Hochschule Fulda - University of Applied Sciences

Fachbereich Angewandte Informatik

Entwicklung eines Videospielprototypen

als ”Ein-Mann-Videospielentwickler“ auf der Unreal Engine 5 mit Hilfe von

KI-Systemen

**Bachelorarbeit**

|  |  |
| --- | --- |
| Vorgelegt von: | Matrikelnummer: |
| Nicolas Justus Taylor | 836415 |

Im Studiengang:

Digitale Medien (B.Sc.)

WS 2023 / 2024

Erstprüfer: Prof. Dr. Christian Fischer

Zweitprüfer: Dipl.-Ing. Peter Klingebiel

Fulda, 23. Oktober 2023

# Abstract

# Inhaltsverzeichnis

[Abstract I](#_Toc148908567)

[Inhaltsverzeichnis II](#_Toc148908568)

[1 Einleitung 1](#_Toc148908569)

[1.1 Motivation 2](#_Toc148908570)

[1.2 Forschungsfrage und Forschungsmethode 3](#_Toc148908571)

[1.3 Gliederung der Arbeit 3](#_Toc148908572)

[2 Theoretischer Hintergrund 4](#_Toc148908573)

[2.1 Begriffsdefinitionen 4](#_Toc148908574)

[2.1.1 Intelligenz 4](#_Toc148908575)

[2.1.2 Künstliche Intelligenz 4](#_Toc148908576)

[2.1.3 KI-Systeme 4](#_Toc148908577)

[2.1.4 Prompt 5](#_Toc148908578)

[2.1.5 NPC 5](#_Toc148908579)

[2.1.6 Ein-Mann-Videospielentwickler 5](#_Toc148908580)

[2.2 Mesh - Vertices, Edges und Faces 6](#_Toc148908581)

[2.2.1 Textur 7](#_Toc148908582)

[3 Methodik 8](#_Toc148908583)

[3.1 Auswahl und Beschreibung der KI-Systeme 8](#_Toc148908584)

[3.1.1 ChatGPT 8](#_Toc148908585)

[3.1.2 Midjourney 9](#_Toc148908586)

[3.1.3 PIFuHD 9](#_Toc148908587)

[3.1.4 Voice.AI 9](#_Toc148908588)

[3.1.5 Adobe Enhanced Speech 9](#_Toc148908589)

[3.2 Beschreibung der Tools und Technologien 9](#_Toc148908590)

[3.2.1 Unreal Engine 5 9](#_Toc148908591)

[3.2.2 Blender 9](#_Toc148908592)

[3.2.3 Blender 10](#_Toc148908593)

[3.2.4 Game Rig Tools (Blender Addon) 10](#_Toc148908594)

[3.2.5 Materialize 10](#_Toc148908595)

[3.2.6 Audacity 10](#_Toc148908596)

[4 Umsetzung 11](#_Toc148908597)

[4.1 Entwicklung der Spieleidee mit Hilfe von ChatGPT 12](#_Toc148908598)

[4.2 Meilenstein: Hauptfigur 15](#_Toc148908599)

[4.2.1 Erzeugen eines 3D-Modells mit Hilfe von PIFuHD 27](#_Toc148908600)

[4.2.2 Verringern von Vertices in BlenderII (Nochmal überarbeiten) 29](#_Toc148908601)

[4.2.3 Nachbesserung des erzeugten Modells in Blender 31](#_Toc148908602)

[4.2.4 Texturieren in Blender 31](#_Toc148908603)

[4.2.5 Rigging in Blender 33](#_Toc148908604)

[4.2.6 Einfügen des Hauptcharakters in Unreal Engine 5 33](#_Toc148908605)

[4.3 Meilenstein: Gebäude 33](#_Toc148908606)

[4.3.1 Erster Ansatz Fachwerkhäuser 33](#_Toc148908607)

[4.3.2 Modellieren und texturieren Einfaches Haus mit Blender 33](#_Toc148908608)

[4.3.3 Zweiter Ansatz: Modellieren eines Fachwerkhaus mit Blender 33](#_Toc148908609)

[4.3.4 Dritter Ansatz Dorfbaukasten 33](#_Toc148908610)

[4.4 Meilenstein: Nebenfiguren 33](#_Toc148908611)

[4.5 Meilenstein: Dialogsystem 33](#_Toc148908612)

[4.5.1 Ansatz 1 mit ChatGPT 33](#_Toc148908613)

[4.5.2 Ansatz 2 Rechersche mit Suchmaschienen im Internet 33](#_Toc148908614)

[4.6 Meilenstein: Sprachausgabe 33](#_Toc148908615)

[5 Probleme in der Umsetzung 34](#_Toc148908616)

[5.1 Ablenkung und Abschweifung 34](#_Toc148908617)

[5.2 Einarbeitungszeit 34](#_Toc148908618)

[5.3 weis nicht alles 34](#_Toc148908619)

[5.4 Konsistenz 34](#_Toc148908620)

[5.5 Urheberrecht 34](#_Toc148908621)

[6 Fazit und Ausblick 35](#_Toc148908622)

[Abbildungsverzeichnis IV](#_Toc148908623)

[Tabellenverzeichnis V](#_Toc148908624)

[Abkürzungsverzeichnis VI](#_Toc148908625)

[Literaturverzeichnis VII](#_Toc148908626)

[Code-Beispiele IX](#_Toc148908627)

[Anhangsverzeichnis X](#_Toc148908628)

[Erklärung 11](#_Toc148908629)

# Einleitung

„Sprechen Sie einfach diese magischen Worte aus: Ich bin ein Game Designer. [...] Haben Sie es gemacht? Wenn ja, dann gratuliere ich Ihnen. Sie sind jetzt ein Game Designer“ (Schell, 2020, S. 37). Dieses Zitat stammt von Jesse Schell; Hochschullehrer für Unterhaltungstechnologie am Entertainment Technology Center in Pittsburgh, USA. Er ermutigt Anfänger in seinem Buch ”Die Kunst des Game Designs”, sich selbst bereits als Gamedesigner zu bezeichnen, auch wenn sie sich noch in den ersten Schritten hin zum Game Designer oder Videospiele Entwickler befinden. Wenn wir den Worten von Jesse Schell Glauben schenken, ist Gamedesigner werden nicht schwer, denn es fängt in erster Linie bei einem selbst an.

Um als professioneller Videospieler zu arbeiten oder einer zu werden zeigt Wang (2023, S. 251) drei Wege auf. Die erste Möglichkeit besteht darin, in einem großen Video-Spielentwicklungsstudio zu arbeiten, das in der Regel aus mehreren hundert Angestellten besteht. In solchen großen Studios ist es ¨üblich mit einem sehr kleinen Aufgabenbereich beschäftigt zu sein und quasi ein Spezialist in diesem Bereich zu werden. Der zweite Weg besteht darin, in einem kleinerem Entwicklerstudio anzufangen, in denen der Entwickler meistens mehrere Aufgabengebiete abdeckt. Der Vorteil bei diesen beiden Wegen ist es, dass ein Entwickler von anderen erfahreneren Entwicklern lernen kann. Der dritte und letzte Weg ist der Weg als Videospielentwickler, welcher im Alleingang oder in einem sehr kleinen Team Videospiele entwickelt. Hierbei ist der Entwickler gezwungen alle Aufgaben zu ¨übernehmen die anfallen, um ein Videospiel umzusetzen.

Um nochmal kurz Bezug auf Jesse Shell zu nehmen, macht es für mich sehr viel Sinn, sich selbst als Game Designer zu sehen und damit den Schritt zu wagen alleine ein eigenes Videospiel zu entwickeln.

Die neusten Entwicklungen in Bereich der Künstlichen Intelligenzen wie den Chatbot ChatGTP und Midjourney haben mich auf die Idee gebracht, ein Videospiel mit Hilfe verschiedener KI-Systeme zu entwickeln. Denn ein Videospiel als einzelne Person zu entwickeln erfordert eine Menge Fähigkeiten in ganz verschiedenen Bereichen wie Grafik Design, Programmierung oder Soundgestaltung (Wang, 2023, S. 251). Daher finde ich es spannend mich mit verschiedenen KI-Systemen zu beschäftigen und herauszuarbeiten, ob und in welchen Bereichen sie Entwickler von Videospielen unterstützen können.

## Motivation

Länger als ich denken kann, spiele ich Videospiele. Das Nintendo Entertainment System gehörte zu meiner Welt als Kind wie mein zu Hause und die Natur draußen. Obwohl wir nie viel Geld hatten, hatten wir doch an dem kleinen Röhrenfernsehr im Wohnzimmer diesen Wunderkasten in dem ich in Super Mario die Prinzessin retten musste, Enten mit dem NES Zapper in Duck Hunt jagte, und in Teenage Mutant Ninja Turtels 2 jeden Sieg mit dem Schlachtruf Cowabunga feierte.

Während ich als Kind nie das letzte Level in Teenage Mutant Ninja Turtel geschafft habe, war ich so frustriert das ich gesagt habe, dass das Spiel niemand schaffen kann. Mein Bruder Patrick hat mich in diesem Moment getröstet und gesagt: ”Doch, es gibt einen der das Spiel schafft, der Entwickler!” Das war der Moment, wo ich verstand, dass hinter jedem Werk auch ein Entwickler stand, und so ein Mensch wollte ich immer werden. Doch ich bin auf einem Dorf mit 300 Einwohnern aufgewachsen und hatte nie die Chance Medienkompetenzen zu erlangen, mit denen ich mir eine Perspektive aufbauen konnte, um Videospielentwickler zu werden.

Während meines Studiums an der Hochschule Fulda beschäftige ich mich immer mehr mit dem Entwickeln von Videospielen. Obwohl ich ein paar Gruppenarbeiten im Rahmen meines Studiums gemeistert habe, bemerkte ich, dass das Thema Videospiele zu entwickeln bei meinem Kommilitonen nie das große Thema war. Also schlug ich den weg als Ein-Mann-Videospielentwickler ein.

Auf meinem Weg als Ein-Mann-Videospielentwickler habe ich bemerkt, dass ich gute Programmierkenntnisse besitze, und auch mit Klängen gut umgehen kann, aber wenn ich ein Stift in die Hand nehme, um was zu schreiben oder zu malen, habe ich immer gemerkt, dass andere immer schneller und besser sind.

Dieser Gedanke, verlor ich im Oktober und Dezember 2022, denn seit dieser Zeit experimentiere ich mit KI-Systemen, die mich dabei unterstützen in diesen Disziplinen kreativ tätig zu sein.

Videospielentwicklung und KI-Systeme sind zwei Welten. Und diese zwei Welten möchte ich mit einer Brücke verbinden.

Im Dezember 2022 Veröffentlicht die Firma OpenAI ChatGPT und das Thema KI und KI-Systeme wird über Nacht ein Thema für die breite Gesellschaft (Riecken, 2023).

## Forschungsfrage und Forschungsmethode

In dieser Bachelorarbeit möchte ich mich mit der Frage auseinandersetzen, welche Möglichkeiten KI-Systeme einem Ein-Mann-Videospieler bei der Entwicklung eines Videospiels bieten. Um diese Frage beantworten zu können werde ich einen Prototyp entwickeln und so viele KI-Systeme einbinden, wie es möglich und sinnvoll ist.

## Gliederung der Arbeit

In Kapitel 2 Theoretischer Hintergrund definiert die benötigten Begriffe, um die vorliegende Bachelorarbeit nachzuvollziehen.

In Kapitel 3 Methodik, werden alle KI-Systeme und zusätzlich verwendete Software vorgestellt, die während des praktischen Teils der Bachelorarbeit verwendet werden.

In Kapitel 4 Umsetzung wird die Entwicklung eines Videospiels mit Hilfe von KI-Systemen anhand eines Prototyps in der Unreal Engine 5 gezeigt.

In Kapitel 5 Allgemeine Probleme mit KI und KI-Systeme werden auf Probleme hingewiesen, die während der Entwicklung des Prototyps entstanden sind.

In Kapitel 6 Fazit und Ausblick wird diskutiert, inwieweit die Forschungsfragen beantwortet werden, kann und gibt zusätzlich einen Ausblick.

# Theoretischer Hintergrund

## Begriffsdefinitionen

### Intelligenz

In der Psychologie ist das am besten erforschte Merkmal die Intelligenz. Nicht nur die Psychologie beschäftigt sich mit der Intelligenz, sondern auch Nachbardisziplinen wie Pädagogik, Sozialwissenschaft und Hirnforschung.

Der Begriff Intelligenz, geistige Fähigkeit, mentale Fähigkeit oder Begabung genannt, werden durch Faktoren wie Gene und Umwelt gegenseitig in der kognitiven Leistungsfähigkeit beeinflusst.

Intelligenz ist kein Privileg der Menschen, sondern auch in der Tierwelt zu beobachten (Rost, 2013, S. 11).

### Künstliche Intelligenz

Künstliche Intelligenz ermöglicht Computern, Aufgaben zu lösen, die sonst die Intelligenz vom Menschen benötigen würde. Zwei grundlegende Aspekte werden von dem Begriff Künstliche Intelligenz umfasst.

* Die Schaffung und Nachahmung menschlichen Verhaltens.
* Die autonome und automatische Aufgabenerledigung, von klar abgegrenzten Aufgabenbereichen.

Autonomes Fahren, intelligente Softwareassistenten sowie Sprach- und Bilderkennung sind heute sehr zuverlässige funktionierende Systeme, die immer mehr Relevanz in Unternehmen und im Alltag haben.

Fliesen neue Informationen zum inneren Modell einer KI, kann die KI selbständig lernen, was zu einer Anpassung der Anforderungen und deren Umweltbedingungen führt (Moring, 2023, S. 8).

### KI-Systeme

Unter dem Begriff “KI-Systeme” werden solche Maschinen verstanden, die in der Lage sind, eigenständig abstrakt beschriebene Aufgaben zu bearbeiten, welche nicht Schritt für Schritt vom Menschen programmiert wurden. Die KI-Systeme basieren auf maschinelles Lernen, wodurch die Systeme die Fähigkeit gewinnen, weiter zulernen und vorab trainierte Modelle zu verbessern (ifaa, 2023) KI-Systeme finden mittlerweile in den verschiedensten Branchen Anwendung wie der Automobilindustrie für das autonome Fahren, in der Logistik, Medizin oder in der Landwirtschaft. Im privaten Bereich sind KI-Systeme in Apps, auf dem Smartphone oder für automatisch generierte Sprach- und Bilderkennung im Einsatz (Moring, 2023, S1, S. 8)

### Prompt

Das englische Wort ”prompt” leitet sich gemäß des Cambridge Dictionary (o. D.) von dem Verb "to prompt“ ab und bedeutet ¨übersetzt sinngemäß, jemanden dazu zu bringen, etwas zu sagen oder zu tun. In Bezug auf Computer bedeutet das Nomen ”prompt”, einem KI-System eine Anweisung zu geben, welche nicht in Computersprache verfasst wird, sondern in der natürlichen Sprache des Menschen (Cambridge Dictionary, o. D.) Ein Prompt kann demnach als eine Eingabeaufforderung verstanden werden, die in Form von beispielsweise Fragen, Aufforderungen oder auch Beschreibungen eines Themas dem KI-System übermittelt werden. Das KI-System generiert auf der Basis der Eingabeaufforderung eine entsprechende Antwort (BM-Experts, 2023).

### NPC

Non-Player Character, kurz NPC, sind in Computerspielen zu finden und spielen eine wichtige Rolle, um eine Spielwelt lebendig zu gestalten [Hack, 2018, S. 293].

