Höhere Technische Bundeslehranstalt Wien 3, Rennweg
IT & Mechatronik

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

Autor richtig setzen

Metatags setzen

Diplomarbeit

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

Layout überprüfen, d.h. sind Bilder, Texte, Überschriften an der richtigen Stelle?

unter der Anleitung von

Matejowsky Peter
Dazinger Robert, Sturm Gerhard

Wien, 21. 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 21. März 2019

Hillinger Stefan

Lichtenstein Benjamin

Röhrer Tobias

Seidel Hans

Inhaltsverzeichnis

Tabellenverzeichnis

kann entfallen falls (fast) leer

Abbildungsverzeichnis

1 Ziele

Ziele hineinschreiben

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

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
2. First level item
    a) Second level item
    b) Second 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‘
- "Doubleqoutes"

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

Fußnoten[1] mit LaTeX sind kein Problem[2] Dank des Befehls \footnote.

Zitate:

> Die quote-Umgebung ist nicht nur für Zitate eine beliebte Form der Texthervorhebung, 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 umbedingt 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

So gibt man Tastenkombinationen an: ⇧+F.

So gibt man Schritte an: Something » new.

3 Blender

3.1 Einleitung

Einleitung bei jedem Kapitel?

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 Raum werden Punkte gesetzt, die auch zu Kanten und Flächen verbunden werden können. Mehrere solcher Punkte, Kanten und Flächen ergeben 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 in einem Bild verwenden.
- Objekte erschaffen und für 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.[17] Populäre Programme dieser Art sind u. a.:

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

Unser Team hat beschlossen die benötigten Modelle in Blender zu erstellen, weil wir mit dem Programm im Unterricht arbeiten und es kostenlos verwendbar ist. Unser Team hat mit der Blender Version 2.79 gearbeitet.

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 Origin[1] des Objekts verändern. Das verändern des Origins ist besonders nützlich, um Objekte auf einem bestimmten Punkt zu verändern. Wie wir im *Kapitel 3.13 " Exportieren von Blender zu Unreal Engine 4"* noch erfahren, hat uns dieses Feature besonders geholfen.

Referenz kursiv

3.3.1 Smooth Shading

Mit Smooth Shading werden die Schatten eines Objektes so berechnet, dass das ganze Objekt Glatt aussieht (*Abbildung 3.7, rechts*). Man kann diesen Effekt, sehr gut für Runde Objekte benutzen, da man dank dieser Funktion, nicht so viele Flächen für ein Objekt benötigt, um es Rund aussehen zu lassen. Man kann also sehr gut die Performance des Spiels verbessern, wenn man anstatt einer hohen Anzahl an Flächen Smooth Shading verwendet.

3.3.2 Flat Shading

Flat Shading lässt Objekte Kantig aussehen (*Abbildung 3.7, links*). Es ist standardmäßig für Objekte ausgewählt und ist für alle Objekte die keine abgerundeten Flächen besitzen gut.

[1] Der Origin ist ein Punkt, der bei jedem Objekt festgelegt ist. Er bestimmt an welchem Ort im Koordinatensystem das Objekt platziert wird. Außerdem rotiert und skaliert das Objekt zu diesem Punkt

3.3.3 Ebenen

Sehr nützlich im Object Mode sind Ebenen. Man kann dadurch mehrere Objekte ein- bzw. ausblenden. Das war Beispielsweise hilfreich, um das Haus zu modellieren und zusammenzufügen (*Kapitel 3.12.1 "Haus"*). Auf diese Art können nämlich Objekte je nach Stockwerk eingeblendet werden.

Neben Ebenen gibt es noch die Möglichkeit, ein Objekt im Local View und im Global View anzuzeigen. Die Standardansicht eines Objektes ist im Global View. Man sieht das ausgewählte Objekt und alle anderen Objekte in derselben Ebene, die nicht ausgeblendet sind. Im Local View kann man alle ausgewählten Objekte seperat von allen anderen Objekten anschauen.

3.4 Edit Mode

Im Edit Mode kann man 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 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 *Kapitel 3.11 "Modellierung von 3D-Objekten"* anhand von Praxisbeispielen gezeigt.

3.5 Modifikatoren

Modikfikatoren verändern Objekte in einer gewissen Art und Weiße. Die Veränderung ist je nach Modifikator unterschiedlich.

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

3.5.1 Boolean-Modifikator[3]

Der Boolean-Modifikator verändert ein Objekt mittels einem zweiten 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, mittig*).

Bei Intersect wird das Objekt mit dem Modifikator an den überschneidenden Stellen in das Zielobjekt eingefügt (*Abbildung 3.1, rechts*).

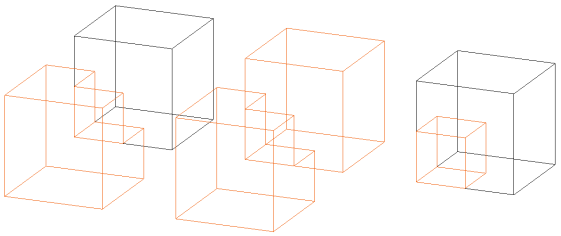


Abbildung 3.1: Difference-, Union- und Intersect-Operation

3.5.2 Mirror-Modifikator[7]

Der Mirror-Modifikator spiegelt ein Objekt um den Origin. Den Origin kann man festlegen indem man im Edit Mode einen Punkt im Koordinatensystem auswählt, dann die Position des Cursors setzt und anschließend im Object Mode den Origin setzt.

passt die Beschreibung vom Origin

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

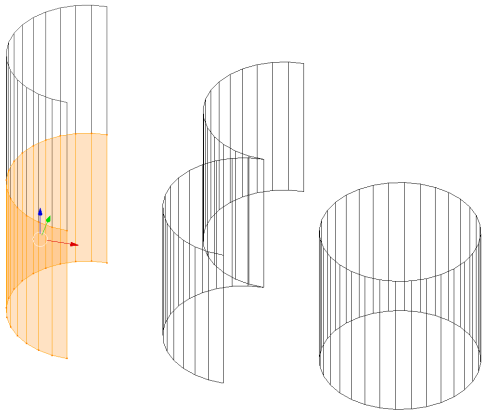


Abbildung 3.2: Mirror-Modifikator mit Spiegelung um die Z-, Y- und X-Achse

3.5.3 Array-Modifikator[1]

Der Array-Modifikator vervielfacht ein Objekt entlang einer Achse. Man kann 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.

