



HÖHERE TECHNISCHE BUNDESLEHRANSTALT Wien 3, Rennweg
IT & Mechatronik

HTL Rennweg :: Rennweg 89b
A-1030 Wien :: Tel +43 1 24215-10 :: Fax DW 18

Diplomarbeit

Autor
richtig
setzen

Metatags
setzen

G.O.I. Graveyard of Immortals

ausgeführt an der
Höheren Abteilung für Informationstechnologie/Medientechnik
der Höheren Technischen Lehranstalt Wien 3 Rennweg

im Schuljahr 2018/2019

durch

**Hillinger Stefan
Lichtenstein Benjamin
Mendel Tobias
Röhrer Tobias
Seidel Hans**

unter der Anleitung von

Matejowsky Peter
Dazinger Robert, Sturm Gerhard

Wien, 7. März 2019

Kurzfassung

Bei unserer Diplomarbeit handelt es sich um ein Horror-Spiel. Wir wollen ein Gruselerlebnis, Spannung, Angst und Schreckmomente in hoher Qualität darbieten. Ein Erlebnis, in welches man sich hineinfallen lassen kann – und das in einer sicheren Umgebung.

Die Umwelt im Spiel ist allerdings gar nicht sicher. Dort muss man sich durch Gräber, Mausoleen und Häuser schleichen, um wichtige Objekte zu sammeln. Doch das wird kein Spaziergang, denn man ist nicht allein.

Das Abenteuer wird begleitet von einer umfangreichen fiktiven Geschichte, welche im Spielverlauf mithilfe von Notizzetteln herausgefunden werden kann. Zusätzlich gibt es auch viele interessante Informationen über bekannte Menschen. Auf Informationstafeln kann man mehr über unsere Geschichte erfahren, oder sein vorhandenes Wissen auffrischen.

Abstract

Our diploma-project is a horror game. We want to provide tension, fear, and jump scares in high quality. An immersive experience – in a safe environment.

The in-game environment, though, is not safe at all. You will have to navigate through graves, mausolea and other buildings to acquire important objects. Don't think that it will be easy, as you are not alone.

The adventure is accompanied by an extensive fictional story, which will be explained in-game via various notes. In addition to this, you will be able to find out more about important real-life personas to expand or refresh your knowledge of the world.

Ehrenwörtliche Erklärung

Ich erkläre an Eides statt, dass ich die individuelle Themenstellung selbstständig und ohne fremde Hilfe verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

Wien, am 7. März 2019

Hillinger Stefan

Lichtenstein Benjamin

Röhner Tobias

Seidel Hans

Inhaltsverzeichnis

Tabellenverzeichnis	xiii
----------------------------	-------------

Abbildungsverzeichnis	xv
------------------------------	-----------

1	Ziele	1
2	Formatierung	3
2.1	Eine Überschrift	3
2.1.1	Eine Unterüberschrift	3
3	Blender	5
3.1	Einleitung	5
3.2	Modellierungsprogramm	5
3.3	Object Mode	6
3.3.1	Smooth Shading	6
3.3.2	Flat Shading	6
3.3.3	Ebenen	6
3.4	Edit Mode	6
3.4.1	Seperate	7
3.5	Modifikatoren	7
3.5.1	Boolean	7
3.5.2	Mirror	7
3.5.3	Array	8
3.5.4	Curve	9
3.5.5	Bevel	9
3.5.6	Decimate	10
3.5.7	EdgeSplit	10
3.6	Viewport Shading	11
3.6.1	Solid	11
3.6.2	Wireframe	11
3.7	Texturierung Vorarbeiten	13
3.8	Charakter Gestaltung	13
3.8.1	Skizzen zu dem Modell	13
3.8.2	Grundmodellierung	13
3.8.3	Sculpting Feinmodellierung	13
3.8.4	Menschliche Relationen	13
3.8.5	Normal-Map erstellen	13

3.8.6	UV-Map erstellen	13
3.9	Rigging	13
3.9.1	Geschichte	13
3.9.2	Allgemein Rigs	13
3.9.3	Rigging in Blender	13
3.9.4	Umsetzung der Rigs	13
3.10	Animation	13
3.10.1	Animationstheorie	13
3.10.2	Geschichte	13
3.10.3	3D-Animation	13
3.10.4	Animation in Blender	13
3.10.5	Umsetzung der Animationen	13
3.11	Modellierung von 3D Objekten	13
3.11.1	Paracelsus Grab	14
3.11.2	Bettdecke	19
3.12	Zusammensetzung mehrerer 3D Objekte	19
3.12.1	Haus	19
3.13	Exportieren von Blender zu Unreal Engine 4	20
3.13.1	3D Modelle	20
3.13.2	Simulationen	20
4	Unreal Engine	21
4.1	Blueprints	22
4.1.1	Classes	22
4.1.2	Nodes	22
4.1.3	HO-Interaktion	22
4.1.4	Grabwächter	22
4.2	Interface	22
4.2.1	Startmenü	22
4.3	Export	22
4.4	Texturen	22
4.4.1	Grundsätzlicher Unterschied zwischen generierten und gemappten Texturen	22
4.4.2	Verschieden Arten von Texturen	22
4.4.3	Bump-Textur und Normal-Textur	22
4.4.4	Belichtungstexture	22
4.4.5	Höhenberichtung mittels Texturen	22
4.4.6	Fotobearbeitung	22
4.4.7	Umsetzung der Texturierung in der Unreal-Engine	22
4.5	Materialien	22
4.5.1	Physikbasierende Materialien	22
4.5.2	Material-Ebenen in der Unreal-Engine	22
4.5.3	Emmission Parameter	22
4.5.4	Material Transition	22
4.5.5	Simulierte Materialien	22
4.6	Lighting	22
4.6.1	Vorwissen	23

4.6.2	Directional Light	23
4.6.3	Sky Light	23
4.6.4	Sky Sphere	23
4.6.5	Post Process Volume	23
4.6.6	Exponential Height Fog	24
4.6.7	Lightmass Importance Volume	24
4.6.8	Light Mobility	24
4.6.9	Das Lighting im Spiel	25
5	Sound	27
6	Website	29
6.1	Intro	29
6.2	CMS	29
6.3	Plugins	30
6.4	Host	30
6.5	Inhalte	30
6.5.1	Diplomarbeit	30
6.5.2	Spiele	30
6.5.3	Sonstiges	30
6.6	Design	30
6.6.1	Template	30
6.6.2	Abänderungen	30
A	Anhang 1	31
	Literaturverzeichnis	33

Tabellenverzeichnis

kann
entfal-
len falls
(fast) leer

Abbildungsverzeichnis

2.1	Ein Bild im Kapitel Chapter mit dem Namen image	4
3.1	Difference-, Union- und Intersect Modifikator	8
3.2	Mirror Modifikator mit Spiegelung um die Z-, Y- und X-Achse	8
3.3	Array Modifikator	9
3.4	Curve Modifikator	9
3.5	Bevel Modifikator	10
3.6	Decimate Modifikator	10
3.7	Edgesplit Modifikator	11
3.8	Import von Bildern	14
3.9	Paracelsus Grab Nachmodellierung	16
3.10	Links: Vase mit den Verzierungen. Rechts: Bezier-Circle	17
3.11	Einfügen der Schwarz/Weiß Textur	18
3.12	Verzierung des Paracelsus Grabs	18
3.13	Bettdecke mit Forcefields(orange)	19
3.14	Objekte des Hauses, auf mehreren Ebenen verteilt	20
4.1	Objekte in der Lighting Only Ansicht von Unreal Engine	26

1 Ziele

Kurz in Worten zusammengefasst, was die Ziele der Diplomarbeit sind.

