

## Projektbericht 2002 - 2004

Impressum

artecLab paper 6

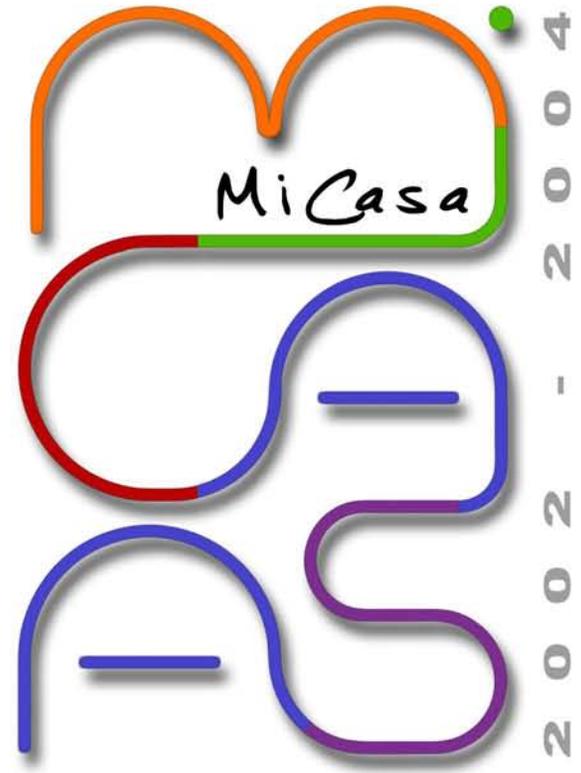
**micasa — mixed reality cave**  
**Projektbericht 2002 – 2004**

Laboratory for Art, Work and Technology  
Universität Bremen  
Enrique-Schmidt-Straße 7 (SFG)  
D-28359 Bremen  
[www.arteclab.uni-bremen.de](http://www.arteclab.uni-bremen.de)

Bremen 2005

Redaktion: Bernd Robben

ISSN 1860-9953  
Copyright © artecLab-paper, Bremen  
Satz und Herstellung im Eigenverlag



# MiCasa

Entwicklung eines Mixed Reality Cave

von

Alexander Alves, Daniel Cermak-Sassenrath, Oliver Dambroth, Sebastian Fischer, Thorsten Freese, Anja Hashagen, Serpil Karatay, Sebastian Kinder, Markus Krause, Sabine Kroh, Daniel Kühn, Tim Pilz, Daniel Pratsch, Thorsten Rosebrock, Germaine Stolper, Edita Stolzenberger, Ersin Sürmeli, Kay Tietje, Till Witt, Patrick Zeising

Ein studentisches Projekt der Universität Bremen  
Wintersemester 2002/2003 bis Sommersemester 2004

Betreut von

Prof. Dr.-Ing. F. Wilhelm Bruns  
Dipl.-Inf. Daniel Cermak-Sassenrath

Bremen, 22. Oktober 2004

# Inhalt

<b>Vorwort</b> .....	1
<b>Einleitung</b> .....	2
<b>Buch I</b>	
DER PROTOTYP .....	4
1 Die Entwicklung des Prototyp .....	5
1.1 Die Rahmenbedingungen und Vorgaben .....	5
1.2 Der Bau des Cave .....	5
1.3 Die Bespannung der Holzrahmen .....	7
1.4 Die vier Wände .....	8
1.5 Der Aufwand und die Kosten .....	11
1.6 Materielle Rahmenbedingungen der Präsentation .....	11
1.7 Die Präsentation des Caves am 22.11.2002 .....	12

---

1.7.1	MiCaDo . . . . .	12
1.7.2	MiCasa . . . . .	14
1.7.3	MiCarpet . . . . .	16
2	Die Anwendungen . . . . .	18
2.1	Bremen im Nebel . . . . .	18
2.2	Das MZH . . . . .	20
3	Die Sensoren und Aktoren . . . . .	21
4	Die Wahl der Grafik-Engine . . . . .	22
4.1	Die Madness Engine . . . . .	22
4.2	Die Half-Life Engine . . . . .	23
4.3	Die Quake III Engine . . . . .	23
4.4	Die NetImmerse Engine . . . . .	23
4.5	Die Vulpine-Engine . . . . .	24
4.6	Die Unreal Engine 2 . . . . .	24
5	Fazit . . . . .	26

## **Buch II**

THE REAL DEAL . . . . .	28
1 Konstruktion des Caves . . . . .	29
1.1 Erfahrungen aus der Konstruktion des Prototypen . . . . .	29

---

1.2	Das Projektionsmaterial . . . . .	31
1.3	Überlegungen zu einer Dachprojektionsfläche . . . . .	33
2	Die Engine . . . . .	34
2.1	Entscheidung für eine Grafikengine . . . . .	34
2.2	Szenengraph der Unreal Engine 2 . . . . .	35
2.3	Materialeditor . . . . .	37
2.4	Programmierung der Unreal Engine . . . . .	46
2.5	Unreal Mutator „Cubecams“ . . . . .	46
2.6	Unreal Gametype „SahneGametype“ . . . . .	47
2.7	MiCasa Stick . . . . .	49
3	Sensorik und Aktorik . . . . .	51
3.1	Die Tanzmatte . . . . .	51
3.1.1	Von der Idee zum Konzept . . . . .	51
3.1.2	Der Aufbau der Tanzmatte . . . . .	53
3.1.3	Die Elektronik . . . . .	55
3.1.4	Entwicklung und Verbesserung des Eingabegeräts . . . . .	56
3.1.5	Materialliste . . . . .	57
3.2	Der A.R.M. . . . .	58
3.2.1	Motivation . . . . .	58

---

3.2.2	Der Entwurf . . . . .	58
3.2.3	Visualisierung des Eingabegerätes . . . . .	61
3.2.4	Anbindung und Konstruktion . . . . .	62
3.2.5	Die Software . . . . .	67
3.2.6	Fazit . . . . .	70
3.3	MiCasa Stick und Laserpointer . . . . .	71
3.3.1	Anbindung der Webcams und des A.R.M. an die Engine . . . . .	71
3.3.2	Vom Zauberstab zum Laserpointer . . . . .	71
3.3.3	Bilderkennung I . . . . .	72
3.3.4	Webcamposition . . . . .	72
3.3.5	Bilderkennung II . . . . .	73
3.3.6	Polarisation . . . . .	73
4	Umgebungen und Weltenbau . . . . .	74
4.1	Das Mehrzweckhochhaus . . . . .	74
4.1.1	Die Ebene 1 des MZH . . . . .	74
4.1.2	Die erste Präsentation und ihre Auswertung . . . . .	81
4.1.3	Eine neue Karte . . . . .	84
4.1.4	Verworfenen oder nicht realisierte Ansätze . . . . .	90
4.1.5	Static Meshes und Shader . . . . .	91