NPCs sind vom Computer gesteuerte Charaktere wie Dorfbewohner, Tiere oder Monster. Alle Charaktere und Tiere, die sich nicht vom anderen Spieler kontrollieren lassen (Breuer, 2012, S.113).

### Ein-Mann-Videospielentwickler

Bereits in der Einleitung wurde kurz beschrieben, dass es verschiedene Wege gibt um als Videospieleentwickler zu arbeiten. Hierbei wurden die drei folgenden Wege aufgezeigt: Das Arbeiten in einem großen Videospielstudio mit sehr vielen Angestellten, das Arbeiten in einem kleineren Entwicklerstudio und der dritte Weg ist es, ein Videospiel allein oder in einem sehr kleinen Team zu entwickeln. Wang (2023, S.251) bezeichnet diesen Entwickler als solo game developer. In dieser Bachelorarbeit verwende ich den Begriff ¨Ein-Mann-Videospielentwickler“ um zu verdeutlichen, dass alle Prozesse bei der Entwicklung des Videospiels von nur einer einzigen Person durchgeführt werden.

### Mesh - Vertices, Edges und Faces

|  |  |  |  |
| --- | --- | --- | --- |
| Ein Bild, das Screenshot, Box, Design, Behälter enthält.  Automatisch generierte Beschreibung | Ein Bild, das Screenshot, Design enthält.  Automatisch generierte Beschreibung | Ein Bild, das Screenshot, Design enthält.  Automatisch generierte Beschreibung | Ein Bild, das Box, Behälter enthält.  Automatisch generierte Beschreibung |
| Vertex | Edge | Face | Mesh |

Vertex Singular oder Vertices Plural, ist ein Begriff auf der 3D-Computergraphik. Ein Vertex repräsentiert einen einzelnen Punk im Raum und bringt die Grundlage für eine Darstellung von Objekten im dreidimensionalen Raum. Durch das Verbinden von mindestens drei Vertices, ist das Bilden eines Polygons (Face) möglich [Pyka, o. D.].

Die Verbindung zweier Vertices ist eine Edge. Ein Mesh besteht aus den Elementen, Vertices, Eges und Faces. Beispielsweise besteht das Mesh eines Würfels aus acht Vertices, zwölf Edges und sechs Faces (Bühler, 2021, S. 7).

### Textur

|  |  |
| --- | --- |
|  |  |
| a\_pale\_brown\_oak\_texture |  |

Bilder, die auf der Oberfläche von 3D-Modellen dargestellt werden, werden Texturen genannt (Bühler, 2021, S.7).

Innerhalb dieser Bachelorarbeit werden Image Texturen verwendet. Diese Image Texturen können von Midjourney erzeugt werden. Zum Beispiel ein generiertes Bild von einer Holzoberfläche. Diese Holzoberfläche wird einem 3D-Modell übergeben um wie ein Holzobjekt zu wirken.

Dreidimensionale Objekte können mit Hilfe von Texturen verändert werden. Diese Veränderungen können zum Beispiel sich in Farbe, Spiegelung, 3D-Eindruck auf der Oberfläche und Transparenz von 3D-Objekten sein. Eine Textur wird in normalen Fall durch ein zweidimensionales Bild repräsentiert. (Quelle fehlt – Adobe?-in Literaturverzeichnis vorhanden)

### T-Pose

Die T-Pose ist eine Körperhaltung in der Videospielcharakter-Entwicklung. um unerwünschte Verzerrungen in der Animation einer Spielfigur zu vermeiden. Sie zeichnet sich darin aus, dass Videospielcharaktere mit beiden Armen von dem Oberkörper gerade ausgestreckt sind. Die Körperhaltung erinnert an den Buchstabe T. Die T-Pose ermöglicht es, dass Videospielfiguren leichter geriggt werden können (Bühling, 2017).

# Methodik

Die zur Umsetzung des Prototyps verwendeten Werkzeuge werden in diesem Kapitel vorgestellt. Die verwendeten Werkzeuge werden nach KI-Systemen und Nicht-KI-Systeme kategorisiert. Innerhalb dieser Kategorien werden die Werkzeuge in der Reihenfolge beschrieben, wie sie in der Umsetzung des Prototyps angewendet worden sind.

## Auswahl und Beschreibung der KI-Systeme

### ChatGPT

Im Juni 2018 wurde das Sprachmodell GPT-1 vom US-Amerikanischen Unternehmen Open AI vorgestellt. ChatGPT ist ein Dialogsystem, welches als Schnittstelle zu dem Sprachmodell GPT-3,5 dient. GPT-3,5 kann mit Hilfe dieser Schnittstelle erstmal mit Usern frei formulierte Anfragen angemessen bearbeiten.

ChatGPT ist in dem Sinne beeindruckend, da es für viele Menschen die Möglichkeit bietet, mit einem Gesprächsagenten zu interagieren (Helfrich-Schkarbanenko, 2023, S. 3). Das Dialogsystem ChatGPT besitzt die Fähigkeit Kurzgeschichten zu schreiben, Fachfragen zu beantworten, Codeblöcke auszugeben oder Texte zusammenzufassen (Nimsdorf, 2023).

In dieser Bachelorarbeit wird ChatGPT beispielsweise dazu verwendet eine Spieleidee zu entwickeln oder Spielfiguren zu konzeptionieren. Des Weiteren wird ChatGPT als Schnittstelle zwischen User und dem KI-System Midjourney eingesetzt. Für die Umsetzung des Videospielprototypen wird das kostenlose Sprachmodell GPT-3,5 verwendet. Der Zugriff auf das Sprachmodell GPT-4 ist über ein Bezahlmodell (ChatGPTPlus) möglich, welches ich aber nicht verwendet habe (OpenAI, 2023).

### Midjourney

Midjourney wurde von David Holz, Midjourney kann aus Text mit Hilfe einer künstlichen

Intelligenz Bilder erzeugen

### PIFuHD

3D Modell von Spielfiguren zu erzeugen

### Voice.AI

Wurde verwendet um Stimmen zu verstellen

### Adobe Enhanced Speech

Verwendet um die Qualit¨at von Sprachaufnahmen zu verbessern

## Beschreibung der Tools und Technologien

### Unreal Engine 5

Die Unreal Engine ermöglicht den Spieleentwickler 3D-Videospiele zu entwickeln. Die

Entwicklung eines Videospiels in der Unreal Engine 5 kann in Echtzeit entwickelt werden, das bedeutet, dass man das Ergebnis seiner Arbeit sofort betrachten kann. Epic Games,

die Entwickler der Unreal Engine 5, beschreiben sie als ”Das weltweit offenste und

fortschrittlichste Tool zur 3D-Erstellung in Echtzeit

### Blender

3D Modelle zu modellieren und texturieren

### Game Rig Tools (Blender Addon)

### Materialize

### Audacity

Verwendet um Stimmen aufzunehmen

# Umsetzung

In diesem Kapitel wird gezeigt, wie mit Hilfe von KI-Systemen der Prototyp in der Unreal Engine 5 umgesetzt wird. Um die Kommunikation mit den KI-Systemen zu visualisieren, erhält jedes KI-System ein eigenes Symbol oder einen Schriftzug. Die Zuordnung der Symbole und Schriftzüge zu dem jeweiligen KI-System sind in Tabelle 1 dargestellt.

Tabelle 1: Symbolübersicht

|  |  |  |  |
| --- | --- | --- | --- |
| Ein-Mann-Videospielentwickler | ChatGPT | Midjourney | PIFuHD |
| Abbildung : Symbol: Ein-Mann-Videospielentwickler  Ein Bild, das Cartoon enthält.  Automatisch generierte Beschreibung  Quelle: Titán Televisor MAX (Pinterest User), o.D | Abbildung : Symbol: ChatGPT  Ein Bild, das Symbol, Grafiken, Schrift, Logo enthält.  Automatisch generierte Beschreibung  Quelle: Zhing'za zï Ïnin (Wikipedia User), OpenAI (Original), 04. September 2023 | Abbildung : Symbol: Midjourney  Ein Bild, das Entwurf, Zeichnung, Strichzeichnung, Clipart enthält.  Automatisch generierte Beschreibung  Quelle: Midjourney, 04. Juli 2023 | Abbildung : Schriftzug:PIFuHD  PIFuHD  Quelle: Eigener Schiftzug |

Quelle: Eigene Darstellung

Alle Prompts für ChatGPT und Midjourney sind in ihrere Form unverändert in dieser Bachelorarbeit abgebildet. Die Prompts beinhalten Rechtschreibfehler und Umgangssprache. Die Wahrscheinlichkeit liegt sehr hoch, dass auf Verzicht auf Rechtschreibfehler und Umgangssprache andere Ergebnisse hervorgerufen hätte.

## Entwicklung der Spieleidee mit Hilfe von ChatGPT

Eine Schlüsselressource der Zukunft ist Kreativität. Kreativität kann mit Kreativitätstechniken gefördert werden, wie zum Beispiel mit Mindmapping, Brainstorming oder der Walt Disney Methode. Ziel dieser Kreativtechniken ist es, neue Ideen zu generieren (Mai, 2019).

In diesem Experiment wird eine weitere Kreativtechnik verwendet, um die Spielidee zu entwickeln - das Nutzen eines KI-Systems. Um genau zu werden, wird ChatGPT dazu verwendet, ein Thema für Videospiele zu finden.

Hierfür wird ChatGPT aufgefordert fünf große Themen in der deutschen Geschichte zu nennen, welche vor dem zweiten Weltkrieg stattgefunden haben. Der exakte Prompt sowie das resultierende Ergebnis sind in Tabelle 2 zu sehen.

Tabelle : Die ersten fünf Themenvorschläge von ChatGPT

|  |  |
| --- | --- |
| Ein Bild, das Cartoon enthält.  Automatisch generierte Beschreibung | nenn mir bitte 5 große Themen in der Deutschen Geschichte, vor den zweiten Weltkrieg. |
| Ein Bild, das Symbol, Grafiken, Schrift, Logo enthält.  Automatisch generierte Beschreibung | Gerne! Hier sind fünf große Themen in der deutschen Geschichte vor dem Zweiten Weltkrieg:  1. Das Heilige Römische Reich Deutscher Nation: Das Reich bestand vom 10. Jahrhundert bis 1806 und war ein wichtiger Einfluss auf die politische und kulturelle Entwicklung Deutschlands.  2. Die Reformation: Die Bewegung begann im 16. Jahrhundert und führte zu bedeutenden Veränderungen in der religiösen Praxis und Theologie, insbesondere durch Martin Luther.  3. Die Aufklärung: Eine intellektuelle Bewegung im 18. Jahrhundert, die sich für Freiheit, Toleranz und Vernunft einsetzte und viele wichtige Persönlichkeiten hervorbrachte, darunter Immanuel Kant.  4. Die Industrialisierung: Im 19. Jahrhundert wurde Deutschland zu einer führenden Industrienation und erlebte eine rasante wirtschaftliche und technologische Entwicklung.  5. Der Erste Weltkrieg: Der Konflikt von 1914-1918 hatte einen enormen Einfluss auf die deutsche Gesellschaft und führte zu politischen, wirtschaftlichen und kulturellen Veränderungen, die die folgenden Jahrzehnte prägten. |

Quelle: Eigene Darstellung, in Anlehnung an ChatGPT

Da die ersten fünf Ideen von ChatGPT nicht zufriedenstellend sind, wird ChatGPT erneut aufgefordert, fünf weitere Themen auszugeben. Der Prompt und die weiteren fünf Themen, die von ChatGPT vorgeschlagen werden, sind in Tabelle 3 zu sehen.

Tabelle : Die zweiten fünf Themenvorschläge von ChatGPT

|  |  |
| --- | --- |
| Ein Bild, das Cartoon enthält.  Automatisch generierte Beschreibung | nen mir bitte weitere 5 |
| Ein Bild, das Symbol, Grafiken, Schrift, Logo enthält.  Automatisch generierte Beschreibung | Gerne! Hier sind weitere fünf große Themen in der deutschen Geschichte vor dem Zweiten Weltkrieg:  1. Die Französische Revolution und Napoleonische Kriege: Die Ereignisse von 1789 bis 1815 beeinflussten auch Deutschland stark und führten zu politischen Veränderungen, insbesondere durch die napoleonische Herrschaft.  2. Die Deutsche Revolution von 1848: Eine Bewegung für demokratische Reformen und Einheit, die jedoch letztendlich scheiterte.  3. Die Bismarcksche Reichsgründung: Die Schaffung des Deutschen Kaiserreichs 1871 durch Kanzler Otto von Bismarck prägte die deutsche Geschichte des 19. Jahrhunderts.  4. Der Erste Weltkrieg und die Weimarer Republik: Die Niederlage im Ersten Weltkrieg und die darauf folgende Gründung der Weimarer Republik waren entscheidend für die Entwicklungen in der deutschen Geschichte.  5. Die nationalsozialistische Machtergreifung: Der Aufstieg der NSDAP und Hitlers Ernennung zum Reichskanzler im Januar 1933 führte schließlich zum Zweiten Weltkrieg und den Schrecken des Holocausts. |

Quelle: Eigene Darstellung, in Anlehnung an ChatGPT

Der Ausgabe nach zu urteilen, besitzt ChatGPT eine gewisse Datenmenge über die deutsche Geschichte. Es wurde an dieser Stelle nicht recherchiert, ob die vorgeschlagenen Themen wirklich in der deutschen Geschichte vorgekommen sind, sondern dienen nur der Ideenfindung.

Als Ein-Mann-Videospielentwickler ist es nun möglich, aus diesen zehn vorgeschlagenen Themen ein Videospiel zu entwickeln. Die Entscheidung fällt auf das zweite Thema der ersten Ausgabe von ChatGPt: Die Reformation, mit Martin Luther als spielbare Hauptfigur.

Innerhalb dieser Bachelorarbeit ist es nicht möglich ein komplettes Videospiel zu entwickeln, was das Leben von Martin Luther widerspiegelt, deshalb liegt die Konzentration auf die Entwicklung eines Prototyps.

Martin Luther war damals überzeugt davon, dass die Menschen durch den richtigen Glauben ihren Sünden verziehen werden, und nicht durch Ablassbriefe der Kirche. Genau diesen Zeitpunkt, bevor Martin Luther angeblich seine 95 Thesen an die Kirchentür in Wittenberg nagelt, wird als Vorlage der Videospielprototyp verwendet [Mr-Wissen2go Geschichte, 2017].

Hierbei soll die Spielwelt des Prototyps aus einem Dorf mit verschiedenen Häusern, Martin Luther als Spielbare Hauptfigur und NPCs bestehen. Die Spielidee für den Prototyp ist, dass Martin Luther durch ein Dorf läuft, verschiedene NPCs trifft und mit diesen in einen Dialog tritt. Diese verschiedenen Personen haben verschiedene Ansichten und Probleme. Durch das Sprechen mit diesen Personen entwickelt Martin Luther seine Inspirationen für seine 95 Thesen.

Die Entwicklung des Prototyps lässt sich in folgende Meilensteine unterteilen:

• Hauptfigur

• Gebäude

• Nebenfiguren

• Dialogsystem

• Sprachausgabe

Jeder oben genannte Meilenstein besitzt in dieser Bachelorarbeit ein eigenes Unterkapitel, in dem die Entwicklung des jeweiligen Meilensteins gezeigt wird.

