



Bachelorarbeit

„INTEGRATION UND VISUALISIERUNG VON AMTLICHEN
GEODATEN DER BAYERISCHEN VERMESSUNGSVERWALTUNG IN
DIE LAUFZEIT- UND ENTWICKLUNGSUMGEBUNG DER UNREAL
ENGINE 5.1 VON EPIC GAMES“



Hochschule für angewandte Wissenschaften München
Fakultät für Geoinformation FKo8

Die Arbeit wurde angefertigt in Kooperation mit:
Landesamt für Digitalisierung, Breitband und Vermessung Bayern (LDBV)

Studiengang:	Kartographie & Geomedientechnik
Semester:	SoSe 2023
Erster Betreuer:	Prof. Dr. Sven Fuhrmann (Hochschule München)
Zweiter Betreuer:	Thomas Meier (LDBV)
Angefertigt von:	Sinan Babernits
Vorgelegt am:	01.08.2023

ZUSAMMENFASSUNG

Diese Bachelorarbeit soll Aufschluss darüber geben, wie aktuelle Soft- und Hardwareentwicklungen eingesetzt werden können, um Visualisierungen mit amtlichen Geodaten in einer Game Engine zu erstellen, ohne ein großes Entwicklerteam zu benötigen.

Die daraus resultierenden Forschungsfragen lauten: Wie lassen sich vorhandene Datensätze des Landesamtes für Digitalisierung, Breitband und Vermessung (LDBV) in die Laufzeit- und Entwicklungsumgebung der Unreal Engine 5.1 von Epic Games integrieren und für eine interaktive Visualisierung aufbereiten? In welchen Anwendungsgebieten können diese Daten interaktiv für das LDBV verwertbar gemacht werden?

Um diese Fragen zu beantworten, wurde ein Workflow erarbeitet, der es ermöglicht, aus den Datensätzen des LDBV ein Stadtmodell zu erstellen und dieses in Unreal Engine zu importieren. Anschließend wurde daraus ein Spiel in der Unreal Engine erstellt, das interaktiv genutzt werden kann.

Das Ergebnis soll zeigen, welche Visualisierungen mit einer Game Engine und den Geodaten der bayerischen Vermessungsverwaltung möglich sind, um diese z. B. auf Messen Bürger:innen präsentieren zu können.

In Zukunft wird es mit weiter voranschreitenden Soft- und Hardwareentwicklungen, die zusätzlich mit KI-gestützten Systemen arbeiten, möglich sein, in einem kleinen Team von Entwicklern aufwändige Anwendungen mit GIS-Funktionalitäten zu entwickeln und die Digitalisierung im Bereich der Geodaten-Visualisierung weiter voranzutreiben.

VORWORT

Während meines Studiums der Kartographie und Geomedientechnik lag mein besonderes Augenmerk auf dem Bereich der Fernerkundung und der 3D-Visualisierung. Mich hat insbesondere das Arbeiten mit 3D-Modellierungssoftware, das Erstellen von Animationen sowie die Visualisierung mittels einer Game Engine sehr interessiert. Mit diesem Bereich bin ich im Modul GMT 5 meines Studiums das erste Mal in Berührung gekommen und konnte Erfahrungen während einer Studienarbeit sammeln.

Während meines Praxissemesters im LDBV unter der Betreuung von Herrn Thomas Meier, der mir vertiefte Einblicke in die Anwendung dieses Themengebietes vermittelte, hat mich die Arbeit mit InfraWorks® und das Erstellen von Animationen mit 3ds Max am meisten fasziniert.

Bei weiteren Recherchen, insbesondere bezüglich der Möglichkeiten, die eine Game Engine wie Unreal Engine von Epic Games bietet, wuchs mein Interesse immer mehr. Aus diesem Grund entschied ich mich dazu, meine Bachelorarbeit diesem Gebiet zu widmen.

DANKSAGUNG

Mit der Fertigstellung meiner Bachelorarbeit möchte ich von ganzen Herzen jenen danken, die mich während meines Studiums und dieser Arbeit unterstützt haben.

Als erstes möchte ich meiner Lebenspartnerin Angela Stanculovic danken, die mich während meines gesamten Studiums und während dieser Arbeit zu jeder Zeit unterstützt hat und so für mich von unschätzbarem Wert war. Mit ihrer Korrekturleistung konnte sie maßgeblich zur Qualität dieser Arbeit beitragen.

Ein herzlicher Dank geht auch an meinen Betreuer der Hochschule München, Herrn Prof. Dr. Sven Fuhrmann. Die professionelle und immer optimistische Begleitung durch den gesamten Prozess waren für mich stets eine Motivation und Orientierung.

Ich danke auch dem Landesamt für Digitalisierung, Breitband und Vermessung für die Unterstützung bei meiner Bachelorarbeit. Mein besonderer Dank gilt hierbei Herrn Thomas Meier. Durch seine Anregungen und sein Fachwissen hat er nicht nur die Richtung dieser Arbeit maßgeblich beeinflusst. Auch durch die Erfahrungen, die ich während meines Praxissemesters im LDBV sammeln konnte, hat Herr Meier einen entscheidenden Einfluss auf den Inhalt dieser Bachelorarbeit genommen.

Ohne die Kombination dieser Unterstützungen hätte mir die Erstellung dieser Arbeit nicht so viel Freude bereitet.

INHALTSVERZEICHNIS

1.	Einleitung	1
1.1	Forschungsfrage	2
1.2	Zielsetzung	3
2.	Literatur.....	4
2.1	Game Engines, Geodaten und Digital Game-Based Learning im Kontext	4
2.1.1	Was ist eine Game Engine und wie ist diese aufgebaut?	4
2.1.2	Geschichtliche Entwicklung von Videospiele und Game Engines	6
2.1.3	Geschichtliche Entwicklung von Unreal Engine.....	7
2.1.4	Geodaten und Game Engines	7
2.1.5	Game-Based Learning	11
2.2	Orientierung und Navigation in virtuellen Umgebungen mit einem Avatar.....	13
2.2.1	Orientierung in virtuellen Umgebungen	13
2.2.2	Navigation in virtuellen Umgebungen	15
2.2.3	Perspektive, Fortbewegung und Interaktion in interaktiven virtuellen Umgebungen.....	18
2.3	Aktuelle Beispiele von Geodaten in virtuellen Umgebungen	21
2.3.1	Beleuchtungssimulation der Stadt Oldenburg	21
2.3.2	Virtual Helsinki.....	22
2.3.3	Project Anywhere	23
3.	Methodenteil	25
3.1	Grundlagen	25
3.1.1	Daten und Datenaufbereitung	26
3.1.2	Software und der Weg in die Entwicklungsumgebung	28
3.2	Arbeitsschritte (Workflow).....	29
3.2.1	InfraWorks® Workflow	29
3.2.2	Twinmotion Workflow	31
3.2.3	Unreal Engine Workflow	32
3.2.4	Unreal Engine Blueprints	39
3.2.4.1	Main Menu	41
3.2.4.2	Options Menu	42
3.2.4.3	Pause Menu	46
3.2.4.4	Level Select.....	47

3.2.4.5	Level	49
3.2.4.6	Third Person Character (Avatar).....	49
3.2.4.7	Drone	52
3.2.4.8	Game Instance	55
3.2.4.9	In & Out Bounds	55
3.2.4.10	MiniMap.....	58
3.2.4.11	Map.....	60
4.	Ergebnisse	62
4.1	InfraWorks®	62
4.2	Twinmotion 2023	63
4.3	Unreal Engine Editor.....	64
4.4	Projekt UI.....	65
4.5	Projekt MiniMap	71
4.6	Projekt Map	72
4.7	Projekt Loadingscreen	73
4.8	Projekt Level.....	78
4.9	Projekt Beispielbilder	84
5.	Diskussion und Fazit	91
5.1	Diskussion.....	91
5.2	Fazit und Ausblick	93
	Literaturverzeichnis.....	96
	Anhang.....	100
	Eidesstattliche Erklärung	102

ABBILDUNGSVERZEICHNIS

Abbildung 1: Vereinfachter Aufbau der Hauptkomponenten einer Game Engine	5
Abbildung 2: Beleuchtungssimulation des Rathausmarkts in Oldenburg	21
Abbildung 3: Alvar Aalto House, Visualisierung in der Unreal Engine	23
Abbildung 4: Senatsplatz Helsinki, Visualisierung in der Unreal Engine	23
Abbildung 5: Project Anywhere, Sicht auf den Globus	24
Abbildung 6: Project Anywhere, XR-Ansicht der 3D-Tiles von CESIUM ion	24
Abbildung 7: QGIS-Auswahl der ATKIS®-Daten.....	26
Abbildung 8: Software	29
Abbildung 9: Workflow InfraWorks®	31
Abbildung 10: Workflow Twinmotion	32
Abbildung 11: Workflow Unreal Engine.....	38
Abbildung 12: Workflow Unreal Engine, detaillierter Ablauf	38
Abbildung 13: Legend Blueprints	39
Abbildung 14: BP & WBP Main Menu	42
Abbildung 15: WBP Options Menu 1	44
Abbildung 16: WBP Options Menu 2	45
Abbildung 17: WBP Options Menu 3.....	45
Abbildung 18: WBP Pause Menu	47
Abbildung 19: WBP Level Select 1.....	48
Abbildung 20: WBP Level Select 2	48
Abbildung 21: BP Level	49
Abbildung 22: BP Third Person Character 1	51
Abbildung 23: BP Third Person Character 2	52
Abbildung 24: BP Drone 1	54
Abbildung 25: BP Drone 2	54
Abbildung 26: BP Game Instance.....	55
Abbildung 27: Bounds Actor im Unreal Engine Level Editor	56
Abbildung 28: Actor BP (BP Bounds)	57
Abbildung 29: Actor BP (Bounds BP) Construction Script	57
Abbildung 30: BP & WBP In & Out Bounds.....	58
Abbildung 31: BP MiniMap.....	59

Abbildung 32: MP MiniMap.....	60
Abbildung 33: WBP Map	61
Abbildung 34: Fertiges InfraWorks®-Modell, bereit für den FBX-Export	62
Abbildung 35: Importiertes Modell, bereit für den Datasmith-Export in Twinmotion.....	63
Abbildung 36: Fertiges Projekt Unreal Engine Editor	64
Abbildung 37: Main Menu	65
Abbildung 38: Options Menu	66
Abbildung 39: Credits	66
Abbildung 40: Pause Menu	67
Abbildung 41: Keys Input	67
Abbildung 42: Level Select	68
Abbildung 43: Levelbeschriftung InfraWorks®-Modell Drohne	68
Abbildung 44: Levelbeschriftung Digitales Orthophoto 2022	69
Abbildung 45: Levelbeschriftung InfraWorks®-Modell Avatar	69
Abbildung 46: In & Out Bounds Timer.....	70
Abbildung 47: MiniMap Avatar.....	71
Abbildung 48: MiniMap Drohne	71
Abbildung 49: Map Avatar	72
Abbildung 50: Map Drohne	72
Abbildung 51: Loadingscreen Digitales Orthophoto 2022	73
Abbildung 52: Loadingscreen Status Symbol.....	73
Abbildung 53: Loadingscreen InfraWorks®-Modell	74
Abbildung 54: Loadingscreen Luftbild Mosaik 1945.....	74
Abbildung 55: Loadingscreen Digitales Orthophoto 1963	75
Abbildung 56: Loadingscreen Webkarte 2023.....	75
Abbildung 57: Loadingscreen Planungskarte 2023.....	76
Abbildung 58: Loadingscreen Uraufnahme 1809	76
Abbildung 59: Loadingscreen Main Menu	77
Abbildung 60: Level Digitales Orthophoto 2022	78
Abbildung 61: Level Luftbild Mosaik 1945.....	79
Abbildung 62: Level Digitales Orthophoto 1963	79
Abbildung 63: Level Webkarte 2023.....	80
Abbildung 64: Level Planungskarte 2023	80

Abbildung 65: Level Uraufnahme 1809	81
Abbildung 66: Level InfraWorks®-Modell zum Vergleich	82
Abbildung 67: Level Digitales Orthophoto 2022 zum Vergleich	82
Abbildung 68: Level Digitales Orthophoto 2022 mit Avatar.....	83
Abbildung 69: Level Webkarte 2023 mit Avatar.....	83
Abbildung 70: Projekt Beispielbild 1.....	84
Abbildung 71: Projekt Beispielbild 2	84
Abbildung 72: Projekt Beispielbild 3.....	85
Abbildung 73: Projekt Beispielbild 4.....	85
Abbildung 74: Projekt Beispielbild 5.....	86
Abbildung 75: Projekt Beispielbild 6.....	86
Abbildung 76: Projekt Beispielbild 7	87
Abbildung 77: Projekt Beispielbild 8	87
Abbildung 78: Projekt Beispielbild 9.....	88
Abbildung 79: Projekt Beispielbild 10	88
Abbildung 80: Projekt Beispielbild 11.....	89
Abbildung 81: Projekt Beispielbild 12	89
Abbildung 82: Projekt Beispielbild 13.....	90
Abbildung 83: Projekt Beispielbild 14.....	90

ABKÜRZUNGSVERZEICHNIS

ALKIS®	<i>Amtliches Liegenschaftskatasterinformationssystem</i>
AR	<i>Augmented Reality</i>
ATKIS®	<i>Amtliches Topographisch-Kartographisches Informationssystem</i>
BIM	<i>Building Information Modeling</i>
BP	<i>Blueprint</i>
CAD	<i>Computer Aided Design</i>
DGM	<i>Digitales Geländemodell</i>
DLM	<i>Digitales Landschaftsmodell</i>
DOM	<i>Digitales Oberflächenmodell</i>
DOP	<i>Digitales Orthophoto</i>
DOPs	<i>Digitale Orthophotos</i>
FPS	<i>Frames per Second</i>
GIS	<i>Geographisches Informationssystem</i>
GPS	<i>Global Positioning System</i>
HUD	<i>Head-up-Display</i>
KI	<i>künstliche Intelligenz</i>
LoD	<i>Level of Detail</i>
MP	<i>Material Parameter</i>
SDK	<i>Software Development Kit</i>
UI	<i>User Interface</i>
VR	<i>Virtual Reality</i>
WBP	<i>Widget Blueprint</i>

1. EINLEITUNG

Ein interaktiv begehbare 3D-Stadtmodell oder Landschaftsmodell, das in einer Game Engine nutzbar gemacht wird, kann zukünftig die Visualisierung von Geodaten auf eine neue Ebene heben.

Im Kontext dieser wissenschaftlichen Arbeit soll eine Differenzierung von einfachen 3D-Stadtmodellen, dem digitalen Zwilling einer Stadt und einem interaktiv begehbaren 3D-Stadtmodell stattfinden, wobei sich diese Arbeit auf die interaktive Nutzung konzentrieren wird.

In klassischen 3D-Stadtmodellen ist es meist nur vorgesehen, eine Stadt oder Landschaft aus einer betrachtenden Perspektive heraus zu erkunden, so wie es z. B. Google Earth bietet, das heutzutage fast jede Person kennt und nutzt. Auch in der Stadtplanung oder bei Forschungsfragen in Bezug auf das Klima (wie z. B. Hochwassersimulationen), für die keine Echtzeit-Sensordaten benötigt werden, kann man 3D-Stadtmodelle nutzen. Die Einsatzgebiete sind heutzutage sehr vielfältig. Bei der Stadtplanung beispielsweise ermöglichen sie es, die Partizipation von Bürger:innen zu erhöhen und erleichtern somit die Entscheidungsfindung. Aber auch die Immobilienbranche und der Tourismus sind wichtige Einsatzgebiete dieser Modelle. Das LDBV verwaltet ein 3D-Landschaftsmodell im BayernAtlas, das auf Grundlage eines Digitalen Geländemodells (DGM) und mit Level of Detail (LoD) 2-Gebäuden erstellt wurde. Auch wenn es sich hierbei nur um eine vereinfachte Form eines 3D-Modells von Bayern handelt, ermöglicht es besonders in Gebieten mit großen Höhenunterschieden aufgrund der 3D-Visualisierung eine bessere Orientierung.

Der digitale Zwilling einer Stadt hingegen wird noch durch Fachinformationen erweitert, wie beispielsweise Daten aus den Stadtverwaltungen, den Stadtwerken oder auch den Verkehrsgesellschaften. Die wichtigste Rolle übernehmen jedoch Echtzeit-Sensordaten einer Stadt, wie z. B. Verkehrsmengen, Wetterdaten, Lärmmessungen sowie Luftschadstoffmessungen. Diese Daten sind für die Stadtverwaltung von größtem Interesse, insbesondere für die Planung der urbanen Mobilität oder für die Partizipation von Bürger:innenentscheidungen bei der Stadtplanung. Auch für Simulationen im Bereich Verkehr und Klimaschutz, wie z. B. die Berechnung von Luftwerten für eine zukunftstaugliche Verkehrsplanung, sind die Daten essenziell. Gerade in der Stadtplanung können Simulationen die klimatischen Auswirkungen von Verkehr und geplanten Bauvorhaben veranschaulichen und den Bürger:innen näherbringen. Das digitale Abbild der Stadt ist für Analysen, Simulationen und Was-Wäre-Wenn-Szenarien ausgelegt und ermöglicht es der Stadtverwaltung, Planungen in einer Stadt derart miteinander zu verbinden, wie es sonst kaum möglich ist. Erst anhand eines digitalen Zwillings und dessen vielseitigen Möglichkeiten kann eine Vielzahl von Bauvorhaben zusammengeführt werden. (IT Referat Landeshauptstadt München, n.d.) Auch in den Bereichen Industrie und Wissenschaft spielen digitale Zwillinge bei Planungen oder in Klimafragen eine wichtige Rolle. Ein digitaler Zwilling

bringt jedoch ein hohes Maß an Aufwand mit sich und ist dadurch für kleinere Projekte nicht sonderlich geeignet. Die kontinuierliche Datenerhebung für diese komplexen Modelle geht mit einem großen Verwaltungsaufwand und rechtlichen Anforderungen aus Datenschutzgründen einher. Eine interaktive Nutzung wie in einem klassischen Videospiel, in dem man das Modell begehen könnte, ist nicht vorgesehen.

Die interaktive Nutzung eines begehbaren 3D-Stadtmodells ist das Thema dieser Arbeit. So können Nutzer:innen auf eine spielerische Weise in ein Produkt eingebunden werden. Das visuelle Erlebnis und die Immersion, die es einem ermöglicht, wie in einem Videospiel in eine virtuelle Welt einzutauchen, können Anwender:innen spielerisch Informationen vermitteln. Auch der Lerneffekt wird durch solch eine Anwendung verstärkt. Wir lernen am einfachsten und effektivsten auf spielerische Art und Weise und die Immersion wie in einem Videospiel verstärkt diesen Effekt zusätzlich. Game-Based Learning ist ein wichtiger Aspekt, den wir in unserer Gesellschaft noch zu wenig wahrnehmen. Würden wir ebenso viel Zeit in die Entwicklung von Spielen stecken, die Wissen vermitteln, wie die Spieleindustrie heutzutage in normale Computerspiele investiert, könnten wir wieder zu einer natürlichen Lernmethode zurückkehren und auf eine spielerische Art und Weise lernen. (Schutz & Schwarz, 2022)

In der Zukunft werden Geodaten in Game Engines, aus denen seriöse Spiele entwickelt werden, in vielen Bereichen genutzt und werden einen wichtigen Beitrag zur Digitalisierung leisten. Die Nutzungsmöglichkeiten werden sich hierbei auf ein breites Spektrum an Game-Based Learning, Erhöhung von Partizipation bei Bürger:innenentscheidungen, politischen Entscheidungen, Klimafragen oder einfachen Bauvorhaben erstrecken.

Neueste Entwicklungen ermöglichen es, die meist fiktiven Game Maps mit realen Daten aus unserer Welt zu verknüpfen. Diese Verbindung besteht im Grunde genommen seit jeher, da fiktive Landschaften die gleiche Topologie und Topographie wie die reale Welt aufweisen. Durch den zunehmenden Realismus, den Videospiele heutzutage bieten, wird diese Überschneidung immer deutlicher, wie beispielsweise in den Spielen Microsoft Flight Simulator oder Aerosoft CityDriver.

1.1 Forschungsfrage

In dieser wissenschaftlichen Arbeit soll die Frage geklärt werden, wie die Daten des LDBV in Unreal Engine 5.1 von Epic Games integriert und interaktiv nutzbar gemacht werden können. Hierfür soll ein Workflow entwickelt werden, der es ermöglicht, Geodaten in die Engine zu importieren und darin eine interaktive Anwendung zu erstellen. Für zukünftige Projekte am LDBV sollen für das Gameplay grundsätzliche Funktionen vorbereitet werden, die als Vorlage mit kleineren Anpassungen in Projekte immigriert werden können und für Präsentationszwecke vom LDBV weiterverwendet werden können. Die Anwendung kann dann auf dieser Grundlage noch mit weiteren themenspezifischen Funktionen ergänzt werden. Die vorliegende Arbeit soll eine

Möglichkeit aufzeigen, wie die Daten des LDBV in einer Game Engine genutzt werden können und wie Game Engines zur Digitalisierung beitragen können.

Daraus ergibt sich die Frage bezüglich der korrekten Vorbereitung der Daten für den Import und welche Softwarelösungen für die Umsetzung der Anwendung verfügbar sind und für diesen Zweck nutzbar gemacht werden können. Zusätzlich soll ein kompletter Workflow dazu entstehen, welche Arbeitsschritte in den jeweiligen Softwarelösungen und Plugins durchgeführt werden müssen, um die Daten optimal vorbereiten zu können und um ein interaktives Videospiel daraus zu erstellen.

1.2 Zielsetzung

Die Zielsetzung dieser Arbeit besteht darin, mit modernster Technik einen Workflow für die Visualisierung amtlicher Geodaten zu erarbeiten, um diese den Bürger:innen näher bringen zu können. Die Geodaten sollen Menschen aller Altersgruppen möglichst barrierefrei zugänglich gemacht werden. Das Interesse an Geodaten soll geweckt und die Bedeutung dieser bewusster gemacht werden. Dadurch kann das Aufgabengebiet der bayerischen Vermessungsverwaltung aufgezeigt und gleichzeitig auch weitere Wege der Digitalisierung im LDBV vorangetrieben werden. Der Einsatzzweck dieser Arbeit im LDBV besteht dann im Allgemeinen darin, Geodaten auf Messen und Präsentationen vorzuführen und diese den Nutzer:innen aller Altersgruppen spielerisch näher bringen zu können.

2. LITERATUR

2.1 Game Engines, Geodaten und Digital Game-Based Learning im Kontext

Seit einigen Jahren ist die Nutzung von Geodaten mit Game Engines ein viel diskutiertes Thema in der Geovisualisierung. Von besonderer Bedeutung sind hierbei aufgrund ihrer Aktualität und Qualität amtlich geführte Geodaten. Zusätzlich hat das amtliche Geoinformationswesen durch die EU-Richtlinie „Infrastructure for Spatial Information in the European Community“, kurz Inspire, eine neue Ausrichtung erhalten, die eine gemeinsame Geodateninfrastruktur schaffen soll und die Daten als Open Data zum Download anbietet, um die Umweltpolitik zu unterstützen. (Dr. Dennis Edler, 2019) Durch diese Zugänglichkeit zu Geodaten für die Öffentlichkeit wird ein wichtiger Grundstein gelegt, um in Zukunft eine breite Palette an Projekten, in denen die virtuelle Realität eine Rolle spielt, mit einheitlichen Geodaten zu versorgen. Es wird sogar Hobby-Entwicklern ermöglicht, auf Grundlage dieser Daten Visualisierungen zu erstellen.

2.1.1 Was ist eine Game Engine und wie ist diese aufgebaut?

Eine Game Engine ist ein spezielles Framework zur Entwicklung von Spielen. Es steuert den Ablauf eines Spiels und fungiert, einfach gesagt, als Herzstück oder Motor, der alle Komponenten eines Spiels miteinander verbindet. Der Begriff Engine beschreibt hierbei metaphorisch die Rolle einer Laufzeit- und Entwicklungsumgebung, die eine zentrale, treibende Softwarekomponente darstellt, welche für die Entwicklung und Ausführung von Computerspielen notwendig ist. Game Engines sind heutzutage wesentlich mehr als nur ein antreibender Motor. Sie stellen gleichzeitig auch ein ganzes Autorensystem zur Verfügung, das nicht nur die einzelnen Komponenten miteinander verbindet, sondern es auch erlaubt, mit Hilfe eines Editors alle Elemente eines Spiels zu erstellen. (Christian Wasser, 2019; Marc Ottensmann, 2016)

Vereinfacht ausgedrückt besteht eine moderne Game Engine aus den Komponenten Rendering, Physik, Künstliche Intelligenz (KI), Audio, Ein- & Ausgabeverwaltung, Netzwerkfunktionalitäten, Skripte & Logik, Dateisystem, Benutzeroberfläche und Werkzeuge & Editor, siehe Abbildung 1.

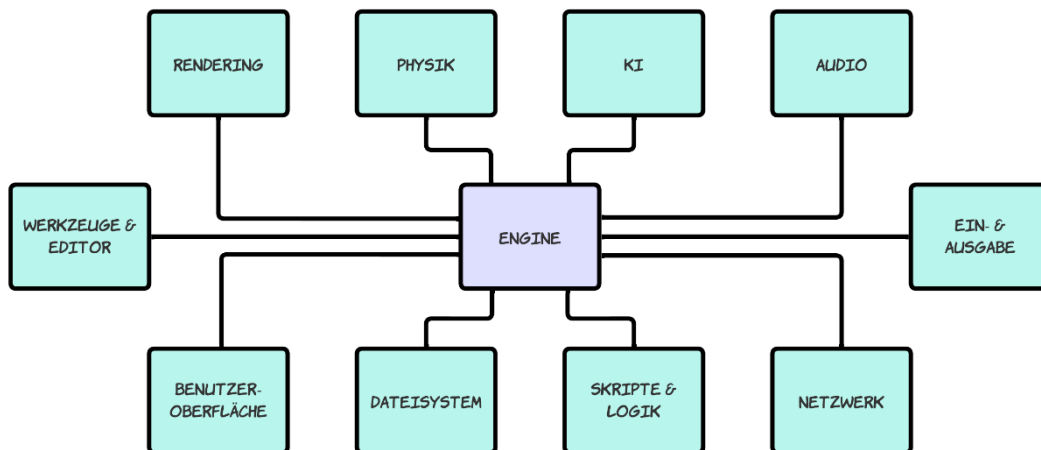


Abbildung 1: Vereinfachter Aufbau der Hauptkomponenten einer Game Engine

Rendering: Grafische Darstellung von Objekten, Texturen, Beleuchtung und Schatten

Physik: Physikalische Abläufe im Spiel, wie Kollisionen (wenn ein Objekt auf den Boden fällt oder ein Character gegen eine Wand läuft) oder Bewegungsabläufe (Fahrphysik von Fahrzeugen, Fallen und Bewegungen)

KI: Verhalten von z. B. Gegnern im Spiel, Entscheidungen und Reaktionen auf Aktionen des Spielers

Audio: Wiedergabe von Soundeffekten, Hintergrundmusik und Sprache, auch im Kontext mit der Umgebung oder Richtung

Ein- & Ausgabe: Steuerung durch Controller, Tastatur und Maus, Virtual Reality (VR)-Headset Tracking oder Lenkrad usw.

Netzwerk: Kommunikation über Netzwerke (meist Internet) z. B. bei Multiplayern

Skripte & Logik: Verhalten, Regeln sowie Abläufe im Spiel über Skripte oder Programmcode

Dateisystem: Laden und Verwalten von Ressourcen wie Texturen, 3D-Modellen, Audiodateien und anderen Daten, die für das Spiel benötigt werden

Benutzeroberfläche: Menüs und Head-up-Display (HUD), wie MiniMap, oder Informationstexte während des Spielablaufes

Werkzeuge & Editor: Verwaltung von Inhalten des Spieles (Autorensystem) und die Schnittstelle moderner Game Engines zur Erstellung des Spiels mit einer geeigneten Benutzeroberfläche, meist auch erweiterbar durch Plugins von Drittanbietern

Diese Auflistung stellt nur einen sehr vereinfachten Aufbau einer Game Engine dar. Es gilt zu beachten, dass die Komponenten je nach verwendeter Engine unterschiedlich

sein können und der Schwerpunkt und die damit einhergehenden Möglichkeiten einer Engine variieren. Gute Entwickler können mit zwei unterschiedlichen Engines durch Anpassungen dasselbe Ergebnis erzielen. Viele Spielestudios setzen auch ihre eigenen Engines zur Erstellung ihrer Spiele ein.

2.1.2 Geschichtliche Entwicklung von Videospielen und Game Engines

Für den Tag der offenen Tür des Kernforschungszentrums am Brookhaven National Laboratory in den Vereinigten Staaten entwickelte der amerikanische Physiker William Higinbotham 1958 das Videospiel „Tennis for Two“. Es kann somit als das erste der Unterhaltung dienende Videospiel betrachtet werden und war der Vorläufer des ersten kommerziell erfolgreichen Automatenspiels „Pong“ von Nolan Bushnell aus dem Jahr 1972. Im selben Jahr erschien auch die erste Heimkonsole Odyssey der Firma Magnavox. Nolan Bushnell konnte ab 1977 mit der Atari 2600 ebenfalls in den wachsenden Markt für Heimkonsolen einsteigen und an seinen Erfolg von 1972 mit „Pong“ anknüpfen. Ein wichtiger Aspekt hierbei war, dass die Atari-Konsole Hard- und Software voneinander trennte, wie es auch schon bei der Odyssey der Fall war. Dadurch boten diese Konsolen die Möglichkeit, die Spiele zu wechseln. Bei den Automatenspielen waren Konsole und Spiel noch fest miteinander verknüpft. In den 80er-Jahren veränderte sich der Markt dahingehend, dass Spieler vor allem aufgrund der Erscheinung des Commodore 64 im Jahr 1982 zunehmend an ihrem PC spielten. (Funk, n.d.; Videospiehalbwissen, n.d.) Bis Anfang der 90er-Jahre wurde jedes Spiel von Grund auf neu entwickelt, ohne eine Game Engine im modernen Sinne zu nutzen. Spieleentwickler verwendeten bis zu diesem Zeitpunkt, wenn möglich, Teile und Codeausschnitte aus Spielen ihrer früheren Entwicklung. (Christian Wasser, 2019)

Erst in den 90er-Jahren begann der Begriff Game Engine populär zu werden. Mit dem Spiel „Wolfenstein 3D“, das im Jahr 1992 erschien, legte id Software den Grundstein für das im Jahr 1993 erschienene Spiel „Doom“. Dieses basierte dann zum ersten Mal auf einer Game Engine im modernen Sinne. Die darauffolgenden Spiele von id Software „Quake 1-3“ waren in einer ähnlichen Struktur wie Doom aufgebaut. Somit musste id Software nicht bei jedem Spiel bei null anfangen, da die Spiele basierend auf derselben Struktur (Engine) des vorherigen Spiels entwickelt wurden. Der Mehrwert dieses Konzeptes war nicht nur für die Entwickler selbst bei der Schaffung neuer Spiele von Vorteil. Zusätzlich erkannte id Software, dass durch die Vergabe von Lizenzen an ihrer Engine Geld verdient werden konnte. Somit war der Grundstein für moderne Game Engines wie Unreal Engine von Epic Games oder Unity von Unity Technologys gelegt, dem heutigen Industriestandard u. a. in der Filmindustrie, in Werbung und Marketing, in Architekturvisualisierungen, in Simulationen und Trainings, in Virtual Reality und Augmented Reality, in Echtzeitvisualisierungen und in wissenschaftlicher Forschung. (Balasubramanian, 2022; Marc Ottensmann, 2016)

2.1.3 Geschichtliche Entwicklung von Unreal Engine

Epic MegaGames (ab 1999 Epic Games) veränderte 1998 mit dem Release des Spiels „Unreal“ die Spieleindustrie nachhaltig. Das Spiel war bekannt für seine fortschrittliche Rendering-Technologie und die Fähigkeit, große, offene Welten darzustellen. Gleichzeitig hatte es eine eigene Skriptsprache (UnrealScript). Schon vor dem Erscheinen des Spiels fragten Entwicklerstudios bei Epic MegaGames nach Lizenzen für die gleichnamige Unreal Engine. Die Unreal Engine wurde zu diesem Zeitpunkt gleichzeitig auch interessant für Forschungsprojekte und das Militär.

Mit Unreal Engine 2 gelang es dann zum ersten Mal, eine für PC-Spiele konzipierte Engine auf Spielekonsolen zu exportieren. Obwohl die Spiele „Unreal“ und „Unreal Tournament 2004“ ein voller Erfolg waren, ist die Lizenzierung der Engine für andere Spielestudios bis heute eine der größten Einnahmequellen für Epic Games. Mit Unreal Engine 5, die 2020 vorgestellt wurde und die derzeit aktuelle Version der Engine darstellt, hat Epic Games mit den Rendering-Technologien Lumen und Nanite wieder einen neuen Standard in der Spieleindustrie gesetzt. Unreal Engine 5 ermöglicht es, Prototypen von Spielen mit enorm verkürzten Arbeitszeiten zu erstellen. Epic Games konnte mit Unreal Engine 5 die Qualität von Echtzeit-Visualisierungen auf ein fotorealistisches Niveau heben.