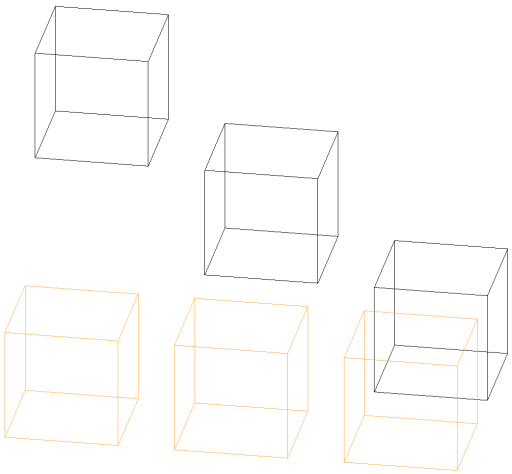


Abbildung 3.3: Array-Modifikator

3.5.4 Curve-Modifikator[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 Bezier-

kurve, in der unteren beide zusammengefügt. Außerdem zu erwähnen ist, dass das Rechteck mehrfach unterteilt ist damit es sich der Bezierkurve anpassen kann.

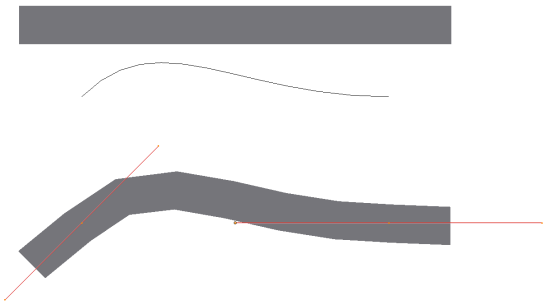


Abbildung 3.4: Curve-Modifikator

3.5.5 Bevel-Modifikator[2]

Mit dem Bevel-Modifikator 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. Der Bevel Modifikator wurde oft verwendet, damit die erstellten Objekte echt aussehen, weil jedes Objekt z.B. eine Tischkante nie abrupt aufhört, sondern zumindest ein bisschen abgerundet ist.

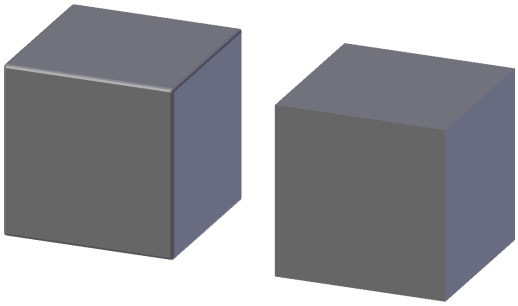


Abbildung 3.5: Bevel-Modifikator

3.5.6 Decimate-Modifikator[5]

Mit dem Decimate-Modifikator kann man bei einem Objekt die Flächen reduzieren. Zum Einsatz kommt er bei komplizierten Objekten, die viele Flächen brauchen, damit sie detailliert genug dargestellt werden können. Man entfernt dann aber möglichst viele Flächen wieder, so dass das Objekt noch gut genug aussieht. Das entfernen der überflüssigen Flächen verbessert die Performance sehr stark. In *Abbildung 3.6* sieht man den ursprünglichen Würfel (orange) und den Würfel mit dem angewendeten Decimate Modifikator.

Vergleich - Detailed Mesh für Texturen.

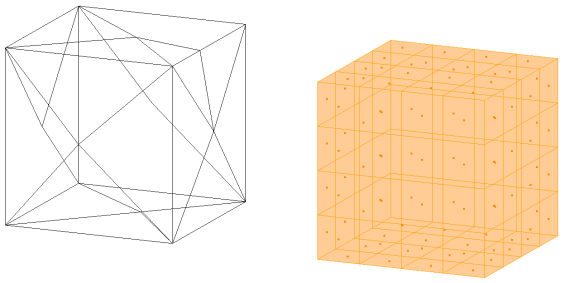


Abbildung 3.6: Decimate-Modifikator

3.5.7 Edge-Split-Modifikator[6]

Mit dem Edge Split Modifikator kann man 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 dem geringsten Winkel zu den Flächen angewendet und rechts wird Smooth Shading überall angewendet.

Mehr zu Smooth Shading findet man in *Kapitel 3.3.1 "Smooth Shading"*.

3.6 Viewport Shading

Mit Viewport Shading, kann man in Blender bestimmen, wie Objekte angezeigt werden. Für uns waren die Varianten Solid und Wireframe relevant.

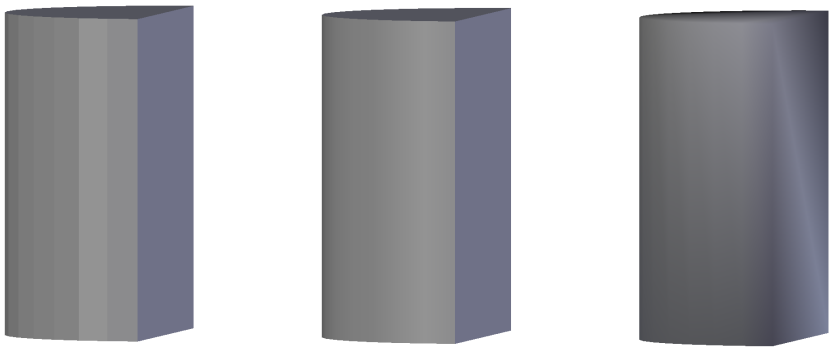


Abbildung 3.7: Edge-Split-Modifikator

3.6.1 Solid

Im Solid Mode wird das Objekt als mit der Standardfarbe grau angezeigt. Außerdem werden die verschiedenen Materialien, die ein Objekt hat in den jeweiligen Farben angezeigt. Mit dem Solid Mode kann man einen guten Eindruck von der Form des fertigen Objekts bekommen.

3.6.2 Wireframe

Im Wireframe Modus werden Objekte so dargestellt, dass man durch sie hindurch sehen kann. Das ist hilfreich, wenn man wissen muss wie Kanten, Punkte oder Flächen auf gegenüberliegenden Seiten des Objekts platziert sind. Außerdem kann man durch das Objekt hindurch mehrere Punkte, Kanten oder Flächen auf einmal auswählen. So kann man zum Beispiel in einer orthographischen Ansicht, alle Kanten am Boden eines Objektes auswählen ohne es drehen zu müssen. Diese Kanten kann man in Folge parallel verschieben.

Referenz Wirefra-me, bild?

Am An-fang von Kapitel Blender

3.7 Texturierung Vorarbeiten

Um Texturen in der Unreal Engine richtig anwenden zu können, müssen zuerst in Blender ein paar Vorbereitungen getroffen werden.

Um mehrere Texturen auf Objekte anwenden zu können, muss man einem Objekt mehrere Materialien zuweisen. Dazu wählt man im Edit Mode die Flächen aus die man einem Material geben möchte und fügt diese dem gewünschten Material hinzu.