---

4.1.6	Fazit . . . . .	98
4.2	Die Kunsthalle . . . . .	101
4.2.1	Realisierung . . . . .	101
4.2.2	Texturen . . . . .	102
4.2.3	Geometrische Objekte . . . . .	106
4.3	Playground . . . . .	112
4.3.1	Die erste Version . . . . .	112
4.3.2	Die finale Umgebung . . . . .	112
4.4	Das Portal . . . . .	117
4.4.1	Entstehungsgeschichte . . . . .	117
4.4.2	Umsetzung . . . . .	119
4.4.3	Die Anordnung der Portaltüren . . . . .	120
4.4.4	Warpzones . . . . .	120
5	Finanzierung . . . . .	124
5.1	Marketing und Sponsorensuche . . . . .	124
5.2	Kosten des Caves . . . . .	124
6	Der Projekttag . . . . .	126
6.1	Die Vorbereitungen . . . . .	126
6.1.1	Das Studierhaus . . . . .	126

---

6.1.2	Der Stand . . . . .	128
6.1.3	Die Bühnenpräsentation . . . . .	133
6.2	Der Ablauf des Projekttagess . . . . .	142
6.2.1	Im Studierhaus . . . . .	142
6.2.2	Die virtuelle Ebene 1 des MZH . . . . .	142
6.2.3	Der A.R.M . . . . .	146
6.2.4	Am Stand im MZH . . . . .	149
6.2.5	Bei der Bühnenpräsentation . . . . .	149
6.2.6	Der letzte Ausklang . . . . .	149

## **Buch III**

DAS PROJEKTLEBEN . . . . .	151
1 Projektwochenenden und Exkursionen . . . . .	152
1.1 Projektwochenende Nienburg . . . . .	152
1.2 Projektwochenende Norderney . . . . .	157
1.3 Projektwochenende Dorum . . . . .	162
1.4 MiCaSa in Berlin . . . . .	166
1.5 MiCasa im Heidepark . . . . .	177
2 Das Streichholz . . . . .	182
3 Faltanleitung für den MiCasa-Würfel . . . . .	184

---

4	LAN-Wochenende . . . . .	188
4.1	Steckbrief/Einladung . . . . .	188
4.2	Erfahrungsberichte . . . . .	188
	<b>Anhang</b> . . . . .	193
1	Ausarbeitungen für den Freitags-Ausklang . . . . .	194
1.1	Sushi-Meister für die Informatiker . . . . .	194
1.2	MiCasas Argumente für die Unreal Engine . . . . .	196
1.3	Die Tanzmatte . . . . .	199
2	Die Präsentation des Prototypen . . . . .	201
3	Die Präsentation zur Halbzeit . . . . .	204
	<b>Abbildungen</b> . . . . .	206
	<b>Glossar</b> . . . . .	214
	<b>Index</b> . . . . .	228

# Vorwort

Projekte an der Universität Bremen können spannende Abenteuer sein. Spannend in mehrfacher Hinsicht.

Ein gutes Projekt erzeugt gruppendedynamische Spannungen bis zum Bersten, zerbricht aber nicht, sondern lernt diese Spannungen auszuhalten. Ein gutes Projekt erzeugt Spannungen zwischen dem fantastisch Wünschbaren und dem realistisch Möglichen. Nicht die stromlinienförmige, termingerechte Umsetzung früh spezifizierter Anforderungen in ein nützliches Produkt sind nach meinem Verständnis von Projektarbeit anzustreben - das lernen die Studen-

ten in der beruflichen Realität schnell - sondern kreative, lustvolle Selbstverwirklichung. Das geht nur an der Universität: „Usability“ versus „Enjoyment“. Nicht die oberflächliche Spaßgemeinschaft, sondern die befreiende Spielgesellschaft. Naive oder zwanghafte Wissenschaftler bezweifeln die Fruchtbarkeit eines derartig unstrukturiert spielerischen Vorgehens. Der Erfolg von MiCasa ist eine Bestätigung „offener“ Projekte. Ich freue mich mit den MiCasa-Mitgliedern über ihren Erfolg. Sowohl der Prozess als auch das Produkt sind beeindruckend. Ich bedauere, dass ich nicht aktiver und gleichberechtigter in dem Projekt mitarbeiten konnte.

Prof. Dr.-Ing. F. Wilhelm Bruns  
Bremen, Sept. 2004

# Einleitung

Jedes Informatikprojekt an der Universität Bremen sieht als letzte Aufgabe der gemeinsamen Projektarbeit die Erstellung eines Projektberichtes vor. Er soll eine ausführliche Erläuterung der erreichten Projektziele beinhalten. Diese schriftliche Auseinandersetzung mit der Projektarbeit schließt sowohl die Ergebnisse erster Diskussionen zur Zielfindung, darauf folgende Konzepte und praktische Versuche zur Konstruktion benötigter Hard- und Software als auch das Endergebnis der zweijährigen Zusammenarbeit ein.

Dieser Projektbericht teilt die vergangenen vier Projektsemester in drei inhaltlich weitgehend unabhängige Abschnitte.

In *Buch I* wird die Zielfindung des Projektes und der erste Prototyp des Cave vorgestellt.

Als eine erste Aufgabe sollte ein Prototyp des Caves innerhalb von vier Wochen gebaut werden. MiCasa setzte sich Rahmenbedingungen und nutzte diese Gelegenheit für erste Überlegungen zum Rahmenbau und zum Experimentieren mit verschiedenen Materialien für die Bespannung

des Cave. Die Ergebnisse dieser Arbeit wurden in einer Präsentation vorgestellt.

Weiterhin wurde intensive Zielfindung für die Auswahl der nötigen Grafik-Engine, mögliche Anwendungen und damit verknüpfte Interaktionen mit einer virtuellen Umgebung betrieben.

*Buch II* beschreibt das Endprodukt von MiCasa. Dies beinhaltet die für den Bau des Caves benötigten Komponenten, deren Entstehung und Funktion.

Die Konstruktion und Bespannung des Rahmens und die verwendete, von MiCasa erweiterte Grafik-Engine bilden das Grundgerüst für die weitere Arbeit zur Durchführung des Konzepts für den Cave.

Anschließend wird die Entwicklung und Realisierung von Eingabegeräten beschrieben. Es wurde eine Tanzmatte für die Fortbewegungen innerhalb der virtuellen Umgebungen gebaut, sowie ein Real-Time Motion Capturing-System zum Erfassen von Armbewegungen in Echtzeit und zum Auswählen und Transportieren von Gegenständen.

den in der virtuellen Umgebung. Zur Benutzung dieser Hardware ist eine Schnittstelle nötig, um Daten in die Grafik-Engine übertragen und verarbeiten zu können.

Zur Anwendung der Interaktionsmöglichkeiten wurden virtuelle Umgebungen geschaffen. Die Größte Umgebung bildet die Ebene 1 des Universitätsgebäudes MZH, deren Entstehung, Verbesserungen und entstandene Probleme erläutert werden. Als weitere Umgebungen liegen der Nachbau der Kunsthalle Bremen zur Vorstellung der Projektmitglieder sowie ein Playground vor, in dem mit der virtuellen Welt interagiert werden kann. Über das Portal kann der Benutzer des Caves eine Umgebung verlassen, um eine andere zu erkunden.