Aufgrund des komplett freien Zugangs im nicht kommerziellen Bereich ist sie zudem ein wertvolles Werkzeug für Indie- und Hobby-Entwickler sowie für die Forschung und im Medienbereich. Erst bei einem kommerziellen Erfolg verlangt Epic Games eine Lizenzgebühr. Dieser Ansatz macht die Unreal Engine besonders attraktiv. (Diedrich, 2023; Marc Ottensmann, 2016; Olaf Bleich et al., 2019)

2.1.4 Geodaten und Game Engines

Für die Visualisierung von Geodaten in Game Engines eignen sich besonders Unreal Engine und Unity. Sie bieten für die Arbeit mit Geodaten Plugins und eine Community mit Dokumentationen. Um mit Geodaten in den Engines arbeiten zu können, stehen Plugins für Softwarelösungen von großen Unternehmen wie Esri, Autodesk, mapbox, Cesium und Blender bereit. Epic Games bietet mittlerweile auch ein Plugin für Unreal Engine, das ein geodätisches Koordinatensystem bereitstellt. Zusätzlich wird von Epic Games die Software Twinmotion bereitgestellt, mit der Computer Aided Design (CAD)- und Building Information Modeling (BIM)-Daten für die Visualisierung aufbereitet und in Unreal Engine importiert werden können. Im Epic Store finden sich noch weitere Drittanbieter-Plugins von kleineren Entwicklern, die es ermöglichen, in der Engine mit Geodaten zu arbeiten. Besonders interessant ist das Plugin Landscaping von Ludic Drive.

Für den Import von Geodaten in Game Engines gibt es mehrere Möglichkeiten und diese folgen drei verschiedenen Ansätzen. In allen drei Fällen ist das Arbeiten mit Drittanbieter-Plugins für Unreal Engine nötig. Eine Ausnahme stellt hierbei nur der Import von DGMs dar, die auch direkt importiert werden können. Die Engines bieten

ohne Plugins nur ein kartesisches Koordinatensystem zur Darstellung von räumlichen Daten. Das direkte Arbeiten mit einem geodätischen Koordinatensystem ist ohne Plugins nicht möglich. Diese Plugins ermöglichen es dann, Geodaten in die jeweiligen Engines zu importieren.

Die erste Möglichkeit besteht darin, ein Modell in einer anderen Softwarelösung wie InfraWorks®, Blender oder in CAD-Programmen vorzubereiten und es ohne ein geodätisches Referenzsystem über Plugins oder weitere Software wie Twinmotion zu importieren. Bei der zweiten Möglichkeit werden Geodaten mit Hilfe eines Plugins über Bibliotheken von bestehenden Unternehmen, darunter Esri, Cesium oder mapbox, importiert. Die Plugins von Esri sowie Cesium stammen direkt von den Herstellern und bieten auch ein geodätisches Koordinatensystem in der Engine selbst. Die dritte Möglichkeit sind Plugins, die das direkte Importieren von Geodaten in die Engine erlauben. Hier ist das Landscaping Plugin von Ludic Drive aus dem Epic Marketplace für die Unreal Engine besonders interessant, da es auch die Möglichkeit bietet, Vektordaten, wie z. B. Straßen, zu importieren und diese in ein 3D-Modell umzuwandeln. Dies wäre hierbei die eleganteste Lösung, da man direkt in der Engine arbeiten kann, ohne dabei auf weitere Softwarelösungen angewiesen zu sein.

Jede dieser Möglichkeiten hat ihre Vor- und Nachteile. In der folgenden Übersicht soll gezeigt werden, wo die Anwendungsbereiche der jeweiligen Lösungen liegen.

Esri ArcGIS Maps Software Development Kit (SDK) Plugin

Mit dem ArcGIS Maps SDK sind richtige Geographisches Informationssystem (GIS)-Funktionalitäten für die 3D-Visualisierung in einer Engine möglich. Auch kann man so mit einer Vielzahl von Geodaten von Esri selbst oder auch mit den eigenen Geodaten arbeiten. Das Plugin bietet gleichzeitig schon Vorlagencodes (Frameworks) für die Engine selbst, um z. B. Routen zwischen zwei Punkten zu berechnen oder Messungen im Modell vorzunehmen. (*ArcGIS Maps SDK for Unreal Engine | ArcGIS Developers*, n.d.)

Gut geeignet, wenn richtige GIS-Funktionalitäten zur Analyse von Geodaten das Ziel sind. Für Unity und Unreal Engine. Hohe Kosten für Lizenzgebühren.

Cesium Plugin

Cesium erlaubt das Streamen von globalen 3D-Tiles, die mit einem photogrammetrischen Verfahren erstellt wurden und macht so interaktiv den ganzen Globus begehbar. Es ist aber begrenzt auf die Datenformate, die Cesium selbst anbietet und bietet auch keine GIS-Funktionalität wie das ArcGIS Maps SDK Plugin. Mit dem kostenpflichtigen Cesium ion kann zusätzlich auf weiteren Content bestehend aus hochwertigen 3D-Städten, Satellitenbildern oder Gebäudemodellen, die teilweise begehbar sind, zugegriffen werden. (*Cesium for Unreal*, n.d.)

Anwendung für die Visualisierung von Globen und Realisierung von Projekten wie Google Earth oder Flugsimulatoren. Durch das photogrammetrische Verfahren ist die

Qualität nicht auf dem Level wie bei ausmodellierten Daten. Für Unity und Unreal Engine. Kosten nur für Cesium ion Produkte.

Ludic Drive Landscaping Plugin

Der Vorteil des Landscaping Plugins liegt darin, dass man direkt in der Engine arbeiten kann und sich der Workflow auf die Arbeit in der Unreal Engine selbst beschränkt. Auch Vektordaten wie Straßen und Flüsse können importiert werden. Diese Daten können dann zu 3D-Straßenmodellen konvertiert oder als Flüsse in der Engine dargestellt werden. Dies ermöglicht es, bei Erstellung von geeigneten 3D-Straßenmodellen eine visuell ansprechende Darstellung zu erreichen, die durch photogrammetrische Modelle heute noch nicht erreicht werden kann. Auch können Modelle aus photogrammetrischen Verfahren von Nachteil für die Performance sein. (*Landscaping in Code Plugins - UE Marketplace*, n.d.; Ludic Drive, n.d.)

Visualisierung direkt in der Unreal Engine mit Geodaten möglich. Geringe Kosten für das Plugin.

InfraWorks®

InfraWorks® ist eine Entwurfsplanungssoftware mit GIS-Funktionalitäten, deren Stärke in der Verarbeitung von Vektordaten wie Verkehrswegen und Nutzungsflächen liegt. InfraWorks® ist unter anderem auch für die Straßenplanung ausgelegt und ermöglicht es z. B., Vektordaten von Verkehrswegen zu einem 3D-Modell zu konvertieren und diese in das DGM zu integrieren. Somit kann man mit InfraWorks® ein hochwertiges 3D-Modell erstellen. Der Nachteil hierbei liegt darin, dass das Landscape aus einem Mesh besteht, das die Performance der Engine verringert. Auch können die nativen Landscape-Funktionen der Engine nicht genutzt werden. Der Workflow muss über mehrere Softwarelösungen und Dateiformatkonvertierungen laufen, um das fertige Modell in eine Engine importieren zu können. (*InfraWorks: Modellieren Sie Gebäude- Und Infrastrukturentwurfskonzepte Im Realen Kontext*, n.d.)

Hochwertige Visualisierungen möglich und realistische 3D-Modelle von Städten und Landschaftsausschnitten. Gut geeignet, um ein Stadt- oder Landschaftsmodell aus Geodaten für eine Engine vorzubereiten. Für ein Autodesk-Produkt geringe Kosten.

Twinmotion 2023

Twinmotion ist eine Echtzeit-3D-Visualisierungssoftware von Epic Games und wurde für Architektur, Städte- und Landschaftsplanung und das Bauwesen konzipiert. Es wird auch in der Automobilbranche, im Modedesign oder im allgemeinen Produktdesign eingesetzt und nutzt die Unreal Engine zur Visualisierung der Daten. Der Vorteil von Twinmotion besteht darin, dass es erlaubt, Modelle aus einer Vielzahl an CAD- und BIM-Softwarelösungen zu importieren. Da es auch von Epic Games bereitgestellt wird, eignet es sich als optimaler Zwischenschritt vor dem Import eines Modells in die Unreal Engine. Twinmotion kann auch digitale Geländemodelle als Landscape importieren. Da es sich um eine reine Visualisierungssoftware handelt, sind

keine interaktiven Anwendungen möglich. (*Twinmotion Is a Cutting-Edge Real-Time Visualization Tool*, n.d.)

Hochwertige Visualisierungen möglich und optimal zum Vorbereiten eines 3D-Modells für den Import in Unreal Engine. Es kann auch DGMs als Landscape importieren. Kosten für die Nutzung im vollen Umfang.

mapbox Plugin

Mit dem mapbox Plugin können die Geodaten von den mapbox-Servern als Landscape in die Unreal Engine importiert werden. Mit Hilfe des Unreal Engine Plugins, das ein geodätisches Koordinatensystem bereitstellt, ist auch eine Georeferenzierung möglich. (delebash, n.d.)

Nur für den Import von Landscapes in die Unreal Engine. Eventuell Kosten für die Nutzung von mapbox.

Blender

Für Blender gibt es auch ein GIS Plugin, das es ermöglicht, direkt in Blender mit Geodaten zu arbeiten. Eignet sich, wenn man viel Modellierungsarbeit an einem Stadt- oder Landschaftsmodell vornehmen möchte. Workflow muss über mehrere Softwarelösungen und Dateiformatkonvertierungen laufen, um das fertige Modell in eine Engine importieren zu können. (Domlysz, n.d.)

Geeignet, wenn man Modellierungsarbeiten am Modell vornehmen möchte. Je nach Workflow für alle Engines geeignet. Keine Kosten, da Open Source.

Sonstige Autodesk-Lösungen

Für Bauplanungen, Kartenerstellung, 3D-Scans, Modellierungen und Gebäude- und Infrastruktur-Planungen werden auch weitere CAD- und BIM-Lösungen für die Visualisierung eingesetzt, die mit Geodaten arbeiten können. Dazu zählen z. B. Civil 3D®, Map 3D®, Maya®, Recap Pro®, 3ds Max® und Revit®. Einige können nativ mit GIS-Daten umgehen, andere benötigen Plugins oder einen speziellen Workflow. Der Workflow muss über mehrere Softwarelösungen und Dateiformatkonvertierungen laufen, um das fertige Modell in eine Engine importieren zu können. Mit Hilfe des Datasmith Plugins kann das exportierte Modell aber auch direkt in Unreal Engine integriert werden. (*Autodesk Gibt Innovatoren Überall Die Möglichkeiten, Neues Wahr Werden Zu Lassen*, n.d.)

Nur für die Vorbereitung und bei spezialisierten Projekten. Eventuell auch für zusätzliche Vorbereitungen von 3D-Modellen. Je nach Workflow für alle Engines geeignet. Hohe Kosten durch Lizenzgebühren.

2.1.5 Game-Based Learning

Die Möglichkeiten, die eine Game Engine beim Lernen eröffnet, werden in Zukunft einen hohen Stellenwert in der Bildung einnehmen. Wir sind heute an einem Punkt angelangt, an dem es für Lehrende in konventionellen Bildungsinstitutionen immer schwieriger wird, Schüler:innen angesichts der Flut an Informationen des digitalen Zeitalters etwas beizubringen. Auch könnten Lernspiele, die nur einmal erstellt werden müssen, für die Bildungspolitik der Lösungsansatz für den Lehrer:innenmangel im 21. Jahrhundert sein. (Schutz & Schwarz, 2022, S. 25)

Wir haben in unserer Gesellschaft leider noch nicht vollständig erkannt, was digitale Lösungen für uns leisten können. Das Lernen ist durch die beschleunigte Flut an Informationen im digitalen Zeitalter des 21. Jahrhunderts immer schwieriger geworden, da wir Fakten ohne genügende Wiederholungen nur flüchtig in unserem Gehirn abspeichern können. Auch sehen wir nicht nur mit unseren Augen, sondern auch mit unserem Gedächtnis und die Wahrnehmung daraus ist eine Konstruktionsleistung unseres Gehirns. Dadurch blenden wir auch das aus, was wir nicht sehen wollen und erkennen z. B. nicht, wie uns die Digitalisierung helfen kann. In Deutschland ist die Digitalisierung im Gesundheitswesen oder in der Verwaltung in den letzten Jahrzehnten dadurch nur langsam fortgeschritten. Umso wichtiger ist es für die Zukunft, die Möglichkeiten, die uns die Digitalisierung bietet, zu nutzen und die dadurch entstandenen Vorteile einzusetzen. Computerspiele im Bildungssektor eröffnen uns vielfältige Alternativen, um künftig besser mit der wachsenden Flut an Informationen umgehen zu können. (Schutz & Schwarz, 2022, S. 4)

Um die Vorteile des Game-Based Learning richtig zu verstehen, ist es wichtig zu wissen, wie wir lernen und wie uns digitale Spiele hierbei helfen können. Die Lerninhalte in heutigen Bildungseinrichtungen sind oft so aufgebaut, dass wir etwas auswendig lernen und es nach der Prüfung wieder vergessen. Bevor es die heutigen Bildungseinrichtungen gab, hat der Mensch spielerisch gelernt. Game-Based Learning kann das Lernen wieder in ein spielerisches Erlebnis verwandeln. (Schutz & Schwarz, 2022, S. 19)

Angesichts der Informationsflut des 21. Jahrhunderts ist das selbstorganisierte Lernen der Schlüssel zum richtigen Lernen. Hierbei gilt es freiwillig, mit Freude und voller Energie zu lernen. Unser Gehirn ist so aufgebaut, dass es uns belohnt, wenn wir lernen. Wir sind nicht glücklich, weil wir lernen. Wir sind glücklich, damit wir lernen. (Schutz & Schwarz, 2022, S. 8) Herausforderungen, die wir nicht beim ersten Mal geschafft haben, machen uns am glücklichsten, wenn wir sie schließlich gemeistert haben. Wenn wir ein Spiel spielen, ist unser Erfolgserlebnis umso größer, je schwieriger es war, das Ziel zu erreichen. An dieser Stelle ist es wichtig zu verstehen, was mit selbstorganisiertem Lernen gemeint ist und was dazu nötig ist. (Schutz & Schwarz, 2022, S. 14)

Merkmale von Lernenden, die wissen, wie das Lernen funktioniert (Schutz & Schwarz, 2022, S. 8):

- ich weiß, wohin ich gehe
- Ich habe die Werkzeuge für diese Lernreise
- Ich beobachte meinen eigenen Lernprozess
- Ich erkenne, wann ich bereit bin für das, was als nächstes kommt
- Ich weiß, was ich als nächstes machen sollte

Gamer:innen können hier womöglich Parallelen zu Computerspielen ziehen und stellen vielleicht fest, dass sie in Spielen nach diesen Merksätzen handeln, auch wenn ihnen das nicht bewusst war. Die Gemeinschaft der Gamer:innen stellt weltweit die größte Gruppe dar, die die Prinzipien des selbstorganisierten Lernens anwendet. Durch den Einsatz von Spielen, die für Lernzwecke erstellt wurden und diesen Prinzipien folgen, kann eine Kultur des selbstorganisierten Lernens für das 21. Jahrhundert aufgebaut werden. (Schutz & Schwarz, 2022, S. 10, S. 29-32)

Dabei stehen nicht mehr Lehrende im Mittelpunkt, sondern Lernende, welche auf ihre eigene Weise und in ihrer eigenen Geschwindigkeit agieren. Wichtig ist aber, dass Lernspiele so aufgebaut sind, dass Lehrende nicht verschwinden, sondern den Schüler:innen beratend beistehen. Die Spielerfahrung soll so aufgebaut sein, dass sich Schüler:innen selbst motivieren, neugierig bleiben und zuversichtlich bis zum Ende des Lehrinhaltes gelangen. Hier kann man auch Parallelen zwischen Game Designern und Lehrenden ziehen. Beide haben die Aufgabe, Spieler:innen oder Lernende mit Motivation, Neugier und Zuversicht bis zum Abschluss des Spiels oder des Lehrinhaltes zu begleiten. (Schutz & Schwarz, 2022, 14-19)

In Zukunft wird es möglich sein, Spiele mit Hilfe von KI zu entwickeln, die zusätzliche Lehrende bereitstellen, die mit Rat und Tat dienen und an die sich Nutzer:innen jederzeit und ganz individuell wenden können. Mit diesen können auch Debatten geführt oder kreativ zusammengearbeitet werden. Somit wären auch alle Kriterien des „8 Learning Events Model“ für das Game-Based Learning erfüllt, welches Erkenntnisse und Methoden aus über 2000 Jahren Pädagogik destilliert und auf acht unterschiedliche Lern-/Lehr-Methoden reduziert hat. (Schutz & Schwarz, 2022, S. 13-16)

- Imitation/Modelling: Ein:e Lehrer:in macht vor, ein:e Schüler:in macht nach.
- Reception/Transmission: Ein:e Lehrer:in trägt vor oder zeigt, ein:e Schüler:in hört/schaut zu.
- Exercising/Guidance: Ein:e Lehrer:in leitet an und führt, ein:e Schüler:in übt selbst.

- Exploration/Documentation: Ein:e Lehrer:in bietet Material an, ein:e Schüler:in exploriert es.
- Experimentation/Reactivity: Ein:e Schüler:in experimentiert, ein:e Lehrer:in reagiert darauf.
- Creation/Confortation: Ein:e Schüler:in erschafft selbst, ein:e Lehrer:in bestärkt und unterstützt.
- Self-reflexion/Co-reflexion: Ein:e Schüler:in reflektiert sich selbst, ein:e Lehrer:in co-reflektiert.
- Debate/Animation: Ein:e Lehrer:in animiert eine Debatte, ein:e Schüler:in debattiert. (Schutz & Schwarz, 2022, S. 15)

2.2 Orientierung und Navigation in virtuellen Umgebungen mit einem Avatar

Bei der Orientierung und Navigation in einer virtuellen Umgebung müssen die richtigen topographischen Objekte in ein Modell implementiert werden, um gewährleisten zu können, dass sich Nutzer:innen beim Begehen eines Stadtmodells an diesen orientieren können. Eine Orientierung in der Umgebung mit Hilfe von Navigationsmöglichkeiten kann aber auch wichtig sein. Hierfür müssen je nach Einsatzbereich einer Anwendung auch verschiedene Möglichkeiten in Betracht gezogen werden. Beispielsweise können 2D-Orientierungsmöglichkeiten zur Wegsuche genutzt werden, wie Übersichtskarten vom Gelände oder eine geführte Navigation zu einem bestimmten Ort, der aufgesucht werden soll. Orientierung und Navigation in einem Raum bilden zusammen den Prozess, einen Zielort in bekannter oder unbekannter Umgebung zu finden. Die Orientierung beschreibt das Zurechtfinden in der Umgebung und führt dadurch oft zu einer zielgerichteten Bewegung, die als Navigation bezeichnet wird. (Myriam Seidt & Katharina Wolter, 1999, S. 7)

Bezüglich der Bewegung ist zu beachten, dass je nach Größe der Umgebung die passende Fortbewegungsmöglichkeit für den gesteuerten Avatar bereitgestellt werden sollte. Wenn ein Modell mehrere Quadratkilometer umfasst, kann es erforderlich sein, an bestimmte Orte mit einem Fahrzeug zu fahren, anstatt zu Fuß gehen zu müssen. Auch die Perspektive, die man bei der Steuerung einnimmt, spielt eine Rolle für die Orientierung und Navigation.

2.2.1 Orientierung in virtuellen Umgebungen

Kartographische Darstellungen mit einer abstrahierten Grundrissdarstellung, wie sie in normalen 2D-Karten meist vorliegt, können die Orientierung besonders in topographisch anspruchsvollen Umgebungen sehr erschweren. Daher wird immer

wieder versucht, mit Aufrissen in Schräg- oder Perspektivansichten besonders wichtige Objekte darzustellen. Der Nachteil einer solchen Darstellung liegt oft darin, dass sie zeichnerisch schwer umzusetzen ist und Objekte, die sich hinter den Abbildungen befinden, verdeckt. (Kohlstock, 2018, S. 105) In einer virtuellen Umgebung, wie in einer Game Engine, sind solche Nachteile nicht gegeben und es eröffnen sich dadurch umfangreiche Visualisierungsmöglichkeiten, in denen Nutzer:innen sich nicht mit kartographischen Abstraktionen auseinandersetzen müssen. Gleichzeitig können Nutzer:innen ein 3D-Stadtmodell aus allen Perspektiven betrachten und auch räumlich anspruchsvolle Gebiete wie Städte können realitätsnah dargestellt werden. Für ein 3D-Modell eines städtischen Raumes müssen jedoch auch wie in einer 2D-Karte grundlegende topographische Objekte vorhanden sein, damit Nutzer:innen den Ort, an dem sie sich befinden, wiedererkennen und sich dort orientieren können. Für ein realitätsnahes 3D-Modell gilt: Je mehr Objekte und Details der Wirklichkeit es abbildet, desto höher werden die Immersion und der Realismus eines solchen Modells. Je nach Anwendungszweck muss jedoch auch hier der Detailgrad im Voraus definiert werden. Ein 3D-Modell, das bei einem Bürger:innenentscheid nur die Abschattungen eines neuen Bauvorhabens vermitteln soll, muss keine detaillierte Fassade des Gebäudes und auch nicht jedes Detail der Vegetation oder der Möblierung der Stadt darstellen. Wenn ein Modell jedoch ein realitätsnahes Erlebnis für ein Computerspiel erzeugen soll, ist es in vielen Anwendungsfällen wichtig, städtische Möblierungen und Vegetation so genau wie möglich abzubilden. Ein Modell, das die Wirklichkeit 1:1 widerspiegelt, ist aber auch in einer virtuellen Umgebung nicht umzusetzen. Folglich muss auch hier, wie in der Kartographie üblich, eine Erfassungsgeneralisierung stattfinden, selbst wenn der Raum in einer virtuellen Umgebung durch einen 1:1 Maßstab nicht verkleinert wird. (Kohlstock, 2018, S. 75–76)

Zur Klärung der Frage, welche topographischen Elemente für ein immersives Erlebnis in einem Stadtmodell notwendig sind, kann man sich auf die grundlegenden topographischen Elemente stützen, die zur Orientierung in der Umgebung eines 2D-Kartenwerkes genutzt werden. Die natürlichen und künstlichen Objekte, die die Erdoberfläche widerspiegeln, werden gegliedert in Situationen (Grundrissobjekte) und das Relief des Geländes (Höhen und Geländeform). Situationen sind alle Objekte, deren Grundriss definiert ist. Dazu zählen:

- Verkehrswege (Wege, Straßen, Gleise)
- Gewässer (Bäche, Flüsse, Kanäle, Seen, Meere)
- Siedlungen (Gebäude)
- Vegetation und Nutzungsflächen (Wälder, Äcker, Gärten, Bäume usw.)
- Topographische Einzelelemente kleineren Ausmaßes (Zäune, Hecken, Mauern, Laternen, Masten usw.)

(Kohlstock, 2018, S. 47–49)

Für eine gute Orientierung und ein immersives Erlebnis in einer virtuellen Umgebung werden das Relief der Erdoberfläche in Form eines DGM sowie Grundrissobjekte von Verkehrswegen, Gewässern, Siedlungen, Vegetation und Nutzungsflächen benötigt. Mit topographischen Elementen, die von den Vermessungsämtern aufgezeichnet werden, lassen sich immersive Umgebungen gestalten, die für eine gute Orientierung in einer virtuellen Umgebung ausreichend sind.

Die einfachste Form dies zu realisieren, besteht darin, Luftbilddaufnahmen oder topographische Karten auf ein DGM zu projizieren und mit Hilfe von LoD₂-Gebäuden ein Modell zu erstellen, wie es beispielsweise schon im BayernAtlas zur Verfügung steht. Hierbei können Bildkarten (Luftbilddaufnahmen, Digitale Orthophotos (DOPs), Luftbildmosaike) oder topographische Kartenwerke (topographische Karten, Planungskarten, Webkarten usw.) für die Situationsabbildung genutzt werden. (Kohlstock, 2018, S. 112, 123) Es ist aber darauf zu achten, dass die Orientierung in einem solchen Modell mit einem Avatar in einem menschlichen Maßstab als Interaktionsmittel begrenzt ist. Diese Beschränkung entsteht durch die Auflösung und den Maßstab der Bildkarten sowie der topographischen Karten, die zur Verwendung kommen. Möglicherweise muss für die Interaktion mit dem Modell und für die Orientierung darin ein geeignetes Fortbewegungsmittel gewählt werden, das auch in einer bestimmten Höhe oder in der Vertikalen agieren kann.

Im Allgemeinen stellen vorhandene Datensätze der Vermessungsämter und des LDBV somit eine gute Grundlage für ein Modell dar, das ein immersives Erlebnis bei der Visualisierung erzeugen kann.

2.2.2 Navigation in virtuellen Umgebungen

Neben der Orientierung im Raum, die über die Topographie und Topologie des Modells stattfindet, ist in einer virtuellen Umgebung auch die Navigation von großer Bedeutung. Diese beschreibt anhand der Orientierung an der Topographie und Topologie im Modell den Weg zu einem Ziel, das man erreichen möchte. Während der Navigation werden durch Entscheidungsfindungen, basierend auf Hinweisen aus der Umgebung sowie dem eigenen Wissen und bestehenden Erfahrungen, neue Erfahrungen gesammelt, wodurch das Wissen über eine Umgebung erweitert wird. Orientierung und Navigation sind somit ein dynamischer Prozess, der aus den Bestandteilen Informationsverarbeitung, Entscheidungsfindung und Entscheidungsausführung besteht. Eine gute Orientierung kann Nutzer:innen zu einer zielgerichteten Bewegung führen, die man als Navigation bezeichnet. (Myriam Seidt & Katharina Wolter, 1999, S. 7–8)

Um Nutzer:innen einer größeren virtuellen Umgebung bei der Navigation zu unterstützen, ist es wichtig, dass sie sich gut in der Umgebung des Modells orientieren können, selbst wenn sie sich zum ersten Mal dort befinden und keine Kenntnisse über die Örtlichkeit besitzen. Die Navigation in einem virtuellen Raum wird immer dann zum Problem, wenn der Raum größer ist als der View der Nutzer:innen. Hier kann, je nach Anwendung und deren Ziel, eine Navigationshilfe wichtig sein. Vor allem wenn

es sich um größere virtuelle Umgebungen handelt, kann man so Problemen bei der Orientierung entgegenwirken. (Carter et al., 2012, S. 90–92; Darken, o.d.)

Um in einer interaktiven Anwendung Nutzer:innen die Navigation zu erleichtern, ist es wichtig, passende Werkzeuge dafür bereitzustellen. Damit sich Nutzer:innen in einer Umgebung schnell zurechtfinden können, müssen zwei Fragen geklärt werden.

Gibt es ein Ziel, das Nutzer:innen erreichen sollen?

Dabei ist zu hinterfragen, was das Ziel ist und zu entscheiden, wie es gefunden werden soll:

- Es gibt ein Ziel, aber Nutzer:innen kennen das Ziel nicht und sollen das komplette Gebiet absuchen. (Umfassende Suche)
- Es gibt ein Ziel, Nutzer:innen kennen das Ziel und sollen es zielgerichtet finden. (Zielgerichtete Suche)
- Es gibt kein Ziel und Nutzer:innen sollen das Gebiet nur erkunden. (Reine Erkundung)

Oft ist die Zielfindung ein Prozess, der aus einer zielgerichteten Suche und einer anschließenden umfassenden Suche besteht. Man kennt den eingegrenzten Standort eines Ziels und kann somit zielgerichtet danach suchen. Am eingegrenzten Standort angekommen, muss dann eine umfassende Suche durchgeführt werden, um das Ziel zu erreichen.

In der realen Welt kommt die umfassende Suche selten vor, wohingegen in einer virtuellen Welt bei Nutzer:innen, die das Modell erstmalig nutzen, dies meist den Regelfall darstellt. (Darken, o.d.)

Welche Navigationsmöglichkeiten gibt es für die Zielsuche und wird dafür eine geführte Navigation benötigt?

Bei dieser Frage soll geklärt werden, welche Form von Navigation für die Zielfindung eingesetzt werden kann und welche Art von Navigationssystem hierfür in Frage kommt.

MiniMap: Diese wird in Computerspielen als Orientierungssystem in größeren Umgebungen verwendet und funktioniert wie ein Global Positioning System (GPS) für virtuelle Umgebungen. Der Standort der Nutzer:innen wird in einem Ausschnitt einer Übersichtskarte der Umgebung angezeigt und folgt meist dynamisch dem Standort der Nutzer:innen.

Map: Die Map zeigt eine Übersichtskarte der gesamten Umgebung, in der sich die Nutzer:innen befinden.

Die MiniMap und die Map sollten stets die aktuelle Position und die Blickrichtung der Nutzer:innen beinhalten, um die Orientierung und Navigation zu erleichtern. (Carter et al., 2012; Darken, o.d.)

Des Weiteren muss geklärt werden, wie Nutzer:innen in der virtuellen Umgebung navigieren sollen. Hierfür kann, je nach Ziel der Anwendung, neben der MiniMap und der Map auch ein geführtes Navigationssystem eingesetzt werden.

Umfassende Suche

Hierbei können Nutzer:innen mit Hilfe der Unterstützung der MiniMap und der Map systematisch die Welt absuchen (falls für die Anwendung gewünscht). Durch die Orientierungshilfe in den beiden Karten wird verhindert, dass dieselben Gebiete mehrmals abgesucht werden. Gleichzeitig kann ein Ziel nach Auffinden in den Karten verortet werden. So wird Nutzer:innen z. B. das wiederholte Auffinden eines zuvor erreichten Ziels erleichtert.

Zielgerichtete Suche

Für die zielgerichteten Suche können, je nach Anwendung, verschiedene Ansätze der Navigation verwendet werden. Die MiniMap und die Map können mit weiteren Orientierungshilfen wie z. B. Zielsymbolen versehen werden, oder es wird eine geführte Navigation implementiert.

Mit Hilfe von Zielen, die in der MiniMap und Map verortet sind, kann eine zielgerichtete Navigation durchgeführt werden. Die Navigation wird dann von den Nutzer:innen selbst ausgeführt. Sie können sich an topographischen Objekten und der Topologie in der Umgebung orientieren und den Objekten (z. B. einem Kirchturm, einer Straße oder entlang eines Flusses) folgen, um ihr Ziel zu erreichen. Mit Hilfe der MiniMap und Map können Nutzer:innen aber auch ohne Orientierung an der Topographie und Topologie direkt in Richtung des Ziels navigieren, das auf der Karte dargestellt wird.

Alternativ können Nutzer:innen auch mittels einer Wegbeschreibung zum Ziel geführt werden, die auf topographischen Objekten und der Topologie in der Umgebung basiert, vergleichbar mit einer Wegbeschreibung, wenn man einen Ortskundigen nach dem Weg fragt.

Die einfachste Form der Navigation für Nutzer:innen ist jedoch ein geführtes Navigationssystem, wie es heute üblich ist. Dabei können Nutzer:innen entweder einer angezeigten Route zum Ziel folgen oder sie werden durch Schritt-für-Schritt-Anweisungen bis zum Ziel geleitet. Beide Systeme können einzeln oder in Kombination verwendet werden.

Reine Erkundung

Bei der reinen Erkundung können Nutzer:innen mit Hilfe der MiniMap und Map Gebiete, die ihnen interessant erscheinen, eingrenzen (falls für die Anwendung gewünscht). Gleichzeitig können sie sich einen Überblick über die gesamte Umgebung verschaffen. (Darken, o.d.; Schwering et al., 2017)

Die korrekte Auswahl der Navigationssysteme kann Frustrationen verhindern, die ohne diese Hilfsmittel entstehen könnten. Gleichzeitig kann Nutzer:innen durch eine gezielte Führung durch eine Umgebung Wissen vermittelt werden. Zu diesem Zweck können Methoden der Navigation aus der realen Welt auf die virtuelle Welt übertragen werden.

2.2.3 Perspektive, Fortbewegung und Interaktion in interaktiven virtuellen Umgebungen

Neben der Orientierung und der Navigation in einer virtuellen Umgebung sind auch die Fortbewegungsmöglichkeiten und die eingenommene Perspektive bei der Bewegung ausschlaggebende Punkte für die Erfahrung und das Immersionserlebnis der Nutzer:innen. In großen virtuellen Umgebungen, in denen ohne ein Fahrzeug viel Zeit benötigt würde, um von einem Ort zum nächsten zu gelangen, ist es notwendig, das passende Fortbewegungsmittel bereitzustellen. So kann z. B. ein Wechsel von einem zu steuernden Avatar zu einem Flugzeug einen erweiterten Bewegungsraum in der Vertikalen ermöglichen und einen anderen Blickwinkel auf die Umgebung eröffnen. (Denisova & Cairns, 2015, S. 147–148)

Die Perspektive bezieht sich auf die Kameraperspektive, die im Spiel eingenommen wird, und spielt eine große Rolle bei der Erfahrung. Man unterscheidet dabei zwischen einer egozentrischen und einer exozentrischen Perspektive. Durch Wechsel der Perspektive und durch unterschiedliche Blickwinkel auf eine Umgebung kann zudem der Lerneffekt verstärkt werden. (Adamo-Villani & Wilbur, 2.1 Virtual Environments for Education)

Egozentrisch

Nutzer:innen nehmen die Egoperspektive in einem Spiel ein, diese wird auch als First-Person-Perspektive bezeichnet. Spieler:innen sehen die dargestellte Umgebung aus den Augen der Spielfigur. Dies ermöglicht einen hohen Grad an Immersion. Da Nutzer:innen in die Position der Spielfigur schlüpfen, können sie sich besser mit ihr identifizieren, was die Erfahrung innerhalb der Anwendung beeinflusst.

