb der Engine auf und benötigt keinerlei zusätzliches Setup nach dem Import. Zudem stellten die Objekte eine gute Basis für neue Assets und Optimierungen dar.

4.1.2 LANDSCAPE

TERRAIN

Unter einem Terrain versteht man die digitale Darstellung einer Geländeoberfläche.

Die topografischen Merkmale umfassen dabei natürliche Landformen wie Hügel, Berge, Täler, Ebenen, Klippen, Flüsse und Seen dar. Die Erzeugung der Höhenunterschiede kann auf zwei verschiedene Arten erfolgen: entweder mithilfe einer originalen Höhenkarte, durch prozedurales Vorgehen oder durch manuelles Modellieren.

Für die prozedurale Erstellung wird dafür häufig spezielle Software wie Gaea, World Creator oder GeoGen verwendet. Die Generierung des Terrains erfolgt mittels Erosion, Noise, Deformation und weiterer ähnlicher Effekte. Das fertige Gelände wird in der Regel entweder als Mesh oder Height-Map zur Verwendung in anderen Softwareanwendungen bereitgestellt.

Im Rahmen der Projektarbeit wurden mit Gaea verschiedene Berge als Mesh erstellt, um den Hintergrund der Szenen zu füllen. Dies sollte auch für das Terrain im Vordergrund erfolgen. Aufgrund zeitlicher Einschränkungen mussten diese jedoch mit den Landscape-Tools der Unreal Engine modelliert werden.



GAEA

Abb. 64: Logo von Gaea

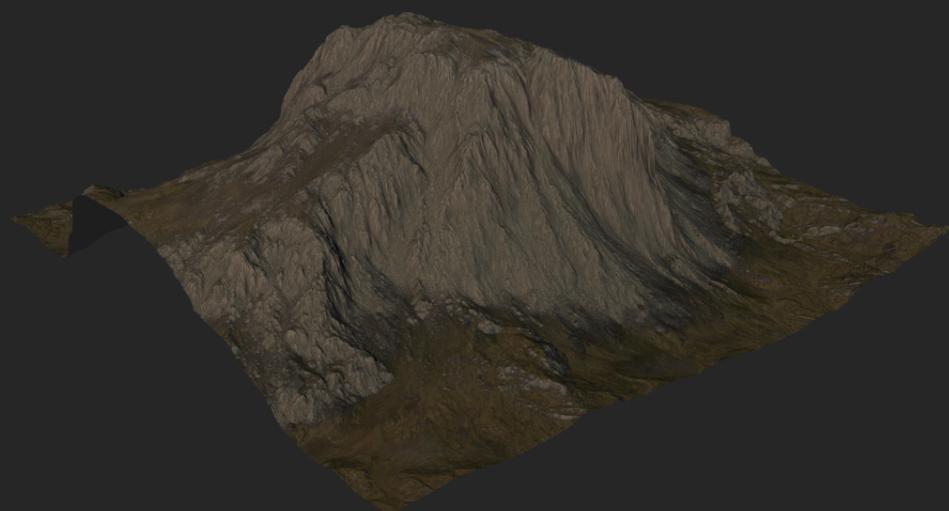


Abb. 65: Beispiel eines mit Gaea erstellten Terrains



Abb. 66: Rendering eines Terrains mit einem Auto-Landscape-Material

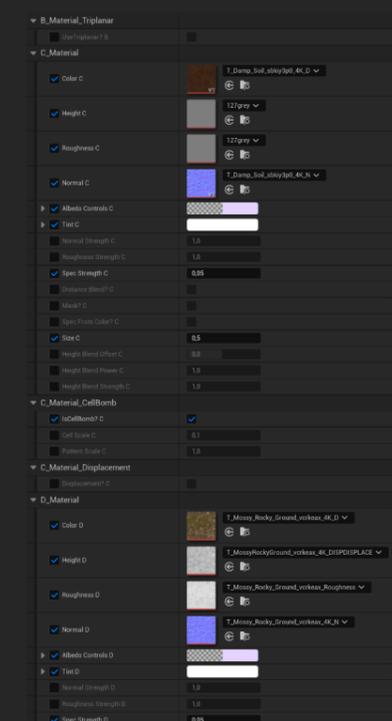


Abb. 67: Einstellungen des Auto-Landscape-Materials

LANDSCAPE-AUTO-MATERIAL

Für das Shading des erstellten Terrains wurde ein Landscape-Auto-Material entwickelt. Dieses verteilt anhand bestimmter Bedingungen wie Höhe und Neigung des Geländes automatisch unterschiedliche Materialien. Dadurch wird beispielsweise flachem Terrain Gras, steilen Hängen Stein und hohen Gipfeln Schnee zugewiesen. Die Materialien lassen sich zudem durch unterschiedliche Masken einzeichnen. Da das Material ziemlich umfangreich ist, wird in den folgenden Abschnitten die Funktionsweise anhand der verwendeten Techniken erläutert.



Abb. 68

LAYERING

Die zuvor beschriebene automatische und manuelle Vergabe der Materialien wird durch die Erstellung verschiedener Material-Layer ermöglicht, welche unterschiedliche Texturen enthalten. Die Zusammenführung der Layer-Masken erfolgte mittels Landscape-Blending-Nodes, um saubere Übergänge zwischen den Materialien zu schaffen. Die Eigenschaften aller Materialien lassen sich dynamisch über eine Landscape-Material-Instance steuern. Die Einstellungen umfassen dabei Texturanpassungen, UV-Transformationen, Displacement- und Blendingoptionen. [61]

TRIPLANAR PROJECTION MAPPING

Das Triplanar Projection Mapping ist eine Methode, um Objekte ohne UV's oder die schwer zu UV-Unwrappen sind, mit Texturen zu versehen. Dabei wird eine Textur aus drei orthogonalen Richtungen (X-, Y- und Z-Achse) auf ein Objekt projiziert und in einem Ergebnis zusammengefasst. Dadurch können Materialien entwickelt werden, die auf jedes Objekt angewendet werden können, ohne dabei Verzerrungen aufzuweisen. Dies war eine notwendige Vorgehensweise für das Landscape-Material, da eine UV-Map eines in Unreal modellierten Landscapes eine Projektion von oben darstellt, welche starke Neigungen nicht korrekt abdecken kann und somit als verzerrt dargestellt wird. [62]

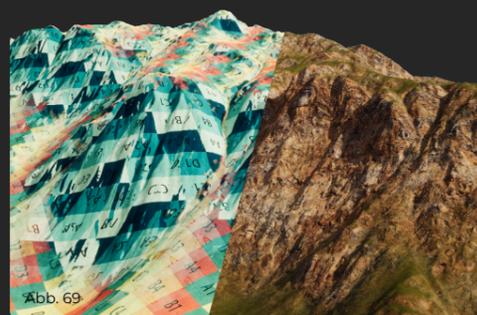


Abb. 69

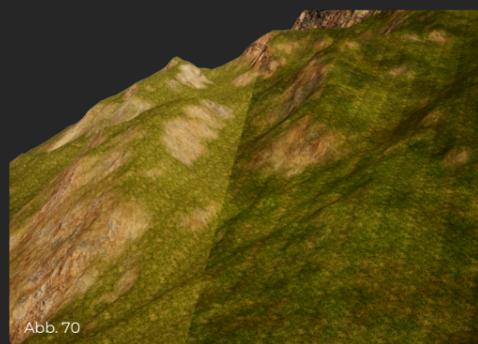


Abb. 70

MICRO VARIATIONS

Mit Micro Variations wurden Unregelmäßigkeiten den Materialeigenschaften (Farbe und Glanz) hinzugefügt. Ihre Erstellung erfolgte durch Multiplikation verschiedener Noises mit unterschiedlichem Tiling. Die dadurch erzielte, geringfügige und subtile Veränderung der Oberfläche des Landscapes hatte eine positive Auswirkung auf den Abwechslungsreichtum und die Natürlichkeit des Materials.

TEXTURE BOMBING

Die Idee hinter dem Texture Bombing besteht in der Unterteilung der UVs eines Meshes in ein Gitter von Zellen, welches zufällig an verschiedenen Positionen der Oberfläche platziert wird. Jede Zelle beinhaltet dabei einen unterschiedlichen Bereich einer ausgewählten Textur. Die zufällige Verteilung der Bereiche verhindert die Bildung eines erkennbaren, wiederholenden Musters durch das Tiling der Textur. Zusätzlich wurden überlappende Zellen geblendet, um Übergänge zwischen den einzelnen "Bombs" zu schaffen. Diese Technik wurde beispielsweise in der "Matrix Awakens Demo" von Epic Games eingesetzt, um Tiling-Repetition für die Straßen zu vermeiden. [63]

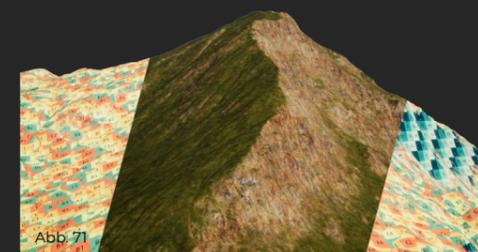


Abb. 71

RUNTIME VIRTUAL TEXTURING

Mit dem Runtime Virtual Texturing (RVT) lassen sich komplexe Materialien auf großen Flächen, wie einem Landscape, performant darstellen. Die Texturen in ihren benötigten Auflösungen werden dabei erst bei Bedarf aus dem Speicher (HDD/SDD) geladen, was sich positiv auf die Speicherauslastung auswirkt. In diesem Zusammenhang wird von MipMap-Leveln gesprochen, die unterschiedliche Auflösungsgrößen einer originalen Textur darstellen.

Höhendaten sowie der Materialeigenschaften des Landscapes erfolgt mithilfe zweier RVT-Volumes. Im Anschluss werden die Daten in unterschiedlichen Runtime Virtual Textures geschrieben. Dafür wird im Landscape-Material bestimmt, welche Eigenschaften gespeichert und ausgegeben werden. Die Materialien der Objekte greifen auf diese Daten zu, wodurch das Blending ermöglicht wird. Dieses Vorgehen wird auch als RVT-Blending bezeichnet. [64][65]

Ein solcher komplexer Anwendungsfall stellte das Blending zwischen der Szenen Geometrie und dem Landscape dar. Die Erfassung der



Abb. 72: Blending zwischen Landscape und Szenengeometrie mittels RVT | Links ohne - Rechts mit

4.1.3 FOLIAGE

KONTEXT

Unter Foliage lässt sich die digitale Darstellung von Vegetation in Form von Pflanzen, Bäumen, Sträuchern, Gräsern, Blumen und Moos bezeichnen. Die Erstellung einer glaubwürdigen, digitalen Abbildung der Natur, insbesondere per Hand, erweist sich als schwierig.

Dabei lassen sich viele Formen und Strukturen der Natur durch mathematische Prinzipien abbilden. Beispiele dafür sind unter anderem Fraktale, der goldene Schnitt, die

Fibonacci-Folge sowie die Mandelbrot-Menge. Dieses Wissen kann folglich genutzt werden, um Foliage generativ zu erzeugen. Dies kann entweder durch ein eigenes System oder durch speziell dafür entwickelte Software, wie etwa Speedtree oder Treelt, erfolgen. Im Rahmen dieser Bachelorarbeit wurde auf den Industriestandard in der Film- und Spielebranche, Speedtree zurückgegriffen. [66]

SPEEDTREE

Speedtree stellt eine node-basierte Software zur Erstellung von virtuellen Bäumen und Pflanzen dar. Hierbei sind Bäume das Hauptaugenmerk des Programms. Durch prozedurale Algorithmen werden komplexe Baum- und Pflanzenstrukturen effizient und einfach erstellt. Die Steuerung des Aussehens erfolgt generativ durch verschiedene Parameter und Modi, wodurch eine Vielzahl an Pflanzen- und Baumarten erzeugt werden können. Die grundlegende Struktur eines Node-Trees in Speedtree lässt sich auf eine Vielzahl an verschiedenen Foliage-Typen anwenden.

Der Aufbau eines Baumes kann wie folgt dargestellt werden:



Abb. 74: Aufbau eines Baumes | Links = Stamm & Wurzeln / Mitte = Äste & Zweige / Rechts = Blätter



Abb. 73: Node-Struktur eines Baumes

Die Software Speedtree verfügt über drei verschiedene Modi, welche in Generator, Node und Freehand unterteilt sind. Jeder dieser Modi kann pro Node separat genutzt werden und weist ähnliche Einstellungsmöglichkeiten auf.

Im Generator-Modus wird durch globale Eigenschaften in Form von Parametern die Erzeugung der Foliage gesteuert. Bedeutet, wenn eine Node beispielweise mehrere Äste beinhaltet, sind alle Äste von diesen Einstellungen betroffen.

Der Node-Modus weist eine gewisse Ähnlichkeit bezüglich der Parameter des Generator auf. Dies ermöglicht es, wie im vorigen Beispiel erläutert, jeden Ast separat generativ zu beeinflussen.

Die Erstellung der Strukturen einer Node erfolgt in Freehand mithilfe von Splines. Dadurch ist es möglich, beispielsweise Äste manuell zu positionieren und nach Bedarf zu manipulieren.

Alle zu einstellenden Eigenschaften sind in verschiedene Menüpunkte unterteilt. Je nach Nodetyp gibt es mehr oder weniger davon. Im Folgenden werden die wichtigsten Punkte beschrieben, die im Laufe des Projektes von Relevanz waren.

Gen: Die Generierungseinstellungen legen die Anzahl und Verteilung der zu generierenden Subnodes (Äste, Blätter usw.) fest. Zudem kann ein Bereich definiert werden, in dem die Subnodes erzeugt werden dürfen.

Spine: Legt die Länge und Form eines Meshes fest. Eigenschaften wie Krümmungen (Twist), Verzerrungen (ZigZag, Noise) und Orientierung (Start angle, Gravity) bestimmen das Erscheinungsbild.

Skin: Bestimmt die Dicke des Meshes, sowie die Übergänge zwischen den Nodes. Mithilfe von einstellbaren Graphen pro Eigenschaft, kann zudem die Breite über die Länge des Meshes dynamisch festgelegt werden.

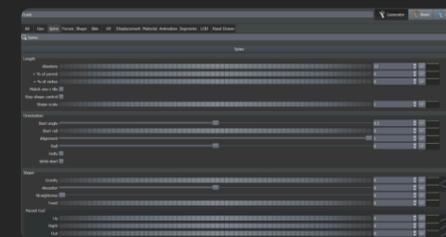


Abb. 75: Speedtree Einstellungsmöglichkeiten



Abb. 76: Beispiel eines erstellten Hero-Trees

BLÄTTER

Für die Erstellung des Blattwerkes kommen im Rahmen der vorliegenden Bachelorarbeit Atlanten der Megascan-Bibliothek zum Einsatz. Die Atlanten stellen realistische Abbildungen von Blättern, Zweigen oder Pflanzen dar, die mittels Photogrammetry eingescannt wurden. Durch Cutouts, die die grobe Form eines Blattes definieren, werden die Detailstufen eines Meshes (LOD's), sowie der Ursprung festgelegt. Im Anschluss werden die einzelnen Cut-outs auf den Baum übertragen, wodurch das Blattwerk dargestellt wird.



Abb. 77: Blätter-Atlas des zuvorigen Baumes (ungefärbt)

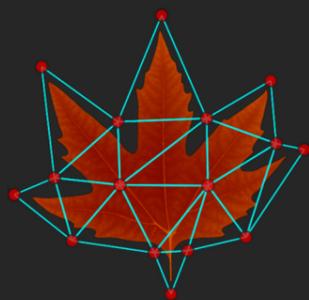


Abb. 78: Mesh-Cutout

GRAS UND PFLANZEN

Die Verwendung von Speedtree beschränkt sich im Animationsfilm auf die Erzeugung verschiedener Bäume. Aufgrund des Zeitdrucks und anderer Priorisierungen wurden größtenteils die Pflanzen und Gräser aus fertigen Asset-Packs und den Megascans entnommen. Dabei wurde darauf geachtet, dass die externe Foliage eine ähnliche Qualität aufweist wie die Bäume.



Abb. 79: Animiertes Gras-Material

WINDBEWEGUNGEN

Ein weiterer wesentlicher Aspekt, der für die Entscheidung zugunsten von Speedtree ausschlaggebend war, ist die Pipeline nach Unreal. Hierbei werden nicht nur das Mesh selbst, sondern auch animierte Materialien beim Import automatisch erstellt. Der animierte Aspekt stellt ein globales Windsystem dar, welches die voreingestellten Windbewegungen einer Foliage auf alle anderen überträgt. Dadurch wirkt die erstellte Foliage deutlich dynamischer, was zu einer höheren Immersion beiträgt.

WORLD-POSITION-OFFSET

Die Generierung der gesamten Windbewegung erfolgt durch die Manipulation des World Position Offset (WPO). Hierbei lassen sich die Positionen der Vertexpunkte einer Geometrie dynamisch im Raum bewegen. Dadurch ist es möglich, natürliche Bewegungen mittels mathematischer Funktionen zu simulieren. In diesem Zusammenhang ermöglicht das WPO auch die Einbindung von Simulationen und Animationen aus externer Software in Form von Vertex Animation Textures (VAT). Diese stellen eine deutlich performantere Variante der Alembics dar. [67][68]

4.1.4 PCG



Abb. 80: Electric Dreams Demo

Das Procedural Content Generation Framework (PCG) stellt ein Toolset innerhalb der Unreal Engine dar, welches die Erstellung von prozeduralen Inhalten ermöglicht. PCG ist dabei als ein nodebasiertes Scattering Tool einzuordnen. Die Verteilung von Assets und Foliage erfolgt in Form von Punkten, wobei in der Regel ein Landscape als Basis zugrunde liegt. Dadurch lassen sich Biome wie Wälder oder Graslandschaften zeit- und ressourceneffizient erzeugen. Die

Electric Dreams Demo von Epic Games demonstriert die vielfältigen Anwendungsmöglichkeiten von PCG. Im Rahmen der Produktion des Animationsfilms wurde PCG insbesondere zur Erstellung komplexer Waldszenen sowie zur einfachen Verteilung von Steinen und Ästen eingesetzt. Die nachfolgende Darstellung des Scattering-Prozesses mittels PCG basiert auf der Generierung eines Waldes. [69]

GENERIERUNG EINES WALDES

1. Im ersten Schritt erfolgt die Verteilung einzelner Punkte auf der Oberfläche des Landscapes mittels eines Surface-Samplers. Dabei wird für jeden Punkt ein zufälliger Density-Wert zwischen 0 und 1 generiert.