Die Suche nach Sponsoren, ohne deren Unterstützung dieses Projekt nicht möglich gewesen wäre, war ebenfalls Bestandteil der Projektarbeit.

Jedes Informatikprojekt findet an der Universität Bremen seinen Höhepunkt am Projekttag. Die Arbeit der vergangenen zwei Jahre wird anhand einer Bühnenpräsentation und eines Projektstandes interessiertem (Fach-)Publikum vorgestellt.

Neben dem Erarbeiten von fachlichem Wissen gehört das Sammeln von Erfahrungen bei der Gruppenarbeit zu den wichtigsten Zielen eines Projektes. Aus diesem Grund wurde dem Projektleben in *Buch III* ein Platz in unserem Projektbericht eingeräumt.

Um die Zusammenarbeit und den Zusammenhalt im Projekt zu fördern, wurden projektinterne Ausflüge nach Nienburg, Norderney und Dorum veranstaltet, auf denen zahlreiche interessante Ideen und kreative Ansätze entwickelt wurden.

Außerdem fand ein LAN-Wochenende im Projekttraum statt und MiCasa reiste auf Einladung von Willi Bruns für ein Wochenende zu ihm nach Berlin. Anekdoten komplettieren Buch III.

Im Anhang befinden sich neben einer DVD mit sämtlichen digitalen Unterlagen zum Projekt Berichte über die von MiCasa gestalteten Freitagsausklänge, einer wöchentlichen Veranstaltung aller drei Teilprojekte MiCaDo, MiCarpet und MiCasa.

# Buch I

---

DER PROTOTYP

---

# 1 Die Entwicklung des Prototyp

## 1.1 Die Rahmenbedingungen und Vorgaben

Der Prototyp sollte modular aufgebaut und nach vier Wochen fertiggestellt sein. Des Weiteren sollte er ausdrücklich nicht zur mehrfachen Benutzung betrachtet und geplant werden.

Der Prototyp diene hauptsächlich dazu, dass sich die Projektmitglieder bei der gemeinsamen handwerklichen Arbeit kennenlernen.

Zur Darstellung virtueller Umgebungen stand die Grafik-Engine Madness von Martin Faust, einem Mitarbeiter des artec, zur Verfügung. Für die Arbeit am Prototyp und die Präsentation der Ergebnisse wurden private Laptops und Computer benutzt.



Abbildung I.1: Es wurde ein Tresor als Schlüsselfach für den Projekttraumschlüssel angeschafft.

Die anschließende Diskussion um den Wegwerf-Cave, gewissermaßen eine Zielfindung für die nächsten vier Wochen, gestaltete sich produktiv. Die Mitglieder kannten sich zu diesem Zeitpunkt noch nicht, hatten zum Teil völlig kontroverse Standpunkte und Ansprüche und dennoch wurden für eine gemeinsame Standpunktbestimmung nur drei Stunden benötigt. Das Für und Wider einzelner Meinungen und Wünsche wurde erwogen, es wurde diskutiert und argumentiert und am Ende eines einzigen Nachmittages kam es zu einem von allen getragenen Ergebnis. (siehe Abbildung I.2).

## 1.2 Der Bau des Cave

Ein Nachmittag im Leben von MiCasa

Nach den ersten Plena stand nun das erste konkrete Arbeitsvorhaben ins Haus. Und die Ankunft an einem der ersten Freitage im Projekttraum barg für die meisten schon die erste Überraschung. Die Rahmengruppe hatte den Rahmen für den provisorischen Cave bereits zusammengebaut. Mit den Maßen 2,00 x 2,67 m pro Wand stand der Prototyp vor den Mitgliedern des Projektes. Nach all den theoretischen Vorbereitungen entfachte dieser Anblick eine wahre Euphorie, obwohl nicht viel mehr als ein paar zusammenge-

- Der Prototyp Der Cave soll kostengünstig sein und sein Bau soll möglichst viele Erfahrungen mit sich bringen.
- Die Konstruktion wird vier Wände und (vorläufig) kein Dach haben.
- Die Innenflächen der einzelnen Wände soll 2,67 Meter mal 2,00 Meter betragen (übliches Seitenverhältnis von 4:3).
- Die Wände können jeweils unterschiedlich konstruiert sein. Zum Bespannen der Wände sollen unterschiedliche Materialien verwendet werden.
- Konkrete Planung und Materialorganisation geschieht arbeitsteilig in Gruppen.

Abbildung I.2: Der Prototyp-Text.

schraubte Latten zu sehen waren. Was bisher lediglich in mehr oder weniger vagen Vorstellungen Einzelner existierte, war nun als ein erstes Ergebnis zu sehen. Und wie so oft in den kommenden Wochen zeigte sich, was für eine hervorragende Idee es war, das Projekt mit der Konstruktion dieses Prototypen zu beginnen.

Beflügelt von dem Gefühl zügig voranzukommen, wurde sogleich mit der Arbeit an den einzelnen Wänden begonnen. Sie sollten mit Folie, Papier und Stoff bespannt werden. Ziel war es, möglichst viele verschiedene Materialien einzusetzen und auszuprobieren. Letztlich hatten alle wenig Erfahrung mit dem Projizieren von bewegten Bildern auf eine selbstgebaute Leinwand und waren deshalb um so gespannter, welche Ergebnisse sich

erzielen lassen würden. Es bildeten sich schnell kleine Gruppen, die sich der Erstellung und Befestigung der Wände widmeten. Sie verteilten sich im Flur und auf den anliegenden Gängen des GW 2. Einzelne liefen herum halfen hier und dort, je nachdem, wo gerade eine helfende Hand gebraucht wurde. Andere tüftelten vollkommen vertieft an geeigneten Bespannungs- und Befestigungstechniken. Tesafilm und Reißzwecken wurden dabei in rauen Mengen verbaut. Es herrschte ein geschäftiges und produktives Treiben von umherlaufenden, schneidenden, nähenden, rufenden, tragenden, helfenden, schimpfenden, lachenden und weinenden Studenten.

Eine nach der anderen wurden die Wände schließlich zurück in den Projektraum getragen, um dort

ihre Feuertaufe mit dem extra ausgeliehenen Beamer zu bestehen. In der Dunkelheit erwartete alle die zweite große Überraschung an diesem Tag. Sowohl die Papierwand als auch die weiße Stoffwand lieferten extrem gute Ergebnisse, obwohl eher zu erwarten war, dass kaum etwas von dem projizierten Bild auf der anderen Seite der Projektionsfläche ankommen würde. Allerdings war die Struktur des Stoffes in der Projektion noch zu erkennen. Auch die Lichtquelle blendete noch relativ stark durch den Stoff. Beim Papier war das Bild zwar etwas matt und kontrastarm, dafür war die Lichtquelle kaum auszumachen und so entstand beim Betrachter ein sehr ausgewogener Eindruck.

Alles in allem war jenes Arbeitsvorhaben ein voller Erfolg. Die gesamte Gruppe war ein Stück enger zusammengerückt und nebenbei war auch die Arbeit noch ein gutes Stück vorangekommen.