Ziele hin-
einschrei-
ben

2 Formatierung

Ein Kapitel

2.1 Eine Überschrift

Ein Todo

2.1.1 Eine Unterüberschrift

Noch ein TODO

2.1.1.1 Eine Unter Unterüberschrift

Die Beschreibung kann im Ordner text im File Formatierung gefunden werden.
underline *kursiv* **fett**

Für einen neuen Absatz einfach eine Zeile auslassen.

Unordered List :

- Item 1
- Item 2

Nested List mit Nummern am Anfang :

1. First level item
 - a) Second level item
 - b) Second level item
2. First level item

Hier ein Link : -> Linkname <- Bitte Hovern.

Sonderzeichen werden bei Latex mit einem Backslash (\) maskiert oder mit Befehlen erzeugt. z.B. \$ & % # | { } § ¶ • ©

- „Deutsche Anführungszeichen“
- ‚Halbe deutsche Anführungszeichen‘
- "Doublequotes"

Zeichen schreibt man mit `\verb + Ein Zeichen + Die zwei +` können beliebige Zeichen sein und geben den Anfang und den Schluss an.

Fußnoten¹ mit \LaTeX sind kein Problem² Dank des Befehls `\footnote`.

Zitate:

Die `quote`-Umgebung ist nicht nur für Zitate eine beliebte Form der Text-hervorhebung, bei der der Text beidseitig eingerückt wird.

Falls das Bild keinen richtigen Abstand hat, keine Panik! Bitte beim Lokalen TEX-Verwalter melden. Bilder können sehr viele Formate haben und müssen nicht unbedingt ein png sein. Für Screenshots kann ich das Programm Lightshot nur empfehlen.

Eine referenz zum Bild -> 2.1 Bitte raufklicken.

Das Bild sollte bei allen eine einheitliche Größe haben. (max. 2 verschieden Größen). Wir sollten das noch absprechen, bzw. ausprobieren wie es am Besten passt.

In Firefox kann man irgendwie mit erhöhter Auflösung Screenshots machen, für gschmeidige Bilder :)



Abbildung 2.1: Ein Bild im Kapitel Chapter mit dem Namen image

So referenziert man auf eine Überschrift : Abschnitt 2.1 eine Überschrift

¹ Die erste Fußnote

² Die zweite Fußnote

3 Blender

3.1 Einleitung

Bei der 3D Modellierung, werden Objekte in einem dreidimensionalen Raum erstellt. Mit anderen Worten, es gibt ein Koordinatensystem mit einer X-, Y- und Z-Achse. In diesem werden Punkte gesetzt, die auch zu Kanten und Flächen verbunden werden können. Mehrere solcher Punkte, Kanten und Flächen ergibt dann ein 3D-Objekt. Diese 3D-Objekte können für alle möglichen Zwecke eingesetzt werden. Damit man sich etwas darunter vorstellen kann, folgen nun ein paar Beispiele für eine mögliche Anwendungen.

Anwendungsmöglichkeiten:

- Echte Objekte nachmodellieren und digital z.B. in einem Bild verwenden.
- Objekte erschaffen und für z.B. Spiele verwenden.
- Objekte für Veranschaulichungen von Zukunftsprojekten erstellen z.B. ein Wohnhaus.

3.2 Modellierungsprogramm

Es gibt viele Programme mit denen man 3D-Objekte erstellen kann. [14] Populäre Programme dieser Art sind u. a.:

- Maya
- Cinema 4D
- LightWave 3D
- Blender
- 3ds Max

- Houdini

Unser Team hat sich entschieden die benötigten Modelle in Blender zu erstellen, weil wir mit dem Programm im Unterricht arbeiten und es kostenlos verwendbar ist.

Mit den Kapiteln Object Mode, Edit Mode und Modifikatoren, folgen Erläuterungen zu den von uns verwendeten Tools die öfters zum Einsatz kamen.

3.3 Object Mode

Der Object Mode in Blender ist dazu da um ein ganzes 3D-Objekt zu verändern. Man kann es Beispielsweise im Koordinatensystem ausrichten, aber auch rotieren, skalieren oder sogar den Mittelpunkt des Objekts verändern. Das verändern des Mittelpunkts ist besonders nützlich, um Objekte auf einem bestimmten Punkt zu verändern. Wie wir im Abschnitt 3.13 "Exportieren von Blender zu Unreal Engine 4" noch erfahren, hat uns dieses Feature besonders geholfen.

3.3.1 Smooth Shading

3.3.2 Flat Shading

3.3.3 Ebenen

3.4 Edit Mode

Im Edit Mode kann man alle möglichen Veränderungen an einem Objekt durchführen. Man kann zum Beispiel Punkte, Kanten und Flächen bewegen, löschen oder hinzufügen. Es gibt aber viel mehr Funktionen, die für unser Projekt mitunter auch in Sachen Effizienz wichtig waren. Eine dieser Funktionen ist der Magnet. Mit ihm kann man Punkte, Kanten oder Flächen zu 100% genau an die Koordinaten anderer Punkte, Kanten oder Flächen verschieben ohne die Werte der Position anzugeben.

Die verwendeten Funktionen werden in Abschnitt 3.11 "Modellierung von 3D Objekten" anhand von Praxisbeispielen gezeigt.

3.4.1 Seperate

3.5 Modifikatoren

In diesem Kapitel, werden die Modifikatoren beschrieben, die für dieses Projekt oft verwendet wurden. Es werden nur Funktionen außerdem nur Funktionen der Modifikatoren beschrieben, die von uns genutzt wurden.

Allgemein kann man zu Modifikatoren noch sagen, dass sie Objekte in einer gewissen Art und Weise, abhängig davon welcher Modifikator verwendet wird, verändern.