Abb. 81



Abb. 82

2. Danach wird die Anzahl der Punkte mithilfe eines Density Filters reduziert. Im vorliegenden Beispiel werden alle Punkte mit einem Wert unter 0,95 weg.

3. Um das Wertespektrum von 0 bis 1 wiederherzustellen, erfolgt eine Neuzuweisung der Density-Werte für alle Punkte durch eine Attribute Noise Node.



Abb. 83



Abb. 84

4. Damit die verteilten Punkte aufgelockert und dynamischer wirken, werden mithilfe der Transform Points Node die Transformation (Location, Rotation, Scale) aller Punkte zufällig verändert.

5. Durch den Einsatz eines Static Mesh Spawners erfolgt ein Austausch der zuvor festgelegten Punkte durch Bäume.



Abb. 85



Abb. 86

6. Dem Wald fehlt es nun an begehbaren Wegen. Mit einer Kombination aus einem Spline Sampler und Bounds Modifier wird die Breite des Weges durch eine Spline definiert. Um die gescatterten Punkte innerhalb des Weges zu entfernen, wird eine Difference Node verwendet.

7. Da sich außerhalb eines Baumes oft Gewächse bilden oder Steine zu finden sind, werden die bereits erzeugten Bäume als Basis für die neuen Punkte verwendet. Mit Create Points Grid wird zunächst ein zweidimensionales Raster aus Punkten erzeugt. Mit Copy Points werden die Eigenschaften der Ausgangspunkte auf das Raster übertragen.



Abb. 87



Abb. 88

8. Durch die Reduzierung der Punkte und etwas Transformationen wird das erzeugte Raster aufgebrochen und wirkt nun deutlich glaubwürdiger. Damit sind die finalen Punkte für das Gewächs erstellt, die nun mit dem Static Mesh Spawner platziert werden.



Abb. 89

9. Im letzten Schritt des PCC-Graphen wird der Wald mit einer Grasfläche überzogen. Dazu werden mit einem Surface Sampler neue Punkte auf der Landschaft erzeugt. Anschließend werden diejenigen Punkte entfernt, welche die Wege überschneiden. Zuletzt werden die Punkte mit einem Density-Filter reduziert und mit Transform Points manipuliert.

10. In einem optionalen Schritt besteht die Möglichkeit, wesentliche Werte des PCC-Graphen durch Blueprint-Variablen für den Viewport zugänglich zu machen. Dazu ist die Get Actor Property Node erforderlich. Innerhalb der Node muss der Variablenname exakt mit dem Äquivalent im Blueprint übereinstimmen.

Point Density	10,0		
▶ Point Size	10,0	10,0	10,0
▶ Scale Min	0,5	0,5	0,5
▶ Scale Max	1,0	1,0	1,0
▶ Rotation Min	0,0	0,0	-180,0
▶ Rotation Max	0,0	0,0	180,0
▶ Offset Min	-100,0	-100,0	0,0
▶ Offset Max	100,0	100,0	0,0
Seed	439701		Abb. 90



Abb. 91: Einsatz eines komplexen PCC-Systems

4.1.5 NANITE

NANITE

Nanite stellt ein virtualisiertes Geometriesystem dar, welches mit der Unreal Engine 5 eingeführt wurde. Die Technologie ermöglicht die Echtzeit-Darstellung von extrem detaillierter Geometrie mit Millionen von Polygonen. Durch automatische Optimierungsmethoden der Szenengeometrie wird dynamisch der benötigte Detailgrad der Meshes festgelegt. Während des Importvorgangs wird eine Analyse des Meshes durchgeführt, woraufhin eine Unterteilung in hierarchische Cluster (Gruppen) erfolgt. Im Rendering wird anhand der Kameraperspektive zwischen den Detailstufen der Cluster dynamisch gewechselt. Dadurch wird der Bedarf an manuell erstellten Level

of Detail (LOD's) oder Normal Map Baking für statische Objekte obsolet. Zusätzlich ist Nanite dazu imstande, die Sichtbarkeit der Meshes zu erkennen, wodurch verdeckte Geometrie aus dem Rendering-Prozess ausgeschlossen werden kann. [70]

Da Nanite nur erhebliche Vorteile für die Performance einer komplexen Szene mit sich bringt und die künstlerischen Freiheiten durch das Wegfallen vieler technischer Limitationen stärkt, war der Einsatz dieser Technologie für die vorliegende Arbeit keiner langen Entscheidung bedürftig. [71]

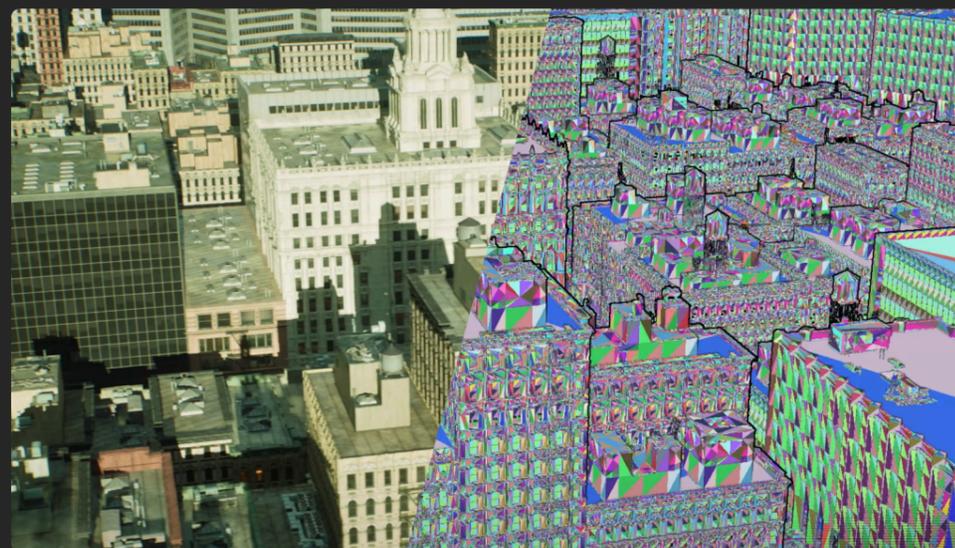


Abb. 92: Darstellung von Nanite-Szenengeometrie - The Matrix Awakens

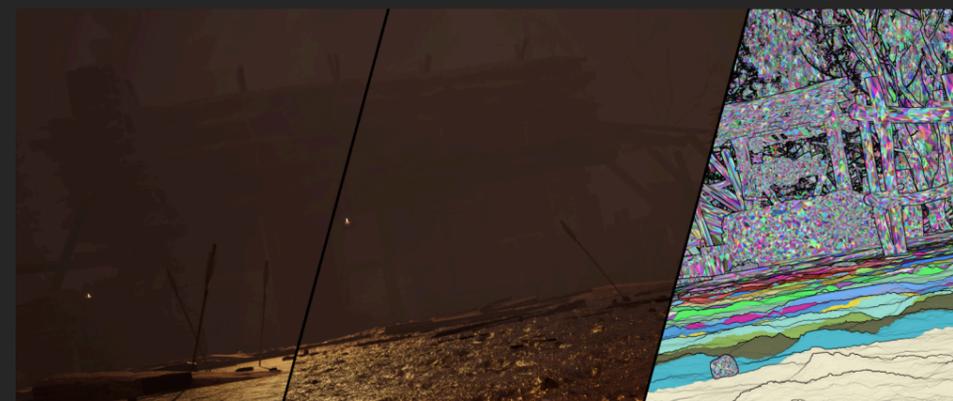


Abb. 93: Beispiel für Tessellation auf einem Landscape | Links = Aus / Mitte = An / Rechts = Nanite Visualisierung

TESSELLATION

Im Zuge der Weiterentwicklung von Nanite konnten auch die Einsatzmöglichkeiten erweitert werden. Diesbezüglich ist die Nanite-Tessellation zu erwähnen. Der Begriff Tessellation bezeichnet die Unterteilung von Polygonen in kleinere Polygone mit dem Ziel einer detaillierten Darstellung einer Oberfläche. Dies ermöglicht beispielsweise die Umwandlung eines Low-Poly-Mesh mithilfe einer Displacement-Map zu einem High-Poly-Mesh mit einer höheren Detailtreue und einer größeren Vielfalt an Strukturen. Im Falle der Nanite-Tessellation kann die Szenengeometrie durch die

Performancevorteile, die automatische Detailoptimierung und die einfache Anwendung stark von der realistischen Darstellung profitieren. Um Tessellation verwenden zu können, benötigt es ein Master-Material, welches die entsprechende Funktion aktiviert hat und Daten in Form einer Displacement Map, die es zu tessellieren gilt. In der verwendeten Unreal Engine 5.3 unterstützen bisher lediglich Static Meshes, Foliage und Landscapes diese Funktion. Für diese Bachelorarbeit erfolgte der Einsatz zunächst nur auf Landscapes. [72]

4.1.6 LIGHTING

LUMEN

Mit der Unreal Engine 5 wurde das neue globale Beleuchtungs- und Reflexionssystem Lumen veröffentlicht. Lumen ermöglicht es, in Echtzeit Einfluss auf die globale Umgebung zu nehmen, durch Lichtveränderungen, wie den Tageszeitenwechsel oder bewegliche Lichtquellen. Mittels der vieldynamischen indirekten Beleuchtung kann somit auf Änderungen von Lichteigenschaften der Geometrie und Materialien sofort eingewirkt werden. Dadurch sind keine Baking-Vorgänge von Lightmaps mehr nötig. Dafür wird

mehr Rechenleistung erfordert. Folglich werden leistungsstarke Rechner sowie Current-Gen-Konsolen (PS5, XBOX Series X/S) für die Verwendung von Lumen vorausgesetzt. [73]

Aufgrund der enormen Zeiteinsparungen und realistischen Ergebnisse wurde Lumen als die optimale Ausgangstechnologie für die Lichtsetzung des Animationsfilms gewählt.



Abb. 94: Lumen im Einsatz

LICHTSETZUNG

Für eine realistische Lichtsetzung wurden auf möglichst glaubwürdige und nachvollziehbare Lichtquellen gesetzt. Die Umgebungsbeleuchtung kann folglich aus einem Directional Light, einem Sky Light und einer Sky Atmosphäre zusammengesetzt

werden. Zusätzlich wird das Licht durch volumetrische Effekte, wie Nebel und Staub, sowie daraus resultierende God-Rays (volumetrische Lichtstrahlen) beeinflusst. Im Folgenden werden die im Animationsfilm genutzten Komponenten erläutert.



Abb. 95: Directional Light und Sky Light

DIRECTIONAL LIGHT

In der Unreal Engine fungiert ein Directional Light als Lichtquelle zur gleichmäßigen und direkten Beleuchtung einer Szene, wobei die Eigenschaften der Sonne oder eines großen Scheinwerfers imitiert werden.

Parallele Lichtstrahlen mit unbegrenzter Reichweite werden hierbei in eine bestimmte Richtung versendet. Basierend auf diesen Lichtstrahlen werden zudem dynamische Schatten erzeugt, die in Abhängigkeit von der Position und Intensität des Lichts sowohl weich als auch scharf sein können. Zusätzlich ermöglicht das Directional Light die Abbildung von Lichtstimmungen verschiedener Tageszeiten sowie statischer Farben. [74]

SKY LIGHT

Das Sky Light stellt eine Lichtquelle dar, welche das Umgebungslicht der Szene simuliert. Die Funktion des Sky Lights besteht in der Ergänzung eines gleichmäßigen, diffusen Lichteinfalls in die Szene. Dazu erfasst es die entfernten Bereiche eines Levels und bestimmt die Lichtverhältnisse. Auf diese Weise wird eine konsistente Darstellung des Himmels sowie der damit verbundenen Beleuchtung und Reflexion gewährleistet. Zudem führt dies zu einer Aufhellung der Schatten in der Szene, was insbesondere in der Außenumgebung zu einem realistischen Schattenwurf führt. [75]



Abb. 96: Sky Atmosphere und Volumetric Clouds

SKY ATMOSPHERE

Mit der Sky Atmosphäre lassen sich die Atmosphäre und ihre Lichtstreuung simulieren. Dies erlaubt die dynamische Darstellung unterschiedlicher Tageszeiten, Wetterbedingungen sowie weiterer atmosphärischer Phänomene. Die Streuung von Licht umfasst dabei die Rayleigh-Streuung, welche für die Färbung des Himmels verantwortlich ist, sowie die Mie-Streuung, welche die Dichte des Himmels bestimmt, also die Menge an Nebel, Staub oder Dunst, die im Himmel vorzufinden ist. [76]

VOLUMETRIC CLOUDS

Ermöglicht die Simulation von dreidimensionalen Wolken, welche unterschiedliche Himmels- und Wetterbedingungen darstellen können. Die volumetrischen Wolken können sich dynamisch bewegen, formen und auflösen sowie Einfluss auf die Lichtstreuung nehmen. Dadurch können Lichteffekte wie God Rays, Schatten und farbliche Veränderungen der Wolken erzeugt werden. [77]

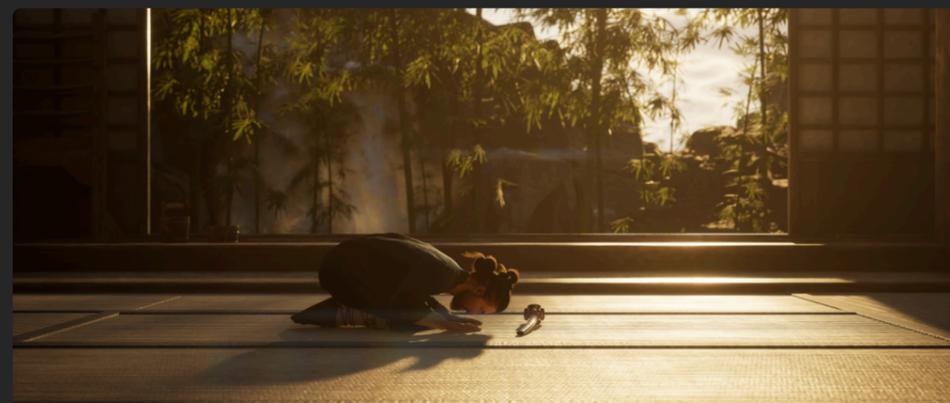


Abb. 97: Volumetric Fog | Oben = Aus / Unten = An

VOLUMETRIC FOG

Volumetric Fog ist die realistische Darstellung von Nebel im dreidimensionalen Raum. Dabei werden Lichtstrahlen auf natürliche Art und Weise vom Nebel gestreut, was zu realistischen Lichteffekten wie God Rays führt. Die Intensität und Farbe des Lichts beeinflussen das Erscheinungsbild des Nebels auf dynamische Weise. [78]

4.2 CHARACTER | 4.2.1 DESIGN

Der erste und grundlegende Schritt bei der Erstellung eines Charakters ist die Definition des Aussehens und des persönlichen Hintergrunds. Durch das Setting und die in der Konzeptionsphase geschriebene Geschichte lässt sich für alle Charaktere ein traditioneller japanischer Kleidungsstil zur Zeit des feudalen Japans ableiten. Zudem stammen alle abgebildeten japanischen Charaktere demselben Dorf ab. Deswegen wurde eine Ähnlichkeit zwischen den Kleidungsstücken, Farben und Muster hergestellt.

Die gewählten Kleidungsstücke umfassen dabei verschiedene Arten des Kimono (Oberkörper) und Hakama (Hose, Rock), sowie Waraji-Sandalen und Tabi-Socken. Zu der Zeit waren

diese Teile weit verbreitet und werden bis heute stark mit der japanischen Kultur assoziiert. Die Kleidung wurde in der Regel locker getragen und der jeweiligen Situation entsprechend angepasst, meist durch Zusammenbinden. [79]

Des Weiteren wurden für den Protagonisten Aoi Details in Form von Bandagen und einem Schwert konzipiert, um die Kampftauglichkeit als Samurai passend zu visualisieren. Zudem wurde die Kleidung so designt, dass sie auch für eine junge Version des Protagonisten verwendet werden kann.



Abb. 98: Traditionelle japanische Kleidung



Abb. 99: Teil der Ootuki Familie, 1874



Abb. 100



Abb. 101



Abb. 102



Abb. 103



Abb. 104

4.2.2 METAHUMANS

METAHUMANS

MetaHumans ist ein modernes Framework für die Erstellung und Animation von realistischen virtuellen Charakteren, die in jeder erdenklichen Weise mit den Möglichkeiten der Unreal Engine eingesetzt werden können. Die Kernfunktionen umfassen einen Charaktereditor, das Verwandeln eines realen oder digitalen Gesichts in einen MetaHuman, sowie das Facial Motion Capturing. Darüber hinaus gewährleistet die nahtlose Integration in die Unreal Engine die Verfügbarkeit aller für die Charakter- und Animationspipeline erforderlichen Elemente, um Rise of a Samurai zu realisieren.



