



Technische Hochschule Ostwestfalen-Lippe

University of Applied Sciences and Arts

Fachbereich 2 - Medienproduktion

Bachelorarbeit zum Thema:

Virtual Production für Animationsfilme in der Unreal Engine

Eine Zusammenfassung der angewendeten Virtual Production Techniken für den Animationsfilm Mr. Sandman(AT)

Vorgelegt von:

Dennis Prüßmeier

Matrikelnummer: 15417063

BA Medienproduktion

Erstprüfer: Prof. Dipl.-Ing. Rico Dober

Zweitprüferin: B. DES. Jennifer Meier

Eingereicht am: 08.02.2023

Lizenz: CC-BY 4.0

Inhaltsverzeichnis

1. Glossar	4
2. Einleitung	7
2.1 Ziel der Arbeit	7
2.2 Sandman Projekt	7
2.2.1 Die Story	7
2.2.2 Praktische Arbeit am Animationsfilm „Mr. Sandman (AT)“	8
2.3 Aufbau der Arbeit	12
3. Definition „Virtual Production“	13
4. Virtual Production für „Mr. Sandman (AT)“	14
4.1 Environments	14
4.1.1 Organisation	14
4.1.2 Landscape Actor Erstellung	15
4.1.3 Gestaltung des Landscapes	20
4.1.4 Landscape Material	21
4.1.5 Importierung von Assets	26
4.1.6 Nanite	27
4.1.7 Foliage Mode	27
4.1.8 Atmosphere Basis Setup	29
4.1.9 Lumen	30
4.1.10 Licht Typen	31
4.1.11 Post Process Volumne	32
4.1.12 Basiseinstellungen der Lichter	33
4.1.13 Light Shafts	34
4.2 Motion Capture	35
4.2.1 Optische Systeme	35
4.2.2 Nicht optische Systeme	37
4.2.3 Motion Capture im Projekt	38
4.3 Virtuelle Kamera und Kamera Tracking	42
4.3.1 Virtuelle Kamera im Projekt	43

5. Ergebnisse der Techniken	46
5.1 Ergebnisse der Environments	46
5.2 Ergebnisse des Motion Captures	49
5.3 Ergebnisse des Kamera Tracking und der Virtuellen Kamera	50
5.4 Fazit	51
6. Abbildungsverzeichnis	52
7. Literaturverzeichnis	54

1. Glossar

Global Illumination

Global Illumination ist das Ergebnis der Kombination von direkten und indirekten Licht. Direktes Licht beschreibt dabei nur den direkten Effekt, den Lichtquellen auf Objekte haben, indirektes Licht hingegen ist das Licht, was durch Reflexion des direkten Lichtes entsteht und dabei die Beleuchtung anderer Objekte beeinflusst (tu-ilmenau, o. D.).

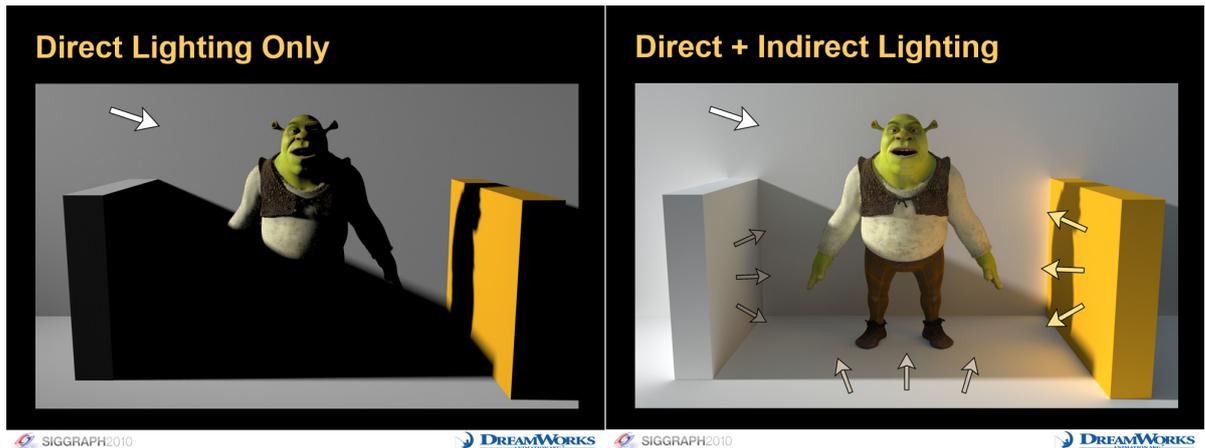


Abbildung 1: Global Illumination - Quelle: colinbarrebrisebois.com

LOD (Level of Detail)

LOD beschreibt den Detailgrad von 3D Modellen. In Videospielen und auch in Animationsfilmen werden verschiedene Detailgrade von Modellen eingesetzt, um die Performance zu steigern. Dafür wird der Detailgrad von weiter entfernten und kleineren Modellen in Abstufungen reduziert (vgl. rastergrid, o. D.). Dies passiert durch Reduzierung der Polygonanzahl des Modelles.

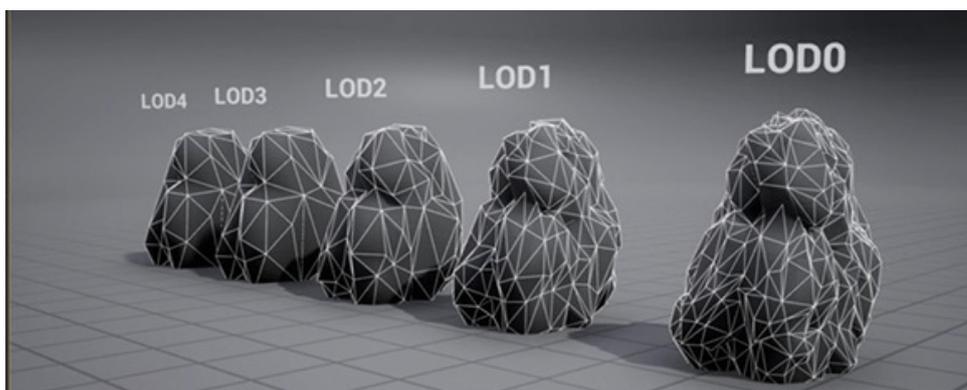


Abbildung 2: LOD Stufen - Quelle: 3d1.com.br

Photogrammetrie

„*Photogrammetrie ist ein Verfahren, um aus Fotos 3D Modelle zu erstellen*“

(Photogrammetrie Software, o. D). Dabei werden aus einer Vielzahl von überlappenden Fotos eines Objektes, 2D- und 3D-Daten extrahiert. Durch die gesammelten Daten wird durch die Verwendung von Software ein 3D-Modell hergestellt (vgl. Photogrammetrie Software, o. D). Dafür muss das Objekt von allen Seiten fotografiert werden.

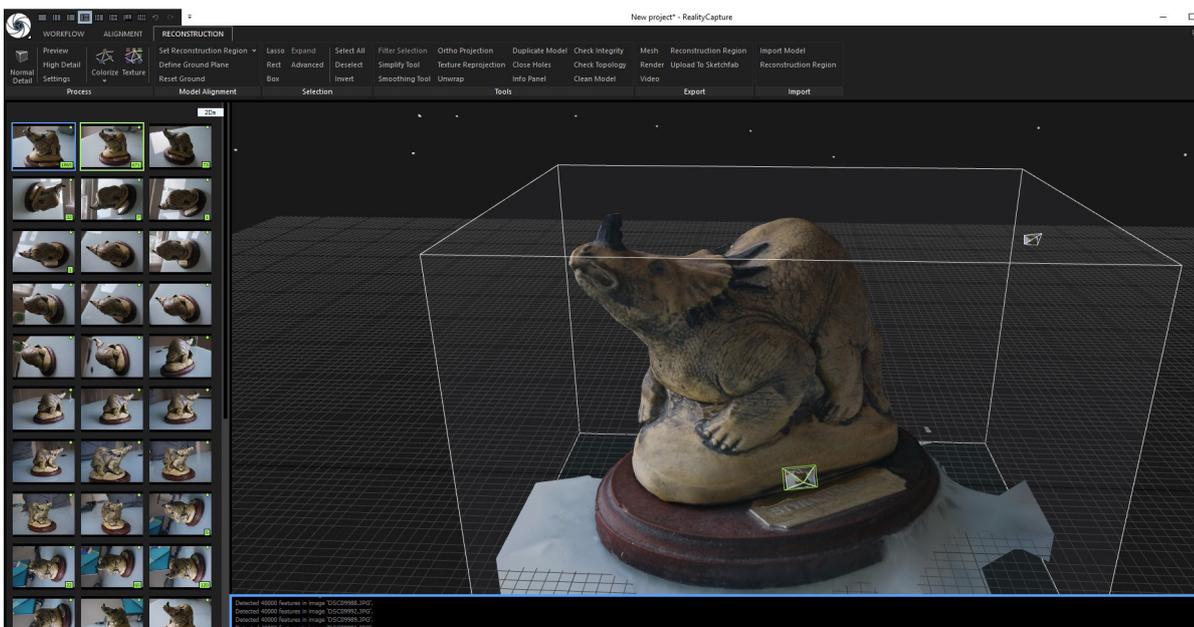


Abbildung 3: RealityCapture - Quelle: peterfalkingham.com

Previsualisierungsphase

Die Previsualisierungsphase (Previs) ist eine der ersten Phasen des Projektes.

In der Previs des Virtual Production Projektes können Filmemacher mit verschiedenen Inszenierungs- und Art-Direction-Optionen wie Beleuchtung, Kameraplatzierung und -bewegung, Bühnenregie und Schnitt experimentieren ohne die erhöhten Kosten der eigentlichen Produktion (vgl. Kadner, 2019, S. 12).

Die Phase erlaubt bereits im frühen Produktionsprozess eine Zusammenarbeit zwischen den Abteilungen und ermöglicht es, fundierte Entscheidungen bezüglich Inszenierungs- und Art-Direction-Optionen zu treffen (vgl. Clark; Okada; Reisner; Stump; Edlund; Levinson et al., 2008).

Riggen

Riggen beschreibt den Vorgang, ein Objekt mit einem System, dem sogenannten Rig, auszustatten, um Translationen und Rotationen des Objektes zu animieren.

Dabei kann ein Rig verschiedene Formen annehmen. Das Rig eines menschlichen Charakters ist vergleichbar mit dem Aufbau eines Skelettes. Im Charakter werden Knochen (Bones) und Gelenke (Joints) platziert, die miteinander verknüpft werden (vgl. Jablotschkin, 2021, S. 12f).

Die Geometrie des Charakters wird an die angelegte Knochenstruktur gebunden. Die Gelenke können anschließend durch verknüpfte Griffe (Controllers) gesteuert werden. Durch die Benutzung von Deformern wird die Geometrie des Charakters bei Bewegung der Gelenke durch die Controller passend verformt (vgl. Beane, 2012, S. 177f).

Unreal Engine

„Die Unreal Engine ist ein komplettes Paket von Entwicklungswerkzeugen für die Spieleentwicklung, Architektur- und Automobilvisualisierung, die Erstellung linearer Film- und Fernsehinhalte, die Produktion von Übertragungen und Live-Events, Schulungen und Simulationen sowie andere Echtzeitanwendungen.“ (Frequently Asked Questions, o. D.)

Die Engine ist dabei Open Source und bis zu 1 Million USD Ertrag kostenfrei für private wie auch kommerzielle Projekte nutzbar (vgl. Frequently Asked Questions, o. D.).

Unreal Engine Actors

Actors beschreiben alle Objekte, die im Level platziert werden können.

Dazu zählen zum Beispiel Kameras oder statische Meshes. Actors unterstützen dabei alle gängigen Transformationen wie Translationen in X, Y und Z-Richtung, Rotationen und Skalierungen (vgl. Actors, o. D.).

Hinweis zur Gleichberechtigung

In dieser Arbeit wird aus Gründen der besseren Lesbarkeit das generische Maskulinum verwendet. Weibliche und anderweitige Geschlechteridentitäten werden dabei ausdrücklich mitgemeint.

2. Einleitung

Bahnbrechende technologische Fortschritte im Laufe der Jahre haben die Entwicklung des Filmemachens verändert und erweitert (vgl. Kadner, 2019, S. 3).

So kommen bei Filmproduktion vermehrt Virtual Production Methoden zum Einsatz (vgl. Reinschmidt, 2023, S. 35).

2.1 Ziel der Arbeit

Die Zielsetzung dieser Arbeit ist es, Einsteigern im Thema „Virtual Production“ (VP) und auch kleineren Teams mit einem niedrigen Budget einen Einblick in die Verwendung von diversen VP Techniken, die in der Herstellung von Animationsfilmen, Videospielsequenzen oder anderen Animationswerken angewendet werden, zu geben.

Dazu bezieht sich die Arbeit hauptsächlich auf die im Animationsfilm Mr. Sandman(AT) verwendeten Techniken unter der Verwendung der von „Epic Games“ entwickelten „Unreal Engine“ und sieht aufgrund des Umfangs, Grundkenntnisse in den Bereichen Film, Animation, 3D Modellierung, Texturierung und der Benutzeroberfläche der Unreal Engine vor. Es wird untersucht, inwiefern die Unreal Engine einen einfachen Einstieg in die Virtual Production für Animationswerke ermöglicht und wo es gegebenenfalls Einschränkungen in der Anwendung der Techniken gibt.

2.2 Sandman Projekt

2.2.1 Die Story

„Der Sandmann ist als freundliche Figur bekannt, die kleinen Kindern Gute-Nacht-Geschichten erzählt. Doch dies war nicht immer so.

In dem Motion-Capture-Animationsfilm „Mr. Sandman (AT)“ wird die Herkunft des Sandmannes in einem düsteren Licht erzählt.

Der Sandmann verkörpert den Schlaf und ist der Zwillingsbruder vom Tod. Der Sandmann entführt Kinder, die nicht schlafen können, in den dunklen Märchenwald. Die beiden Brüder stehen unter der Macht ihrer Schwester Ker, die sich von den Seelen der Kinder ernährt.

Eines Tages entführt der Sandmann die junge Mia in den Märchenwald. Doch Mia ist ganz anders, als die anderen Kinder, die der Sandmann bisher entführt hat und er fängt an seine Taten zu hinterfragen. Als die Schatten des Märchenwaldes nach der jungen Seele von Mia suchen, kommt Hilfe aus unerwarteter Stelle für den Sandmann und Mia: Frau Holle versteckt Mia und zeigt dem Sandmann einen anderen, besseren Weg.

Ein Kampf entsteht zwischen den drei Geschwistern, die dunklen Mächte werden besiegt und der Sandmann wird zu der guten Figur, die wir kennen und den Kindern einen angenehmen Schlaf beschert“ (Holzweißig, 2022).

2.2.2 *Praktische Arbeit am Animationsfilm „Mr. Sandman (AT)“*

Die praktische Arbeit des Projektes beschäftigt sich mit der Anfertigung des Animationsfilmes „Mr. Sandman(AT)“. Dabei wird das Projekt in einem Team aus fünf Personen durch die Verwendung von mehreren Virtual Production Methoden durchgeführt.

Wie Jan Andres Reinschmidt in seiner Arbeit über die Möglichkeiten zur kosteneffizienten Umsetzung von Virtual Production schreibt, war eine klare Departmentaufteilung aufgrund des Umfangs und der Teamgröße nur schwer möglich. So musste jedes Teammitglied in verschiedenste Aufgabenbereiche einspringen (vgl. Reinschmidt, 2023, S. 37).

Im Folgenden werden Bereiche und Aufgaben beschrieben, in denen ich maßgeblich beteiligt war.

- **Technik**

Unter dieser Kategorie fallen alle Hardwareverknüpfungen wie Kabelverbindungen, Installation von Software wie SteamVR oder der für die Kalibrierung des VIVE Systemes notwendigen Software. Ebenso fällt unter diese Kategorie die Kalibrierung der von Captury gelieferten Kameras und Software. Sowie das Kalibrieren des VIVE Systemes für die Benutzung von Markern und des VR Headsets.