## Meilenstein: Hauptfigur



Abbildung : Hautfigur Martin Luther in der Unreal Engine 5,

Quelle: Eigene Darstellung

Abbildung 5 zeigt die Hauptfigur Martin Luther. Martin Luther wurde mit Hilfe der KI-Systeme ChatGPT, Midjourney und PIFuHD entwickelt. Weitere Technologien, die verwendet wurden für die Entwicklung der Hauptfigur, sind Blender und Google-Collab.

Tabelle : Martin Luther in Action

|  |  |  |
| --- | --- | --- |
|  |  |  |

Martin Luther kann in der Unreal Engine 5 laufen, springen, sich drehen und mit Gegenständen kollidieren. Wie die Hauptfigur entsteht und diese Eigenschaften bekommt wird in diesem Abschnitt erklärt.

Martin Luther kann zusätzlich Türen auf und zu machen und sich mit NPC unterhalten. Diese Fähigkeiten werden im Abschnitt; Meilenstein Gebäude und im Abschnitt Dialogsystem erklärt.

**Abbildung 4.3: UE5: Martin Luther im Spiel**

Das Ziel ist, ChatGPT dazu zu bringen, einen Prompt für Midjourney zu generieren, welche Bilder für PIFuHD erzeugt. Die Entwicklung der Hauptfigur beginnt in dem ChatGPT um eine Beschreibung der Hauptfigur Martin Luther aufgefordert wird. Außerdem wird in dem Prompt angegeben, dass die Ausgabe von ChatGPT als Prompt fur Midjourney verwendet wird. Zusätzlich sollen die von Midjourney generierten Bilder PIFuHD kompatibel sein.

Tabelle : ChatGPT: Erster Versuch zur Erstellung eines Prompts für Midjourney

|  |  |
| --- | --- |
| Ein Bild, das Cartoon enthält.  Automatisch generierte Beschreibung | Bitte schreib mir ein prompt für midjourney um eine vorlage zu bekommen die ich in PIFuHD benutzen kann. Die Person soll martin luther ähnlich sehen. |
| Ein Bild, das Symbol, Grafiken, Schrift, Logo enthält.  Automatisch generierte Beschreibung | "Martin Luther bei der Arbeit an seiner Bibelübersetzung"  Beschreibe eine Szene, in der Martin Luther bei der Arbeit an seiner Bibelübersetzung ist. Er sitzt an einem Tisch mit Büchern und Schreibwerkzeugen um sich herum. Die Atmosphäre kann entweder ruhig und friedlich oder hektisch und angespannt sein, je nachdem, welche Phase der Übersetzung er gerade durchläuft. Stelle sicher, dass das Gesicht des Martin Luther ähnlich aussieht wie das historische Abbild des berühmten Reformators und dass er typische Kleidung dieser Zeit trägt. |

Quelle: Eigene Darstellung, in Anlehnung an ChatGPT

Der erste Promt für ChatGPT dient dazu herauszufinden, ob ChatGPT Prompts für Midjourney erstellen kann. Das Ergebnis des ersten Versuchs ist in Tabelle 5 zu sehen.

Dieser Prompt hat ein Ergebnis auf Deutsch zurückgegeben, was für Midjourney nicht zu empfehlen ist. Es ist möglich, Prompts auf deutsch für Midjourney zu verfassen, dafür ist die Wahrscheinlichkeit höher, eine falsche Interpretation zu bekommen. Denn Englisch ist Midjourneys Hauptsprache[luber-2023].

Durch einen Prompt zur Übersetzung, wird der vorige Prompt von ChatGPT ins Englische übersetzt. Den Prompt und das Ergebnis der Übersetung ist in Tabelle 6 zu betrachten.

Tabelle : ChatGPT: Übersetzung des vorherigen Promps von ChatGPT

|  |  |
| --- | --- |
| Ein Bild, das Cartoon enthält.  Automatisch generierte Beschreibung | auf englisch bitte |
| Ein Bild, das Symbol, Grafiken, Schrift, Logo enthält.  Automatisch generierte Beschreibung | "Martin Luther working on his Bible translation"  Describe a scene where Martin Luther is working on his Bible translation. He is sitting at a table with books and writing tools around him. The atmosphere can be either quiet and peaceful or hectic and tense, depending on which phase of the translation he is currently going through. Make sure that Martin Luther's face looks similar to the historical depiction of the famous reformer and that he is wearing typical clothing of that time. |

Quelle: eigene Darstellung, in Anlehnung an ChatGPT

Dieser ins Englische übersetzte Prompt wurde in Midjourney mit Hilfe von Kopieren und Einfügen übergeben. Midjourney hat mit diesem Prompt folgendes Bild generiert.

Abbildung : Midjourney: Erster Versuch Martin Luther Prompt von ChatGPT

Ein Bild, das Text, Menschliches Gesicht, Buch, Kleidung enthält.

Automatisch generierte Beschreibung

Quelle: Midjourney

Das Ergebnis, von dem ins Englische übersetzte Prompt zeigt Abbildung 6. In Abbildung 6 ist eine Figur, die Martin Luther ähnlich ist, zumindest einem Mönch an einem Schreibtisch ähnlichsieht. Mit diesem Versuch kann erkenne werden, das Midjourney Prompts solcher Art umgehen kann und richtig Interpretiert, aber dieses Ergebnis ist nicht für PIFuHD zu verwenden.

Ziel ist es eine Vorlage für PIFuHD zu bekommen. PIFuHD benötigt Bilder die gewisse Vorrausetzung erfüllen.

Folgende Tips für die Bilder werden von PIFuHD empfohlen:

* Hochauflösende bilder mit 1024 x 1204 Pixel
* Bilder mit einer Person
* Eine Direkte von vorne gerchtete Kameraeinstellung
* Der ganze Körper ist abgelichtet
* Einge gute Beleuchtung
* Paraleler Kamerawinkel zum Boden
* Ein einfacher Hintergrund
* Nur Menschen und keine Künstlichen Bilder von Personen wie Anime-Figuren

Hochskalierte Bilder von Midjourney werden in der Auflösung 1024 x 1024 erzeugt, was schon mal die erste Voraussetzung erfüllt. Die Letzte Voraussetzung kann nicht erfüllt werden, da alle Bilder von Personen von Midjourney künstlich sind. Es wird trotzdem experimentiert 3D-Modelle zu erzeugen mit Hilfe von PIFuHD.

Eine Recherche auf Youtube zeigt, dass das Verwenden von einer Midjourney-Formel klare Ergebnisse hervorbringen kann (Wade McMaster - Creator Impact, 2023).

Die Idee, das ChatGPT Midjourney-Promts generiert, die PIFuHD-Kompatible bilder generiert wird verworfen. Der neue Ansatz ist ein schritt zusätzlich zu machen, und ChatGPT beizubringen, PIFuHD-Kompatible Midjourney-Promts zu generieren.

Mit dem folgenden Prompt wird ChatGPT dazu aufgefordert, die Midjourney-Formel zu benutzen.

Tabelle : ChatGPT: aufforderung zur benutzung der Midjourney-Promt-Formel

|  |  |
| --- | --- |
| Ein Bild, das Cartoon enthält.  Automatisch generierte Beschreibung | Here is a MidJourney Prompt Formula:  (image we're prompting), (5 descriptive keywords), (camera type), (camera lens type), (time of day), (style of photograph), (type of film)  Please respond with "yes" if you understand the formula |
| Ein Bild, das Symbol, Grafiken, Schrift, Logo enthält.  Automatisch generierte Beschreibung | Yes, I understand the formula. |

Quelle: eigene Darstellung, in Anlehnung an ChatGPT

In der Tabelle 7 hat ChatGPThat bestätigt, das er die Midjourney-Formel versteht. Mit dem nächsten Prompt wird geprüft, ob ChatGPT die Midjourney-Formel richtig anwendet und für Midjourney besser strukturiertere Prompts generiert. Getestet wird das mit einer Aufforderung einen Abendteurer mit Fuchskopf.

Tabelle : ChatGPT: Fünf Midjourney-Prompts - Videospielcharakter mit Fuchskopf

|  |  |
| --- | --- |
| Ein Bild, das Cartoon enthält.  Automatisch generierte Beschreibung | Bitte erstelle mir 5 MidJourney Promps für ein Videospielcharakter. Der Videospielcharakter ist ein Mensch mit Fuchskopf. |
| Ein Bild, das Symbol, Grafiken, Schrift, Logo enthält.  Automatisch generierte Beschreibung | Hier sind fünf MidJourney Prompts für deinen Videospielcharakter:  1. A humanoid fox warrior standing atop a cliff, overlooking a sprawling forest below. The warrior is wielding a sword and wearing armor made of leather and metal. (warrior, fox, cliff, aerial camera, dawn, dramatic, black and white film)  2. A fox-headed person sitting on a tree branch, overlooking a river. They are dressed in simple clothing and holding a quill and parchment. (scholar, fox, tree, close-up camera, afternoon, tranquil, sepia film)  3. A fox-headed adventurer emerging from a cave, holding a torch in one hand and a map in the other. (adventurer, fox, cave, wide-angle camera, night, adventurous, high-contrast film)  4. A fox-headed farmer tending to their crops in a peaceful meadow, surrounded by colorful wildflowers. (farmer, fox, meadow, medium shot camera, morning, serene, color film)  5. A fox-headed thief sneaking through a medieval city at night, using the shadows to remain undetected. (thief, fox, city, low-angle camera, night, mysterious, black and white film) |

Quelle: eigene Darstellung, in Anlehnung an ChatGPT

Tabelle 8 zeigt den ersten Versuch eine Ausgabe von ChatGPT mit der Midjourney-Formel wurde ein Ergebnis generiert, welches mit Klammern ausgegeben wird. Mit einem einfachen Prompt werden diese Klammern für zukünftige Prompts vermieden.

|  |  |
| --- | --- |
| Ein Bild, das Cartoon enthält.  Automatisch generierte Beschreibung | Bitte nochmal, aber ohne die Klammern. |
| Ein Bild, das Symbol, Grafiken, Schrift, Logo enthält.  Automatisch generierte Beschreibung | Gerne, hier sind die 5 MidJourney Prompts für deinen Videospielcharakter, einen Mensch mit Fuchskopf:  1. A humanoid fox warrior standing atop a cliff, overlooking a sprawling forest below. The warrior is wielding a sword and wearing armor made of leather and metal. Keywords: warrior, fox, cliff, aerial camera, dawn, dramatic, black and white film.  2. A fox-headed person sitting on a tree branch, overlooking a river. They are dressed in simple clothing and holding a quill and parchment. Keywords: scholar, fox, tree, close-up camera, afternoon, tranquil, sepia film.  3. A fox-headed adventurer emerging from a cave, holding a torch in one hand and a map in the other. Keywords: adventurer, fox, cave, wide-angle camera, night, adventurous, high-contrast film.  4. A fox-headed farmer tending to their crops in a peaceful meadow, surrounded by colorful wildflowers. Keywords: farmer, fox, meadow, medium shot camera, morning, serene, color film.  5. A fox-headed thief sneaking through a medieval city at night, using the shadows to remain undetected. Keywords: thief, fox, city, low-angle camera, night, mysterious, black and white film. |

In Tabelle XXX ist zu sehen, das ChatGPT die Klammern aus den Midjourney-Promts von entfernt hat. Die von ChatGPT erzeugten Prompts werden in Midjourney getestet.

Ein Bild, das PC-Spiel, Natur, Höhle enthält.

Automatisch generierte Beschreibung

Das Ergebnis aus dem Prompt aus Tabelle XXX ist in Abbildung XX zu betrachten. Der Test mit einem Videospielcharakter mit Fuchskopf, lässt noch keine Schlüsse zu, zu sagen, ob das Ergebnis, besser ist als die Ausgabe von Midjourney ind Abbildung XXX davor. Ein weiter Test mit Martin Luther muss durchgeführt werden.

Für einen weiteren Versuch wird ChatGPT aufgefordert, Prompts für Midjourney zu erstellen, mit Hilfe der Midjourney-Formel, diesmal mit Martin Luther als Videospielcharakter.

|  |  |
| --- | --- |
| Ein Bild, das Cartoon enthält.  Automatisch generierte Beschreibung | Bitte erstelle mir 5 MidJourney Promps für ein Videospielcharakter. Der Videospielcharakter ist Martin Luther. |
| Ein Bild, das Symbol, Grafiken, Schrift, Logo enthält.  Automatisch generierte Beschreibung | Gerne, hier sind fünf MidJourney Prompts für deinen Videospielcharakter Martin Luther:  1. Martin Luther preaching to a crowd of people in a church, with light streaming in from stained glass windows. Keywords: preacher, church, crowd, medium shot camera, afternoon, dramatic, color film.  2. Martin Luther writing at a desk, surrounded by books and religious texts. Keywords: scholar, desk, books, close-up camera, morning, serious, sepia film.  3. Martin Luther walking through a forest, holding a Bible in his hand and deep in thought. Keywords: pilgrim, forest, Bible, low-angle camera, dawn, introspective, black and white film.  4. Martin Luther debating with other theologians in a grand hall, with a crowd of people watching from the sidelines. Keywords: debater, hall, theologians, wide-angle camera, afternoon, intense, color film.  5. Martin Luther kneeling in prayer at an altar, surrounded by candles and religious icons. Keywords: devout, altar, prayer, close-up camera, night, spiritual, sepia film. |

Ein Ergebnis aus Tabelle XXX wird in Midjourney getestet.

Ein Bild, das Mann, Kleidung, Person, Menschen enthält.

Automatisch generierte Beschreibung

Abbildung XXX zeigt ein etwas farbarmes Ergebnis. Um etwas mehr Farbe in die von Midjourney erzeugten Bilder zu bekommen, wird ChatGPT dazu aufgefordert.

|  |  |
| --- | --- |
| Ein Bild, das Cartoon enthält.  Automatisch generierte Beschreibung | Bitte nochmal, aber die Bilder sollen etwas farbenfroh sein. |
| Ein Bild, das Symbol, Grafiken, Schrift, Logo enthält.  Automatisch generierte Beschreibung | Natürlich, hier sind fünf MidJourney Prompts für deinen Videospielcharakter Martin Luther, mit etwas farbenfroheren Bildern:  1. Martin Luther preaching to a crowd of people in a colorful church, with light streaming in from stained glass windows. Keywords: preacher, church, crowd, medium shot camera, afternoon, vibrant, color film.  2. Martin Luther writing at a desk in a bright study, surrounded by books and religious texts. Keywords: scholar, desk, books, close-up camera, morning, serious, vivid, color film.  3. Martin Luther walking through a forest full of colorful trees, holding a Bible in his hand and deep in thought. Keywords: pilgrim, forest, Bible, low-angle camera, dawn, introspective, colorful, color film.  4. Martin Luther debating with other theologians in a grand hall with colorful tapestries and ornate furnishings, with a crowd of people watching from the sidelines. Keywords: debater, hall, theologians, wide-angle camera, afternoon, intense, colorful, color film.  5. Martin Luther kneeling in prayer at an altar in a vibrant, colorful chapel, surrounded by candles and religious icons. Keywords: devout, altar, prayer, close-up camera, night, spiritual, colorful, color film. |

Ein Bild, das Kunst, Kleidung, Person enthält.

Automatisch generierte Beschreibung

Abbildung XXX zeigt ein sehr farbenfrohes Ergebnis. Tabelle XXX zeigt die Midjourney-Promts die von ChatGPT erzeugt wurden. Gegenüber Tabelle XXX fallen die Schlüsselworter colorful und color film auf.