### 1.3 Die Bespannung der Holzrahmen

Größtmögliche Mobilität und geringer Aufwand beim Auf- und Abbau des Caves waren mit die wichtigsten Faktoren, die es bei der Rahmenkonstruktion zu bedenken galt. Diese Faktoren waren natürlich nicht nur für den Rahmen selbst wichtig, sondern auch für die Projektionsflächen.

Dauerhaft fixierte Projektionsflächen könnten beim Transport leicht beschädigt werden und würden zudem verhindern, dass der Cave in seine Einzelteile zerlegt werden kann. Die Konsequenz wäre, dass sämtliche Rahmenseiten jeweils als Ganzes mit äußerster Umsicht transportiert werden müssten. Während des Transportes könnten die Projektionsflächen leicht beschädigt werden und der Rahmen könnte nicht platzsparend transportiert werden. Demnach musste eine Technik gefunden werden, bei der die Projektionsflächen nicht dauerhaft am Rahmen fixiert sind. Außerdem muss sich eine solche Projektionsfläche ohne großen Aufwand an die Rahmen anbringen und bei Beschädigung in kurzer Zeit auch austauschen lassen.

Es wurde eine Methode entwickelt, mit der sich die Seiten des Caves recht schnell bespannen lassen.

Bei dieser Technik ist davon auszugehen, dass es sich bei dem Material der Projektionsflächen um einen nähbaren Stoff handelt. In einen solchen Stoff kann jeweils an den schmalen Seitenrändern ein Saum eingenäht werden, der genug Platz bietet, ein Stahlrohr unterzubringen. Nachdem die Projektionsfläche so mit dem Stahlrohr verarbeitet wurde, wird durch die Stahlrohre jeweils ein Seil gezogen, sodass oben und unten aus den Rohren die Seilenden herausragen. Die Länge der Seilenden sollte dabei etwas mehr als die Hälfte der

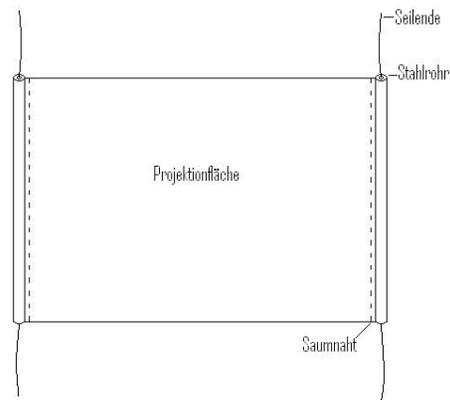


Abbildung I.3: Bespannung der Rahmen (Seitenansicht).

Rahmenbreite betragen (siehe Abbildung I.3).

Nun wird diese Konstruktion auf dem Boden ausgebreitet und eine fertig zusammengesetzte Seite des Rahmens (mit der Außenseite nach oben) auf die Projektionsfläche gelegt. Jetzt werden die Seilenden mit dem jeweiligen Seilende des anderen Seiles auf der gleichen Seite verbunden. Die Verbindung der Seilenden geschieht durch eine Ratsche, mit der die so verbundenen Rohre in Spannung zueinander gebracht werden (siehe Abbildung I.4). Auf diese Weise wird die Projektionsfläche um die Seite des Cave-Rahmens gespannt. So lassen sich die Seiten des Caves schnell und ohne großen Aufwand bespannen, sofern das Spannungsmaterial die entsprechenden Belastun-

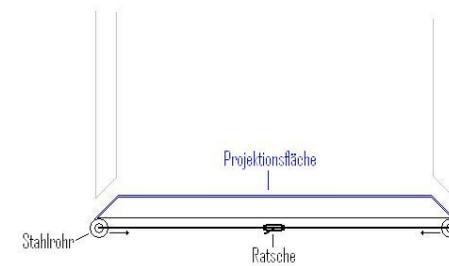


Abbildung I.4: Bespannung der Rahmen (Draufsicht).

gen aushält. Zudem wird der Transport vereinfacht und der schnelle Austausch beschädigter Projektionsflächen möglich.

## 1.4 Die vier Wände

### Weißer Stoff

Eine Seite des Caves wurde mit weißer Leinwand aus Bettwäsche in beschriebener Technik bespannt.

Die Projektion war sauber und klar, da der vorhandene Stoff aufgrund der benötigten Größe aber aus zwei Teilen zusammengenäht werden musste, zeichnete sich die Naht innerhalb des Bil-

des ab. Sollte also Stoff als Bespannungsmaterial für den Cave gewählt werden, muss dieser eine entsprechende Größe haben.

### **Papier**

Auf der Suche nach kostengünstigen Bespannungsalternativen kam die Idee auf, unbedrucktes Zeitungspapier zu verwenden. Papier ist relativ fest und lichtdurchlässig aber undurchsichtig. Der AStA stellte den dankbaren Projektmitgliedern kostenlos genügend Papier zur Verfügung. Dieses Zeitungspapier hat einen hellen, leicht gelblichen Grauton und würde eine Projektion nicht farblich verfälschen.

Bei der Bespannung des Rahmens mit Reißzwecken und Tesafilm zeigte sich, dass das Papier nicht flexibel genug war. Die aus mehreren Bahnen bestehende Projektionsfläche musste straff gespannt werden. Da der vorläufige Rahmen aber zuviel Spiel besitzt und sich beim Tragen etwas neigt, bestand bei jeder Art von Transport die Gefahr, das Papier über die Maßen hinaus zu strapazieren und die Papierfläche zu zerreißen. Zwei Projektmitglieder entwickelten die zeitaufwendige Methode der absolut parallelen, schleichenden Fortbewegung bis zur Perfektion.

Das projizierte Bild war klar und gestochen scharf. Damit gehörte diese schwierig zu transportierende Projektionsfläche zu den Favoriten,

obwohl sich bei der Überlappung einzelner Papierbahnen wie bei der Bespannung mit weißem Stoff dunkle Schatten zeigten.

Um eine Papierfläche als Leinwand für das Projekt nutzen zu können, muss noch eine andere Befestigungsmöglichkeit erarbeitet werden, damit die nötige Flexibilität gegeben ist. Außerdem könnte über andere Papierarten diskutiert werden wie beispielsweise Architektenzeichenpapier, das stärker und halbtransparent ist.

### **Plastikfolie**

Eine weitere Alternative zu Papier und Leinwand fand sich in Plastikfolie. Es wurde dafür durchscheinende Abdeckplane aus einem Baumarkt besorgt.

Da das Material für die Projektion gespannt werden musste, die Folie aber nur bedingt Spannung aushält, wurden die Ränder mit Tesafilm verstärkt. Die so haltbarer gemachten Folienränder wurden mit Reißzwecken an der Rahmenkonstruktion befestigt.

Wie spätestens am Tag der Präsentation augenfällig wurde, ist die Verteilung des Lichtes auf der Projektionsfläche sehr ungleichmäßig, sodass die Projektion kaum zu erkennen war. Im Gegensatz zu den Materialien Papier und Stoff war der hinter der Wand stehende Beamer durch die Folie deut-

lich zu erkennen. Daher kommt dieses Material für den herzustellenden Cave nicht in Frage.