3.5.1 Boolean

[3] Der Boolean Modifikator macht aus zwei Objekten ein Objekt. Dazu gibt es drei Boolean-Operationen, die man anwenden kann.

Diese heißen:

- Difference
- Union
- Intersect

Bei Difference werden bei zwei überlappenden Objekten, die überlappenden Teile ausgeschnitten. Beim Zielobjekt ist nun ein Loch an der Gewünschten stelle (Abbildung 3.1 links).

Bei Union werden zwei Objekte zu einem Objekt zusammengefügt (Abbildung 3.1 mitte).

Bei Intersect wird das Objekt mit dem Modifier, an den Überschneidenden stellen in das Zielobjekt eingefügt (Abbildung 3.1 rechts).

3.5.2 Mirror

[7] Der Mirror Modifikator spiegelt ein Objekt um den eigenen Mittelpunkt. Den Mittelpunkt kann man festlegen indem man im Editmode einen Punkt auswählt und mit SHIFT + S -> Cursor to Selected den Cursor setzt und anschließend im Objectmode mit STRG + SHIFT + ALT + C -> Origin to 3D Cursor den Mittelpunkt setzt.

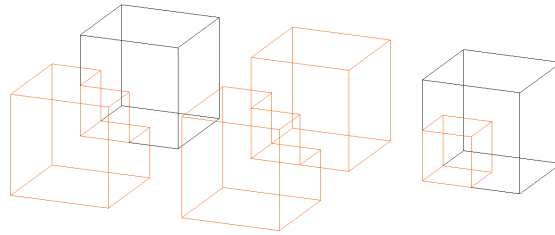


Abbildung 3.1: Difference-, Union- und Intersect Modifikator

In Abbildung 3.2 kann man die Anwendung des Mirror Modifikatoren sehen. Der Orangene Teil des Objekts, ist in allen drei Beispielen das Grundobjekt, welches gespiegelt wird.

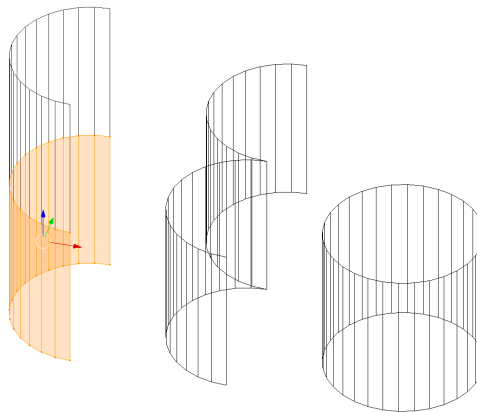


Abbildung 3.2: Mirror Modifikator mit Spiegelung um die Z-, Y- und X-Achse

3.5.3 Array

[1] Der Array Modifier vervielfacht ein Objekt entlang einer Achse. Außerdem kann man den relativen oder absoluten Abstand zwischen den Objekten angeben, damit diese regelmäßig wiederholt werden. In Abbildung 3.3 sieht man den angewendeten Array Modifikator. Die ausgewählten (orange umrandeten) Würfel werden auf der Y-Achse wiederholt, die anderen auf der Y- und Z-Achse.

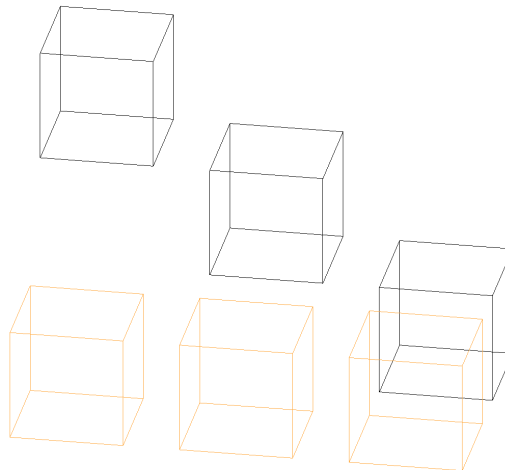


Abbildung 3.3: Array Modifikator

3.5.4 Curve

[4] Mit dem Curve Modifikator, kann man ein beliebiges Objekt an eine Bezierkurve anpassen. In Abbildung 3.4 sieht man in der oberen Hälfte, das Objekt und die Bezierkurve, in der unteren beide Zusammgefügt.

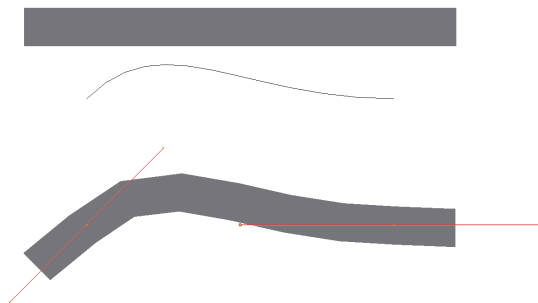


Abbildung 3.4: Curve Modifikator

3.5.5 Bevel

Mit dem Bevel Modifikator [2] kann man Kanten durch Flächen, die nahe der Kanten hinzugefügt werden, abrunden (Abbildung 3.5). Wichtige Einstellungen sind der Abstand von der am äußersten hinzugefügten und der ursprünglichen Kante und die Anzahl der Kanten die hinzugefügt werden.

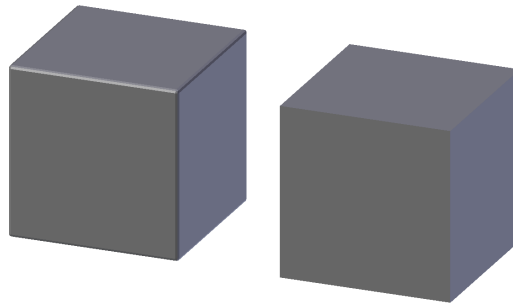


Abbildung 3.5: Bevel Modifikator

3.5.6 Decimate

Mit dem Decimate Modifier [5] kann man bei einem Objekt die Faces reduzieren. Zum Einsatz kommt der Decimate Modifier bei komplizierten Objekten die viele Faces brauchen um detailliert modelliert zu werden. Man entfernt dann aber so viele Faces wieder, dass das Objekt noch gut genug aussieht. Dieser Prozess verbessert am Ende sehr stark die Performance.

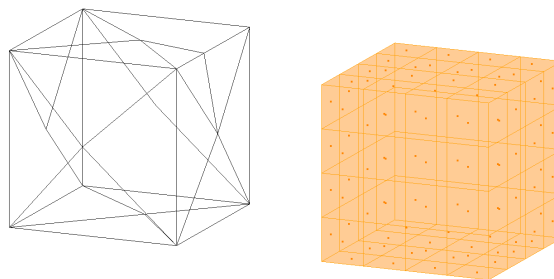


Abbildung 3.6: Decimate Modifikator

3.5.7 EdgeSplit

Mit dem Edgesplit Modifikator kann man [6] bei einem Objekt bestimmen, ab welchem Winkel Smooth Shading angewendet wird. In Abbildung 3.7 links, wird Smooth Shading gar nicht angewendet, in der Mitte wird es nur an den Kanten mit den geringsten Winkel zu den Flächen angewendet und rechts wird Smooth Shading überall angewendet.