Ein weiterer Abschnitt des Technikbereiches ist die Behebung von Software wie auch Treiberfehler, die im Laufe des Projektes auftreten. Da der Linux-Rechner, auf dem die von Captury gelieferte Software betrieben wird, durch eine Netzwerkverbindung mit dem Windows Rechner, auf dem die Unreal Engine betrieben wird, verbunden werden muss, ist eine weitere wichtige Aufgabe die Herstellung einer Netzwerkverbindung und wenn nötig, die Problembehandlung bei auftretenden Schwierigkeiten.

- **Charakterdesign (Tod)**

In der Geschichte des Filmes treten mehrere Charaktere auf, die es zu gestalten gilt. Meine Aufgabe dabei war das Entwerfen des Charakters „Tod“ nach gegebener Charakterisierung durch die im Script vorhandenen Informationen. Dazu zählt das Entwickeln des Charakters durch Skizzieren verschiedenster Versionen und das Sculpten des Charakters in Blender.



Abbildung 4 & 5: Tod Modell in Blender - Quelle: Eigene Abbildung

Der Charakter wird anschließend als Vorlage für die Anfertigung eines MetaHuman genutzt. Somit erhalten wir hochqualitative Modelle mit vorgefertigten Rigs für Körper und Gesicht. Die Kleidung kann dabei selbst angefertigt werden und ist nicht auf die dürftige Auswahl der von MetaHuman gestellte Kleidung angewiesen.



Abbildung 6 & 7: Tod als MetaHuman - Quelle: Eigene Abbildung

- **Environment**

Für die Story ist es notwendig, virtuelle Drehorte, in denen die Handlung spielt, umzusetzen. Da es sich um mehrere Umgebungen handelt, wurden diese im Team aufgeteilt. Bei meiner zugewiesenen Umgebung handelt es sich um eine dichte Waldumgebung, die der Haupthandlungsort des Filmes darstellt.

Der Wald soll gruselig wirken und kommt in der Geschichte in einer Nacht vor. Die Beleuchtung des Waldes ist daher hauptsächlich durch ein Mondlicht gewährleistet. Da ein atmosphärisches Licht allerdings nicht in allen Situationen ausreichend ist, wurde die nähere Umgebung, in der die Handlung spielt, separat durch manuell platzierte Lichter detailreicher ausgeleuchtet. So konnte sichergestellt werden, dass die eigentliche Handlung des Filmes ausreichend beleuchtet war und der im Hintergrund sichtbare Wald als solcher erkannt werden kann.



Abbildung 8: Waldumgebung - Quelle: Eigene Abbildung

Die Umsetzung des Waldes wurde dabei durch die in der Unreal Engine integrierten Tools durchgeführt und durch Assets und Scripts aus dem Epic Games Marktplatz und der Quixel Asset Library zusammengestellt. Eine genauere Betrachtung der Vorgehensweise bei der Erstellung der Umgebung folgt im Kapitel 4.

- **Dreharbeiten**

Aufgrund der Größe des Teams konnte während der Dreharbeiten keine feste Rollenverteilung gewährleistet werden. So war ich zuständig für die Bedienung der Unreal Engine, dazu zählt die Vorbereitung der Unreal Engine für die Benutzung von Motion Capture und der Arbeit mit VIVE Motion Trackern für die Verwendung von virtuellen Kameras. Ebenso gehört dazu die Aufnahme der virtuellen Kamera durch Benutzung des Sequenzers.

Eine weitere Rolle während der Dreharbeiten war die schauspielerische Verkörperung des Charakters „Sandman“. Dabei spielte ich sowohl die Bewegungen des Körpers als auch die später aufgezeichnete Mimik.

- Nachbearbeitung

Da die aufgenommenen Daten von Captury Studio sowie die aufgenommenen Daten der Unreal Engine eine Nachbearbeitung benötigen, ist eine weitere Aufgabe nach den Dreharbeiten die Nachbearbeitung der gespeicherten Daten. Dazu zählen das Entfernen von Noise und das Retargeting des Captury Skeletons auf das durch MetaHumans gestellte Skeleton.

2.3 Aufbau der Arbeit

Die Bachelorarbeit besteht aus dem hier zu lesenden theoretischen Part und dem praktischen Part, der aus der Arbeit am Animationsfilm „Mr. Sandman(AT)“ besteht.

Um als themenfremde Person einen genaueren Einblick in die Thematik zu bekommen, beginnt der theoretische Part mit einer genauen Definition des Begriffes „Virtual Production“. Dabei wird erklärt, was unter dem Begriff zu verstehen ist.

Der Hauptteil der theoretischen Arbeit besteht aus mehreren Unterpunkten, die die Techniken, die im Animationsfilm „Mr. Sandman (AT)“ verwendet wurden, erklären und die Implementation in die Unreal Engine, wie sie im praktischen Projekt geschieht, darstellen. Dabei werden bei Einstellungsmöglichkeiten verschiedener Actors, Texturen, Materials, etc. nur auf die wichtigsten oder im Projekt benutzen Einstellungen Bezug genommen. Es werden mögliche Fehler, Chancen und die Ziele der Techniken angesprochen und die Ergebnisse abschließend gewertet.

3. Definition „Virtual Production“

Durch den steigenden Gebrauch von „visual effects“ (VFX) in Filmen und Serien verändert sich die Arbeit am Filmset. Der Unterschied zwischen dem, was während der Dreharbeiten am Set vom Film zu sehen ist und wie der Film am Ende der Postproduction aussieht, wird immer größer (vgl. Rogers, 2020).

Filmcrews und Schauspieler müssen dies beachten und sich die Effekte, Charaktere oder Umgebungen/Hintergründe, die erst nach den Dreharbeiten fertig sind, während der Dreharbeiten vorstellen. Dies sorgt immer wieder für Schwierigkeiten am Set (vgl. muthmedia, o. D.).

VP ermöglicht eine Verschiebung der üblichen Produktionslinie.

Die Arbeit an den VFX startet deutlich früher und läuft laut Kevin Baillie, Method Studios' Creative Director, parallel neben den Dreharbeiten anstatt wie ursprünglich linear (vgl. Kadner, 2019, S. 4).

Der Virtual Production Field Guide von Epic Games beschreibt VP als einen Oberbegriff für ein Spektrum von Computer unterstützten Produktions- und Visualisierungsmethoden (vgl. Kadner, 2019, S. 3). Durch die Verwendung von virtuellen Kameras, Motion Capture Systemen und 3D Software können Filmemacher interaktiv in der Produktionsphase digitale Szenen visualisieren und erkunden (vgl. Autodesk, 2009, S. 1).

Die Produktionsumgebungen ändert sich bei der Verwendung von VP in ein reines Studiosetting. Statt Green-/Bluescreen wird in Live Action Filmen auf hochauflösende LED Wände zurückgegriffen, auf denen unter der Verwendung von Game Engines wie der Unreal Engine virtuelle Hintergründe abgebildet werden. So können Schauspieler direkt vor den Hintergründen aufgenommen werden (muthmedia, o. D.).

“Our latest developments enable final pixel quality in real time, which empowers content creators across all stages of production, and equip them to shoot their visual effects in camera. It's a game changer.”

– Marc Petit, general manager at Epic Games (Deloitte, 2020).

4. Virtual Production für „Mr. Sandman (AT)“ in der Unreal Engine

4.1 *Environments*

Environments sind virtuelle Umgebungen bestehend aus Terrain, Licht und Assets, die in Game Engines gebaut werden, um Umgebungen für Videospiele, Animationswerke oder durch Virtual Production auch für Serien und Filme zu schaffen.

Diese Umgebungen ersetzen den Dreh an einer real existierenden Location.

Da diese Umgebungen rein virtuell sind, bieten sie viele Vorteile gegenüber echten Umgebungen. So besteht die Möglichkeit, durch Verwendung des Sequenzers für Aufnahmen in jeder Phase der Produktion Änderungen an der Umgebung vorzunehmen. Regisseure können direkt während der Dreharbeiten Ausleuchtungen und Position von Objekten nach Belieben in Echtzeit verändern.

Es ermöglicht sogar durch Verwendung von VR Techniken eine direkte Erkundung der Umgebungen. Durch die Veränderung des Terrains, Modelle und der Texturierungen lassen sich so fotorealistische oder auch stilisierte Umgebungen ganz nach Wunsch kreieren.

Da durch das Drehbuch für das Projekt „Mr. Sandman“, bestimmte Drehorte in der Story vorgesehen sind, ist es nötig, diese Umgebungen anzufertigen. Gesondert werden wir die Anfertigung einer Waldumgebung betrachten, in der der erste Abschnitt des Animationsfilmes spielt.

4.1.1 *Organisation*

Umgebungen, besonders Wälder bestehen aus einer Vielzahl von unterschiedlichen Assets und Texturen, die erstellt und verwaltet gehören. Es ist daher sehr ratsam, eine passende Struktur zur Speicherung anzulegen. Dies sollten bereits bei der Erstellung der Assets in Programmen wie Maya oder Blender geschehen.

Die Unreal Engine nutzt den sogenannten Content Browser zum Verwalten und Strukturieren der Inhalte. In diesen befinden sich alle Basis Objekte, der Game Engine sowie auch alle importierten oder in der Engine erstellten Objekte und bildet somit den Hauptbereich für die Erstellung, Importierung, Organisierung und Management der Inhalte der Engine (vgl. Content Browser, o. D.).

Der Content Browser ermöglicht es, eine eigene Ordnerstruktur anzulegen, was ein exaktes Replizieren der Ordnerstruktur vor dem Import ermöglicht.

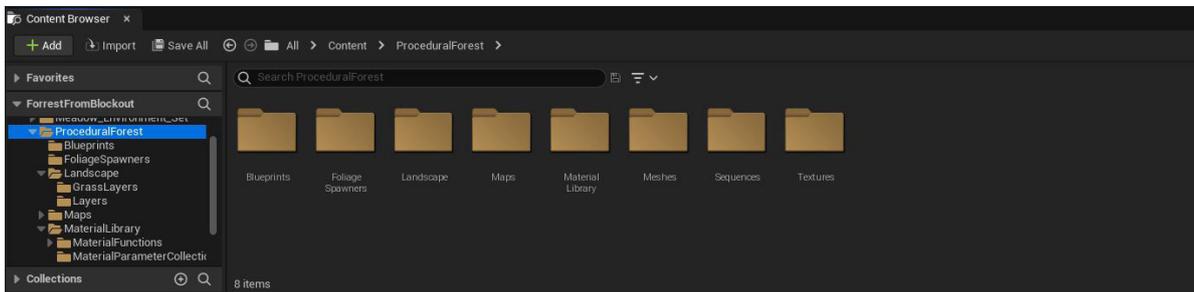


Abbildung 9: Content Browser - Quelle: Eigene Abbildung

Die Unreal Engine bietet dabei Filtermechaniken, um Assets einfacher im Content Browser zu lokalisieren. Dabei ermöglicht der Content Browser einfache Interaktionen mit den Assets. Durch einfaches Klicken auf zum Beispiel ein Asset des Typen Blueprint, öffnet sich passend der Blueprint Editor für das ausgewählte Asset. Die Engine passt sich dabei dem Typ des ausgewählten Assets an und öffnet den passenden Editor oder das passende Detailpanel (vgl. Content Browser, o. D.).

4.1.2 Landscape Actor Erstellung

Die Arbeit an der Umgebung beginnt bereits in der Previsualisierungsphase in der die Umgebungen anhand von Konzeptzeichnungen und Analyse des Drehbuches grob geblockoutet wird.

Dadurch wird es uns ermöglicht, bereits sehr früh in der Produktion einen Eindruck zu bekommen, wie die Umgebung aussehen könnte. Ebenso können Abstände zwischen zum Beispiel Bäumen und Abstände zwischen den genaueren Handlungsstandorten, indem sich die Charaktere aufhalten, genauer eingeschätzt und bestimmt werden.

Da der Wald im Gegensatz zu einem Kinderzimmer eine natürliche Umgebung ist, ist es nötig, in der Unreal Engine ein sogenanntes Landscape(Landschaft) anzulegen.

Das Landscape System in der Unreal Engine ist eine Kollektion von Werkzeugen, die es einem ermöglichen, große Außenumgebungen zu realisieren (vgl. Landscape Quick Start Guide, o. D.).

Zum Gestalten einer solchen Landschaft ist es anfangs notwendig, einen Landscape Actor zu erstellen. Der Landscape Actor ist eine Grundfläche, auf der später durch die Verwendung von Heightmaps Verformungen vorgenommen werden können. Um diesen Actor anzulegen, ist ein Öffnen des Landscape Mode über die Main-Toolbar notwendig. Der Managementtab des Landscape Mode bietet hier die Optionen zur Erstellung oder Bearbeitung von Landscapes (vgl. Landscape Quick Start Guide, o. D.).

Die „New“ Auswahl im Management Tab bietet eine Vielzahl an Einstellungen, die ausschlaggebend für Größe, Platzierung, Einteilung und Performance der Fläche sind. Neben den bekannten Optionen wie die Positionierung und Rotation der Fläche auf der X,Y und Z Achse bietet uns die Unreal Engine die Möglichkeit extern erstellte Heightmaps bereits in der Erstellungsphase des Actors zu importieren. Ebenso ist es bereits hier möglich, ein erstelltes Landscape Material auszuwählen. Das Auswählen von einem Landscape Material oder das Importieren von extern erstellten Heightmaps ist nach der Erstellung des Actors weiterhin möglich (vgl. Landscape Quick Start Guide, o. D.).

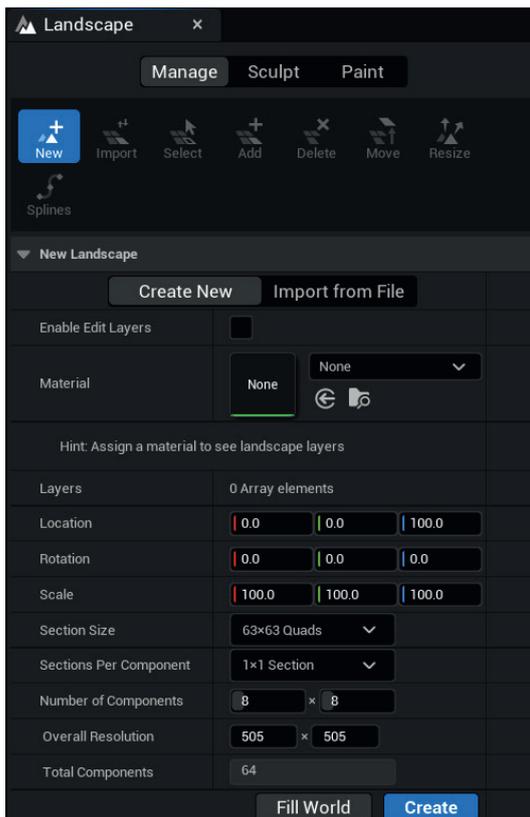
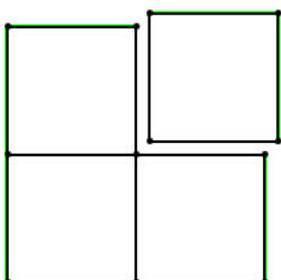


Abbildung 10: Management Tab des Landscape Mode - Quelle: Landscape Quick Start Guide, o. D.

Die richtige Balance zwischen Auflösung und Performance zu finden stellt sich häufig schwieriger heraus als gedacht. Heightmap Dimensionen zu wählen ist damit ein wichtiger Bestandteil der Erstellung des Landscapes und kann im späteren Verlauf der Gestaltung des Landscapes nicht mehr so einfach korrigiert werden. Jedes Landscape ist unterteilt in mehrere Landscape Components, die alle quadratisch sind und dieselbe Größe teilen. Landscape Components sind Unreal Engines Basiseinheit für Rendering, Berechnung der Sichtbarkeit von im Landscape platzierten Actors und Assets und der Berechnung von Kollisionen (vgl. Landscape Technical Guide, o. D.).



Zu sehen ist ein Landscape Actor (Grüne Umrahmung) bestehend aus vier Landscape Components.

Abbildung 11: Landscape Actor bestehend aus vier Landscape Components - Quelle: Landscape Technical Guide, o. D.