Das nächste Ziel ist, das Martin Luther in einer T-Pose sich befindet. Der Folgende Prompt fordert ChatGPT dazu auf, Auskunft darüber zu geben, ob ChatGPT die T-Pose kennt. Des Weiteren sollen die Midjourney-Promts den ganzen Körper von Martin Luther zeigen.

|  |  |
| --- | --- |
| Ein Bild, das Cartoon enthält.  Automatisch generierte Beschreibung | Bitte nochmal, aber Martin Luther sein körper ist ganz zu sehen, und er befindet sich in der T-Pose. Kennst du die T-Pose in denen Videospiele kreiert werden? Wenn ja, bitte Antwort mit, "Ja, kenne ich". |
| Ein Bild, das Symbol, Grafiken, Schrift, Logo enthält.  Automatisch generierte Beschreibung | Ja, ich kenne die T-Pose in der Videospiel-Entwicklung. Hier sind fünf MidJourney Prompts für deinen Videospielcharakter Martin Luther, in denen er in einer T-Pose zu sehen ist:  1. Martin Luther standing in a T-Pose on the steps of a grand cathedral, with the sun setting behind him. Keywords: church, sunset, cathedral steps, wide-angle camera, dramatic, color film.  2. Martin Luther in a T-Pose standing in a lush garden, surrounded by colorful flowers and foliage. Keywords: nature, garden, flowers, medium shot camera, peaceful, color film.  3. Martin Luther in a T-Pose standing on the banks of a tranquil river, with mountains in the background. Keywords: nature, river, mountains, wide-angle camera, serene, color film.  4. Martin Luther in a T-Pose standing in a grand hall filled with ornate furnishings and tapestries. Keywords: hall, ornate, furnishings, medium shot camera, regal, color film.  5. Martin Luther in a T-Pose standing on a hill overlooking a medieval city, with the sun shining brightly. Keywords: city, hill, sun, wide-angle camera, triumphant, color film. |

Ein Bild, das Statue, Kunst, Kleidung, Kirche enthält.

Automatisch generierte Beschreibung

Das Ziel Bilder von Martin Luther in der T-Pose zu bekommen hat sich an dieser Stelle nicht erfüllt. Die Bilder zeigen eine Statur und keinen Menschen. Es ist zu vermuten, dass Hintergründe die Person im Vordergrund farblich beeinflusst. In diesem Fall ist eine Braun-Graues Kathedrale aus Stein im Hintergrund. Diese Braun-Graue Kathedrale beeinflusst wahrscheinlich die Person im Vordergrund, und verwandelt sie in eine Statur.

Um solche unerwünschten Einflüsse zu vermeiden, wird ChatGPT dazu aufgefordert noch einmal die Prompts zu generieren. Zusätzlich sollen die Midjourney-Promt mit neutralem Hintergrund und direkter Kameraeinstellung auf Martin Luther generiert werden.

|  |  |
| --- | --- |
| Ein Bild, das Cartoon enthält.  Automatisch generierte Beschreibung | Bitte nochmal, und beschreibe, das es für eine Vorlage eines Videospiel dienen soll. Der Hintergrund darf gerne neutral sein. Die Kamera direkt auf die Figur. |
| Ein Bild, das Symbol, Grafiken, Schrift, Logo enthält.  Automatisch generierte Beschreibung | Hier sind fünf MidJourney Prompts für Martin Luther, die als Vorlage für einen Videospiel-Charakter dienen sollen. Die Figur steht in der T-Pose, der Hintergrund ist neutral und die Kamera zeigt direkt auf die Figur:  1. Martin Luther in T-Pose, wearing his signature robes and holding a Bible in one hand, standing on a plain white background. Keywords: neutral background, T-Pose, robes, Bible, close-up camera, color film.  2. Martin Luther in T-Pose, holding a quill pen and standing in front of a plain gray background. Keywords: neutral background, T-Pose, quill pen, gray, medium shot camera, sepia film.  3. Martin Luther in T-Pose, holding a hammer and standing in front of a plain black background. Keywords: neutral background, T-Pose, hammer, black, close-up camera, black and white film.  4. Martin Luther in T-Pose, wearing his signature robes and holding a Bible in one hand, standing in front of a plain blue background. Keywords: neutral background, T-Pose, robes, Bible, medium shot camera, color film.  5. Martin Luther in T-Pose, wearing his signature robes and holding a quill pen, standing in front of a plain white background. Keywords: neutral background, T-Pose, robes, quill pen, close-up camera, sepia film. |

Ein Bild, das Person, Menschliches Gesicht, Kleidung, Mann enthält.

Automatisch generierte Beschreibung

Wir sehen einen Mönch der Martin Luther ähnlich sieht. Der Hintergrund ist neutral, und die Kameraeinstellung ist direkt. Es ist ein realitätsnahe Darstellung. Was nicht funktioniert hat, ist das die Person ganz zu sehen ist, sie befindet sich nicht in der T-pose.

Nach all den Versuchen, ChatGPT dazu zu bringen Prompts für Midjourney zu generieren, wird auf ChatGPT verzichtet. Der Prozess immerwieder zwischen ChatGPT und Midjourney zu wechseln kostet Zeit.

Für die entwicklung der Hauptfigur werden die Midjourney-Prompts selbst verändert. Das hat den vorteil, das die Zyklen zwischen den Midjourney-Prompts und deren anpassung kürzer ist.

Nach weitern Versuchen wurde mit folgenden Prompt dieses Ergebnis von Midjourney generiert.

|  |  |  |
| --- | --- | --- |
| Ein Bild, das Cartoon enthält.  Automatisch generierte Beschreibung | Ein Bild, das Entwurf, Zeichnung, Strichzeichnung, Clipart enthält.  Automatisch generierte Beschreibung | Ein Bild, das Entwurf, Zeichnung, Strichzeichnung, Clipart enthält.  Automatisch generierte Beschreibung Upscale V 4 |
| Martin Luther in T - Pose, wearing his signature robes, standing in front of a plain blue background. Keywords: neutral background, T - Pose, whole body, robes, Bible, color film. | Ein Bild, das Kleidung, Mantel, Cartoon, Kostümdesign enthält.  Automatisch generierte Beschreibung | Ein Bild, das Kleidung, Mantel, Person, Robe enthält.  Automatisch generierte Beschreibung |

Midjourney erzeugt mit jedem Prompt vier Vorschaubilder. Der Benutzer kann entscheiden, ob er auf Grundlage eines dieser vier Vorschaubilder, vier neue Bilder generieren lassen möchte, ein Bild hochskalieren lässt oder vier komplett neue Vorschaubilder auf Grundlage des verwendeten Prompts erzeugen lassen möchte.

In der Abbildung Tabelle XXX ist der verwendete Prompt, der selbst erstellt wird. Mit diesem Prompt erzeugt Midjourney vier Vorschaubilder. Sehr oft in unterschiedlichen Stilrichtungen. Anschließend wird eine hochskalierte Version von Version 4 erzeugt.

Mit diesem hochskalierten Bild wird ein erster Test mit PIFuHD durchgeführt, Das Bild erfüllt soweit alle Kriterien, bis auf das es keine echte Fotografie ist, sondern ein künstlich erzeugtes Bild durch ein KI-System. Das Testen und erzeugen eines 3D-Modells mit Hilfe von PIFuHD wird im nächsten Abschnitt erläutert.

### Erzeugen eines 3D-Modells mit Hilfe von PIFuHD

Im vorrigen Abschnit wurde versucht mit Hilfe von ChatGPT und Midjourney Bilder zu generieren. Diese Bilder sollen dazu dienen 3D-Modelle mit Hilfe von PIFuHD zu erzugen.

Der Prozess, des Erzeugen von 3D-Modelle wird in diesem Abschnitt gezeigt und erläutert.

Als erstes wird das Hochskalierte Bild aus Tabelle XXXStelle wird getestet, ob dieses Bild für PIFuHD-kompatiebel ist.

PIFuHD ist durch eine Google-Suche zu finden. Der Erste Vorschlag von Google zeigt einen Git-Hub-Link an. Im kopf dieser Git-Hub Seite findet man weit oben den Link zur Demo die auf Google-Colab veröffentlicht ist.

Über das Anmelden eines Google-Kontos und über das Verbinden mit der gehosteten Laufzeig wird Google-Colab die Demo von PIFuHD zum Laufen gebracht. Anschließend startet PIFuHD über den Menüpunkt Laufzeit → Alle ausfuhren oder alternativ CTRL + F9.

Weiter unten in der Website befindet sich der Abschnitt Config input data. An dieser Stelle wartet PIFuHD auf eine Eingabeaufforderung in Form eines Buttons Durchsuchen.., wo das Bild von Martin Luther übergeben werden kann.̈

Ist die Übergabe erfolgreich, erfolgt nach einer kurzen Bearbeitungszeit, die Ausgabe über die Ordnerstruktur links.

Das 3D-Objekt wird als obj-Datei gespeichert und befindet sich in der Ordnersturkur pifuhd → results → pifuhd-final → recon. Zusätlich zu der obj-Datei bietet PIFuHD eine png-Datei von einer Normalmap und ein fünf sekündiges Video an.

Das Ergebnis was PIFuHD erzeugt hat ist in folgenden Abbild zu sehen.

|  |  |
| --- | --- |
| Ein Bild, das Kleidung, Mantel, Person, Robe enthält.  Automatisch generierte Beschreibung | Ein Bild, das Statue, Kunst, Skulptur enthält.  Automatisch generierte Beschreibung |

Dieser Test zeigt, dass es möglich ist, mit Hilfe von verschiedenen KI-Systeme 3D-Figuren zu erzeugen. Nach diesem Ergebnis wird versucht ein noch besseres Ergebnis zu erzeugen. Im Bauchbereich und des Rechten Beins ist das Modell unvollständig. Möglicherweise sind diese Bereiche etwas unbelichtet.

Nach dem ersten Test mit PIFuHD ist für der nächste Test die T-Pose wichtig, denn die Hauptfigur soll geriggt werden und die T-Pose kann das Riggen später vereinfachen.

Nach weiteren Versuchen wurde ein Ergebnis erzeugt, welches für die Hauptfigur im Prototyp verwendet wird.

|  |  |  |
| --- | --- | --- |
| Ein Bild, das Cartoon enthält.  Automatisch generierte Beschreibung | Ein Bild, das Entwurf, Zeichnung, Strichzeichnung, Clipart enthält.  Automatisch generierte Beschreibung | 3D-Modell von PIFuHD, gerendert in Blender |
| the famous martin luther from germany, robe from the Renaissance, T - Pose for gamedesign, standing in front of a plain blue background. neutral magenta background, T - Pose, whole body, face looking in the camera, color film | Ein Bild, das Kleidung, Mantel, Person, Ärmel enthält.  Automatisch generierte Beschreibung | Ein Bild, das Tanz, Kleidung, Kunst, Tanzen enthält.  Automatisch generierte Beschreibung |

In der Tabelle XXX ist wird gezeigt welcher Prompt verwendet wird um das Bild zu erzeugen welches für PIFuHD verwendet wird. Das 3D-Modell das mit PIFuHD erzeugt ist zeigt wenige Fehler auf, die aber im Nächsten Abschnit mit Hilfe von der 3D-Software Blender verbessert werden.

### Verringern von Vertices in Blender

|  |  |
| --- | --- |
| Ein Bild, das Screenshot, Text, Multimedia-Software, Digitales Compositing enthält.  Automatisch generierte Beschreibung | Ein Bild, das Text, Multimedia-Software, Screenshot, Software enthält.  Automatisch generierte Beschreibung |
| Ein Bild, das Text, Screenshot, Multimedia-Software, Software enthält.  Automatisch generierte Beschreibung | Ein Bild, das Screenshot, Multimedia-Software, Grafiksoftware, Software enthält.  Automatisch generierte Beschreibung |

In einem Kurzen Test wurde getestet ob die Unreal Engine 5 mit einem Unbehandelten 3D-Modell von Martin Luther lauffägig ist. Es wurde das 3D-Modell geriggt und in Blender Importiert. Das ergebnis war, das die Unreal Engine 5 nach wenigen sekunden Abstürzt. Es wird vermutet, das das Mesh, was PIFuHD zu viele Verticies unf Faces besitzt.

Das Ziel ist, die Anzahl der Verticies und Poligone zu verringern um Rechenkapazität in de Unreal Engine 5 zu sparen.

Der erste Schritt ist, dass von PIFuHD erzeugte 3D-Modell in Blender zu laden. In Blender ist es möglich, über den Menüpunkt File → Import → Wavefront die obj-Datei zu importiert.

Nachdem die Obj-Datei in Blender importiert ist, wird sie im Eddit Mode in Blender bearbeitet. Im Edit Mode kann durch die Auswahl eines Vertex mit der rechten Maustaste ausgewählt werden. Mit STRG + L werden alle mit dem Modell verbundenen Vertices ausgewählt. Das Modell erscheint nun orange und alle Teile, die nicht direkt mit dem großen 3D-Figur von Martin Luther verbunden ist, sind noch grau. Mit der Tastenkombination STRG + I wird die Auswahl invertiert, nun sind alle Vertices innerhalb des großen 3D-Modell grau und alle außerhalb liegenden Vertices Orange. Mit dem Shortcut X ist es möglich alle Vertices zu löschen die außerhalb des 3D-Modells liegen.

Nachdem alle Vertices gelöscht sind, die außerhalb des 3D-Modell von Martin sind, wird im anschließenden Arbeitsschritt die restlichen Vertices mit der Blender-Funktion Merge verschmolzen.

Damit alle Vertices des 3D-Modells zueinander mit einem gewissen Abstand gemerged werden, werden alle Vertices über den Shortcut A alle Vertices ausgewählt. Durch das anschließende Betätigen des Shortcuts M wird ein Kontextmenü von Blender angezeigt, welche Merge-Funktion benutzt werden kann. Durch die Auswahl der Merge-Funktion By Distance wird unten links ein weiteres Kontextmenü geöffnet, wo die Distanz eingestellt wird, welchen Mindestabstand die Vertices zueinander haben dürfen, um miteinander verschmolzen zu werden. In diesem Beispiel wird die Distanz von 0,0001 m auf 0,001 m verringert.

Das Ergebnis von den zwei Arbeitsschritten wurden die ursprünglichen 105.000 Vertices auf 96.000 Vertices und 211.000 Polygone auf 192.000 Polygone verringert.

### Nachbesserung des erzeugten Modells in Blender

Nachdem der Allgemeine Polycount verringert ist, ist der nächste Schritt unsaubere Stellen, die bei der Erzeugung des 3D-Modells durch PIFuHD entstanden sind nachzubessern. In diesem Beispiel besitzt Martin Luther an der rechten Hand, einen aus der Hand ragenden Stab. Dieser Stab wurde im Edit Mode mit dem Shortcut B ausgewählt und anschließend nach der Auswahl mit dem Shortcut M und der Merge-Funktion At Center, die ausgewählten Bereiche zu einem Vertex verbunden. Dieser Vertex kann nun mit dem Shortcut G so platziert werden, dass die Hand etwas natürlicher wirkt.

Blender bietet weitere Funktionen wie zum Beispiel den Sculp Mode, um das 3D-Modell zu optimieren. Für die Entwicklung des Prototyps ist dieser Zustand jedoch ausreichend und der Sculp Mode wurde nicht verwendet.