Exozentrisch

Nutzer:innen betrachten die Spielfigur und stehen hinter dieser. Die Spielfigur kann während der Steuerung beobachtet werden. Dadurch entsteht eine gewisse Distanz zur Spielfigur. Nutzer:innen identifizieren sich hier nicht so stark mit dem Avatar wie bei der egozentrischen Perspektive. (Denisova & Cairns, 2015, S. 147–148)

Erfahrene Nutzer:innen bevorzugen oft die egozentrische Perspektive. Diese bietet ein immersiveres Erlebnis und die Spielfigur kann präziser gesteuert werden. Im Gegensatz dazu kann die Spielfigur aus der exozentrischen Perspektive leichter gesteuert werden, was besonders für Anfänger von Vorteil ist. Die Immersion ist aber auch bei Anfängern in der egozentrischen Perspektive höher, obwohl die erleichterte Steuerung wiederum einen positiven Einfluss auf die erlebte Immersion hat. (Denisova & Cairns, 2015, S. 147–148)

Neben der eingenommenen Perspektive spielt auch das Fortbewegungsmittel in der Anwendung eine wichtige Rolle. Im Folgenden soll ein Überblick über die am häufigsten vorkommenden Fortbewegungsmittel in Computerspielen gegeben werden:

Avatar

Der Avatar als künstlicher Stellvertreter der Nutzer:innen kann verschiedenste Arten der Fortbewegung nutzen. Darunter sind die gängigsten das Gehen, Laufen, Sprinten und Springen. In einigen Sonderfällen kann er eventuell auch fliegen oder, je nachdem, wen oder was der Avatar verkörpert und welche Aufgaben er übernehmen soll, jede erdenkliche Form der Fortbewegung nutzen. Er kann in menschlicher Form auftreten, aber auch als Tier, Roboter oder in Form eines beliebigen anderen Wesens.

Fahrzeuge

Mit Hilfe von Fahrzeugen können Nutzer:innen Strecken zwischen zwei Orten schneller zurücklegen als mit einem menschlichen Avatar, der nur gehen kann. Daher werden sie oft zu diesem Zweck genutzt. Sie benötigen jedoch eine größere Umgebung, in der sich Nutzer:innen mit diesen bewegen können. Fahrzeuge können aber auch je nach Anwendungszweck des Spiels bestimmte Aufgaben erfüllen und in jeglicher Form zur Anwendung kommen. Zu den gängigsten Fahrzeugen zählen Autos, Motorräder und Boote.

Luftfahrzeuge

Luftfahrzeuge ermöglichen es Spieler:innen, die Umgebung zusätzlich in der Vertikalen zu erkunden. Auch mit dieser Form der Fortbewegung können Spieler:innen große Strecken in kurzer Zeit zurücklegen. Gleichzeitig bietet es die Möglichkeit, durch Erkundungen in der Vertikalen einen anderen Blickwinkel auf eine Umgebung zu erhalten. Zu den gängigsten Formen von Luftfahrzeugen gehören Flugzeuge, Helikopter und Drohnen.

Teleportation

Eine Sonderform der Fortbewegung stellt die Teleportation dar. Sie wird in virtuellen Umgebungen gerne genutzt, wenn lange Distanzen zurückgelegt oder sogar Reisen in andere Länder oder Welten oder durch die Zeit ermöglicht werden sollen, ohne dass Nutzer:innen eine Strecke dafür zurücklegen müssen.

In Spielen wird heutzutage eine Vielzahl von Fortbewegungsmöglichkeiten angeboten. Der Fantasie oder dem Anwendungszweck sind dabei keine Grenzen gesetzt. Gleichzeitig bieten sie die Möglichkeit, das Erlebnis in einer Anwendung abwechslungsreicher zu gestalten.

Bei der Steuerung der Fortbewegungsmittel ist auch das zum Einsatz kommende Eingabegerät von Bedeutung. Die gängigste Form stellt die Eingabe mit Tastatur und Maus dar. Je nach Zielgruppe und Anwendungsfall kann es aber auch von Vorteil sein, einen speziellen Gamecontroller für die Interaktion mit der Anwendung zu verwenden, um die Fortbewegungsmöglichkeit und Perspektive zu steuern.

Für die Nutzung mit Geodaten ist neben dem Avatar insbesondere die Fortbewegung mit Luftfahrzeugen von Interesse, da diese den Nutzer:innen die Möglichkeit bieten, die Umgebung auch in der Vertikalen zu erkunden und aus einer anderen Perspektive zu betrachten. Amtliche Geodaten können so auf spielerische Art und Weise erlebbar gemacht und barrierefrei aus allen Perspektiven betrachtet und präsentiert werden.

2.3 Aktuelle Beispiele von Geodaten in virtuellen Umgebungen

2.3.1 Beleuchtungssimulation der Stadt Oldenburg

Im Rahmen seiner Masterarbeit mit dem Thema „Untersuchung zum Potenzial von Game Engines für Beleuchtungs-Simulationen im 3D-Stadtmodell“ hat Helge Olberding eine interaktive Anwendung für die Stadt Oldenburg geschaffen, die Beleuchtungs-Simulationen in Echtzeit bietet. Das Ziel der Arbeit war, die Stadt interaktiv begehen zu können und Beleuchtungstests realitätsnah zu simulieren. Dadurch können Aufwendungen für Beleuchtungstests, die sonst vor Ort stattfinden müssten, reduziert werden. In der Anwendung können verschiedene Arten von Lampenmodellen interaktiv im Modell platziert und durch Datum und Uhrzeit die Sonnen- und Mondstände dynamisch beeinflusst werden. Dies ermöglicht realitätsnahe Visualisierungen bei der Planung und Platzierung von städtischer Beleuchtung, siehe Abbildung 2. Durch die Möglichkeiten der Interaktion und die Option, verschiedene Beleuchtungsvarianten auszuprobieren, eignet sich das Modell auch gut für Präsentationen und die interaktive Einbeziehung des Publikums in die Planung. (Olberding, 2018a, 2018b)

Für die Umsetzung wurde auf CityGML-Daten für die Gebäudemodelle und ein DGM der Stadt Oldenburg zurückgegriffen. Diese Daten wurden mittels 3ds Max® in die Unreal Engine 4 integriert. (Olberding, 2018a, S. 27)

Durch den Einsatz solcher Modelle kann nicht nur der Planungsaufwand bei Beleuchtungssimulationen reduziert werden, sondern auch der Klimaschutz wird unterstützt, indem die städtische Beleuchtung auf das notwendige Minimum reduziert wird. Zudem ermöglicht es die interaktive Einbeziehung der Bürger:innen in die Planung.



Abbildung 2: Beleuchtungssimulation des Rathausmarkts in Oldenburg

(Olberding, 2018a, S. 65)

2.3.2 Virtual Helsinki

Virtual Helsinki wurde von der finnischen Tech-Firma ZOAN entwickelt. In dieser Anwendung kann die finnische Hauptstadt mit einer VR-Brille erkundet werden. Ursprünglich war das Ziel, die Stadt für den smarten Tourismus zu präsentieren und eine nachhaltige Form des Reisens zu bieten. Gleichzeitig sollte aber auch Werbung für den realen Tourismus in der Stadt gemacht werden, auch wenn dies womöglich etwas im Widerspruch zueinander steht. Die Anwendung ist auch für das Metaverse in Nutzung und ermöglicht es auf Anfrage, virtuelle Veranstaltungen wie z. B. Konzerte, Messen oder Sportevents in der digitalen Version der Stadt zu erleben. Zukünftige Bauvorhaben wurden visualisiert, um Anwender:innen einen Blick auf die Zukunft der Stadt zu ermöglichen. Gleichzeitig besteht die Möglichkeit, in Teilen der Stadt historische Stadttouren zu unternehmen und Helsinki Anfang des 20. Jahrhunderts zu erleben. Nutzer:innen können Gebäude begehen, die in überragender Qualität visualisiert wurden, wie beispielsweise das Alvar Aalto House, das Zuhause des berühmtesten Architekten Finnlands, siehe Abbildung 3. Es werden auch virtuelle Einkaufsmöglichkeiten geboten, bei denen Nutzer:innen auf einer Shoppingtour durch die Geschäfte der Stadt Produkte kaufen können und diese anschließend per Paketdienst nach Hause geliefert bekommen. Der Realismus ist durch die aufwändigen Modellierungsarbeiten sehr hoch, siehe Abbildung 4. Zusätzlich werden Echtzeit-Augmented Reality (AR)-Anwendungen angeboten, die z. B. bei Konzerten aufwändige, mit der Musik synchronisierte Visualisierungen bieten und über das Smartphone betrachtet werden können. (Epic Games, 2020; Potor, 2018; Ronen Bekerman, 2019; ZOAN - *Virtual Helsinki*, n.d.)

Für die Umsetzung des Projekts konnte ein 3D-Scan-Modell, das von der Stadt Helsinki bereitgestellt wurde, als Grundlage für die Proportionen und Positionen der Objekte in der Stadt verwendet werden. Die Stadt wurde dann mit 3D-Modellen nachmodelliert, die mit 3ds Max[®] und ZBRUSH erstellt und in Unreal Engine 4 interaktiv ausgestaltet wurden. Die Modellierungsarbeiten wurden anhand von Bildern durchgeführt, die von der ganzen Stadt aus verschiedenen Perspektiven aufgenommen wurden. Die meiste Arbeit wurde durch traditionelle 3D-Modellierung ausgeführt, kombiniert mit speziellen photogrammetrischen Verfahren. Skulpturen und Ornamente in der Stadt wurden mit ZBRUSH, einem speziellen Grafikprogramm für das digitale Sculpting, erstellt. (Ronen Bekerman, 2019)

Obwohl Virtual Helsinki aufgrund des hohen manuellen Arbeitsaufwandes mit klassischen Modellierungsverfahren nur noch in der Grundlage mit Geodaten aus der Vermessung verknüpft ist, zeigt es jedoch eindrucksvoll, was mit Unreal Engine im Bereich der 3D-Stadtvisualisierung möglich ist. Auch die Anwendungsmöglichkeiten eines solchen Modells sind beeindruckend.



Abbildung 3: Alvar Aalto House, Visualisierung in der Unreal Engine

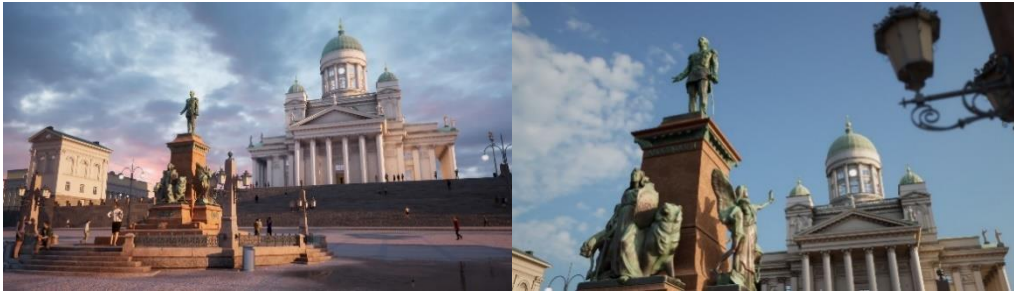


Abbildung 4: Senatsplatz Helsinki, Visualisierung in der Unreal Engine

(Ronen Bekerman, 2019)

2.3.3 Project Anywhere

Project Anywhere wurde von Cesium, Epic Games, NVIDIA und Microsoft als Demoprojekt entwickelt, um die Möglichkeiten von photogrammetrisch gewonnenen Geodaten in Unreal Engine mit dem Cesium Plugin zu präsentieren. Die Anwendung wurde für die Nutzung mit VR-Brillen und Mixed-Reality-Anwendungen ausgelegt. In der Anwendung können Nutzer:innen die ganze Welt erkunden und sich frei von Ort zu Ort teleportieren. Ähnlich wie in Google Earth kann der ganze Globus betrachtet werden, siehe Abbildung 5, oder eine bestimmte Region auf der Erde, siehe Abbildung 6. Das kostenlose Demoprojekt ist im Epic Game Store verfügbar, und Entwickler:innen können es in der Entwicklungsumgebung der Unreal Engine nutzen, um davon zu lernen oder es als Grundlage für eigene Projekte zu verwenden. (Epic Games, 2021; Mohammed, 2021)

Mit dem Projekt können die weltweit zur Verfügung stehenden 3D-Tiles von Cesium, die in einem hochauflösenden photogrammetrischen Verfahren gewonnen wurden, in die Unreal Engine 4 gestreamt werden. (Epic Games, 2021)

Anhand von Project Anywhere kann ein Einblick in die Möglichkeiten von AR- und VR-Anwendungen in Verbindung mit weltweiten Geodaten vermittelt werden. Entwickler können sich so einen Überblick über Visualisierungsmöglichkeiten und die Qualität von photogrammetrisch gewonnenen Geodaten verschaffen.

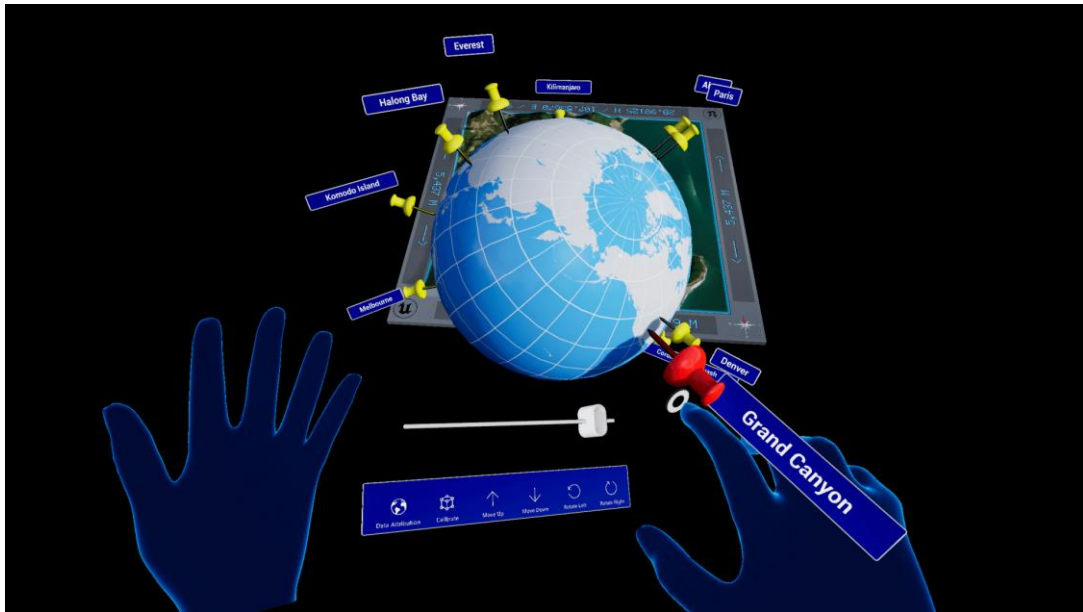


Abbildung 5: Project Anywhere, Sicht auf den Globus



Abbildung 6: Project Anywhere, XR-Ansicht der 3D-Tiles von CESIUM ion
(Epic Games, 2021)

3. METHODENTEIL

3.1 Grundlagen

Um herauszufinden, mit welchen Methoden es möglich ist, die Geodaten des LDBV in Unreal Engine zu importieren, wurden einige Testmodelle erstellt. So sollte ein Weg für die Integration der Geodaten des LDBV in Unreal Engine gefunden werden. Wie bereits in Kapitel 2.1.4 beschrieben, kommen hierfür InfraWorks® sowie weitere Autodesk-Lösungen oder das Blender GIS Plugin in Frage. Auch das Ludic Drive Landscaping Plugin und das ArcGIS Maps SDK Plugin for Unreal Engine stellen hierfür interessante Lösungen dar.

Das LDBV favorisiert einen Workflow über InfraWorks®, da hiermit schon einige Modelle erstellt wurden. Daher wären bereits 3D-Modelle vorhanden, die mit Unreal Engine weiterverarbeitet und interaktiv nutzbar gemacht werden könnten. Auch ist im LDBV schon eine gewisse Expertise in der Arbeit mit InfraWorks® vorhanden. Die Arbeit mit InfraWorks® stellt auch eine gute Lösung dafür dar, die benötigten Geodaten vorzubereiten und Wege und Straßen in einer generalisierten Form in ein 3D-Modell umzuwandeln und zu visualisieren. Dadurch konnte die Methode, bei der vor der Arbeit in Unreal Engine ein 3D-Modell erstellt werden muss, um dieses anschließend in Unreal Engine zu importieren, auf die Arbeit mit InfraWorks® beschränkt werden. Diese Methode dient nun als Grundlage für die weitere Arbeit. Das interessante Ludic Drive Landscaping Plugin wurde aufgrund von Zusatzkosten, die für Testzwecke entstanden wären, nicht in die Tests eingeschlossen.

Neben der Auswahl der geeigneten Methode zum Erstellen eines Modells und zum Import in Unreal Engine musste auch die Funktionalität, die das fertige Projekt haben sollte, abgeklärt werden. Gemäß der Vorgabe des LDBV sollte das in die Engine importierte Modell mit einem Avatar begehbar sein. Zudem sollte ein Umschalten zwischen dem erstellten 3D-Modell und weiteren Modellen, die aus einem DGM mit integrierten Texturen aus Kartenmaterial oder Orthophotos bestehen, ermöglicht werden. Die fertige Anwendung sollte außerdem Nutzer:innen aller Altersgruppen weitestgehende Barrierefreiheit bieten und über ein User Interface bedienbar sein. Zusätzlich zu diesen Funktionen wurde die Entscheidung getroffen, eine MiniMap und Map für die Navigation zu integrieren, wie in Kapitel 2.2.2 beschrieben. Auch bezüglich der Fortbewegung und Perspektive zur Betrachtung und Erkundung des Modells, wie in Kapitel 2.2.3 beschrieben, wurde entschieden, dass zwischen einer exozentrischen Perspektive bei der Steuerung des Avatars und einer egozentrischen Perspektive bei der Steuerung einer Drohne umgeschaltet werden kann.

Bei korrekter Umsetzung kann so eine gute Grundlage für eine Anwendung mit Geodaten für interaktive Präsentationszwecke für das LDBV geschaffen werden.

3.1.1 Daten und Datenaufbereitung

Das LDBV hat die verwendeten Daten für das zu bearbeitende Gebiet fertig aufbereitet bereitgestellt und sie konnten direkt weiterverarbeitet werden.

Amtliches Topographisch-Kartographisches Informationssystem (ATKIS®)

Die ATKIS®-Vektordaten bilden die Grundlage der topographischen Karte 1:25000 des Digitalen Landschaftsmodells (Basis-DLM) und sind bundesweit einheitlich geregelt.

Da aber nur ein kleiner Teil der Daten benötigt wird und manche der Daten Überschneidungen aufweisen, müssen diese manuell und spezifisch für das Modell ausgewählt werden. Die Daten werden in dieser Auswahl mit QGIS, siehe Abbildung 7, auf das Wesentliche reduziert, damit alle topographischen Objekte der Landschaft zur Orientierung im Modell, siehe Kapitel 2.2.1, vorhanden sind.

Aus den ausgewählten ATKIS®-Daten wird mit InfraWorks® das 3D-Modell generiert. Diese Daten werden in InfraWorks® auch auf das DGM übertragen und dort integriert. Sie beinhalten alle Nutzungsflächen, Verkehrswege sowie Steh- und Fließgewässer.

Alle Informationen zu den einzelnen Layern können dem ATKIS®-Objektartenkatalog Basis-DLM in der aktuellen Version 6.0 entnommen werden. (*Arbeitsgemeinschaft Der Vermessungsverwaltungen - Adv-Online, n.d.*)

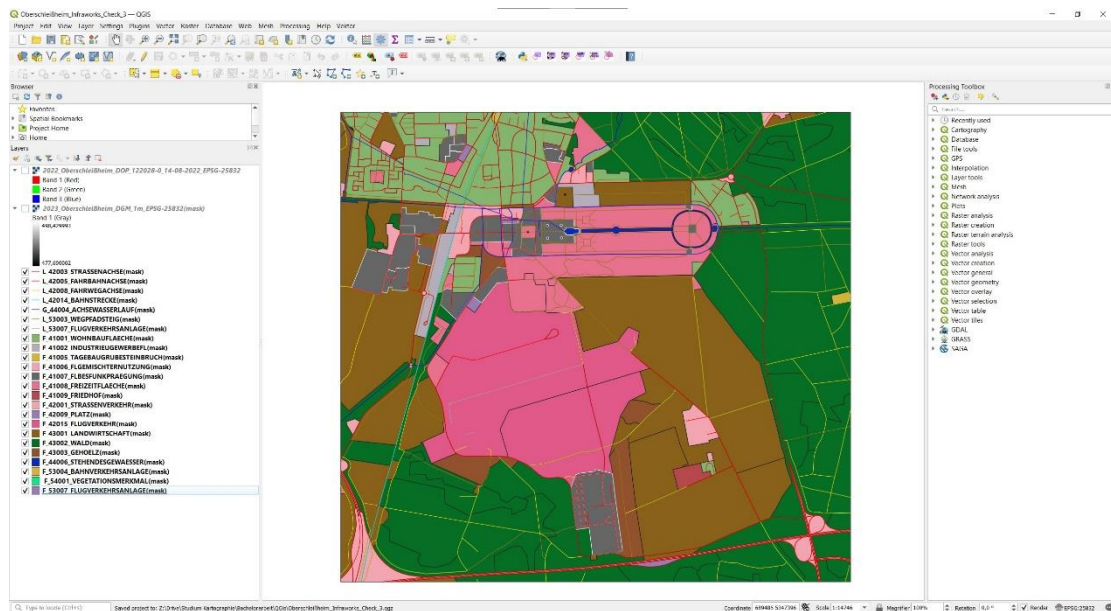


Abbildung 7: QGIS-Auswahl der ATKIS®-Daten.

DGM

Die mittels Airborne Laserscanning gewonnenen Daten des DGM beschreiben die Erdoberfläche ohne Vegetation und Bebauung in Form einer Punktwolke.

Das DGM wurde mit den Gitterweiten 1m und 5m bereitgestellt, um daraus im 3D-Modell die Erdoberfläche zu visualisieren und bildet im InfraWorks®-Modell die Grundlage der Lage und Höhen im Modell.

LoD2-Gebäudemodelle

Die verwendeten LoD2-Modelle werden aus den Gebäudegrundrissen vom Amtlichen Liegenschaftskatasterinformationssystem (ALKIS®), den Daten des Airborne Laserscannings und einem luftbildbasierten Digitalen Oberflächenmodell (DOM) erstellt. Die LoD2-Gebäude bilden auch die Dachform der Gebäude ab. Das LDBV bietet neben den LoD2-Gebäuden auch LoD1-Gebäude an. Die LoD1-Gebäude beschreiben die Gebäude nur als Klötzchen-Modell und basieren auf den ALKIS®-Gebäudegrundrissen und einer mittleren Gebäudehöhe aus den Daten des Airborne Laserscannings.

Durch die Arbeit mit InfraWorks® können die Gebäude einfach als Point Feature importiert werden.

Rasterdaten (Luftbilder und Kartenwerke)

Für die Bachelorarbeit stellte das LDBV digitale True Orthophotos, analoge Orthophotos (digitalisiert), historische Luftbilder von 1941-1945 (digitalisiert), historische Flurkarten von 1808-1864 (digitalisiert) sowie eine Planungskarte und eine Webkarte bereit. Alle analogen Bilder und Karten wurden mit Präzisionsscannern digitalisiert. Neben den Orthophotos und digitalen Karten sind auch alle weiteren historischen Rasterdaten wie Luftbilder und Kartenwerke georeferenziert.

Mit InfraWorks® ist es auch möglich, Rasterdaten in das DGM zu integrieren. So können z. B. für eine Zeitreihe alle Karten und Orthophotos, die aus Rasterdaten bestehen, im Modell betrachtet werden. Gleichzeitig dienen sie in der Anwendung als Hintergrundkarte für die MiniMap und Map, siehe Kapitel 2.2.2.

Dank der Arbeit mit InfraWorks®, welches die Daten analog zu einem GIS mit einem geodätischen Koordinatenreferenzsystem importieren kann, lassen sich alle Daten ohne Positionierungsarbeiten importieren. Die Verkehrswege sind aber aufgrund des Maßstabes der topographischen Karte 1:25000 des Basis-DLM nicht alle lagerichtig und beinhalten eine Verdrängung, die in InfraWorks® manuell entfernt werden muss. Diese Anpassungsarbeit ist je nach Modellgröße und Ausmaß der Verdrängung des Kartenausschnittes, die im Basis-DLM notwendig war, mehr oder weniger aufwändig.

3.1.2 Software und der Weg in die Entwicklungsumgebung

Für die finale Arbeit wurde ein Modell von Oberschleißheim mit Schlossanlage und Flughafen erstellt. Die ersten Tests wurden aber mit einem vorhandenen InfraWorks®-Modell von Unterschleißheim durchgeführt, um vor der Erstellung des Modells von Oberschleißheim erste Tests mit Unreal Engine durchführen zu können. Der erste wichtige Punkt war hierbei, das Modell aus InfraWorks® in die Unreal Engine zu importieren. Hierfür muss das Modell aus InfraWorks® im FBX-Format exportiert werden. Anschließend kann das Modell über verschiedene Wege in Unreal Engine importiert werden.

Der direkte Weg über die exportierte FBX-Datei von InfraWorks® ist eine Option. Aufgrund von Problemen, die hierbei durch die Struktur und den Aufbau eines komplexen Modelles beim Export in das FBX-Format entstehen können, empfiehlt es sich jedoch, die FBX-Datei mit Datasmith für den Import in Unreal Engine vorzubereiten. Ohne diesen Schritt wäre ein hoher Arbeitsaufwand nötig, um die FBX-Dateien hierarchisch korrekt zu strukturieren und Geometrien sowie Materialien für die Unreal Engine korrekt aufzubereiten, um diese so ressourcenschonend wie möglich visualisieren und verarbeiten zu können.

Datasmith übernimmt diesen sehr zeitaufwändigen Teil für eine Vielzahl von CAD-Programmen und steht für diese als Plugin zur Verfügung. Damit erhalten exportierte 3D-Modelle im Datasmith-Format eine für Unreal Engine optimal lesbare Struktur. Auf diese Weise können die Daten auch für Echtzeitanwendungen in Unreal Engine optimal genutzt werden. Fehler in der Visualisierung oder Fehler, die Performanceprobleme verursachen, werden vermieden, siehe Kapitel 3.2.2. Darüber hinaus bietet Datasmith die Möglichkeit, das Modell über eine Direktlink-Funktion nach dem Import in Unreal Engine nochmal in der Software, die zum Export in das Datasmith-Format verwendet wurde, nachzubearbeiten. Änderungen an einem Modell können so in Unreal Engine über eine Synchronisation aktualisiert werden, ohne das ganze Modell erneut importieren zu müssen. Da aber für InfraWorks® zum momentanen Zeitpunkt kein Datasmith Plugin zur Verfügung steht, ist diese Funktion in dieser Arbeit nicht zur Verwendung gekommen. Das exportierte Modell konnte nur über 3ds Max® oder Twinmotion importiert werden. Beide Lösungen wurden in Tests ausprobiert und andere Möglichkeiten wurden nicht mehr in Betracht gezogen.

Mit 3ds Max® kann das FBX-Modell von InfraWorks® importiert und mit Hilfe des Datasmith Plugins in eine Datasmith-Datei exportiert werden. Wenn keine weiteren Modellierungsarbeiten mit 3ds Max® nötig sind, ist es aus Kostengründen aber besser, mit Twinmotion 2023 zu arbeiten. Deshalb wurde für die Arbeit die Entscheidung getroffen, das Modell über Twinmotion zu importieren. Twinmotion bietet hierbei auch noch weitere Vorteile gegenüber 3ds Max®, wie in Kapitel 2.1.4 beschrieben. Mit Twinmotion 2023 wurde der Datasmith-Export direkt in die Anwendung integriert. So ist es möglich, das Modell ohne weitere Plugins für Twinmotion zu exportieren, siehe Abbildung 10, und dieses mit Hilfe des Datasmith Importer Plugins für Unreal Engine

zu importieren. Somit wurde für den gesamten Workflow die Software, wie in Abbildung 8 dargestellt, ausgewählt.



Abbildung 8: Software



Nach erfolgreichem Import des Modells in Unreal Engine wurden auch die grundlegenden Funktionalitäten der Anwendung in das Testmodell Unterschleißheim implementiert, wie in Kapitel 3.1 beschrieben. Erwähnenswert wäre hier, dass in diesem Testmodell beim Wechsel zwischen den einzelnen Modellen das aktuelle Modell, in dem sich der Avatar befand, nur ausgeblendet und das nachfolgende Modell eingeblendet wurde. Aufgrund von Performanceproblemen im Modell Oberschleißheim wurde für jedes Modell ein eigenes Level erstellt. Beim Wechsel zwischen diesen Leveln wird so der Avatar oder die Drohne mit den letzten Standortkoordinaten aus dem vorherigen Level im neuen Level respawned.

3.2 Arbeitsschritte (Workflow)

3.2.1 InfraWorks® Workflow

Abbildung 9 zeigt eine Übersicht der Arbeitsschritte in InfraWorks®. Abbildung 34 zeigt das fertige Modell in InfraWorks®.

1. Nach der Auswahl der ATKIS®-Daten in QGIS, siehe Kapitel 3.1.1, und Überlegungen, wie die unterschiedlichen Verkehrswege und Nutzungsflächen visualisiert werden sollen, wird eine generalisierte Zuordnung vorgenommen (z. B. Autobahn, Landstraße, Straßen mit Gehwegen, Straßen mit Gehwegen und Beleuchtung, Feldweg, Fahrradweg usw.; Wiese, Acker, Felder, Waldboden, Wohngebiete usw.). Diese Vorüberlegung hilft bei der Erstellung und Zuordnung der Modelle für die Verkehrswege und der Materialien der

- Nutzungsflächen und trägt gleichzeitig zu einer übersichtlichen Zuordnung der ATKIS®-Verkehrswege in InfraWorks® bei. Es ist anzumerken und für die spätere Arbeit in Unreal Engine wichtig, dass die Zuordnung der Materialien für die Nutzungsflächen gut überlegt sein sollte, da beim Export alle Flächen mit gleichem Material zusammenhängen. Bei der endgültigen Zuweisung der Materialien für die Nutzungsflächen in Unreal Engine kann so enorm viel Zeit gespart werden. Durch die Verschmelzung aller Flächen gleichen Materials beim Export müssten diese sonst bei einer nachträglichen Änderung aufwändig aufgetrennt werden, um ihnen die passenden Materialien zuweisen zu können.
2. Erstellung eines neuen Projektes in InfraWorks® und Import von DGM, LoD2-Gebäuden und eines aktuellen Digitalen Orthophotos (DOP). Die LoD2-Gebäude müssen als Point Feature importiert werden, als Import Type wurde City Furniture ausgewählt. Andernfalls stimmt die Position der Gebäude im Modell nicht.
 3. Erstellung der Modelle (Styles) für die ausgewählten Verkehrswege in InfraWorks®.
 4. Import der ausgewählten ATKIS®-Nutzungsflächen und Zuweisung der Materialien für diese. Import aller stehenden Gewässer mit Materialzuweisung. Import der ausgewählten ATKIS®-Verkehrswege und Zuweisung der Styles.
 5. Anpassung aller ATKIS®-Daten, die eine Verdrängung aufweisen. Manuelle Anpassungen am ganzen Modell, orientiert an einem aktuellen DOP. Hierfür müssen alle Nutzungsflächen über das Surface Layers Panel ausgeblendet werden.
 6. Erstellung der Modelle (Styles) für alle Brücken, Unterführungen, Bahnübergänge usw. und anschließende Zuordnung der Styles im InfraWorks®-Modell.
 7. Einzeichnen fehlender benötigter Strukturen, die nicht von den ATKIS®-Daten abgedeckt sind, wie Verkehrswege, Flüsse, Nutzungsflächen und Stehgewässer.
 8. Export des Modells im FBX-Format. Dabei ist es wichtig, das Modell in mehrere Dateien zu splitten. InfraWorks® bietet die Möglichkeit, Ground (Erdoberfläche bestehend aus DGM, Nutzungsflächen und Verkehrswegen), City Furniture (hier LoD2-Gebäude) und Linear Decorations (nachträglich eingefügte 3D-Modelle wie z. B. Ampeln oder Bahnschranken) separat zu exportieren. So können diese Elemente in der Unreal Engine separiert angesprochen werden. Anschließend muss das DGM mit jedem Luftbild oder jeder Karte, die später Verwendung finden sollen, nochmal exportiert werden. Hierfür können alle nicht benötigten Surface Layer (ATKIS®-Daten, Rasterdaten und zusätzlich erstellte Feature) in InfraWorks® deaktiviert werden und beim Export wird nur der Ground exportiert.