Bevor man eine UV-Map erstellt, muss man sicher gehen, dass beim Modellieren keine Punkte im 3D-Modell überlappen. Dazu wechselt man in den Edit Mode und wählt das ganze Objekt aus und entfernt anschließend alle doppelten Punkte im Objekt.

Jetzt kann man die UV-Map erstellen. Sie gibt an, an welchem Ort eine Textur auf einem Objekt erscheint. Im UV/Image Editor Panel kann man, wenn man im 3D-View Panel in den Edit Mode wechselt und das ganze Objekt auswählt, die UV-Map sehen.

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

3.10.4 Animation in Blender

3.10.5 Umsetzung der Animationen

Ein großer Teil in der Spielentwicklung, war das Modellieren. In diesem Kapitel wird gezeigt, wie man kompliziertere Modelle macht und auf was man achten muss.

Es ist sehr wichtig, dass das Spiel mit einer guten Performance spielbar ist. Deswegen wurde beim Modellieren darauf geachtet, dass es pro Modell so wenig Flächen wie möglich gibt [12], denn diese müssen vom Computer berechnet werden. Darunter kann dann die Performance im Spiel leiden. Zwei wesentliche Gebäude des Spieles, sind das Haus des Grabwächters und das Mausoleum inklusive aller enthaltenen Objekte. Weiters gibt es einige Objekte die für die Story relevant sind und Objekte, die modelliert wurden um das Spiel zu füllen.

Bei Mausoleum und Cutscene wurde das auch gemacht adden.

Modellierte Objekte referenzieren

Auflistung der Objekte:

Zweck	Objekt	Entwickelt von	Anmerkung
Haus			*Kapitel 3.12.1 "Haus"*
	Stufen	Lichtenstein	*Kapitel 3.11.3 "Stufen im Haus"*
	Haustür	Lichtenstein	
	Tür	Lichtenstein	
	Fenster	Lichtenstein	
	Sessel	Lichtenstein	
	Toilette	Lichtenstein	
	Waschbecken	Lichtenstein	
	Dusche	Lichtenstein	
	Bett	Lichtenstein	
	Bettdecke	Lichtenstein	*Kapitel 3.11.4 "Bettdecke"*
	Polster	Lichtenstein	
	Nachtkasten	Lichtenstein	
	Nachtkastenlampe	Lichtenstein	
	Kleiderkasten	Lichtenstein	
	Steckdose	Lichtenstein	
	Küche Arbeitsfläche	Lichtenstein	
	Küche Schubladen	Lichtenstein	
	Küche Herd	Lichtenstein	
	Küche Backrohr	Lichtenstein	
	Deckenleuchte	Lichtenstein	
Mausoleum			
	Stufen	Lichtenstein	
	Säule	Lichtenstein	
	Verzierung	Lichtenstein	
	Tafel 1	Lichtenstein	
	Tafel 2	Lichtenstein	
	Altar	Lichtenstein	
	Hauptgrab	Lichtenstein	
	Hauptgrab klein	Lichtenstein	
	Nebengrab	Lichtenstein	
	Wandgrabstein	Lichtenstein	
Hauptobjekte			
Füllobjekte			
	Nebengrab 1	Lichtenstein	
	Nebengrab 2	Lichtenstein	
	Nebengrab 3	Lichtenstein	
	Nebengrab 4	Lichtenstein	
	Paracelsus Grab	Lichtenstein	*Kapitel 3.11.2 "Paracelsus Grab"*
	Fass	Lichtenstein	
	Türstopper	Lichtenstein	

3.11.1 Hauptgrab

Das Hauptgrab, ist einem echten Grab, welches unser Team auf einem Friedhof gesehen hat, nachempfunden. Der Grabsockel besteht aus einem angepassten Würfel mit einem Bevel Modifikator. Der Grabdeckel wurde ebenfalls aus einem Würfel modelliert und anschließend mit einem Bevel-Modifikator so bearbeitet, dass er seine Einkerbungen bekommen hat. Der Griff ist, ein mit einem Boolean-Modifikator zusammengefügter Würfel und Torus[2]. Der Griff wurde mittel Array-Modifikator vervielfältigt und im passenden Winkel am Grabdeckel platziert. Anschließend wurden beide Objekte zusammengefügt.

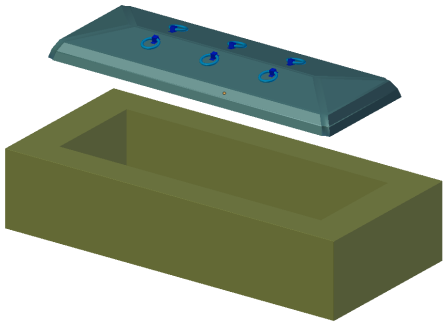


Abbildung 3.8: Hauptgrab

Verwendete Modifikatoren: *Kapitel 3.5.5 "Bevel"*; *Kapitel 3.5.1 "Boolean"*; *Kapitel 3.5.3 "Array"*

3.11.2 Paracelsus Grab

Das Grab des Paracelsus ist ein Modell, welches einem echten Objekt nachempfunden ist. Um ein möglichst originalgetreues Ergebnis zu erhalten, modelliert man das Objekt von einem Foto nach. Dafür importiert man zuerst ein Foto in Blender (*Abbildung 3.9*). Man kann die Durchsichtikgeit des Bildes einstellen, bei welchen Winkel und auf welcher Achse das Bild sichtbar ist. In diesem Fall wurde das Bild auf 100% Sichtbarkeit eingestellt, damit man während dem Modellieren die Linien im Bild noch gut erkennt. Außerdem, wurde es nur auf einer orthographischen Ansicht angezeigt, damit man es nur in der Ansicht sieht, in der es sinnvoll ist, die Maße nachzumodellieren.

[2] Der Torus ist ein Objekt in Blender. Seine Gestalt ähnelt der eines Rings.