Abb. 105: Aoi als Jugendlicher (Oben) und Erwachsener (Unten)

METAHUMAN CREATOR

Der MetaHuman Creator stellt eine cloudbasierte Plattform in Form eines Charaktereditors dar. Auf Basis von bereits vorhandenen Scans, welche reale Menschen darstellen, kann eine neue digitale Person durch eine Mischung der verschiedenen Gesichtsproportionen erstellt werden. Im nächsten Schritt werden die erstellten Gesichtsproportionen mit weiteren Editor-Tools für Gesicht und Körper verfeinert. Zusätzlich kann durch eine Auswahl an verschiedenen Haaren und Kleidung ein vielfältiges Spektrum von Charakteren erstellt werden.

Aufgrund der einfachen und effizienten Arbeitsweise konnten alle benötigten Modelle innerhalb kürzester Zeit erstellt werden. Diese umfassten den Protagonisten sowie eine Auswahl an männlichen und weiblichen Charakteren für die Crowds.



Abb. 106: MetaHumans für die Crowds



Abb. 107: Mesh to MetaHuman

MESH TO METAHUMAN

Eine alternative Methode zur Erstellung eines Charakters stellt Mesh to MetaHuman dar.

Die Erstellung eines vollwertigen MetaHumans ist mittels eines Scans des eigenen Gesichts oder eines bereits bestehenden 3D-Modells möglich. In beiden Fällen erfolgt die dreidimensionale Erfassung der Gesichtszüge anhand verschiedener Perspektiven des Kopfes, welche als "Identity" verarbeitet werden. Das Resultat kann im MetaHuman Creator weiter verfeinert werden und mittels Quixel Bridge nahtlos als vollständiger Charakter mit vollem Funktionsumfang in der Unreal Engine genutzt werden.

Diese Vorgehensweise wurde nicht direkt für die Erstellung eines Charakters genutzt, jedoch war Mesh to MetaHuman zur Nutzung des MetaHuman Animators für das Facial Mocap notwendig.

METAHUMAN ANIMATOR

Der MetaHuman Animator stellt das Facial Mocap Tool der MetaHumans dar. Damit können Gesichtsanimationen in einer extrem realistischen und präzisen Art und Weise anhand eines Videos aufgenommen werden. Eine detaillierte Erläuterung sich im Kapitel "Animation" dieser Arbeit.



Abb. 108: MetaHuman Animator

INTEGRATION

Nachdem die Erstellung der Charaktere abgeschlossen war, konnten diese mittels der nahtlosen Integration von Quixel Bridge innerhalb der Unreal Engine abgerufen und importiert werden. Dazu werden alle relevanten Daten zur Darstellung eines MetaHumans in einem Charakter-Blueprint zusammengefasst. Dazu gehören auch alle erforderlichen Schritte zur Nutzung von Realtime-Haar-Simulationen sowie der vollständigen Animationspipeline von Unreal.

4.2.3 AUFBAU EINES CHARACTERS

RIG

Damit ein 3D-Charakter animiert werden kann, benötigt es eine digitale Repräsentation eines Skelettes, im Englischen als "Rig" bezeichnet. Dieses besteht aus einer hierarchischen Anordnung von Knochen (Bones), welche das Verformen eines Meshes mittels Bewegen und Rotieren ermöglicht. Aufgrund der Struktur besteht eine gewisse Abhängigkeit zwischen den Knochen (Bones), die durch den Einsatz von Controllern, Constraints und Deformern beeinflusst werden kann. Dies gewährleistet, dass beispielsweise der Fuß stets in Kontakt mit dem Boden bleibt, insbesondere bei einer Beugung des Beines oder einer Verformung der Haut im Bereich des Ellbogens. In diesem Zusammenhang werden die Manipulationsmöglichkeiten eines Rigs durch ein Steuerungssystem aus verschiedenen Gizmos kontrolliert.

Der Rigging-Prozess kann manuell oder automatisiert durchgeführt werden. Zu den bekannten Beispielen für automatisierte Tools zählen Mixamo, AccuRig und Blender-Rigify.

In Bezug auf die genutzten MetaHumans ist das Rigging nicht erforderlich, da bereits ein Skelett existiert. In diesem Zustand ist eine Steuerung des Skeletts innerhalb der Unreal Engine nicht möglich, sodass Posen und Animationen in einer externen Software erstellt werden müssen. Das Control Rig bietet eine Lösung dafür, da es viele Möglichkeiten für die Manipulation eines MetaHumans zulässt. Allerdings lassen sich externe Animationen nicht mit dem Control Rig kontrollieren. Im Rahmen dieses Projekts wurde sich gegen das Control Rig entschieden und auf externe Software wie Cascadeur gesetzt. Mehr dazu im Animations Teil dieser Arbeit. [80]

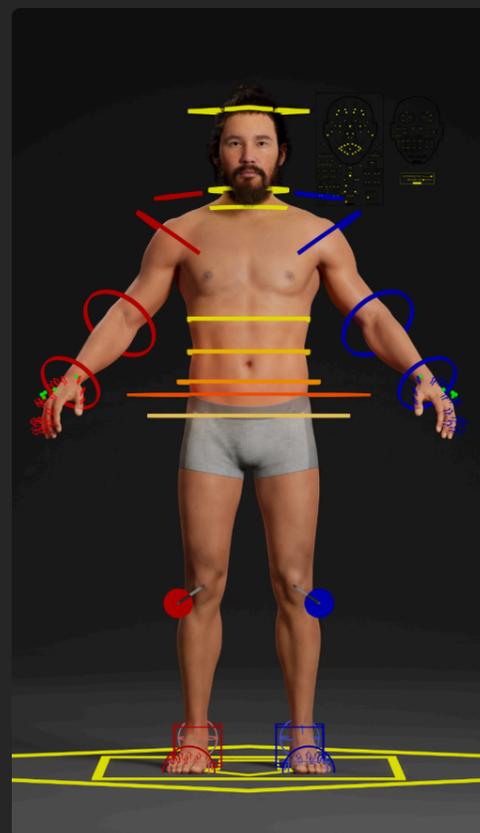


Abb. 109: MetaHuman Control-Rig

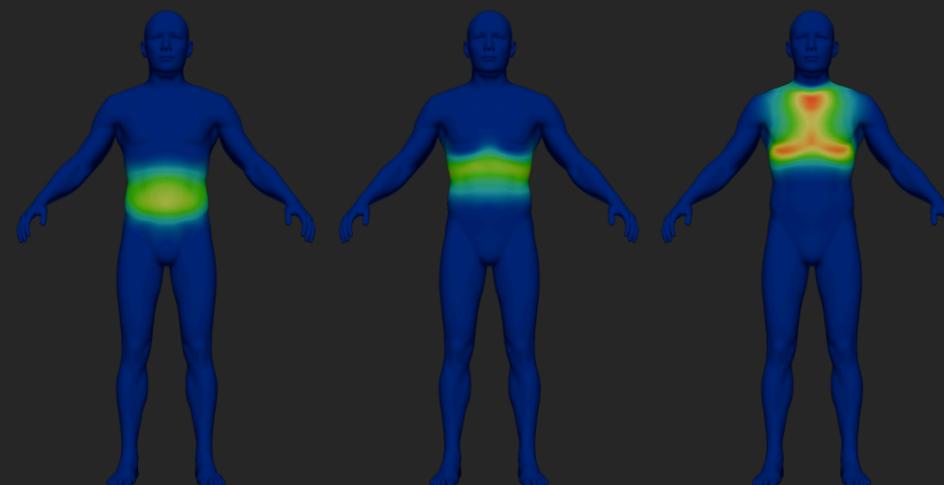


Abb. 110: Weight Painting MetaHuman | Spine 2 - 4

WEIGHT PAINTING

Die Technik des Weight Paintings dient der Steuerung der Einflüsse von Skelettknochen auf die umliegende Geometrie. Zu diesem Zweck wird jedem Vertex ein Gewichtswert pro Knochen zugewiesen, welcher die Stärke der Beeinflussung durch die Bewegungen des jeweiligen Knochen definiert. Auf diese Weise lässt sich die Geometrie durch die Animation der Knochen deformieren. Der Wert kann dabei einen Zahlenwert zwischen 0 und 1 annehmen, wobei eine Visualisierung durch ein Farbspektrum erfolgt.

Das Weight Painting lässt sich entweder traditionell über manuellem Weg oder inzwischen auch automatisch durchführen, wobei sich die traditionelle Methode als die präziseste erwiesen hat.

Im Fall dieser Arbeit wurde das Weight Painting ausschließlich für die Kleidung benötigt, da die MetaHumans bereits Skin Weights besitzen. Zudem wurde aus zeitlichen und experimentellen Gründen der automatische Weg in Form des Chaos Cloth Panels gewählt. Eine detaillierte Erläuterung erfolgt im nachfolgenden Abschnitt zum Thema Kleidung. [81]

4.2.4 CLOTH

CHARACTERMODEL

Der Zugriff auf die Daten eines MetaHuman schafft die Grundlagen für weitere Anpassungsmöglichkeiten, weshalb im Rahmen dieser Arbeit das Körpermesh als Basis für die Erstellung der Kleidung diente.

Dafür musste der Körper zunächst in zwei Schritten exportiert werden, da der Kopf und der Körper als separate Objekte voneinander getrennt sind. In diesem Fall wurden beide Objekte in Blender importiert und als ein Mesh zusammengeführt, um sie für die Simulation der Kleidung in Marvelous Designer vorzubereiten.



Abb. 111: Kombiniertes Körper des MetaHuman

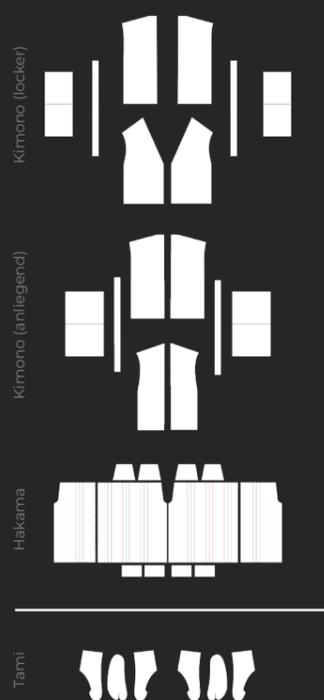


Abb. 112: Schnittmuster

CLOTH SIMULATION

Die Erstellung der Kleidung erfolgte durch die Simulationssoftware Marvelous Designer, welche für virtuelle Kleidung und Textilien genutzt wird. Wie beim konventionellen Entwerfen von Kleidungsstücken werden im ersten Schritt von Marvelous 2D-Schnittmuster erstellt. Dazu wurden Referenzen gesucht, welche die Schnittmuster der benötigten Kleidungsstücke abbilden.

Diese Referenzen wurden dann im Pattern-Editor mit den verfügbaren Werkzeugen abgepauscht und auf die korrekten Proportionen des Charakters skaliert. Die erstellten 2D-Pattern werden dabei simultan in den dreidimensionalen Raum überführt. Zusätzlich werden automatisch passende UVs generiert.

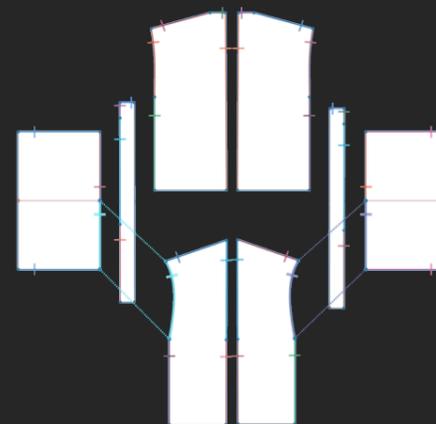


Abb. 113: Sewing (gleiche Farben werden zusammengenäht)

Im nächsten Schritt erfolgt die Bestimmung der physikalischen Eigenschaften der einzelnen Kleidungsstücke. Dabei werden durch die Variation der Stoffdichte, Elastizität und des Faltenwurfs Eigenschaften realer Textilstoffe, wie Baumwolle oder Seide, nachgeahmt. Dies hat einen direkten Einfluss auf die Simulation und steuert, in welchem Umfang die Kleidungsstücke an einem Körper anliegen.



Abb. 115: Fertig simulierte Kleidung

Im Anschluss an die Erstellung der Schnittmuster, galt es diese zusammenzunähen. Zu diesem Zweck wurde das "Sewing"-Werkzeug benötigt. Dabei wurden die Schnittmuster entweder über die Kanten mittels "Segment-Sewing" oder manuell mit "Free-Sewing" zusammengenäht.

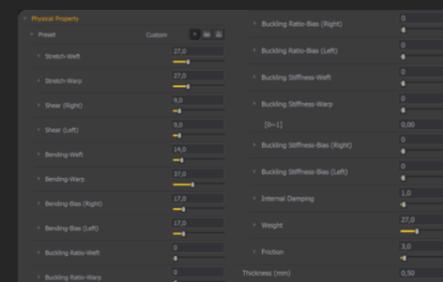


Abb. 114: Eigenschaften am Beispiel eines Kimonos

Um der Cloth-Simulation eine Grundlage zu geben, wie die Kleidung übereinander liegen, werden die einzelnen Kleidungsstücke in verschiedene Layer unterteilt. Dieser Schritt ist von entscheidender Bedeutung, da andernfalls die Kleidung während der Simulation Fehler aufweist.

Für die Simulation werden die Kleidungsstücke der Layer-Reihenfolge nach einzeln simuliert, um ein korrektes Verhalten der Kleidung zu gewährleisten. Nach der Positionierung der einzelnen Schnittmuster eines Kleidungsstücks im 3D-Viewport kann die Simulation gestartet werden. Dabei besteht jederzeit die Möglichkeit, die Simulation zu unterbrechen, um Änderungen am Schnittmuster oder am Stoffverhalten vorzunehmen.



Abb. 116: Aufbereitete Kleidung | Links = Unbearbeitet / Mitte = Überarbeitet / Rechts = Details

AUFBEREITUNG DER KLEIDUNG

Nachdem die Kleidung fertig simuliert wurden, galt es diese nach Blender zur weiteren Überarbeitung zu exportieren. Dafür wurde das Mesh mittels Sculpting den Wünschen entsprechend verfeinert und fehlerhafte Stellen bereinigt. Dabei blieb die Komplexität des Meshes unverändert.

Im Fall des Samurai wurden darüber hinaus die fehlenden Schuhe sowie Accessoires wie Bandagen, Bambushut und Schwert mittels 3D-Modellierung hinzugefügt. Zudem wurde in Substance Painter die finale Kleidung mit Materialien versehen.

Aio: <https://www.youtube.com/watch?v=FIL1pWM4CZU>

NPCs: <https://www.youtube.com/watch?v=THd0Lla2Rew>



Abb. 117: Finale texturierte Kleidung



Static Mesh



Skin Weights



Weight-Map



Physics Asset

Abb. 118: Bestandteile des Chaos Cloth Systems

CHAOS CLOTH PANEL

Das Chaos Cloth Panel stellt ein spezielles Tool zur Echtzeit-Simulation von Kleidung und anderen Stoffen dar. Es handelt sich dabei um eine Weiterentwicklung des bereits in der Unreal Engine 4.2.7 eingeführten Chaos-Systems, welches nun einen nodebasierten Ansatz verfolgt. Zudem wird eine für die Umsetzung relevante Funktion in Form des automatischen Übertragens der Skin Weights eines Skeletal Meshes auf die Kleidung ermöglicht. Die Funktionsweise lässt sich dabei in folgenden Schritten beschreiben: [82]

1. Die erstellte Kleidung in das Cloth Asset importiert.
2. Im Anschluss erfolgt die Übertragung der Skin Weights des Körpers in Form von Vertex-Daten auf die Kleidung. Dies gewährleistet, dass sich die Kleidung bei Bewegung des Körpers automatisch mitbewegt.
3. Die von der Simulation beeinflussbaren Bereiche des Meshes werden durch eine Maske in Form einer Weight-Map eingezeichnet. Dabei werden schwarze Bereiche von der Simulation nicht berücksichtigt, während weiße Bereiche vollständig von der Simulation erfasst werden.
4. Anschließend wird das Verhalten der Kleidung durch die Simulationseinstellungen bestimmt und auf die maskierten Bereiche übertragen.
5. Damit die Simulation mit dem Körper interagieren kann, werden die notwendigen Daten mit der Collision und dem Physics-Assets ermittelt.
6. Optional können zusätzliche Turbulenzen in Form von Wind in die Simulation einbezogen werden.

4.3 ANIMATION | 4.3.1 GRUNDLAGEN

ANIMATION

Grundlegend lässt sich die Animation als schnelle Abfolge von einzelnen Bildern, die als Bewegung wahrgenommen werden, beschreiben. Die Anzahl der Bilder pro Sekunde bestimmt, wie flüssig eine Bewegung für den Betrachter erscheint. In diesem Zusammenhang wird oft von Frames gesprochen.

Die Animation hat ihren Ursprung im Zeichentrick und wurde traditionell in Form von handgezeichneten Sequenzen dargestellt. Dabei hat sich die Pose-to-Pose-Animation als klassische Methode zur Erstellung von Animationen etabliert. Dabei werden

zunächst die "Extremes" erstellt, welche die Anfangs- und Endpunkte einer Bewegung repräsentieren. Im Anschluss werden mit den "Breakdowns" die Übergangsposen zwischen den Extrempunkten dargestellt. Die "Inbetweens" schließlich sorgen für eine fließende Bewegung, indem sie die einzelnen Posen miteinander verbinden.