### Texturieren in Blender

Damit die Hauptfigur am Ende im Prototyp nicht grau ist, bekommt das Modell Martin Luther noch eine Textur. Für die Textur wir das von Midjourney erzeugte Bild aus Abbildung XXX verwendet, welches PIFuHD verwendet hat, um das 3D-Modell zu generieren.

Im folgenden Abschnitt wird gezeigt, wie das 3D-Modell in Blender texturiert wird. Blender bietet die Möglichkeit ein zweites Bearbeitungsfenster hinzuzufügen, welches als UV-Editor benutze wird. Über den Menüpunkt Image → Open wird das Bild von Martin Luther geladen. Damit die Textur des 3D-Modells korrekt angezeigt wird, wird, rechts über das Propertie-Fenster → Material Properties → der gelben Punkt neben Base Color → Image Textur die Materialeigenschaft Texture Image eingestellt. Im Propertie-Fenster erscheint die Möglichkeit, ebenfalls das Bild aus Abbildung 4.18 zu laden, welche die Standard-Materialeigenschaft in eine Image Textur umwandelt.

Nach diesen Voreinstellungen ist es möglich, das 3D-Modell Blender-Funktion Unwrap auszupacken werden. Um die Unwarp-Funktion Unwrap fromView zu benutzen, muss die View (Betrachtung auf das 3D-Modell) im Edit Mode so eingestellt werden, dass es die View von der Textur Vorlage entspricht. In diesem Beispiel ist es möglich, durch das Betätigen des Shortcuts 1 auf dem Numpad zu benutzen. Numpad 1 stellt die View direkt von vorne auf das 3D-Modell ein. Anschließend werden alle Vertices durch den Shortcut A markiert. Durch das Verwenden des Shortcut U und die Wahl der Unwrap-Funktion Unwrap from View werden alle Vertices auf die Texturvorlage projiziert.

Im UV-Editor wird das in 2D projizierte 3D-Modell so vergrößert, dass es mit der Vorlage übereinstimmt.

Ein Problem nach dem Auspacken des 3D-Modells, nach dieser beschriebenen Methode ist, dass das 3D-Modell von vorne genauso texturiert ist wie von hinten. Lösung dieses Proplems ist, das die Faces von dem Rücken und den Hinterkopf extra ausgewählt werden und im UV Editor an stellen plaziert werden, die besser passeln.

Zum Beispiel werden die Blauen stellen auf dem Rücken im Edit Mode mit dem Shortcut B markiert und anschließend im UV-Editor in braune Bereiche des Mantels verschoben.

Nachdem der Rücken und den Kopf nachtexturiert sind, fehlt noch der letzte wichtige Schritt, damit Martin Luther in der Unreal Engine 5 zum Leben erweckt werden kann, und zwar das Rigging.

### Rigging in Blender

Damit der Hauptcharakter nicht nur in seiner T-Pose verweilt, sondern sich mit einer Laufanimation fortbewegt und mit einer Sprunganimation springt, benötigt das 3D-Modell von Martin Luther ein Rig.

Bevor das 3D-Modell in Blender geriggt wird, benötigt Blender das Addon Game Rig Tools von CGDive was durch eine Google-suche zu finden ist.

Nachdem das Game Rig Tool von CGDive als Addon in Blender hinzugefügt ist, kann über die Funktion des Addons Initiate Mannequin ein Unreal Engine 5 kompatibles RIG zu dem 3D-Modell hinzugefügt werden. Über das UE5 Manny TWEAK das über die Scene Collection auswählbar ist, können die einzelnen Bones des Rigs innerhalb des 3D-Modells platziert werden. Mit dem Shortcut G im Pose Mode ist es nun möglich, die einzelnen Bones so anzuordnen, wie es Analog am menschlichen Körper möglich ist, bis hin zu einzelnen Finger-Bones.

Nachdem die Bones vom UE5 Manny TWEAK fertig platziert sind, kann über das Addon-Fenster von Game Rig Tool, nur das Unreal-Rig sichtbar gemacht werden, in dem der Weiße Punkt neben Unreal betätigt wird.

Nachdem nur noch das 3D-Modell und das Unreal-Rig sichtbar ist, wird in den Object Mode gewechselt. Um zwei Objekte auszuwählen ist es wichtig das bei Gedrückter Shift-Taste, erst das 3D-Modell von Martin Luther ausgewählt wird, dann das Unreal-Rig.

Nachdem das 3D-Modell von Martin Luther und das Unreal-Rig im Objekt Mode in der korrekten Reihenfolge ausgewählt ist, kann über den Menüpunkt File → Export → FBX, das geriggte Modell exportiert werden. Beim exportieren werden folgende Optionen verwendet:

• Limit to Selected Objects

• Object Types; Amature und Mesh

• Transform → Apply Transform → Ja

• Geometry → Smoothing → Face

• Armature → Add Leaf Bones → Nein

• Bake Animation → Nein

Nachdem diese Einstellung wird ein Datei-Namen für die fbx-Datei gewählt und den Button mit der bezeichnung Export FBX exportiert und gespeichert. ̈

### Importieren des Haupcharakters in Unreal Engine 5  ̈

Der Haupcharakter ist so weit fertig zum Importieren in die Unreal Engine 5, nach dem Import kann der Hauptcharakter im Prototyp getestet werden.

Über den Epic Games Launcher kann ein neues Projekt erstellt werden. In diesem Beispiel wird das Third-Person-Template verwendet. Das Third-Person-Template ist mit einer Spielfigur ausgestattet, welche von dem Spieler gesteuert werden kann. Damit nicht der Standart-Charakter, sondern Martin Luther die Spielfigur ist, sind noch folgende letzte Schritte notwendig.

In der Entwicklungsumgebung Unreal Engine 5 gibt es eine Blueprint Class vom Typ Charakter mit der bezeichnung BP\_ThirdPerson.

Die BP\_ThirdPersonCharacter befindet sich unter dem Ordner All → Content → ThirdPerson → Blueprints. Uber den Button Import der sich im Contentbrowser befindet, kann die von Blender exportierte fbx-Datei importieren werden.

Nach dem auswählen der fbx-Datei, öffnet sich die FBX-Importoptionen, wo das SK Mannequin für das Skeleton ausgewählt werden muss. Durch den anschließenden betätigen der Buttons Import All wird 3D-Modell und das entsprechende Material in der Unreal Engine 5 im Contentbrowser angezeigt.

Um das Skeletal Mesh zu ersetzen, wird der BP\_ThirdPersonCharacter mit einem Doppelcklick zur bearbeitung geöffnet. Die BP\_ThirdPersonCharacter besteht aus verschiedenen Components, von diesen verschiedenen Componets wird das Mesh(CharacterMesh0) ausgewählt. Nach dem ich das Mesh(CharacterMesh0) ausgewählt ist, kann ich in den geöffneten Details das Skeletal Mesh Asset durch das von Martin Luther getauscht werden. Automatisch wird auch die dazu gehörige Textur geändert.

Nach diesen Schritten kann Martin Luther im Prototyp als Hauptcharakter benutzen werden. Um das zu Testen ist das betätigen des grünen Play-Symbol notwendig.

Der Meilenstein Hauptfigur ist Erledigt. Es wurde eine Hauptfigur erstellt, die mit der Unreal Engine 5 Kompatibel ist.

## Meilenstein: Gebäude

Nach der Erstellung der Hauptspielfigur, ist der nächste Meilenstein die Erstellung einer Spielwelt. Die Idee ist, ein Dorf in der Unreal Engine 5 zu erschaffen. In diesem Dorf befinden sich verschiedene Gebäude.

In diesem Kapitel wird gezeigt, wie Gebäude mit Hilfe verschiedener KI-Systemen in der Unreal Engine 5 realisiert werden. In diesem Kapitel und in den folgenden Meilensteinen wird nicht mehr detailliert auf alle Bearbeitungsschritte wie bei der Erstellung der Hauptfigur eingegangen, sondern nur auf Schritte, die sich im Prozess stark im Gegensatz zur Erstellung der Hauptfigur unterscheiden.

Fachwerkhäuser wie sie in Deutschland zu finden sind, werde seit mindestens 1353 gebaut wie zum Beispiel das Schiefe Haus in Ehingen was 1353 gebaut wurde oder das Spital am Viehmarkt ebenfalls in Ehingen welches 1532 gebaut wurde (Ege, 2013, S 6).

Im Jahr 1517 hat Martin Luther angeblich seine 95 Thesen and das Kirchtor in Wittenberg genagelt (Mr Wissen 2 Go).

Als Ein-Mann-Vidospiel-Entwickler ist es möglich aus diesen beiden Tatsachen eine Spielwelt zu erstellen, die einen Geschichtlichen Kulturellen Kontext zur deutschen Geschite hat.

Die idee ist es ein ein Fachwerk-Dorf zu Entwickeln welches so 1517 gegeben hätte. Der Architekturstil ist Fachwerk.

Wie verschiedene Ideen und Ansätze umgesetzt werden, werden in den folgenden Kapitel näher erläutert.

### Erste Ansatz das simple Fachwerkhaus

Der erste Ansatz ist, Häuser mit Hilfe von einem einfachen 3D-Modell umzusetzen, die in Blender erstellt werden. Dieses 3D-Modell bestand aus einem Quader mit einer Spitze. Kurz, ein einfaches Haus.

Mit Midjourney werden Texturen generiert, die Fachwerkhäuser nachempfunden sind. Diese Texturen und das einfache Haus werden mit Hilfe von Blender verbunden. Diese einfachen Fachwerkhäuser können als FBX-Datei in Blender exportiert, in Unreal Engine 5 importiert und in der Landschaft verteilt werden.

### Zweiter Ansatz: Modellieren eines Fachwerkhauses mit Blender

Der zweite Ansatz ist, mit Blender individuelle Häuser zu gestalten. Es werden, wie beim ersten Ansatz Bilder von Midjourney erzeugt. Im Gegensatz zum ersten Ansatz, wird kein Würfel mit einer Spitze in Blender texturiert, sondern die Balken werden mit der Moddelierungsfunktion Extrude modelliert.

Extrud ist eine Moddelierungsfunktion, mit dem man ein Face von einem 3D-Modell herausheben kann. Bei diesem Herausheben entstehen wieder neue Faces, die jeweils wieder extrudiert werden können. Nachdem das Gefache (die sichtbaren Balken eines Fachwerkhauses) des Fachwerkhauses fertig ist, können die Wände mit Hilfe von Planes erzeugt werden.

Der Vorteil dieses Ansatzes ist, dass das 3D-Modell mehr Oberflächenstruktur aufweist, und die Balken einen Schatten werfen in der Unreal Engine 5.

Der Nachteil dieses Ansatze ist das es sehr viel Zeit dauert, Das Haus was in Abbildung XXX zu sehen ist, wurde über fünf Tage Modelliert.

### Dritter Ansatz Zusammenbauen eines Fachwerkhaus mit Blender

Anstatt das Gefache aus einem Mesh zu modellieren, wie in Versuch Zwei, werden hier für jeden Balken ein einzelner Cube von Blender im Objekt Mode erzeugt. Diese Cubes werden So Skaliert, das sie mit der Vorlage die mit Hilfe von Midjourney erzeugt wird, entsprechen.

Mit Hilfe von Planes werden auch hier wie im zweiten Ansatz Wände erzeugt.

Das mit dieser Methode erzeugte Haus hat besteht 124 Cubes und 33 Planes. Aber auch diese Methode hat zwei Tage gebraucht, um ein Fachwerkhaus zu gestalten.

Es musste eine Methode her, die es ermöglicht innerhalb von Stunden Individuelle Fachwerkhäuser zu gestalten. Diese Methode wird im folgenden Kapitle Ansatz Vier erklärt.

### Vierter Ansatz: Dorfbaukasten

Eine Inspiration für den dritten Ansatz ist Valheim, ein Survival-Spiel von den Iron Gate Studio. In diesem Spiel gibt es ein Baukastensystem, in dem der Spieler sein eigenes Haus bauen kann.

In Valheim kann der Spieler verschiedene vordefinierten Wände, Balken, Fußböden, Dächer und Materialarten auswählen, um damit seine Behausung zu gestalten.

Hinzu kommen andere Elemente wie Zäune und Gemüse um ein Garten zu erschaffen, Werkbänke, Tische und Stühle um eine Inneneinrichtung zu kreieren und sogar Teppiche aus Tierfelle und Trophäen die man an die Wand hängen kann um eine Dekorative Charakter in die Behausung eines Spielers zu schaffen.(Valheim Bausystem)

Aus dieser Inspiration wird ein Vierter Ansatz entwickelt, und zwar der so genannten Dorfbaukasten.

Dieser Dorfbaukasten besteht ebenfalls aus Wänden, Balken, Dächer, Dachgiebel, Fußböden und Tapeten.

Bei der Betrachtung eines Fachwerkhauses, kann man zu Schluss kommen, das ein Fachwerkhaus aus einfachen Geometrischen Formen bestehen. Zum Beispiel Geometrien wie Quader für Wände, Balken, Dächer, Türen und Fenster. Der Dachgiebel von einem Fachwerkhaus ist ein dreieckiges Prisma. Diese einzelnen Elemente werden nachgebaut, so dass man in der Unreal Engine 5 nur noch die einzelnen Bauelemente zusammenstecken muss.

Ein weiteres Ziel ist das die einzelnen Bauelemente in der Unreal Engine 5 über den integrierten Blueprint-Editor zu programmieren, damit die Materialeieigenschaft sich untereinander unterscheiden. Neben den Materialeieigenschaften werden die einzelnen Bauelemente so Programmiert, das die Bauelemente in ihrer Länge oder Breite über Parameter bestimmen lassen.

Anhand folgenden Beispiel wird gezeigt wie das Bauelement Balken Programmiert wird:

Der Balken ist eine Blueprint Class vom Typ Actor.

Durch das Erzeugen einer neuen Actor Class, kann durch ein Doppelklick diese Actor geöffnet werden. Mit Hilfe des Buttons ADD Componets wird ein Cube hinzugefügt und die Skalierung in diesem Beispiel auf 40, 40, 100 verändert. Das bedeutet der Balken ist 40cm Breit und Tief, und hat eine Länge von 100cm.

Im Construction Script kann mit Hilfe von Nodes dem Balken besondere Eigenschaften programmiert werden wie, zum Beispiel die zufällige Auswahl des Materials oder das bestimmen der Länge mit Hilfe eines Parameters.

Nachdem Alle Bauelemente Programmiert wurden, kann in der Unreal Engine 5 die einezelnen Bauelemente in die Spielwelt per Drag & Drop gezogen werden. Mit der Einstellung Set the Position Grid Snap value kann eingestellt werden, das die Bauelement an eine art imagineräs gitter plaziert werden kann. Der Wert wurde in der Umsetzung auf 50 gestellt. Über die Einstellung Enabeles or disables snaping tot he Grid when dragging objects ground kann diese Einstellung an und ausgestellt werden. Für die Umsetzung wird die Einstellung angestellt.

Mit den Beiden eingerichteten Einstellung verhällt sich der Dorbaukasten wie in dem Videospiel Valheim, was als Ein-Mann-Videospielentwickler sehr viel Zeit spart.