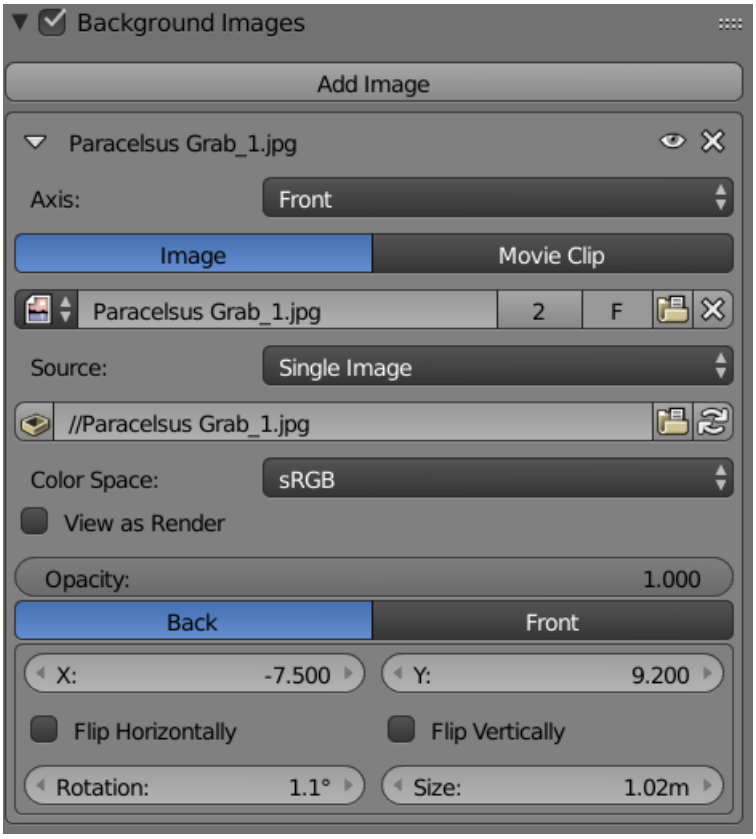


Abbildung 3.9: Import von Bildern

Nun modelliert man mit Formen das Grab nach und verschiebt im Edit Mode einzelne Punkte, so dass sie die gewünschte Form darstellen. Man muss allerdings darauf achten, dass in diesem Bild (*Abbildung 3.10*) Verzerrungen aufgrund des Aufnahmewinkels des Fotos auftreten, deshalb sind einige Stellen nicht genau, sondern durch eine Schätzung der Größen nachmodelliert worden.

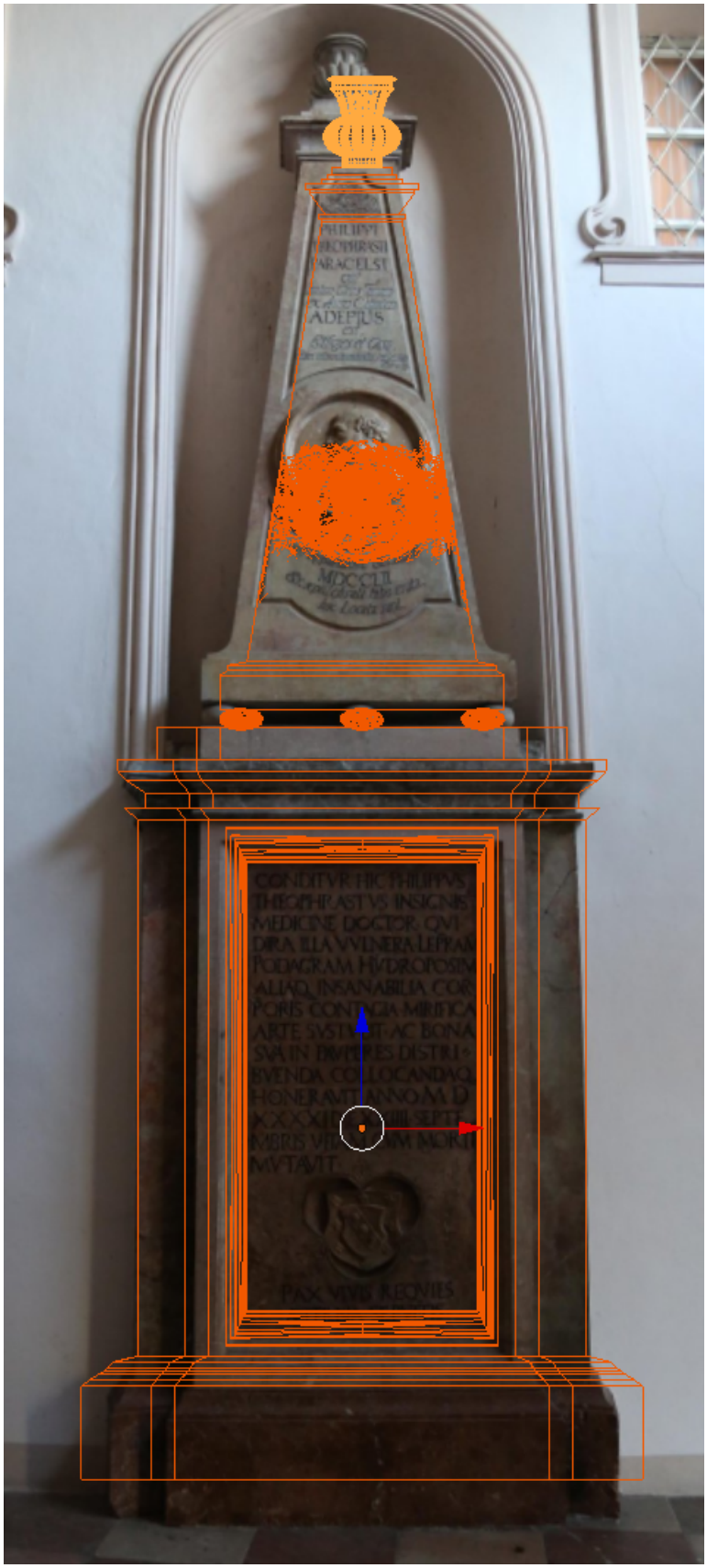


Abbildung 3.10: Paracelsus Grab Nachmodellierung (Lizenz zum Bild: [8], Autor: unbekannt - Wikimedia)

Muss man Nutzungsrechte für das Bild einfügen?

3.11.2.1 Vase

Um die Vase für das Paracelsus Grab zu modellieren, wurde eine Sphere benutzt und anschließend im Edit Mode verändert. Anschließend wurden die Verzierungen für die Vase gemacht. Dazu wurde zwei Curves erstellt (*Abbildung 3.11*). Die obere für die Einkerbungen und die untere für die Verzierung der Vase. Damit die Curves eine Dicke haben, muss man sie mit einem Bezier-Circle verbinden, der die Dicke der Bezier-Curve bestimmt. Nach dem Erstellen der Curves, wurden diese zu einem Mesh konvertiert, damit sie anschließend mittels Boolean-Modifikator mit der Vase verbunden bzw. ausgeschnitten werden können.