Diese grundlegende Animations-Technik lässt sich auch auf andere Bereiche wie die Computeranimation übertragen. In diesem Zusammenhang wird von sogenannten Keyframes gesprochen. [83]

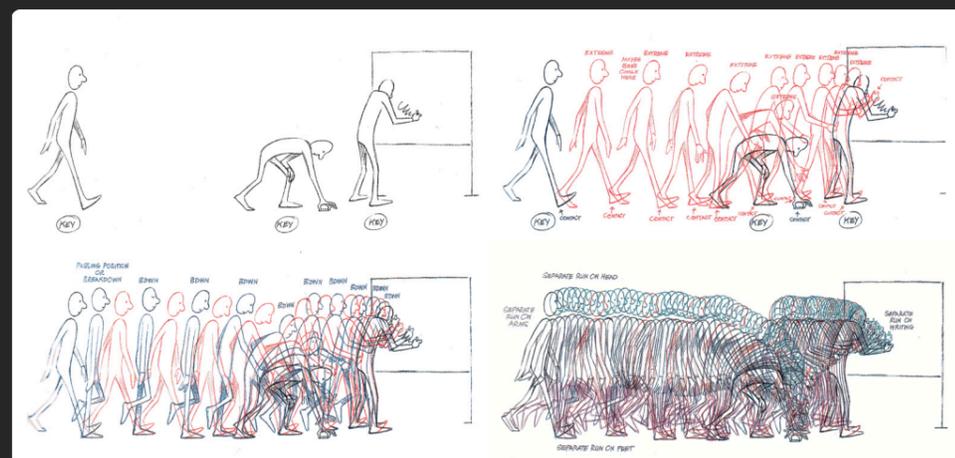


Abb. 119: "The Animator's Survival Kit" - Richard Williams

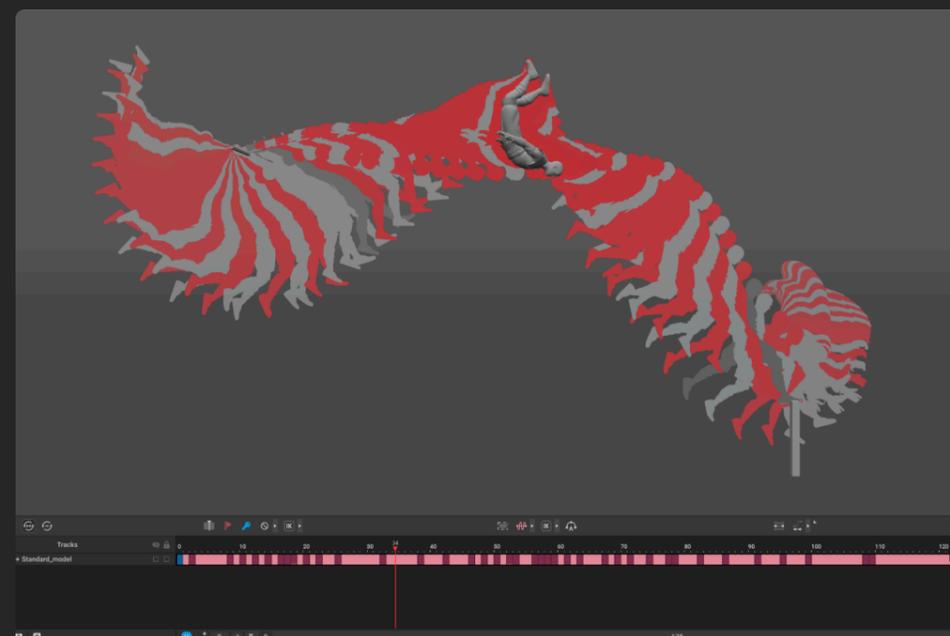


Abb. 120: Keyframe-Animation

KEYFRAME-ANIMATION

Die Keyframe-Animation stellt eine im 2D- und 3D-Bereich etablierte Technik zur zeitlichen Veränderung der Eigenschaften eines Objektes oder Charakters dar. Die Eigenschaften umfassen dabei beispielsweise Posen, Transformationen sowie Farben. Das zugrundeliegende Prinzip entspricht dem der Pose-to-Pose Animation. Die Erstellung der wesentlichen Stellen einer Animation erfolgen als Keyframes. Durch Interpolationen zwischen den Keyframes werden jedoch keine "Inbetweens" für eine flüssige Bewegung mehr benötigt, wodurch sich der Arbeitsaufwand im Vergleich zur Zeichentrickanimation signifikant reduziert. Mit den Vorteilen ergeben sich jedoch auch mehr Möglichkeiten, sodass Animationen zunehmend komplexer und realistischer werden.

In Bezug auf die vorliegende Arbeit lässt sich festhalten, dass der manuelle Prozess des Keyframing einen äußerst zeitintensiven Vorgang darstellt, bis die Bewegungen eines Menschen als natürlich und glaubwürdig wahrgenommen werden. Daher findet der manuelle Prozess lediglich bei konzeptionellen Zwecken Anwendung, beispielsweise beim Animatic. Für den finalen Film wurde nach einer alternativen Methode gesucht, und es stellte sich heraus, dass das Motion Capturing eine deutlich geeignetere Vorgehensweise für die Erstellung der Animationen darstellt. Hierbei werden reale Bewegungen aufgezeichnet und automatisch in Keyframes umgewandelt. Eine detaillierte Erläuterung dieser Methode erfolgt im nachfolgenden Abschnitt.

4.3.2 MOTION CAPTURING

ALLGEMEIN

Unter Motion Capturing, auch als Mocap bezeichnet, wird ein Verfahren verstanden, durch welches menschliche Bewegungen digital erfasst und aufgenommen werden. In diesem Kontext wird auch oft der Begriff Motion-Tracking verwendet. Dadurch lassen sich realistische Animationen für 3D-Charaktere aufnehmen, die mit manuellem



Abb. 121: Optisch - Anzug mit Markern, die von Kameras erfasst werden



Abb. 122: AI - Markerlose Aufnahmen durch KI-basierte Algorithmen

Für die Umsetzung wurde die KI-basierte Lösung in Form von MoveAI gewählt, da andere Methoden entweder mit hohen Kosten oder einem hohen Aufwand verbunden

Keyframing schwer zu erreichen wären. Es existieren verschiedene Arten von Motion Capturing, die für die Arbeit relevanten umfassen das Full-Body Motion Capture und das Facial Capture. [84]

Darüber hinaus gibt es verschiedene Aufnahmemethoden. Die gängigsten sind dabei folgende:



Abb. 123: Inertial - Sensoren am Körper, die die Bewegung messen

wären. Zudem wird die gewählte Methode aufgrund ihrer hohen Flexibilität und einfachen Bedienbarkeit zunehmend im professionellen Bereich eingesetzt.

MOVE AI

Die KI-basierte Motion-Capture-Technologie MoveAI ermöglicht die Erfassung menschlicher Bewegungen ohne die Notwendigkeit der Verwendung traditioneller Marker oder spezieller Anzüge. Stattdessen erfolgt die Generierung von 3D-Animationen durch eine Kombination aus Kameras und Algorithmen anhand von Videomaterial. In diesem Kontext stellt MoveAI eines der einfachsten und ausgereiftesten Tools in diesem Bereich dar. Im Vergleich zu den Konkurrenten erwies es sich als geeignetste Wahl für dieses Projekt. Das Mocap lässt sich dabei in einem Single-Camera- oder Multi-Camera-System aufnehmen. Aufgrund der ungewissen Kosten und der teuren Hardwareanschaffungen wurde zur ersten Methode gegriffen.

Der Aufbau des Motion-Capturing-Setups umfasste ein iPhone sowie ein Stativ. Da der Aufnahmeort nach bisherigen Erkenntnissen keinen Einfluss auf die Qualität des Mocaps hat, konnten die Aufnahmen sehr einfach und flexibel gestaltet werden. Der intuitive Ablauf beginnt mit einer A-Pose zur Kalibrierung des Akteurs, gefolgt von der Aufnahme der gewünschten Bewegungen. Nach einer kurzen Verarbeitungszeit von wenigen Minuten wird das Mocap als FBX-Datei zum Download bereitgestellt.



Abb. 124: Move AI Mocap | Links = Aufnahme / Mitte = Rohdaten / Rechts = Animation auf dem UE5-Manny

RETARGETING

Das Retargeting beschreibt den Prozess, bei welchem Bewegungsdaten von einem Quell-Skelett auf ein Ziel-Skelett übertragen werden. Der Hintergrund dafür liegt in einer unterschiedlichen Anordnung und Benennung der Knochen zwischen den Skeletten. Dadurch besteht die Möglichkeit einer fehlerhaften Übertragung der Daten auf ein anderes Skelett. Beim Retargeting werden daher die entsprechenden Bones dem Ziel-Skelett zugeordnet (gemappt). In den meisten Fällen wird dazu auch eine identische Pose in Form einer T- oder A-Pose benötigt (beispielweise Unreal Engine, iClone, Cascadeur). [85]

Ein wesentlicher Anwendungsfall umfasst das Retargeting von Mocap-Daten, wie es beispielsweise wegen MoveAI zum Einsatz kommt. Durch das neue Auto-Retargeting-Tool der Unreal Engine 5.4 konnte dieser Prozess deutlich vereinfacht werden, da ein

entsprechendes Preset zur Verfügung stand. Das neue Feature unterstützt in diesem Zusammenhang die bekanntesten Rigs und vereinfacht das Retargeting für die Unreal Engine erheblich. [86]

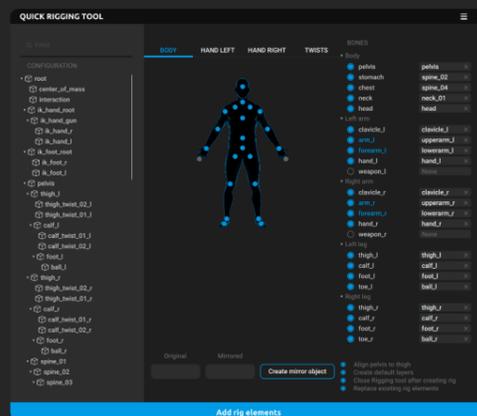


Abb. 125: Beispiel für das Remapping von Bones - Cascadeur



Abb. 126: Unterschiedliche Skelette mit derselben Animation (Retargeting)

CLEANUP

Aufgrund der Möglichkeit, dass Mocap-Daten kleinere oder größere Probleme beinhalten, ist eine manuelle Aufbereitung der Animationen erforderlich. Insbesondere während des AI-Mocap-Prozesses sammeln sich Fehlerquellen wie Foot-Sliding, Glitches oder zittrige Bewegungen. Die Animationssoftware Cascadeur erwies sich dabei als effizientes Tool zur Bereinigung dieser Probleme. Da ein mit dem MetaHuman kompatibles Skelett als Voreinstellung verfügbar war, wurde dieses als Basis für die Aufbereitung der Animationen verwendet.

Im ersten Schritt wurden die Animationen zunächst in ihre essentiellen Keyframes aufgeteilt und die Fehler bereinigt. Hierbei erwies sich das Auto-Posing-Feature, welches mittels KI jede Änderung nach korrekter Statik überprüft und automatisch korrigiert, als außerordentlich hilfreich. Dadurch

konnten natürliche Posen für den Charakter sichergestellt werden.

Danach wurden sämtliche übrigen Keyframes entfernt und mittels passender Interpolation geschlossen, wodurch die grundlegende Animation wiederhergestellt werden konnte.

Der Animation fehlt es jedoch noch an minimalen Bewegungen, sogenannter Secondary-Motion. Diese wurde mithilfe des Auto-Physics-Features hinzugefügt. Hierbei handelt es sich um KI-basierte Algorithmen, welche die Bewegungen analysieren und physikalische Einflüsse mit einberechnen. Dadurch wirken die Animationen deutlich realistischer und natürlicher. Dabei wird ein grüner Doppelgänger generiert, welcher als Preview für die Änderungen dient, bis diese auf das Original angewendet werden. Dieser letzte Schritt stellt die finale Animation dar.

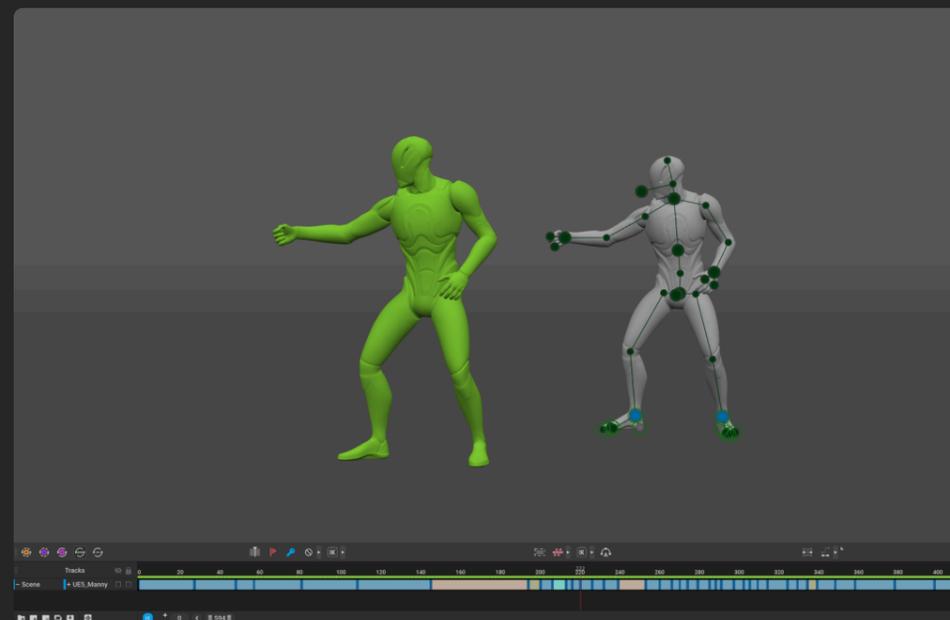


Abb. 127: Überarbeitete Mocap Daten

BLENDSHAPES

Im Gegensatz zum Animieren mit Bone-Animations stellt das Animieren mit Blendshapes, auch bekannt als Morph Targets oder Shape Keys, die Interpolation zwischen mehreren relativen Zuständen des Meshes dar. Die Zustände (Shapes) repräsentieren dabei modifizierte Kopien des Basis-Meshes in Form von Posen und Ausdrücken, wie etwa ein Lächeln, ein Stirnrunzeln oder geschlossene Augen. Die vordefinierten Verformungen ermöglichen eine präzise Steuerung der Blendshapes, wobei deren Anwendungsbereich vor allem bei Gesichtsausdrücke, Lippenbewegungen und Muskeln gebrauch finden. Die gleichzeitige Kombination mehrerer Blendshapes erlaubt die Generierung komplexer und realistischer Bewegungen. Dieses Prinzip findet ebenfalls Anwendung bei der Steuerung des Gesichts eines MetaHumans. [87]

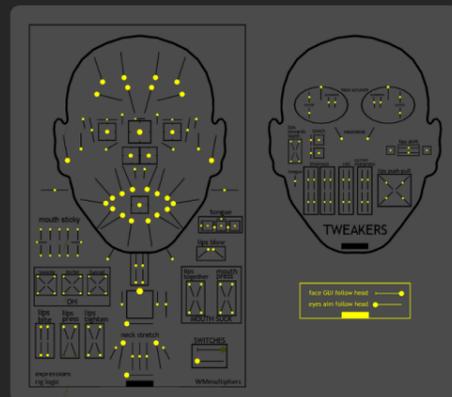


Abb. 128: MetaHuman Facial Control Rig



Abb. 129: Verschiedene Mimiken durch das Manipulieren der Blendshapes



Abb. 130: Erstellung einer "MetaHuman Identity" | Links = Kalibrierung / Rechts = Digitales Abbild

METAHUMAN ANIMATOR

Da sich die Gefühlslage und Stimmung einer Person sehr gut über die Mimik darstellen lässt, sollte dieser Aspekt einen wesentlichen Bestandteil des Animationsfilms darstellen. Allerdings sind realistische Gesichtsbewegungen sehr komplex und über manuelles Keyframing schwer zu erreichen. Aus diesem Grund wurden alle benötigten Bewegungen des Gesichts mittels Motion Capturing aufgenommen.

Zunächst wird ein Kalibrierungs-Clip aufgenommen. Diese Aufnahme wird vorausgesetzt, um die Gesichtszüge des Akteurs auf die Blendshapes eines MetaHuman abzustimmen. Zusätzlich wird ein digitales Abbild des Kopfes in Form eines Meshes erzeugt.

Nach erfolgreich abgeschlossener Kalibrierung kann das richtige Facial-Mocap aufgenommen werden. Die Aufnahmen werden nun mittels der erstellten "MetaHuman Identity" verarbeitet und können nach Fertigstellung als Animationssequenz exportiert werden.

Im finalen Schritt erfolgt die Umwandlung des Facial-Mocap in ein Animation-Asset, sodass eine Anwendung auf alle MetaHumans gewährleistet ist. Die daraus resultierenden Ergebnisse lassen sich kaum von professionellen Animationen unterscheiden. Dadurch stellt das Facial-Mocap eine wertvolle Ergänzung für den Animationsfilm dar.