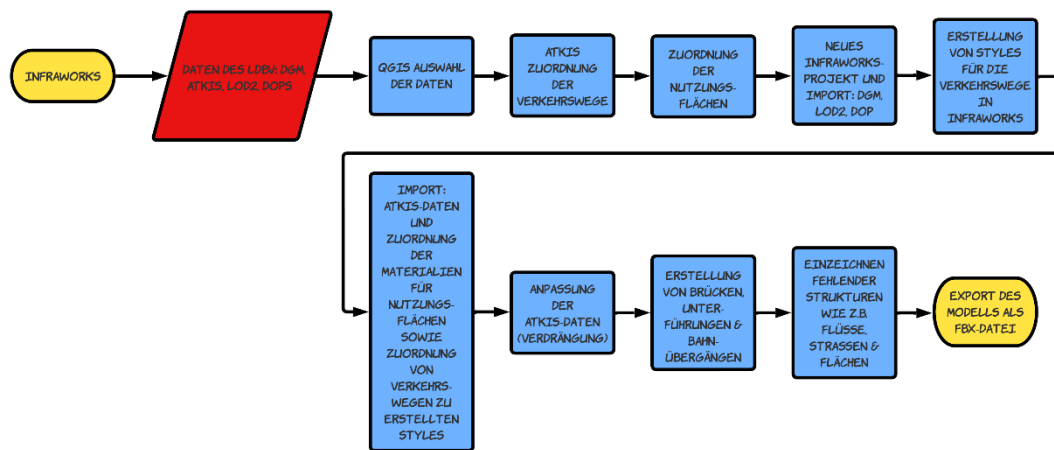


Abbildung 9: Workflow InfraWorks®



3.2.2 Twinmotion Workflow

Abbildung 10 zeigt eine Übersicht der Arbeitsschritte in Twinmotion. Abbildung 35 zeigt das Modell in Twinmotion.

1. Import aller FBX-Dateien aus dem InfraWorks®-Export für jedes Modell separat. Dabei soll die Hierarchie der Datei erhalten bleiben und bei einem entstehenden Materialkonflikt sollen beide Materialien behalten werden.
2. Export als Datasmith-Datei.

Bei Testversuchen mit Twinmotion 2022 war es nur möglich, das importierte Modell im Twinmotion-Dateiformat abzuspeichern und es mit dem Twinmotion Plugin für Unreal Engine aus der abgespeicherten Twinmotion-Datei zu importieren. Hierbei gehen aber die Vorteile von Datasmith verloren. Im Testmodell Unterschleißheim, das wesentlich kleiner war als das fertige Modell von Oberschleißheim, entstanden hierbei keine Probleme. Im späteren Modell von Oberschleißheim wurde durch das Datasmith-Format ein deutlicher Performancegewinn erreicht. Dieser entstand durch Reduzierung der Videospeicherauslastung aufgrund optimierter Texturen und verbesserter Integration der Texturen in die Geometrien. So entstanden deutliche Verbesserungen in der Frame Rate während der Laufzeit der Anwendung. Auch die Dauer des Imports des Modells konnte dadurch von mehreren Stunden auf wenige Minuten verkürzt werden.

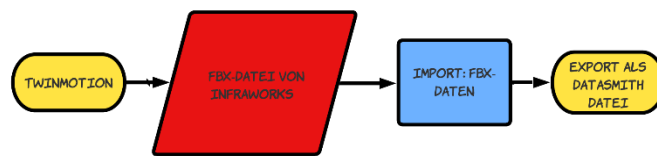
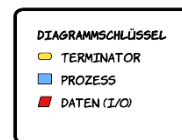


Abbildung 10: Workflow Twinmotion



3.2.3 Unreal Engine Workflow

Abbildung 11 zeigt eine Übersicht des gesamten Ablaufes der Arbeit in Unreal Engine, ein detaillierterer Ablauf kann zusätzlich Abbildung 12 entnommen werden. Abbildung 36 zeigt das fertige Projekt im Unreal Engine Editor.

Mit Unreal Engine 5 wurden Lumen und Nanite eingeführt. Mit dem Lumen Global Illumination System und dem Nanite Rendering von Unreal Engine konnten Performance sowie Qualität der Visualisierungen in Echtzeitanwendungen immens gesteigert werden. Sie sind in Unreal Engine 5 als Default Settings im Projekt aktiviert. Beide Methoden benötigen aber auch eine hohe Hardwareperformance siehe Kapitel 5.1. Bei größeren Umgebungen oder verringerter Hardwareleistung ist es eventuell nötig, auf Schatten, die während der Laufzeit in Echtzeit generiert werden, zu verzichten und diese stattdessen in die Texturen des Modells zu integrieren. Das Verfahren, welches dies ermöglicht, ist das Light Baking. Allerdings kann dieses in Unreal Engine 5 mit Lumen derzeit nicht realisiert werden. Deshalb muss am Ende der Arbeit eventuell eine andere Beleuchtungsmethode als Lumen verwendet werden und für das Light Baking die Beleuchtung des Modells berechnet und in Light Maps gespeichert werden. In der vorliegende Arbeit wurde mit Lumen und Nanite gearbeitet.

1. **Projekt- und Engine-Vorbereitungen** wie die Auswahl der grundlegenden Projekteinstellungen und die Installation der benötigten Plugins aus dem Epic Games Store. Sicherung und Packen des Projekts für die fertige Anwendung. Anlegen einer sinnvollen Ordnerstruktur.

Das Anlegen einer sinnvollen Ordnerstruktur, z. B. gegliedert für unterschiedliche Level, Blueprints und andere Themenbereiche, ist wichtig, um das Projekt übersichtlich zu halten. Nachträgliche Veränderungen an der

Ordnerstruktur können dann durch entstandene ungelöste Abhängigkeiten der Dateien Probleme verursachen, die oft nur mit hohem Arbeitsaufwand gelöst werden können.

- a. Plugins installieren: Benötigte Plugins aus dem Epic Games Store installieren. Dazu zählt das Async Loading Screen Plugin (Truong Bui, n.d.), und falls mit Twinmotion-Dateien gearbeitet wird, Datasmith Twinmotion Importer Plugin (*Datasmith Twinmotion Importer Plugin (Beta 4) in Code Plugins - UE Marketplace*, n.d.) und Twinmotion Content Plugin (*Twinmotion Content for Unreal Engine Plugin (Beta 4) in Code Plugins - UE Marketplace*, n.d.). Die Twinmotion Plugins wurden nach dem Umstieg auf Twinmotion 2023 nicht mehr benötigt. Alle weiteren Plugins können direkt aus der Plugin-Bibliothek in der Unreal Engine aktiviert werden.
 - b. Projekt erstellen: Neues Projekt erstellen, dazu gewünschte Vorlage auswählen. Für diese Arbeit wurde aus den Templates Games>Third Person gewählt und das Projekt mit den Default Settings für Blueprints erstellt.
 - c. Plugins aktivieren: Nach dem Erstellen des Projekts müssen die benötigten Plugins aktiviert werden. Dazu zählen zunächst das Async Loading Screen Plugin und das Datasmith Importer Plugin.
 - d. Projekteinstellungen: Die Default Settings für das Projekt werden übernommen und im Verlauf der Arbeit nach Bedarf angepasst.
 - e. Sicherung des Projekts: Sicherungen des Projekts sollten nach jedem Arbeitstag oder größeren Änderungen durchgeführt werden. Um das Projekt ohne Probleme wiederherstellen zu können, muss das gesamte Projekt gesichert werden. (Pfad: Documents/Unreal Projects/[Projectname])
 - f. Packen des Projekts als .exe: Das Packen (Exportieren) des Projekts stellt den finalen Schritt dar, um die Anwendung auszuliefern oder zu testen. Dieser Schritt sollte wie die Sicherung des Projektes nach jedem Arbeitstag oder größeren Änderungen durchgeführt werden. Wenn hier Probleme auftreten, kann die mögliche Ursache leichter eingegrenzt werden. Während der Entwicklungsphase wird das Projekt im Development Modus gepackt (Unreal Engine Default Settings).
2. **Level erstellen und Modell importieren** sowie grundlegende Einstellungen an den Leveln und anschließende Fertigstellung des Modells für die Anwendung.
- a. Level erstellen: Leeres Level zum Projekt hinzufügen. Für jedes Modell wird ein separates Level erstellt. Im Projekt sind insgesamt sieben Modelle vorhanden.
 - b. Licht und visuelle Effekte: Das erstellte Level beinhaltet keine Beleuchtung oder Atmosphäre, daher müssen diese Elemente zuerst hinzugefügt werden, um im Viewport etwas sehen zu können. Hierfür werden grundlegende Elemente für die Beleuchtung und die visuellen Effekte für eine realistische Beleuchtung zum Level hinzugefügt:

Directional Light (die Sonne)

Skylight (Streulicht durch Lichtbrechung der Atmosphäre)

Sky Atmosphere (Atmosphäre, die das Sonnenlicht bricht und den Himmel z. B. blau erscheinen lässt. Ist wie die Hemisphäre der Umgebung, in der sich das Modell befindet.)

Volumetric Cloud (Wolken in der Atmosphäre)

Exponential Height Fog (Bodennebel, in der Höhe veränderbar; ermöglicht ein sanftes Überblenden zwischen Himmel und Boden und kann so die Weitsicht im Projekt verbessern. Das Überblenden findet dann zwischen der Sky Atmosphere, die wie eine Hemisphäre fungiert, und der hypothetischen, gegenüberliegenden Hemisphäre statt.)

PostProcessVolume (Einstellungen der Beleuchtung und visueller Effekte; kann auf die ganze Welt angewendet werden oder durch mehrere PostProcessVolume auf bestimmte Bereiche im Spiel. Wenn das Volumen nicht für die ganze Welt angewendet wird, treten die Einstellungen erst in Kraft, wenn das Volumen im Spiel betreten wird. Die Einstellungen im PostProcessVolume sind eingestellten Werten der einzelnen Elemente oder den Projekt Settings übergeordnet.)

- c. World Settings: Unter dem Punkt GameMode Override wird BP_ThirdPersonGameMode ausgewählt. So kann das Spiel mit dem Avatar in der Third Person-Ansicht gestartet werden. Beim Testen kann man sich so mit dem Avatar durch das Level bewegen.
- d. Licht und visuelle Effekte Mobility: Die Mobility kann für die einzelnen Elemente verändert werden und beeinflusst, wie das Licht oder visuelle Effekte berechnet werden. Je nach Wahl der Beleuchtungsmethode und zur Verbesserung der Performance muss diese eventuell umgestellt werden, um Light Maps erstellen zu können. Grundsätzlich zu unterscheiden ist hierbei zwischen:
 - Static:** Licht und Schatten in der Szene werden nur einmal berechnet und in sogenannte Light Maps gespeichert (geringe Hardwareperformance).
 - Stationary:** Licht und Schatten können in der Szene für Objekte, die sich nicht bewegen, als Light Maps gespeichert werden, während das Licht für Objekte, die sich bewegen, während der Laufzeit dynamisch in Echtzeit berechnet wird.
 - Movable (Default in Unreal Engine 5):** Licht und Schatten in der Szene werden dynamisch für alle Objekte im Spiel während der Laufzeit in Echtzeit berechnet (höchste Hardwareperformance nötig).
- e. Import Datasmith Modell: Mithilfe des Datasmith Plugins wird das Modell aus dem Export von Twinmotion importiert. Einstellungen bleiben hierbei auf Default.
- f. Collision hinzufügen: Nach dem Import hat das Modell kein Collision Model. Der Avatar fällt beim Start durch das Modell. Alle Objekte und Meshes im Modell, denen ein Collision Model zugewiesen werden soll, werden hierfür ausgewählt und können über ein Bulk Edit gleichzeitig bearbeitet werden. Damit die Collision akkurat funktioniert, muss die

- Collision Complexity auf Use Complex Collision As Simple gestellt werden. Nach diesem Schritt ist das Modell vollständig begehbar.
- g. LoD2-Anpassungen: Entfernen von Brücken aus den LoD2-Daten. Einfügen von Ausschnitten in die LoD2-Gebäude für benötigte Hausdurchgänge, die nicht in den LoD2-Gebäuden vorhanden sind.
 - h. Anpassungen des Modells: Beim Import des Modells ist ein Fehler bezüglich der Fahrbahnmarkierung aufgetreten. Diese schwebte ca. 40cm über dem Grund und musste um diesen Wert in der Z-Achse verschoben werden. Dadurch, dass beim Export alle Oberflächen mit gleichen Materialien zusammengefügt werden, muss bei gewünschter Änderung von Materialien eine Separierung der betreffenden Stellen im Mesh oder von Objekten erfolgen. Nach der Separierung kann das Objekt oder Mesh einzeln angesprochen werden.
 - i. Zusätzliche Modellierung: Im Projekt sind einige Brunnen enthalten, die nachträglich eine Umrandung erhalten haben, um die Brunnen besser von der Umgebung abgrenzen zu können.
 - j. Nanite aktivieren: Wenn Nanite für Objekte im Modell aktiviert werden soll, kann dies gleichzeitig für die gewünschte Auswahl geschehen. Beim Modell selbst muss aber darauf geachtet werden, dass die einzelnen Objekte keine unterschiedlichen LoD-Modelle besitzen, wie es bei der Vegetation der Fall ist. Nanite bringt hier somit keine Verbesserung in der Visualisierung mit sich, aber eventuell eine bessere Performance. Nanite kann jederzeit für einzelne Objekte aktiviert und deaktiviert werden. Nanite wurde für alle Objekte im Modell aktiviert, die keine hochgenaue Collision benötigen. Da durch Nanite kein hochgenaues Collision-Modell verwendet werden kann, würde der Avatar sonst z. B. durch Lücken durch das Modell fallen oder bei Gebäudemodellen durch die Wände gehen.
 - k. Quixel Bridge Material: Über die Quixel Bridge Library, die in Unreal Engine integriert ist, können eine Vielzahl von hochwertigen Materialien und fertigen 3D-Modellen für die Gestaltung des Modells kostenlos heruntergeladen und einfach im Projekt integriert werden. Dafür wurde für alle Oberflächen im Modell eine passende Auswahl an Materialien getroffen und diese anschließend Verkehrswegen und Nutzungsflächen zugewiesen.
3. **Blueprints** sind als visuelle Scriptsprache eine Alternative zur klassischen Programmierung mit einer Programmiersprache und werden im Projekt für die Umsetzung von Funktionen und Interaktionsmöglichkeiten anstelle der klassischen Programmierung verwendet, siehe Kapitel 3.2.4.
 4. **Foliage (Vegetation)** zum Projekt hinzufügen.
 - a. Quixel Bridge Foliage: Aus der Quixel Bridge Library können Vegetation und sonstige 3D-Modelle heruntergeladen und im Modell eingesetzt werden.
 - b. Epic Marketplace Megascans: Megascans ist eine Onlinebibliothek mit hochwertigen 3D-Modellen und Vegetation. Die eingesetzten Bäume

- stammen aus der Megascans Library. Diese können aus dem Epic Game Store bezogen und von dort direkt dem Projekt hinzugefügt werden.
- c. Setup Foliage: Einstellung der einzelnen Pflanzen und Bäume. Nach der Auswahl der Vegetation können umfangreiche Anpassungen am Foliage vorgenommen werden. Bei Bäumen können z. B. Jahreszeit, Windstärke, oder Gesundheitszustand angepasst werden. Diese Anpassungen können über einen Foliage Actor auch für alle Modelle aus der Quixel Bridge oder der Megascans Library gleichzeitig erfolgen. Es sind aber auch Änderungen an jedem einzelnen Objekt möglich.
 - d. Nanite aktivieren: Bei Aktivierung von Nanite werden für die Vegetation keine LoD-Modelle verwendet, die normalerweise zum Einsatz kommen würden, um in der Ferne liegende Vegetation performanceschonend darzustellen. So wird ein Umschalten zwischen den LoD-Modellen, das normalerweise deutlich sichtbar wäre, verhindert. Nanite benötigt dadurch aber auch ein gewisses Maß an Hardwareperformance, auch wenn die Technologie hinter Nanite sehr performanceschonend ist. Nanite wird im gesamten Projekt für jegliche Vegetation eingesetzt.
 - e. Foliage hinzufügen: Mit dem Unreal Engine Foliage Tool steht ein mächtiges Tool zur Verfügung, mit dem Vegetation in das Modell eingefügt werden kann. Es können z. B. unterschiedliche Baumarten in zufallsgenerierten Größen in das Level eingefügt werden. So kann auch ein realistischeres Bild bei der Verteilung der Vegetation im Modell erzeugt werden.
5. **Sonstiges: Animationen und Camera Renderings** für abschließende Visualisierungen im Projekt. Unreal Engine ist eine mächtige Anwendung für Video- und Bildrenderings und wird auch in der Filmindustrie, Architektur oder in Produktvisualisierungen eingesetzt.
- a. Animationen: Animierte Objekte im Modell wie Flugzeuge, Autos, Fußgänger usw., die sich durch das Modell bewegen. Animationen für sonstige Darstellungen. Nutzung während der Laufzeit der Anwendung, oder auch für Videos.
 - b. Camera Renderings Videos: z. B. Präsentationszwecke oder benötigte Zwischensequenzen im Spiel.
 - c. Camera Renderings Bilder: z. B. für Präsentationszwecke benötigte Bilder.
6. **Abschließende Einstellungen** des Projekts.
- a. Bake Light?: Je nach Anwendungszweck müssen eventuell noch performanceverbessernde Maßnahmen getroffen werden. Lumen als Beleuchtungsmethode ist sehr performanceintensiv. Wenn ein Erstellen von Light Maps gewünscht ist, um die Performance zu erhöhen, muss Lumen momentan noch durch eine andere Beleuchtungsmethode ersetzt werden, siehe Kapitel 5.1.
 - b. Texture Streaming Poolsize?: Im Allgemeinen sollten alle Texturen im Projekt für die Anforderungen in der Auflösung angepasst werden, um die bestmögliche Performance bieten zu können und

ressourcenschonend mit der Hardware umzugehen. Gerade bei einem großen Projekt treten schnell Probleme mit der Größe des Videospeichers auf. Der Texture Streaming Poolsize reicht dann nicht aus, um alle Texturen in den Videospeicher der Grafikkarte zu laden. Dies resultiert in einer rechenintensiven Echtzeitreduzierung der Texturauflösung und gleichzeitig in einer schlechteren Qualität der Texturen während der Laufzeit der Anwendung. Daher sollten alle Texturen der Materialien in der Auflösung angepasst werden, siehe Kapitel 5.1. Hierfür können auch alle Texturen zusammen über ein Bulk Edit angepasst werden.

- c. Packen des Projekts: Für das finale Packen des Projekts werden noch ein paar Einstellungen in den Project Settings getroffen.
Unter Project>Packaging wird die Build Configuration von Development auf Shipping gestellt und ein Haken bei Full Rebuild gesetzt.
Unter Project>Descriptions können noch Projektname, Beschreibungen für die Lizenzierungen und weitere Beschreibungen eingefügt werden.
Unter Project>Maps & Modes wird die Game Default Map ausgewählt, das Level, welches beim Start als erstes geöffnet wird. Das ist normalerweise das Main Menu.
Unter Project>Supported Platforms können die Betriebssysteme gewählt werden, auf denen die Anwendung laufen soll.
Unter Project>Target Hardware wird ausgewählt, auf welcher Hardware die Anwendung verwendet werden soll.
Unter dem Punkt Platforms können noch vom Betriebssystem abhängige Einstellungen getroffen und ein Game Icon eingefügt werden. Für die vorliegende Anwendung wurde Windows als Plattform gewählt.
Anschließend wird die Anwendung im Shipping Modus gepackt.

Detaillierte Informationen zu den getroffenen Einstellungen oder Funktionen einzelner Bestandteile der Unreal Engine-Entwicklungsumgebung können der Unreal Engine-Dokumentation entnommen werden. (*Unreal Engine 5.1 Documentation*, n.d.)

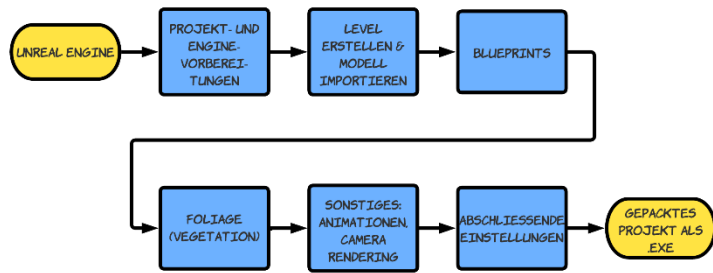


Abbildung 11: Workflow Unreal Engine

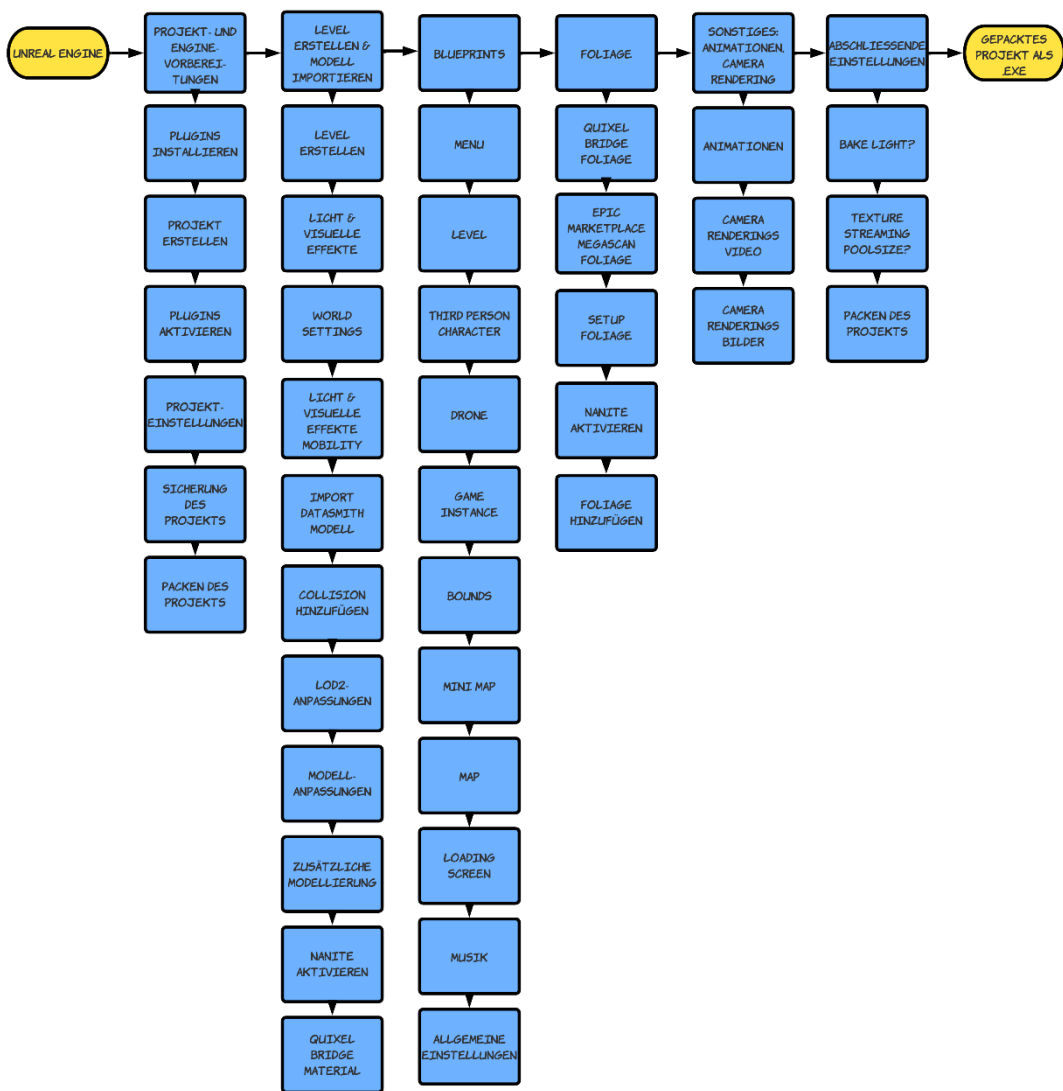


Abbildung 12: Workflow Unreal Engine, detaillierter Ablauf

3.2.4 Unreal Engine Blueprints

Für alle Flowcharts in Kapitel 3.2.4 gilt Abbildung 13.

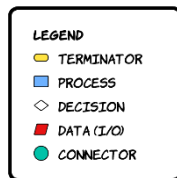


Abbildung 13: Legend Blueprints

Begriffserklärungen für die Blueprints in Kapitel 3.2.4.1 bis Kapitel 3.2.4.11.

Blueprint (BP)

Ein Blueprint ist ein visuelles Skriptsystem, das es ermöglicht, ohne Programmierkenntnisse z. B. Logik, Steuerung, Interaktion mit Objekten, Programmabläufe oder Abläufe im User Interface (UI) usw. zu definieren.

Widget Blueprint (WBP)

Ein Widget Blueprint ist wie das Blueprint ein visuelles Skriptsystem, das speziell für das UI zur Verwendung kommt. Es ermöglicht die Gestaltung des Designs von Menüs oder Anzeigen im Spiel mit Hilfe eines visuellen Editors.

Input Modes

Legt fest, wie Benutzer:inneneingaben verarbeitet werden. Die korrekte Auswahl bei der Programmierung ist wichtig, damit Nutzer:innen mit der Anwendung interagieren können.

Game Only: Der Fokus liegt auf der Steuerung des Spiels, um z. B. den Player zu steuern. Es wird normalerweise kein Mouse Cursor angezeigt. Interaktionen mit dem User Interface sind in der Standardeinstellung nicht möglich.

UI Only: Der Fokus liegt auf der Interaktion mit dem User Interface. Die Steuerung des Players ist nicht möglich. Es wird normalerweise ein Mouse Cursor angezeigt, um mit dem User Interface interagieren zu können.

Game and UI: Der Fokus liegt darauf, während des Spiels Eingaben im User Interface machen zu können. Bei der Aktivierung kann das User Interface bedient werden, aber auch die Steuerung des Players ist noch möglich. Um zu verhindern, dass Falscheingaben in Menüs an den Player weitergegeben werden, wird das Spiel in der vorliegenden Anwendung beim Öffnen des Menüs pausiert. Je nach Art der Anwendung kann aber auch eine gemischte Form der Eingabe erwünscht sein.

Game Settings

Über die Game Settings können eine Reihe von verschiedenen Einstellungen der Anwendung vorgenommen werden, z. B. Lautstärke, Screen Resolution, Frame Rate, V-Sync und noch weitere Grafikqualitäts- und Spieleinstellungen. Unreal Engine bietet hierfür mehrere vorgefertigte Funktionen, mit denen die Einstellungen über Blueprints einfach geändert werden können.

Viewport

Der Viewport ist im Level Editor der Bereich, in dem die Anwendung visuell dargestellt wird. In ihm wird das Level oder UI in einer Echtzeitvorschau gezeigt und er bietet Interaktionsmöglichkeiten mit z. B. Objekten im Level.

Level Editor

Der Level Editor ist der Überbegriff für das User Interface der Entwicklungsumgebung der Unreal Engine.

Player Location and Rotation

Die Player Location and Rotation setzt sich aus mehreren Koordinaten und Drehwinkeln für die Position des Avatars und der Camera im Level zusammen.

Player Location: Beschreibt die X-, Y- und Z-Koordinaten, auf denen sich der Avatar befindet.

Player Rotation: Beschreibt die X-, Y- und Z-Drehwinkel, die der Avatar im Spiel einnimmt.

Camera Location: Beschreibt die X-, Y- und Z-Koordinaten, auf denen sich die Player Camera befindet (im Third Person Modus hinter dem Player).

Camera Rotation: Beschreibt die X-, Y- und Z-Drehwinkel, die die Player Camera im Spiel einnimmt (Blickrichtung).

Start Position

Die Start Position bezeichnet den Punkt im Level, an dem der Player am Anfang des Spiels startet.

Game Instance

Die Game Instance ist ein high-level Manager, der während der Laufzeit des Spiels z. B. Variablen, Funktionen und Events level- oder objektübergreifend speichert und ausführt.

Weiter Informationen können der Unreal Engine-Dokumentation entnommen werden. (*Unreal Engine 5.1 Documentation*, n.d.)

Sonstige Vorbereitungen vor der Arbeit an den Blueprints:

Allgemeine Einstellungen

In den Editor Preferences kann unter General>Blueprint Editor Settings unter dem Tab Experimental eine Funktion zur Deaktivierung von einzelnen Nodes im Blueprint Editor aktiviert werden. Dadurch können einzelne Nodes beim Testen von Blueprints deaktiviert werden.

Musik

Zum Abspielen von Hintergrundmusik wird ein Sound Class Mix erstellt, in dem die Systemlautstärke geregelt werden kann und in dem auch weitere Parameter zur Soundeinstellung vorhanden sind. Zusätzlich wird ein Sound Cue angelegt, in dem Musiktitel platziert werden können, die abgespielt werden sollen. Über ein Custom Event in der Game Instance kann so das Abspielen der Musik von jedem Level aus erfolgen.

Loadingscreen

Das Async Loading Screen Plugin ermöglicht es, Loadingscreens asynchron während des Ladens eines Levels anzuzeigen. Erst wenn das Level vollständig geladen ist, wird der Loadingscreen vom Viewport entfernt. Mit Hilfe des Plugins können auch Intro-Videos zu Beginn der Anwendung abgespielt werden. Die Loadingscreens können individuell für jedes Level mit verschiedenen Texten, Bildern oder Sequenzen angepasst werden. So müssen beim Wechsel eines Levels nur die Indexnummern der zuvor erstellten Texte, Bilder oder Sequenzen aufgerufen werden, um den gewünschten Loadingscreen anzuzeigen. Siehe Abbildung 51 bis Abbildung 59.

3.2.4.1 Main Menu

Abbildung 14 zeigt den Aufbau und Ablauf im Main Menu BP und WBP. Abbildung 37 zeigt das Main Menu.

Das Main Menu besteht aus einem extra Level und einem WBP. Im Level BP wird die Hintergrundmusik aktiviert, der Input Mode für das Level wird auf UI Only gesetzt, die Einstellungen aus dem Options Menu werden geladen und das WBP wird zum Viewport hinzugefügt. Falls das Spiel zum ersten Mal gestartet wurde, wird ein Hardware Benchmark durchgeführt und die Einstellungen für das System werden automatisch gesetzt und gespeichert.

Das Main Menu WBP beinhaltet vier ausführbare Buttons:

1. Button Start Game: Die Einstellungen für das Loadingscreen Plugin werden gesetzt und der Loadingscreen gestartet. Das Main Menu WBP wird vom

Viewport entfernt. Das Start Level wird im Hintergrund geladen und der Input Mode auf Game Only gestellt.

2. Button Options Menu: Das Main Menu WBP wird vom Viewport entfernt und das Options Menu WBP zum Viewport hinzugefügt.
3. Button Credits: Das Main Menu WBP wird vom Viewport entfernt und das Credits WBP zum Viewport hinzugefügt. Abbildung 39 zeigt das Credits WBP.
4. Button Quit Game: Das Main Menu WBP wird vom Viewport entfernt und das Spiel beendet.

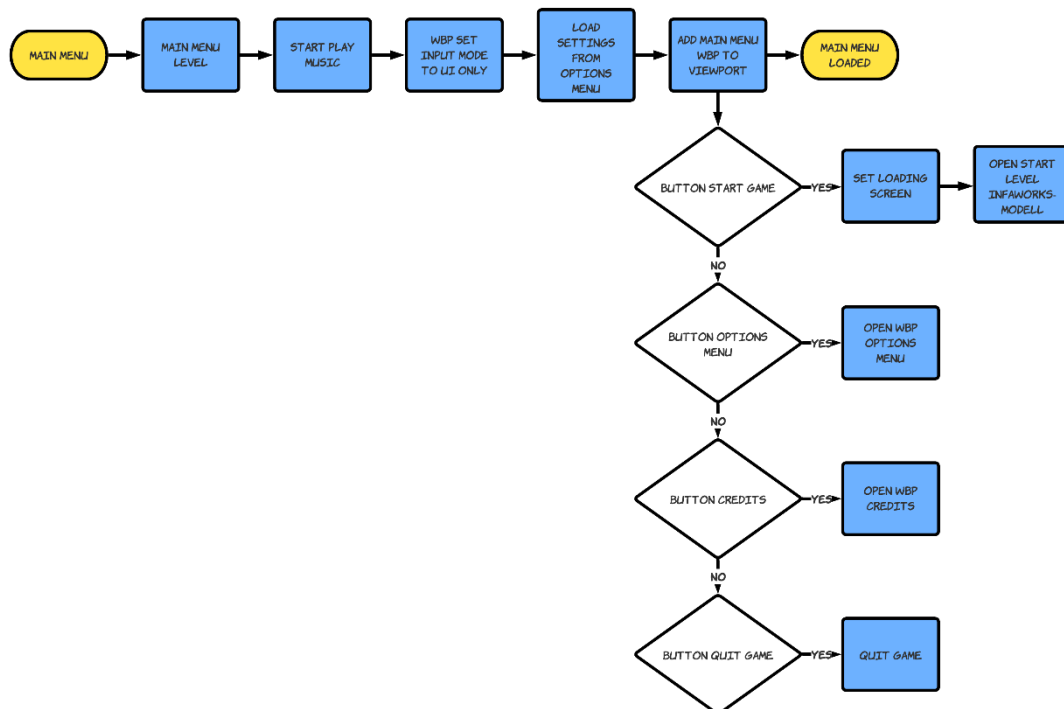


Abbildung 14: BP & WBP Main Menu

3.2.4.2 Options Menu

Abbildung 15 bis Abbildung 17 zeigen den Aufbau und Ablauf im Options Menu WBP. Abbildung 38 zeigt das Options Menu.