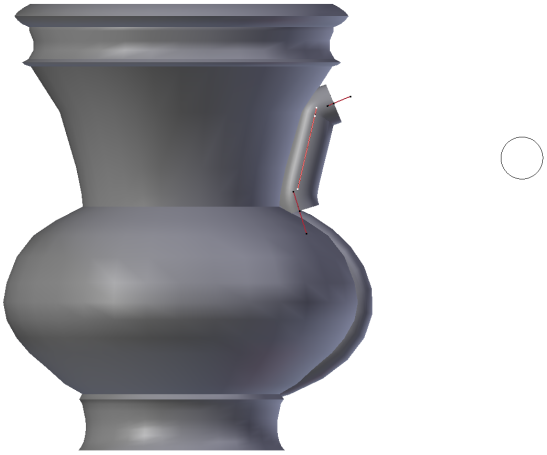


Abbildung 3.11: Links: Vase mit den Verzierungen. Rechts: Becier-Circle

Verwendete Modifikatoren: *Kapitel 3.5.4 "Curve"*; *Kapitel 3.5.1 "Boolean"*

3.11.2.2 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 unterteilt. Dann wurde im Sculpt Mode eine Schwarzweiß Textur, welche aus dem ursprünglichen Bild des Paracelsus Grabes gefertigt wurde, eingefügt (*Abbildung 3.12, rotes Rechteck*). Somit kann man mit dem Brush, Höhen und Tiefen Zeichnen. Außerdem wurden die Werte Strenght und Radius angepasst, damit die Brushstärke so hoch ist, das bei der Bearbeitung die Flächen weit genug verschoben

Referenz bei Hans

werden und die Brushfläche groß genug ist, damit das Gesicht auf das Grab hinauf passt. Danach wurde der Brush angewendet. Letztendlich sind noch mit dem Decimate Modifikator so viele Flächen wie möglich entfernt worden, ohne dass das Aussehen der Statue stark beeinträchtigt wurde.

Proofreading

Verwendete Modifikatoren: *Kapitel 3.5.6 "Decimate"*

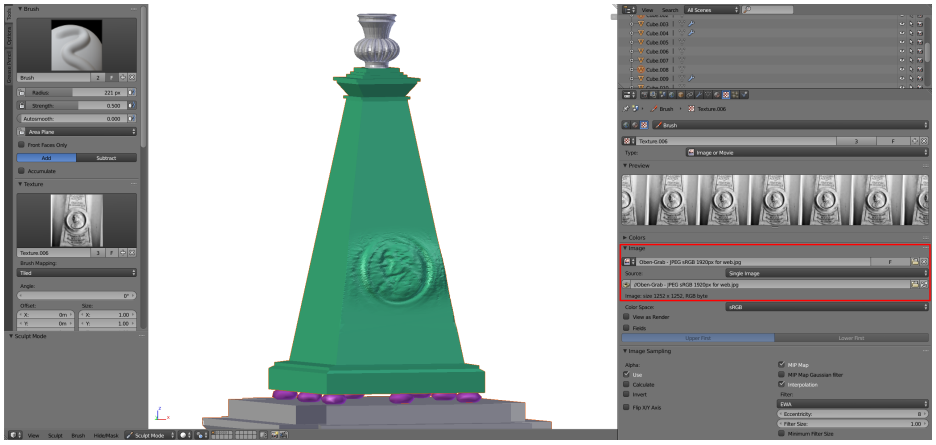


Abbildung 3.12: Einfügen der Schwarz/Weiß Textur (Lizenz zum ursprünglichen Bild der Schwarz/Weiß Textur: [8])

3.11.2.3 Verzierungen

Nachdem die Verzierung des Paracelsus Grabes (*Abbildung 3.13*) 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 zusammengefügt. Die Verzierung selber, wurde mit einer Fläche modelliert, die dann im Edit Mode so angepasst wurde, dass sie aussieht wie eine Verzierung.

Abbildung 3.13: Verzierung des Paracelsus Grabs

Verwendete Modifikatoren: *Kapitel 3.5.1 "Boolean"*

3.11.3 Stufen im Haus

Um die Stufen (*Abbildung 3.14*) zu modellieren, wurden echte Stufen abgemessen, um die Stufen mit einer möglichst realistischen Höhe und Tiefe modellieren zu können. Die Stufen wurden mit einem Würfel modelliert, der mit einem Array Modifikator wiederholt wurde. Das Geländer und die Halterung des Handlaufes zur Wand wurden mit den gleichen Verfahren wie die Stufen gefertigt, nur dass als Basisobjekt kein Würfel sondern ein Zylinder genommen wurde. Die Stufen sind ein relativ einfaches Modell, jedoch ist es wichtig zu beachten, dass die Abstände sowohl im Modell selbst, als auch innerhalb des Hauses realistisch sind.

Um das Gelände von der Höhe an das Haus anzupassen, wurde es zum Schluss mit einem Boolean-Modifikator abgeschnitten. Der Handlauf wurde ebenfalls mit einem Boolean Modifikator gekürzt. Das gesamte Geländer wurde mit Smooth Shading (*Kapitel 3.3.1 "Smooth Shading"*) versehen, damit es runder aussieht.