Abb. 131: Facial Mocap Performance

4.4 VFX | 4.4.1 EMBERGEN

FEUERSIMULATION

Die Echtzeit-Simulationssoftware EmberGen wurde speziell für die Erstellung und Bearbeitung von Feuer, Rauch, Explosionen und anderen volumetrischen Effekten entwickelt. Das Tool zeichnet sich durch einen einfachen und kompakten Workflow sowie qualitativ hochwertige

Ergebnisse aus. Aus diesen Gründen wurde EmberGen für die Erstellung des benötigten Feuers genutzt. Die grundlegende Funktionsweise des Simulationsaufbaues lässt sich wie folgt zusammensetzen:

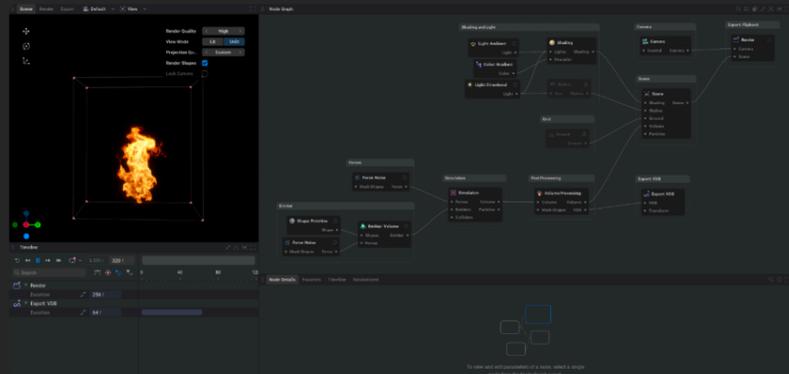


Abb. 132: Feuersimulation in EmberGen

Emitter: Definiert die grundsätzliche volumetrische Form und Position des Emitters. Diese Daten werden als Voxel der Simulation übergeben.

Simulation: Hier werden die Einstellungen für die Berechnung der Simulation definiert, wobei einwirkende physikalische Einflüsse mitberücksichtigt werden. Im Beispiel von Feuer lassen sich unter anderem die Dichte, Temperatur, Turbulenzen und Lebensdauer bestimmen.

Post Processing: Mittels unterschiedlicher Post-Processing Einstellungen kann die visuelle Darstellung mit unterschiedlichen Effekten gesteuert werden. Im vorliegenden Beispiel wurde den Flammen dafür etwas Motion Blur hinzugefügt.

Shading and Light: Bestimmt das generelle Shading (Farbe, Leuchtkraft, Übergänge) der Simulation.

Export: EmberGen bietet die Möglichkeit, entweder als Flipbook oder VDB zu exportieren. Im Folgenden werden beide Optionen näher erläutert.

FLIPBOOK

Ein Flipbook stellt eine weit verbreitete Darstellungsmethode von Animationen oder Simulationen in zweidimensionaler Form dar. Dieses besteht aus einer Serie von Einzelbildern, welche jeweils eine Phase der Animation darstellen. In der Regel werden die einzelnen Bilder der Sequenz innerhalb eines zusammenhängenden Bildes (Texture Atlas) zusammengefasst. Durch einen iterativen Vorgang wird zu den Positionen der Einzelbilder gesprungen und innerhalb eines Materials als Animation abgespielt. Diese Methode ist zwar sehr effizient, besitzt jedoch keinerlei Dreidimensionalität. Da dieser Aspekt hohe Priorität hatte, fanden Flipbooks keinen Einsatz in dieser Arbeit.



Abb. 133: Feuersimulation als Flipbook

VDB

Eine Volume Data Base, kurz VDB, stellt eine offene 3D-Datenstruktur (OpenVDB) dar, welche zur Speicherung, Manipulation und Darstellung von volumetrischen Daten dient. Dies ermöglicht eine effiziente und einfache Übertragung von Simulationen, wie beispielsweise Feuer, Nebel oder Wolken, in andere Anwendungen. Im Vergleich zu Flipbooks ermöglicht das volumetrische Format eine deutlich glaubwürdigere Darstellung der Simulationen. Für diese Arbeit wurde es daher als priorisiertes Format gewählt. Die Verarbeitung der Daten in Unreal erfolgte durch den Einsatz eines Sparse-Volume. Dieses verarbeitet nur die relevanten Teile eines Volumens, um Ressourcen zu sparen. Dabei werden die Daten hierarchisch in einer Art Baumstruktur abgerufen.[88]



Abb. 134: Feuersimulation als VDB

4.4.2 NIAGARA

SYSTEM

Das Niagara-System stellt das VFX-Tool der Unreal Engine dar. Mit diesem ist es möglich, visuelle Effekte wie Feuer, Explosionen oder Nebel in Echtzeit dynamisch und interaktiv darzustellen. Dabei weist Niagara eine modulare und flexible Architektur auf. Diese lässt sich anhand des "Falling Leave Systems" der Arbeit wie folgt erklären.

EMITTER

Ein Emitter fungiert als Container für unterschiedliche Module. Durch die Kombination mehrerer Emitter lassen sich komplexe Partikelsysteme abbilden. Ein Beispiel hierfür ist das Feuer, welches aus den einzelnen Emittern Flammen, Rauch und Funken besteht. Jeder Emitter kann dabei unabhängig voneinander konfiguriert und angesteuert werden. Im Fall des "Falling Leave Systems" war lediglich ein Emitter notwendig, um die komplexe Funktionalität abzubilden.

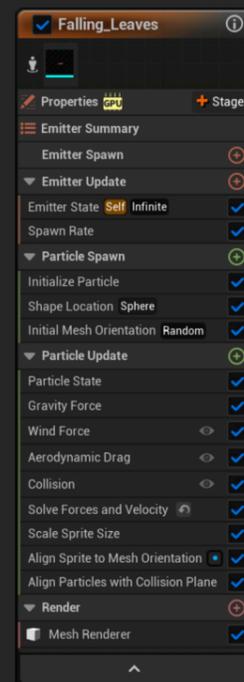


Abb. 135: Niagara Emitter mit Modulen

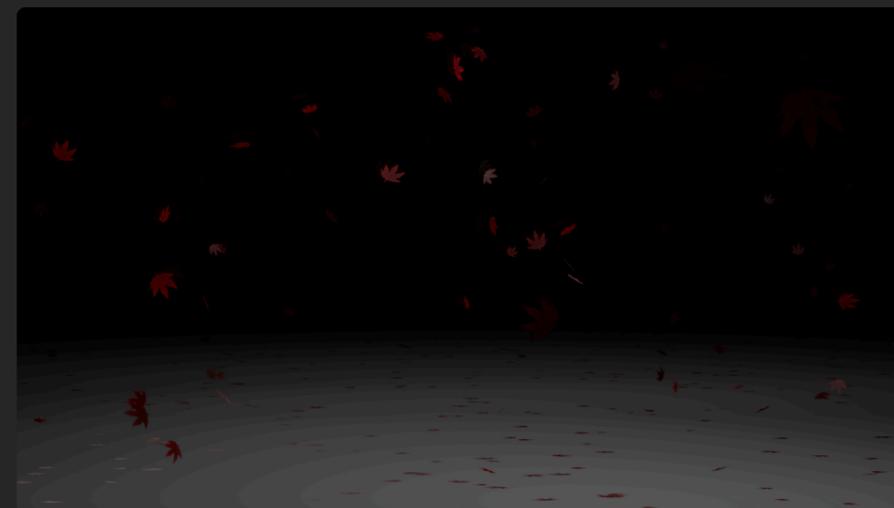


Abb. 136: Falling Leave System

MODULE

Das Verhalten der Emitter wird durch einzelne Module bestimmt, welche verschiedene Aspekte wie Lebensdauer, Geschwindigkeit, Farbe, Größe und Bewegung der Partikel definieren.

Emitter Spawn: Wird einmal zu Beginn des Partikelsystems abgerufen, um die grundlegenden Eigenschaften und Zustände des Emitters zu bestimmen. Dazu gehören die Menge, Lebensdauer und Loop-Eigenschaften der zu erzeugenden Partikel. Da in diesem Beispiel fortlaufend neue Partikel erstellt werden, wurden die Eigenschaften im folgenden "Emitter Update" festgelegt.

Emitter Update: Die Funktionsweise ist vergleichbar mit der von Emitter Spawn, allerdings werden die Eigenschaften des Emitters zu jedem Frame der Laufzeit neu gesetzt. Dadurch ist es möglich, die Anzahl und die zeitlichen Abstände der Blätter zufällig zur Laufzeit neu zu bestimmen.

Particle Spawn: Erstellt die einzelnen Partikel mit gewissen Initial-eigenschaften. Dadurch wurden jedem Blatt zum Partikelstart eine zufällige Position, Größe und Ausrichtung innerhalb eines festgelegten Bereichs zugewiesen.

Particle Update: Aktualisiert kontinuierlich das Verhalten und die Eigenschaften der erzeugten Partikel. Dadurch konnte das Fallen der Blätter simuliert werden. Dafür werden zu jedem Frame die Position, Geschwindigkeit, einwirkenden Kräfte (Wind, Gravitation), Ausrichtung und Kollision der Blätter neu berechnet. Zusätzlich wurde bei Kontakt mit dem Boden eine Bedingung zur automatischen Ausrichtung eingebaut, um eine natürliche Fallbewegung zu gewährleisten.

Render: Bestimmt die visuelle Darstellung der Partikel. Die gängigsten Module stellen dabei der Sprite-, Mesh- und Light Render dar. In diesem Fall wird der Mesh Renderer benötigt, um die Geometrie eines Blattes zu simulieren.

4.5 RENDERING | 4.5.1 RENDERER

Der finale Schritt des 3D-Workflows umfasste das Rendering. Hierbei werden alle Daten einer 3D-Szene in Form eines zweidimensionalen Bildes oder einer ganzen Bildsequenz ausgegeben. Die benötigten Renderer unterscheiden sich in Bezug auf ihre Leistungsfähigkeit, Formatunterstützung, Ausgabequalität, Lichtmodellen sowie ihrem grundsätzlichen Look voneinander. Die genannten Unterschiede bestimmen, welcher Renderer für ein spezifisches Projekt am geeignetsten ist.

Die Unreal Engine bietet in diesem Kontext lediglich einen einzigen internen Realtime-Renderer. Dieser lässt sich jedoch in der visuellen Qualität und Flexibilität kaum von einem Offline-Renderer unterscheiden. Das Zusammenspiel von Qualität, Funktionalität und der enormen Zeitersparnis macht das Rendering innerhalb der Unreal Engine zu einem perfekten Gesamtpaket.

4.5.2 RENDERINGTOOLS

MOVIE SCENE CAPTURE

Die "Movie Scene Capture" stellt ein grundlegendes Tool für einfache und schnelle Renderingexporte dar. Hierbei wird die simulierte Szene in Echtzeit gerendert und weist keinerlei zusätzliches Post Processing auf. Die Software eignete sich in erster Linie für konzeptionelle Renderings, wie das Animatic oder kurze Previews. Zudem entsprach die Renderqualität nicht den Anforderungen für das finale Video. Aus diesem Grund wurde ein besseres Rendertool verwendet.



MOVIE RENDER QUEUE

Die Movie Render Queue (MRQ) war dabei das deutlich modernere und fortschrittlichere Werkzeug für das Rendering des Animationsfilms. Dieses Tool ermöglichte die bestmöglichen Renderergebnisse, die der Qualität eines Offline-Renderers (Cycles, Arnold, RedShift) entsprachen. Es bietet erweiterte Einstellungen sowie eine detaillierte Kontrolle über die Renderpipeline. Für das Projekt wurden folgende Einstellungen getroffen:



Abb. 137: Movie Scene Capture (Links) und Movie Render Queue (Rechts) im Vergleich

RENDERSETTINGS

Anti-Aliasing: Wird dazu verwendet, um das sogenannte Aliasing zu reduzieren. Dieses erzeugt bei scharfen Kanten und dünnen Linien eine gezackte Darstellung, auch Treppcheneffekt genannt. Mit dem Anti-Aliasing werden diese Kanten geglättet. Diese Methode erwies sich für interaktive Inhalte als sehr geeignet, für das Rendering der Cinematics jedoch als unbrauchbar. Dies ist auf die bemerkbaren Imperfektionen im Motion Blur zurückzuführen. Aus diesem Grund wurde Anti-Aliasing deaktiviert und stattdessen der Temporal Sample Count erhöht. Dies hatte positive Effekte auf die Bildqualität sowie auf das Aliasing.

Color Output: Bestimmt den Farbraum des Rendering. Dabei wurde die Tone Curve deaktiviert, um physikalisch korrekte Licht- und Farbwerte zu erhalten. Diese stellen einen guten Startpunkt für das Color Grading dar. Die Entscheidung fiel auf Linear sRGB, da auf diese Weise die bestmöglichen Voraussetzungen für das Color Grading geschaffen werden.

Console Variables: Um Post-Processing-Effekte sowie andere Rendering-Optionen zu steuern, wurden Werte von Engine-Variablen modifiziert. Dabei wurde beispielsweise die interne Rendering-Auflösung hochskaliert und die Stärke des Motion Blur reduziert, um ein möglichst scharfes Bild zu erhalten.

Game Overrides: Überschreibt mehrere grundlegende Einstellungen der Szenensimulation. Dadurch wurde gewährleistet, dass die bestmöglichen Optionen für eine optimale Bildqualität gewählt werden.

EXR-Dateiformat: Stellt das qualitativ hochwertigste Dateiformat für das Rendering dar. Gegenüber dem JPG- und PNG-Format umfasst EXR einen höheren Dynamikumfang, mehrere Renderpässe, höhere Farbtiefe und verlustfreie Kompression. Hierbei wurde eine Sequenz von Bildern exportiert, die in der externen Software Davinci Resolve zu einem ganzen Video zusammengeführt wurde.

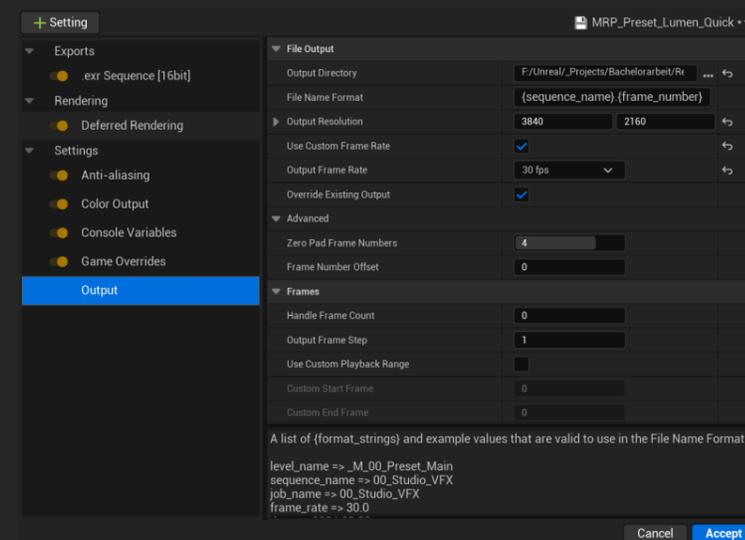


Abb. 138: Rendersettings

4.6 AUDIO | 4.6.1 ELEVENLABS

ELEVENLABS

Der Markt für Audio-Tools, die auf Basis künstlicher Intelligenz Stimmen, Soundeffekte oder Musik generieren, ist äußerst umfangreich. Da dieser Bereich zur Zeit der Umsetzung bereits weit fortgeschritten war, wurde nach einer optimalen Lösung für das Sounddesign gesucht. Das Tool "ElevenLabs" erfüllte dabei den Großteil aller erforderlichen Funktionen und zeichnete sich durch qualitativ hochwertige Ergebnisse aus, welche die der getesteten Konkurrenzprodukte übertreffen. Die relevanten Funktionen waren dabei folgende:

VOICE CLONING

Unter Voice Cloning wird die digitale Nachbildung der Stimme einer realen Person mittels Sprachsynthese verstanden. Dafür wird eine Sammlung von unterschiedlichen Sprachaufnahmen einer Stimme benötigt. Die Qualität und Präzision der geklonten Stimme ist dabei stark von der Menge und Diversität der Audioaufnahmen abhängig. Auf Basis dieser Sprachdaten erfolgt im Anschluss das Training eines entsprechenden Modells, welches die charakteristischen Merkmale der Stimme, darunter Tonhöhe, Klangfarbe, Betonung und Sprachmuster, nachahmt. Nach dem Training kann mittels Text- oder Spracheingabe ein gewünschter Dialog mit der künstlich generierten Stimme vertont werden.

Für die Erzählerstimme des Animationsfilms wurden dafür etwa eine halbe Stunde an verfügbaren Sprachaufnahmen einer passenden Stimme zusammengetragen, editiert und ElevenLabs zum Voice-Cloning übergeben. Hierfür wurde das Instant-Voice-Cloning-Feature genutzt.

TEXT TO SPEECH

Um sicherzustellen, dass die KI-Stimme einen gewünschten Dialog vertont, wurde der Dialog in Form eines Prompts an ElevenLabs übermittelt. Mithilfe von "Text to Speech" analysierte die KI den Inhalt des Dialogs, um die passenden Betonungen sowie den emotionalen Kontext zu bestimmen. Dies diente als Grundlage für die anschließende Generierung der Voice Lines. Dabei konnten weitere Prompting-Eigenschaften sowie unterschiedliche Parameter gesteuert werden, um die Ausdrucksweise zu beeinflussen. Das folgende Beispiel veranschaulicht dies:

"Samurai <break time="1.5s" /> What is your destiny?" <break time="1s" /> he said loud and seriously.



TEXT TO SFX

Dem gleichen Ansatz folgt die Methode "Text to SFX", welche anhand eines Prompts eine Auswahl an Soundeffekten generiert. Die Qualität der Ausgabe hängt dabei stark von der Menge und Formulierung der mitgelieferten Informationen ab. Hierbei wurden zahlreiche der im Film genutzten Soundeffekte durch dieses Feature erstellt, jedoch benötigte es einige Versuche, bis ein passendes Ergebnis generiert wurde.