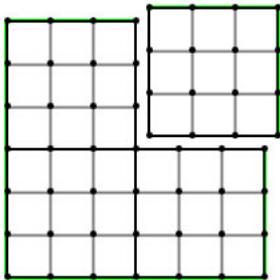
Nach der Wahl der Anzahl der Landscape Components ist es optional möglich, diese Components in Sektionen einzuteilen. Dabei besteht die Auswahl zwischen einer 1x1 Unterteilung und einer 2x2 Unterteilung. Bei einer 1x1 Unterteilung wird jeder Landscape Component als eine eigenständige Sektion angesehen. Bei einer 2x2 Unterteilung wird jeder Landscape Component in vier Untersektionen aufgeteilt, was die Auflösung des Landscapes erhöht. Sektionen werden von der Unreal Engine für Berechnungen von LODs benutzt (vgl. Landscape Technical Guide, o. D.).

Landscapes mit kleineren Sektionen können LODs besser darstellen, besitzen aber eine schlechtere Performance. Durch den Gebrauch von größeren Sektionen sinkt zwar die Anzahl Components, was eine besser Performance bietet, allerdings können dadurch LODs nicht mehr so gut dargestellt werden, was einen Einfluss auf das grafische Aussehen des Landscapes hat (vgl. Creating Landscapes, o. D.).

Die Größe jeder Sektion wird von der Anzahl der Vertices bestimmt.

Die Anzahl der Vertices ergibt sich durch eine Gridunterteilung der Sektionen.

Unreal Engine gibt eine maximale Gridunterteilung von 256x256 vor (vgl. Landscape Technical Guide, o. D.), empfiehlt allerdings für eine ausreichende Größe und Performance eine Einteilung von 63x63 (vgl. Creating Landscapes, o. D.). Diese aus der Grideinteilung entstehenden Quadrate werden **Quads** genannt (vgl. Landscape Technical Guide, o. D.).



Zu sehen ist ein Landscape Component das in vier Sektionen unterteilt wurde. Jede Sektion wurde besteht aus neun (3x3) Quads.

Abbildung 12: Landscape Component bestehend aus vier Untersektionen - Quelle: Landscape Technical Guide, o. D.

Die Größe des Landscapes setzt sich also aus der Anzahl von Quads in jeder Sektion, die Anzahl der Sektionen in jedem Landscape Component und der Anzahl der Landscape Components zusammen. Die Dimensionen des Landscapes lässt sich mit einer mathematischen Formel und den gewählten Werten errechnen (vgl. Landscape Technical Guide, o. D.).

Formel: **(A*Quads + 1, B*Quads + 1)**

A und B beschreiben die Anzahl an Landscape Components in die X und Y Richtung.

Quads beschreibt die Anzahl der Quads pro Landscape Component

(vgl. Landscape Technical Guide, o. D.).

Beispiel:

Ein Landscape Component wird durch die 2x2 Unterteilung in vier Untersektionen unterteilt. Jede dieser Untersektionen wird durch Bestimmung der Sektionsgröße in 63x63 Quads unterteilt. Dies bedeutet, dass der Landscape Component 126x126 Quads vorweist. Wenn der Landscape Actor aus 32x32 Landscape Components besteht, kommen wir auf 4032 Quads in X und Y Richtung. Um nun eine extern erstellte Heightmap zu importieren, müsste die Heightmap eine Dimension von 4033x4033 vertices besitzen (vgl. Landscape Technical Guide, o. D.).

Unreal Engine vermerkt, dass jeder Component einen Render-Thread benötigt, der CPU Rechenleistung verbraucht. Auch Sektionen brauchen CPU Rechenleistung. Es wird daher geraten, die Anzahl von Sektionen und Components so gering wie möglich zu halten, um eine gute Performance zu gewährleisten (vgl. Landscape Technical Guide, o. D.).

Um die Auswahl der Werte einfacher zu gestalten, bietet uns die Unreal Engine eine Übersicht über empfohlene Einstellungen an. Diese Einstellungen ermöglichen die maximale Landschaftsgröße mit minimaler Component Anzahl (vgl. Landscape Technical Guide, o. D.).

Overall size (vertices)	Quads / section	Sections / Component	Landscape Component size	Total Landscape Components
8129 x 8129	127	4 (2x2)	254x254	1024 (32x32)
4033 x 4033	63	4 (2x2)	126x126	1024 (32x32)
2017 x 2017	63	4 (2x2)	126x126	256 (16x16)
1009 x 1009	63	4 (2x2)	126x126	64 (8x8)
1009 x 1009	63	1	63x63	256 (16x16)
505 x 505	63	4 (2x2)	126x126	16 (4x4)
505 x 505	63	1	63x63	64 (8x8)
253 x 253	63	4 (2x2)	126x126	4 (2x2)
253 x 253	63	1	63x63	16 (4x4)
127 x 127	63	4 (2x2)	126x126	1
127 x 127	63	1	63x63	4 (2x2)

Abbildung 13: Empfohlene Landscape Größen - Quelle: Landscape Technical Guide, o. D.

4.1.3 Gestaltung des Landscapes

Nachdem wir nun einen Landscape Actor erstellt haben, haben wir jetzt die Möglichkeit, wenn wir nicht eine extern erstellte Heightmap benutzen, den Landscape Actor so zu gestalten, wie es vorgesehen ist. Dazu bietet uns das Landscape Tool der Unreal Engine eine Auswahl an Werkzeugen. Um diese Werkzeuge zu nutzen, ist es notwendig, zum Sculpt Tab des Landscape Modes zu wechseln (vgl. Sculpt Mode, o. D.). Aufgrund des Umfangs und der Anzahl der Werkzeuge werde ich mich bei der Erklärung auf die wichtigsten Werkzeuge und Einstellungsmöglichkeiten beschränken, die bei der Erstellung der Waldszene verwendet wurden.

Das erste Werkzeug, welches wir genauer untersuchen, ist das *Sculpt* Werkzeug.

Mit dem Sculpt Werkzeug ist es uns möglich, die Heightmap des Landscape Actors an bestimmten Stellen, in Form des ausgewählten Brushes, anzuheben oder abzusenken. Dabei wird in der betreffenden Region die Höhendaten der Heightmap des Landscapes erhöht oder verringert. Die Stärke der Operation und die Größe des Brushes lassen sich dabei in den Optionen einstellen (vgl. Landscape Sculpt Tool, o.D.). Bei den Arbeiten am „Mr. Sandman“ Projekt wurde das Sculpt Werkzeug hauptsächlich für die sich im Wald befindenden Hügel und Täler benutzt.

Da das Sculpt Werkzeug öfters unschöne Kanten hervorbringt, liefert uns das Landscape Tool ein Werkzeug, um harte Kanten abzuflachen und ein flüssigeres Aussehen zu gewährleisten. Das *Smooth* Werkzeug wird dabei wie auch das Sculpt Werkzeug mit einem Brush benutzt. Auch dieses Werkzeug besitzt Einstellungen betreffend Stärke der angewandten Operation (vgl. Landscape Smooth Tool, o. D.).

Das dritte Werkzeug, das in der Kreierung des Waldes öfters benutzt wurde, ist das *Flatten* Werkzeug. Da Motion Capture bestimmte Limitierungen in der Bewegungsfreiheit der Schauspieler hat, ist es notwendig für Stellen, an denen sich die Charaktere bewegen werden, das Flatten Werkzeug zu benutzen. Durch Benutzung des Flatten Werkzeug werden Teile des Landscapes auf dieselbe Höhe angehoben oder abgesenkt wie initial ausgewählt. Der Initialwert wird durch Mausklick bestimmt. Durch das Halten des Mausklicks kann das Werkzeug durch einen Brush alle Flächen, über die der Brush fährt, auf die ausgewählte Initialhöhe angleichen.

Genauso wie beim Sculpt und Smooth Werkzeug gibt es die üblichen Optionen zur Änderung der Brushgröße oder der Stärke der Operation, aber auch spezielle Einstellungsmöglichkeiten wie den *Flatten Mode* (vgl. Landscape Flatten Tool, o. D.).

Der Flatten Mode gibt uns eine Auswahl zwischen drei verschiedenen Modi. Beim *Both* Modus werden Werte über und unter dem gewählten Initialwert an den Initialwert angeglichen. Der *Raise* Modus ignoriert Werte über dem Initialwert. Werte unter dem Initialwert werden angeglichen. Der letzte Modus ist der *Lower* Modus der nur Werte über dem Initialwert angleicht und Werte unter dem Initialwert nicht beachtet (vgl. Landscape Flatten Tool, o. D.).

Durch die Verwendung von externen Programmen wie *World Machine* oder *GAEA* lassen sich ebenfalls hochqualitative Landscapes erstellen. Diese Programme sind speziell für die Erstellung von Terrain gedacht und weisen damit Features auf, die der Landscape Mode der Unreal Engine nicht vorweist. Dabei werden die Landscapes nicht gesculpted, sondern simuliert. *World Machine* und *GAEA* besitzen ebenfalls Tools zur Exportierung zu bekannten Engines wie Unreal Engine und Unity. Nachdem wir nun den Landscape Actor erstellt und ihn nach unseren Wünschen geformt haben, können wir ein sogenanntes Landscape Material anlegen.

4.1.4 Landscape Material

Ein Landscape Material ist ein spezielles Material, welches das Material System der Unreal Engine benutzt. Dieses Material bietet Nodes, die speziell auf Landscapes ausgerichtet sind. Wie auch normale Materials findet die Bearbeitung des Landscape Materials im Material Editor der Unreal Engine statt (vgl. Landscape Material Layer Blending, o.D.).

Um ein Landscape Material zu erstellen, navigieren wir zuerst durch Benutzung des Content Browsers in das von uns gewählte Verzeichnis. Dazu bietet sich an, für das Landscape Material eigenständige Unterverzeichnisse im Ordner des Landscape Actors zu erstellen. Hierbei erstellen wir einen Ordner mit dem Namen Textures und einen Ordner mit dem Namen Materials.

Anschließend sollte man sich einen Überblick verschaffen, welche Texturen zur Gestaltung der Landschaft benötigt werden. Im Falle der zu erstellenden Waldumgebung wären dies Texturen wie Laubböden, Moosböden oder steinige Untergründe. Diese Texturen dienen nur dem Untergrund, auf denen später 3D Objekte wie Laubblätter platziert werden. Die Texturen sollten möglichst zum Stil des Filmes passen.

Wenn eine fotorealistische Umgebung gewünscht wird, bietet sich an, für diese Texturen Fotos von echten Untergründen zu verwenden. Eine andere Möglichkeit wäre die Erstellung der Texturen durch ein Programm wie Photoshop oder 3D Software wie Mudbox oder ZBrush.

Ebenso empfiehlt es sich Displacement, Normal und Bump Maps für diese Texturen anzulegen, um Details im Untergrund ohne die Verwendung von 3D Modeling hervorzuheben. Im Verzeichnis Materials ist es uns möglich, durch Rechtsklick eine Vielzahl von Auswahlmöglichkeiten anzulegen. Um ein Landscape Material anzulegen, wählen wir durch Rechtsklick in der Kategorie *Materials* das Asset *Material* aus.

Nachdem das Material erstellt wurde, kann ein frei wählbarer Name zugeordnet werden. Durch Doppelklick auf das neu erstellte Material im Content Browser öffnet sich der Material Editor, in dem wir nun das Material bearbeiten und nach Wunsch einstellen können.

Im Material Editor fangen wir zuerst damit an, ein *LandscapeLayerCoords* Node zu erstellen. Dieser Node ist die Basis, mit der wir anfangen, unser Node Netzwerk aufzubauen und hilft uns dabei, UV Koordinaten zu generieren, die dafür verwendet werden, das Material auf den Landscape Actor anzuwenden (vgl. Landscape Quick Starter Guide, o. D.).

Damit wir später mehrere Texturen verwenden können und diese auch miteinander mischbar sind, ist es als Nächstes nötig, ein *LandscapeLayerBlend* Node anzulegen. Nachdem der Node erstellt und durch Klicken ausgewählt wurde, erscheint ein Einstellungsmenü, indem nun ein paar Einstellungen vollzogen werden sollten. Um mehrere Texturen zu ermöglichen, muss vorher im *LandscapeLayerBlend* Node für jeden Untergrundtyp ein weiterer Layer angelegt werden.

Diese Layer können anschließend nach dem Untergrundtyp benannt werden. Im folgenden Bild sehen wir den aktuellen Status des Landscape Materials mit den zusätzlichen Layern im LandscapeLayerBlend Node (vgl. Landscape Material Layer Blending, o. D.).

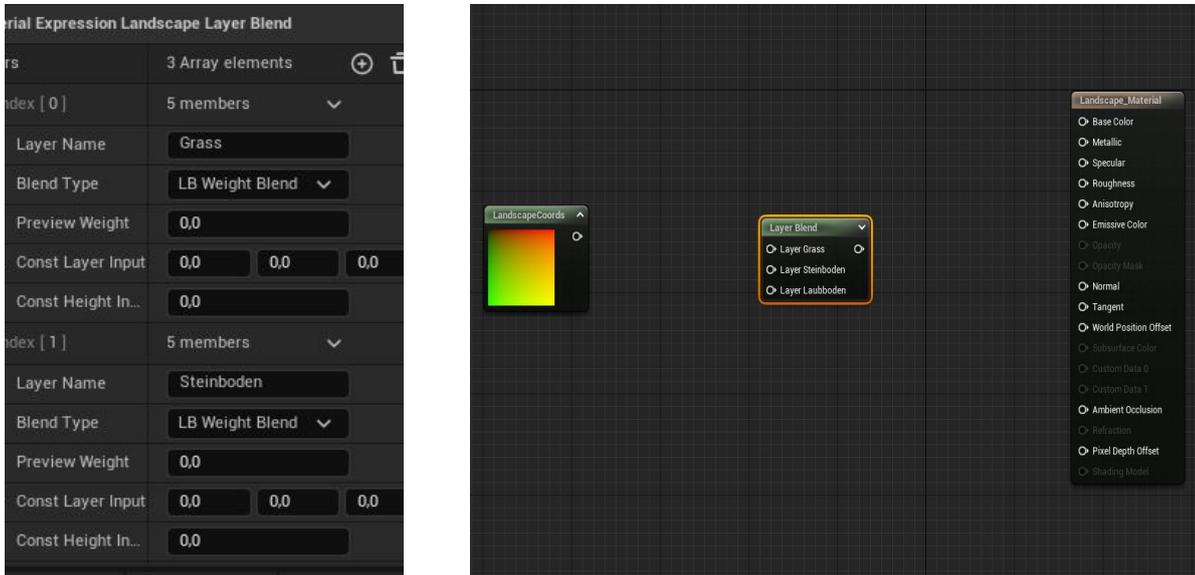


Abbildung 14 & 15: Aktueller Stand des Landscape Materials - Quelle: Eigene Abbildung

In den Einstellungen des LandscapeLayerBlend Nodes haben wir nach Anlegen der Layer noch die Möglichkeit, für jedes der Layer eigene Einstellungen zu treffen. Eine wichtige Auswahl, die wir treffen können, ist der Blend Typ. Der Blend Typ bestimmt die Berechnungsart, die Unreal Engine benutzt, um festzulegen, welcher Untergrund aktuell sichtbar gemacht wird. Auswählen können wir zwischen drei Typen, die jeweils für bestimmte Situationen besser geeignet sind als die anderen (vgl. Landscape Quick Starter Guide, o. D.).

Während Height Blending besser für Situationen gedacht ist, in denen eine Textur klar abgegrenzt über einer anderen liegen soll, funktioniert Weight Blending besser für fließende Übergänge zwischen Texturen. Der dritte Typ, der uns zur Verfügung steht, ist das Alpha Blending. Beim Alpha Blending benutzt die Unreal Engine Weight Informationen zusammen mit einem Alpha Layer, um die Ergebnisse zu ermitteln (vgl. Landscape Quick Starter Guide, o. D.).

Nachdem nun die Typen eingestellt worden sind, sollten als Nächstes die Texturen der verschiedenen Untergründe eingefügt werden. Dafür müssen wir für jede Texture, Bump-, Displacement- und Normal Map eine *Texture Sample* Node erstellen. In den Einstellungen des Texture Samples müssen anschließend noch die importierten Texturen ausgewählt werden (vgl. Landscape Material Layer Blending, o. D.).