Abbildung I.5: Bespannung der Projektionswand mit Folie

### Patchwork aus Stoff

Nach diversen Experimenten mit Materialien wie Watteline, Pergamentpapier etc. entschieden sich die Projektmitglieder dafür, die drei oben ge-

nannten Materialien ernsthaft auf ihre Projektionseigenschaften zu testen. Der vierte Rahmen konnte also ohne jede Absicht für eine brauchbare Projektionsfläche bespannt werden.

Diesen vierten Rahmen mit bunten Stoffstücken im Patchworkstil zu bespannen, gibt viele Möglichkeiten zum Experimentieren. Welche Stofffarben lassen beispielsweise welches Licht durch und inwiefern werden Farben verändert und welche Wirkung hat dies auf den Betrachter. Eine Schneiderin aus dem Projekt hatte nicht nur die Kenntnisse diese Leinwand zu nähen, sondern auch die nötigen Stoffstücke und eine entsprechende Nähmaschine.

Diese Wand könnte später zum Spielen mit Sensoren und Aktoren benutzt werden. Beispielsweise Glühlampen hinter den einzelnen Stoffstücken und -farben und Buzzern davor könnte die Auseinandersetzung mit Aktoren und Sensoren ermöglichen.

Als Befestigungstechnik wurde für diese Wand das Antackern gewählt. Wie erwartet, erwies sich diese Methode als wenig funktional. Das Bespannen des Rahmens war zwar problemlos möglich, doch ein Entfernen der Bespannung, zum Beispiel um den Stoff zu waschen, gelang nicht ohne Weiteres.

## 1.5 Der Aufwand und die Kosten

Eine Stehlampe, zehn Verlängerungskabel, zwei Kabeltrommeln, sieben Mehrfachsteckdosen, zwei Switches, vier Beamer, zwei Laptops, zwei Standrechner, vier leere Ordner, hundertzwei- undvierzig Heftzwecken, siebenundfünfzig Meter Tesafilm, vier Bettbezüge, eine Abdeckfolie, eine Menge Platz, zehn Meter Seil, zwei Frösche, achtundvierzig Schrauben, sieben Winkel, zwei Scharniere, zwei Wischtücher, achtzehn Quadratzentimeter Alufolie, dreiundzwanzig Meter weißes Nähgarn, siebzig Meter grünes Nähgarn, einen Kopfkissenbezug, drei Vorhänge, hundertdreizehn Tackerklammern, siebenunddreißig Meter und achtundzwanzig Zentimeter Holzleiste, sechs Quadratmeter und sechzehn Quadratmeter Papier, zweiundfünfzig Meter Netzkabel, vierzig Zentimeter dünnes Kabel, ein Tacker, eine Bohrmaschine, ein Akku-Bohrer, zwölf Meter Absperrband, drei Tische, eine Nähmaschine, ein Stück Schneiderkreide, ein Handmaß, eine Schere, ein Zentimetermaß und vier Stahlrohre wurden für Bau und Präsentation des Prototypen verwendet. Dafür wurden drei Euro und fünfundsiebzig Cent aus der Projektkasse aus-

gegeben und alle anderen Kosten wurden privat getragen.

## 1.6 Materielle Rahmenbedingungen der Präsentation

Für die Präsentation des Prototypen stellte das artec einen Beamer bereit. Ein vom artec an MiCasa überreichter, neuer Switch erwies sich als defekt, sodass kurzfristig ein Switch von MiCaDo ausgeliehen werden musste, um das für die Präsentation notwendige Netzwerk einzurichten.

An dieser Stelle besonderen Dank an die AG Rechnernetze, die AG Rechnerarchitektur und den Technik-Pool der Angewandten Informatik für geliehene Beamer, Verlängerungskabel und Mehrfachsteckdosen. Ebenso vielen Dank an die Haustechnik im GW 2 für das pünktliche Ausschalten des Lichtes am Präsentationsort.

Andere technische Bestandteile der Präsentation, wie vier arbeitsfähige Computer, diverse Verlängerungskabel und Mehrfachsteckdosen, Werkzeug und verschiedenstes Verbrauchsmaterial kamen aus Beständen der Projektteilnehmer.

## 1.7 Die Präsentation des Caves am 22.11.2002

Mica - Die Reportage: Schonungslos, Kritisch, Analytisch

### 1.7.1 MiCaDo

- Der Cave:

Der Prototyp von MiCaDo bestand aus drei Leinwänden, die an der Decke aufgehängt wurden. Die Leinwände bestanden aus Stoff und hatten oben und unten je eine Holzstange (oben zum Aufhängen, unten als Gewicht und damit für die Spannung). Angeordnet waren die Leinwände im 30 Grad Winkel, so dass sich daraus eine Art Cockpitsicht ergab.

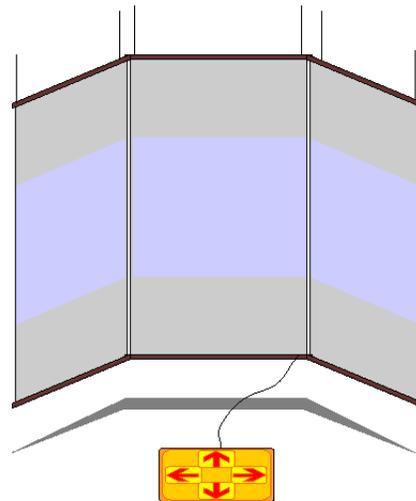


Abbildung I.6: Skizze des MiCaDo-Caves.

Nicht unbeachtlicher Vorteil der Aufhängung war ihre Transportabilität - die Leinwände ließen sich einfach zusammenrollen. Nachteile dieser rahmenlosen Konstruktion: Die Leinwände geraten durch Luftzug in starke Bewegung und schlagen zudem durch die Aufhängung leichte Wellen. Angeschlossen war auch eine Druckmatte, die es Benutzern ermöglichte, sich einer virtuellen Welt der Madness Engine zu bewegen.

- Präsentation:

Zunächst hat ein Repräsentant von MiCaDo das Publikum ein bißchen durch die projizierte Welt geführt, während Stimmen aus dem Off und Gemurmel von MiCaDolern ein bißchen zum Cave erklärt haben. Anschließend durfte dann noch eine geneigte Testperson das Vergnügen haben. Nun ja, der Autor dieser Zeilen wurde freundlich genöti... gebeten, besagte geneigte Testperson zu sein.

Die Kontrolle via Druckmatte erwies sich als nicht ganz so simpel, wie es vorher aussah. Nach verschiedenlicher Bekanntschaft mit den Wänden und Hindernissen der virtuellen Umgebung bekam der Nutzer ein gewisses Gefühl für die Steuerung.