4.6.2 UDIO

UDIO

Der letzte fehlende Bestandteil des Sounddesigns war die musikalische Untermalung. Da ElevenLabs zum Zeitpunkt der Umsetzung noch keine Möglichkeit zur Generierung von Musik anbot, wurde mit Udio eine potenzielle Lösung gefunden. UDIO ist eine webbasierte Plattform, die auf Basis eines vorgegebenen Prompts, welcher die zu erstellende Musik beschreibt, ein passendes Musikstück generiert. Dabei kann das komplette Musikspektrum an Instrumenten, Genres und Stimmen abgedeckt werden. Zusätzlich können neben unterschiedlichen Modellen und Anpassungen auch Lyrics genutzt werden, welche entweder selbst geschrieben oder automatisch generiert werden können. Die Ergebnisse sind vielversprechend, wie folgendes Beispiel zeigt:

Link: <https://www.udio.com/songs/3eZuxQcj1JfzCd9c8VjDjp>

Prompt: glam metal, 1980s, hair metal, hard rock, album rock, anthemic, female vocalist, energetic, passionate, melodic,

Letztendlich wurde sich gegen die Verwendung von KI generierter Musik entschieden, da die erstellten Ergebnisse noch Optimierungspotenzial aufwiesen. Für eine ideale Melodie wäre ein größerer Zeitaufwand sowie ein weiteres, kostenpflichtiges Abo erforderlich gewesen.

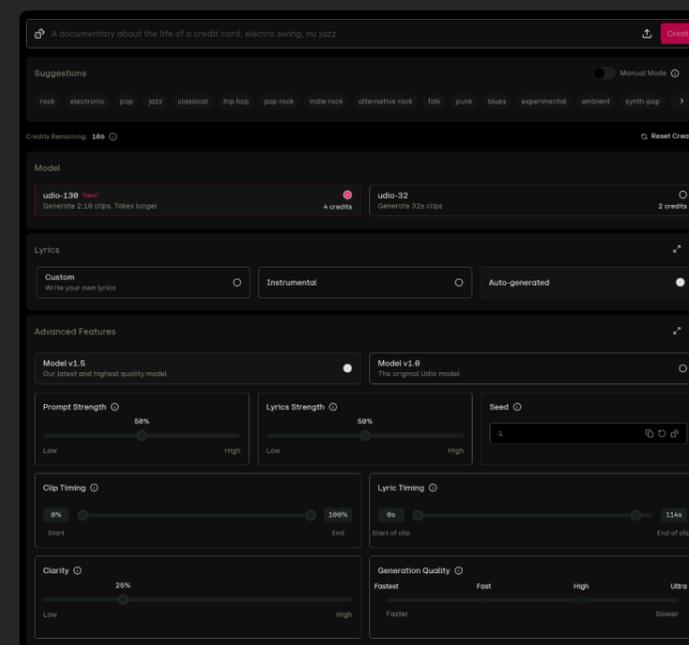
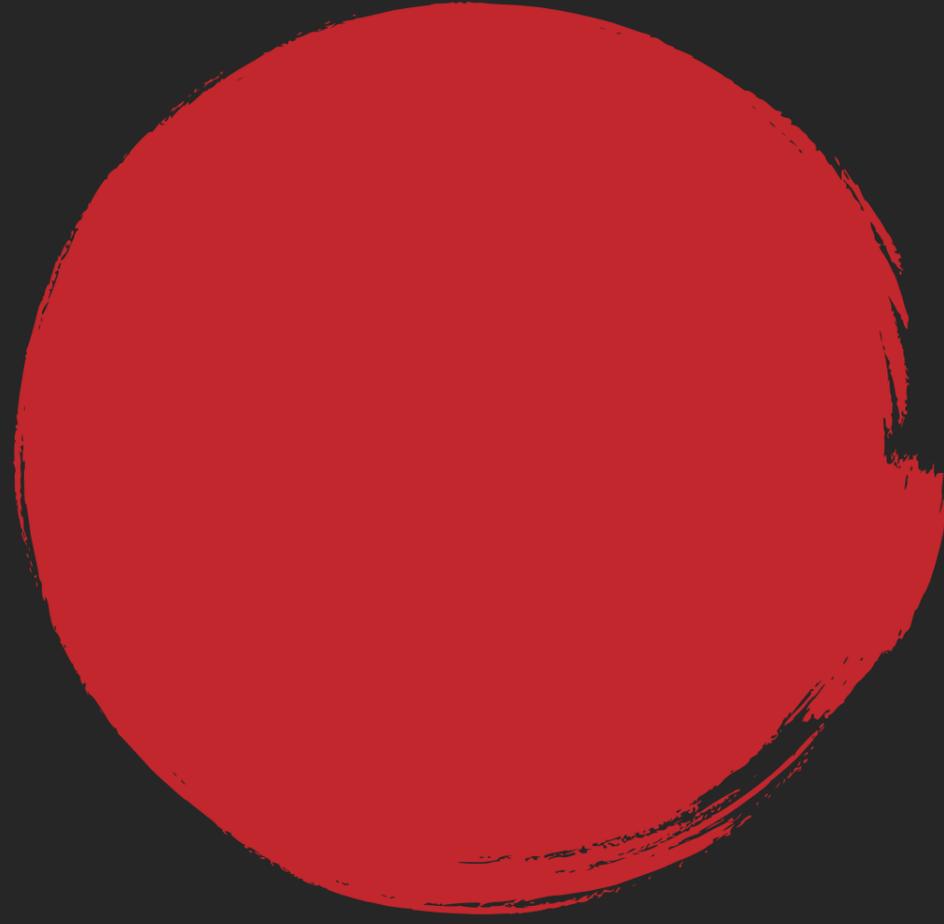


Abb. 139: Udio Weboberfläche

05 ERGEBNIS



<https://www.youtube.com/watch?v=a659EePDWwU>



Abb. 140: Szene 01 - Shot 01



Abb. 143: Szene 02 - Shot 03



Abb. 141: Szene 02 - Shot 01



Abb. 144: Szene 02 - Shot 04



Abb. 142: Szene 02 - Shot 02



Abb. 145: Szene 02 - Shot 05



Abb. 146: Szene 02 - Shot 06



Abb. 149: Szene 02 - Shot 07



Abb. 147: Szene 03 - Shot 01



Abb. 150: Szene 04 - Shot 01



Abb. 148: Szene 03 - Shot 02



Abb. 151: Szene 05 - Shot 01



Abb. 152: Szene 06 - Shot 01



Abb. 155: Szene 07 - Shot 03



Abb. 153: Szene 07 - Shot 01



Abb. 156: Szene 08 - Shot 01



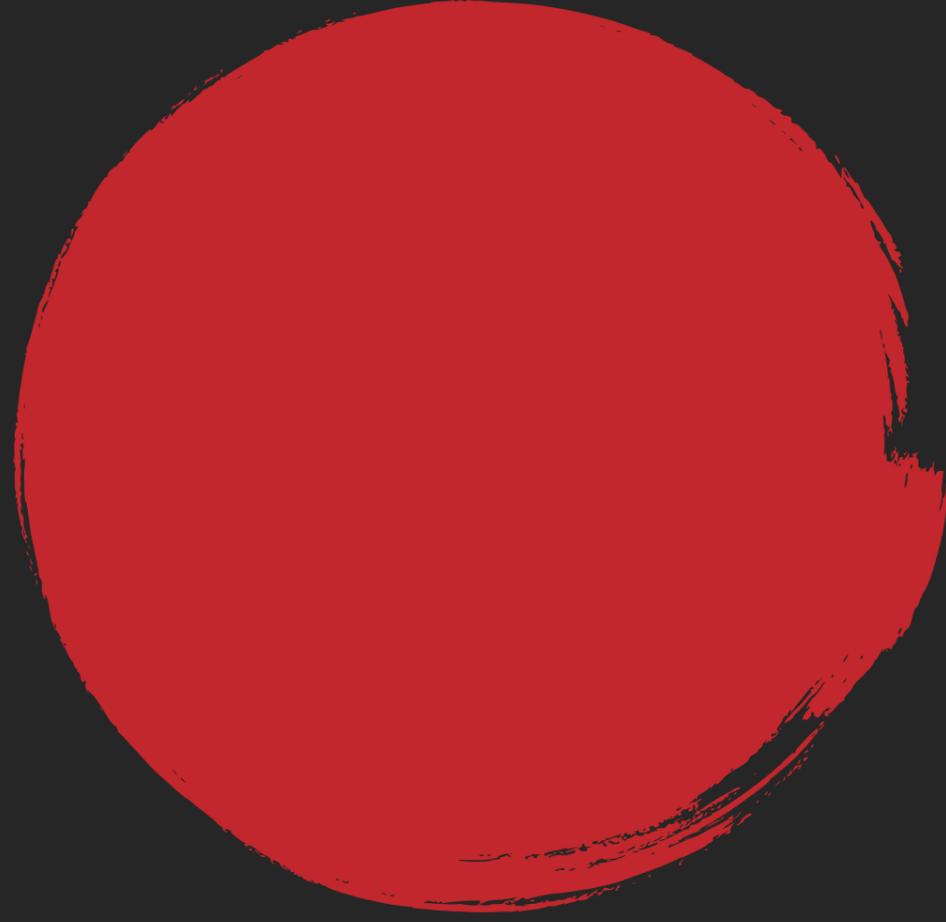
Abb. 154: Szene 07 - Shot 02



Abb. 157: Szene 08 - Shot 02

FAZIT

06



6.1 HERAUSFORDERUNGEN

VORWORT

Bei der Durchführung der Arbeit haben sich zahlreiche Probleme und Herausforderungen in verschiedenen Bereichen herausgestellt. Für eine

bessere Übersicht wurden dafür die Hürden thematisch voneinander getrennt. Dadurch ergeben sich folgende Bereiche:

KONZEPT

Auf konzeptioneller Ebene haben sich vor allem über die Zeit einige Probleme ergeben, da immer deutlicher wurde, dass der Umfang der Arbeit eine Person übersteigt.

Ansatz, den Einsatz von KI so umfassend wie möglich zu gestalten, wurde selbstverständlich verfolgt. Allerdings waren in bestimmten Bereichen die Technologien noch nicht hinreichend ausgereift, um einen signifikanten Mehrwert zu generieren. Im Rahmen der Recherche wurden diverse Technologien, wie die Generierung von 3D-Meshes und Texturen, untersucht. Die Ergebnisse waren jedoch zum jetzigen Stand unausgereift, sodass sie im weiteren Verlauf nicht berücksichtigt wurden. Im Fokus standen daraufhin die Bereiche Animation, Konzeption und Audio, welche in den ersten Tests überzeugende Ergebnisse lieferten.

In der Konsequenz mussten zwei geplante Szenen sowie wenige Shots verworfen werden, um ein vorzeigbares Produkt vorweisen zu können. Zudem konnten wenige Szenen nicht wie geplant umgesetzt werden. Dies machte eine Anpassung des Scripts sowie der mit der Handlung verwobenen visuellen Ausarbeitung erforderlich, um die gewünschte Geschichte dennoch innerhalb des geplanten Rahmens umzusetzen.

Eine weitere Herausforderung bestand in der Abwägung, in welchem Umfang künstliche Intelligenz in dieser Arbeit sinnvoll eingesetzt werden kann. Der

CLOTH

Nach einer gewissen Einarbeitungszeit in Marvelous Designer stellte sich die Erstellung von Kleidung sowie deren Überarbeitung in Blender als durchaus angenehm dar. Allerdings ergaben sich Schwierigkeiten bei der Anwendung des Weight-Paintings und des daraus resultierenden Clippings im Nackenbereich. Zur Erklärung, der Kopf und Körper eines Metahuman stellen jeweils separate Objekte dar. Verticies

des Körpermeshes, die unterhalb der Kleidung liegen, konnten ohne Probleme in Blender entfernt werden. Allerdings konnte das überarbeitete Kopf-Mesh nicht mehr in Unreal importiert werden, ohne Probleme mit den Facial Mocap Daten zu bekommen. Aus Zeitgründen musste dieses Ärgernis in Kauf genommen und das Clipping für diese Arbeit vorerst akzeptiert werden.

MOCAP

Die Aufbereitung der Mocap-Daten erwies sich als anspruchsvoller und zeitaufwändiger als ursprünglich angenommen. Der Einsatz von AI Motion Capturing erwies sich als vorteilhaft, da die grundlegenden Bewegungen bereits vorhanden waren. Eine Verwendung der Daten für das finale Produkt war ohne vorherige Überarbeitung jedoch nicht möglich. Gründe dafür waren Foot-Sliding, teilweise fehlendes oder schlechtes

Handtracking oder seltsame Positionierung in der Welt aufgrund der Perspektive. Diese Erkenntnisse waren bereits zu Beginn des Projektes sowie den ersten Tests vorhanden. Erst im finalen Einsatz wurde deutlich, wie aufwendig die Aufbereitung der Mocap-Daten war, um ein optimales Ergebnis zu erzielen. Dies resultierte in einer zeitlichen Herausforderung.

RENDERING

Das Rendering des Animationsfilms war ein besonders problematischer Arbeitsschritt während der Produktion.

Insbesondere die Metahumans wiesen eine hohe Fehleranfälligkeit auf. Aufgrund der wiederkehrenden Glitches, welche durch die standardmäßig aktivierte Haarsimulation in Kombination mit dem Sequencer verursacht wurden, wurde die Entscheidung getroffen, auf dieses Feature zu verzichten. Da eine Vielzahl von Nutzenden ebenfalls auf dieses Problem gestoßen sind, ist zu hoffen, dass eine Verbesserung in zukünftigen Versionen der Unreal Engine erfolgt.

Ein weiteres signifikantes Problem stellte das Rendering einiger Partikeleffekte dar. Zum Verständnis, jedes einzelne Partikelsystem verfügt über eigene zeitbasierte Simulationseinstellungen. Diesbezüglich standen die lokale Zeit des Rechners oder eine relative Zeit, welche während des Aufrufs des Partikelsystems berechnet wird, zur Auswahl. Da die MRQ den Realtime Viewport als zeitliche Basis nimmt und dadurch in Echtzeit abläuft, jedoch die einzelnen Bilder pro Frame durch das Postprocessing nicht mehr in Echtzeit gerendert werden, können

Probleme, wie bei Partikelsystemen, auftreten. Daher ist eine Anpassung der Simulationsparameter erforderlich, sodass die Zeit vom gerenderten Frame abhängt. Nach einiger Recherche konnte eine Kombination aus passenden Einstellungen sowie dem Lebenszyklus jedes einzelnen Partikelsystems im Sequencer auf den gewünschten Wert geforced werden. Dadurch wurden identische Resultate wie im Viewport erzielt.

Des Weiteren ergaben sich kleinere visuelle Herausforderungen, wie Schatten-Artefakte, Depth-of-Field-Bugs oder problematische Tessellation mit Nanite, welche eine umfangreiche Recherche und Tests erforderten. In den meisten Fällen konnten die Probleme durch Konsolenbefehle in der "DefaultEngine.ini" gelöst werden, welche zum Start des Projekts aufgerufen wird.

6.2 RESÜMEE

Das Ergebnis dieser Bachelorarbeit ist ein über zweiminütiger Animationsfilm, der den Weg eines Samurais thematisiert. Der Schwerpunkt lag dabei auf der Entwicklung verschiedener moderner Workflows, die mit dem Einsatz von künstlicher Intelligenz kombiniert wurden.

Dabei wurde untersucht, inwiefern KI in der Lage ist, Arbeitsschritte zu übernehmen, oder ob sie lediglich eine unterstützende Rolle einnimmt.

In einer umfassenden Recherche wurden etablierte sowie neuartige Workflows und (KI-)Tools ermittelt und hinsichtlich ihres Potenzials zur Bereicherung des Entwicklungsprozesses geprüft. Des Weiteren bot sich die Möglichkeit, sich mehr mit der japanischen Geschichte auseinanderzusetzen und dabei tiefere Einblicke in die Werte und Ideale eines Samurai zu gewinnen.

Auf konzeptioneller Ebene wurde das Schreiben einer Heldenreise mit dem Schwerpunkt Visual Storytelling thematisiert. Des Weiteren hatte die technische Machbarkeit der konzipierten Erzählung Einfluss auf den Schreibprozess. Dies ermöglichte es, die technische und gestalterische Konzeption eines Projektes in vollem Umfang kennenzulernen.

Die Umsetzung des Animationsfilms konnte unter Berücksichtigung zukunftsweisender Methoden und Werkzeuge hinsichtlich der Industrie in nahezu allen Bereichen abgedeckt werden. Der Einsatz von Künstlicher Intelligenz erfolgte lediglich in jenen Bereichen, in welchen die entsprechenden Technologien bereits so weit fortgeschritten waren, dass sie eine Bereicherung für den Entwicklungsprozess darstellten. Darüber hinaus konnten tiefe Einblicke in die technischen Möglichkeiten der Unreal Engine gewonnen und die Vorteile einer Echtzeit-Engine genutzt werden. Mithilfe von Tools wie Blender, Substance Painter, Marvelous Designer,

Speedtree, Cascadeur, EmberGen und MoveAI konnte eine umfangreiche 3D-Pipeline nach Unreal aufgebaut werden.

Auf persönlicher Ebene hat mir die Durchführung dieses Projekts eine unbeschreibliche Menge an Wissen gebracht, welches ich zu Beginn des Projekts in dieser Tiefe noch nicht hatte. Im Nachgang des Projektes fühle ich mich vorbereiteter und sicherer für das Arbeiten in der 3D-Industrie. Zusätzlich konnte ich meine Perspektive auf das Thema KI weiter vertiefen.