Das Options Menu ist ein WBP. Beim Öffnen des Options Menu werden gespeicherte Spieleinstellungen geladen und angewendet. Bildschirmauflösungen vom PC werden in eine Combo Box geladen. Für die Darstellung im Viewport werden alle Buttons, die in den Game Settings eingestellt sind, mit einem Highlight versehen.

Mit einem Schieberegler kann das Master Volume der Sound Class angepasst werden, um die Lautstärke im Spiel zu regulieren. Der Wert des Schiebereglers kann zwischen

0 und 1 verstellt werden und wird direkt in der Sound Class angewendet. Damit der Schieberegler beim erneuten Öffnen des Options Menu seinen Wert beibehält und die eingestellte Lautstärke anzeigt, wird dieser zusätzlich in der Game Instance als Variable gespeichert und beim erneuten Öffnen des Options Menu von dort geladen. Allerdings bleibt dieser Wert nur während der Laufzeit der Anwendung bestehen und wird beim Beenden der Anwendung wieder zurückgesetzt.

Ausführbare Elemente im Options Menu WBP:

1. Combo Box Change Resolution: Über ein Dropdown Menü wird die Auflösung ausgewählt. Die Auswahl wird in den Game Settings gespeichert und geladen.
2. Buttons für die Game Settings: Der ausgewählte Button wird in den Game Settings gespeichert. Der Button wird mit einem Highlight im Viewport dargestellt. Die Einstellungen aus den Game Settings werden geladen.
Zu diesen Buttons gehören:
 - a. Buttons Window Mode: Umschalten zwischen Fullscreen und Windowed Mode.
 - b. Buttons Frames per Second (FPS) Limit: Limitiert die Frame Rate.
 - c. Buttons View Distance: Stellt die Grenze ein, ab wann Objekte in der Anwendung ausgeblendet werden. Funktion ist nur gegeben, wenn Objekte einen Desired Max Draw Distance Wert besitzen und dieser nicht 0 entspricht. Die Einstellungen beeinflussen mehrere Werte gleichzeitig. Diese sind in der Datei BaseScalability.ini gespeichert (Pfad: [UE_InstallPath]/Engine/Config folder) (*Unreal Engine 5.1 Documentation Scalability Reference*, n.d.)
 - d. Buttons Post Processing Quality: Stellen die Qualität der Post Processing Effekte ein. Die Einstellungen beeinflussen mehrere Werte gleichzeitig. Diese sind in der Datei BaseScalability.ini gespeichert (Pfad: [UE_InstallPath]/Engine/Config folder) (*Unreal Engine 5.1 Documentation Scalability Reference*, n.d.)
 - e. Buttons Anti-Aliasing: Stellen die Kantenglättung in der Anwendung ein. Die Einstellungen beeinflussen mehrere Werte gleichzeitig. Diese sind in der Datei BaseScalability.ini gespeichert (Pfad: [UE_InstallPath]/Engine/Config folder) (*Unreal Engine 5.1 Documentation Scalability Reference*, n.d.)
 - f. Buttons Texture Quality: Stellen die Qualität der Texturen ein. Die Einstellungen beeinflussen mehrere Werte gleichzeitig. Diese sind in der Datei BaseScalability.ini gespeichert (Pfad: [UE_InstallPath]/Engine/Config folder) (*Unreal Engine 5.1 Documentation Scalability Reference*, n.d.)
 - g. Buttons Shadow Quality: Stellen die Qualität der Shadows ein. Die Einstellungen beeinflussen mehrere Werte gleichzeitig. Diese sind in der Datei BaseScalability.ini gespeichert (Pfad: [UE_InstallPath]/Engine/Config folder) (*Unreal Engine 5.1 Documentation Scalability Reference*, n.d.)
 - h. Buttons V-Sync: Stellen die Synchronisierung der Frame Rate mit der Bildaktualisierungsrate (Hz) des Monitors an oder aus (verhindert Tearing).

3. Button Apply Optimal Settings: Ein Hardware Benchmark wird durchgeführt und die Ergebnisse in den Game Settings gespeichert und anschließend geladen.
4. Button Save Settings: Einstellungen werden nochmal in den Game Settings gespeichert und anschließend geladen (ist theoretisch nicht notwendig).
5. Button Close WBP: Das Options Menu wird vom Viewport entfernt. Wenn das Spiel pausiert ist, wird das Pause Menu zum Viewport hinzugefügt. Wenn das Spiel nicht pausiert ist, wird das Main Menu zum Viewport hinzugefügt.

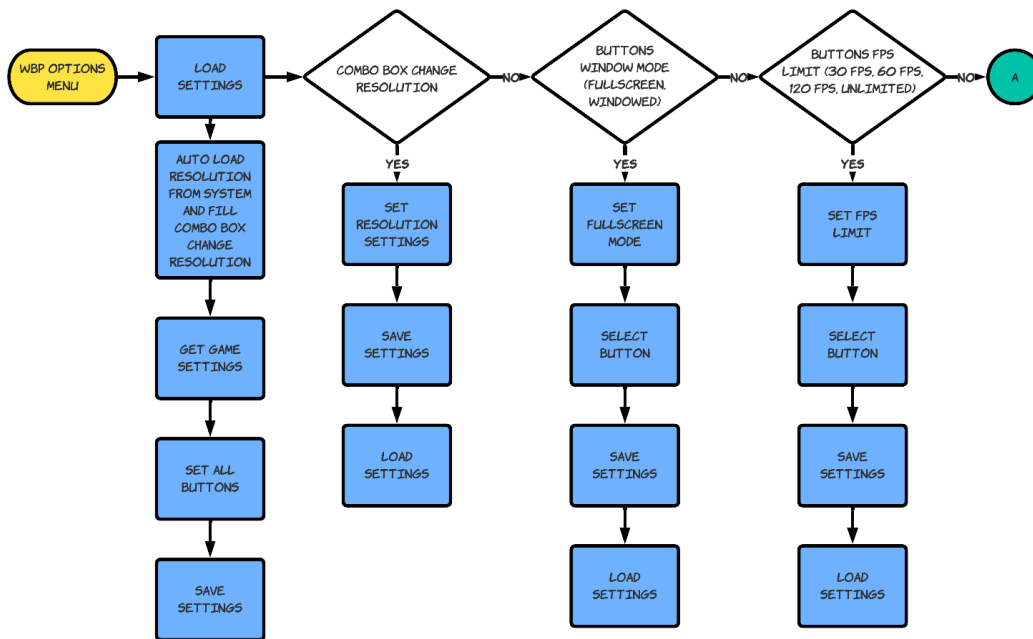


Abbildung 15: WBP Options Menu 1

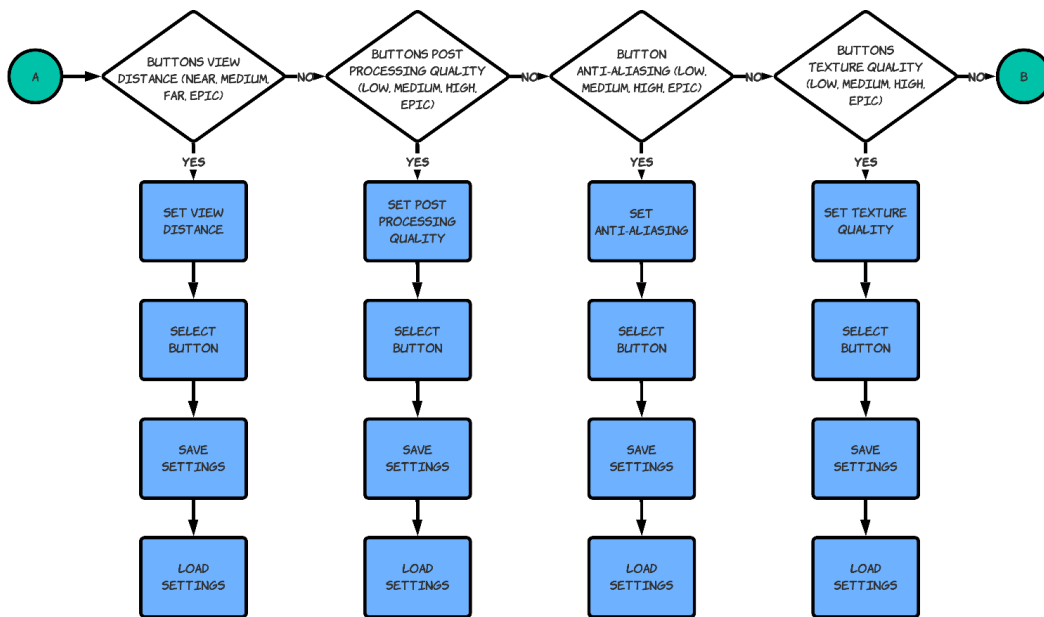


Abbildung 16: WBP Options Menu 2

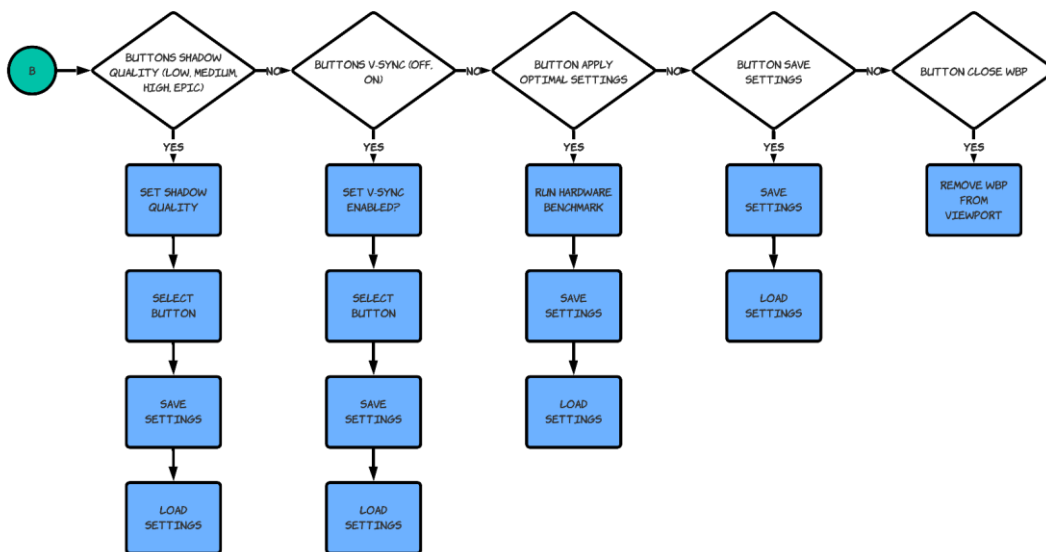


Abbildung 17: WBP Options Menu 3

3.2.4.3 Pause Menu

Abbildung 18 zeigt den Aufbau und Ablauf im Pause Menu WBP. Abbildung 40 zeigt das Pause Menu.

Das Pause Menu ist ein WBP. Beim Öffnen des Pause Menu wird das Spiel pausiert und der Input Mode auf Game and UI gesetzt. Der Mouse Cursor wird sichtbar.

Ausführbare Elemente im Options Menu WBP:

1. Button Options Menu: Das Pause Menu WBP wird vom Viewport entfernt und das Options Menu WBP zum Viewport hinzugefügt.
2. Button Keys Input: Das Pause Menu WBP wird vom Viewport entfernt und das Keys Input WBP zum Viewport hinzugefügt. Abbildung 41 zeigt das Keys Input WBP.
3. Button Return to Main Menu: Die Einstellungen für das Loadingscreen Plugin werden gesetzt und der Loadingscreen gestartet. In den Variablen der Game Instance werden aktuelle Player Position und Rotation auf die Startposition zurückgesetzt. In den Variablen der Game Instance wird die Drohne auf nicht aktiv gesetzt (falls aktiv). Das Options Menu WBP wird vom Viewport entfernt. Das Main Menu Level wird im Hintergrund geladen.
4. Button Close WBP: Der Input Mode wird auf Game Only gesetzt. Der Mouse Cursor wird ausgeblendet. Das Pause Menu WBP wird vom Viewport entfernt und die Pausierung des Spiels wird aufgehoben.

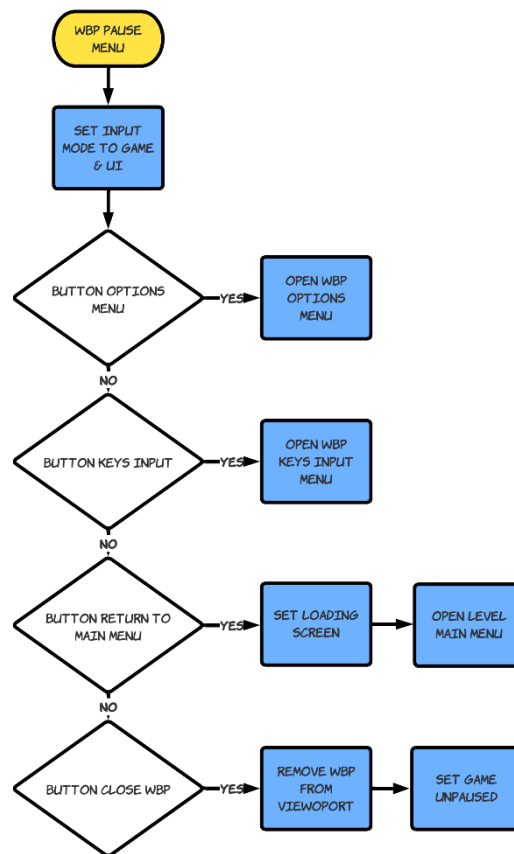


Abbildung 18: WBP Pause Menu

3.2.4.4 Level Select

Abbildung 19 und Abbildung 20 zeigen den Aufbau und Ablauf für das Level Select WBP. Abbildung 42 zeigt das Menu für den Level Select.

Das Level Select ist ein WBP. Beim Öffnen des Level Select wird das Spiel pausiert und der Input Mode auf Game and UI gesetzt. Der Mouse Cursor wird sichtbar.

Ausführbare Elemente im Level Select WBP:

1. Buttons für die sieben zur Verfügung stehenden Level: Die Einstellungen für das Loadingscreen Plugin werden gesetzt und der Loadingscreen gestartet. Prüfung in der Game Instance, ob die Drohne aktiv ist. Speicherung der aktuellen Player Location and Rotation in der Game Instance, damit der Player im neuen Level an derselben Stelle startet. Das ausgewählte Level wird im Hintergrund geladen.
2. Button Close WBP: Der Input Mode wird auf Game Only gesetzt. Der Mouse Cursor wird ausgeblendet. Das Level Select WBP wird vom Viewport entfernt und die Pausierung des Spiels wird aufgehoben.

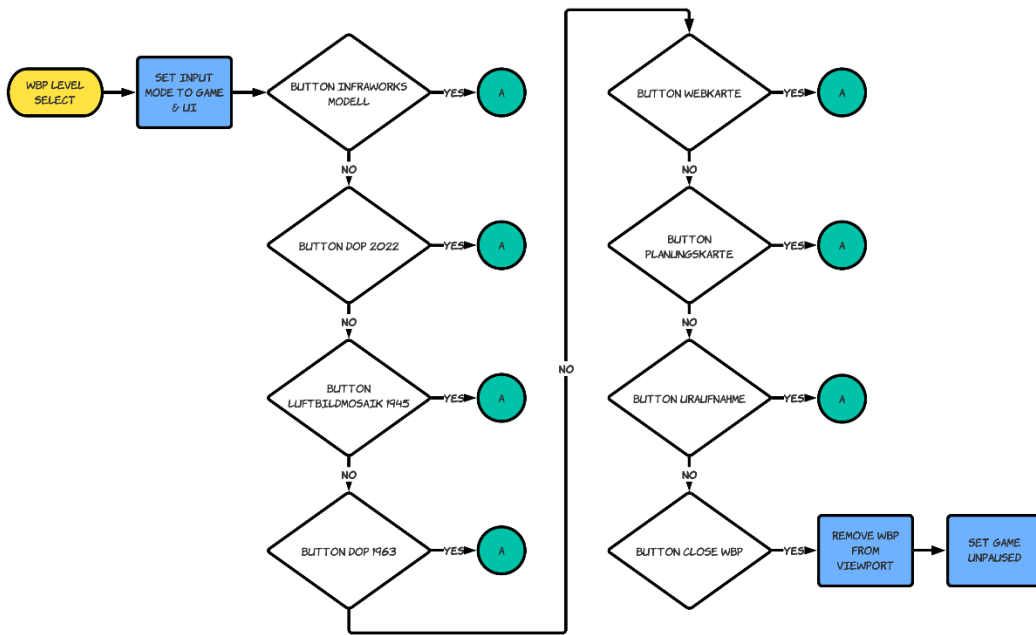


Abbildung 19: WBP Level Select 1

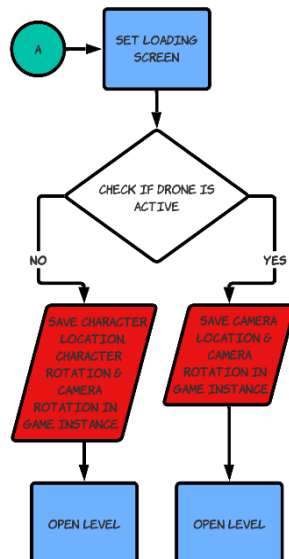


Abbildung 20: WBP Level Select 2

3.2.4.5 Level

Abbildung 21 zeigt den Aufbau und den Ablauf im Level BP.

Jedes Level verfügt über ein eigenes Level BP. Es stehen insgesamt sieben begehbare Level zur Verfügung. Siehe Abbildung 60 bis Abbildung 69.

Starten eines begehbaren Level BP:

Die Hintergrundmusik wird aktiviert. Der Input Mode wird auf Game Only gesetzt. Der Mouse Cursor wird ausgeblendet. Ein WBP mit Levelbeschriftung wird zum Viewport hinzugefügt und nach fünf Sekunden wieder vom Viewport entfernt, siehe Abbildung 43 bis Abbildung 45. Prüfung in der Game Instance, ob die Drohne aktiv ist. Falls diese aktiv ist, wird das MiniMap WBP des Third Person Character vom Viewport entfernt und das MiniMap Drone WBP für die Drohne zum Viewport hinzugefügt.

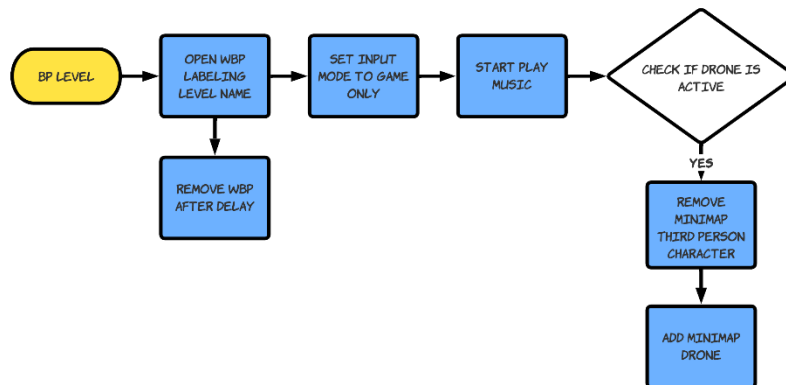


Abbildung 21: BP Level

3.2.4.6 Third Person Character (Avatar)

Abbildung 22 und Abbildung 23 zeigen den Aufbau und Ablauf im Third Person Character BP.

Das Third Person Character BP wird beim Start eines Levels ausgeführt.

Prüfung in der Game Instance, ob die Drohne aktiv ist: Falls diese nicht aktiv ist, werden für den Avatar Player Location and Rotation aus der Game Instance geladen und so die Startposition des Avatars festgelegt. Falls die Drohne in der Game Instance als aktiv gespeichert ist, wird die Steuerung auf die Drohne umgestellt. Das MiniMap WBP des Third Person Character wird vom Viewport entfernt und das MiniMap Drone WBP wird für die Drohne zum Viewport hinzugefügt. Für die Drohne und den Avatar

werden Player Location and Rotation aus der Game Instance geladen und so die Startposition des Avatars und der Drohne festgelegt.

Prüfen, ob sich der Player innerhalb der Levelgrenzen befindet, In Bounds und Out of Bounds Event: Wenn der Player die Levelgrenzen überschreitet, wird das Out of Bounds (außerhalb der Levelgrenzen) Event ausgelöst. Wenn der Player umkehrt und die Levelgrenzen erneut überschreitet, wird das In Bounds (innerhalb der Levelgrenzen) Event ausgelöst. Bei aktivem Out of Bounds Event wird die F-Taste (Umschalten zwischen Avatar und Drohne) und L-Taste (Levelwechsel) deaktiviert. In einer lokalen Variable des Third Person Character BP oder des Drone BP wird gespeichert, dass der Player Out of Bounds ist. Das Bounds WBP wird zum Viewport hinzugefügt. Ein Timer zeigt an, wie viel Zeit der Player hat, bis er an die Startposition des Levels respawned wird. Beim Respawn wird das In Bounds Event ausgelöst. Das In Bounds Event aktiviert die F-Taste und L-Taste. In der lokalen Variable des Third Person Character BP oder des Drone BP wird gespeichert, dass der Player In Bounds ist. Das Bounds WBP wird vom Viewport entfernt und der Timer zurückgesetzt.

Steuerung im Third Person Character BP:

1. Toggle Walking G-Key: Die Gehgeschwindigkeit des Avatars wird auf einen niedrigeren Wert gesetzt. Beim erneuten Drücken der Taste wird der Wert wieder zurückgesetzt.
2. Button Shift: Beim Drücken und Halten der Taste wird die Gehgeschwindigkeit auf einen erhöhten Wert gesetzt (Sprinten).
3. Toggle Pause Menu ESC-Key: Wenn das Spiel nicht pausiert ist, wird das Pause Menu WBP zum Viewport hinzugefügt. Beim Öffnen des Pause Menu wird das Spiel pausiert und der Input Mode auf Game and UI gesetzt. Der Mouse Cursor wird sichtbar. Wenn das Spiel pausiert ist, wird keine Aktion ausgeführt, damit sich nicht mehrere WBP übereinander öffnen.
4. Toggle Level Select L-Key: Beim Betätigen der Taste wird geprüft, ob der Player Out of Bounds ist, in diesem Fall wird keine Aktion ausgeführt.

Wenn das Spiel nicht pausiert ist, wird das Level Select WBP zum Viewport hinzugefügt. Beim Öffnen des Level Select wird das Spiel pausiert und der Input Mode auf Game and UI gesetzt. Der Mouse Cursor wird sichtbar. Wenn das Spiel pausiert ist, wird keine Aktion ausgeführt, damit sich nicht mehrere WBP übereinander öffnen.

5. Toggle Map K-Key: Wenn das Spiel nicht pausiert ist, wird das Map WBP zum Viewport hinzugefügt. Beim Öffnen der Map wird das Spiel pausiert und der Input Mode auf Game and UI gesetzt. Der Mouse Cursor wird sichtbar. Wenn das Spiel pausiert ist, wird keine Aktion ausgeführt, damit sich nicht mehrere WBP übereinander öffnen.
6. Toggle Help H-Key: Wenn das Spiel nicht pausiert ist, wird das Help WBP zum Viewport hinzugefügt. Beim Öffnen des Help WBP wird das Spiel pausiert und der Input Mode auf Game and UI gesetzt. Der Mouse Cursor wird sichtbar. Wenn das Spiel pausiert ist, wird keine Aktion ausgeführt, damit sich nicht mehrere WBP übereinander öffnen.

7. Toggle Drone F-Key: Speicherung der aktuellen Player Location and Rotation in der der Game Instance. In den Variablen der Game Instance wird die Drohne auf aktiv gesetzt. Die Steuerung wird auf die Drohne umgestellt und diese dem Level hinzugefügt. Das MiniMap WBP des Third Person Character wird vom Viewport entfernt und das MiniMap Drone WBP für die Drohne zum Viewport hinzugefügt.

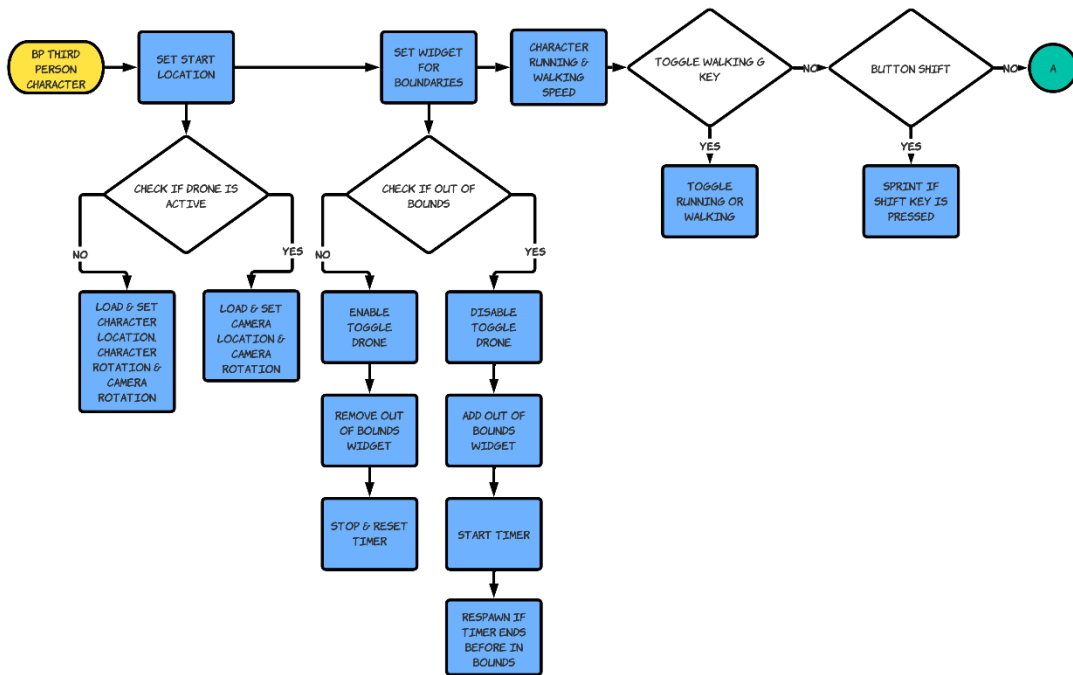


Abbildung 22: BP Third Person Character 1

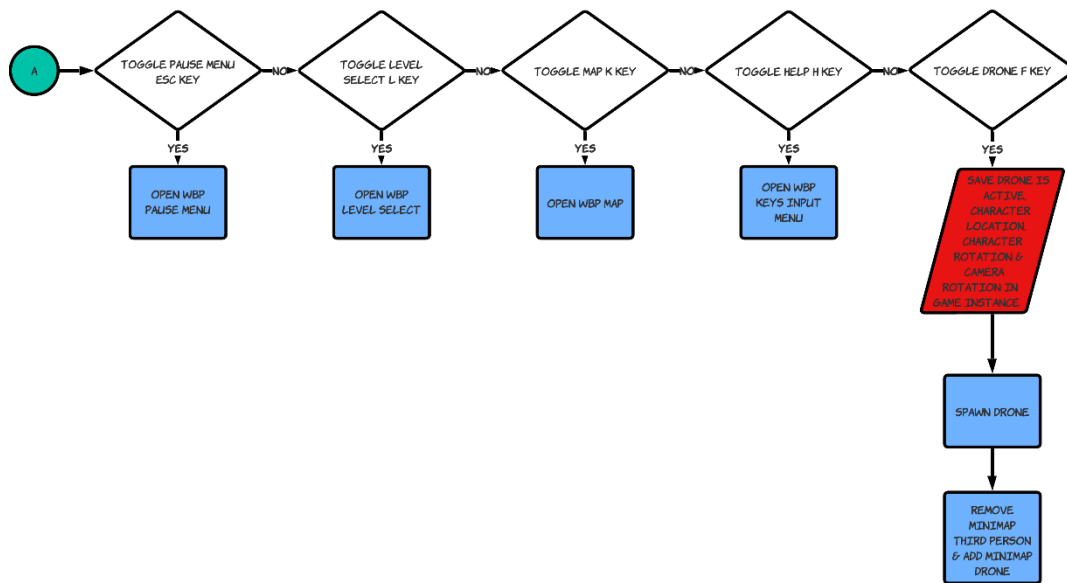


Abbildung 23: BP Third Person Character 2

3.2.4.7 Drone

Abbildung 24 und Abbildung 25 zeigen den Aufbau und Ablauf im Drone BP.

Das Drone BP ist ähnlich aufgebaut wie das Third Person Character BP. Unterschiede bestehen darin, dass die Drohne beim Umschalten mit der F-Taste auf den Avatar komplett aus dem Level entfernt wird. Das Drone BP wird beim Start eines neuen Levels mit aktivierter Drohne über das Third Person Character BP aufgerufen. Zur Steuerung stehen die Tasten für Gehen und Sprinten nicht zur Verfügung. Vergleich Abbildung 24 und Abbildung 22.

Prüfen, ob sich der Player innerhalb der Levelgrenzen befindet, In Bounds und Out of Bounds Event: Wenn der Player die Levelgrenzen überschreitet, wird das Out of Bounds (außerhalb der Levelgrenzen) Event ausgelöst. Wenn der Player umkehrt und die Levelgrenzen erneut überschreitet, wird das In Bounds (innerhalb der Levelgrenzen) Event ausgelöst. Bei aktivem Out of Bounds Event wird die F-Taste (Umschalten zwischen Avatar und Drohne) und L-Taste (Levelwechsel) deaktiviert. In einer lokalen Variable des Third Person Character BP oder des Drone BP wird gespeichert, dass der Player Out of Bounds ist. Das Bounds WBP wird zum Viewport hinzugefügt. Ein Timer zeigt an, wie viel Zeit der Player hat, bis er an die Startposition des Levels respawned wird. Beim respawn wird das In Bounds Event ausgelöst. Das In Bounds Event aktiviert die F-Taste und L-Taste. In der lokalen Variable des Third

Person Character BP oder des Drone BP wird gespeichert, dass der Player In Bounds ist. Das Bounds WBP wird vom Viewport entfernt und der Timer zurückgesetzt.

Steuerung im Drone BP:

1. Toggle Pause Menu ESC-Key: Wenn das Spiel nicht pausiert ist, wird das Pause Menu WBP zum Viewport hinzugefügt. Beim Öffnen des Pause Menu wird das Spiel pausiert und der Input Mode auf Game and UI gesetzt. Der Mouse Cursor wird sichtbar. Wenn das Spiel pausiert ist, wird keine Aktion ausgeführt, damit sich nicht mehrere WBP übereinander öffnen.
2. Toggle Level Select L-Key: Beim Betätigen der Taste wird geprüft, ob der Player Out of Bounds ist, in diesem Fall wird keine Aktion ausgeführt. Wenn das Spiel nicht pausiert ist, wird das Level Select WBP zum Viewport hinzugefügt. Beim Öffnen des Level Select wird das Spiel pausiert und der Input Mode auf Game and UI gesetzt. Der Mouse Cursor wird sichtbar. Wenn das Spiel pausiert ist, wird keine Aktion ausgeführt, damit sich nicht mehrere WBP übereinander öffnen.
3. Toggle Map K-Key: Wenn das Spiel nicht pausiert ist, wird das Map WBP zum Viewport hinzugefügt. Beim Öffnen der Map wird das Spiel pausiert und der Input Mode auf Game and UI gesetzt. Der Mouse Cursor wird sichtbar. Wenn das Spiel pausiert ist, wird keine Aktion ausgeführt, damit sich nicht mehrere WBP übereinander öffnen.
4. Toggle Help H-Key: Wenn das Spiel nicht pausiert ist, wird das Help WBP zum Viewport hinzugefügt. Beim Öffnen des Help WBP wird das Spiel pausiert und der Input Mode auf Game and UI gesetzt. Der Mouse Cursor wird sichtbar. Wenn das Spiel pausiert ist, wird keine Aktion ausgeführt, damit sich nicht mehrere WBP übereinander öffnen.
5. Toggle Third Person Character F-Key: In den Variablen der Game Instance wird die Drohne auf nicht aktiv gesetzt. Die Steuerung wird auf den Avatar umgestellt und die Drohne aus dem Level entfernt. Das MiniMap Drone WBP der Drohne wird vom Viewport entfernt und das MiniMap WBP für den Third Person Character zum Viewport hinzugefügt.