Der nächste Schritt, der zu vollziehen ist, ist es, die von uns in den letzten Schritten erstellten Nodes zu verbinden. Als Erstes verbinden wir den Output vom LandscapeCoords Node mit dem UV Input der Texture Sample Nodes. Nun wird der Layer Blend Node mit den Texture Sample Nodes verbunden. Für mehr Kontrolle über Parameter bietet es sich an, einen weiteren Layer Blend Node zu erstellen, der die extra Maps wie Displacement, Bump und Normal händelt. Dabei wird jeweils der RGB Output der Texture Samples mit dem dazugehörigen Input des Blend Nodes verbunden (vgl. Landscape Quick Starter Guide, o. D.).

Abschließend werden die Outputs der Layer Blend Nodes mit dem jeweiligen Endslot verbunden. Dabei wird das Ergebnis des Layer Blend Node der Texturen mit dem Base Color Input verbunden. Das Ergebnis des Layer Blend Node der Maps wird mit dem passenden Endslot verbunden. So wird der Blend Node für Normal Maps auch mit dem Normal Slot verbunden (vgl. Landscape Quick Starter Guide, o. D.).

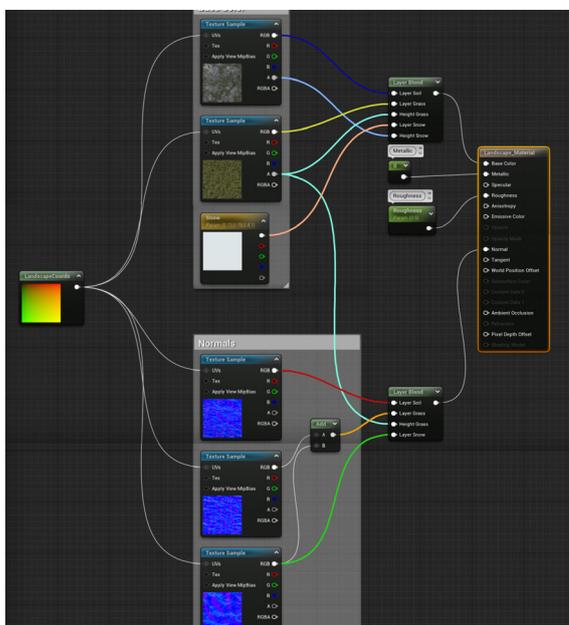


Abbildung 16: Beispiel fertiges Landscape Material - Quelle: Landscape Quick Start Guide, o. D.

Nachdem wir jetzt ein einfaches Landscape Material kreiert haben, müssen wir es dem zuvor von uns erstellten Landscape Actor zuweisen. Dazu navigieren wir im Content Browser zum Speicherplatz des erstellten Landscape Actors. Im Detailpanel des Actors wird das Material in der Auswahl *Landscape Material* eingetragen (vgl. Landscape Material Layer Blending, o. D.).

Das Material kann nun durch Painting auf dem Landscape Actor aufgetragen werden. Im Landscape Tool ist es dazu nötig, in den Landscape Paint Tab zu wechseln. Bevor wir aber anfangen, den Landscape Actor nach Wunsch mit dem Landscape Material zu designen, müssen wir noch *Landscape Layer Info Objects* anlegen, in denen die vom Painten entstandene Informationen gespeichert werden. In den Einstellungen des Paint Tabs befinden sich im Abschnitt *Target Layers* unsere im Landscape Material angegebenen Layers. Rechts neben den von uns angelegten Layer befindet sich ein Plus Icon, welches uns durch Auswahl ein Menü öffnet, indem wir ein Object anlegen können (vgl. Landscape Quick Starter Guide, o. D.).

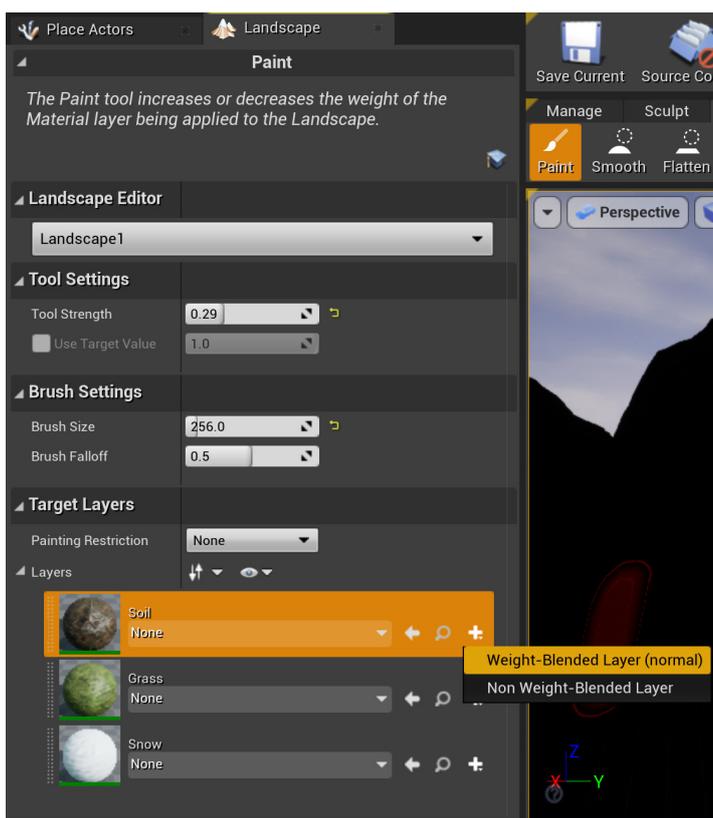


Abbildung 17: Blend Layer - Quelle: Landscape Quick Start Guide, o. D.

Nun kann mit dem Painten des erstellten Materials auf dem Landscape Actor angefangen werden. Dazu öffnen wir im Landscape Tool den Paint Tab.

Im Einstellungsbereich des Paint Tabs ist es möglich, wenn wir mehrere Landscape Actors haben, das zu bearbeitende Landscape auszuwählen (vgl. Landscape Quick Starter Guide, o. D.). Für das Painten des Landscapes wird genauso wie beim Sculpten ein Brush benutzt, der in der Größe und Intensität eingestellt werden kann. Nun können die erstellten Materials durch Benutzung des Brushes auf den vorher angelegten und gesculpteten Landscape Actor aufgetragen werden.

4.1.5 Importierung von Assets

Der nächste Schritt nach dem Auftragen des Landscape Materials ist das Importieren von Assets, die später im Landscape platziert werden sollen. Diese Assets können durch 3D Software wie Autodesk Maya oder Blender selbst entworfen und texturiert werden oder sie können von der von Epic Games zur Verfügung gestellten Quixel Library stammen.

Die Quixel Megascans Library ist eine Sammlung hochqualitativer Texturen und Models, die durch Fotogrammetrie erstellt worden sind (vgl. Quixel Megascans, o. D.). Diese Assets können durch die Quixel Bridge direkt in die Unreal Engine importiert werden. Dabei hat der Anwender die Auswahl zwischen mehreren Auflösungen, in denen der Import möglich ist. Für fotorealistische Texturen bietet die Quixel Bridge Auflösungen bis zu 8k an (vgl. Quixel Bridge, o. D.). Sollte man eigene Modelle importieren wollen, so passiert dies unter Verwendung des Content Browsers. Im Content Browser ist es möglich, sich durch das mit Rechtsklick öffnende Menü unter der Option *Import to / Game*, Assets zu importieren. Die zweite Möglichkeit, die uns der Content Browser zur Importierung bietet, ist der Import Button in der Navigationsleiste. In den sich öffnenden Importierungseinstellungen kann man nun für den Typ der importierten Datei passende Einstellungen treffen.

Nach der Importierung des Assets steht es uns im Content Browser für weitere Bearbeitungen zur Verfügung. Die Assets können nun per Drag-and-Drop aus dem Content Browser in der Landschaft platziert werden. Um große Bereiche wie zum Beispiel Grasflächen zu realisieren, benötigt man Tausende dieser Assets, die durch ihre hohe Auflösung und Anzahl enorm viel Performance benötigen. Um die beste Performance für Videospiele oder Realtime Produktion von Animationsfilmen zu gewährleisten, wird dringend die Benutzung von LODs für die Assets empfohlen.

4.1.6 Nanite

Eine weitere Möglichkeit, die uns die Epic Games seit dem Release der Unreal Engine 5 bietet, ist das *Nanite* System. Nanite ist ein neues, von der Unreal Engine entwickeltes System, welches ein neues internes Mesh Format und Rendering System benutzt, um hochauflösende Modelle Performance freundlich und in hoher Anzahl zu realisieren. Das System erstellt ebenfalls eigene LODs der Modelle, was ein eigenes Anfertigen von LODs ersetzt. Nanite analysiert und wandelt Meshes von Assets in hierarchische Cluster von Dreiecksgruppen um. Diese Cluster werden passend zur Kameraperspektive in unterschiedliche Detailstufen dargestellt. Der Wechsel der LOD Stufen findet dabei fließend und ohne visuelle Kanten zwischen den Clustern statt. Nanite lädt dabei die benötigte Daten ebenfalls fließend, was dafür sorgt, dass nur wirklich benötigte Daten von Details im Speicher geladen werden, welches einen positiven Einfluss auf die Performance hat (vgl. Nanite Virtualized Geometry, o. D.).

Nanite hat allerdings auch technische Einschränkungen, die es zu beachten gilt. Eine dieser Einschränkungen bezieht sich auf sogenannte *Aggregate Geometry*. Es handelt sich dabei um sehr feine Geometrie wie Grass, Haare und Blätter, dessen Fläche zu klein ist, um von Nanite berechnet zu werden. Um Nanite zu aktivieren, haben wir während der Importierung bereits die Möglichkeit, ein Nanite Container zu erstellen. Aber auch wenn wir die Assets bereits importiert haben, ist es uns möglich, Nanite für diese Assets zu aktivieren. Dafür können wir einfach durch Rechtsklick auf das Asset im Content Browser über die Option *Static Mesh Actions / Nanite* aktivieren. Es ist ebenso möglich, Nanite direkt in der Detailansicht des Assets zu aktivieren, dafür ist es notwendig, das gewünschte Asset über den Content Browser doppelt anzuklicken. Der sich öffnende Editor beinhaltet Detailinstellungen, in denen man auch die Möglichkeit findet, Nanite zu aktivieren (vgl. Nanite Virtualized Geometry, o. D.).

4.1.7 Foliage Mode

Da Hunderte oder Tausende Assets per Drag-and-Drop in der Landschaft zu platzieren etwas umständlich wäre, ermöglicht die Unreal Engine es uns, genauso wie beim Sculpten oder Texturieren des Landscape Actors, durch Benutzung des Foliage Mode, Assets in der Landschaft zu platzieren. Der Foliage Mode ist wie der Landscape Mode über die Main-Toolbar erreichbar. Durch Aktivieren dieses Modes erhält man Zugriff auf eine Auswahl an Tools, die es einem ermöglichen, einzelne oder eine Auswahl an Assets in die Landschaft zu platzieren oder sie wieder zu entfernen (vgl. Foliage Mode, o. D.).

Wie auch im Sculpt Tab des Landscape Mode werden wir nur die wichtigsten Tools betrachten, die im Projekt „Mr. Sandman“ benutzt wurden. Durch das *Paint Tool* ist es möglich, anhand eines Brushes Assets auf den Landscape Actor zu platzieren. Dafür müssen die Assets, die platziert werden sollen, vorher in den Einstellungen des Tools hinzugefügt werden. In den Einstellungen ist dafür eine geeignete Schaltfläche mit der Bezeichnung *+ Foliage* angelegt. Nun tauchen die Assets in den Einstellungen auf und sind durch Klicken auswählbar. Nachdem wir soeben unsere Static Meshes ausgewählt haben, können wir für jedes Mesh weitere individuelle Einstellungen treffen (vgl. Foliage Mode, o. D.).

Diese Einstellungen beinhalten unter anderem Abstand zwischen platzierten Assets und Variationen in der Größe des platzierten Assets. Der Brush weist wie auch beim Sculpt und Texture Tool Einstellungsmöglichkeiten bezüglich Größe und Intensität auf.

Durch Führen des Brushes über den Landscape Actor lassen sich jetzt die ausgewählten Meshes platzieren. Das Gegenstück zum Paint Tool ist das *Erase Tool*, mit dem es möglich ist, die zuvor aufgetragenen Meshes wieder aus dem Landscape Actor zu entfernen. Dabei ist es möglich, die zu löschenden Meshes durch Markieren in den Einstellungen des Tools genau zu bestimmen (vgl. Foliage Mode, o. D.).

Weitere nützliche Tools, die bei der Detailarbeit helfen werden, ist das *Single Tool* und das *Move Tool*. Mit dem Move Tool lassen sich einzelne Meshes, die auf dem Landscape Actor platziert worden sind, auswählen und mit dem Gizmo transformieren. Das Single Tool ermöglicht es, ein in den Einstellungen markiertes Mesh einzeln auf den Landscape Actor zu platzieren. Alle Meshes, die unter der Verwendung des Foliage Tool platziert worden sind, werden automatisch in Gruppen zusammengefügt, die durch Hardware Instancing gerendert werden. Durch das Rendern der Gruppen benötigt Unreal Engine weniger Leistung als das Rendern jedes einzelnen Meshes (vgl. Foliage Mode, o. D.).

Nachdem nun die gewünschte Szenerie durch Platzierung der Assets erschaffen wurde, sollte sie, damit sie für Animationsfilme oder Videospiele verwendet werden kann, ausgeleuchtet werden. In der Unreal Engine haben wir eine Reihe an Lichttypen und vorgefertigten Systemen, die die gesamte Umgebung oder nur einzelne Stellen grob oder auch detailreicher ausleuchten können. Im Folgenden werden die im „Mr. Sandman“ Projekt verwendeten Lichttypen und Systeme erläutert. Die zu erstellende Waldumgebung ist eine Umgebung, die sich unterm freien Himmel befindet. Dadurch ist basierend auf der Tageszeit Sonne oder Mond sichtbar und dient als Haupt Lichtquelle.

4.1.8 Atmosphäre Basis Setup

Um eine solche Lichtstimmung zu simulieren, wurde auf dem in der Unreal Engine integrierten *Environment Light Mixer* zurückgegriffen. Der Light Mixer ist ein Fenster, indem alle im Landscape platzierten atmosphärischen Licht Componets aufgelistet und editierbar sind. Der Mixer ermöglicht es ebenso, neue Lichter oder Atmosphären anzulegen. Über die Menu Bar des Editors durch Auswahl des Windows -> Env. Light Mixer lässt sich das Fenster des Light Mixers öffnen (vgl. Environment Light Mixer, o. D.).

In der Oberfläche des Mixers ist es nun möglich, in der Toolbar Lichter und Atmosphären zu erstellen. Für eine Outdoorumgebung empfiehlt sich das Erstellen eines *Sky Lights*, *Atmospheric Light 0*, *Sky Atmosphere*, *Volumetric Cloud* und *Height Fog* (vgl. Environment Light Mixer, o. D.). Durch Platzierung dieser Componets wird bereits ein funktionsfähiges Atmosphäre System erstellt.