Für mich persönlich ist künstliche Intelligenz eine Technologie, die bleiben wird. Sie erleichtert uns das Leben in einigen Bereichen des Alltags. Auch die von mir erforschten Gebiete im 3D-Bereich weisen für die Zukunft ein hohes Potenzial auf. Im Moment sehe ich KI aber mehr als einen "Assistenten", der mir hilft, einzelne langwierige Prozesse zu vereinfachen und komplexe Sachverhalte greifbarer zu machen. Die einzelnen Puzzleteile muss der Mensch aber immer noch selbst zusammensetzen, denn oft steckt mehr dahinter als nur ein Algorithmus. Wir stehen noch zu sehr am Anfang, um Prognosen für die Zukunft abzugeben. Ich persönlich kann nur sagen, dass ich es nicht bereue, künstliche Intelligenz für dieses Projekt eingesetzt zu haben und ich bin gespannt, was die Zukunft bringen wird.

6.3 AUSBLICK

Das Projekt konnte im Wesentlichen gemäß dem Konzept und den angestrebten Zielen umgesetzt werden. Allerdings besteht in bestimmten Elementen noch Optimierungspotenzial, weshalb eine weitere Überarbeitung des Projektes angestrebt wird.

Die Erstellung eines Animationsfilms in diesem Umfang erfordert in der Regel ein umfangreicheres Wissen in den unterschiedlichen Bereichen sowie ein Team, um alle Phasen der Animationspipeline angemessen zu berücksichtigen um ein optimales Ergebnis zu erzielen. Im vorliegenden Fall waren aufgrund der Zeit gewisse Kompromisse erforderlich, um den gesamten Prozess eigenständig zu bewältigen.

Ein Kompromiss bestand in der Streichung zweier Szenen sowie weniger Shots, welche im Nachhinein den Film aufgerundet hätten. Außerdem sollen die vorhandenen Szenen überarbeitet werden, um die visuelle Darstellung glaubwürdiger zu gestalten. Diesbezüglich sind ebenfalls kleinere Elemente wie fliegende Vögel oder fallende Blätter im Hintergrund zu nennen.

Des Weiteren besteht Optimierungspotenzial hinsichtlich der Animationen. Mit mehr Zeit könnten diese noch realistischer zum Ausdruck kommen. Außerdem besteht die Möglichkeit, der Kleidung mehr Dynamik zu verleihen, indem sie mit dem Wind interagieren kann. Aus diesem Grund erscheint es sinnvoll, das Thema Realtime-Cloth-Simulation, das mit dem verwendeten Chaos-Cloth-System möglich ist, erneut aufzugreifen.

Die vorliegende Arbeit hat mich in meiner Entscheidung bestärkt, meine berufliche Laufbahn im 3D-Bereich fortzusetzen. Dabei ist es mein Ziel, meine Kenntnisse im Umgang mit der Unreal Engine zu vertiefen und weitere Workflows kennenzulernen, die in Kombination mit Unreal möglich sind. Mein nächstes Ziel ist der Berufseinstieg in einem 3D-Studio als Unreal Artist.

07 ZUSATZ



7.1 LITERATURVERZEICHNIS

[1] Youtube | 2009 | 20th Century Fox | Avatar Exclusive - Behind The Scenes (The Art of Performance Capture)

https://www.youtube.com/watch?v=P2_vB7zx_SQ

[Zugriff: 10.08.2024]

[2] IGN | 2024 | Frandesca Rivera | Breaking Down the Across the Spider-Verse LEGO Scene With the Teen Who Animated It

<https://www.ign.com/articles/spider-man-across-the-spider-verse-lego-scene-breakdown-easter-eggs-teen>

[Zugriff: 10.08.2024]

[3] Steam | 2024 | Black Myth: Wukong

https://store.steampowered.com/app/2358720/Black_Myth_Wukong/

[Zugriff: 31.08.2024]

[4] Steam | 2023 | Hogwarts Legacy

https://store.steampowered.com/app/990080/Hogwarts_Legacy/

[Zugriff: 31.08.2024]

[5] Epic Games | 2017 | Fortnite

<https://www.fortnite.com/>

[Zugriff: 31.08.2024]

[6] Unreal Engine | 2022 | How Sony Pictures Imageworks created a real-time thriller for Netflix's Love, Death + Robots

<https://www.unrealengine.com/en-US/spotlights/how-sony-pictures-imageworks-created-a-real-time-thriller-for-netflix-s-love-death-robots>

[Zugriff: 10.08.2024]

[7] Unreal Engine | 2024 | Magnopus brings Amazon's Fallout series to life with virtual production powered by Unreal Engine

<https://www.unrealengine.com/en-US/spotlights/magnopus-brings-amazon-s-fallout-series-to-life-with-virtual-production-powered-by-unreal-engine>

[Zugriff: 10.08.2024]

[8] Unreal Engine | 2024 | Jeff Farris | Forging new paths for filmmakers on "The Mandalorian"

<https://www.unrealengine.com/en-US/spotlights/magnopus-brings-amazon-s-fallout-series-to-life-with-virtual-production-powered-by-unreal-engine>

[Zugriff: 10.08.2024]

[9] Youtube | 2023 | Drama | Unrecord - Official Early Gameplay Trailer

<https://www.youtube.com/watch?v=5qvVNzsJyB0>

[Zugriff: 10.08.2024]

[10] Vimeo | 2021 | Sava Zivkovic | Irradiation

<https://vimeo.com/585797977>

[Zugriff: 10.08.2024]

[11] Youtube | 2022 | Move AI | Siggraph Talk Short - EA & Move.ai

<https://www.youtube.com/watch?v=z0aNKvZR8Tk>

[Zugriff: 10.08.2024]

[12] Gamesindustry | 2024 | Sophie McEvoy | EA believes over 50% of its development processes will be "positively impacted" by gen AI

<https://www.gamesindustry.biz/ea-believes-over-50-of-its-development-processes-will-be-positively-impacted-by-gen-ai>

[Zugriff: 10.08.2024]

[13] Wikipedia | 2024 | Epic Games

https://de.wikipedia.org/wiki/Epic_Games

[Zugriff: 11.08.2024]

[14] Pocketgamer | 2020 | Jon Jorden | Realpolitik and Tim Sweeney's epic vision

<https://www.pocketgamer.biz/realpolitik-tim-sweeney-epic-vision/>

[Zugriff: 11.08.2024]

[15] Statista | 2023 | J. Clement | Gross revenue generated by Epic Games worldwide from 2018 to 2026

<https://www.statista.com/statistics/1234106/epic-games-annual-revenue/>

[Zugriff: 11.08.2024]

[16] Slashdata | 2023 | Did you know that 60% of game developers use game engines?

<https://www.slashdata.co/post/did-you-know-that-60-of-game-developers-use-game-engines>

[Zugriff: 11.08.2024]

[17] Stepico | 2024 | Petro Prots | Games Made With Unreal Engine: What's All the Hype About?

<https://stepico.com/blog/games-made-with-unreal-engine/#:~:text=Unreal%20Engine%20seamlessly%20integrates%20real,an-d%20Bright%20Memory%3A%20Infinite>

[Zugriff: 11.08.2024]

[18] Unreal Engine | 2024 | Licensing Options

<https://www.unrealengine.com/en-US/licensing>

[Zugriff: 11.08.2024]

[19] Unreal Engine | 2024 | What is virtual production?

<https://www.unrealengine.com/en-US/explainers/virtual-production/what-is-virtual-production>

[Zugriff: 11.08.2024]

[20] Epic Games | 2024 | Unreal Engine 5

<https://www.unrealengine.com/en-US/unreal-engine-5>

[Zugriff: 11.08.2024]

[21] Unreal Engine | 2024 | Electric Dreams Environment Sample Project

<https://www.unrealengine.com/en-US/electric-dreams-environment>

[Zugriff: 11.08.2024]

[22] Quixel | 2019 | Teddy Bergsman Lind | Quixel joins forces with Epic Games

<https://quixel.com/blog/2019/11/12/quixel-joins-forces-with-epic-games>

[Zugriff: 11.08.2024]

[23] Unreal Engine | 2024 | MetaHuman - Realistic Person Creator

<https://www.unrealengine.com/en-US/metahuman>

[Zugriff: 11.08.2024]

[24] Blender Foundation | 2024 | Blender

<https://www.blender.org/>

[Zugriff: 12.08.2024]

[25] BlenderBrit | Blender is becoming an industry standard

<https://www.blenderbrit.co.uk/blog-1-1/blender-is-being-adopted-by-the-industry>

[Zugriff: 12.08.2024]

[26] Blender Foundation | 2024 | Blender Studio

<https://studio.blender.org/welcome/>

[Zugriff: 12.08.2024]

[27] Adobe | 2024 | Substance 3D

<https://www.adobe.com/de/creativecloud/3d-ar.html>

[Zugriff: 12.08.2024]

[28] Adobe | 2019 | Scott Belsky | Adobe Acquires Allegorithmic, the Leader in 3D Editing and Authoring for Gaming and Entertainment

<https://blog.adobe.com/en/publish/2019/01/23/adobe-acquires-allegorithmic-substance-3d-gaming>

[Zugriff: 12.08.2024]

[29] Adobe | 2024 | Substance 3D Painter 2023

https://store.steampowered.com/app/2718190/Substance_3D_Painter_2024/

[Zugriff: 12.08.2024]

[30] CLO Virtual Fashion | 2024 | CLO-SET
<https://www.clovirtualfashion.com/>
 [Zugriff: 12.08.2024]

[31] Nekki Limited | 2024 | Cascadeur
<https://cascadeur.com/>
 [Zugriff: 12.08.2024]

[32] Nekki Limited | 2024 | Cascadeur About
<https://cascadeur.com/about>
 [Zugriff: 12.08.2024]

[33] Unity Technologies | 2024 | Speedtree
<https://store.speedtree.com/>
 [Zugriff: 12.08.2024]

[34] Wikipedia | 2024 | Speedtree
<https://de.wikipedia.org/wiki/SpeedTree>
 [Zugriff: 12.08.2024]

[35] JangaFX | 2024 | JangaFX
<https://jangafx.com/>
 [Zugriff: 12.08.2024]

[36] Wikipedia | 2024 | Mongoleninvasionen in Japan
https://de.wikipedia.org/wiki/Mongoleninvasionen_in_Japan
 [Zugriff: 22.03.2024]

[37] Wikipedia | 2024 | Mongol invasions of Japan
https://en.wikipedia.org/wiki/Mongol_invasions_of_Japan
 [Zugriff: 22.03.2024]

[38] Wikipedia | 2023 | Schlacht von Bun'ei
https://de.wikipedia.org/wiki/Schlacht_von_Bun%E2%80%99ei
 [Zugriff: 22.03.2024]

[39] Youtube | 2020 | 4P - Das Spielmagazin | Ghost of Tsushima: Mongolen vs. Samurai | Geschichtsstunde
<https://www.youtube.com/watch?v=pMHjiClriuE&t=1251s>
 [Zugriff: 22.03.2024]

[40] Wikipedia | 2023 | Schlacht von Kōan
https://de.wikipedia.org/wiki/Schlacht_von_K%C5%8Dan
 [Zugriff: 22.03.2024]

[41] Wikipedia | 2024 | Samurai
<https://de.wikipedia.org/wiki/Samurai>
 [Zugriff: 21.03.2024]

[42] Youtube | 2022 | MrWissen2go Geschichte | Terra X | Die Wahrheit über Samurai
<https://www.youtube.com/watch?v=fDtNEcFxiMI&t=328s>
 [Zugriff: 21.03.2024]

[43] Wikipedia | 2024 | Bushidō
<https://de.wikipedia.org/wiki/Bushid%C5%8D>
 [Zugriff: 21.03.2024]

[44] OpenAI | 2024 | What is ChatGPT?
<https://help.openai.com/en/articles/6783457-what-is-chatgpt>
 [Zugriff: 16.08.2024]

[45] Hugging Face | 2024 | What is Text-to-Image?
<https://huggingface.co/tasks/text-to-image>
 [Zugriff: 16.08.2024]

[46] Studio Binder | 2024 | Kyle DeGuzman | What is a Mood Board - Definition, Examples & How They Work
<https://www.studiobinder.com/blog/what-is-a-mood-board-definition/>
 [Zugriff: 23.08.2024]

[47] Katharina Glück | 2022 | Die Heldenreise erklärt
<https://www.katharinagluck.de/post/die-heldenreise-erklart>
 [Zugriff: 23.08.2024]

[48] Studio Binder | 2024 | Alyssa Maio | What is a Storyboard? The Fundamentals to Get You Started
<https://www.studiobinder.com/blog/what-is-a-storyboard/>
 [Zugriff: 23.08.2024]

[49] Adobe | 2024 | Animatics: Definition und Technik
<https://www.adobe.com/de/creativecloud/animation/discover/animatics.html#:~:text=Eine%20Animatic%20ist%20ein%20animiertes,wie%20eine%20Sequenz%20aussehen%20wird>
 [Zugriff: 23.08.2024]

[50] Autodesk | 2024 | What is 3D Modeling & How Do You Use It?
<https://www.autodesk.com/solutions/3d-modeling-software>
 [Zugriff: 24.08.2024]

[51] Wikipedia | 2024 | Polygon mesh
https://en.wikipedia.org/wiki/Polygon_mesh
 [Zugriff: 24.08.2024]

[52] Autodesk | 2024 | What is 3D Modeling & How Do You Use It?
<https://www.autodesk.com/solutions/3d-modeling-software>
 [Zugriff: 24.08.2024]

[53] Pixune | Nazanin Shahbazi | 2024 | What Are the Differences Between 3D Modeling and Sculpting?
<https://pixune.com/blog/3d-modeling-vs-sculpting/>
 [Zugriff: 24.08.2024]

[54] Western Kentucky University | Joon Sung | Retopology
<https://people.wku.edu/joon.sung/edu/anim/3d/modeling/retopology/retopology.html>
 [Zugriff: 24.08.2024]

[55] Blender | 2024 | Manuel - Unwrapping
https://docs.blender.org/manual/en/2.79/editors/uv_image/uv_editing/unwrapping/index.html
 [Zugriff: 24.08.2024]

[56] Beyond Extent | 2023 | Timothy Dries | Texel Density
<https://www.beyondextent.com/deep-dives/deepdive-texeldensity>
 [Zugriff: 24.08.2024]

[57] Adobe | 2023 | What is Baking?
<https://helpx.adobe.com/substance-3d-bake/getting-started/what-is-baking.html>
 [Zugriff: 25.08.2024]

[58] Adobe | 2024 | What is 3D texturing?
<https://www.adobe.com/products/substance3d/discover/3d-texturing.html>
 [Zugriff: 25.08.2024]

[59] Maarten Hof | 2023 | What Is Channel Packing? How To Set It Up In Substance Painter
<https://maartenhof.com/what-is-channel-packing-and-how-to-set-it-up-with-substance-painter/>
 [Zugriff: 25.08.2024]

[60] Unreal Engine | 2024 | Documentation - Materials
<https://dev.epicgames.com/documentation/en-us/unreal-engine/unreal-engine-materials>
 [Zugriff: 05.05.2024]

[61] Unreal Engine | 2024 | Documentation - Using Material Layers
https://dev.epicgames.com/documentation/en-us/unreal-engine/using-material-layers-in-unreal-engine?application_version=5.0
 [Zugriff: 03.05.2024]

[62] Ryan DowlingSoka | 2022 | Triplanar, Dithered Triplanar, and Biplanar Mapping in Unreal
<https://ryandowlingsoka.com/unreal/triplanar-dither-biplanar/>
 [Zugriff: 07.05.2024]

[63] Nvidia | 2007 | R. Steven Glanville | Developer - Chapter 20. Texture Bombing
<https://developer.nvidia.com/gpugems/gpugems/part-iii-materials/chapter-20-texture-bombing>
 [Zugriff: 08.05.2024]

[64] Unreal Engine | 2022 | Astaraa | Community - Guide to Virtual Textures for Noobs
<https://dev.epicgames.com/community/learning/tutorials/58vb/unreal-engine-guide-to-virtual-textures-for-noobs>
 [Zugriff: 03.05.2024]

[65] Youtube | 2022 | Game Dev Academy | QUICK TRICK to Realistically BLEND ASSETS with LANDSCAPE | UE5 Runtime Virtual Texturing Tutorial Noobs
<https://www.youtube.com/watch?v=ATfzfvtflo>
 [Zugriff: 03.05.2024]

[66] Youtube | 2021 | Terra X History | Kann man die Natur vermessen? | Harald Lesch | Ganze Folge | Terra X
<https://www.youtube.com/watch?v=UOLqWRPk8-4>
 [Zugriff: 26.08.2024]

[67] Unreal Engine | UDK - World Position Offset
<https://docs.unrealengine.com/udk/Three/WorldPositionOffset.html>
 [Zugriff: 26.08.2024]

[68] Medium | 2020 | Luiz Otavio Vasconcelos | Texture Animation: Applying Morphing and Vertex Animation Techniques
<https://medium.com/tech-at-wildlife-studios/texture-animation-techniques-1daecb316657>
 [Zugriff: 26.08.2024]

[69] Unreal Engine | 2024 | Electric Dreams Environment Sample Project
<https://www.unrealengine.com/en-US/electric-dreams-environment>
 [Zugriff: 25.08.2024]

[70] Wikipedia | 2023 | Level of Detail
https://de.wikipedia.org/wiki/Level_of_Detail
 [Zugriff: 27.08.2024]

[71] Unreal Engine | 2024 | Documentation - Nanite Virtualized Geometry
<https://dev.epicgames.com/documentation/en-us/unreal-engine/nanite-virtualized-geometry-in-unreal-engine>
 [Zugriff: 27.08.2024]

[72] Youtube | 2024 | Unreal Engine | An Artist's Guide to Using Nanite Tessellation | Unreal Fest 2024
<https://www.youtube.com/watch?v=6igUsOp8FdA>
 [Zugriff: 27.08.2024]