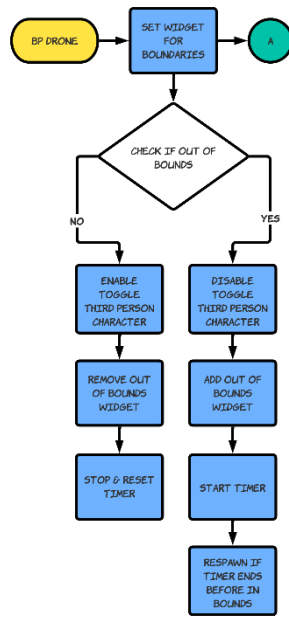


Abbildung 24: BP Drone 1

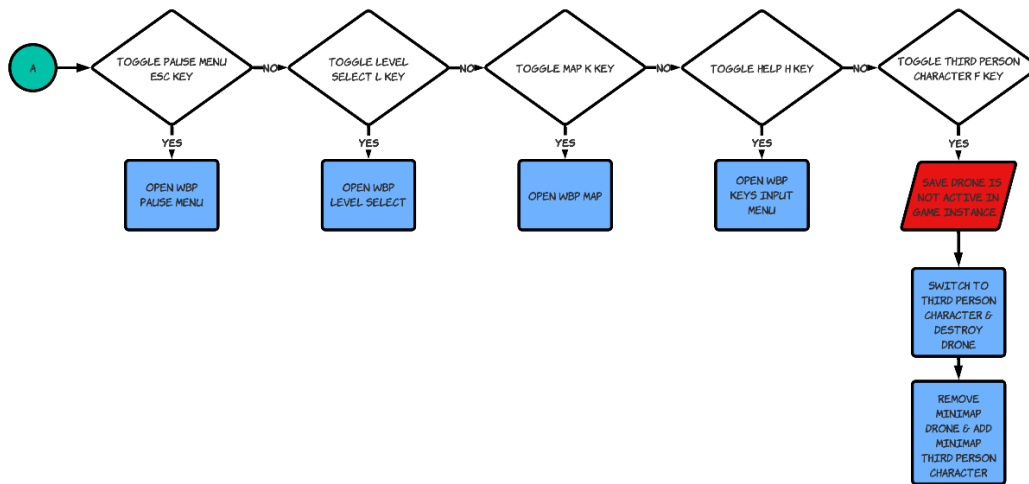


Abbildung 25: BP Drone 2

3.2.4.8 Game Instance

Abbildung 26 zeigt den Aufbau und Ablauf in der Game Instance BP.

Das Game Instance BP erlaubt es während der Laufzeit des Spiels z. B. Variablen, Funktionen und Events level- oder objektübergreifend zu speichern und auszuführen. Es fungiert als high-level Manager während der gesamten Laufzeit der Anwendung. Beim Beenden des Spiels werden alle gesetzten Einstellungen oder Variablen auf die Standardwerte der Game Instance zurückgesetzt. Die Game Instance ist nicht zu verwechseln mit einem permanenten Speicherstand im Spiel, der auch nach erneutem Start der Anwendung zur Verfügung steht.

In der Game Instance werden Character Location, Character Rotation, Camera Location, Camera Rotation, Is Drone Active und Master Sound Value als Variablen während der Laufzeit der Anwendung gespeichert.

Auch wird über die Game Instance ein Play Background Music Event ausgelöst. So muss dieses Event nicht in jedem Level neu implementiert werden und kann aus jedem Level angesprochen werden.

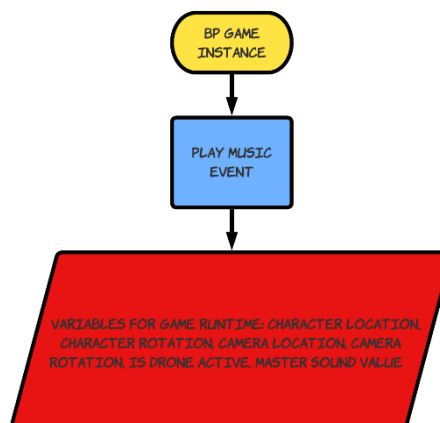


Abbildung 26: BP Game Instance

3.2.4.9 In & Out Bounds

Abbildung 30 zeigt den Aufbau und Ablauf für das In & Out Bounds BP und WBP. Abbildung 46 zeigt das In & Out Bounds WBP.

Die In & Out Bounds Funktion besteht aus einem WBP (Bounds WBP) und einem Actor BP (Bounds BP). Das WBP beinhaltet neben der Anzeige im Viewport auch eine Funktion, die den Text der ablaufenden Sekunden aus einer Variablen liest und so dynamisch ändert, siehe Abbildung 46.

Das Bounds BP kann im Level beliebig oft platziert werden. Dieses besteht aus zwei BoxCollisions und einem Cube, der als Trennwand fungiert und zwischen den beiden BoxCollisions platziert ist, siehe Abbildung 28. Die Trennwand dient als Hilfe, um bei der Platzierung im Level das Bounds BP sehen zu können. Dafür wurde der Trennwand ein durchsichtiges Material (Glas) zugewiesen. Zusätzlich wurde auf der Trennwand ein Richtungspfeil platziert, der bei der Positionierung im Level Unterstützung bei der korrekten Ausrichtung bietet, siehe Abbildung 27. Während der Laufzeit der Anwendung ist das Bounds BP nicht zu sehen.

Damit sich die BoxCollisions und die Trennwand im Bounds BP beim Ändern der Größe des Actors im Level Editor dynamisch verändern, wird die Größe der BoxCollisions und der Trennwand im Bounds BP über ein Construction Script in Abhängigkeit von der Größe im Level Editor dynamisch angepasst, siehe Abbildung 29.

Das In & Out Bounds BP wird über zwei Events aktiviert. Diese starten, wenn sie vom Avatar oder von der Drohne berührt werden und sind richtungsabhängig, je nachdem ob der Player die Levelgrenzen verlässt oder wieder betritt.

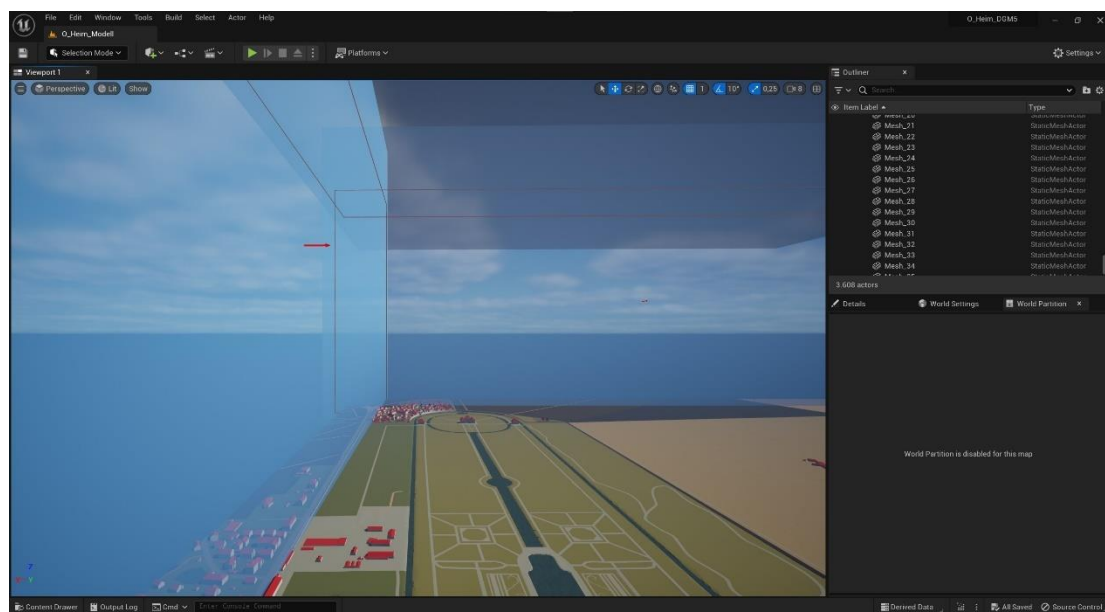


Abbildung 27: Bounds Actor im Unreal Engine Level Editor

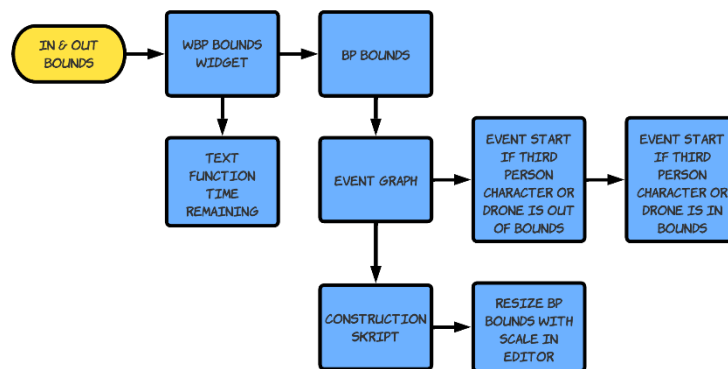


Abbildung 30: BP & WBP In & Out Bounds

3.2.4.10 MiniMap

Abbildung 31 und Abbildung 32 zeigen den Aufbau und Ablauf für das MiniMap BP und WBP. Abbildung 47 und Abbildung 48 zeigen die MiniMap.

Die MiniMap besteht aus einem BP und WBP mit einem dynamischen Material. Das MiniMap BP fügt das MiniMap WBP beim Start des Levels zum Viewport hinzu.

Über ein Event wird während der gesamten Laufzeit der Anwendung für jeden Frame die Player Location and Rotation in ein Material Parameter (MP) MiniMap gespeichert. Im MP MiniMap können Skalar- und Vektorparameter gespeichert werden, die in einem Material ausgelesen werden können. Ansonsten steht in einem Material keine Funktion zum Abrufen dieser Parameter zur Verfügung.

Das MiniMap WBP zeigt im Viewport eine Übersichtskarte, deren Ausschnitt sich dynamisch mit dem aktuellen Standort des Players verändert.

Die dynamische Änderung wird dadurch ermöglicht, dass im WBP anstatt einer einfachen Textur ein veränderbares Material zur Verwendung kommt. Dieses Material wird aus der Textur der Übersichtskarte erstellt.

Die gespeicherte Player Location aus dem Material Parameter wird im MiniMap Material abgerufen und auf die Texturkoordinaten der Übersichtskarte umgerechnet. Das Material selbst besitzt hierbei die Textur der Übersichtskarte. Diese wird in einer bestimmten Zoomstufe angezeigt und die Koordinaten des Players im Level werden auf die Koordinaten der Textur der Übersichtskarte umgerechnet. So zeigt der Ausschnitt in der MiniMap vergrößert den Bereich, in dem sich der Player zum momentanen Zeitpunkt befindet.

Zusätzlich wird im Material eine Textur des Player Symbols (Symbol mit Richtungsanzeige) auf die Textur der Übersichtskarte gelegt. Damit das Player Symbol auch die Blickrichtung des Players anzeigt, wird es über eine weitere Umrechnung aus

den gespeicherten Rotationswerten des Material Parameter rotiert. So stimmt die Blickrichtung des Player Symbols in der MiniMap mit der Blickrichtung des Players in der Anwendung überein.

Für die Umsetzung sind für den Avatar und für die Drohne jeweils ein Material und ein WBP notwendig.

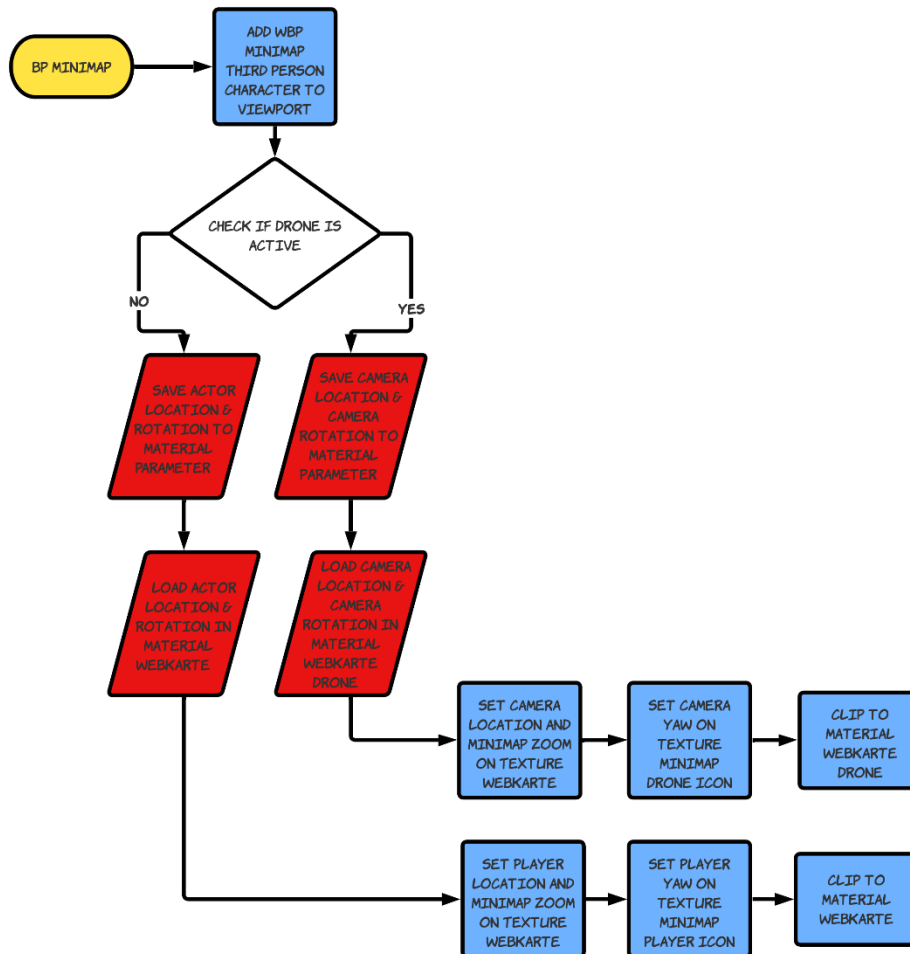


Abbildung 31: BP MiniMap



Abbildung 32: MP MiniMap

3.2.4.11 Map

Abbildung 33 zeigt den Aufbau und Ablauf für das Map WBP. Abbildung 49 und Abbildung 50 zeigt die Map.

Die Map ist ein WBP. Beim Öffnen der Map wird das Spiel pausiert und der Input Mode auf Game and UI gesetzt. Der Mouse Cursor wird sichtbar.

Im Gegensatz zur MiniMap werden beim Map WBP die Location and Rotation des Players auf Player Symbole im WBP angewendet und müssen nicht dynamisch sein. So müssen die Koordinaten nicht in einem Material angewendet werden. Auch ist keine zweite Map notwendig, da für den Avatar und für die Drohne jeweils ein Player Symbol im Map WBP zur Verfügung steht. Die Player Symbole sind im WBP außerhalb des Viewports platziert, so wird das nicht zur Verwendung gekommene Player Symbol beim Aufrufen der Map nicht im Viewport angezeigt.

Um die Player Symbole korrekt zu positionieren, wird die aktuelle Player Location im Spiel auf die Texturkoordinaten der Map umgerechnet. Damit wird das Player Symbol auf der Hintergrundkarte platziert. Damit das Player Symbol auch die Blickrichtung des Players anzeigt, wird es über eine weitere Umrechnung der Player Rotation rotiert. So stimmt die Blickrichtung des Player Symbols in der Map mit der Blickrichtung des Players in der Anwendung überein. Das Player Symbol zeigt den aktuellen Standort und die Blickrichtung des Players auf der Map.

Ausführbare Elemente im Map WBP:

Button Close WBP: Der Input Mode wird auf Game Only gesetzt. Der Mouse Cursor wird ausgeblendet. Das Map WBP wird vom Viewport entfernt und die Pausierung des Spiels wird aufgehoben.

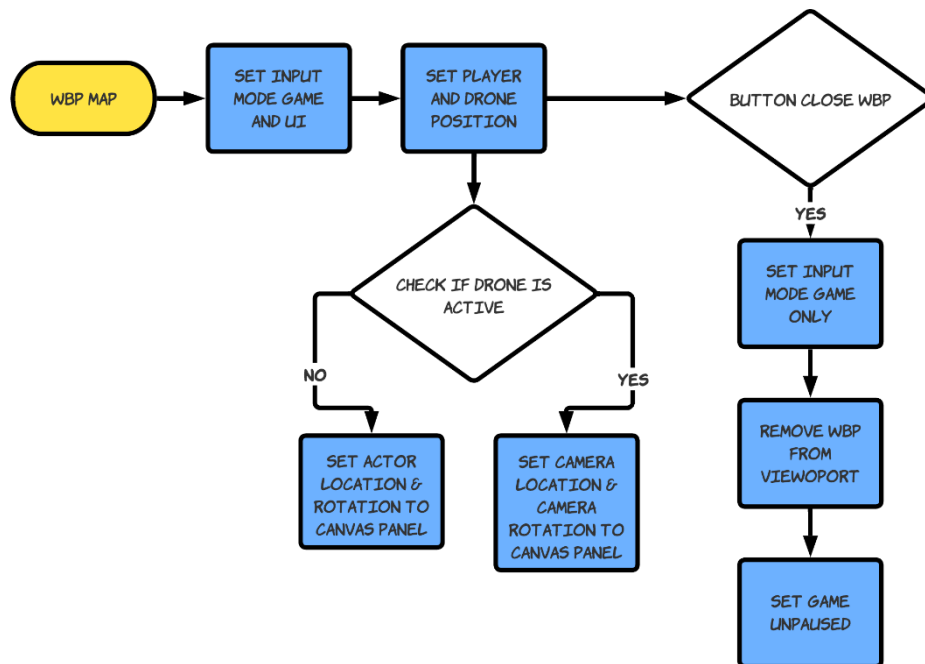


Abbildung 33: WBP Map

4. ERGEBNISSE

4.1 InfraWorks®

Abbildung 34 zeigt das finale Modell von Oberschleißheim im Viewport von InfraWorks®. Das Surface Layers Fenster zeigt die zur Verwendung gekommenen Geodaten für das Surface und welche von diesen aktuell im Modell aktiviert sind.

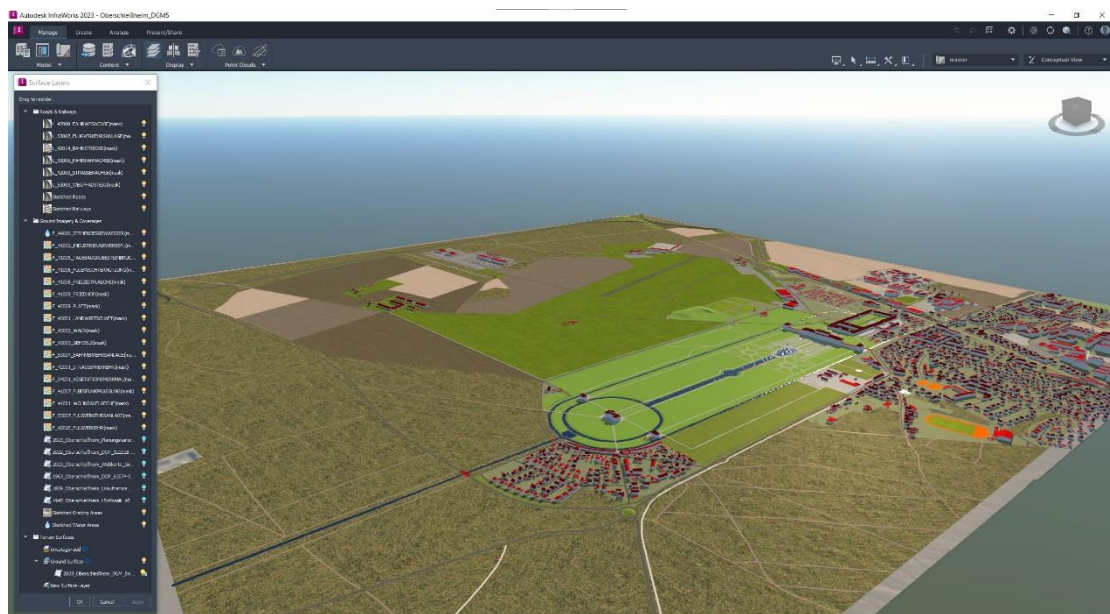


Abbildung 34: Fertiges InfraWorks®-Modell, bereit für den FBX-Export

4.2 Twinmotion 2023

Abbildung 35 zeigt das importierte InfraWorks®-Modell in Twinmotion. Twinmotion ist im Projekt nur für die Datenkonvertierung zum Einsatz gekommen. Das Modell wird nach dem Import im FBX-Format in Twinmotion als Datasmith-Format exportiert und ist dann bereit für den Import in Unreal Engine.

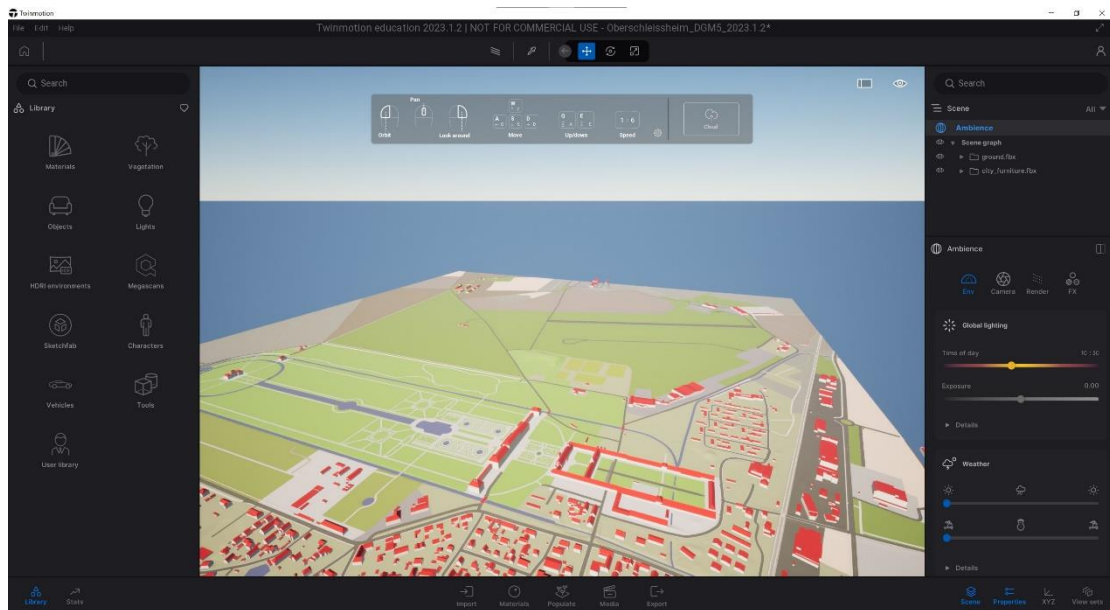


Abbildung 35: Importiertes Modell, bereit für den Datasmith-Export in Twinmotion

4.3 Unreal Engine Editor

Abbildung 36 zeigt das fertige Projekt im Unreal Engine Editor. Im Outliner ist die Struktur des Projekt zu sehen. Auch sieht man um das Modell herum die In & Out Bounds Actor BP, die im Level platziert wurden.

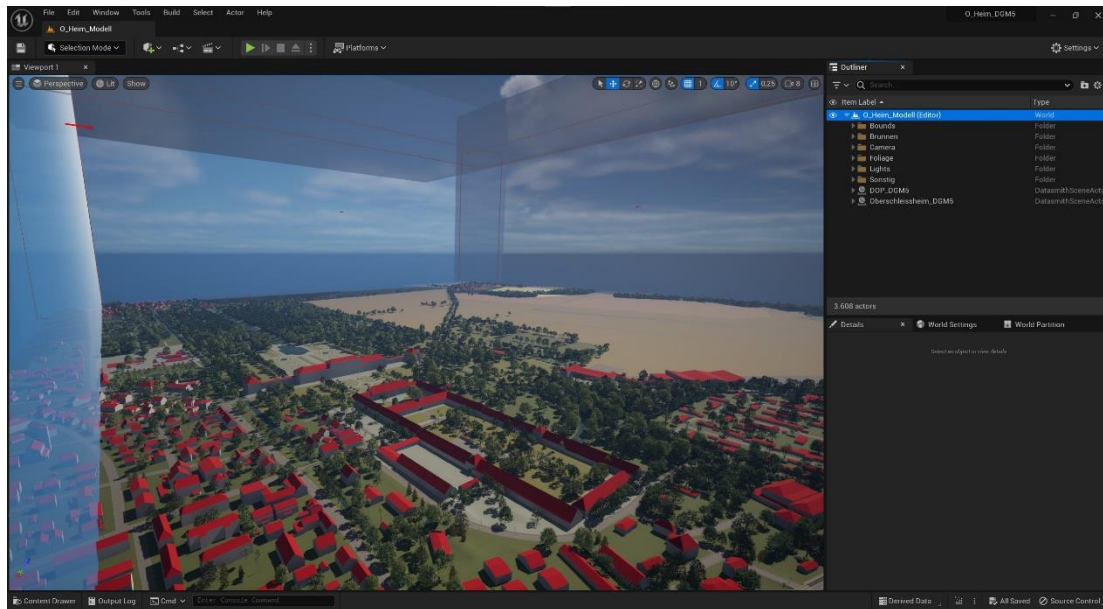


Abbildung 36: Fertiges Projekt Unreal Engine Editor

4.4 Projekt UI

Abbildung 37 bis Abbildung 46 zeigen das fertige Design vom UI. Alle Beschriftungen für das gesamte UI der Anwendung wurden auf Deutsch erstellt. Alle Buttons im UI ändern die Farbe, wenn der Mauszeiger über einem Button hovers, siehe der Spiel Starten-Button in Abbildung 37. Zusätzlich werden verschiedene Töne beim Hovern oder beim Klicken auf einen Button abgespielt und eine Hintergrundmusik ist zu hören. Im Options Menu, siehe Abbildung 38, sind die ausgewählten Game Settings gelb hinterlegt. Während des Spiels wird im UI der momentane View des Avatars oder der Drohne verschwommen angezeigt, siehe Abbildung 40 bis Abbildung 42. Abbildung 43 bis Abbildung 45 zeigen das Levelbeschriftung WBP, das beim Starten des Spiels oder beim Levelwechsel für fünf Sekunden eingeblendet wird. Zur Hilfestellung bei der Steuerung wird zusätzlich ein Hinweis eingeblendet, wie das Keys Input WBP aufzurufen ist. Für das Out of Bounds Event wird beim Verlassen der Levelgrenzen das in Abbildung 46 gezeigte WBP mit ablaufendem Timer angezeigt.