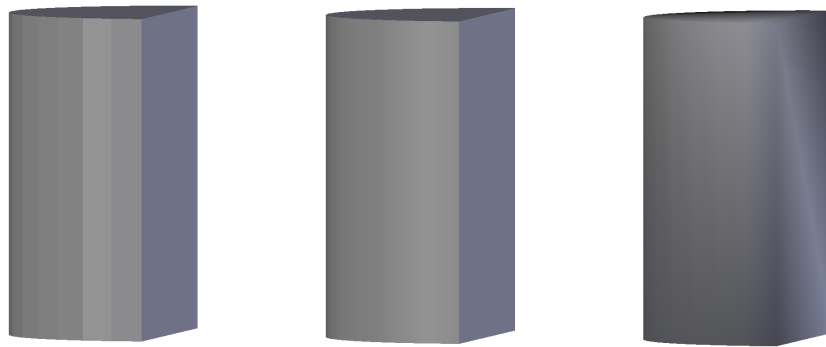


Abbildung 3.7: Edgesplit Modifikator

3.6 Viewport Shading

3.6.1 Solid

3.6.2 Wireframe

3.7 Texturierung Vorarbeiten

3.8 Charakter Gestaltung

3.8.1 Skizzen zu dem Modell

3.8.2 Grundmodellierung

3.8.3 Sculpting Feinmodellierung

3.8.4 Menschliche Relationen

3.8.5 Normal-Map erstellen

3.8.6 UV-Map erstellen

3.9 Rigging

3.9.1 Geschichte

3.9.2 Allgemein Rigs

3.9.3 Rigging in Blender

3.9.4 Umsetzung der Rigs

3.10 Animation

3.10.1 Animationstheorie

3.10.2 Geschichte

3.10.3 3D-Animation

Es ist sehr wichtig, dass das Spiel am Ende mit einer guten Performance spielbar ist. Deswegen wurde beim Modellieren darauf geachtet, das es so wenig Flächen wie möglich pro Modell gibt [11], denn diese müssen vom Computer berechnet werden. Darunter kann dann die Performance im Spiel leiden.

3.11.1 Paracelsus Grab

Das Grab des Paracelsus ist ein Modell welches einem echten Objekt entspricht. Um ein originalgetreues Ergebnis zu erhalten, modelliert man das Objekt von einem Foto nach. Dazu muss man ein Foto in Blender importieren. Um ein Foto zu importieren muss man sich im "Object Mode" befinden und "N" drücken. Im Menü sieht man dann das Feld Background Images (Abbildung 3.8). Dort kann man dann den Pfad des Bildes angeben und einstellen bei welchem Winkel zu welcher Achse sichtbar ist. In diesem Fall ist das Bild sichtbar, wenn man genau auf die Achsen X und Z schaut. Man kann außerdem die durchsichtigkeit des Bildes einstellen, in diesem Fall ist es zu 100% sichtbar, damit man während dem modellieren die Linien im Bild noch gut erkennt.

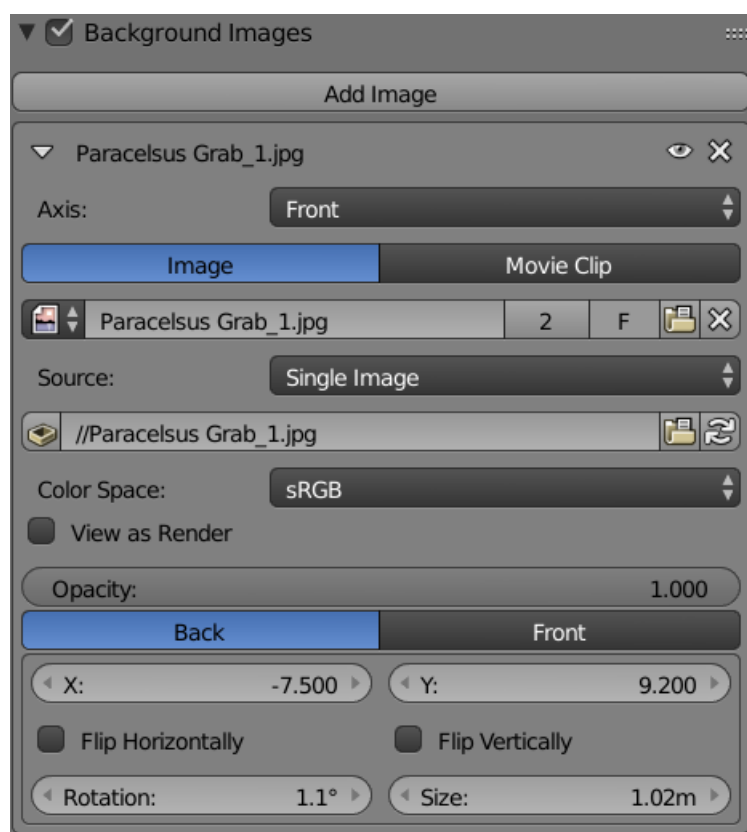


Abbildung 3.8: Import von Bildern

Nun modelliert man mit Formen das Grab nach und verschiebt im Editmode einzelne

Punkte so das sie passen. Man muss allerdings darauf achten, das in diesem Bild (Abbildung 3.9) Verzerrungen aufgrund des Aufnahmewinkels des Fotos auftreten, deshalb sind einige Stellen nicht genau, sondern durch eine Schätzung der Größen nachmodelliert worden.