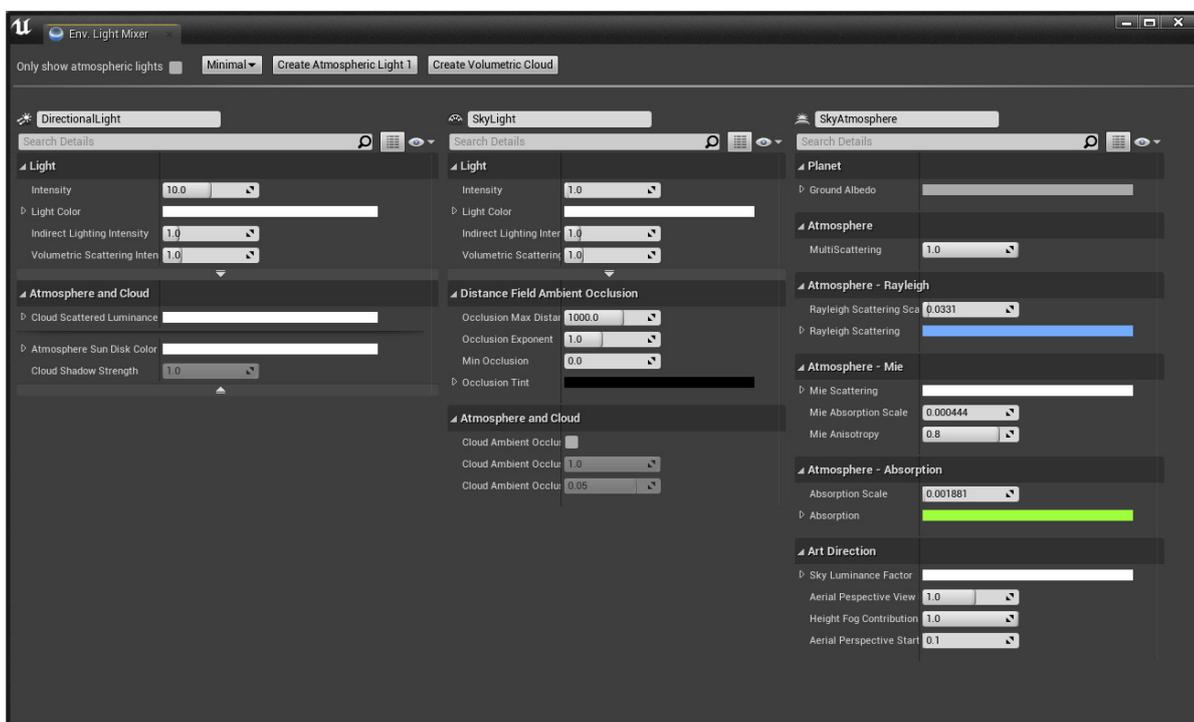


Abbildung 18: Light Mixer - Quelle: Environment Light Mixer, o. D.

Durch Erstellung des Atmospheric Light 0 platziert die Unreal Engine automatisch ein Directional Light Component in der Landschaft. Dieser Component wird für die Lichtstimmung am Tag zuständig sein. Auf Wunsch kann auch ein *Atmospheric Light 1* erstellt werden, der dadurch entstehende Directional Light Component wird für das Mondlicht zuständig sein. Optional kann es auch als zweite Sonnenquelle genutzt werden. Nur in Verbindung mit einem Sky Atmosphere Component wird eine Sonnenscheibe und Atmosphäre sichtbar (vgl. Environment Light Mixer, o. D.). Das Directional Light kreiert eine Lichtquelle, die unendlich weit entfernt ist und dessen Schattenwurf parallel zur Quelle verläuft. Nachdem das Licht im Editor platziert wurde, kann es mit der Rotation des Gyzmos oder durch STRG+L und anschließender Bewegung der Maus verschoben werden. Dadurch ändert sich der Winkel des einfallenden Lichtes und passend zum Stand der Sonne die Farbe des Lichtes (vgl. Directional Lights, o. D.).

4.1.9 Lumen

Sollte man ein Projekt in der Unreal Engine 5 erstellen, stellt die Unreal Engine das *Lumen* System zur Verfügung. Lumen ist Unreal Engines dynamisches Global Illumination und Reflections System, was es ermöglicht, indirektes Licht in Echtzeit zu generieren. Ebenso ermöglicht Lumen das durch eines Emissive Material ausgestrahlte Licht in das Global Illumination einfließen zu lassen. Durch die Verwendung von Lumen wird es ebenso möglich, Global Illumination in Reflexionen zu sehen. Lumen ermöglicht es dabei, ohne tieferegehende Einstellungen hochqualitative Ergebnisse im Bereich Global Illumination und Reflexionen zu erzielen und sollte deswegen, wenn mit der Unreal Engine 5 gearbeitet wird, benutzt werden (vgl. Lumen Global Illumination and Reflections, o. D.).

Sollte das Projekt in der Unreal Engine 5 begonnen worden sein, ist Lumen automatisch von Beginn an aktiviert, sollte jedoch eine Portierung von einem Projekt der Unreal Engine 4 vorliegen, sind Veränderungen der Projekteinstellungen nötig, um Lumen zu benutzen. Dafür muss über die Menü Bar unter dem Abschnitt Edit auf Projekt Settings navigiert werden. In den Projektsettings können wir nun Lumen aktivieren.

Die dafür benötigten Einstellungen finden sich im Abschnitt Rendering in der Engine Kategorie. Folgende Einstellungen wurden im Projekt „Mr. Sandman“ getätigt, um Lumen zu benutzen (vgl. Lumen Global Illumination and Reflections, o. D.).

- Dynamic Global Illumination Method -> Lumen
- Reflection Method -> Lumen
- Ray Lightning Mode -> Surface Cache
- Software Raytracing Mode -> Detail Tracing
- Shadow Map Method -> Virtual Shadow Map
- Support Hardware Ray Tracing -> true
- Use Hardware Ray Tracing when available -> true

Hardware Ray Tracing ist dabei nur verfügbar, sollte das System, auf dem die Unreal Engine läuft, eine Ray Tracing kompatible Grafikkarte besitzen.

4.1.10 Licht Typen

Aber nicht nur durch das Directional Light, Sky Light oder Emessive Materials lässt sich Licht in die Umgebung einfügen. Unreal Engine bietet uns zur weiteren Detailausleuchtung mehrere Light Types, die wir nach beliebiger Anzahl in der Landschaft platzieren können. Zu diesen Typen gehören unter anderem das *Point Light*, *Spot Light* und das *Rect Ligh*, welche im Create Abschnitt der Main Toolbar zu finden ist.

- **Point Light**

Das Point Light sendet Licht von einer Sphäre in alle Richtungen aus. Das Licht ist dabei vergleichbar mit einer Glühbirne (vgl. Light Types and Their Mobility, o. D.).

- **Spot Light**

Das Spot Light sendet Licht von einer Sphäre in eine Richtung in Form eines Kegels aus. Dabei benutzt das Licht einen inneren Kegel und einen äußeren Kegel. Der innere Kegel bietet volle Helligkeit, die sich nach außen zum äußeren Kegel abschwächt. Das Licht ist dabei vergleichbar mit einer Taschenlampe.

- **Rect Light**

Das Rect Light streut Licht von einer viereckigen Fläche aus. Dabei ermöglichen uns im Licht integrierte *Barn Doors* den Winkel des an den Rändern ausgehenden Licht zu beeinflussen.

Da nun Lichter in der Umgebung platziert worden sind, können diese detailreicher eingestellt werden. In der Detailansicht der Lichter finden wir viele Optionen, die es uns ermöglichen, das Licht nach Wunsch einzustellen.

In der Unreal Engine ist standardmäßig ein Auto Exposure aktiviert. Dieser kann vor allem bei der Ausleuchtung von Umgebungen für fehlerhafte Einschätzungen von Helligkeit sorgen. Bei bestimmten Betrachtungswinkeln werden zum Beispiel Wände heller dargestellt als in anderen Winkeln (vgl. Post Process Effects, o. D.).

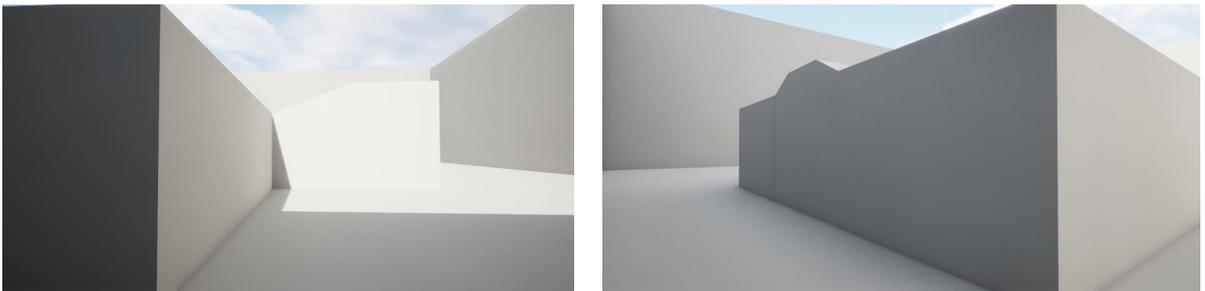


Abbildung 19 & 20: Post Prozess Volumne Auto Exposure - Quelle: Eigene Abbildung

4.1.11 Post Process Volumne

Um diese Fehlerquelle auszuschließen, ist es nötig, ein *Post Process Volumne (PPV)* anzulegen. Ein PPV ist ein frei platzierbarer Container, der im durch den Container eingegrenzten Bereich bestimmbare Einstellungen vornimmt. Die Einstellungsmöglichkeiten betreffen dabei verschiedenste Bereiche, die von Kameraeinstellungen bis hin zu Ray Tracing und Rendereinstellungen reichen, was das PPV zu einem der wichtigsten Bestandteile des Landscape macht. Der Umfang der Einstellungsmöglichkeiten ist immens und wird deswegen im Folgenden nur im Umfang der im Projekt angewendeten Einstellungen besprochen.

Die Einstellungen des PPV treten dabei nach den angewendeten Einstellungen der Actors und Components ein. Die Größe des Containers kann dabei selbst bestimmt werden und betrifft nur den gewählten Bereich oder sie kann die ganze Szene betreffen. Sollte man sich dazu entscheiden, dass das PPV die ganze Szene betreffen soll, so kann dies über die Aktivierung der Option *Infinite Extend (Unbound)* in den Einstellungen des PPV erreicht werden. Dadurch wird die Größe und Platzierung des PPV ignoriert und dessen Einstellungen wirken sich auf die ganze Szene aus (vgl. Post Process Effects, o. D.). Um die erwähnte Fehlerquelle des Auto Exposure zu deaktivieren, muss im PPV die Option

Metering Mode in der Kategorie *Lens* -> *Exposure* auf *Manuel* geschaltet werden. Dadurch werden die im PPV angegebenen Kamerasettings benutzt, um Brennweite und Blende zu bestimmen (vgl. Auto Exposure, o. D.). Ebenso erlaubt es, einen festen Blendenwert anzugeben, der nicht nur Lichtbedingungen der Umgebung beeinflusst. Durch anschließende Deaktivierung der Option *Apply Physical Camera Exposure* erhält man eine lineare Helligkeit des Exposures der sich dann nicht mehr an Lichtstimmungen anpasst.

Im PPV kann jetzt, ohne die Intensity Einstellungen des Directional Light oder anderer Lichter zu ändern, durch Veränderung des *Exposure Compensation* Wertes, die wahrgenommene Helligkeit der Umgebung beeinflusst werden. Der Wert dient dabei als Variable, der in die Berechnung des Exposures benutzt wird.

4.1.12 Basiseinstellungen der Lichter

Für die Einstellung der Lichter gibt es keine festen Werte, die immer passen. Die Lichter müssen an die erstellte Umgebung angepasst werden. Im Folgenden werden die grundlegenden Einstellungsmöglichkeiten präsentiert.

In der Toolbar des Mixer Fensters, ist die Tiefe der Einstellungsmöglichkeiten der kreierte Components einstellbar. Die Auswahlmöglichkeiten liegen dabei bei *Minimal*, welche nur die nötigsten Einstellungsmöglichkeiten präsentiert, *Normal*, welche allgemeine Einstellungen anzeigt und *Normal+Advanced*, welche alle möglichen Einstellungsmöglichkeiten der Components ermöglicht (vgl. Environment Light Mixer, o. D.). Für den Anfang empfiehlt sich den Minimal Einstellungsmodus zu verwenden.

- **Directional Light**

Die nun verfügbaren Einstellungen des Directional Light im Mixer Fenster bieten uns die Möglichkeit, die Stärke des ausgesendeten Lichtes anhand der Intensity Einstellung zu bestimmen. Im Directional Light kann dem ausgestrahlten Licht ebenso eine Färbung gegeben werden. Dabei stehen frei wählbare RGB Werte, HSV oder Hex Werte zur Verfügung. Optional kann auch im normalen Einstellungsmodus des Mixers, Kelvin als Einstellung für die Wärme des Lichtes benutzt werden. Durch die Einstellung *Indirect Light Intensity* ist es möglich, die Stärke des indirekten Lichtes, das durch Reflexionen an Objekten ausgestrahlt und in Lumens Global Illumination aufgenommen wird, zu bestimmen.

- **Sky Light**

Das Sky Light besitzt dieselben Einstellungsmöglichkeiten bezüglich Intensität und Farbe des Lichtes wie das Directional Light. Zusätzlich bietet das Sky Light noch weitere wichtige Einstellungen, die einen enormen Effekt auf das Endresultat der Lichtgebung haben. Sollte im Laufe der Aufnahmen eine dynamische Veränderung des Lichtes stattfinden, sollte die *Real Time Capture* Einstellung aktiviert werden. Bedenken sollte man allerdings, dass in diesem Modus Volumetric Fog vom Exponential Height Fog nicht mehr unterstützt wird.

Für die platzierten Point, Spot und Rect Lights, die nicht im Mixer zur Verfügung stehen, gibt es überschneidende Einstellungsmöglichkeiten. So ist es ebenfalls möglich, die Intensität des ausgestrahlten Lichtes sowie den Farbton oder Wärme des Lichtes einzustellen.

Durch die Einstellung *Attenuation Radius*, die nur im Point und Spot Light verfügbar ist, kann das ausgestrahlte Licht in einem sphärischen Bereich begrenzt werden.

Die Größe der Quelle, aus der das Licht generiert wird, kann dabei durch den *Source Radius* eingestellt werden.

4.1.13 Light Shafts

Light Shafts oder auch God Rays genannt, können vom Directional Light simuliert werden, um atmosphärische Lichteinstreuungen zu erzeugen. Diese Einstreuungen sorgen für Realismus und Tiefe in der Szene (vgl. Light Shafts, o. D.). Um Light Shafts anzulegen, muss zuerst Volumetric Fog im Exponential Height Fog aktiviert werden. Anschließend kann im Directional Light durch Veränderung des *Volumetric Scattering Intensity* Wertes die Stärke und Sichtbarkeit der Light Shafts, die durch das vom Directional Light ausgesendete Licht in Verbindung mit dem Volumetric Fog, entstehen, angepasst werden.

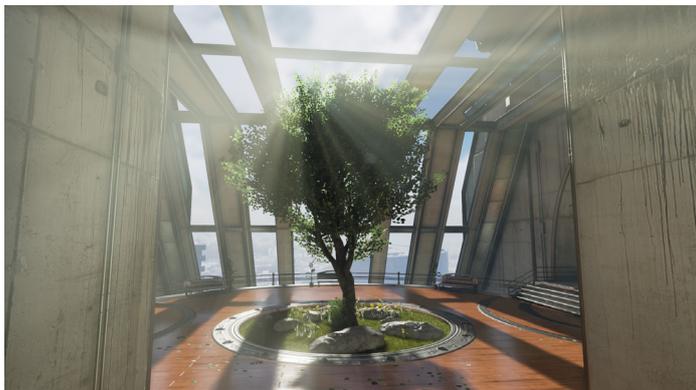


Abbildung 21: Light Shafts - Quelle: Light Shafts, o. D.

4.2 *Motion Capture*

Im Animationsfilm „Mr. Sandman“ wurde statt einer herkömmlichen, rein händischen Keyframeanimation der Charaktere auf Motion Capture (MoCap) zurückgegriffen. Beim MoCap spielen reale Schauspieler ihre 3D-Charaktere.

Dabei werden Position, Bewegung und Rotation von Schauspieler oder Objekten durch die Verwendung von Kameras und Markern aufgezeichnet und die dadurch gesammelten Daten der Bewegungen auf den 3D-Charakter übertragen (vgl. Bennett & Carter, 2014, S. 5) oder zur Studie von Bewegungen genutzt (vgl. Kitagawa & Windsor, 2008, S. 1).

MoCap dient dabei nicht als kompletter Ersatz der Keyframe Animation, sondern als Ergänzung, um Arbeitswege zu erleichtern und dem Publikum die Feinheiten einer lebenden Figur durch Benutzung eines menschlichen Darstellers zu ermöglichen (vgl. Kadner, 2019, S. 59). MoCap Systeme können in zwei Hauptkategorien eingeteilt werden. Optische Systeme und nichtoptische Systeme.