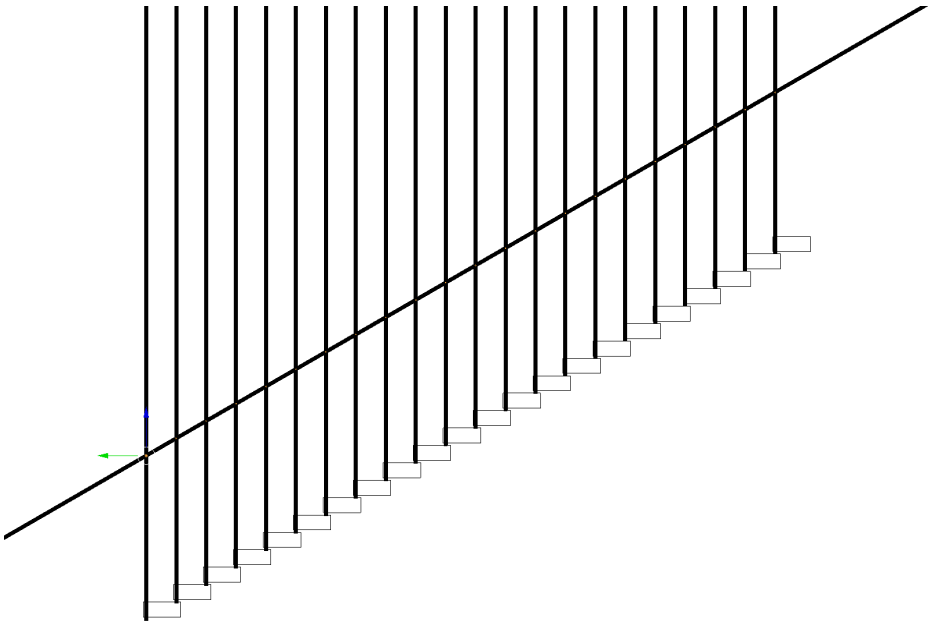


Abbildung 3.14: Stufen mit Geländer

3.11.4 Bettdecke

Die Bettdecke sollte etwas zerknüllt aussehen. Deshalb wurde sie mit einer Simulation modelliert. Damit sich die Decke verformt, muss sie genug Flächen haben. Dies kann man erreichen, indem man eine Plane[3] erstellt, in den Edit Mode wechselt und diese in mehrere Flächen unterteilt. Anschließend wurde auf die Decke ein Solidify-Modifikator angewendet, damit diese auch eine Dicke hat. Dann wurden Forcefields hinzugefügt (*Abbildung 3.15*) um die Decke in Bewegung zu versetzen. Damit sich die Decke aber auch wirklich bewegt muss man sie auswählen, in den Physics Tab wechseln und die Option Cloth auswählen, damit sich die Decke wie ein Stoffstück verhält. Wenn man nun auf der Timeline auf Play drückt, bewegt und verformt sich die Bettdecke. Wenn die Form passend ist pausiert man die Simulation. Dadurch, das die Decke animiert wurde, ist sie kein Objekt mehr. Deshalb muss man sie am Schluss noch in ein Objekt umformen, damit man die Decke frei bewegen kann, ohne das sie in die Form vor der Simulation zurückspringt.

Proofreading

[3] Eine Plane ist ein Objekt in Blender, welches einer Fläche gleicht.

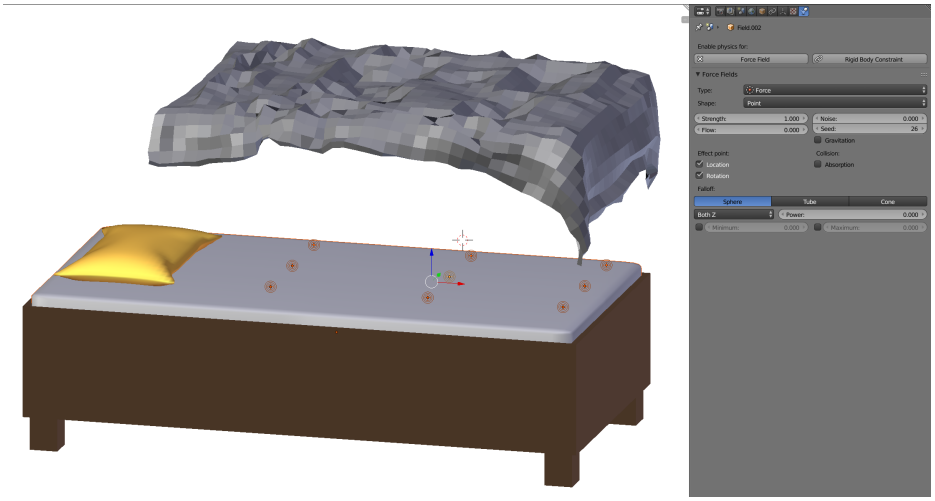


Abbildung 3.15: Bettdecke mit Forcefields(orange)

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. Um die Größe besser zu planen, wurde eine Skizze mit Raumplan für das Haus angefertigt und mit realen Hausmaßen abgeschätzt, ob alle Objekte in den jeweiligen Raum passen. Außerdem müssen alle Maße im Haus z.B. Türen, Gänge und Stockwerkhöhe im Haus ungefähr an die Größe der Charaktere angepasst sein. Die Objekte im Haus sind angelehnt an echte Objekte. Zum Beispiel sind die Stufen (*Kapitel 3.11.3 "Stufen im Haus"*) abgemessen 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 3.16* 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 zwischen der Wand und den Objekten noch auf einen realistischen Abstand angepasst.

Nachdem alle Objekte für das Haus erstellt worden sind, wurden diese einzeln exportiert und in einer neuen Blender Datei wieder zusammengefügt. Genauere Informationen zum Exportieren, kann man unter dem *Kapitel 3.13 "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 für einen anderen Zweck noch einmal benötigt.

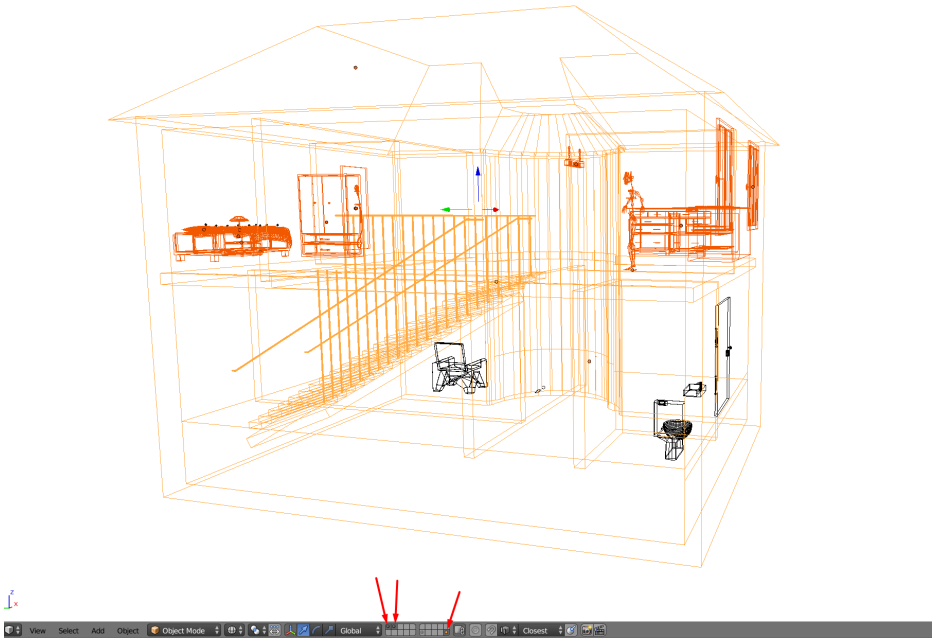


Abbildung 3.16: Objekte des Hauses, auf mehreren Ebenen verteilt

section einfügen

3.12.2 Cutscene

Die Cutscene ist eine Sequenz, die zu einem bestimmten Zeitpunkt im Spiel abgespielt wird. Im Spiel gibt es zwei Cutscenes, eine Cutscene am Anfang und eine am Ende. Folgend wird die Cutscene am Schluss vom Spiel beschrieben, weil sie aufwändiger zu erstellen war.