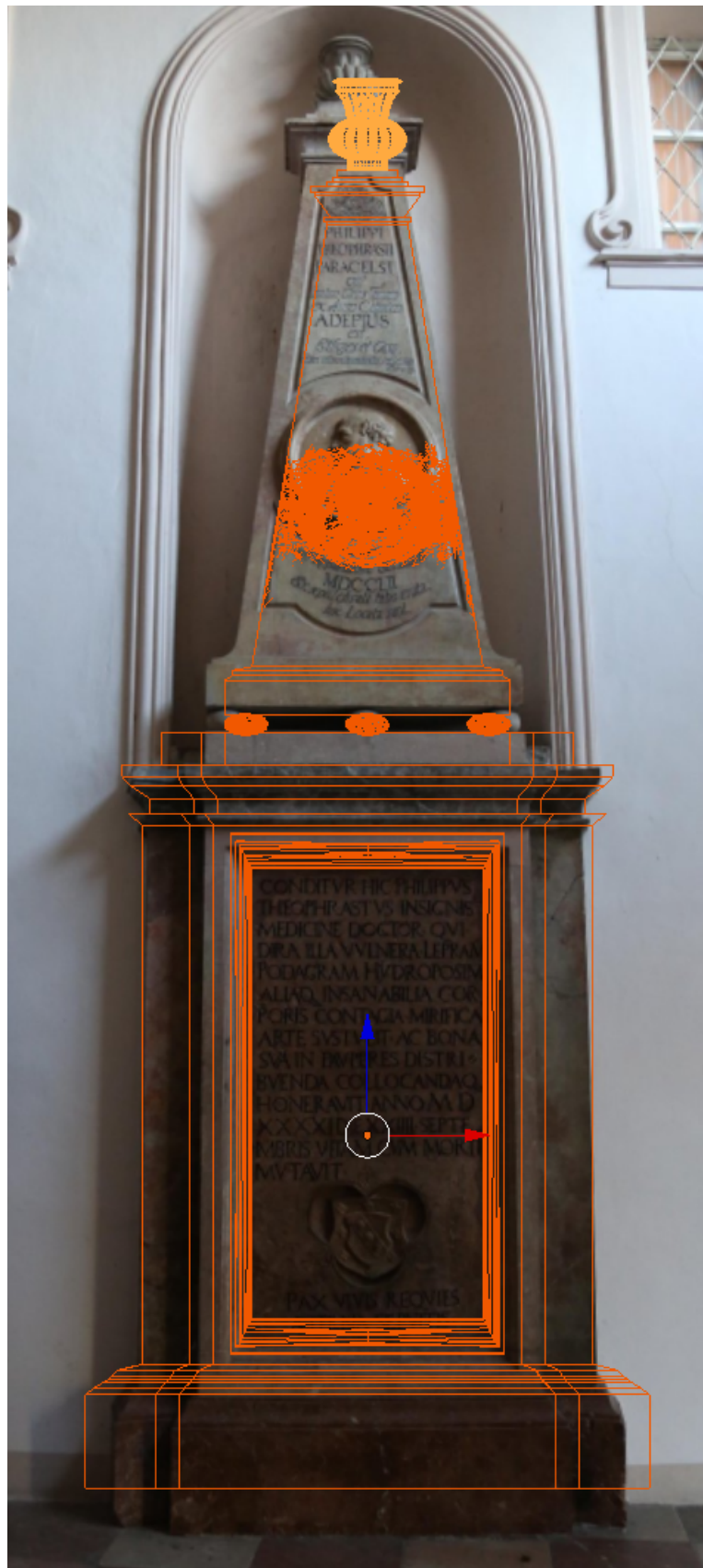


Abbildung 3.9: Paracelsus Grab Nachmodellierung

3.11.1.1 Vase

Um die Vase für das Paracelsusgrab zu modellieren, wurde eine Sphere benutzt und anschließend im Editmode verändert. Anschließend wurden die Verzierungen für die Vase gemacht. Dazu wurde zwei Curves erstellt (Abbildung 3.10), die obere für die Einkerbungen und die untere für die Verzierung der Vase. Damit die Curves eine Breite haben, muss man sie mit einem Bezier-Circle verbinden, der die Breite bestimmt. Nach dem erstellen der Curve, wurde sie mit Alt + C zu einem Mesh konvertiert, damit sie dann mit der Vase verbunden werden können.



Abbildung 3.10: Links: Vase mit den Verzierungen. Rechts: Bezier-Circle

3.11.1.2 Gesicht

Für die Modellierung des Gesichts, wurde ein Brush verwendet. Diese Methode braucht zwar viele Flächen, was die Performance des Spiels beeinträchtigen kann, war aber notwendig, damit nachher noch eine andere Textur mit einer eigenen Bumpmap verwendet werden kann. Um das Gesicht zu erstellen wurde zuerst auf einer Seite der Säule, die Fläche mit dem Shortcut W + Subdivide (Edit Mode) unterteilt. Dann wurde im Sculptmode eine Schwarzweiß Textur eingefügt (Abbildung 3.11), damit man mit dem Brush Höhen und Tiefen Zeichnen kann. Außerdem wurden die Werte Strength und Radius angepasst. Nun wurde der Brush angewendet. Danach sind noch mit dem Decimate-Modifikator so viele Flächen wie möglich entfernt worden, ohne dass das Aussehen der Statue stark beeinträchtigt wurde.

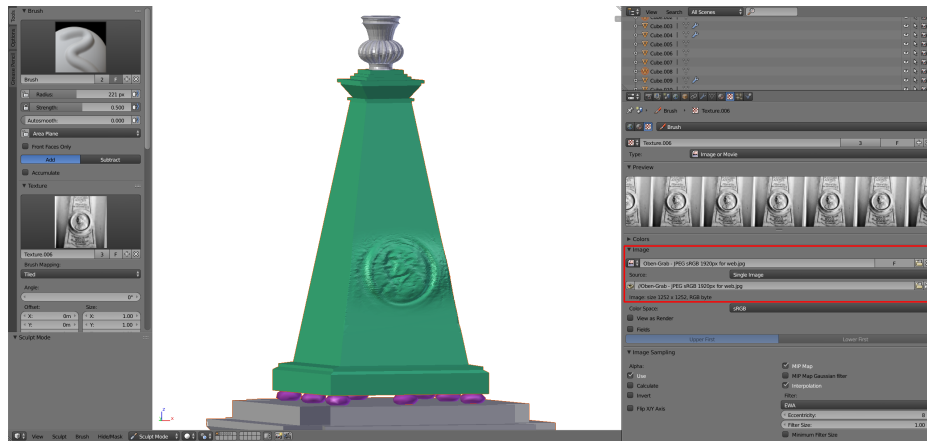


Abbildung 3.11: Einfügen der Schwarz/Weiß Textur

3.11.1.3 Verzierungen

Nachdem die Verzierung des Paracelsus Grabes (Abbildung 3.12) einer bereits vorhandenen Form sehr ähnlich sieht, nämlich der Verzierung des Mausoleums, wurde diese übernommen und mit dem Sockel des Grabes zusammengefügt. Dazu wurde zuerst mit einem Würfel und dem Boolean-Modifikator ein Stück aus dem Sockel herausgeschnitten, damit sich die Verzierung nicht mit dem Sockel überschneidet. Danach wurden der Sockel und die Verzierung mit STRG + J zusammengefügt. Die Verzierung selber, wurde mit einer Fläche modelliert, die dann im Editmode so angepasst wurde, dass sie aussieht wie eine Verzierung.