4.2.1 Optische Systeme

Optische Systeme benutzen Kameras die die aufzunehmende Person oder Objekt aus mehreren Winkeln gleichzeitig aufnehmen. Die Aufnahmen werden in einem Computer verarbeitet. Dadurch kann die Person oder das Objekt in ein dreidimensionales Objekt transformiert werden (vgl. viscircle, o. D.). Ebenso werden Positionierungsdaten im dreidimensionalen Raum gesammelt. Optische Systeme werden dabei in zwei Kategorien eingeteilt. Systeme mit Markern und Markerlose Systeme.

Systeme mit Markern nutzen an dem Subjekt befindende Marker, die die Kameras erfassen. Damit diese Marker für die Kameras erkennbar werden, reflektieren sie entweder Licht (passive Marker) oder sie senden Licht aus (aktive Marker).

Passive Marker sind aus reflektierende Materialien, die aus sphärischen oder runden Formen bestehen. Form und Größe der Marker ist dabei abhängig von den für die Aufnahmen benutzen, Kameras. Diese Marker sind direkt am Subjekt durch zum Beispiel einen sogenannten *MoCap Suit* befestigt (vgl. Kitagawa & Windsor, 2008, S. 8). Um die Marker zu erfassen, senden die Kameras durch LED Licht oder Infrarot Strahlen aus, die dann von den Markern reflektiert werden. Anhand der Zeit, die das Licht braucht, um nach der Reflexion zurückzukehren, wird die Position des Markers im Raum durch Triangulierung berechnet (vgl. Bennett & Carter, 2014, S. 5).

Während der Dreharbeiten sollte allerdings beachtet werden, dass durch zu schnelle Bewegungen oder engerer Annäherung von Markern und aufgenommenen Subjekten, die Marker von der Motion Capture Software verwechselt werden, was zu Fehlern in der Verarbeitung der Daten führt (vgl. viscircle, o. D.).

Aktive Marker reflektieren kein Licht, sondern sind selbst Lichtquellen, die durch die Kameras erkannt werden. Dabei ist selbst nach Sichtverlust jeder Marker genau zu identifizieren (vgl. viscircle, o. D.). Durch die benötigte Stromversorgung der Lichtquellen muss beachtet werden, dass der Schauspieler durch Stromkabel in der Bewegungsfreiheit eingeschränkt wird.

Markerlose Systeme sind hingegen Systeme, die keine Marker benutzen. Dabei senden die um das Subjekt platzierten Kameras das Videomaterial an eine Software, die anhand von Mustererkennung die Bewegungen des Subjektes analysieren kann. Durch diese Technik ist es den Darstellern möglich, bequeme Kleidung zu tragen, mit der man ebenfalls Stunts machen könnte, ohne in der Bewegungsfreiheit eingeschränkt zu sein oder Ausrüstung zu beschädigen (vgl. viscircle, o. D.).



Abbildung 22: Vicon Optical Suits - Quelle: vfxvoice.com <https://www.vfxvoice.com/what-mocap-suit-suits-you/>

4.2.2 Nicht optische Systeme

Nichtoptische Systeme verwenden hauptsächlich Sensoren anstatt Kameras, um Daten zu ermitteln.

Mechanische Systeme

Bei der Verwendung von mechanischen Motion Capture Systemen tragen die aufzunehmende Person einen mechanischen Anzug mit Sensoren an den Gelenken. Diese Sensoren, die auch *Potentiometer* genannt werden (vgl. Votruba, 2001, S. 2), ermitteln anhand der Winkel der Gelenke (vgl. Kitagawa & Windsor, 2008, S. 26) Daten, die anschließend an die verarbeitende Software gesendet werden. Mechanische Systeme haben allerdings Schwierigkeiten, globale Translationen darzustellen. So gibt es häufig Probleme in der Darstellung von Höhen zum Beispiel bei der Besteigung von Treppen. Ebenso sind sie durch den zu tragenden Anzug extrem in der Bewegungsfreiheit einschränkend (vgl. Kitagawa & Windsor, 2008, S. 26).

Inertial Systeme

Inertial Systeme benutzen Sensoren an den Gelenken der aufgenommenen Subjekte und übermitteln ihre Position im relativen Raum kabellos an Empfänger. Da keine Kameras aufgebaut und angeschlossen werden müssen, ist das System sehr transportabel (vgl. Bennett & Carter, 2014, S. 5).

Magnetische Systeme

Bei magnetischen Systemen des Motion-Trackings werden Sensoren an den Gelenken des Subjektes platziert. Diese Sensoren messen den relativen Abstand zwischen Sensor und dem am nächsten liegenden Transmitter (vgl. Bennett & Carter, 2014, S. 5). Die Sensoren übermitteln dabei ihre Translation und Rotation direkt an die Software. Translations- und Rotationsberechnungen der Software sind nicht nötig (vgl. Kitagawa & Windsor, 2008, S. 25f).

Magnetische Systeme gehören zu den weniger verbreiteten Systemen, da sie anfällig für magnetische und elektronische Interferenzen, ausgehend von metallischen oder elektronischen Gerätschaften in der Umgebung, sind (vgl. Bennett & Carter, 2014, S. 5). Die generierten Daten aus magnetischen Sensoren sind ebenfalls tendenziell verrauscht (vgl. Kitagawa & Windsor, 2008, S. 25f), was optische oder inertial Systeme effektiver macht (vgl. Bennett & Carter, 2014, S. 5).

4.2.3 Motion Capture im Projekt

Im Projekt „Mr. Sandman“ wurde ein optisches Markerloses System des deutschen Start-ups *The Captury* verwendet. Durch die Verwendung des Motion Capture System ist es im Projekt möglich, die Bewegungen der Schauspieler in Echtzeit auf die sich in der Unreal Engine befindenden Charaktere zu übertragen.

„*Captury Live is a complete software + hardware solution for real-time markerless motion capture that provides everything that is needed to perform markerless capture of up to 3 actors.*“ (Captury Live Documentation, o. D.)

Hardware

Das Setup des Systems besteht aus Captury Live, welches aus acht verknüpften RGB Video Kameras, die um die Darsteller platziert werden, besteht (vgl. Jürgens & Correia & Masu, 2020, S. 3). Dabei wurden die Kameras durch eine LAN-Kabel-Verbindung an einen *D-Link Stackable Smart Managed Gigabit Switch* der *DGS-1510* Serie verbunden. Dieser Switch ermöglicht es dem ebenfalls am Switch angeschlossenen Rechner, der im Sandman Projekt ein Linux Kernel vorweist und mit dem Ubuntu Distribution läuft, auf die acht von den IP-Kameras übertragenden Video Streams zuzugreifen.

Das Linux-System händelt durch die Verwendung der von Captury bereitgestellten Software (Captury Studio) die eingehenden Videostreams (vgl. The Captury Documentation, o. D.). Die Software sendet anschließend die generierten Daten durch eine per WLAN oder LAN hergestellte Netzwerkverbindung an einen zweiten Rechner, der im Projekt ein Windows Rechner ist, auf dem die Unreal Engine mit dem geöffneten Projekt läuft.

Capture Volumne mit 8 Kameras

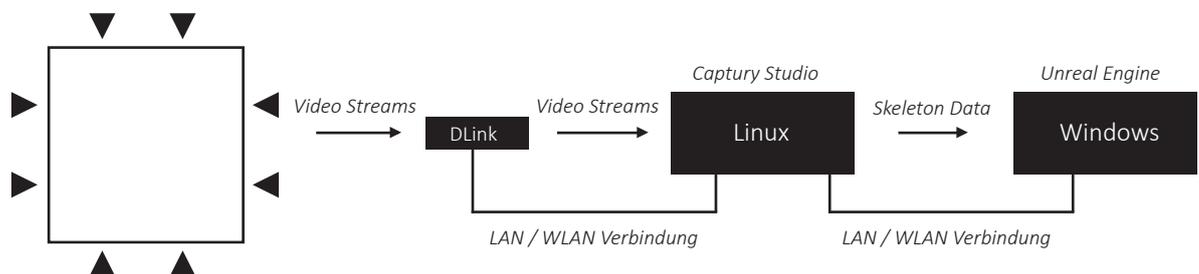


Abbildung 23: Hardware Aufbau im Projekt - Quelle: Eigene Abbildung, in Anlehnung an: Jürgens & Correia & Masu, 2020

Software

Damit der Switch einwandfrei läuft, ist die Installation des für den Switch vorgesehenen Treibers auf dem Linux-System notwendig. Dabei ist darauf zu achten, dass bestimmte Kernel Versionen des Linux-Systems nicht von dem vorgesehenen Treiber unterstützt werden oder Fehler aufweisen. Um das System zu benutzen, ist die Installation der von Captury zur Verfügung gestellten Software (Captury Live) vorausgesetzt.

Wenn das System eingerichtet oder bewegt wird, ist es nötig, die benutzten Kameras neu zu kalibrieren. Captury Live unterteilt diesen Prozess in drei Schritte:

Fokussieren der Kameras

Um die Kameras optimal zu fokussieren, ermöglicht die Software durch eine Kantenvisualisierung den aktuellen Fokuspunkt der Kamera auszumachen.

Dafür ist es nötig, den Focus Tab in der Toolbar auszuwählen.

Der Fokus der Kameras muss manuell über Drehung des an den Kameras befindenden Fokusrades verändert werden. Dabei zu beachten ist, dass eine spätere Veränderung des Fokus oder Position der Kamera eine komplette Rekalibrierung des Systems benötigt (vgl. The Captury Documentation, o. D.).

Lens Kalibrierung

Als nächsten Schritt ist es notwendig, in der Software die internen Parameter der Kameras zu bestimmen. Dafür wird ein sogenanntes *Calibrationsboard* benutzt. Das Calibrationboard ist ein flacher Untergrund, auf dem ein Gittermuster geklebt oder gedruckt ist.

Um die Kameras zu kalibrieren, muss zuerst zum *Calibration* Tab und anschließend auf *Lens* in der Toolbar gewechselt werden. Nachdem anschließend die Option *Capture Calibration Board* ausgewählt wurde, können die Kameras kalibriert werden.

Das Board muss dazu jeder Kamera präsentiert werden und füllend für die Kamera zu sehen sein (vgl. The Captury Documentation, o. D.).

Sobald alle Kamera Kalibrierungen abgeschlossen sind, wird durch das Betätigen der *Stop Capturing Calibration Board* Schaltfläche mit dem Sammeln neuer Kalibrierungsdaten aufgehört. Durch anschließender Auswahl von *Calibrate Lens* werden die gesammelten Kalibrierungsdaten verwertet und anschließend passende Kameraeinstellungen auf die Kameras angewendet (vgl. The Captury Documentation, o. D.).

Volume Kalibrierung

Da nun die Kameras platziert und eingestellt sind, folgt nun die Kalibrierung des Volumes, indem sich die Darsteller aufhalten, während ihre Bewegungen analysiert werden.

Um dieses Volume zu bestimmen, ist es notwendig, wie bei der Kalibrierung der Kameras, das Calibration Board zu benutzen. Hierfür ist es nötig, erneut *Capture Calibration Board* auszuwählen. Bei der Kalibrierung des Volumes ist es im Gegensatz zur Kalibrierung der Kameras wichtig, dass nicht nur eine Kamera ein freies Sichtfeld auf das Calibration Board hat. Den aktuellen Stand der Kalibrierung wird in Captury Studio durch Farbmarkierungen in den Kamerafenstern sowie in Prozentangaben angegeben (vgl. The Captury Documentation, o. D.).

Sobald genug Daten gesammelt wurden, kann auch hier durch Betätigung der Stop Capturing Calibration Board Schaltfläche das Sammeln von Daten beendet werden. Um das Volume mit den gesammelten Daten zu generieren, muss in der Nutzeroberfläche der Software die *Calibrate Volume* Taste betätigt werden. Sobald die Kalibrierung fertiggestellt ist, wurde die Position der Kameras gesetzt, um das Volume zu vervollständigen, benötigt die Software noch die Kalibrierung der Bodenfläche.

Um den Boden zu kalibrieren, reicht es, das Kalibrierungsboard für alle Kameras sichtbar in der Mitte des gewünschten Volumes zu legen. Der *Floor* wird anschließend durch das Betätigen der *Set Floor* Schaltfläche gesetzt, sobald er von der Software erkannt wird. Nachdem alles kalibriert wurde, ist es möglich, mit dem Tracking der Darsteller anzufangen. Sollte die Positionen und Kalibrierung der Kameras festgelegt sein und sich im Laufe der Arbeit am System nicht mehr ändern, ist es möglich, die gesammelten Kalibrierungsdaten abzuspeichern, um nach Neustart oder Absturz des Systems die Kalibrierung wiederherzustellen, ohne eine erneute Kalibrierung durchführen zu müssen. Um in Captury Studio eine neue Person anzulegen, gibt es zwei Methoden:

Die erste Methode, die durch Betätigen der *Search New* Schaltfläche ermöglicht wird, erlaubt es der Software in einem definierten Bereich automatisch nach einer neuen Person zu suchen. Der Bereich wird in der Software durch einen Kreis auf dem Boden markiert und kann durch Benutzung der Maus und Drücken der linken Maustaste verschoben und vergrößert werden.

Alternativ kann auch manuell durch Ziehen der *<new>* Schaltfläche ein Skeleton auf die Stelle gezogen werden, an dem die Person steht, die initialisiert werden soll. Wenn die Person erkannt wird, bildet sich eine Punktwolke um die Person (vgl. The Captury Documentation, o. D.). Damit die Person einfacher von der Software erkannt wird, sollte sie sich in einer Position aufstellen, in der ihre Gliedmaßen für die Software gut voneinander unterscheidbar sind. Durch die *Scaling Status* Anzeige der Software wird der aktuelle Stand der Berechnung des Skeletons nach der Initialisierung der Person angezeigt (vgl. The Captury Documentation, o. D.).

Nachdem nun die Person in Captury Studio hinzugefügt wurde, können die getrackten Daten durch eine Netzwerkverbindung an einen zweiten Rechner gesendet werden, auf dem die Unreal Engine betrieben wird. Die Software kann gleichzeitig die generierten Daten für die Post Production speichern oder als Sicherungsspeicher fungieren, wenn die Daten in anderen Programmen weiter verarbeitet werden sollen. Damit die Unreal Engine die von Captury Studio gesendeten Daten interpretieren kann, muss ein von Captury hergestelltes Plugin aus dem Epic Games Store installiert werden. Ebenso muss anschließend in der geöffneten Engine über die Menü Bar das Plugin Fenster geöffnet und das *Live Link* und das *CapturyLiveLink* Plugin aktiviert werden. Nach dem Aktivieren der Plugins ist ein Restart der Engine notwendig.

Nach Aktivierung der Plugins kann der *Live Link Client* über die Menu Bar unter dem Windows -> Virtual Production Menü erreicht werden. Im geöffneten Client muss nun eine neue *Source* hinzugefügt werden (vgl. Live Link, o. D.). Dabei steht uns nach der Aktivierung des von Captury entwickelten Plugins die Option *Captury Live* zur Verfügung, in der die lokale IPv4-Adresse des Rechners angegeben werden muss, auf dem Captury Studio betrieben wird.

Sobald die Verbindung hergestellt ist, ist es möglich, im Animation Blueprint der importierten Charaktere ein Live Link Node anzulegen. Der Live Link Node bietet die Auswahlmöglichkeit zwischen den von Captury getrackten Personen. Sobald eine Auswahl getroffen und das Blueprint Compiled wurde, werden die Bewegungen der getrackten Person auf den Charakter übertragen. Wichtig zu beachten ist dabei, dass die Skelettstruktur von Captury genutzt werden sollte, um die in der Unreal Engine laufende Charaktere zu riggen. Alternativ bietet die Unreal Engine auch die Option zum Retargeting des Skeletons.

4.3 Virtuelle Kamera und Kamera Tracking

Um Szenen im Animationsfilm aufzunehmen, wird wie auch im Live Action Dreh eine Kamera benutzt. Diese virtuelle Kamera kann entweder durch die Softwareparameter der Kamera und Animationen gesteuert werden oder durch ein greifbares Objekt außerhalb der Software. Diese Objekte können der Form einer existierenden Kamera gleichen oder in deutlich reduziert Form benutzt werden (vgl. Bennett & Carter, 2014, S. 5f).