Bei der Schlusssequenz legt sich der Hauptcharakter in sein Grab hinein und begräbt sich anschließend selbst, indem die Mischmaschine umgedreht wird und der Zement in das Grab fließt. Der Zement wurde mit einer Simulation gemacht. Damit der Zementfluss auch real aussieht, musste man die Keyframes[4] und Objekte so setzen, dass immer weniger Zement aus der Mischmaschine hinaus kommt (*Abbildung 3.17*). Damit der Zement auch hinausfließen kann, wurde die Mischmaschine bewegt.

Die ganze Scene wurde anschließend in die Unreal Engine importiert. Exportiert wurde die Cutscene wie eine Simulation (*Kapitel 3.13.2 "Simulationen"*). Mit Blueprints wurde dann die Funktion hinzugefügt, dass sich der Spieler zu einem gewissen Zeitpunkt

[4] Keyframes sind Werte von Objekten, die zu einem Bestimmten Zeitpunkt gespeichert werden, damit eine Animation auf Basis dieser Werte berechnet werden kann.

Richtung Grab bewegt und anschließend selbst begräbt. Während den Bewegungen, ist der Input für Bewegungen gesperrt. Am Schluss wird der Bildschirm schwarz. Dadurch wird das Ende des Spiels symbolisiert.

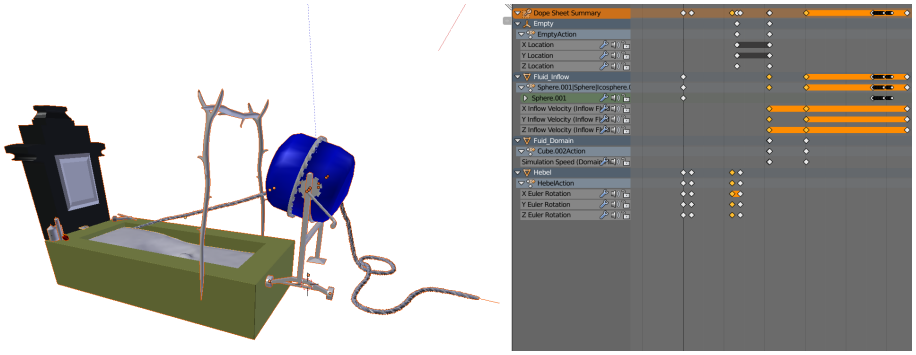


Abbildung 3.17: links: Cutscene Modellierung, rechts: Keyframes

3.13 Exportieren von Blender zu Unreal Engine 4

Damit alle Objekte die in Blender modelliert wurden, richtig in der Unreal Engine anzeigen zu können, muss man ein paar Einstellungen an den Objekten machen und die Maße in der Datei verändern.

In der Datei selbst, muss man im Property-Panel die Length auf Metric, den Angle auf Degree und die Unit Scale auf 0.01 stellen. Das ist notwendig, damit die Objekte in der Unreal Engine die richtige Skalierung haben. Damit man diese Einstellungen nicht bei jeder Datei neu einstellen muss, kann man die Einstellungen im Info-Panel unter File » Save Startup File speichern.

Bei allen Objekten, die man exportieren möchte, muss man darauf achten, dass das Objekt auf den Location Koordinaten den Wert 0 hat, damit das Objekt in der Unreal Engine seinen Origin in der Mitte des Objektes hat und um diesen gedreht, verschoben und skaliert werden kann. Um das Objekt entsprechend auf 0 zu verschieben wird der Origin des Objektes mittig im Objekt gesetzt. Dazu drückt man ⇧ + Ctrl + Alt + C und wählt Origin to Center of Mass (Volume) aus. Damit wurde der Origin des Objektes neu gesetzt. Das ist sehr nützlich, falls der Origin wo anders ist und man das Objekt nicht um seine eigene Achse drehen, verschieben und skalieren kann. Anschließend setzt man mit ALT + G die Location Koordinaten auf 0.

Zum Schluss ist es noch wichtig welches Dateiformat benutzt wird, um die Daten so zu speichern, dass beide Programme die enthaltene Information lesen können. Die Wahl des richtigen Formates, wird in den Unterkapiteln abgedeckt.

3.13.1 3D-Modelle

Um 3D-Modelle zu exportieren, muss man sie mit Dateien der Endung .fbx exportieren. Damit man nicht etwas falsches exportiert, ist es empfehlenswert nur ausgewählte Objekte zu exportieren. Dazu wählt man dann alle Objekte aus die man exportieren möchte, wählt im Info-Panel File > Export > FBX (.fbx) aus und wählt dann in den Exporteinstellungen unter Main den Punkt Selected Objects aus. Um Smooth Shading (*Kapitel 3.3.1 "Smooth Shading"*) auch zu exportieren, muss man unter Geometrie im Punkt Smoothing den Wert Face angeben. Jetzt kann man die Datei mit Export FBX exportieren.

3.13.2 Simulationen

Bei Simulationen, muss man die Dateien als Alembic Datei exportieren. Dazu wählt man wieder alle Objekte aus, die man exportieren möchte und wählt diesmal beim Exportieren das Dateiformat Alembic (.abc) aus. Bei den Exporteinstellungen kann man vor dem Exportieren den Punkt Triangulate auswählen. Es kann nämlich sein, dass die Dateien in der Unreal Engine sonst nicht erkannt werden.

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

4.4.7 Umsetzung der Texturierung in der Unreal-Engine

4.6.1 Vorwissen

Eine uns bekannte Beleuchtungsmethode ist die 3-Punkt-Beleuchtung. Bei ihr wird klassischer weiße 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.

Weiters ist sind uns verschiedene Lichtarten bereits aus Blender bekannt, die es in Unreal Engine auch gibt.

Dazu zählen:

1. Point Light
2. Sun (entspricht dem Directional Light in Unreal Engine)
3. Spot Light
4. Hemi (entspricht dem Skylight in Unreal Engine)

Außerdem gibt es 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 kann. [9]

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. [13] 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. [10]

4.6.7 Lightmass Importance Volume

Mit der Lightmass Importance Volume, kann man Bereiche im Spiel einstellen, wo das Licht genau berechnet werden soll. In Bereichen in denen das Licht genauer berechnet wird, prallt das Licht öfters von Oberflächen ab und wird pro abprallen (Bounce) schwächer und erzeugt somit genauere Schatten. 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. [11]

4.6.8 Point Light

Das Point Light strahlt in alle Richtungen für eine begrenzte Reichweite. Daher eignet es sich gut als Lampe.

4.6.9 Spot Light

Das Spotlight leuchtet in eine Richtung für eine begrenzte Reichweite. Es eignet sich ebenfalls gut als Lampe.