[73] Unreal Engine | 2024 | Documentation - Lumen Global Illumination and Reflections
<https://dev.epicgames.com/documentation/en-us/unreal-engine/lumen-global-illumination-and-reflections-in-unreal-engine>
 [Zugriff: 27.08.2024]

[74] Unreal Engine | 2024 | Documentation - Directional Lights
https://dev.epicgames.com/documentation/en-us/unreal-engine/directional-lights?application_version=4.27
 [Zugriff: 27.08.2024]

[75] Unreal Engine | 2024 | Documentation - Sky Light
https://dev.epicgames.com/documentation/en-us/unreal-engine/sky-light?application_version=4.27
 [Zugriff: 27.08.2024]

[76] Unreal Engine | 2024 | Documentation - Sky Atmosphere
https://dev.epicgames.com/documentation/en-us/unreal-engine/sky-atmosphere?application_version=4.27
 [Zugriff: 27.08.2024]

[77] Unreal Engine | 2024 | Documentation - Volumetric Clouds
https://dev.epicgames.com/documentation/en-us/unreal-engine/volumetric-clouds?application_version=4.27
 [Zugriff: 27.08.2024]

[78] Unreal Engine | 2024 | Documentation - Volumetric Fog
<https://dev.epicgames.com/documentation/en-us/unreal-engine/volumetric-fog-in-unreal-engine>
 [Zugriff: 27.08.2024]

[79] Wikipedia | 2024 | Japanese clothing
https://en.wikipedia.org/wiki/Japanese_clothing
 [Zugriff: 09.04.2024]

[80] School of Motion | EJ Hassenfratz | What is 3D Rigging for Animation?
<https://www.schoolofmotion.com/blog/what-is-3d-rigging-for-animation>
 [Zugriff: 28.08.2024]

[81] Blender | 2024 | Documentation - Weight Paint
https://docs.blender.org/manual/en/latest/sculpt_paint/weight_paint/introduction.html
 [Zugriff: 28.08.2024]

[82] Unreal Engine | 2023 | Tim Brakensiek | Community - Panel Cloth Editor
<https://dev.epicgames.com/community/learning/tutorials/pv7x/unreal-engine-panel-cloth-editor>
 [Zugriff: 28.08.2024]

[83] Wikipedia | 2024 | Animation
<https://de.wikipedia.org/wiki/Animation>
 [Zugriff: 28.08.2024]

[84] Wikipedia | 2024 | Motion Capture
https://de.wikipedia.org/wiki/Motion_Capture
 [Zugriff: 28.08.2024]

[85] Medium | 2022 | NVIDIA Omniverse | What is Animation Retargeting?
<https://medium.com/@nvidiaomniverse/what-is-animation-retargeting-4aadab383032>
 [Zugriff: 28.08.2024]

[86] Youtube | 2024 | Jobutsu | UE5.4: Game-Changing Rigging Update (You Won't Believe How Easy This Is)
<https://www.youtube.com/watch?v=eed8rKcA44Q>
 [Zugriff: 28.08.2024]

[87] Autodesk | 2007 | What are blend shapes?
https://download.autodesk.com/us/maya/docs/Maya85/wwhelp/wwhimpl/common/html/wwhelp.htm?context=Animation&file=What_are_blend_shapes.html
 [Zugriff: 28.08.2024]

[88] JangaFX | Said Al Attrach | VDB: A Deep Dive
<https://jangafx.com/insights/vdb-a-deep-dive>
 [Zugriff: 29.08.2024]

7.2 ABBILDUNGSVERZEICHNIS

Abb. 1: Avatar - Behind The Scenes
https://www.youtube.com/watch?v=P2vB7zx_SO
 [Zugriff: 10.08.2024]

Abb. 2: Vierzehnjähriger - Across the Spider-Verse
<https://assets-prd.ignimgs.com/2023/09/06/spider-verse-lego-scene-00-00-22-22-still002-1693958464833.jpg>
 [Zugriff: 10.08.2024]

Abb. 3: Black Myth: Wukong - Unreal Engine
<https://www.techspot.com/images2/news/bigimage/2024/08/2024-08-28-image-26-i-1100.webp>
 [Zugriff: 20.08.2024]

Abb. 4: Mandalorian - Virtual Production
<https://ruef.com/wp-content/uploads/2020/10/the-mandalorian-virtual-backgrounds.jpg>
 [Zugriff: 10.08.2024]

Abb. 5: MoveAI - Mocap durch KI

Abb. 6: Unreal Engine 5 | Next-Gen Demo
https://cdn2.unrealengine.com/Unreal+Engine%2Fblog%2Fa-first-look-at-unreal-engine-5%2FUnreal_Engine_5_Nanite-2491x1401-110edc43c67b7d81be869bf703cb40a72e557c96.jpg?resize=1&w=1920
 [Zugriff: 10.08.2024]

Abb. 7: Cyberpunk Project

Abb. 8: Amethyst Island

Abb. 9: Modern House Interior

Abb. 10: Alien VR

Abb. 11: Runa

Abb. 12: Welten Saga II - Olympia
https://faber-courtial.de/wp-content/uploads/2024/05/WKEB_203_08OL_001_030_Olympia_02-scaled.jpg
 [Zugriff: 12.07.2024]

Abb. 13: Welten Saga II - Olympia
https://faber-courtial.de/wp-content/uploads/2024/05/WKEB_203_08OL_001_020_Olympia_01-scaled.jpg
 [Zugriff: 12.07.2024]

Abb. 14: Welten Saga II - Cusco
https://faber-courtial.de/wp-content/uploads/2024/05/WKEB_204_07CS_001_010_Cusco_03-scaled.jpg
 [Zugriff: 12.07.2024]

Abb. 15: Unreal Engine Logo

Abb. 16: Matrix Awakens - Unreal Engine Technical Demo
<https://cdn2.unrealengine.com/body-4-1920x1080-frosty-blog-1920x1080-4f06882352c5.png?resize=1&w=1920>
 [Zugriff: 12.08.2024]

Abb. 17: Blender Logo

Abb. 18: Substance Painter Logo

Abb. 19: Marvelous Designer Logo

Abb. 20: Cascadeur Logo

Abb. 21: Speedtree Logo

Abb. 22: EmberGen Logo

Abb. 23: Kublai Khan
https://de.wikipedia.org/wiki/Kublai_Khan#/media/Datei:YuanEmperorAlbumKhubilaiPortrait.jpg
 [Zugriff: 13.08.2024]

Abb. 24: Brief an das Shogunat
<https://upload.wikimedia.org/wikipedia/commons/e/e2/LetterFromKhubilaiToJapan1266.jpg>
 [Zugriff: 13.08.2024]

Abb. 25: Göttliche Winde - Kamikaze
[https://de.wikipedia.org/wiki/Mongoleninvasionen_in_Japan#/media/Datei:Kikuchi_Yoosai_-_Mongol_Invasion_\(m%C5%8Dko_sh%C5%ABrai\)_-_Tokyo_National_Museum.jpg](https://de.wikipedia.org/wiki/Mongoleninvasionen_in_Japan#/media/Datei:Kikuchi_Yoosai_-_Mongol_Invasion_(m%C5%8Dko_sh%C5%ABrai)_-_Tokyo_National_Museum.jpg)
 [Zugriff: 13.08.2024]

Abb. 26: Mongolen Invasion - Karte
<https://www.welt-der-samurai.de/pics/karte-mongoleninvasion.gif>
 [Zugriff: 13.08.2024]

Abb. 27: Samurai entern mongolische Schiffe
https://upload.wikimedia.org/wikipedia/commons/2/2a/Takezaki_suenaga_ekotoba3.jpg
 [Zugriff: 13.08.2024]

Abb. 28: Schlacht an der Hakata-Bucht
<https://japanlink.de/wp-content/uploads/1999/05/kamikaze.jpg>
 [Zugriff: 13.08.2024]

Abb. 29: Japanischer Samurai nimmt Tributzahlungen von Unterworfenen
https://de.wikipedia.org/wiki/Samurai#/media/Datei:Samurai_and_Ainu_Fuzoku_Ema.jpg
 [Zugriff: 13.08.2024]

Abb. 30: Generierte 3D-Assets im Vergleich - Unbearbeitet mit Wireframe
<https://www.closeup.de/kaufen/guardians-of-the-galaxy-2-lifesize-foam-figur-groot-95440>
 [Zugriff: 16.08.2024]

Abb. 31: Generiertes Facemesh - Hyperhuman ChatAvatar

Abb. 32: Beispiel für AI-Mocap mit MoveAI

Abb. 33: Ghost of Tsushima - Concept Art
<https://www.flickr.com/photos/playstationblogeurope/3801172302/in/album-72157662068619388/>
 [Zugriff: 20.03.2024]

Abb. 34: Ghost of Tsushima - Fan Art
<https://florentlebrun.artstation.com/projects/J9Jzzz>
 [Zugriff: 20.03.2024]

Abb. 35: Ghost of Tsushima - Concept Art
<https://www.flickr.com/photos/playstationblogeurope/38011708392/in/album-72157662068619388/>
 [Zugriff: 20.03.2024]

Abb. 36: Ghost of Tsushima - Concept Art
<https://www.artstation.com/artwork/J9rVEm>
 [Zugriff: 20.03.2024]

Abb. 37: Darstellung eines herrenlosen Samurai - Ronin
https://www.reddit.com/r/wallpaper/comments/13p41cn/ghost_of_tsushima_3840x2160/#lightbox
 [Zugriff: 21.03.2024]

Abb. 38 - 47: Storyboard

Abb. 48: Geometrie eines Hauses

Abb. 49: Sculpting eines Grabsteins

Abb. 50: UV´s eines Würfels anhand eines Checker-Pattern

Abb. 51: UV-Mapping des Gesichts eines MetaHumans mit einem Checker-Pattern

Abb. 52: Normal Map Baking am Beispiel eines Grabsteins

Abb. 53: Texturierter Grabstein

Abb. 54: Texture 1 - Albedo

Abb. 55: Texture 2 - Packed

Abb. 56: Texture 3 - Normal

Abb. 57: Schritt 1 - Grundlegende Textureslots

Abb. 58: Schritt 2 - UV-Einstellungen (Tiling, Rotation, Offset)

Abb. 59: Schritt 1 & 2 - Ergebnis

Abb. 60: Schritt 3 - Anpassungen der Textureigenschaften

Abb. 61: Schritt 4 - Surface Imperfections

Abb. 62: Schritt 5 - Verschmutzungen (Dirt-Layer)

Abb. 63: Megascans in einer Szene

Abb. 64: Logo von Gaea

Abb. 65: Beispiel eines mit Gaea erstellten Terrains

Abb. 66: Rendering eines Terrains mit einem Auto-Landscape-Material

Abb. 67: Einstellungen des Auto-Landscape-Materials

Abb. 68 - 71: Landscape-Auto-Material

Abb. 72: Blending zwischen Landscape und Szenengeometrie mittels RVT

Abb. 73: Node-Struktur eines Baumes

Abb. 74: Aufbau eines Baumes

Abb. 75: Speedtree Einstellungsmöglichkeiten

Abb. 76: Beispiel eines erstellten Hero-Trees

Abb. 77: Blätter-Atlas des zuvorigen Baumes (ungefärbt)

Abb. 78: Mesh-Cutout

Abb. 79: Animiertes Grass Material

Abb. 80: Electric Dreams Demo
<https://cdna.artstation.com/p/assets/images/images/063/399/954/4k/jakob-keudel-opal-marketing-heromonolith-001b-png-copy.jpg?1685452962>
[Zugriff: 25.08.2024]

Abb. 81 - 90: Generierung eines Waldes

Abb. 91: Einsatz eines komplexen PCG-Systems

Abb. 92: Darstellung von Nanite-Szenengeometrie - The Matrix Awakens
<https://youtu.be/WU0qvPcc3jQ?si=LiEhKaMh3GAnhcX-&t=52>
[Zugriff: 25.08.2024]

Abb. 93: Beispiel für Tessellation auf einem Landscape

Abb. 94: Lumen im Einsatz

Abb. 95: Directional Light und Sky Light

Abb. 96: Sky Atmosphere und Volumetric Clouds

Abb. 97: Volumetric Fog

Abb. 98: Traditionelle japanische Kleidung
https://factsanddetails.com/media/2/20090810-clothes%20intoa_02.gif
[Zugriff: 28.00.2024]

Abb. 99: Teil der Ootuki Familie, 1874
https://en.wikipedia.org/wiki/Japanese_clothing#/media/File:Ootuki_family_1874.jpg
[Zugriff: 28.00.2024]

Abb. 100, 102 - 104: The Art of Ghost of Tsushima / Dark Horse Comics / 2020 / ISBN-13: 978-1506713557 / Seite 14, 15, 20, 41

Abb. 101: Ghost of Tsushima - Outfits
<https://cdnb.artstation.com/p/assets/images/images/029/222/977/large/eunhye-lee-ronin-armor-el-dyes.jpg?1596829439>
[Zugriff: 09.04.2024]

Abb. 105: Aoi als Jugendlicher und Erwachsener

Abb. 106: MetaHumans für die Crowds

Abb. 107: Mesh to MetaHuman

Abb. 108: MetaHuman Animator

Abb. 109: MetaHuman Control-Rig

Abb. 110: Weight Painting MetaHuman | Spine 2 - 4

Abb. 111: Kombiniertes Körper des MetaHuman

Abb. 112: Schnittmuster

Abb. 113: Sewing

Abb. 114: Eigenschaften am Beispiel eines Kimonos

Abb. 115: Fertig simulierte Kleidung

Abb. 116: Aufbereitete Kleidung

Abb. 117: Finale texturierte Kleidung

Abb. 118: Bestandteile des Chaos Cloth Systems

Abb. 119: The Animator's Survival Kit von Richard Williams / Farrar und Faber / 2009 / ISBN-13: 978-0571238330 / Seite 64 - 66

Abb. 120: Keyframe-Animation

Abb. 121: Optisch - Anzug mit Markern, die von Kameras erfasst werden
<https://cdn.britannica.com/36/254436-050-039C8FDE/optical-motion-capture-system-demonstrated-by-postgraduate-students-sam-pike-and-jake-teale-of-falmouth-university-south-west-england-2015.jpg>
[Zugriff: 13.07.2024]

Abb. 122: AI - Markerlose Aufnahmen durch KI basierte Algorithmen
https://beforesandafters.com/wp-content/uploads/2023/04/IMG_5138.jpg
[Zugriff: 13.07.2024]

Abb. 123: Inertial - Sensoren am Körper, die die Bewegung messen
https://3dvi.com/wp-content/uploads/2021/10/SSP2_Still_4.jpg
[Zugriff: 13.07.2024]

Abb. 124: Move AI Mocap

Abb. 125: Beispiel für das Remapping von Bones - Cascadeur

Abb. 126: Unterschiedliche Skelette mit derselben Animation (Retargeting)

Abb. 127: Abb. 127: Überarbeitete Mocap Daten für die Trainingsszene

Abb. 128: MetaHuman Facial Control Rig

Abb. 129: Verschiedene Mimiken durch manipulieren der Blendshapes

Abb. 130: Erstellung einer MetaHuman Identity

Abb. 131: Facial Mocap Performance

Abb. 132: Feuersimulation in EmberGen

Abb. 133: Feuersimulation als Flipbook

Abb. 134: Feuersimulation als VDB

Abb. 135: Niagara Emitter mit Modulen

Abb. 136: Falling Leave System

Abb. 137: Movie Scene Capture und Movie Render Queue im Vergleich

Abb. 138: Rendersettings

Abb. 139: Udio Weboberfläche

Abb. 140 - 157: Finale Renderings

7.3 ABKÜRZUNGSVERZEICHNIS

AAA = Produktion mit großem Budget

VFX = Visuelle Effekte

KI/AI = Künstliche Intelligenz

UE = Unreal Engine

NPC = Nicht spielbarer Charakter

7.4 GENDERVERMERK

In der vorliegenden Bachelorarbeit wird im Sinne der Inklusivität eine geschlechtergerechte oder geschlechtsneutrale Sprache verwendet, wo immer dies möglich ist.

Im Fall der Verwendung des generischen Maskulinums, soll dies keine Wertung oder Benachteiligung des weiblichen oder anderweitigen Geschlechtern implizieren. Die gewählten Formulierungen dienen ausschließlich dem Textfluss und der Lesbarkeit.

7.5 DEEPL DISCLAIMER

Die in der Arbeit von mir verfassten Texte wurden mit DeepL Write auf Stil und Rechtschreibung geprüft und, wo sinnvoll, durch DeepL Write verbessert.

<https://www.deepl.com/de/write>

7.6 EIDESSTATTLICHE VERSICHERUNG

Hiermit versichere ich, dass diese Abschlussarbeit von mir persönlich verfasst ist und dass ich keinerlei fremde Hilfe in Anspruch genommen habe. Ebenso versichere ich, dass diese Arbeit oder Teile daraus weder von mir selbst noch von anderen als Leistungsnachweise andernorts eingereicht wurden. Wörtliche oder sinngemäße Übernahmen aus anderen Schriften und Veröffentlichungen in gedruckter oder elektronischer Form sind gekennzeichnet. Sämtliche Sekundärliteratur und sonstige Quellen sind nachgewiesen und im Literaturverzeichnis aufgeführt. Das Gleiche gilt für graphische Darstellungen und Bilder sowie für alle Internet-Quellen. Ich bin ferner damit einverstanden, dass meine Arbeit zum Zwecke eines Plagiats Abgleichs in elektronischer Form anonymisiert versendet und gespeichert werden kann. Mir ist bekannt, dass von der Korrektur der Arbeit abgesehen werden kann, wenn die Erklärung nicht erteilt wird.

Ort und Datum

Unterschrift



RISE OF A SAMURAI
VON ALEXANDER BAUR