Abbildung 3.12: Verzierung des Paracelsus Grabes

3.11.2 Bettdecke

Die Bettdecke sollte etwas zerknüllt aussehen. Deshalb wurde sie mit einer Simulation modelliert. Damit die Decke sich verformt, muss sie genug Flächen haben. Dies kann man erreichen, indem man eine Plane erstellt, in den Editmode wechselt und mit W + Subdivide mehrere Flächen erzeugt. Anschließend wurde auf die Decke ein Solidify Modifikator angewendet, damit sie auch eine Dicke hat. Dann wurden Forcefields hinzugefügt (Abbildung 3.13) um die Decke in Bewegung zu bringen. Damit sich die Decke aber auch wirklich bewegt muss man die Decke auswählen, in den Physics Tab wechseln und Cloth auswählen. Jetzt verhält sich die Decke wie ein Stoffstück. Wenn man nun auf der Timeline auf Play drückt, bewegt sich die Bettdecke und verformt sich. Wenn die Form passt, pausiert man die Simulation. Dann drückt man Alt + C und wählt Mesh from Curve/Meta/Surf/Text aus, somit kann man das fertige Modell frei bewegen.

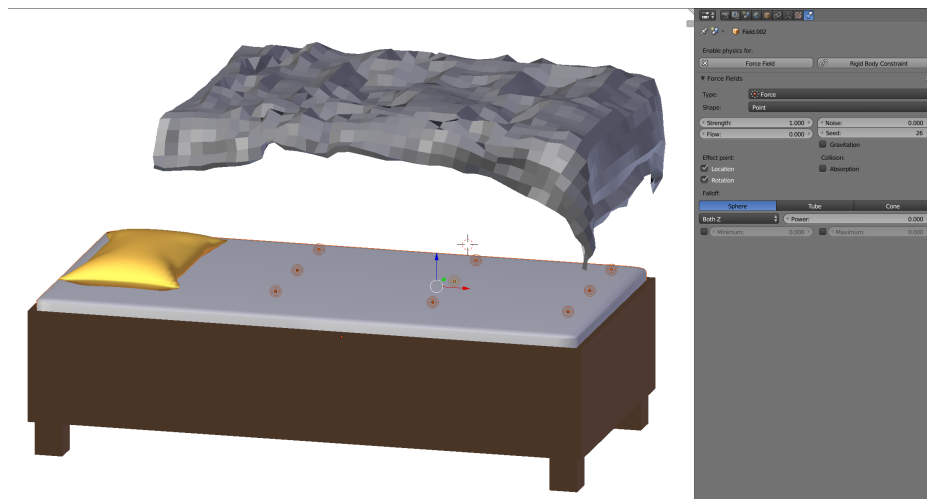


Abbildung 3.13: Bettdecke mit Forcefields(orange)

3.12 Zusammensetzung mehrerer 3D Objekte

bessere
Über-
schrift

3.12.1 Haus

Um das Haus zu modellieren, musste zuerst die Größe gut eingeplant werden, denn es musste so groß sein, dass einige Objekte hineinpassen. Außerdem muss alles z.B. Türen, Gänge und die Stockwerkhöhe im Haus ungefähr an die Größe der Charaktere angepasst sein. Die Objekte im Haus sind angelehnt an echte Objekte, z.B. sind die Stufen von echten Stufen abgemessen worden und in realistischen Maßen an das Haus angepasst worden.

Beim Modellierungsvorgang, wurde zuerst das Haus modelliert und anschließend die Objekte im Haus. Die Objekte wurden pro Stockwerk auf Ebenen (Die roten Pfeile auf Abbildung 3.14 zeigen die Ebenen auf) in Blender verschoben, damit man gesamte Stockwerke ausblenden kann. Um die Objekte richtig zu platzieren, wurde auf Orthogonale Ansichten umgeschaltet und das Magnettool benutzt um die Objekte ganz genau zu den Wänden des Hauses zu verschieben. Anschließend wurde der Abstand noch auf einen realistischen Abstand angepasst.

Nachdem alle Objekte für das Haus erstellt worden sind, wurden diese einzeln exportiert und in einem neuen Blender File wieder zusammengefügt. Genauere Informationen zum Exportieren, kann man unter dem Kapitel Exportieren von Blender zu Unreal Engine 4 finden. Die Objekte werden neu zusammengefügt, damit keine Nebenprodukte die beim modellieren im Haus angefallen sind, im zusammengefügten Modell bestehen bleiben. Außerdem ist es gut alle Objekte einzeln zu haben, falls man sie zu einem anderen Zweck noch einmal benötigt.

r3

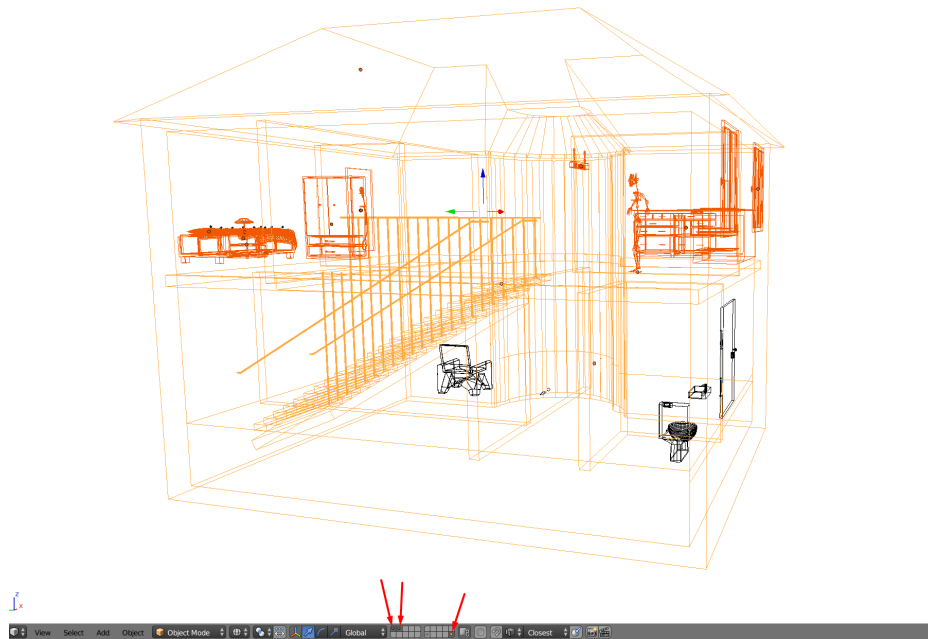


Abbildung 3.14: Objekte des Hauses, auf mehreren Ebenen verteilt

3.13 Exportieren von Blender zu Unreal Engine 4

3.13.1 3D Modelle

3.13.2 Simulationen

r1

r3

Was ist
eine Zeile
darüber

4 Unreal Engine

4.1 Blueprints

4.1.1 Classes

4.1.2 Nodes

4.1.3 HO-Interaktion

4.1.4 Grabwächter

4.2 Interface

4.2.1 Startmenü

4.3 Export

4.4 Texturen

4.4.1 Grundsätzlicher Unterschied zwischen generierten und gemappten Texturen

4.4.2 Verschieden Arten von Texturen

4.4.3 Bump-Textur und Normal-Textur

4.4.4 Belichtungstexture

4.4.5 Höhenberichtung mittels Texturen

4.4.6 Fotobearbeitung

4.6.1 Vorwissen

Eine uns bekannte und Beleuchtungsmethode ist die 3-Punkt-Beleuchtung. Bei ihr wird klassischer weißes ein Objekt von 3 Seiten mittels Key-, Fill- und Backlight ausgeleuchtet. Dadurch entsteht eine Tiefe und eine Stimmung die je nach Lichteinfall und Lichtstärke variieren kann.