4.6.10 Light Mobility

4.6.10.1 Static

Beim Static Light wird das Licht direkt am Anfang berechnet und in einer Lightmap gespeichert. Das heißt man kann es während dem Spiel nicht ändern. Dafür wird die Performance verbessert, da die Schatten nicht immer neu berechnet werden müssen. [14]

4.6.10.2 Stationary

Das Stationary Light kann man während dem Spielen nicht bewegen, allerdings kann man die Farbe und die Intensität verändern. Es kann aber nur das direkte Licht, also kein abprallendes Licht (Licht prallt normalerweise öfters ab und wird immer schwächer) verändert werden. Somit stellt es vom Rechenaufwand und der Funktion einen Kompromiss zwischen Static und Movable Light da. [15]

4.6.10.3 Movable

Das Movable Light wird während dem Spiel berechnet. Dadurch kann man es während dem Spiel bewegen und vollständig ändern. Es ist das Performance lastigste Light. [16]

4.6.11 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
7. Point Light
8. Spot Light

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

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

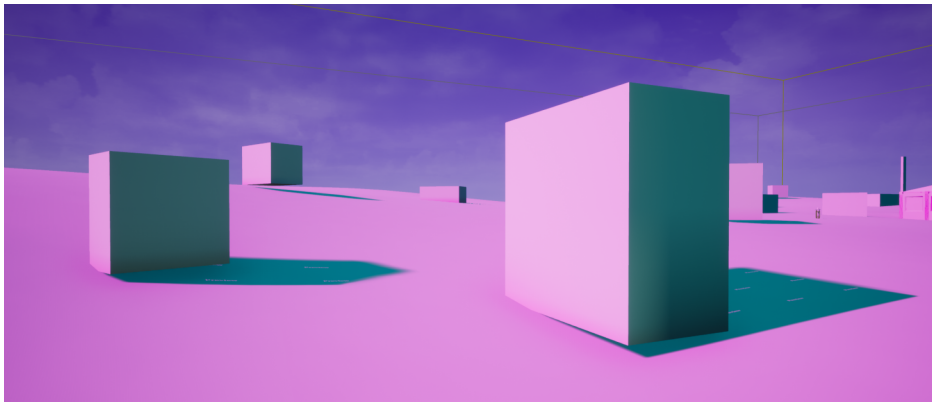


Abbildung 4.1: Objekte in der Lighting Only Ansicht von Unreal Engine

Um eine Nachtatmossphere zu bekommen, wurde die Sonne auf der y-Achse auf 90°

gesetzt. Dadurch kann man nun Sterne sehen, welche man in der Sky Sphere heller oder dunkler stellen kann.

Um ein 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.

Um eine passende Atmosphäre in den Gebäuden zu schaffen wurden je nach Lampenart, Spot Lights oder Point Lights verwendet.

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 können 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 Modifikator*. https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/array.html, Abruf: 2019-02-20

[2] Blender: *Bevel Modifikator*. https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/bevel.html, Abruf: 2019-02-20

[3] Blender: *Boolean Modifikator*. https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/booleans.html, Abruf: 2019-02-15

[4] Blender: *Curve Modifikator*. https://docs.blender.org/manual/en/latest/modeling/modifiers/deform/curve.html, Abruf: 2019-02-20

[5] Blender: *Decimate Modifikator*. https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/decimate.html, Abruf: 2019-02-20

[6] Blender: *Edgesplit Modifikator*. https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/edge_split.html, Abruf: 2019-02-21

[7] Blender: *Mirror Modifikator*. https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/mirror.html, Abruf: 2019-02-15

[8] Commons, Wikimedia: *Paracelsus Grab*. https://commons.wikimedia.org/wiki/File:Salzburg_Grab_Paracelsus_kl.jpg, Abruf: 2019-03-16

[9] Games, Epic: *Directional Light*. https://docs.unrealengine.com/en-us/Engine/Rendering/LightingAndShadows/LightTypes/Directional, Abruf: 2019-03-04

[10] Games, Epic: *Exponential Height Fog*. https://docs.unrealengine.com/en-us/Engine/Actors/FogEffects/HeightFog, Abruf: 2019-03-07

[11] Games, Epic: *Lightmass Importance Volume*. https://docs.unrealengine.com/en-US/Engine/Rendering/LightingAndShadows/Lightmass/Basics, Abruf: 2019-03-07

[12] Games, Epic: *Performance Guidelines for Artists and Designers*. https://docs.unrealengine.com/en-us/Engine/Performance/Guidelines, Abruf: 2019-02-07

[13] Games, Epic: *Post Process Volume*. https://docs.unrealengine.com/en-us/Engine/Rendering/PostProcessEffects, Abruf: 2019-03-07

[14] Games, Epic: *Types of Lights*. https://docs.unrealengine.com/en-us/Engine/Rendering/LightingAndShadows/LightMobility/StaticLights, Abruf: 2019-03-16

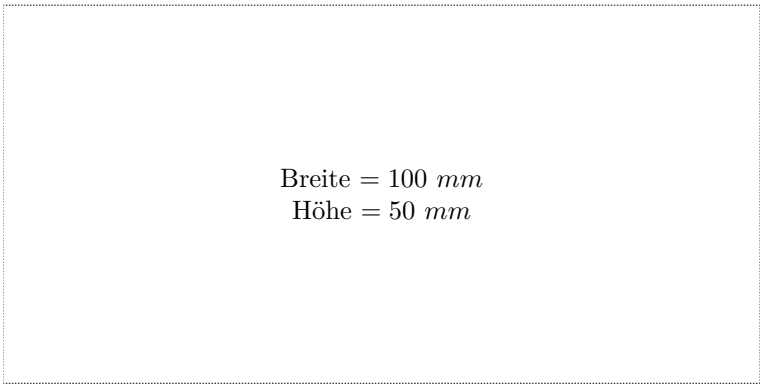
[15] Games, Epic: *Types of Lights*. https://docs.unrealengine.com/en-us/Engine/Rendering/LightingAndShadows/LightMobility/StationaryLights, Abruf: 2019-03-16

[16] Games, Epic: *Types of Lights*. https://docs.unrealengine.com/en-us/Engine/Rendering/LightingAndShadows/LightMobility/DynamicLights, Abruf: 2019-03-16

[17] Wikipedia: *Zitat — Wikipedia, Die freie Enzyklopädie*. https://de.wikipedia.org/wiki/3D-Grafiksoftware, Abruf: 2019-01-09

Nummerierung im Text sortieren

— Druckgröße kontrollieren! —



— Diese Seite nach dem Druck entfernen! —

Diese Seite nach dem Druck entfernen!