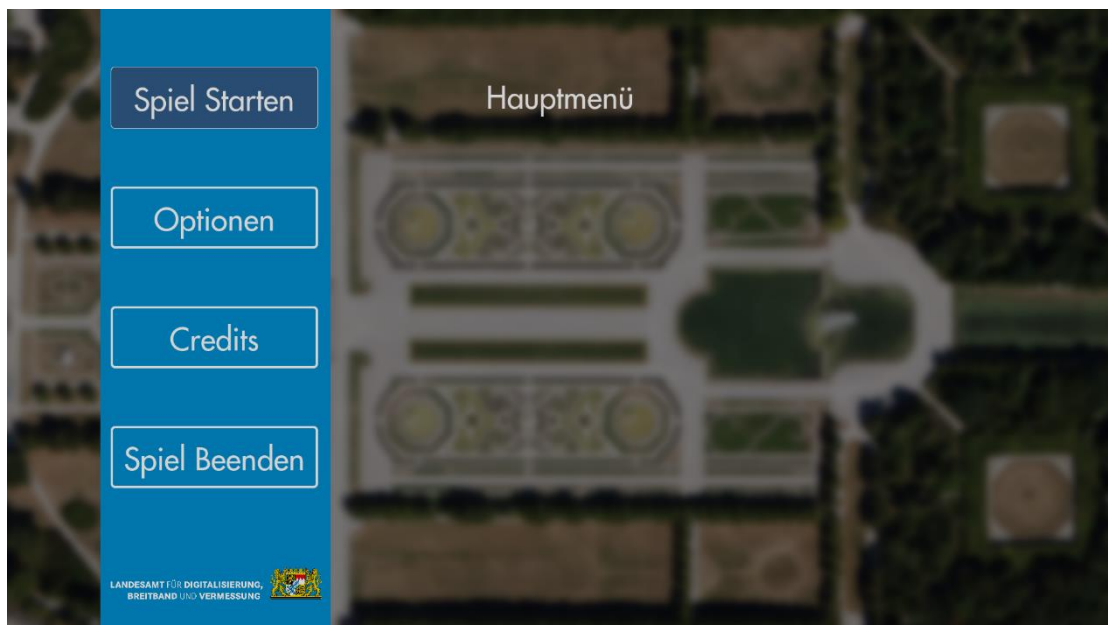


Abbildung 37: Main Menu



Abbildung 38: Options Menu



Abbildung 39: Credits

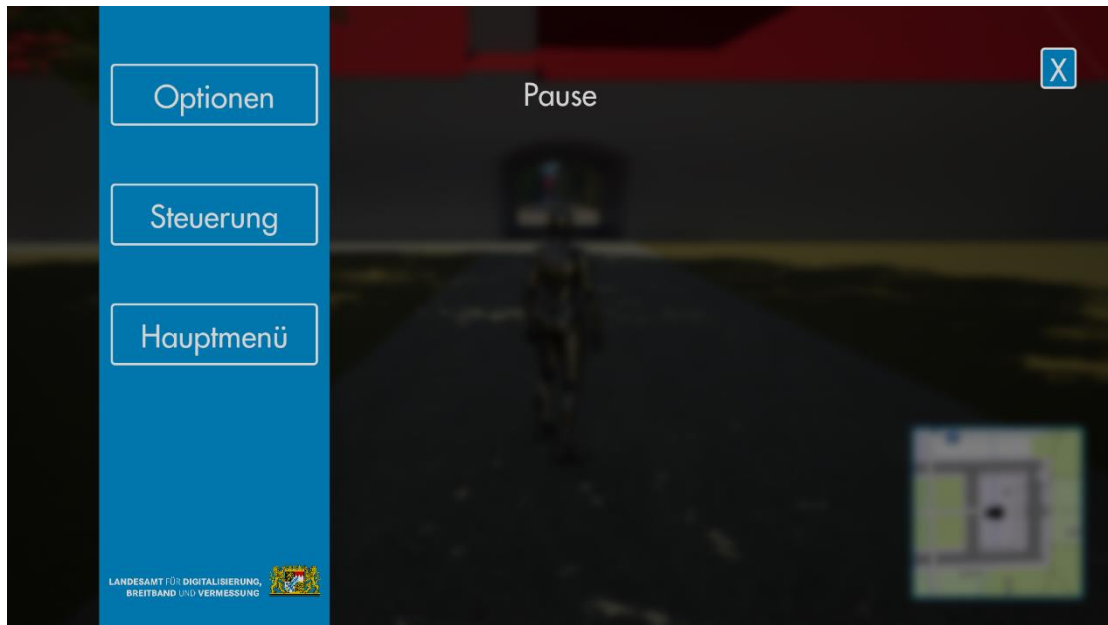


Abbildung 40: Pause Menu

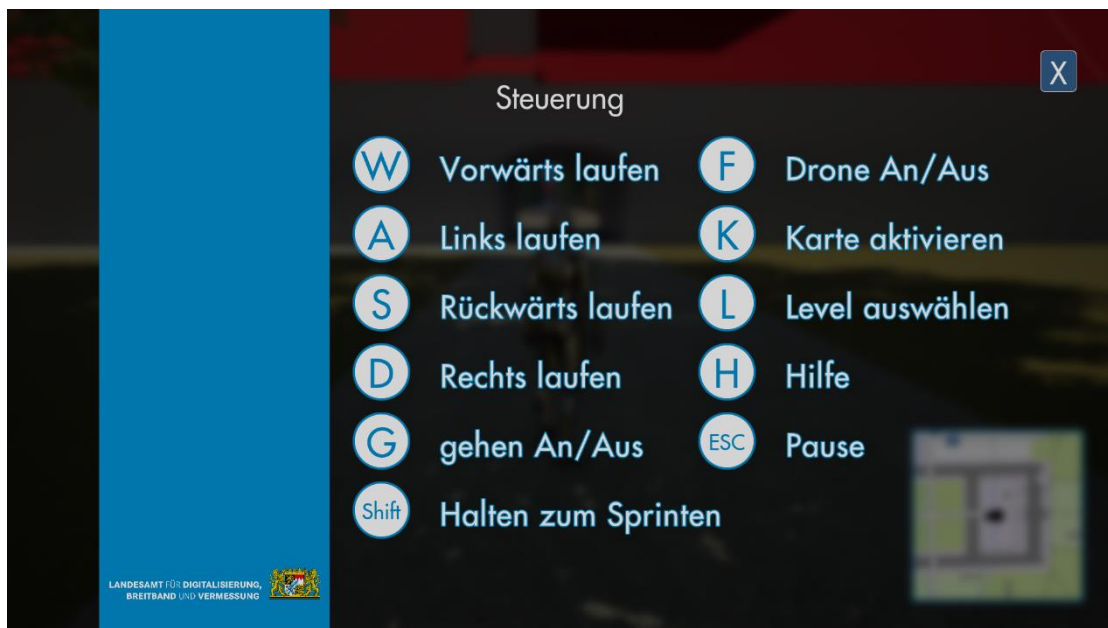


Abbildung 41: Keys Input

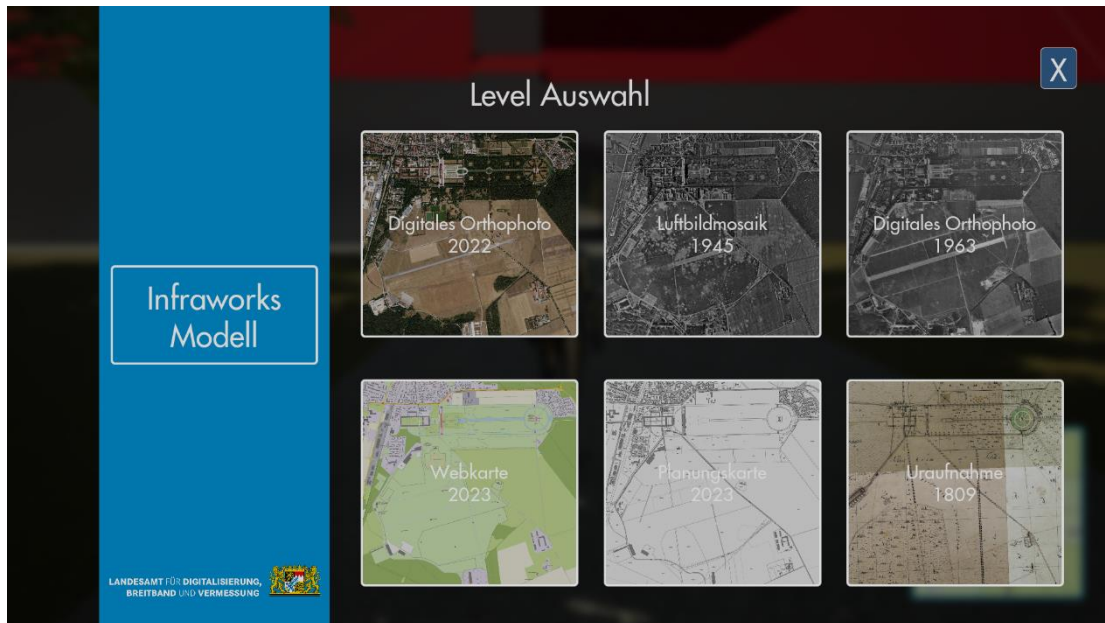


Abbildung 42: Level Select

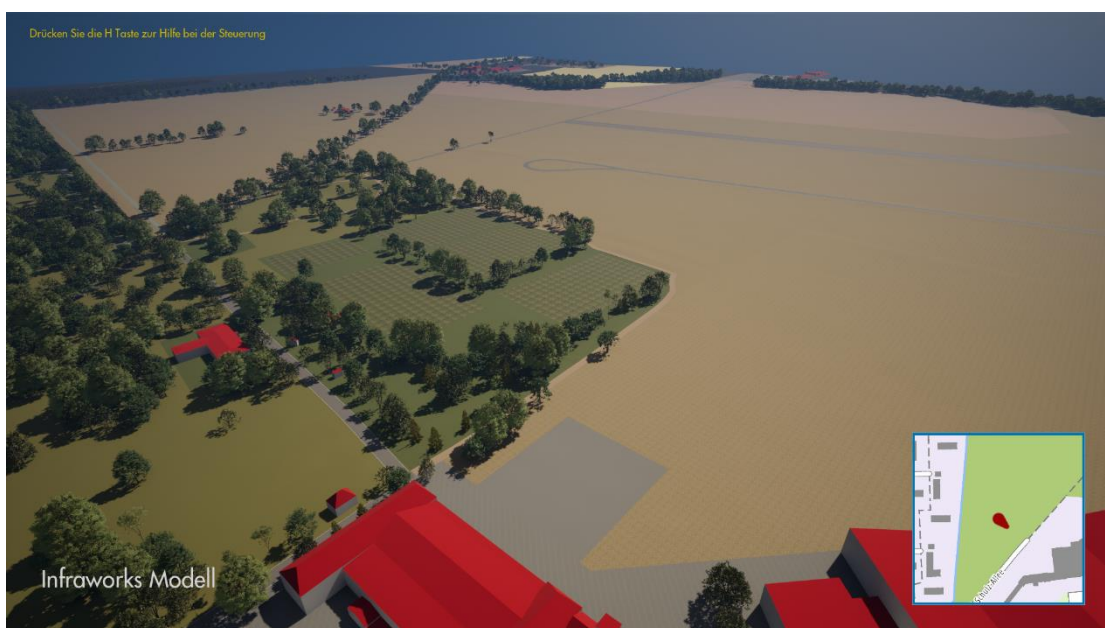


Abbildung 43: Levelbeschriftung Infracore®-Modell Drohne



Abbildung 44: Levelbeschriftung Digitales Orthophoto 2022

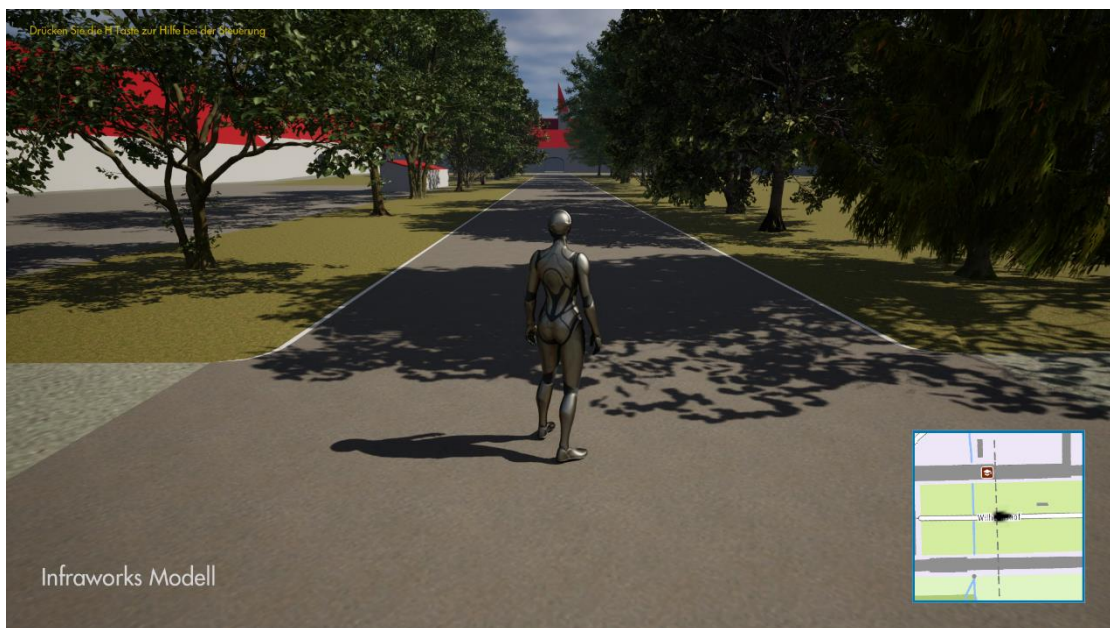


Abbildung 45: Levelbeschriftung InfraWorks®-Modell Avatar



Abbildung 46: In & Out Bounds Timer

4.5 Projekt MiniMap

Abbildung 47 und Abbildung 48 zeigen die MiniMap jeweils für den Avatar und die Drohne. Damit sofort erkannt werden kann, in welchem Modus sich das Spiel befindet, ist das Player Symbol des Avatars schwarz und das der Drohne rot.

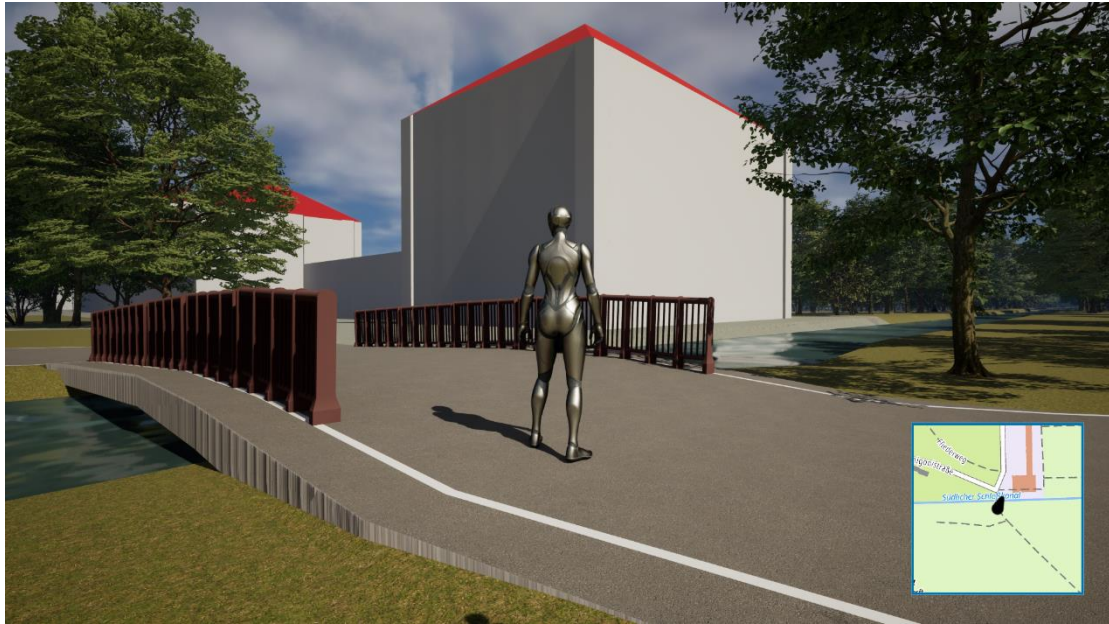


Abbildung 47: MiniMap Avatar

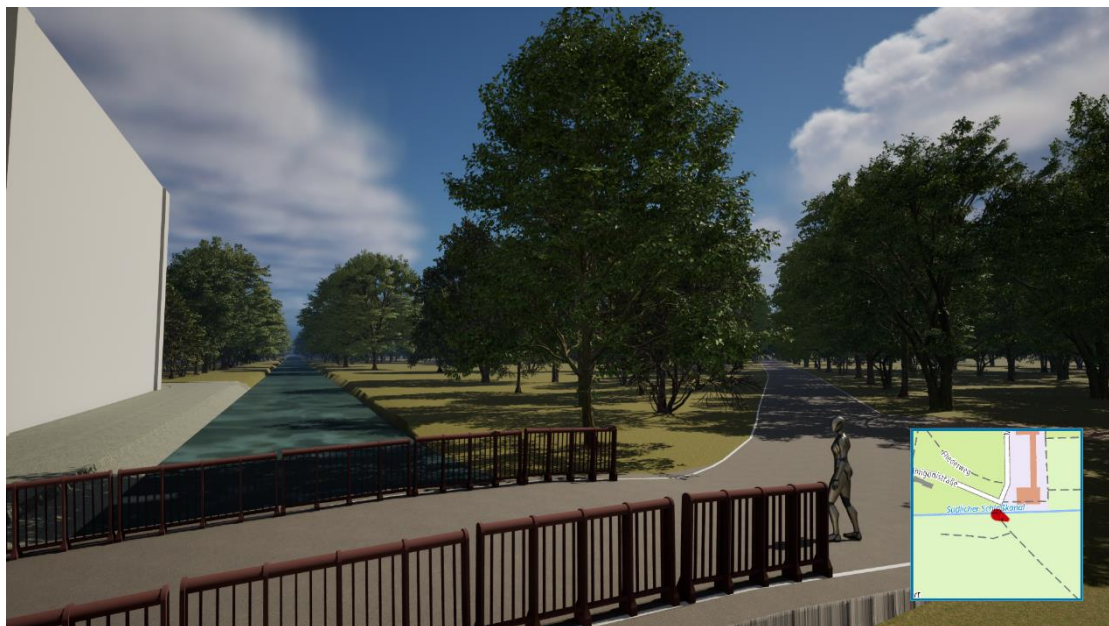


Abbildung 48: MiniMap Drohne

4.6 Projekt Map

Abbildung 49 und Abbildung 50 zeigen die Map jeweils für den Avatar und die Drohne. Damit sofort erkannt werden kann, in welchem Modus sich das Spiel befindet, ist das Player Symbol des Avatars schwarz und das der Drohne rot.



Abbildung 49: Map Avatar



Abbildung 50: Map Drohne

4.7 Projekt Loadingscreen

Abbildung 51 bis Abbildung 59 zeigen die unterschiedlichen Loadingscreens der Level. Für jedes Level wird der passende Loadingscreen mit Levelbeschriftung und Übersichtskarte angezeigt. Damit erkannt werden kann, ob das Spiel noch lädt und sich nicht aufgehängt hat, blinkt das LDBV-Logo, siehe Abbildung 51 und Abbildung 52 zum Vergleich.



Abbildung 51: Loadingscreen Digitales Orthophoto 2022



Abbildung 52: Loadingscreen Status Symbol



Abbildung 53: Loadingscreen InfraWorks®-Modell



Abbildung 54: Loadingscreen Luftbild Mosaik 1945

Digitales Orthophoto 1963

LANDSAMT FÜR DIGITALISIERUNG, ERBTRAND UND VERMESSUNG  Lade



Abbildung 55: Loadingscreen Digitales Orthophoto 1963

Webkarte 2023

LANDSAMT FÜR DIGITALISIERUNG, ERBTRAND UND VERMESSUNG  Lade



Abbildung 56: Loadingscreen Webkarte 2023

Planungskarte 2023

Lade



Abbildung 57: Loadingscreen Planungskarte 2023

Uraufnahme 1809

Lade



Abbildung 58: Loadingscreen Uraufnahme 1809

Zurück zum Hauptmenü

LANDSAMT FÜR DIGITALISIERUNG, BREITBAND UND VERMESSUNG  Lade

Abbildung 59: Loadingscreen Main Menu

4.8 Projekt Level

Abbildung 60 bis Abbildung 65 zeigen die sechs Level im Spiel, die nur aus einem DGM mit Texturierung bestehen und zusätzlich zum Hauptlevel mit dem InfraWorks®-Modell erstellt wurden. Abbildung 66 und Abbildung 67 zeigen einen Vergleich des identischen Ausschnittes des InfraWorks®-Modells und des Digitalen Orthophotos 2022. Abbildung 68 und Abbildung 69 zeigen die Level des Digitalen Orthophotos 2022 und der Webkarte 2023, wenn der Avatar zur Steuerung eingesetzt wird.



Abbildung 60: Level Digitales Orthophoto 2022



Abbildung 61: Level Luftbild Mosaik 1945



Abbildung 62: Level Digitales Orthophoto 1963

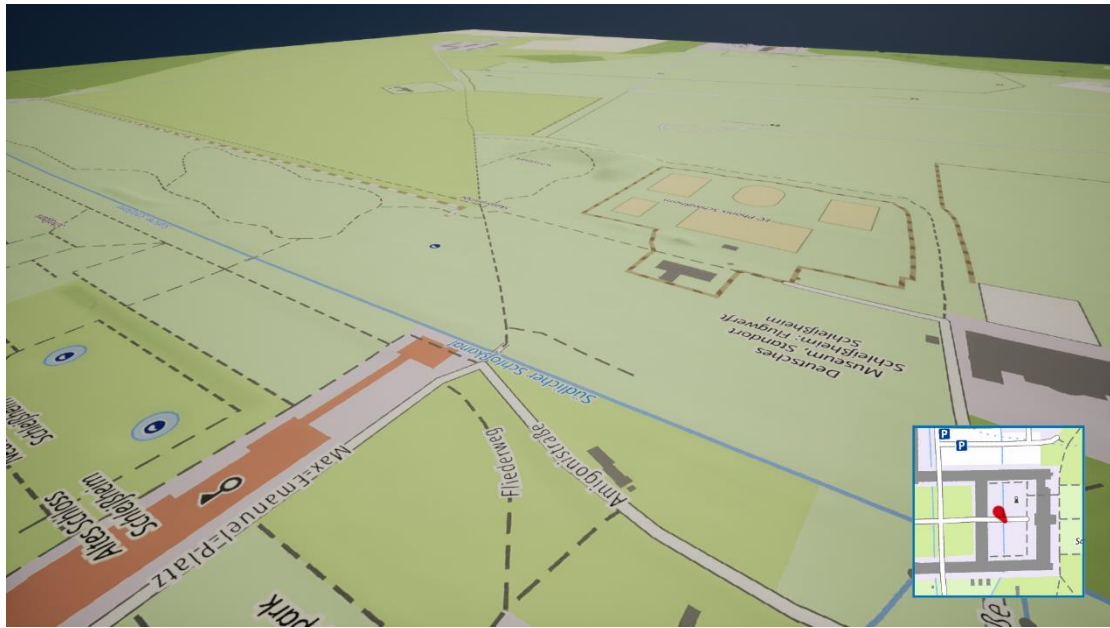


Abbildung 63: Level Webkarte 2023



Abbildung 64: Level Planungskarte 2023



Abbildung 65: Level Uraufnahme 1809

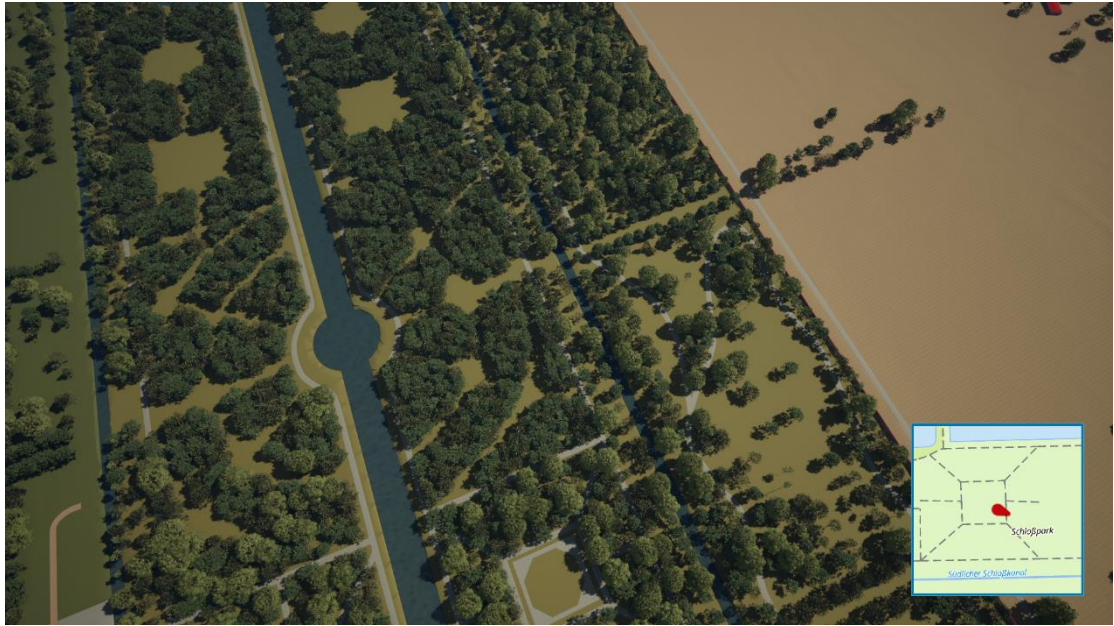


Abbildung 66: Level InfraWorks®-Modell zum Vergleich



Abbildung 67: Level Digitales Orthophoto 2022 zum Vergleich



Abbildung 68: Level Digitales Orthophoto 2022 mit Avatar



Abbildung 69: Level Webkarte 2023 mit Avatar

4.9 Projekt Beispielbilder

Abbildung 70 bis Abbildung 83 zeigen Beispielbilder des fertigen InfraWorks®-Levels.