Warum sah das dann bei Simon so einfach und flüssig aus? Ganz einfach: Er hat die Steuerung

im Sitzen bedient. So konnte er, analog zu einer Cursorsteuerung per Tastatur, entsprechende Tastenfelder drücken und loslassen, um während des Gehens nach links oder rechts zu laufen oder einfach stehenzubleiben. Im Stehen würde der Besucher nach kurzer Zeit das Gleichgewicht verlieren, da mit den Füßen nicht so schnell Tastenfelder gewechselt werden können und gleichzeitige Eingaben z. B. für Laufen oder eine Bewegung in eine bestimmte Richtung schwierig durchzuführen sind.

Das Fazit: Für die Cavesteuerung erschien die Druckmatte nicht praktisch. Sie ist wenig intuitiv und benötigt Übung. Zumal es für das menschliche Begreifen schlichtweg nicht üblich ist, eine Bewegung wahrzunehmen, die durch Herumlaufen entsteht, während aber gleichzeitig die Füße völlig atypische Bewegungen ausführen müssen, um diese Bewegung zu erzeugen. Würde dem Benutzer noch abverlangt, gleichzeitig etwas mit den Händen zu tun (wie eine Waffe zu führen, einen Knopf zu drücken etc.), wäre das wahrscheinlich er damit wahrscheinlich überfordert.

Was die Darstellung der Welt im Cave betrifft, so war der Effekt ansehnlich. Wenngleich der Abstand vom Betrachter zur Projektionsfläche größer sein müsste. Kommentare dazu waren etwa „erdrückend“, „zu viel Bewegung“ und „man weiß gar nicht, wohin man immer so schnell gucken soll“. Das menschliche Bewusstsein kann die vir-

tuelle Bewegung nur mühsam verarbeiten, mitunter auch deshalb, weil die normalerweise zugehörige Bewegung der Beine fehlt. Und bei einem geringen Abstand sind die vielen Bewegungen um einen herum dann zuviel Information auf einmal. Hingegen wurde verschiedentlich positiv angemerkt, dass der Übergang zwischen den Leinwänden mit dem 30 Grad Winkel angenehmer sei als bei 90 Grad Lösungen. Für einen geschlossenen Cave bräuchte man dann lediglich die Kleinigkeit von zwölf Beamern ...

- Das Ding zum Fliegen:

MiCaDoler Lars hat als Abschluss der Präsentation noch eine kleine Fisherprice-Tüftlei vorgeführt, die zur Realisierung einer Anwendungsidee dienen soll: Der Benutzer soll - ähnlich einem Vogel - fliegen können.

Lars hatte auch einige Erklärungen dazu gegeben: Das vorgeführte Eingabegerät soll irgendwo am Boden stehen und verfügt über zwei bewegliche Hebel links und rechts. Diese sind über Fäden mit den Händen des Benutzers verbunden. Die Bewegungen, die der Benutzer nun mit seinen Armen macht, werden an die Hebel weitergegeben und sollen dann vom Programm als Eingabe interpretiert werden. So wird eine Bewegung, bei der gleichzeitig beide Arme auf und ab bewegt werden, als Flügelschlag interpretiert, der den Benutzer virtuell höher steigen lässt. Rechts-

und Linkskurven können ausgeführt werden, indem der Benutzer je einen Arm nach oben und einen nach unten hält (die Kurve wird in Richtung des Arms geflogen, der unten ist). Keine Armbewegungen resultieren in Gleitflug, wobei man damit langsam an Höhe abnimmt.

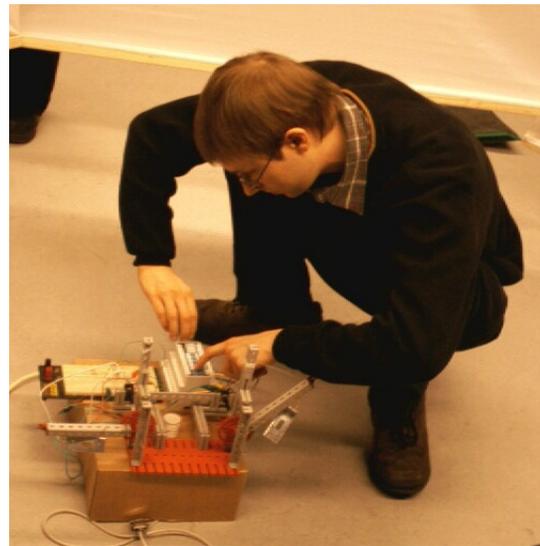


Abbildung I.7: Das Gerät soll Flugbewegungen erkennen und dem Programm als Input liefern.

Dazu gehört noch das Ausgabegerät: Der Benutzer liegt auf einer Plattform, die über Pneumatikzylinder bewegt werden soll (hat sich hier Willi eingebracht?! Man weiß es nicht ...). Das gibt dem Benutzer das Gefühl, Kurven sowie Steig- und Sinkflüge am eigenen Leib mitzuerleben. Ein

Ventilator soll für simulierten Flugwind (und eine kleine Erkältung) sorgen.

### 1.7.2 MiCasa

- Der Cave:

Eine Idee hinter der Entwicklung des Prototypen war mitunter, ein wenig herumzuprobieren und Erfahrungen zu sammeln, ein Gefühl dafür zu bekommen, wie beispielsweise verschiedene Materialien wirken, welche Dimensionen ein Cave haben muss etc. Diese Idee hat wohl nur das Projekt MiCasa verfolgt. Dessen Cave bestand aus vier Rahmen für die Wände, jeweils etwa 2 1/2 Meter breit und 2 Meter hoch. Bespannt waren sie mit vier verschiedenen Materialien: Zeitungspapier, Folie, weißem Stoff und bunten Stoffflicken.

Für die Präsentation wurden die vier Wände verschraubt, eine mit Scharnieren versehene Wand diente als Eingang. Der Cave war - als einziger der präsentierten Caves - geschlossen. Jede Seite wurde mit einem Beamer bestrahlt.

Die Bewegung durch die Welt wurde allerdings vorgegeben und konnte nicht von den Besuchern beeinflusst werden.

Da die Präsentation gewisse Räumlichkeiten voraussetzte und zudem die sehr fragile Papierwand eine Reise ins SFG möglicherweise nicht überstanden hätte, wurde die Präsentation im



Abbildung I.8: Volle Breitseite: Insgesamt vier unterschiedlich bespannte Wände mit vollflächigen Projektionen bei MiCasa.



Abbildung I.9: Im MiCasa-Cave war reichlich Platz für viele Teilnehmer.

Tischtennisareal der großen Höhle der Grenzfahrungen, dem GW 2, durchgeführt.

- Die Präsentation:

Zu Beginn wurden dem Publikum in einer kurzen Einführung die Intentionen des Projektes nahegebracht, wie zum Beispiel das experimentelle Verwenden verschiedener Materialien. Es wurde noch die Alternative zu einer Druckmatte erläutert - Drucksensoren, die direkt unter einem Paar Badelatschen angebracht werden konnten und mit denen realistisches Laufen im Cave möglich ist.

Die Papier- und die weiße Stoffwand haben zwar aufgrund der großen Entfernung der Beamer zu den Wänden (und der damit einhergehend schwä-

cher werdenden Intensität des Lichtes) nicht ihre volle Wirkung entfalten können, konnten aber dennoch demonstrieren, ob und inwiefern sich die jeweiligen Materialien zur Projektion eignen. Dass es einige Schwierigkeiten bei der Ausrichtung der projizierten Bilder gab, war dabei eher nebensächlich.