So können Kamera-Nachbauten mit Einstellungsmöglichkeiten für Blende oder Verschlusszeit benutzt werden (vgl. Autodesk, 2009, S. 7) oder einfache Bauten, die nur das Gewicht einer Kamera nachempfinden. Dabei werden am Objekt Tracking Marker montiert, die die Position und Rotation der Kamera in der Software in Echtzeit manipulieren. Dies erlaubt Filmemachern, die virtuellen Charaktere im gleichen Wege zu filmen wie bei einem Live-Action-Dreh (Bennett & Carter, 2014, S. 5f).

Diese Technik ermöglicht es auch, bei Produktionen ohne virtuelle Charaktere angewendet zu werden. In diesen Produktionen werden Schauspieler vor einer LED Wand mit einer Kamera an der Marker montiert sind gefilmt, die die Position und Rotation der Kamera an einen in der Unreal Engine existierenden Kamera-Actor übertragen. Das Bild des Kamera -Actors wird dabei in Echtzeit auf die LED Wand hinter den Schauspielern übertragen. Dies erlaubt eine Mischung von realen Schauspielern, Sets und Props mit virtuellen Hintergründen in Echtzeit.



Abbildung 24: Virtual Production in „The Mandalorian“ - Quelle: fxguide.com

4.3.1 Virtuelle Kamera im Projekt

Um die durch die Schauspieler und Capture gesteuerten Charaktere in der Unreal Engine aufzunehmen, wurde eine *Cine Camera Actor* in der Unreal Engine angelegt, dessen Bewegung und Rotation von einem Motion Tracking Marker gesteuert werden, der an einem Rig montiert wurde. An diesem Rig ist ebenso ein Monitor montiert, der der filmenden Person ermöglicht, ein Live Bild des Cine Camera Actor zu sehen.



Abbildung 25 & 26: Kamera Rig mit VIVE Tracker - Quelle: Eigene Abbildung

Nachdem das Wunschrig aufgebaut wurde, wird durch ein montierten *VIVE Tracker 3.0* die Bewegungen des Riges auf den Cine Camera Actor übertragen. Dazu müssen die Base Stations von VIVE wie auch die Tracker kalibriert werden. Ebenso ist die Verwendung von *SteamVR* eine Voraussetzung. Nachdem das VIVE System kalibriert und einsatzfähig ist, sind die aktivierten Marker im Live Link Fenster unter der Option *LiveLinkXR Source* sichtbar. Die Tracker können anschließend mit einer *Local Update Rate in Hz* von 90 hinzugefügt werden.

Die eigentliche Kamera besteht aus einer Verschachtelung von im Editor platzierten Actors. Als Erstes wird ein normaler Actor über das Place Actor Menü oder über die Create Schaltfläche der Main-Toolbar erstellt. Dieser Actor dient, als Möglichkeit der Kamera Position einen manuellen Offset der Transformationsdaten zu geben. Dadurch kann die Kamera abseits der Daten des Trackers verschoben werden.

Dieser Offset Actor dient in der Verschachtelung des Kamera-Setups als höchstes Element und überträgt somit seine Offsetdaten auf die hierarchisch unter sortierten Actors. Der nächste Schritt ist die Erstellung des Actors, der die vom Tracker gesendeten Daten zur Transformation der Kamera benutzt. Hierfür ist es zuerst notwendig, ein Blueprint anzulegen.

Ein *Blueprint Class* des Types *Actors* kann über die Add Schaltfläche im Content Browser angelegt werden. Durch Doppelklick auf die angelegte Blueprint Class, öffnet sich das Blueprint Editor Fenster. In diesem Fenster ist es möglich, der Klasse weitere Components hinzuzufügen. Zuerst sollte ein *Live Link Controller Component* der Blueprint Class hinzugefügt werden, in dessen Einstellungen im Live Link Bereich unter der Auswahl *Subject Representation* der zuvor erwähnte Motion Tracker ausgewählt wird. Der Tracker bestimmt nun die Transformation und Rotation des Actors und allen hierarchisch untergeordneten Elementen.

Der letzte Actor, der erstellt werden muss, ist der *Cine Camera Actor*. Der Cine Camera Actor ist ein erweiterter Camera Actor, der durch zusätzliche Einstellungsmöglichkeiten das Verhalten einer Filmkamera repliziert. Durch Einstellungen des Objektivs und der Fokussierung ermöglicht der Cine Camera Actor realistische Szenen abzubilden und dabei die Qualitäts- und technischen Standards der Industrie einzuhalten (vgl. Cine Camera Actor, o. D.).

Die Einstellungsmöglichkeiten des Cine Camera Actors erlauben die Veränderung der Sensor Dimensionen sowie die freie Wahl der Parameter der Kameralinse. Wichtig zu beachten ist dabei, dass die Transformations- und Rotationseinstellungen des Cine Camera Actor nicht verändert werden sollten.

Die Position des Cine Camera Actors wird nun nicht durch die eigenen Transformations- und Rotationseinstellungen bestimmt, sondern durch die Einstellungen des übergeordneten Tracking- und Offset-Actor.



Abbildung 27: Kamera Setup Hierarchie - Quelle: Eigene Abbildung

Damit die Person, die die Kamera bedient, das Live Bild des Cine Camera Actors über einen am Rig montierten Monitor sehen kann, müssen Plugins in der Engine wie auch Hardware am Rechner installiert werden. Bei der erforderlichen Hardware handelt es sich um die *Blackmagic Decklink Card*, die im Rechner, auf dem die Unreal Engine läuft, verbaut werden muss. In der Unreal Engine muss nun das Plugin *Blackmagic Media Player* und *Media Framework Utilities* aktiviert werden. Das Aktivieren des Plugins erfordert einen Neustart der Unreal Engine.

Im geöffneten Projekt ist es jetzt möglich, im Content Browser unter dem Bereich Media ein *Blackmagic Media Output* zu erstellen. Im geöffneten Blackmagic Media Output muss jetzt die verbaute Decklink Card mit der gewünschten Auflösung und FPS angegeben werden.

Als nächsten Schritt kann der *Media Capture* Fenster der Unreal Engine über die Menu Bar geöffnet werden. Sollte die Unreal Engine 5 verwendet werden, befindet sich das Media Capture Fenster unter dem Virtual Production Abschnitt.

Im Media Capture Fenster kann jetzt im Bereich Media Viewport Capture ein Viewport Capture hinzugefügt werden. In diesem muss jetzt unter Media Output der vorher angelegte Blackmagic Media Output eingetragen werden. Ebenso muss unter der *Cameras* Option der angelegte Cine Camera Actor eingetragen werden.

Durch Betätigen der *Capture* Schaltfläche wird nun ein Live Bild der Kamera an den mit der Blackmagic Decklink Card verbundenen Monitor gesendet. Jetzt ist es möglich durch Bewegung des Riges die Kamera in der Engine zu steuern und gleichzeitig das Bild der Kamera auf dem am Rig befindenden Monitor zu sehen.

5. Ergebnisse der Techniken

Im weiteren Text werden die Ergebnisse der vorher erklärten Techniken am Beispiel „Mr. Sandman“ analysiert. Dabei werden auftretende Komplikationen in der Anwendung, Qualität der Ergebnisse und die unterstützenden Möglichkeiten der Unreal Engine besprochen.

5.1 Ergebnisse der Environments

Die Unreal Engine wurde ursprünglich für Videospiele geschaffen (vgl. Plante, 2012). Viele der Features, die für Videospiele entwickelt wurden, kommen nun in der virtuellen Produktion für Filme zu guten. So wird die Erstellung von Umgebungen in der Unreal Engine 4 oder 5 durch integrierte Tools wie dem Landscape Mode unterstützt. Dieser Mode ermöglicht es, mit einfach zu bedienenden Tools Umgebungen umzusetzen, dabei lassen sich in wenigen Klicks erste Ergebnisse erzielen. Aber auch detailreichere Bearbeitungen der Umgebungen sind durch tiefgehende Einstellungen möglich. Die Engine ermöglicht auch den Import von Heightmaps zur Gestaltung der Landschaft. Damit ist die Benutzung von externen Programmen wie World Machine oder Gaea möglich.

Da die Umgebungen aus vielen einzelnen Assets bestehen, ermöglicht die Unreal Engine einen einfachen Import von externen Objekten, Skeletons und Animationen. Dabei ist es möglich, die Objekte zusammen mit den dazugehörigen Texturen und Animationen zu importieren. Der Import erfolgt dabei in wenigen Klicks und unterstützt gängige Formate wie OBJ und FBX.

Durch den Material Editor der Unreal Engine ist es in der Engine möglich, Texturen anzufertigen. Dadurch ist man dabei nicht zwingend auf externe Programme angewiesen. Dabei benötigt die Arbeit mit dem Material Editor Erfahrung, da die Masse und Funktion der benutzbaren Nodes erlernt werden muss.

Das Platzieren der Assets funktioniert genauso wie das Gestalten der Landschaft durch ein einfach zu benutzendes Paint Tool oder durch Drag-and-Drop der Assets vom Content Browser in die Landschaft. Die Qualität der Assets im Landscape hängt natürlich von den erstellten Assets ab. Sollte nicht die Kapazität bestehen, um eine Vielzahl hochqualitativer Assets zu modellieren und zu texturieren, bietet die Unreal Engine durch die Megascans Library eine Sammlung von hochqualitativen Assets und Texturen, die auch in kommerziellen Projekten benutzt werden dürfen.



Abbildung 28: Platzierte Assets - Quelle: Eigene Abbildung

Die Möglichkeiten der Lichtsetzung in der Unreal Engine variiert in der verwendeten Version. Das Platzieren von Basis Lichtern wie Point Lights oder Spot Lights erfolgt durch Drag-and-Drop oder durch die Main Toolbar. Die Einstellungsmöglichkeiten reichen dabei in Kombination mit der Benutzung eines PPV von einfachen Grundeinstellungen der Helligkeit bis hin zu erweiterten Einstellungen bezüglich Light Shafts oder Ray Tracing.

Auch zur Erstellung von Atmosphären bietet die Unreal Engine durch den Light Mixer die Möglichkeit, schnell erste Ergebnisse zu erzielen, die anschließend detailreicher ausgearbeitet werden können. So kann zum Beispiel relativ schnell eine Atmosphäre mit Wolken und Sonne geschaffen werden, die im Falle des Sandman Projektes laufend verfeinert wird.



Abbildung 29: Atmosphäre im Sandman Projekt - Quelle: Eigene Abbildung

Das in der Unreal Engine 5 verfügbare Lumen System ist sehr einfach zu aktivieren und sorgt durch in Echtzeit berechnete Global Illumination für eine realistische und Performance freundliche Lichtstimmung.

Durch das ebenfalls neue Nanite System der Unreal Engine lassen sich auch hochauflösende Assets Performance freundlich in größeren Mengen benutzen. Dabei wird einem ebenfalls das Erstellen von LODs für in der Distanz sichtbare Assets abgenommen.

Insgesamt erfordert die Erstellung einer Umgebung eine gewisse Einarbeitungszeit in die vielen Kategorien, aus denen sie zusammengesetzt ist. Auch die Erstellung der Umgebung kann basierend auf der gewünschten Komplexität, Detailreichtum und Größe einen großen Zeitaufwand kosten.

Die Unreal Engine ermöglicht es allerdings durch eine große, aktive Community und gegebenen Tools, die ein intensives und einfaches ausprobieren von Parameter ermöglichen, eine steile Lernkurve im Bereich Environments zu erzielen.

5.2 Ergebnisse des Motion Captures

Der Aufbau des Motion Capture Systemes von Captury ist simple und benötigt lediglich ein paar Minuten. Dadurch, dass zur Benutzung von Captury nur acht Kameras und ein Rechner benötigt werden, ist das ganze System sehr transportabel. Die Kalibrierung der von Captury entwickelten Software ist durch nur wenige Klicks durchführbar und kann, sollte das System statisch aufgebaut werden, für spätere Neustarts abgespeichert werden.

Worauf geachtet werden sollte, sind Treiberversionen und dessen Kompatibilität mit Betriebssystemen und anderer Software. So ist im Projekt der Treiber des verwendeten Switches nicht mit allen Kernel Versionen des Linux Systems kompatibel, was der Software verhindert, auf die von den Kameras gesendete Videostreams zuzugreifen.

Zur Übertragung der Daten in die Unreal Engine muss eine Netzwerkverbindung zwischen den Rechnern hergestellt werden. Eine Netzwerkverbindung herzustellen kann mitunter zu Problemen führen, da es viele Einflussquellen wie DHCP, Port oder Firewall gibt, die die Verbindung blockieren können.

Captury stellt der Unreal Engine im Epic Games Store ein Plugin zur Verfügung, was mit der Benutzung des Live Link Plugins einen sehr einfachen Weg ermöglicht, die gesendeten Tracking Daten in Echtzeit in der Engine zu verwenden.

Dafür muss ein Retargeting der von Captury gesendeten Skeleton Daten, auf das Skeleton des Charakters, der sich in der Unreal Engine befindet, stattfinden. Es kann allerdings auch das von Captury benutzte Rig für die in der Unreal Engine verwendeten Charaktere benutzt werden.

Die Resultate des Motion Trackings können in der Unreal Engine für die Dreharbeiten verwendet werden, allerdings ist zu beachten, dass durch das optische Markerlose System es immer wieder zu Fehlern kommen kann, wenn sich die Darsteller zu nah kommen oder Gliedmaßen nicht für die Kameras sichtbar sind. Für das Endprodukt sollten die aufgenommenen Daten durch externe Programme wie Maya bereinigt werden, damit Animationen vereinfacht und *Noise* reduziert wird.

5.3 Ergebnisse des Kamera Tracking und der Virtuellen Kamera

Das Kamera-Tracking benötigt die Installation und Kalibrierung des VIVE Systems und die Installation von SteamVR. Die Installation und Kalibrierung der Systeme sind im jeweiligen Installationsvorgang dokumentiert. Nach der Installation und Einrichtung des VIVE Systemes können die aktivierten Marker direkt über das Live Link Plugin in die Engine hinzugefügt werden. Dieser Schritt benötigt wie auch bei der Hinzufügung der von Captury gesendeten Daten nur wenige Klicks und funktioniert problemlos.

Das Anlegen des Blueprints und der Hierarchie besteht nur aus wenigen Schritten und ist durch die einfach zu bedienenden Schaltflächen und der Drag-and-Drop Funktion im *Outliner* durchzuführen.

Sollte man optisches Feedback der Kamerasicht sehen wollen, ist es nötig eine Black Magic Decklink Card mit SDI Anschluss im PC zu verbauen. Je nach gewünschter Version kann diese zwischen 345€ und 1115€ kosten (Stand 19.01.2023) (vgl. blackmagicdesign, o. D.).

Die Aktivierung des benötigten Plugins sowie auch die Erstellung des Black Magic Media Output erfolgen auf die übliche Weise und sind damit schnell aktiviert und angelegt. Durch eine SDI Kabelverbindung von der Decklink Card zum Monitor kann nun das Bild der Kamera übertragen werden.

Ein Problem, das auftreten wird, sind die verschiedenen Positionen der Kamera und der Darsteller im Captury und VIVE Volumne. Da für die Kamera und die Schauspieler nicht dasselbe Tracking System benutzt wird, unterscheiden sich die Größe und Rotation der Volumnes. Dies führt dazu, dass obwohl der Kameramann und der Darsteller sich direkt gegenüber stehen, die Kamera in der Engine und der gespielte Charakter sich nicht gegenüber stehen. Dies kann durch Veränderung der Transformations- und Rotationseinstellungen des Offset Actors in der Hierarchie der Kamera angepasst werden.

5.4 Fazit

Die Unreal Engine ist ein mächtiges Tool zur Erstellung virtueller Welten. Durch die Benutzung einer Game Engine stehen uns viele hilfreiche Tools zur Verfügung, die statt zur Gestaltung von Videospieldwelten auch für die Produktion von Filmen genutzt werden können.

Dabei bietet die Unreal Engine die Freiheit, die Umgebung durch die Benutzung der einfach zu bedienenden integrierten Tools, nach eigenen Belieben und unabhängig vom Art Style zu gestalten. Durch nur wenig Aufwand und Einarbeitungszeit lassen sich einfache Umgebungen gestalten. Doch auch für höher Gesetze Ziele, lässt sich die Unreal Engine durch tiefer gehende Einstellungsmöglichkeiten benutzen. Damit ist sie für Hobbynutzer als auch in der professionellen Produktion einsetzbar.