## Meilenstein: Nebenfiguren

Im Dritten Meilenstein Nebenfiguren wird beschreiben, wie für den Prototypen Nebenfiguren, kurz NPCs, erstellt werden. Ähnlich wie beim Erstellen der Hauptfigur werde durch ChatGPT die verschiedene NPCs beschrieben. Zu dieser Beschreibungen gehören Alter, Geschlecht, Beruf und charakterliche Eigenschaften.

|  |  |
| --- | --- |
| Ein Bild, das Cartoon enthält.  Automatisch generierte Beschreibung | Hi ChatGPT. Ich möchte ein Skript für eine Handlung in einem Videospiel schreiben.  Meine Hauptfigur Martin luther trifft auf ein paar Dorfbewohner und unterhält sich mit ihnen.  Martin luther kann zwischen jeweils drei fragen- oder Antwortmöglichkeiten aussuchen, der eventuell Einfluss auf die anderen Gespräche haben.  In meinem Prototyp möchte ich nicht mehr als 10 NPC verwenden.  Bitte beschreibe diese NPC mit Name, Alter, Geschlecht, Beruf und Gemütszustand. |
| Ein Bild, das Symbol, Grafiken, Schrift, Logo enthält.  Automatisch generierte Beschreibung | Natürlich, ich helfe Ihnen gerne bei der Erstellung der NPCs für Ihr Videospiel-Skript. Hier sind zehn Dorfbewohner, denen Martin Luther begegnen kann:  1. Name: Anna  Alter: 30  Geschlecht: Weiblich  Beruf: Bäuerin  Gemütszustand: Ängstlich, da ihre Ernte schlecht war und sie sich um ihre Familie sorgt.  2. Name: Johann  Alter: 55  Geschlecht: Männlich  Beruf: Schmied  Gemütszustand: Stolz, weil er kürzlich ein beeindruckendes Schwert geschmiedet hat.  (…)  10. Name: Jakob  Alter: 35  Geschlecht: Männlich  Beruf: Dorfwächter  Gemütszustand: Misstrauisch, da er Gerüchte über Diebe im Dorf gehört hat.  Diese Charaktere könnten Martin Luther eine Vielzahl von Informationen, Aufgaben und Handlungsmöglichkeiten bieten, je nachdem, welche Fragen oder Antworten er wählt. Viel Erfolg bei der Entwicklung Ihres Videospiel-Prototyps! |

ChatGPT hat 10 NPCs mit den Angeforderten Eigenschaften generiert. Der Lesbarkeit halber, wurden nur NPC Eins und Zwei sowie NPC Zehn aus der Ausgabe von ChatGPT in dieser Bachelorarbeit abgebildet. Für den Prototyp werden Anna und Johann umgesetzt.

Mit diesen Beschreibungen werden Prompts für Midjourney-Formel selbst entwickelt und Midjourney übergeben. Folgende Bilder werden PIFuHD übergeben, um daraus 3D-Modell zu generieren.

|  |  |  |
| --- | --- | --- |
| Ein Bild, das Cartoon enthält.  Automatisch generierte Beschreibung | 25 jears old female farmer, working cloth from the Renaissance, standing in T-Pose for gamedesign, standing in front of a plain orange background. neutral pale orange background, whole body, face looking in the camera, color film, T-Pose, T Pose, tpose | 50 jears old smith in working cloth from germany, working cloth from the Renaissance, TPose for gamedesign, on white background, whole body, face looking in the camera, color film |
| Ein Bild, das Entwurf, Zeichnung, Strichzeichnung, Clipart enthält.  Automatisch generierte Beschreibung |  |  |
| PIFuHD |  |  |

Da die NPCs dazu dienen, dem Spieler zu begegnen und keine Lauf- oder Sprunganimation benötigen, brauchen die NPCs auch nicht in der T-Pose dargestellt werden. Für die NPCs wird eine natürliche Pose gewählt.

Nachdem dem beiden Bilder von Midjourney erzeugt werden, werden sie mit Hilfe von PIFuHD zu 3D-Modellen umgewandelt.

In Blender erfolgt eine Nachbearbeitung der NPCs, indem die Anzahl der Vertices reduziert, die 3D-Modelle texturiert und stark verformte Stellen nachgebessert werden.

Für den Prototyp werden die NPCs ohne Rig als FBX-Datei exportiert.

In der Unreal Enginge 5 werden die 3D-Modell importiert und anschließend Blueprintklassen von typ Character erzeugt. Diese Blueprintklassen dienen dazu, um sie zu NPCs zu programmieren, die mit dem Hauptcharakter interagieren können.

Die NPC-Blueprintklassen bekommen jeweils per Drag & Drop das Static Mesh von Anna oder Johann.

Die NPCs sind soweit Erstellt und können in der Spielwelt verteilt werden in Abbildung XXX sehen wir die Spielfigurne in der Spielwelt. Anna ist eine Bäuerin und sie wurde in der nähe eines Ackers plaziert. Johann ist ein Schmied und für ihn wurde eine Schmiede gebaut.

|  |  |
| --- | --- |
|  |  |

Die in den Abbildung XXX und XXX gezeigten 3D-Modellen wie zum Beispiel die Mistgabel in Annas Hand und der Schmiedeaboss zu Johann Linken sind aus der In-Engine-Conten-Browser Quixel Bridge.

|  |  |
| --- | --- |
|  |  |

In Abbildung XXX und XXX sehen wir Martin Luther mit den beiden NPCs Anna und Johann. Anhan diesem Beispiel kann man den unterschied erkenne ob Das 3D-Modell welches von PIFuHD erzeugt wird, geriggt oder nicht geriggt wird.

Martin Luthers 3D-Modell sieht um dem Hüftbereich sehr breit aus, und insgesammt ein Kopf kleliner im gegensatz zu Anna und Johann. Das liegt daran, das Martin Luther nicht sauber geriggt ist. Durch das Riggen wird das Mesh verform. Den grad der Verformung kann man beheben in dem das Rig in Blender neu erzugt wird.

Ein weitere Unterschie, den Martin Luther zu den beiden NPCs Johann und Anna aufweist ist, das Martin ein gut texturierten Rücken und Hinterkopf hat.

Obwohl Anna und Johann auf Abbildung XXX mit dem Rücken zur Kamera stehen, kann man ihre Gesichter erkenn.

Dieses Problem wurde behoben, in dem Anna vor einer großen Pflanze gestellt wurde und Johann mit dem Rücken zu einer Wand in seiner Schmiede.

Das nächste ziel in der Entwicklung des Protorypen ist das entwickeln eines Dialogsystems und das Einbinden von Sounddateien die es erlaubt Martin Luther mit den NPCs sich zu unterhalten.

Das erreichen dieser beiden Ziele werden in den folgenden Meilensteinen erläutert.

## Meilenstein: Dialogsystem

Um ein Dialogsystem in der Unreal Enginge 5 zu entwickeln, wurden zwei Ansätze benötigt.

Das Dialogsystem soll als Infrastruktur dienen damit die Sound Datei von den jeweiligen Spielfiguren richtig abgespielt werden. Das Dialogsystem wird mit Hilfe der Visuellen Entwicklungsumgebung der Unreal Engine 5 Blueprint umgesetzt.

### Erster Ansatz: Entwicklung des Dialogsystem mit Hilfe von ChatGPT

Der erste Ansatz ist mit Hilfe von ChatGPT ein Dialogsystem zu entwickeln. ChatGPT besitzt die Fähigkeit Codeblöcke in verschiedenen Programmiersprachen zu schreiben. Die Idee ist es ChatGPT aufzufordern ein Dialogsystem zwischen den NPCs und Martin luther zu generiern.

Folgender Prompt wurde gegenüber ChatGPT formuliert.

|  |  |
| --- | --- |
| Ein Bild, das Cartoon enthält.  Automatisch generierte Beschreibung | Hallo ChatGPT, kannst du mir bitte helfen ein Dialogsystem für mein Prototy zu erstellen? Ich entwickle mein Videospiel mit der Unreal Engine 5.  Ich möchte mein Hauptcharakter mit anderen NPC reden lassen.  Ich hatte mir dabei gedacht, das die Keyboardtaste "E" diese Aktion durchführen soll. |
| Ein Bild, das Symbol, Grafiken, Schrift, Logo enthält.  Automatisch generierte Beschreibung | Ja, natürlich! Ich kann dir eine allgemeine Anleitung geben, wie du ein einfaches Dialogsystem für dein Videospiel mit der Unreal Engine 5 erstellen kannst. Beachte jedoch, dass dies nur ein grundlegendes Beispiel ist und je nach deinem Spiel und den spezifischen Anforderungen des Dialogsystems weitere Anpassungen erforderlich sein könnten. Hier sind die Schritte:  Schritt 1: Erstelle einen neuen Blueprint für den NPC  - Klicke mit der rechten Maustaste im Content Browser auf "Blueprint-Klasse" und wähle "Character" aus, um einen neuen Blueprint für den NPC zu erstellen.  - Öffne den neuen Blueprint und füge eine Textvariable hinzu, um den aktuellen Dialogtext des NPCs zu speichern. Nenne sie z.B. "CurrentDialogText".  - Erstelle eine Funktion, die den Dialog des NPCs aktualisiert. Du kannst sie "UpdateDialogText" nennen und die gewünschten Dialogtexte als Parameter übergeben.  Schritt 2: Die Interaktion einrichten  - Öffne die Blueprint-Klasse deines Hauptcharakters und erstelle eine Variable vom Typ "bool" mit dem Namen "bCanInteract". Diese Variable wird verwendet, um festzustellen, ob der Spieler in der Nähe eines NPCs ist und mit ihm interagieren kann.  - Implementiere eine Funktion namens "Interact" oder ähnlich. In dieser Funktion prüfst du, ob "bCanInteract" wahr ist und wenn ja, rufst du die Funktion "StartDialog" des NPCs auf.  Schritt 3: Dialogsystem in den NPCs einrichten  - Öffne den NPC-Blueprint und füge eine Kugelkomponente hinzu, die als Kollisionsbereich für die Interaktion dient. Nenne sie z.B. "InteractionSphere".  - Erstelle in der NPC-Blueprint-Eventgraph einen "OnComponentBeginOverlap"-Event, der ausgelöst wird, wenn der Spieler in die InteractionSphere des NPCs eintritt. Verbinde diesen Event mit einem Knoten "Set bCanInteract" und setze den Wert auf "wahr".  - Erstelle einen weiteren Event namens "OnComponentEndOverlap", der ausgelöst wird, wenn der Spieler die InteractionSphere verlässt. Verbinde ihn mit einem Knoten "Set bCanInteract" und setze den Wert auf "falsch".  - Erstelle eine benutzerdefinierte Funktion namens "StartDialog". In dieser Funktion rufst du die Funktion "UpdateDialogText" des NPCs auf, um den aktuellen Dialogtext einzustellen. Zusätzlich kannst du ein Dialogfenster oder einen Text im UI anzeigen, um den Dialogtext für den Spieler sichtbar zu machen.  Schritt 4: Eingabeaufforderung für die Interaktion hinzufügen  - In der Eventgraph des Hauptcharakter-Blueprints erstellst du einen "InputAction"-Event, der durch Drücken der Taste "E" ausgelöst wird.  - In diesem Event fügst du einen Knoten "Branch" hinzu, um zu überprüfen, ob "bCanInteract" wahr ist.  - Wenn die Bedingung erfüllt ist, rufe die Funktion "Interact" des Hauptcharakters auf.  Das war eine grundlegende Anleitung, um dein Dialogsystem einzurichten. Natürlich kannst du es weiter ausbauen, indem du mehrere Dialogoptionen für den Spieler hinzufügst, die Geschichte fortschreiten lässt oder Animationen für die Charaktere einbaust. Aber hoffentlich hilft dir dies als Ausgangspunkt, um loszulegen. Viel Erfolg bei deinem Projekt! |

Mit der ersten Ausgabe in Tabelle XXX ist es selbst nicht möglich gewesen, mit dieser Anweisung ein Dialogsystem zu entwickeln. Möglich Uhrsachen könnten mit diesem Ansatz sein, dass in schritt Zwei von einer Funktion Start Dialog geschrieben wird, aber diese Funktion gibt es nicht an dieser Stelle, sondern die Funktion Start Dialog wird in Schritt Drei erste erzeugt. Diesen Fehler wird über folgenden Prompt ChatGPT mitgeteilt.

|  |  |
| --- | --- |
| Ein Bild, das Cartoon enthält.  Automatisch generierte Beschreibung | in schritt 2 kann ich nicht die funktion "StartDialog" aufrufen. |
| Ein Bild, das Symbol, Grafiken, Schrift, Logo enthält.  Automatisch generierte Beschreibung | Entschuldigung für das Missverständnis. Du hast recht, da der Dialog in den NPCs stattfindet, sollten wir die Funktion nicht "StartDialog" nennen. Stattdessen rufen wir die Funktion zum Aktualisieren des Dialogtexts direkt im NPC-Blueprint auf. Hier ist der aktualisierte Schritt 2:  Schritt 2: Die Interaktion einrichten  - Öffne die Blueprint-Klasse deines Hauptcharakters und erstelle eine Variable vom Typ "bool" mit dem Namen "bCanInteract". Diese Variable wird verwendet, um festzustellen, ob der Spieler in der Nähe eines NPCs ist und mit ihm interagieren kann.  - Implementiere eine Funktion namens "Interact" oder ähnlich. In dieser Funktion prüfst du, ob "bCanInteract" wahr ist und wenn ja, führst du eine Trace-Funktion aus, um zu überprüfen, welcher NPC in der Sicht des Spielers liegt. Dann rufst du die Funktion zum Aktualisieren des Dialogtexts direkt im ausgewählten NPC auf.  Hier ist ein Beispiel für die Funktionsimplementierung im Hauptcharakter-Blueprint:  ```  UFUNCTION(BlueprintCallable)  void Interact()  {  if (bCanInteract)  {  // Linien-Trace durchführen, um den NPC zu treffen  FVector Start = FirstPersonCameraComponent->GetComponentLocation();  FVector End = Start + FirstPersonCameraComponent->GetForwardVector() \* InteractionRange;  FHitResult HitResult;  FCollisionQueryParams QueryParams;  QueryParams.AddIgnoredActor(this); // Ignoriere den Hauptcharakter  if (GetWorld()->LineTraceSingleByChannel(HitResult, Start, End, ECC\_Visibility, QueryParams))  {  AMyNPCCharacter\* NPC = Cast<AMyNPCCharacter>(HitResult.GetActor());  if (NPC)  {  NPC->UpdateDialogText("Hallo, ich bin der NPC. Willkommen im Spiel!");  // Hier könntest du auch den Dialogtext für den NPC festlegen  }  }  }  }  ```  In diesem Beispiel gehe ich davon aus, dass dein Hauptcharakter eine Kamera-Komponente namens "FirstPersonCameraComponent" hat und dass du eine Variable "InteractionRange" festlegst, um die Reichweite der Interaktion festzulegen.  Das Dialogsystem im NPC-Blueprint bleibt unverändert. Die Funktion "UpdateDialogText" im NPC-Blueprint, die wir zuvor erstellt haben, sollte den Dialogtext festlegen, den der NPC anzeigen soll, wenn er mit dem Spieler interagiert wird.  Ich hoffe, das klärt die Verwirrung. Wenn du weitere Fragen hast oder zusätzliche Unterstützung benötigst, stehe ich gerne zur Verfügung! |

An diesen Punkt des Experiments entstand Verwirrung, da von einer Trace-Funktion geschrieben w urde, obwohl in der vorrigen Ausgabe eine InteractionSphere erzeugt wurde.