Die bunte Patchworkwand hat die Besucher hauptsächlich irritiert (zumal die Projektion eher schlecht zu sehen war) - trotzdem hat sie einige Besucher zu interessanten Ideen beflügelt und kleine Diskussionen bezüglich Farbeffekten angeregt.

Apropos Effekt: Besonders interessant fanden viele Besucher auch den doch eher surrealistischen

Effekt der Folienwand und hatten angeregt, dabei doch mal mit verschiedenen Farben, Lichteffekten und ambienter Musik zu experimentieren. Zudem mag die Folie durch ihre ständige wellenartige Bewegung durch Windzüge eine besondere Wirkung bei der Realisierung einer Unterwassersimulation entfalten.

### 1.7.3 MiCarpet

- Der Cave:

MiCarpet hatte sich ebenfalls für einen quaderförmigen Cave entschieden - drei Wände, rechtwinklig zusammengebaut. Die Rahmenkonstruktion war professionell gemacht: mit Winkeln verschraubte Holzbalken. Bespannt waren für die Präsentation drei der vier Wände mit halbtransparentem Zeichenpapier, wie es beispielsweise von Architekten verwendet wird. Das bewirkt eine beachtliche Farbbrillanz, hat allerdings zwei - wenn auch nur kleine - Nachteile: Zum Einen wirken helle Szenen mitunter etwas milchig und zum Anderen reflektiert das Material leicht die Projektionen der anderen Wände.

Die Art der Bespannung war nicht uninteressant: die oberen und unteren Gerüstbalken hatten je eine Reihe von 20 Nägeln, die Leinwände oben und unten eine Reihe von Gummibändern. Die Leinwände wurden so einfach mit den Gummibändern an den Nägeln befestigt und erhielten

auf diese Weise auch Straffheit.

Die Projektion deckte nur etwa den mittleren Teil der Leinwände ab, unten blieb etwa ein halber Meter ohne virtuelle Welt.

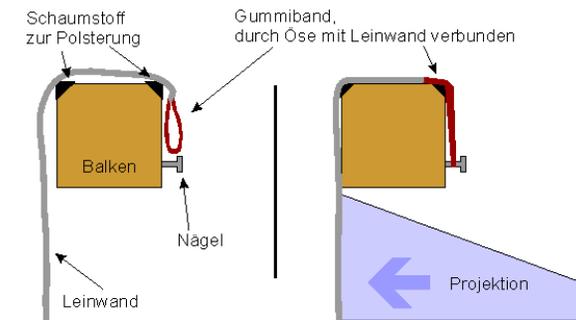


Abbildung I.10: MiCarpet hat die Leinwände mit einer interessanten Bespannungstechnik am Rahmen angebracht.

Die Bedienung sollte über eine Druckmatte erfolgen, die allerdings nicht angeschlossen war. Es handelte sich um dieselbe Tanzmatte, die auch MiCaDo verwendet hatte.

- Die Präsentation:

MiCarpet hat die interessante Taktik gewählt, den Cave für sich selbst sprechen zu lassen. Daher hat niemand großartig etwas über die Konstruktion erzählt oder die Ideen, die dahintersteckten. Die Bewegung durch die virtuelle Welt wurde hinter den Kulissen vorgegeben, jedoch konnten sich interessierte Besucher in den Cave wagen.

Zwar war die Intensität und damit auch der Effekt der projizierten Madness Wanderung durchaus ansehnlich, aber der Cave war unangenehm klein. Das machte die Wanderung zuweilen zu einer Grenzerfahrung für das Wahrnehmungssystem - je schneller man sich durch die Welt be-

wegte, um so unangenehmer war der Effekt. Einige derjenigen, die den Kopf in den Cave gesteckt haben, erzählten auch etwas von Schwindelgefühlen, die bei der kurzen Entfernung zur Leinwand entstanden.

## 2 Die Anwendungen

### 2.1 Bremen im Nebel

Die Grundidee besteht darin, verschiedene Anwendungsmöglichkeiten des Caves mit Aktorik und Sensorik innerhalb eines großen Rahmens zu verbinden und so zu präsentieren.

Dieses Ziel soll erreicht werden, indem alle Einzelprojekte innerhalb eines virtuellen Stadtplanes Bremens stattfinden. Dieser gemeinsam genutzte Raum kann die Verbindung zwischen Projekten sehr unterschiedlichen Inhaltes und sehr unterschiedlicher Form darstellen.

So ist es möglich in Kleingruppen von wenigen Projektmitgliedern spezielle Ideen zu realisieren und trotz der spezifischen Interessen dennoch zum Gesamtprojekt beizutragen. Damit können die technischen Möglichkeiten, die ein Cave mit sich bringt, in vielfältigster Weise erforscht und auch genutzt werden.

Es entstanden mehrere Ansätze zur Umsetzung von Gebäuden und Umgebungen aus dem Bremer Raum.

- **Kunsthalle:**

Die Kunsthalle Bremen soll als Ausgangspunkt dienen. Die verschiedenen Bilder der Kunsthalle stellen Durchgangsstellen zu den anderen Maps dar.

- **Glocke:**

Da die Glocke nicht nur ein interessantes Gebäude, sondern auch ein Veranstaltungsort ist, bietet sie sich für Experimente mit Musik an. Denkbar sind sowohl die Umsetzung menschlicher Bewegungen in Töne und Formen durch das Einlesen von Standort und Bewegung, als auch die direkte Steuerung durch Eingabegeräte.

- **Bremer Rathaus:**

Das Rathaus ist eines der schönsten und ältesten Gebäude in Bremen. Es bietet die Möglichkeit, sowohl historische als auch aktuelle Themen aufzubereiten. Je nach konkreter Ausgestaltung ist eine Nutzung dieser Anwendung durch die Tourismusindustrie denkbar.

- **Bürgerschaft:**

Von dem Stuhl eines Abgeordneten aus kann mit vier Kameras eine Rundumsicht von einer Sitzung der Bürgerschaft gefilmt werden. Interaktionsmöglichkeiten sind dann allerdings nicht vorstellbar, denn BürgerInnen können auch in der Realität nicht direkt in die politischen Prozesse eingreifen.

- **Hauptbahnhof Bremen:**

Da von diesem belebten Platz viel historisches Bildmaterial vorhanden ist, bietet er sich für eine Zeitreise an. Sowohl die Gebäude als auch Men-

schen und ihre Bekleidung können hier im Wandel der Zeit dargestellt werden.

- Knoop's Park:

Knoops's Park bildet den Hintergrund für das bekannte Buch „Sommer in Lesmona“. Das Buch wäre der Hintergrund für die Anwendung.

- Schnoor:

Der Schnoor ist einer der verwinkeltsten Stadtteile Bremens. Dadurch ist er besonders für Labyrinthspiele und Schnitzeljagden geeignet. Außerdem bietet er sich aufgrund seines Alters für virtuelle Führungen mit klassischem Bildungsanspruch an.