Durch den Release der Unreal Engine 5 stehen uns das Nanite und Lumen System zur Verfügung, dessen Performance und Beleuchtungsqualität eine deutliche Verbesserung gegenüber der Möglichkeiten der Unreal Engine 4 bieten. Das Nanite System erlaubt es hochqualitative Assets in großer Anzahl Performance freundlich darzustellen. Dies ist ein sehr großer Pluspunkt, da die Produktion des Animationsfilmes hauptsächlich in Echtzeit stattfindet. Durch bereits integrierte Plugins wie Live Link ermöglicht es die Engine bereits durch wenige Klicks multiple externe Systeme wie das VIVE System zu verknüpfen. Dadurch lassen sich Tracking Daten der VIVE Tracker direkt ohne das aufwendige Schreiben von C++ Code verwerten.

Da die Unreal Engine durch eine Vielzahl von Nutzer (vgl. gamedesigning, 2023) sowie auch professionellen Firmen genutzt wird, bieten viele externe Firmen ihre Produkte bereits mit mitgelieferten Plugins zur Kompatibilität zur Unreal Engine an. So ist es möglich, Modelle aus Maya direkt über eine Schaltfläche in die Unreal Engine zu importieren. Auch das im Projekt verwendete Motion Capture System der Firma The Captury bietet bereits ein existierendes Plugin auf dem Epic Games Marktplatz an.

Die Unreal Engine bietet durch gegebene Tools und Plugins einen einfachen Einstieg in die virtuelle Produktion. Jedoch ist zu beachten, dass obwohl vorhandene Tools Arbeit ersparen, sie erst erlernt werden müssen. Durch tiefer gehende Einstellungsmöglichkeiten diverser Tools, ist eine Einarbeitungszeit notwendig, um die bestmöglichen Ergebnisse im Bereich Qualität und Performance zu erzielen.

6. **Abbildungsverzeichnis**

Abbildung 1: colinbarrebrisebois.com (November, 2015) - Zugegriffen: 22.01.2023.

<https://colinbarrebrisebois.com/2015/11/06/finding-next-gen-part-i-the-need-for-robust-and-fast-global-illumination-in-games/>

Abbildung 2: 3d1.com.br (Februar, 2019) - Zugegriffen: 22.01.2023.

<https://3d1.com.br/noticia/68459>

Abbildung 3: peterfalkingham.com (Mai, 2019) - Zugegriffen: 22.01.2023.

<https://peterfalkingham.com/2019/05/17/photogrammetry-testing-reality-capture-commercial-software/>

Abbildung 10: Unreal Engine Landscape Quick Start Guide - Zugegriffen: 22.01.2023.

<https://docs.unrealengine.com/5.0/Images/building-virtual-worlds/landscape-outdoor-terrain/landscape-quick-start/10-t-landscape-tools.png>

Abbildung 11: Unreal Engine Landscape Technical Guide - Zugegriffen: 22.01.2023.

<https://docs.unrealengine.com/5.0/Images/building-virtual-worlds/landscape-outdoor-terrain/landscape-technical-guide/landscape-tech-components.jpg>

Abbildung 12: Unreal Engine Landscape Technical Guide - Zugegriffen: 22.01.2023.

<https://docs.unrealengine.com/5.0/Images/building-virtual-worlds/landscape-outdoor-terrain/landscape-technical-guide/landscape-tech-component-sections.jpg>

Abbildung 13: Unreal Engine Landscape Technical Guide - Zugegriffen: 22.01.2023.

<https://docs.unrealengine.com/5.0/en-US/landscape-technical-guide-in-unreal-engine/>

Abbildung 16: Unreal Engine Landscape Quick Start Guide - Zugegriffen: 22.01.2023.

<https://docs.unrealengine.com/5.0/Images/building-virtual-worlds/landscape-outdoor-terrain/landscape-quick-start/32-t-landscape-material-final.png>

Abbildung 17: Unreal Engine Landscape Quick Start Guide - Zugegriffen: 22.01.2023.

<https://docs.unrealengine.com/5.0/Images/building-virtual-worlds/landscape-outdoor-terrain/landscape-quick-start/40-t-landscape-blend-layer.png>

Abbildung 18: Unreal Engine Environment Light Mixer - Zugegriffen: 22.01.2023.

https://docs.unrealengine.com/5.0/Images/building-virtual-worlds/lighting-and-shadows/environmental-lighting/environment-light-mixer/1_EnvLightMixer.webp

Abbildung 21: Unreal Engine Light Shafts - Zugegriffen: 22.01.2023.

<https://docs.unrealengine.com/5.0/Images/building-virtual-worlds/lighting-and-shadows/features-of-lights/light-shafts/5LightShafts.png>

Abbildung 22: vfxvoice.com - Zugegriffen: 22.01.2023.

https://www.vfxvoice.com/wp-content/uploads/2019/05/PIX-2-Vicon_suit1.jpg

Abbildung 24: fxguide.com - Zugegriffen: 22.01.2023.

<https://www.fxguide.com/wp-content/uploads/2020/02/HUC-010258.jpg>

7. Literaturverzeichnis

Actors. (o. D.). Epic Games. Online verfügbar unter <https://docs.unrealengine.com/5.0/en-US/actors-in-unreal-engine/>, zuletzt geprüft am 08.02.2023.

Autodesk (2009): The New Art of Virtual Moviemaking. Online verfügbar unter http://images.autodesk.com/apac_grtrchina_main/files/the_new_art_of_virtual_moviemaking_-_autodesk_whitepaper1.pdf, zuletzt geprüft am 24.10.2022.

Auto Exposure. (o. D.). Epic Games. Online verfügbar unter <https://docs.unrealengine.com/5.0/en-US/auto-exposure-in-unreal-engine/>, zuletzt geprüft am 23.01.2023.

Beane, Andy (2012): 3D Animation Essentials: Wiley.

Bennett, Joel; Carter, Chris (2014): Adopting virtual production for animated filmmaking. Singapore.

Blackmagic Design (o. D.): DeckLink. Online verfügbar unter <https://www.blackmagicdesign.com/de/products/decklink>, zuletzt geprüft am 24.01.2023.

CaptureLive Dokumentation. (o. D.). The Capture. Online verfügbar unter <http://doc.capture.com/CaptureLive/index.html>, zuletzt geprüft am 24.01.2023.

Cine Camera Actor. (o. D.). Epic Games. Online verfügbar unter <https://docs.unrealengine.com/5.0/en-US/cinematic-cameras-in-unreal-engine/>, zuletzt geprüft am 24.01.2023.

Clark, Curtis; Okada, Daryn; Reisner, David; Stump, Dave; Edlund, Richard; Levinson, Lou et al. (2008): Technology Committee Report. American Society of Cinematographers. Online verfügbar unter <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7269937>, zuletzt geprüft am 23.01.2023.

Content Browser. (o. D.). Epic Games. Online verfügbar unter <https://docs.unrealengine.com/5.0/en-US/content-browser-in-unreal-engine/>, zuletzt geprüft am 22.11.2022.

Creating Landscapes. (o. D.). Epic Games. Online verfügbar unter <https://docs.unrealengine.com/5.0/en-US/creating-landscapes-in-unreal-engine/>, zuletzt geprüft am 22.11.2022.

Deloitte (2020): The future of content creation: Virtual production. Unlocking creative vision and business value. Online verfügbar unter <https://www2.deloitte.com/content/dam/Deloitte/us/Documents/technology-media-telecommunications/us-tmt-the-future-of-content-creation-virtual-production.pdf>, zuletzt geprüft am 07.04.2022.

Directional Lights. (o. D.). Epic Games. Online verfügbar unter <https://docs.unrealengine.com/5.0/en-US/directional-lights-in-unreal-engine/>, zuletzt geprüft am 23.01.2023.

Environment Light Mixer. (o. D.). Epic Games. Online verfügbar unter <https://docs.unrealengine.com/5.0/en-US/environment-light-mixer-in-unreal-engine/>, zuletzt geprüft am 23.01.2023.

Foliage Mode. (o. D.). Unreal Engine. Online verfügbar unter <https://docs.unrealengine.com/5.0/en-US/foliage-mode-in-unreal-engine/>, zuletzt geprüft am 30.01.2023.

Frequently Asked Questions. (o. D.). Unreal Engine. Online verfügbar unter <https://www.unrealengine.com/en-US/faq>, zuletzt geprüft am 23.01.2023.

gamedesigning (2023): How to Choose the Best Video Game Engine. Online verfügbar unter <https://www.gamedesigning.org/career/video-game-engines/>, zuletzt geprüft am 28.01.2023.

Holzweißig, Julia (2022): Sandman Packaging. Online verfügbar unter https://drive.google.com/file/d/1Agf-Vut4i1sQ5woYDI37clbv8XeYWnbM/view?usp=share_link, zuletzt geprüft am 23.01.2023.

Jablotschkin, Larissa (2021): Tutorial-Video = Problemlösung? Eine Untersuchung anhand der Arbeit mit Adobe After Effects. TH-OWL, Lemgo.

Jürgens, Stephan; Correia, Nuno N. Correia; Masu, Raul (2020): Designing Glitch Procedures and Visualisation Workflows for Markerless Live Motion Capture of Contemporary Dance.

Kadner, Noah (2019): The Virtual Production Field Guide. Epic Games. Online verfügbar unter <https://cdn2.unrealengine.com/vp-field-guide-v1-3-01-f0bce45b6319.pdf>, zuletzt geprüft am 06.04.2022.

Kadner, Noah (2021): The Virtual Production Field Guide. Volume 2. Epic Games. Online verfügbar unter <https://cdn2.unrealengine.com/Virtual+Production+Field+Guide+Volume+2+v1.0-5b06b62cbc5f.pdf>, zuletzt geprüft am 06.04.2022.

Kitagawa, Midori; Windsor, Brain (2008): MoCap for artists. Workflow and techniques for motion capture. Amsterdam: Elsevier/Focal Press.

Landscape Flatten Tool. (o. D.). Epic Games. Online verfügbar unter <https://docs.unrealengine.com/5.0/en-US/landscape-flatten-tool-in-unreal-engine/>, zuletzt geprüft am 23.01.2023.

Landscape Material Layer Blending. (o. D.). Epic Games. Online verfügbar unter <https://docs.unrealengine.com/5.0/en-US/landscape-material-layer-blending-in-unreal-engine/>, zuletzt geprüft am 23.01.2023.

Landscape Quick Start Guide. (o. D.). Epic Games. Online verfügbar unter <https://docs.unrealengine.com/5.0/en-US/landscape-quick-start-guide-in-unreal-engine/>, zuletzt geprüft am 22.11.2022.

Landscape Sculpt Tool. (o. D.). Epic Games. Online verfügbar unter <https://docs.unrealengine.com/5.0/en-US/landscape-sculpt-tool-in-unreal-engine/>, zuletzt geprüft am 23.01.2023.

Landscape Smooth Tool. (o. D.). Epic Games. Online verfügbar unter <https://docs.unrealengine.com/5.0/en-US/landscape-smooth-tool-in-unreal-engine/>, zuletzt geprüft am 23.01.2023.

Landscape Technical Guide. (o. D.). Epic Games. Online verfügbar unter <https://docs.unrealengine.com/5.0/en-US/landscape-technical-guide-in-unreal-engine/>, zuletzt geprüft am 23.01.2023.

Light Shafts. (o. D.). Epic Games. Online verfügbar unter <https://docs.unrealengine.com/5.0/en-US/using-light-shafts-in-unreal-engine/>, zuletzt geprüft am 23.01.2023.

Light Types and Their Mobility. (o. D.). Epic Games. Online verfügbar unter <https://docs.unrealengine.com/5.0/en-US/light-types-and-their-mobility-in-unreal-engine/>, zuletzt geprüft am 23.01.2023.

Live Link. (o. D.). Epic Games. Online verfügbar unter <https://docs.unrealengine.com/5.0/en-US/live-link-in-unreal-engine/>, zuletzt geprüft am 24.01.2023.

Lumen Global Illumination and Reflections. (o. D.). Unreal Engine. Online verfügbar unter <https://docs.unrealengine.com/5.0/en-US/lumen-global-illumination-and-reflections-in-unreal-engine/>, zuletzt geprüft am 27.01.2023.

muthmedia (o. D.): Alles zum Thema Virtual Production. Von der Funktionsweise bis zu den Vorteilen virtueller Produktionen. Hg. v. muthmedia. Online verfügbar unter <https://nur-muth.com/blog/was-ist-virtual-production-erklaerung-vorteile-funktionsweise>, zuletzt geprüft am 30.10.2022.

Nanite Virtualized Geometry. (o. D.). Epic Games. Online verfügbar unter <https://docs.unrealengine.com/5.0/en-US/nanite-virtualized-geometry-in-unreal-engine/>, zuletzt geprüft am 23.01.2023.

Photogrammetrie Software. (o. D.). Autodesk. Online verfügbar unter <https://www.autodesk.de/solutions/photogrammetry-software>, zuletzt geprüft am 23.01.2023.

Plante, Chris (2012): Better with age: A history of Epic Games. How Epic Games went from being the underdog based in a garage to one of the most successful game development studios on the planet. polygon.com. Online verfügbar unter <https://www.polygon.com/2012/10/1/3438196/better-with-age-a-history-of-epic-games>, zuletzt geprüft am 24.01.2023.

Post Process Effects. (o. D.). Epic Games. Online verfügbar unter <https://docs.unrealengine.com/5.0/en-US/post-process-effects-in-unreal-engine/>, zuletzt geprüft am 23.01.2023.

Quixel (o. D.): Bridge. Online verfügbar unter <https://quixel.com/bridge>, zuletzt geprüft am 23.01.2023.

Quixel (o. D.): Megascans. Online verfügbar unter <https://quixel.com/megascans>, zuletzt geprüft am 23.01.2023.

rastergrid (2010): GPU based dynamic geometry LOD. Online verfügbar unter <https://www.rastergrid.com/blog/2010/10/gpu-based-dynamic-geometry-lod/>, zuletzt geprüft am 23.01.2023.

Reinschmidt, Jan Andres (2023): Möglichkeiten zur kosteneffizienten Umsetzung von Virtual Production. TH-OWL, Detmold.

Rogers, Sol (2020): Virtual Production And The Future Of Filmmaking. An Interview with Ben Grossmann, Magnopus. Hg. v. Forbes. Online verfügbar unter <https://www.forbes.com/sites/solrogers/2020/01/29/virtual-production-and-the-future-of-filmmaking-an-interview-with-ben-grossman-magnopus/?sh=715f77aa75d4>, zuletzt geprüft am 24.10.2022.

Sculpt Mode. (o. D.). Epic Games. Online verfügbar unter <https://docs.unrealengine.com/5.0/en-US/landscape-sculpt-mode-in-unreal-engine/>, zuletzt geprüft am 23.01.2023.

tu-ilmenau (o. D.): Global Illumination. Online verfügbar unter <https://www.tu-ilmenau.de/universitaet/fakultaeten/fakultaet-wirtschaftswissenschaften-und-medien/profil/institute-und-fachgebiete/fachgebiet-virtuelle-welten-und-digitale-spiele/forschung/global-illumination>, zuletzt geprüft am 23.01.2023.

viscircle: Einsteigerguide: Was versteht man eigentlich unter Motion Capture? Online verfügbar unter <https://viscircle.de/einsteigerguide-was-versteht-man-eigentlich-unter-motion-capture-2/>, zuletzt geprüft am 24.12.2022.

Votruba, Peter (2001): Ablauf eines Motion Capture - Prozesses. Online verfügbar unter https://www.cg.tuwien.ac.at/courses/Seminar/SS2001/mocap/MotionCapture_Ablauf.pdf, zuletzt geprüft am 24.12.2022.

EIGENSTÄNDIGKEITSERKLÄRUNG

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Bachelorarbeit selbstständig und ohne unerlaubte fremde Hilfe angefertigt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Löhne, den 08.02.2023

A handwritten signature in black ink, reading "Dennis Prüßmeier". The signature is written in a cursive style and is positioned above a horizontal line that extends to the right.

Dennis Prüßmeier

Danke,

Rico, Jennifer, Malte, Julia, Leonie, Jan, Louisa, Marion und Reiner
für Eure Hilfe.