Es gibt verschiedene Lampen, um Personen oder Objekte auszuleuchten. Je konzentrierter das Licht von einer Lampe wegstrahlt und auf ein Objekt fällt, desto härter ist der Schatten den es wirft. Wenn eine Lampe weiter weg ist, kommt weniger Licht am Objekt an.

4.6.2 Directional Light

Das Directional Light simuliert ein Licht was unendlich weit weg ist. Somit kommt das Licht nur von einer Seite, weshalb man es gut als Sonne verwenden. [8]

4.6.3 Sky Light

Das Sky Light strahlt Licht von allen Seiten aus. Man kann somit die Farbe und Lichtstärke auf der Rückseite von Objekten kontrollieren.

4.6.4 Sky Sphere

Die Sky Sphere gibt definiert den Himmel in Unreal Engine. Mit ihr, kann man Wolken und Sterne bearbeiten. Manche Einstellungen funktionieren nur in Relation mit anderen Lichtquellen. Man kann zum Beispiel nur Sterne sehen, wenn man die Sonne in einem bestimmten Winkel platziert. Außerdem kann man mehrere Farbeinstellungen für den Himmel vornehmen.

4.6.5 Post Process Volume

Mit der Post Process Volume kann man das Aussehen im Spiel nachbearbeiten. [12] Wir haben sie dazu benutzt um die Helligkeit im Spiel auf einen gewissen Bereich zu beschränken.

4.6.6 Exponential Height Fog

Mit dem Exponential Height Fog kann man Nebel über die ganze Welt erzeugen. Dieser Nebel hat unten eine höhere Dichte als weiter oben. Weiters kann man zwei Farben einstellen. Die von der Seite wo das Sonnenlicht kommt und die von der Gegenüberliegenden Seite, das heißt der Nebel hat je nach Seitenansicht eine andere Farbe. [9]

4.6.7 Lightmass Importance Volume

Mit der Lightmass Importance Volume, kann man Bereiche im Spiel einstellen, wo das Licht genau berechnet werden soll. Das ist besonders wichtig, da der Spieler nur in einem Bestimmten Bereich Licht mit guter Qualität sehen kann und somit Rechenaufwand für Bereiche in denen es nicht so ist, gespart wird. [10]

4.6.8 Light Mobility

4.6.8.1 Static

Beim Static Light wird das Licht direkt am Anfang berechnet. Das heißt man kann es während dem Spiel nicht ändern, dafür wird die Performance verbessert, da die Schatten nicht immer neu berechnet werden müssen. [13]

4.6.8.2 Stationary

Das Stationary Light kann man während dem Spielen nicht bewegen, allerdings kann man die Farbe und die Intensität verändern. Somit stellt es vom Rechenaufwand und der Funktion einen Kompromiss zwischen Static und Movable Light da. [13]

4.6.8.3 Movable

Das Movable Light kann man während dem Spielen vollständig verändern. Dadurch dass man seine Position verändern kann und die Schatten dadurch neu berechnet werden müssen, ist es das Performancelastigste Light.

4.6.9 Das Lighting im Spiel

Das Ziel beim Lighting im Spiel war es, eine düstere Stimmung zu bekommen. Dazu mussten mehrere Objekte die das Licht und den Himmel bestimmen aufeinander abgestimmt werden.

Diese Objekte sind:

1. Directional Light
2. Sky Light
3. Sky Sphere
4. Post Process Volume
5. Exponential Height Fog
6. Lightmass Importance Volume

Um die Farben abzustimmen und die gesamte Helligkeit zu regeln, muss man an drei Objekten Einstellungen vornehmen.

1. Directional Light
 - a) Licht fällt von einer Seite auf ein Objekt.
2. Sky Light
 - a) Licht fällt von allen Seiten auf ein Objekt.
3. Sky Sphere
 - a) Bestimmt die Farbe des Himmels in relation zu dem Directional Light und Sky Light.

Um das Ganze zu veranschaulichen, wurden die Werte etwas verändert und in die Lighting Only Ansicht der Unreal Engine gewechselt. In Abbildung 4.1 kann man nun sehen, dass von der Sonnenseite ein helles lilanes Licht (das Licht des Directional Lights), und von der Schattenseite ein eher dunkles, blaues Licht (das Licht des Sky Lights) kommt. Die Farbe des Himmels ist violett. Dies wurde durch die Sky Sphere bestimmt.

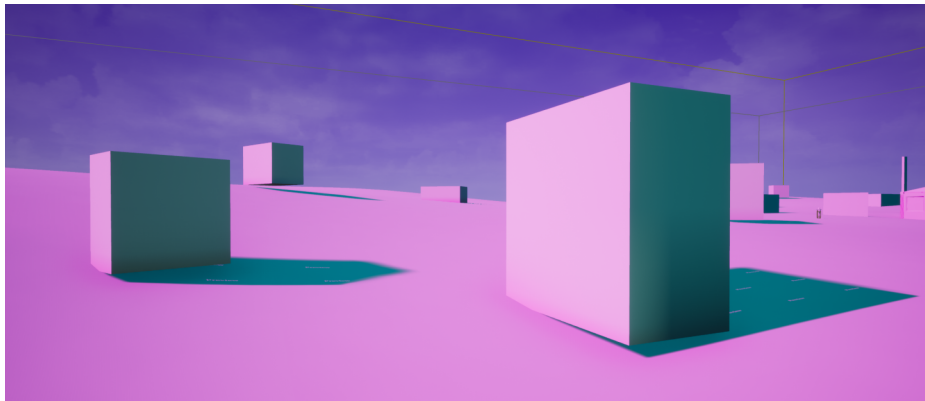


Abbildung 4.1: Objekte in der Lighting Only Ansicht von Unreal Engine

Um eine Nachtatmosphäre zu bekommen, wurde die Sonne auf der y-Achse auf 90° gesetzt. Dadurch kann man nun Sterne sehen, welche in der Sky Sphere heller oder dunkler stellen kann.