Es ist möglich d as ChatGPT falsche Inromationene bekommen hat, da ein Fehler in Schritt zwei gemeldet wurde, aber Als Entwickler Schritt 3 von der Ausgabe von Tabelle XXX durchgeführt hat, und zwar das erstellen einer InteractionSphere.

Mit dem Folgenden Prompt wird gefragt, warum eine Trace-Funktion benötigt wird, obwohl schon eine InteractionSphere erzugt wurde

|  |  |
| --- | --- |
| Ein Bild, das Cartoon enthält.  Automatisch generierte Beschreibung | warum ein trace? ich habe doch eine interactionssphere mit der ich prüfen kann, ob mein hauptcharakter in der nähe eines npc ist. |
| Ein Bild, das Symbol, Grafiken, Schrift, Logo enthält.  Automatisch generierte Beschreibung | Du hast recht, mein Fehler. Es ist nicht notwendig, einen Linien-Trace durchzuführen, wenn du bereits eine Interaktions-Sphäre für den NPC hast, um die Nähe zum Hauptcharakter zu überprüfen. Wir können direkt in der Interaktions-Sphäre des Hauptcharakters prüfen, ob sie mit einem NPC kollidiert. Hier ist die korrigierte Implementierung für Schritt 2:  Schritt 2: Die Interaktion einrichten  - Öffne die Blueprint-Klasse deines Hauptcharakters und erstelle eine Variable vom Typ "bool" mit dem Namen "bCanInteract". Diese Variable wird verwendet, um festzustellen, ob der Spieler in der Nähe eines NPCs ist und mit ihm interagieren kann.  - Implementiere eine Funktion namens "Interact" oder ähnlich. In dieser Funktion prüfst du, ob "bCanInteract" wahr ist und wenn ja, überprüfst du, ob der Hauptcharakter tatsächlich mit einem NPC kollidiert. Wenn ja, rufst du die Funktion zum Aktualisieren des Dialogtexts direkt im NPC auf.  Hier ist ein Beispiel für die Funktionsimplementierung im Hauptcharakter-Blueprint:  ```cpp  UFUNCTION(BlueprintCallable)  void Interact()  {  if (bCanInteract)  {  TArray<AActor\*> OverlappingActors;  InteractionSphere->GetOverlappingActors(OverlappingActors, ANPCCharacter::StaticClass());  if (OverlappingActors.Num() > 0)  {  ANPCCharacter\* NPC = Cast<ANPCCharacter>(OverlappingActors[0]);  if (NPC)  {  NPC->UpdateDialogText("Hallo, ich bin der NPC. Willkommen im Spiel!");  // Hier könntest du auch den Dialogtext für den NPC festlegen  }  }  }  }  ```  In diesem Beispiel gehe ich davon aus, dass du eine Kugelkomponente namens "InteractionSphere" im Hauptcharakter-Blueprint hast, die als Interaktionsbereich für NPCs fungiert.  Die Funktion "UpdateDialogText" im NPC-Blueprint bleibt unverändert und sollte den Dialogtext festlegen, den der NPC anzeigen soll, wenn er mit dem Spieler interagiert wird.  Ich hoffe, dass dies jetzt klarer ist. Vielen Dank für den Hinweis, und wenn du weitere Fragen hast, stehe ich gerne zur Verfügung! |

An dieser Stelle wurde das Experiment abgebrochen mit Hilfe von ChatGPT ein Dialogsystem zu entwickeln. Ziel ist es mit Hilfe von Blueprint ein Dialogsystem zu entwickeln da das Vorwissen mit Hilfe von C++ nicht vorhanden ist. Es ist auch möglich dass die Falschen Fragen an ChatGPT gestellt wurden sind.

Im Nächsten Abschnitt wird ein Dialogsystem Engwickelt mit Hilfe von Suchmaschienen wie Google und YouTube.

### Zweiter Ansatz: Entwicklung des Dialogsystem mit Hilfe von Suchmaschinen im Internet

Nach dem erfolglosen Entwickeln eines Dialogsystems mit Hilfe von ChatGPT wird eine andere Informationsquelle verwendet – Suchmaschienen.

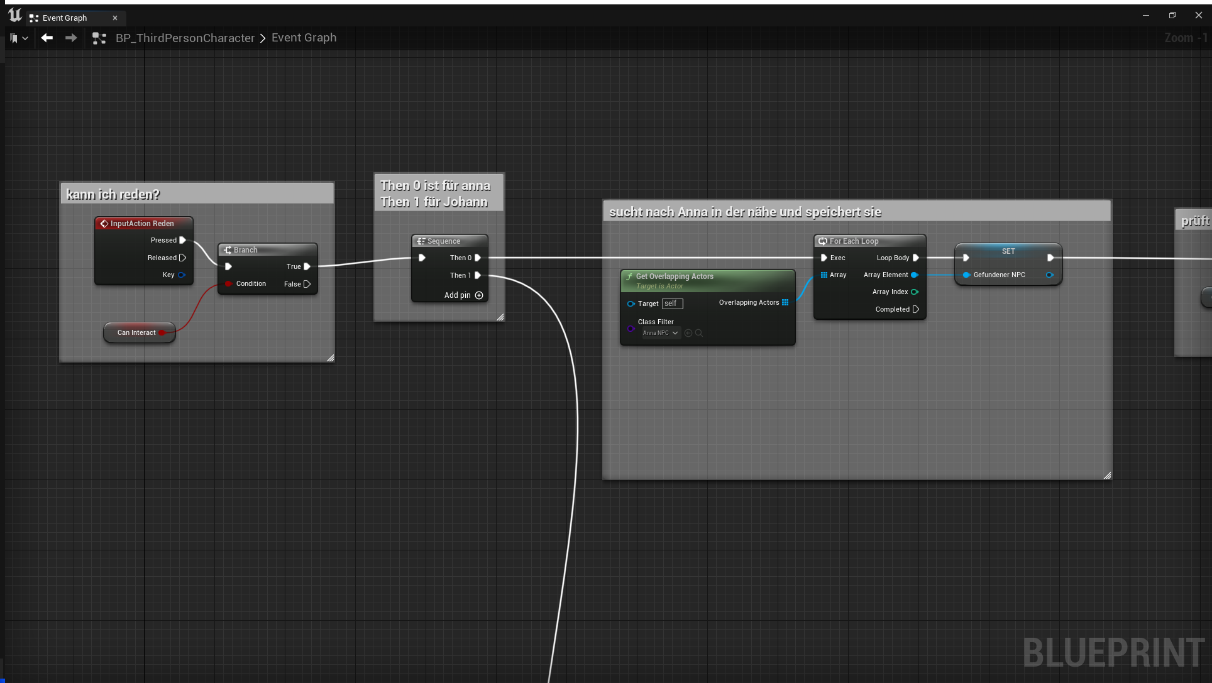
Mit der Suchmaschiene Google wurde Video auf Youtube gefunden, was erklärt wie ich ein Dialogsystem in Unreal Engine 5 erstellt wird. Das Youtube-Video trägt den Titel How to Make a Simple Dialogue System in Unreal Engine 5 von dem YouTube-User Gorka Games.

ChatGPT hat mir gegensatz zu dem Youtube-Video nicht mitgeteilt das es eine möglcihkeit gibt zwischen Blueprint-Klassen zu Komunizieren, und diese geschieht über Interfaces (Gorka Games, 2022).

Das Interface wird den Grünen Button ADD+ → Blueprint → Blueprint Interface, wird ein neues Interface erzugt. Mit doppelklick auf diese neue erzeugt Blueprint Interface wird das Interface geöffnet. In diesem Interface werden zwei neue Funktionen hinzugefügt. Einmal Talk und einmal die Funktion TalkEnd. Beide Funktionen diene zu komunikation zwischen den NPCs.

Zu dem beschriebenen Dialogsystem wurde ein zusätlicher Dilay in der Länge von dem Soundfile hinzugefugt und ein Bool, der auf false steht, falls eine Interaktion gerade nicht möglich sein sollte.

Dieser Bool verhindert, wenn ein Charakter gerade noch redet, ein weiteres Gespräch anfängt. Zusätzlich verhindert der Bool, dass keine Dialoge schnell hintereinander gestartet werden und die Charaktere aussprechen lassen.



Das Dialogsystem beginnt mit dem Hauptcharakter Martin Luther. Die Blueprintklasse PB\_ThirdPersonCharacter besitzt einen Event Graph. In diesem Event Graph ist es möglich zu programmieren, wie er sich im Spiel verhalten soll. In diesem Fall soll Martin Luther mit Anderen NPCs reden können.

Der Event Graph beginnt mit dem ImputAction Reden, welche darauf wartet das die Taste E betätigt wird. Wenn die Taste E im Spiel betätigt wird, wird als erstes geprüft, ob Martin Luther reden darf, das geschieht über den Bool CanInteract. Ist CanInteract true, dann folgt eine Sequenz. Die Sequenz dient dazu, dass ein Blueprint-Skript übersichtlich bleibt, es kann nämlich sehr schnell passieren, dass Blueprintskript unaufgeräumt wirken. In diesem Fall wird dient die Sequenz dazu den Logischen teil mit Anna von dem Logischen Teil mit Johann zu trennen.

Hinter Sequenz 0 steht der NPC Anna, woran dieses Beispiel erklärt wird. Anna und Johann sind fast identisch Programmiert, es werden nur besondere Inhalte verändert, zum Beispiel wird Anna etwas anderes gefragt, als Johann, aber die Logik dahinter ist gleich.

In der Schleife nach der Sequence werden alle Objekte der Klasse Anna gesucht die in der Nähe von Martin Luther befinden. Die nähe wird durch eine Hitbox repräsentiert. Beide haben eine Interaction (Sphere Collision) von zwei Meter, das bedeutet, das Anna und Martin Luther im Spiel mindestens vier Meter voneinander entfernt sein müssen damit GetOverlapingActor Anna findet. Diese gefundene Anna wird, falls sie in der nähe ist in GefundenenNPC gespeichert.

Ein Bild, das Screenshot, Multimedia-Software, Grafiksoftware, Software enthält.

Automatisch generierte Beschreibung

Nach dem speichern der gefundenen Anna wird geprüft, ob die gefundene Anna das Interface besitzt. Besitzt die Gefundene Anna das Interface – weiter.

Nach der Interface Prüffung, wir geprüft ob das Array vom Typ String noch nicht zu Ende ist. In diesem Array befinden sich in Textform was Martin Luter sagt. Falls es noch was zu sagen gibt - weiter.

Nach der Prüfung ob es was zu sagen gibt, wird CanInteract auf false gesetzt, damit der Spieler nicht immer wieder auf die Taste E drückt um weitere Gespräche anzufangen.

Ein Bild, das Screenshot, Text, Multimedia-Software, Software enthält.

Automatisch generierte Beschreibung

Nachdem CanInteract auf false gesetzt wurde, wird auf dem Bildschirm der Text ausgegeben, was Martin Luther zu Anna sagt.

An diese an der nächsten Stelle wird der Sound von Martin Luther gestartet. Der Sound wird aus dem Arrey SpracheMitAnna gelesen, der mit Sound-Dateien gefüllt ist.

Ein Bild, das Text, Screenshot, Multimedia-Software, Grafiksoftware enthält.

Automatisch generierte Beschreibung

Als nächstes, wird ein Delay verwender, mit der jeweiligen Länge, von der Sound-Datei, die Martin Luther spricht.

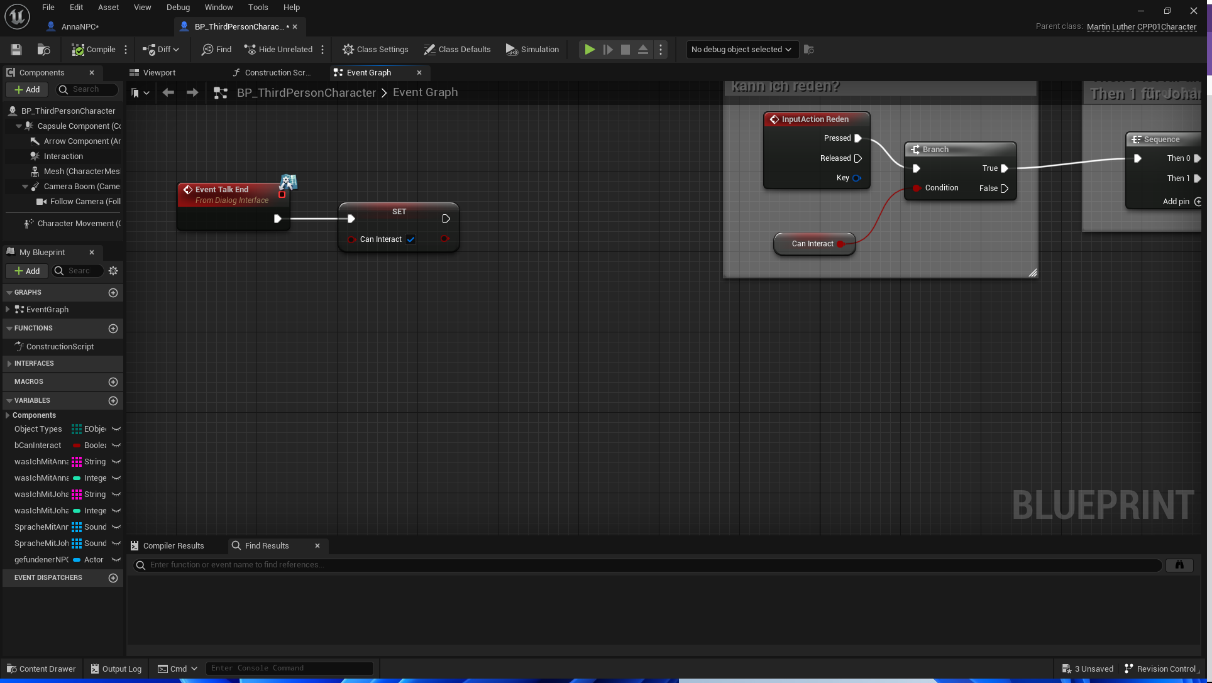
Der Delay bewirk das ist nicht die nächste Funktion aus. Ausgelöst wird bevor Martin Luther ausgesprochen hat.

Als nächstes wird das Interface aufgerufen. Das aufrufen des Interfaces hat die Folge, das Anna die Auforderung Talk bekommt.

Nach dem Aufruf des Interfaces wird der Sprachindes WasIchMitAnnaRedeIndex von null auf ein hochgezählt oder nach jedem weiteren Gespräch eins dazu addiert, bis dis der WasIchMitAnnaRedeIndex auserhalb höher als die größe der Arrays SpracheMitAnna doer WasIchMItAnnaRede ist.

Zum Schluss erhält Martin Luther eine Antwort über das Interface was das Event TalkEnd auslöst. Das auslösen des Events setzt den Bool CanInteract auf true.

Martin Luther kann nun wieder Mit Anna oder Johann reden.



Über das Interface erhält Anna die Nachricht das sie Antworten soll. Das Event Talk beginnt das Blueprint-Skript.

In dem Array Dialog wird geprüft ob es noch was zu reden gibt. Geprüft wird das anhand des TalkIndex. Wenn der Talkindex kleiner ist als die Länge des Stringarrays Dialog - weiter.

Aus dem Stringarray Dialog, wird als nächstes der String was von Anna gesprochen wird auf dem Bildschirm ausgegeben.