Abbildung 70: Projekt Beispielbild 1

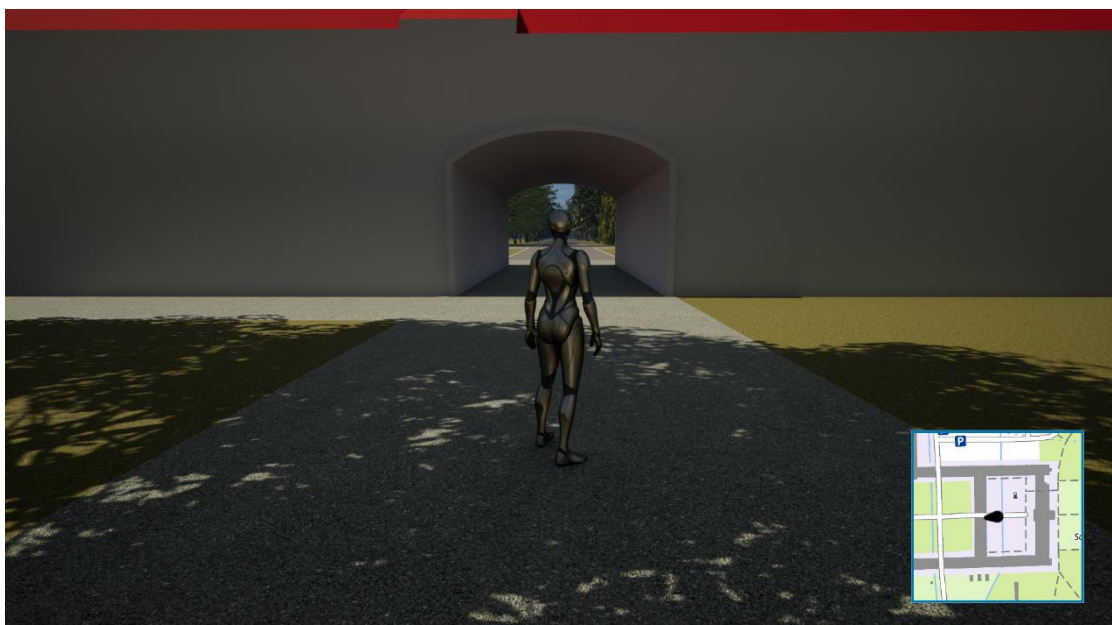


Abbildung 71: Projekt Beispielbild 2

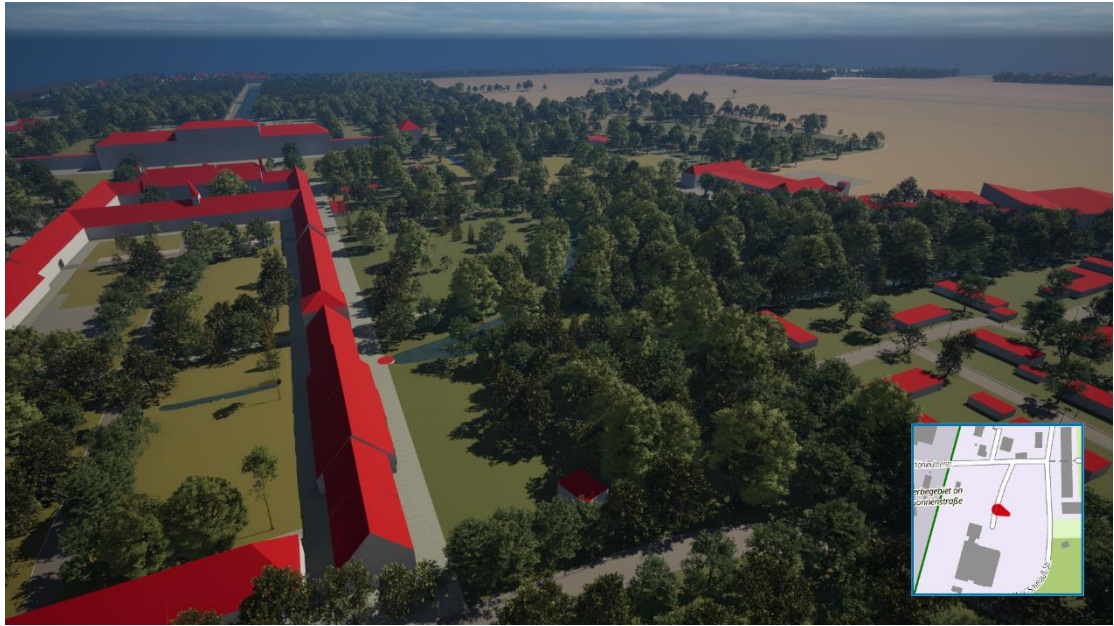


Abbildung 72: Projekt Beispielbild 3

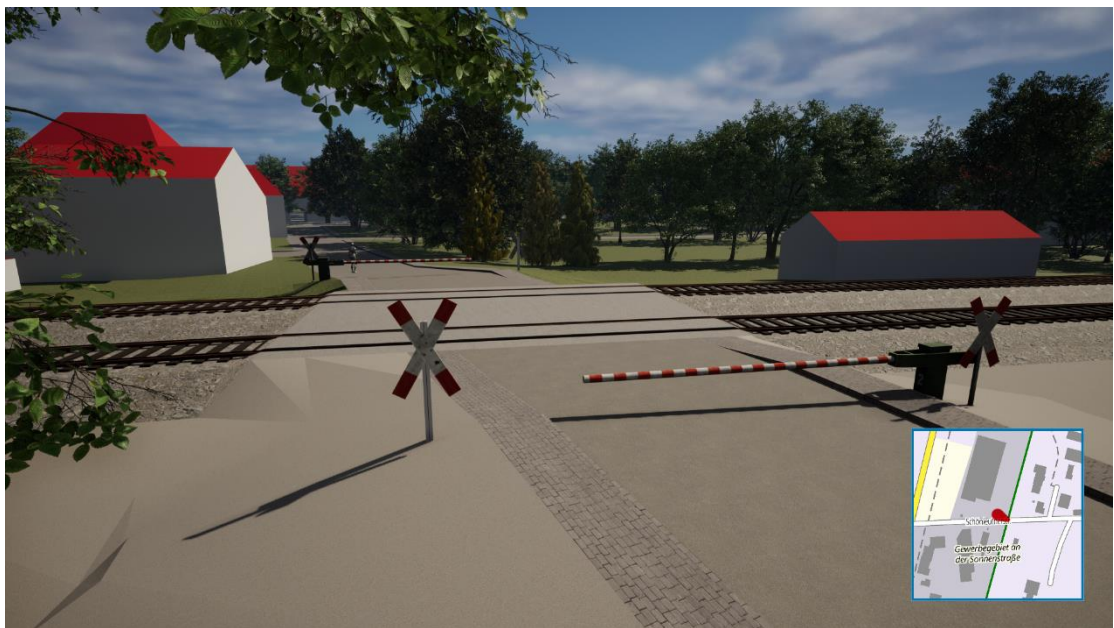


Abbildung 73: Projekt Beispielbild 4

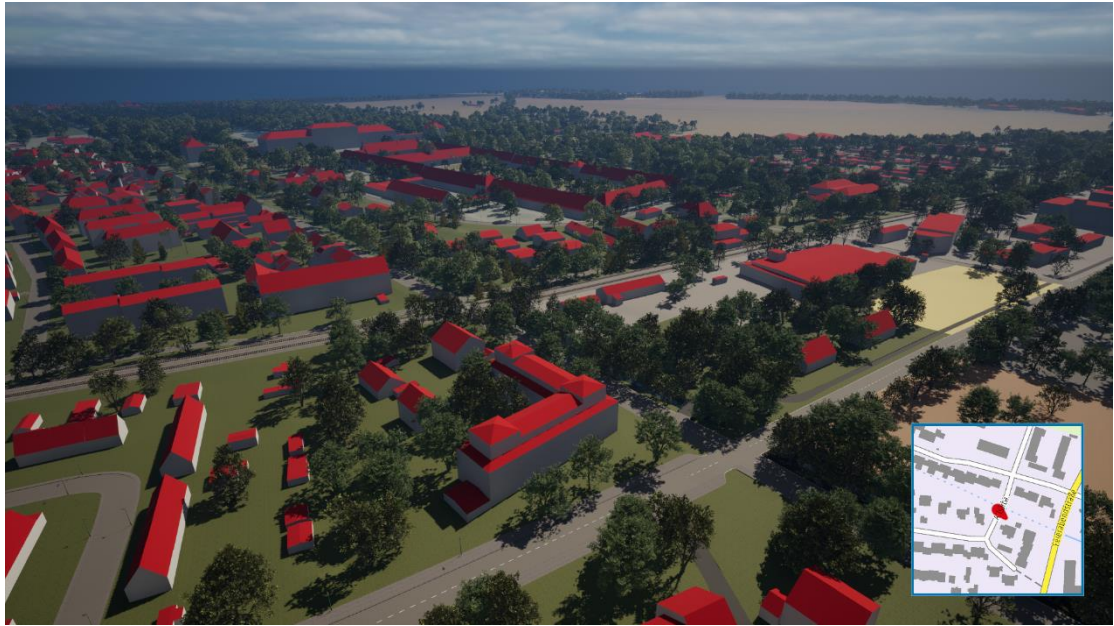


Abbildung 74: Projekt Beispielbild 5

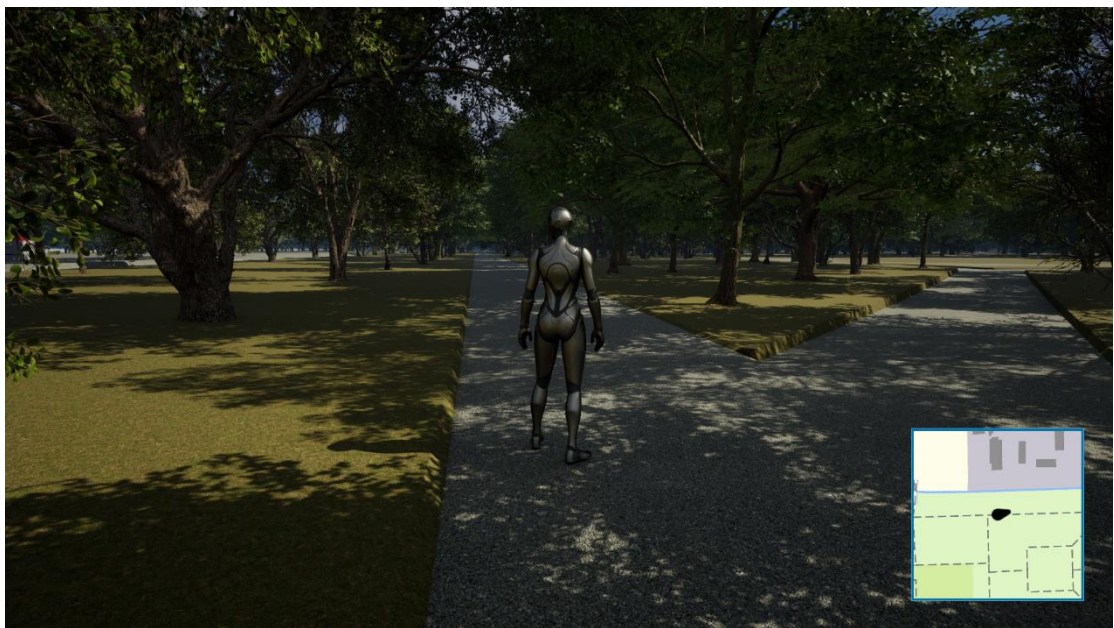


Abbildung 75: Projekt Beispielbild 6

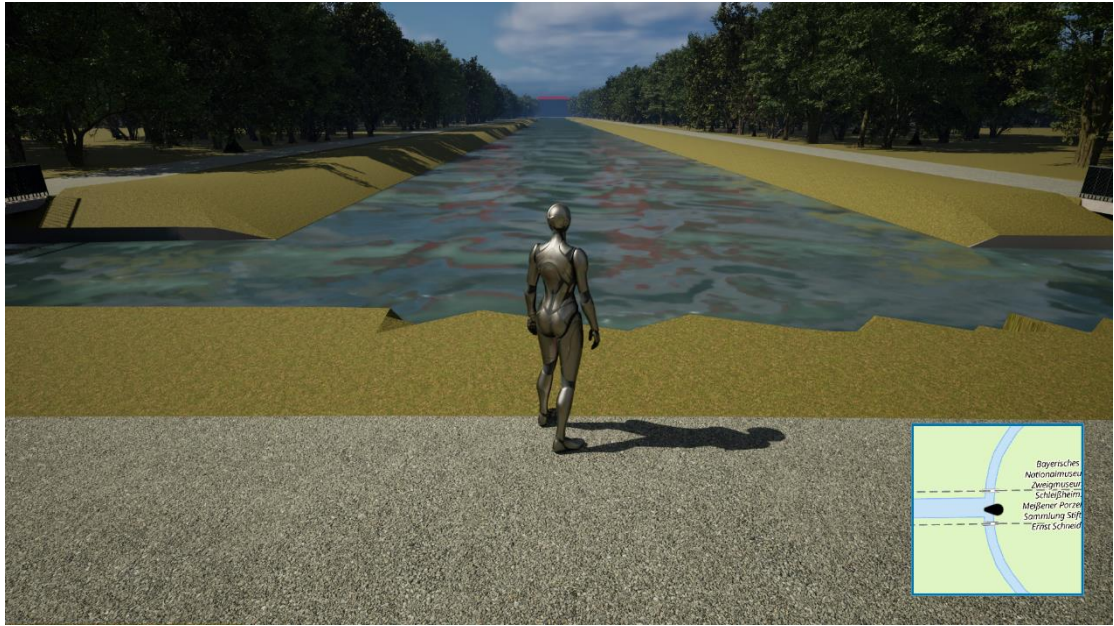


Abbildung 76: Projekt Beispielbild 7

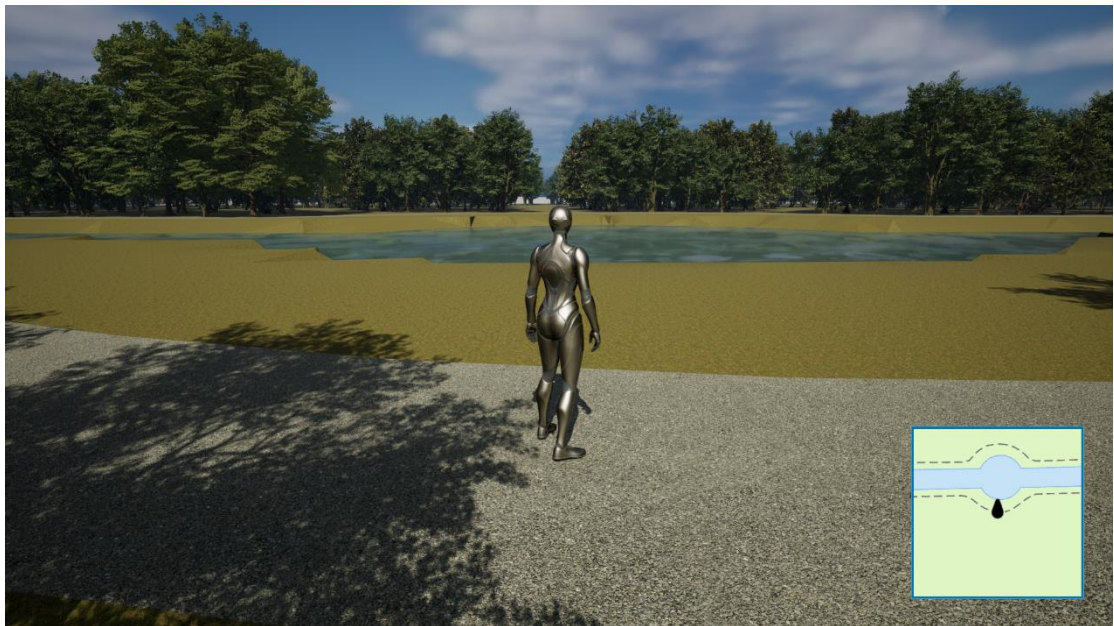


Abbildung 77: Projekt Beispielbild 8



Abbildung 78: Projekt Beispielbild 9

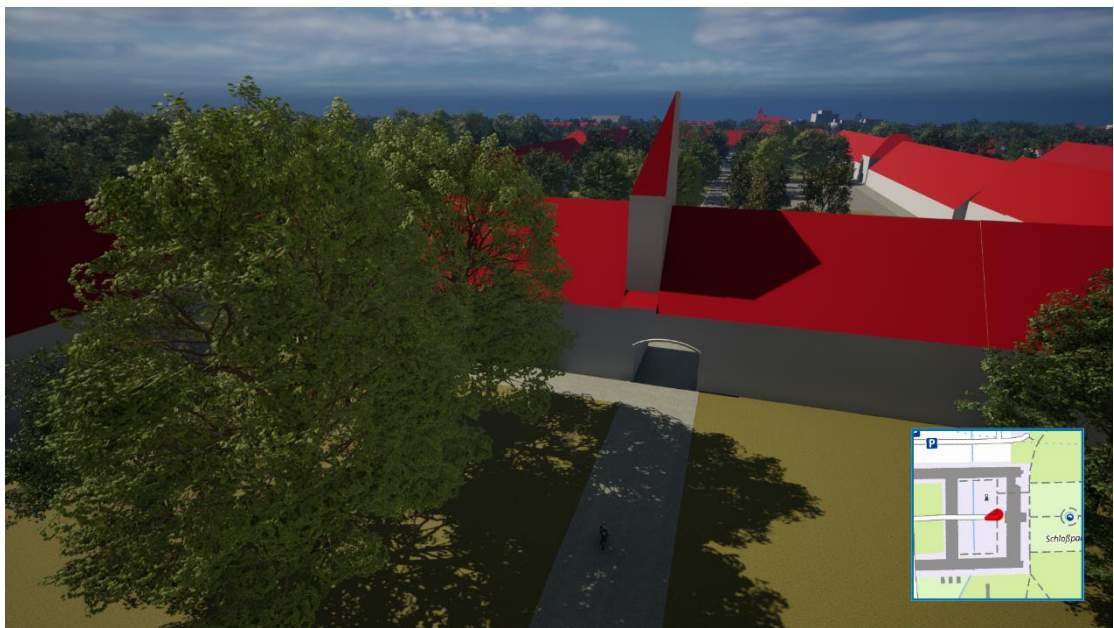


Abbildung 79: Projekt Beispielbild 10



Abbildung 80: Projekt Beispielbild 11

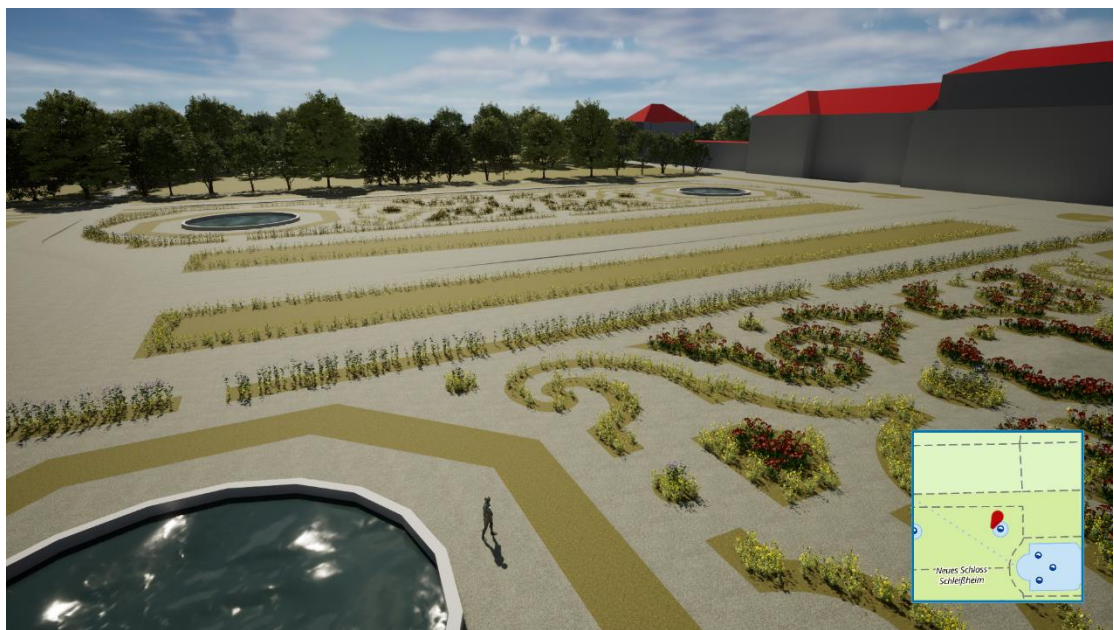


Abbildung 81: Projekt Beispielbild 12



Abbildung 82: Projekt Beispielbild 13



Abbildung 83: Projekt Beispielbild 14

5. DISKUSSION UND FAZIT

5.1 Diskussion

Diese Arbeit zeigt, wie selbst im Rahmen einer Bachelorarbeit Geodaten des LDBV mit Hilfe einer Game Engine visualisiert und in einer interaktiven Anwendung präsentiert werden können. Obwohl das Modell von Oberschleißheim visuell nur minimal aufbereitet wurde und keine Optimierung an Materialien durchgeführt oder eine der Realität entsprechende Vegetation eingefügt wurde, demonstriert es das Potenzial von amtlichen Geodaten in einer Game Engine. Obwohl den LoD2-Gebäudemodellen Texturen fehlen, bietet die Anwendung dennoch eine ansprechende Darstellung und ein immersives Erlebnis. Reale Gebäudetexturen würden die Orientierung im Modell aber noch zusätzlich verbessern. Damit in der vorliegenden Anwendung die Realität nicht zu stark verzerrt wird, wurde den LoD2-Gebäudemodellen für die Fassade und Dächer keine fiktive Textur zugewiesen. So wird ein Verfälschen der Daten verhindert. Für die Anwendung sind weitere Optimierungen möglich, diese wären jedoch aufgrund des zeitlich eingeschränkten Rahmens einer Bachelorarbeit nicht realisierbar gewesen.

Für eine höhere visuelle Qualität könnten die im Projekt verwendeten Materialien individuell optimiert werden. Zusätzlich ermöglicht das Mesh Paint Tool das Aufbereiten harter Kanten, etwa zwischen zwei angrenzenden Nutzungsflächen. Durch das Tool kann ein fließender Übergang zwischen den Materialien der Nutzungsflächen geschaffen werden und so eine realistischere Darstellung entstehen.

Das Modell könnte durch detailliertere Vegetation wie Weizen-, Maisfelder oder Gräser und durch 3D-Modelle von Gesteinen sowie herumliegendes Laub und Äste weiter verbessert werden. In Wäldern könnte durch eine Bodenvegetation auch ein realistischeres Bild erreicht werden. Mit einem Baumkataster, das auch die Baumarten enthält, oder mit allgemeinen Informationen zur vorkommenden Vegetation im abgebildeten Gebiet, kann mit einer Auswahl von passenden 3D-Modellen die Vegetation im Modell an die tatsächliche Vegetation in der Realität angepasst werden. Auch wenn dies nur in einer generalisierten Form geschehen würde, würde das Modell dadurch nochmal realistischer wirken.

Eine ergänzende Stadtmöblierung und das Platzieren von weiteren 3D-Modellen wie parkenden Autos oder z. B. von Flugzeugen am Flughafen Oberschleißheim würde die Immersion zusätzlich erhöhen. Der benötigte Detailgrad hängt jedoch auch von der spezifischen Anwendung ab. Für solche Verbesserungen ist eventuell auch eine bessere Vorbereitung des Modells in InfraWorks® notwendig.

Auch die Einbindung von Animationen von z. B. Segelflugzeugen, Fußgängern und fahrenden Autos könnte das Immersionsgefühl im Modell weiter verstärken. Allerdings würden Erweiterungen mit detaillierter Vegetation, vielen 3D-Modellen und Animationen auch die Hardwareanforderungen erhöhen. Daher wären umfangreiche

Optimierungen an der Performance notwendig, um das Projekt nicht nur auf High-End-Hardware lauffähig zu machen.

Das Projekt nutzt Lumen für die Beleuchtung und Nanite für das Rendering der Vegetation, was bereits hohe Anforderungen an die Performance stellt. Mit hohen Grafikeinstellungen läuft das Projekt momentan nur auf einem Mittelklasse-Gaming-PC. Die für die Erstellung des Projekts verwendete Hardware besteht aus einem AMD Ryzen™ Threadripper™ 1920X Prozessor, einer NVIDIA GeForce RTX 2070 Super Grafikkarte mit 8GB VRAM und 64 GB DDR4 Arbeitsspeicher. Damit werden bei hoher Grafikeinstellung in 1080p 25-30 FPS erreicht. Bei Testversuchen konnte durch das Erstellen von Light Maps die FPS auf ca. 120 gesteigert werden. Da für die Erstellung von Light Maps aber noch umfangreiche Tests mit Berechnungen der Beleuchtung nötig gewesen wären, wurde auf die Nutzung von diesen im Projekt verzichtet, da 25-30 FPS für den geplanten Einsatz ausreichend sind. Zukünftige Projekte könnten je nach Anwendungszweck und verwendeter Hardware jedoch Light Baking nutzen, um die Performance zu verbessern und die Hardware effizienter zu nutzen. Insbesondere wenn keine dynamische Beleuchtung mit verschiedenen Sonnenständen oder mit dynamischer Tageszeit erforderlich ist, wäre dieser Schritt zu empfehlen.

Um die Performance weiter zu optimieren, sollten die Materialien im Projekt so angepasst werden, dass Ihre Texturierung nicht überdimensioniert ist. Viele Materialien aus der Quixel Bridge-Bibliothek oder der Megascans-Bibliothek nutzen in höchster Qualität eine Auflösung von 8k für alle verwendeten Texturen, die für die meisten Objekte im Projekt nicht benötigt wird. Für die meisten Objekte und Anwendungsfälle sind Texturauflösungen in 2k für die Materialien ausreichend. Mit Unreal Engine kann die benötigte Auflösung beim Einstellen der Texturauflösungen mit Hilfe von Optimierungstools visuell getestet werden. Zudem müssen nicht alle Bestandteile eines Materials die gleiche Auflösung besitzen; Diffuse- und Roughnessmaps können beispielsweise in einer niedrigeren Auflösung eingestellt werden als die Normalmaps eines Materials. Durch solche Anpassungen lässt sich erheblich VRAM einsparen und die benötigte Texture Streaming Pool Size kann minimiert werden. Es ist ratsam, diese Anpassungen für alle Texturen im Projekt vorzunehmen. Im vorliegenden Projekt wurden alle Texturen so weit angepasst, bis es keine Probleme mehr mit der Texture Streaming Pool Size gab. Diese Anpassungen könnten in Kombination mit den Optimierungstools von Unreal Engine aber tiefgreifender optimiert werden.

Beim Erstellen des Modells in InfraWorks® wird bei der Integration von Verkehrswegen das DGM an diesen Stellen automatisch geglättet. Durch diese Glättungsprozesse am DGM sind einige Höhendaten verloren gegangen. Deutlich wird dies nur im Schlosspark Oberschleißheim im Bereich der großen Fontäne, da dort normalerweise ein kleiner Wasserfall vorhanden ist. Eventuell gibt es in InfraWorks® noch Möglichkeiten, diese Glättungen beim Integrieren der Verkehrswege in bestimmten Bereichen zu verhindern. Auch können mit InfraWorks® die Höhen von Gewässern modifiziert werden. Diese Funktion sollte angewendet werden, um Gewässer im DGM einige Meter tiefer anzulegen. In der Unreal Engine würde dies

realistischere Wasserflächen ermöglichen. Anstatt der Oberfläche von Gewässern ein Material zuzuweisen, könnten diese Bereiche mit einem Volumen aufgefüllt werden, das das Wasser simuliert. Dies würde zu einer realistischeren Gewässerdarstellung führen und sogar Flussrichtungen könnten dargestellt werden. Der Avatar würde dann beim Betreten ins Wasser eintauchen, anstatt über das Wasser zu laufen. Das aktuelle DGM mit einer 5-Meter-Maschenweite zeigt beim Absenken der Gewässer am Ufer jedoch Artefakte, die dann möglicherweise geglättet werden müssen, siehe Abbildung 76 und Abbildung 77. Eine Glättung ist auch nachträglich in Unreal Engine möglich, jedoch mit einem hohen Arbeitsaufwand verbunden.

Hierfür wäre es am besten, das DGM vor dem Import in InfraWorks® hinsichtlich der Elevation der Gewässer anzupassen, ähnlich wie es z. B. auch bei hydrologischen Simulationen gemacht wird. Das DGM könnte nach der Bearbeitung nochmal zusätzlich geglättet werden, um Artefakte zu vermeiden. Das Ziel wäre hier nicht die Gewässeroberfläche, sondern den Gewässergrund darzustellen. So könnten störende Artefakte, die durch die harte Kante beim Herabsetzen von Gewässern in InfraWorks® entstehen, eliminiert und ein flaches Ufer modelliert werden. Zum DGM wäre noch anzumerken, dass anfangs ein DGM mit einer 1-Meter-Maschenweite verwendet wurde. Aufgrund der hohen Polygonanzahl hat dieses aber die Performance negativ beeinflusst. Da das DGM nicht in ein natives Landscape konvertiert wird, wie es in Unreal Engine üblich wäre, und diesem dadurch zusätzlich ein hochgenaues Collision Model zugewiesen werden muss, wurde zur Reduzierung der Polygone im Projekt das DGM 1 durch ein DGM 5 ausgetauscht. Für die reine Präsentation von Geodaten ist das DGM 5 auch ausreichend.

5.2 Fazit und Ausblick

Die Digitalisierung des 21. Jahrhunderts ermöglicht es leichter denn je, komplexe Visualisierungen mit Geodaten zu erstellen und diese zusätzlich als interaktive Anwendungen bereitzustellen. Dies ermöglicht Nutzer:innen, Geodaten spielerisch zu erkunden und ohne Wissen über kartographische Abstraktionen für sich zu nutzen. So können auch Fehlinterpretationen vermieden werden, die beim Lesen von Karten entstehen können. Interessanterweise können hierbei vorhandene 2D-Datensätze der Vermessungsämter, die auch als Grundlage für kartographische Werke verwendet werden, mittels Software in eindrucksvolle 3D-Visualisierungen umgewandelt werden. Die Erstellung digitaler Stadt- und Landschaftsmodelle war lange Zeit aufgrund fehlender Software oder spezieller Hardware, die benötigt wurde, eine Herausforderung. Der barrierefreie Zugang zu Geodaten, die heutzutage ebenfalls als Open Data zur Verfügung stehen, erleichtert es nun auch kleinen Unternehmen oder Einzelpersonen, diese für Anwendungen zu nutzen. Dieser Bereich befindet sich noch in der Anfangsphase der Entwicklung und die Möglichkeiten sind nicht vollkommen erforscht. Zudem werden zukünftige Entwicklungen, besonders im Bereich der KI und mit weiteren Soft- und Hardwareinnovationen, die Methoden der Geovisualisierung weiter vorantreiben.

Gerade in der thematischen Kartographie wird dies zukünftig interessante Möglichkeiten eröffnen. Thematische Karten visualisieren spezifische Themen oder Konzepte in geographischen Räumen und werden in Bereichen wie Planung und Entwicklung, Bildung und Forschung, Marketing und Geschäftsanalysen sowie im Tourismus eingesetzt. Wenn Visualisierungen, die sonst in thematischen Karten bereitgestellt wurden, in einer Game Engine interaktiv ausgestaltet werden, eröffnen sich neue Dimensionen der Darstellung. Diese interaktiven Modelle bieten dank ihres Realismus und ihrer simulationsbasierten Ansätze einen barrierefreien Zugang zu den Daten und ermöglichen ein besseres Verständnis des zu vermittelnden Wissens. Traditionelle Karten könnten dieses oft nicht im selben Umfang vermitteln. Eine solche Umsetzung kann gleichzeitig auch das Interesse bei Nutzer:innen oder die Partizipation bei der Nutzung erhöhen.

Ein Anwendungsbeispiel ist die interaktive Touristenführung. Nachdem man einen Ort virtuell erkundet hat, kann der tatsächliche Besuch durch das Vorwissen bereichert werden, da man Zusammenhänge besser einordnen und vertiefen kann. Der virtuelle Tourismus kann nicht nur Menschen mit Behinderungen eine Chance bieten, sehenswürdige Orte zu erleben, er eröffnet zudem die Möglichkeit, interessante Lokalitäten im Voraus zu erkunden und Wissen über diese zu sammeln. Dies kann beim späteren Besuch bei der Orientierung und bei der individuellen Eingrenzung von Points of Interest helfen. Die Planung von Wanderwegen kann durch vorherige virtuelle Erkundung von Steigungen und Aussichtspunkten verbessert werden.

Ein weiteres Beispiel in Bezug auf das Projekt dieser Arbeit wäre eine interaktive Schlossführung in Oberschleißheim, bei der Besucher:innen nicht nur das Schloss, sondern auch dessen historische Aspekte entdecken können oder tiefgreifenderes Wissen anhand von historischen Zeitsprüngen über die Gartenanlage des Schlosses erlangen. Solche interaktiven Führungen könnten auch die Geschichte des Zweiten Weltkrieges in Oberschleißheim anschaulich machen.

Das vorliegende Modell von Oberschleißheim kann aber auch als Gemeindeführer für Einheimische genutzt werden, um den Bürger:innen Bauvorhaben näher bringen zu können und diese in die Planung mit einzubeziehen. Mit passenden Funktionalitäten in der Anwendung kann auch ein Mitbestimmen oder Mitgestalten erleichtert werden.

Das Besondere an dieser Arbeit ist, dass es aktuell nur wenige Anwendungen gibt, die amtliche Geodaten zu reinen Präsentationszwecken nutzen, wie es z. B. in diesem Projekt umgesetzt wurde. Insbesondere das Potenzial, lokalbezogenes Wissen zu vermitteln und möglichst barrierefrei zugänglich zu machen, ist hierbei immens.

Durch die Visualisierung von Geodaten mittels einer Game Engine kann das LDBV z. B. auf Messen das Interesse der Bürger:innen wecken. Interaktive Anwendungen können die Relevanz der Arbeit des LDBV aufzeigen und die Neugier auf Geodaten wecken. Vermessung und Kartographie sind essenzielle Instrumente, die in zahlreichen Bereichen wie z. B. Orientierung und Navigation, Grenzfestlegung, Stadtplanung, Wissenschaft, Ressourcenmanagement, kulturellem Verständnis und technologischen Fortschritten eingesetzt werden. In der technologisierten Welt von

heute können Nutzer:innen anhand von interaktiven Anwendungen für Geodaten begeistert werden.

Durch die Bachelorarbeit sind zudem weitere Fragen für zukünftige Untersuchungen aufgekommen:

Wie kann das World Partition System von Unreal Engine mit Geodaten des LDBV verwendet werden? Dies könnte das Potenzial haben, ganz Bayern wie in einem Open-World-Spiel darzustellen.

Wie können Gebäude im Modell mit Texturen versehen werden?

Wie kann das Modell mit einem präzisen, geodätischen Koordinatensystem versehen und darin georeferenziert werden?

Welche Möglichkeiten bietet das Landscaping Plugin von Ludic Drive für Unreal Engine mit den Geodaten vom LDBV?

Inwiefern könnte eine KI wie z. B. ChatGPT sinnvoll in eine Anwendung integriert werden? Bei Stadtführungen könnte ChatGPT z. B. dazu verwendet werden, aufkommende Fragen zu beantworten.

LITERATURVERZEICHNIS

- Adamo-Villani, N. & Wilbur, R. *Effects of platform (immersive versus non-immersive) on usability and enjoyment of a virtual learning environment for deaf and hearing children* [Veröffentlichung]. Purdue University.
https://www.researchgate.net/publication/267558417_Effects_of_platform_immersive_versus_non-immersive_on_usability_and_enjoyment_of_a_virtual_learning_environment_for_deaf_and_hearing_children
- Arbeitsgemeinschaft der Vermessungsverwaltungen - AdV-Online. (n.d.). Zugriff am 16. Juli 2023, verfügbar unter <https://www.adv-online.de/icc/extdeu/broker.jsp?uMen=4ad505ea-127b-b941-2df2-65a572e13d63>
- ArcGIS Maps SDK for Unreal Engine | ArcGIS Developers. (n.d.). Esri. Zugriff am 5. Juli 2023, verfügbar unter <https://developers.arcgis.com/unreal-engine/>
- Autodesk gibt Innovatoren überall die Möglichkeiten, Neues wahr werden zu lassen. (n.d.). Autodesk®. Zugriff am 9. Juli 2023, verfügbar unter <https://www.autodesk.de/>
- Balasubramanian, K. (2022, 15. September). *Game Engines: All You Need to Know*. Gameopedia. <https://www.gameopedia.com/game-engines-all-you-need-to-know-about/>
- Carter, C., Mehdi, Q. & Hartley, T. (2012). Navigational techniques to improve usability and user experience in RPG games. In *2012 17th International Conference on Computer Games (CGAMES)* (S. 86–93). IEEE.
<https://doi.org/10.1109/CGames.2012.6314557>
- Cesium for Unreal. (n.d.). Cesium. Zugriff am 5. Juli 2023, verfügbar unter <https://cesium.com/platform/cesium-for-unreal/>
- Christian Wasser. (2019, 26. Dezember). *Was ist eine Engine? | SambZockt Show #Unity #Unreal4 #Godot* [Video]. Youtube.
<https://www.youtube.com/watch?v=va95hEZ87bQ>
- Darken, R. (o.d.). *Wayfinding in largescale virtual environments* [Veröffentlichung]. The George Washington University.
https://www.researchgate.net/publication/240430895_Wayfinding_in_largescale_virtual_environments
- Datasmith Twinmotion Importer Plugin (Beta 4) in Code Plugins - UE Marketplace. (n.d.). Epic Games. Zugriff am 19. Juni 2023, verfügbar unter <https://www.unrealengine.com/marketplace/en-US/product/21cab2d84a794b7bab359eb1ba5d3e74>
- delebash. (n.d.). *GitHub - delebash/UnrealMapboxBridgePlugin: Import heightmaps into Unreal from real world locations using Mapbox*. GitHub. Zugriff am 6. Juli 2023, verfügbar unter <https://github.com/delebash/UnrealMapboxBridgePlugin>
- Denisova, A. & Cairns, P. (2015). First Person vs. Third Person Perspective in Digital Games. In B. Begole, J. Kim, K. Inkpen & W. Woo (Hrsg.), *Proceedings of the*

- 33rd Annual ACM Conference on Human Factors in Computing Systems (S. 145–148). ACM. <https://doi.org/10.1145/2702123.2702256>
- Diedrich, S. (2023, 18. Januar). *Unreal Engine im Wandel der Zeit: Die größten Grafikkraher aus 25 Jahren*. GameStar. <https://www.gamestar.de/artikel/25-jahre-unreal-engine-die-schoensten-spiele,3388797.html>
- Domlysz. (n.d.). *GitHub - domlysz/BlenderGIS: Blender addons to make the bridge between Blender and geographic data*. GitHub. Zugriff am 6. Juli 2023, verfügbar unter <https://github.com/domlysz/BlenderGIS>
- Dr. Dennis Edler (2019). Geodaten auf dem Weg in die Game Engine. *Rubin Wissenschaftsmagazin*, 2019(2), 49. <https://news.rub.de/wissenschaft/2019-10-31-standpunkt-geodaten-auf-dem-weg-die-game-engine>
- Epic Games. (2020, 25. März). *Building virtual cities with Unreal Engine*. Unreal Engine. <https://www.unrealengine.com/de/spotlights/building-virtual-cities-and-digital-twins-with-unreal-engine>
- Epic Games. (2021, 4. Oktober). *Introducing Project Anywhere XR: a free sample project for mixed reality 3D geospatial visualization*. Unreal Engine. Cesium. <https://www.unrealengine.com/de/blog/introducing-project-anywhere-xr-a-free-sample-project-for-mixed-reality-3d-geospatial-visualization>
- Funk, L. (n.d.). *Von den Anfängen der Videospiele zu den ersten Computerspielen*. Zugriff am 3. Juli 2023, verfügbar unter <https://www.ludologie.de/spiele/computerspiele/von-den-anfaengen-der-videospiele-zu-den-ersten-computerspielen/>
- InfraWorks: Modellieren Sie Gebäude- und Infrastrukturentwurfskonzepte im realen Kontext*. (n.d.). Autodesk®. Zugriff am 19. Juni 2023, verfügbar unter https://www.autodesk.de/products/infracore/overview?mktvar002=afc_de_buyonline&AID=12185349&PID=8227014&gclsrc=aw.ds&SID=jkp_CjoKCQjwirqk BhCTARIsAAHz7K36SouXw4lFDMnVir66M72ygl6u9KdDi6sfsC7ryklmN8PiXA3TikaAtqrEALw_wcB&cjevent=8a8c414e0e4211ee800daea40a18ba73&affname=8227014_12185349&cjdata=MXxZfDB8WXww&term=1-YEAR&tab=subscription&plc=IW36oP
- IT Referat Landeshauptstadt München. (n.d.). *Digitaler Zwilling München*. Zugriff am 19. Juni 2023, verfügbar unter <https://muenchen.digital/twin/>
- Kohlstock, P. (2018). *Kartographie: Eine Einführung* (4. Aufl.). *utb-studi-e-book: Bd. 2568*. Ferdinand Schöningh. <https://elibrary.utb.de/doi/book/10.36198/9783838549194>
<https://doi.org/10.36198/9783838549194>
- Landscaping in Code Plugins - UE Marketplace*. (n.d.). Unreal Engine. Zugriff am 5. Juli 2023, verfügbar unter <https://www.unrealengine.com/marketplace/en-US/product/landscaping>
- Ludic Drive. (n.d.). *Landscaping - GIS data Importer for Unreal Engine*. Zugriff am 10. Juli 2023, verfügbar unter <https://landscaping.ludicdrive.com/>

- Marc Ottensmann. (2016). *Game-Engines get real with Unreal & Co* [Masterarbeit]. Hochschule Ostwestfalen-Lippe.
- Mohammed, S. (2021, 4. Oktober). *Project Anywhere XR Now Available for Free on Unreal Engine Marketplace*. Cesium. <https://cesium.com/blog/2021/10/04/project-anywhere-xr-now-available-for-free-on-unreal/>
- Myriam Seidt & Katharina Wolter. (1999). *Unterstützung der Orientierung und Navigation in realen Räumen durch den Einsatz von Gebäudekarten und DesktopVR* [Studienarbeit]. Universität Hamburg.
- Olaf Bleich, Benedikt Plass-Fleßenkämper & Maria Beyer-Fistrich. (2019, 14. Juli). *Unreal und die Unreal Engine: Eine unneale Erfolgsgeschichte*. PC Games. <https://www.pcgames.de/Unreal-Spiel-40639/Specials/unreal-Engine-Spiele-1294786/>
- Olberding, H. (2018a). *Untersuchung zum Potenzial von Game Engines für Beleuchtungssimulationen im 3D-Stadtmodell* [Masterarbeit]. Jade, Wilhelmshaven Oldenburg Elsfleth. <https://doi.org/10.13140/RG.2.2.25661.20969>
- Olberding, H. (2018b). *Untersuchung zum Potenzial von Game Engines für Beleuchtungssimulationen im 3D-Stadtmodell* [Masterarbeit Präsentation]. Jade, Wilhelmshaven Oldenburg Elsfleth.
- Potor, M. (2018, 29. November). *Virtual Helsinki: Finnische Hauptstadt will mit VR-Reisen das Klima retten*. Basic thinking. <https://www.basicthinking.de/blog/2018/11/29/virtual-helsinki-vr-reisen-klimaschutz/>
- Ronen Bekerman. (2019, 11. April). *Virtual Helsinki Case Study*. <https://www.ronenbekerman.com/virtual-helsinki-case-study/>
- Schutz, T. & Schwarz, D. (2022). *Digital game-based learning: Komplexe Problemlösungen und Kompetenzen für Bildung, Wirtschaft und Politik entwickeln. essentials*. Springer Gabler. <https://doi.org/10.1007/978-3-658-37524-9>
- Schwering, A., Krukar, J., Li, R., Anacta, V. J. & Fuest, S. (2017). *Wayfinding Through Orientation*. *Spatial Cognition & Computation*, 17(4), 273–303. <https://doi.org/10.1080/13875868.2017.1322597>
- Truong Bui. (n.d.). *Async Loading Screen in Code Plugins - UE Marketplace*. Epic Games. Zugriff am 19. Juni 2023, verfügbar unter <https://www.unrealengine.com/marketplace/en-US/product/async-loading-screen?sessionInvalidated=true>
- Twinmotion Content for Unreal Engine Plugin (Beta 4) in Code Plugins - UE Marketplace. (n.d.). Epic Games. Zugriff am 19. Juni 2023, verfügbar unter <https://www.unrealengine.com/marketplace/en-US/product/twinmotion-content-for-unreal-engine-plugin-beta>

Twinmotion is a cutting-edge real-time visualization tool. (n.d.). Epic Games. Zugriff am 19. Juni 2023, verfügbar unter <https://www.twinmotion.com/en-US>

Unreal Engine 5.1 Documentation. (n.d.). Epic Games. Zugriff am 19. Juni 2023, verfügbar unter <https://docs.unrealengine.com/5.1/en-US/>

Unreal Engine 5.1 Documentation Scalability Reference. (n.d.). Epic Games. Zugriff am 18. Juli 2023, verfügbar unter <https://docs.unrealengine.com/5.1/en-US/scalability-reference-for-unreal-engine/>

Videospielhalbwissen. (n.d.). *Tennis For Two (1958)*. Zugriff am 3. Juli 2023, verfügbar unter <https://videospielhalbwissen.de/timeline/tennis-for-two-1958/>

ZOAN - Virtual Helsinki. (n.d.). Zugriff am 2. Juli 2023, verfügbar unter <https://zoan.com/cases/virtual-helsinki/>

ANHANG



[Dokumentation InfraWorks](#)



[Dokumentation Twinmotion](#)



[Dokumentation Unreal Engine](#)



[Videopräsentation fertiges Projekt](#)



[Fertiges Projekt .exe](#)

EIDESSTATTLICHE ERKLÄRUNG

Ich versichere, dass ich diese Arbeit selbstständig angefertigt, nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen oder Hilfsmittel benützt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.



Unterschleißheim, den 01.08.2022

Sinan Babernits