Um die Umgebungshelligkeit anzupassen, wurde in der Post Process Volume, die Helligkeit auf einen bestimmten Wert eingegrenzt.

Damit das Spiel eine düstere Stimmung bekommt, wurde noch Nebel hinzugefügt. Die effizienteste Methode war, einen Exponential Height Fog einzubauen, welcher Nebel im ganzen Spiel erzeugt. Dieser wurde relativ Dicht eingestellt und hat eine zum Himmel passende Farbe bekommen. Zusätzlich wurden in der Sky Sphere noch die Wolken und Sterne angepasst, damit man sie noch gut durch den Nebel sehen kann.

Damit die Beleuchtung nicht zu viel Rechenaufwand in anspruch nimmt, wurde noch eine Lightmass Importance Volume hinzugefügt, welche den Bereich eingrenzt, indem das Licht genau berechnet wird. Dieser Bereich liegt über der Spielwelt.

5 Sound

6 Website

6.1 Intro

Für eine Diplomarbeit an der HTL3R ist es Angabe, eine Website zu erstellen. Deswegen, und weil wir nach außen hin gut repräsentiert werden wollen, haben wir eine Website aufgebaut. Diese ist sowohl auf Deutsch als auch auf Englisch verfügbar, damit wir möglichst viele Menschen erreichen können.

6.2 CMS

Anfangs war geplant, die Website mithilfe des Frameworks Angular.js zu erstellen. Jedoch stellte sich nach Gesprächen mit Lehrpersonen heraus, dass dies zu viel Arbeit sein würde und wir unser Hauptaugenmerk auf das Spiel selbst richten sollten. Deswegen suchten wir uns ein CMS, ein Content Management System aus. Ein CMS ist ein fertiges System, welches auf einem Webserver läuft, und es dem User ermöglicht eine Website zu erstellen, ohne dafür HTML, CSS, JavaScript oder eine serverseitige Programmiersprache wie z.B. PHP oder Java kennen zu müssen.

Für uns standen konkret folgende CMS zur Auswahl:

- Wordpress
- TYPO3
- Joomla
- Drupal

Da wir uns davor noch nie mit Joomla oder Drupal auseinandergesetzt hatten, vielen dies weg. TYPO3 lernten wir in der Schule, jedoch war dieses CMS für unseren Anwendungsfall nicht geeignet, da wir nur eine einfache, statische Website brauchten. Mit TYPO3 ist der Aufwand ohne viel Inhalt bereits recht groß, jedoch steigt er nicht stark, wenn man sehr viel Inhalt hat. Bei Wordpress ist der Anfangsaufwand recht

klein, jedoch steigt er mit dem Inhalt umso stärker an. Da ich persönlich mit Wordpress schon Erfahrung hatte und

6.3 Plugins

6.4 Host

6.5 Inhalte

6.5.1 Diplomarbeit

6.5.2 Spiele

6.5.3 Sonstiges

6.6 Design

6.6.1 Template

6.6.2 Abänderungen

A Anhang 1

was auch immer: technische Dokumentationen etc.

Zusätzlich sollte es geben:

- Abkürzungsverzeichnis
- Quellenverzeichnis (hier: Bibtex im Stil plaindin)

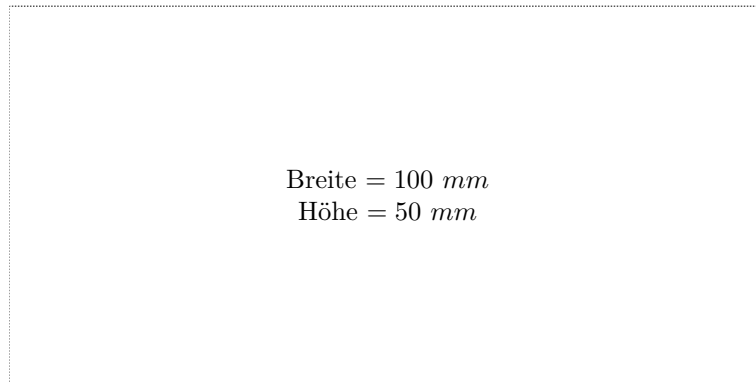
Wie geht das?

Literaturverzeichnis

- [1] BLENDER: *Array Modifier*. <https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/array.html>, Abruf: 2019-02-20
- [2] BLENDER: *Bevel Modifier*. <https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/bevel.html>, Abruf: 2019-02-20
- [3] BLENDER: *Boolean Modifier*. <https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/booleans.html>, Abruf: 2019-02-15
- [4] BLENDER: *Curve Modifier*. <https://docs.blender.org/manual/en/latest/modeling/modifiers/deform/curve.html>, Abruf: 2019-02-20
- [5] BLENDER: *Decimate Modifier*. <https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/decimate.html>, Abruf: 2019-02-20
- [6] BLENDER: *Edgesplit Modifier*. https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/edge_split.html, Abruf: 2019-02-21
- [7] BLENDER: *Mirror Modifier*. <https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/mirror.html>, Abruf: 2019-02-15
- [8] GAMES, Epic: *Directional Light*. <https://docs.unrealengine.com/en-us/Engine/Rendering/LightingAndShadows/LightTypes/Directional>, Abruf: 2019-03-04
- [9] GAMES, Epic: *Exponential Height Fog*. <https://docs.unrealengine.com/en-us/Engine/Actors/FogEffects/HeightFog>, Abruf: 2019-03-07
- [10] GAMES, Epic: *Lightmass Importance Volume*. <https://docs.unrealengine.com/en-US/Engine/Rendering/LightingAndShadows/Lightmass/Basics>, Abruf: 2019-03-07

- [11] GAMES, Epic: *Performance Guidelines for Artists and Designers*.
<https://docs.unrealengine.com/en-us/Engine/Performance/Guidelines>, Abruf:
2019-02-07
- [12] GAMES, Epic: *Post Process Volume*.
<https://docs.unrealengine.com/en-us/Engine/Rendering/PostProcessEffects>,
Abruf: 2019-03-07
- [13] GAMES, Epic: *Types of Lights*. <https://docs.unrealengine.com/en-us/Engine/Rendering/LightingAndShadows/LightTypes/Directional>, Abruf:
2019-03-04
- [14] WIKIPEDIA: *Zitat — Wikipedia, Die freie Enzyklopädie*.
<https://de.wikipedia.org/wiki/3D-Grafiksoftware>, Abruf: 2019-01-09

— Druckgröße kontrollieren! —



— Diese Seite nach dem Druck entfernen! —

Diese
Seite
nach dem
Druck
entfer-
nen!