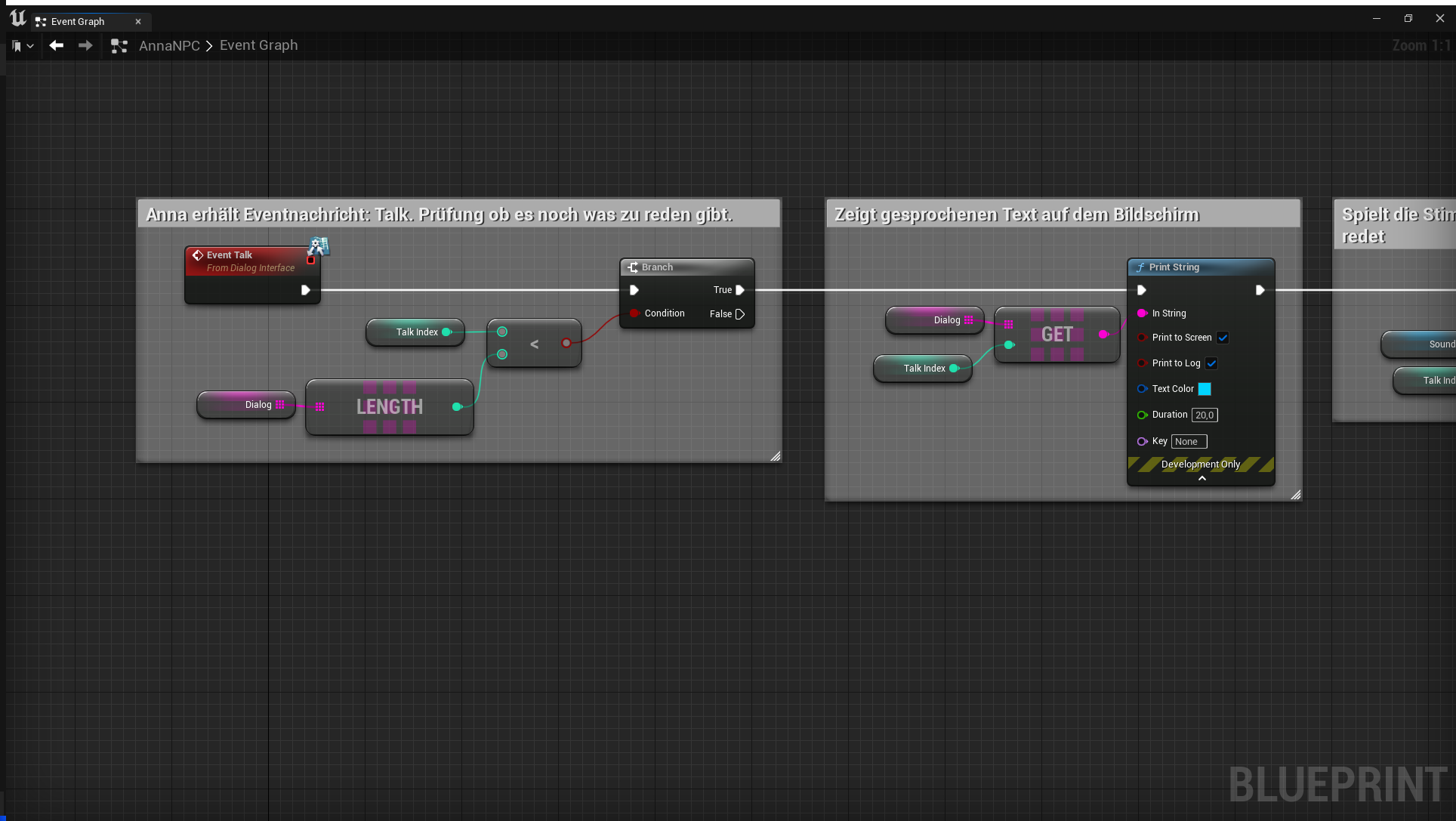
Im nächsten Schritt, wird der Sound von Anna womit Anna antwortet, auf abgespielt.

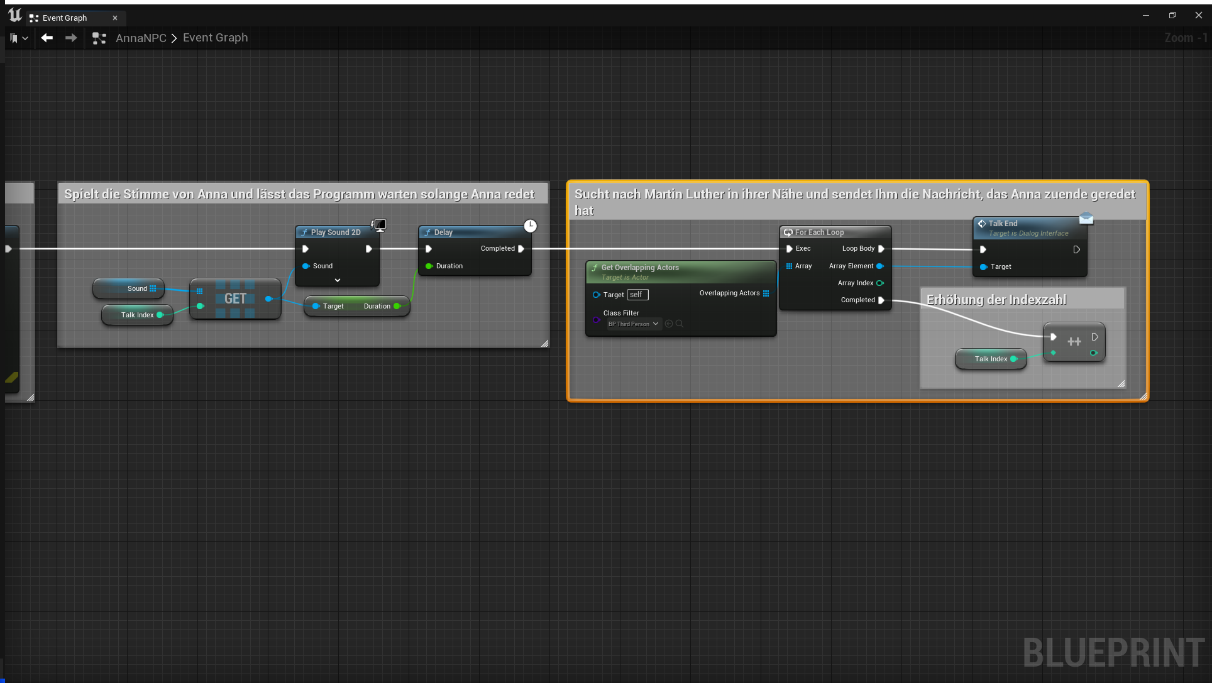
Nach dem die Sounddatei gestartet wird wird ein Delay erzugt, der diie Antwort zu Martin Luther verzögern soll. Die Länge des Delays hat die länge der jeweiligen Sound-Datei aus dem Sound-Arrays.

Nachdem die jeweilige Sound-Datei abgespielt wurde, wird Martin Luther in annas nähe gesucht.

Das Interface Event TalkEnd sendet eine Nachricht an den gefundenen Martin Luther.

Nach dem Finden von Martin Luther, wird TalkIndex 1 hoch addiert.





## Meilenstein: Sprachausgabe

Nachdem ich das Dialogsystem erstellt und implementiert habe, fehlt nur noch der In-

halt, was die Charaktere miteinander reden. Dazu brauche ich Aufnahmen von Stimmen.

Kurz, Martin Luther sowie die beiden NPCs Anna und Johann, brauchen Synchronstim-

men.

Fur das erzeugen der drei Synchronstimmen habe ich mit zwei weiteren KI-Systeme expe-  ̈

rimentiert, und m ̈ochte mein Vorgehen und das Ergebnis in diesem Kapitel pr ̈asentieren.

Voice AI ist ein KI-System, die Stimmen ver ̈andern kann. Zum Beispiel kann man seine

Stimme so manipulieren, das sie wie die von Kanye West oder dem amtierende US-

Pr ̈asident Joe Biden klingt.

Adobe Enhanced ist ein KI-System, das deine gesprochene Stimme so klingen l ̈asst, dass

sie in einem hochwertigen Tonstudio aufgenommen wird.

Zu diesen beiden KI-Systemen, benutze ich mein neun Jahre altes Logitech G35 Headset

um meine Stimme aufzunehmen. Um die Stimmen fur die NPCs zu erzeugen, habe ich  ̈

drei verschiedene Experimente durchgefuhrt.  ̈

• Erst Adobe Enhance Speache dann Voice AI Als erstes wurde der Klang meiner

stimme mit Audacity und meinem Logitech G35 Headset aufgenommen.

• Erst Voice Ai dann Adobe Enhance Speech

• Erst Adobe Enhance Speech dann Voice AI dann wieder Adobe Enhance Speech

# Fazit und Ausblick

Die Umsetzung eines Videospielprototypen hat gezeigt das es möglich ist mit Hilfen von KI-Systemen ein Videospiel zu entwickeln.

Besonders die Tatsache, das mit Hilfe von natürlich formulierten Text, ein 3D-Modell von einer Videospielfigur bekommt, ist eine große Erleichterung als Ein-Mann-Videospielentwickler.

ChatGPT versteht mit seinem Sprachmodell die natürliche Sprache, mit Rechtschreibfehlern, Grammatikfehler und Umgangssprache und verarbeitet sie in einen Midjourney-Prompt.

Dieser von ChatGPT erzeugten Midjourney-Prompts wird von Midjourney zu einem Bild generiert, die Fotorealistische sind.

Diese Fotorealistischen Bilder, die keine echten Fotos sind, wird von PIFuHD verarbeitet, die darauf trainiert ist, echte Fotos von Personen zu 3D-Modellen zu generieren, in ein 3D-Modell zu verwandeln.

Das zeigt wie menschlich diese KI-Systeme sind. Alle drei dieser soeben genannten KI-Systeme besitzen eine gewisse Toleranz, die man in der Regel nicht von Nicht-KI-Systemen kennt, sondern eher von kreativen Personen, die sich damit auskennen.

# Abbildungsverzeichnis

# Tabellenverzeichnis

# Abkürzungsverzeichnis

# Literaturverzeichnis

Adobe. (o.D.). *Was sind 3D-Texturen und wie kann ich sie erstellen?*. Abgerufen am 21. Oktober 2023, von https://www.adobe.com/de/products/ substance3d/discover/how-to-create-3d-textures.html

BM-Experts. (2023, 07. Juni). *Prompt: Was ist das und wie kann er eingesetzt werden?* BM-Experts GmbH, von https://bm-experts.de/definitionenfaq/ definitionen/prompt-was-ist-das-und-wie-kann-er-eingesetzt-werden/

Breuer, K. (2012). *Computerspiele programmieren: Künstliche Intelligenz für künstliche Gehirne*. De Gruyter Oldenbourg.

Bühler, P. (2021). *3D mit Blender: Modeling - Animation – Rendering* (1. Aufl.). Springer Fachmedien Wiesbaden, von https://doi.org/10.1007/978-3-658-36214-0

Bühling, R. (2017, 29. September). *T-Pose*. GameDev-Profi. Abgerufen am 21. Oktober 2023, von https://www.gamedev-profi.de/lexikon/t-pose/

Cambridge Dictionary (o.D.). *Prompt*. Cambridge University Press & Assessment. Abgerufen am 06. Oktober 2023, von <https://dictionary.cambridge.org/dictionary/english/prompt>.

Edge, R. (2013). Fachwerk in Ehingen/Donau.Online-Publikation. https://nbn-resolving.org/urn:nbn:de:101:1-2023092412511020867766

Helfrich-Schkarbanenko, A. (2023). *Mathematik und ChatGPT: Ein Rendezvous am Fuße der technologischen Singularität*. Springer Spektrum. https://doi.org/10.1007/978-3-662-68209-8

ifaa-Institut für angewandte Arbeitswissenschaft. (2023, 14. August). KI-Künstliche Intelligenz und Lernende Systeme. Analysen und Vorhersagen durch maschinelles Lernen verändern die Arbeitswelt 4.0. Arbeitswissenschaft. Abgerufen am 07. Oktober 2023, von <https://www.arbeitswissenschaft.net/themenfelder/kuenstliche-intelligenz-lernende-systeme>

Mai, J. (2019, 20. März 20). *Kreativitätstechniken: Übersicht 20 genialer Tipps & Methoden*. Karrierebibel. Abgerufen am 18. Oktober2023, von https://karrierebibel.de/kreativitaetstechniken/

Midjourney (2023, 04. Juli). Symbol: Midjourney. [Grafik]. Chip. Abgerufen am 25 August 2023, von https://www.chip.de/downloads/webapp-Midjourney-Bilder-KI\_184386674.html

Moring, A. (2023). *Künstliche Intelligenz und Intuition: Robuste und nachhaltige Entscheidungen in digitalen Arbeitswelten* (1. Aufl.). Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-42018-5>

Nimsdorf, S. (2023, 3. April). *Was ist eigentlich … ChatGPT?* Bidt- Bayerisches Forschungsinstitut für Digitale Transformation. Abgerufen am 22. 10. 2023, von https://www.bidt.digital/was-ist-eigentlich-chatgpt/

OpenAI. (2023, 1. Februar). *Introducing ChatGPT plus*. Openai. Abgerufen am 19. Oktober 2023, von https://openai.com/blog/chatgpt-plus

Pyka, M. (o. D.). *Elemente der 3D-Welt*. Tu-chemnitz. Abgerufen am 22. Oktober 2023, von https://www-user.tu-chemnitz.de/~heha/hsn/chm/SelfDXD.chm/directxgraphics/theorie/dg\_ele.html

Riecken, M (2023, März). *Wie Sprachmodelle wie ChatGPT unsere Gesellschaft verändern werden.* Bildungsportal Niedersachsen. Abgerufen am 20.10.2023, von https://bildungsportal-niedersachsen.de/fileadmin/2\_Portale/Medienbildung/Dateien/lernenmit\_ki\_nlq.pdf.

Rost, D. H. (2013). *Handbuch Intelligenz* (1. Aufl.). Beltz. http://d-nb.info/1034153641/04

Schell, J. (2020). *Die Kunst des Game Designs: Bessere Games konzipieren und entwickeln.* (M. Feilen, & K. Lorenzen, Übers.). (3. Aufl.). mitp-Verlag. (Originalwerk veröffentlicht 2020)

Titán Televisor MAX (Pinterest User). o.D. Symbol: Ein-Mann-Videospielentwickler. [Grafik]. Pinterest. Abgerufen am 20. August 2023, von https://www.pinterest.ch/pin/17170042322949698/

Wade McMaster - Creator Impact (05. Mai 2023). *Chat GPT + Midjourney V5.2 = Insane AI Art! (Process to Generate Prompts)*. [Video]. YouTube. Abgerufen am 02. Juli 2023, von https://www.youtube.com/watch?v=0RCvbd5uLDY

Wang, W. (2023). *The Structure of Game Design* (1. Aufl.). International Series on Computer Entertainment and Media Technology. Springer International Publishing.

Zhing'za zï Ïnin (Wikipedia User), OpenAI (Original). (2023, 4. September). Symbol: ChatGPT [Grafik]. Wikipedia. Abgerufen am 10. September 2023, von https://de.m.wikipedia.org/wiki/Datei:ChatGPT\_logo.svg

# Code-Beispiele

# Anhangsverzeichnis

# Erklärung

Ich erkläre, dass ich die vorliegende Bachelor-/Masterarbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

(Unterschrift) Fulda, den …………..

|  |  |  |
| --- | --- | --- |
| Ort, Datum |  | Unterschrift |