- Bremer Häuser:

Eine Straße mit den typischen Bremer Häusern eignet sich besonders gut, um den Wandel von Wohnumgebungen darzustellen. Die Grundrisse der Häuser in einer Straße sind oft identisch, zumindest aber ähnlich. Deshalb bietet sich diese Umgebung dazu an, jedes Haus im Stil einer anderen Epoche einzurichten und so in vergleichbaren Umgebungen den Wandel in der Innenarchitektur zu veranschaulichen.

- Hafenbecken Bremerhaven:

Eine simulierte Fahrt mit dem dort in der Realität vertäuten U-Boot ist möglich. Da die heimische Flora und Fauna unter Wasser sehr langweilig ist, scheint ein reines Bildungsangebot in

dieser Richtung weniger geeignet.

Auch ein Spaziergang auf dem Hafengrund ist denkbar vielleicht verbunden mit einer Schatzsuche.

- Bildung:

Der Cave bietet sich dafür an, Abläufe hinter normalerweise verschlossenen Türen, zum Beispiel in Unternehmen, darzustellen.

- Was passiert mit einem Brief vom Briefkasten zum Briefträger?
- Welche Stellen passiert ein Überweisungsschein?
- Wer arbeitet in der Bürgerschaft eigentlich was, wie und wann?
- Darstellung von (historisch) interessanten Gebäuden mit Führung und Interaktionsmöglichkeiten

- Politik:

Der Cave ist geeignet, einen Beitrag zur Lösung von Umgestaltungsproblemen in Stadtteilen zu leisten. Anwohner und Politiker können das Szenario mit den Veränderungen vor der Umgestaltung begehen und die aktuelle Planung gegebenenfalls ändern.

## 2.2 Das MZH

Der virtuelle Nachbau des MZH bietet viele Gestaltungs- und Interaktionsmöglichkeiten.

Die Umsetzung des Spiels Bomberman in einer dreidimensionalen Umgebung. Hierbei ist auch eine Version mit mehreren Spielern (sogar in unterschiedlichen Caves denkbar). Auch die Umsetzung von anderen bekannten Spielen wie PacMan ist möglich, Interaktionen mit Monstern erhöhen den Spaß am Spiel.

Die geheimnisvolle Welt der 9. Ebene des MZH könnte erkundet werden, die durch den Lastenfahrstuhl erreichbar sein soll. Dort kann dann der Swimmingpool benutzt oder sich eine Margarita an der Bar genehmigt werden.

Bewegungsfolgen des Nutzers können auf einer Kontaktmatte in konkrete Bewegungen im virtuellen Raum umgesetzt werden. So wird ihm die Möglichkeit geboten, sich frei im Szenario zu bewegen und mit seiner Umgebung zu interagieren.

### 3 Die Sensoren und Aktoren

- Taschenlampe:

Mit einer virtuellen Taschenlampe könnte der Weg durch einen dunklen Raum gefunden werden, der den normalen, bekannten physikalischen Eigenschaften der normalen Umgebung widerspricht. So könnte der Weg sich zum Beispiel in Spiralform durch die virtuelle Welt ziehen, der dem Nutzer des Caves jegliches Gefühl für die Raumkoordination nimmt.

- Gegenstände bewegen:

Es könnte eine Schnittstelle realisiert werden, die es dem Benutzer ermöglicht Gegenstände im virtuellen Raum zu bewegen beziehungsweise zu manipulieren. Als Beispiel für eine Anwendung könnte eine Person im Cave einen Graph durch

Bewegungen manipulieren und sich diese Änderungen direkt auf eine Formel auswirken.

- Zaubern:

Durch Bewegen des Arms oder der Hand könnte es möglich sein, einen Befehl oder eine Aktion im Cave auszulösen. So zum Beispiel das Verschwinden von Gegenständen, Bewegen von Objekten etc.

- Simulation:

Ein Stuhl mit Bedienelementen könnte eine Flugsimulation im Weltraum ermöglichen. Diese Anordnung könnte natürlich auch eingesetzt werden, um einen virtuellen Bagger oder andere große Geräte zu steuern.

## 4 Die Wahl der Grafik-Engine

Um eine virtuelle Welt darstellen zu können, wird eine Grafik-Engine benötigt. Vom artec wurde dem Projekt die Madness Engine zur Verfügung gestellt. Die Madness Engine wurde bei der Präsentation des Prototyps (vgl. Kapitel 1.7.2) eingesetzt. Dabei zeigte sich, dass die Madness Engine noch nicht ausgereift war und so auch andere Engines für die zukünftige Projektarbeit in Betracht gezogen werden konnten.

Die Anforderungen an eine Grafik-Engine sind sehr vielfältig. Zum Einen wurde eine moderne, leistungsfähige Grafik-Engine benötigt, bei der ein Zugriff auf den Sourcecode der Engine selbst oder einer Anwendung möglich ist. Zum Anderen ist es wichtig, dass ausreichende Software zur Verfügung steht, um mit dieser Engine zu arbeiten und Anwendungen dafür zu entwickeln. Dazu gehören Mapeditoren, Programmierprogramme, Textur- und Modellingsoftware, etc.

Unter diesen Aspekten werden im Folgenden die Half-Life Engine von Valve (<http://www.valvesoftware.com/>), die Quake III Engine von ID Software (<http://www.idsoftware.com/>), die NetImmerse Engine von ND/ Gamebryo (<http://www.ndl.com/>), die Vulpine-Engine der Firma Vulpine (<http://www.vulpine.de/>) und die Unreal Engine 2 der Firma Epic Games Inc. (<http://www.epicgames.com/>) diskutiert.

### 4.1 Die Madness Engine

Diese Engine basiert auf der Quake III Engine und wurde von Martin Faust, einem Mitarbeiter des artec, entwickelt. Die Entwicklung war zum Zeitpunkt des Auswahlverfahrens noch nicht abgeschlossen. Der Vorteile dieser Engine liegt im direkten Kontakt zum Entwickler, wodurch eine schnelle Unterstützung bei Fragen und Problemen gewährleistet ist. Außerdem gibt es keine Probleme mit eventuellen Lizenzen, da der komplette Sourcecode zugänglich ist.

Ein Nachteil bestand darin, dass die Engine nicht fertig entwickelt war, viele Funktionen waren noch nicht implementiert, außerdem gab es noch einige Probleme mit unvollständigem Netcode. Auch die Qualität der Darstellung war nicht mit der aktueller, kommerzieller Engines zu vergleichen.

Die Engine wird zwar fortlaufend weiterentwickelt, es war jedoch nicht sichergestellt, ob eine ausreichende Entwicklungsstufe rechtzeitig, also innerhalb der nächsten zwei Jahre, erreicht werden würde.

Letztendlich wäre der Aufwand für eine Weiterentwicklung zu groß gewesen und daher kam diese Engine für das Projekt nicht in Frage.

## 4.2 Die Half-Life Engine

Die Half-Life Engine kommt im Spiel „Half-Life“ aus dem Jahr 1998 zum Einsatz. Aufgrund ihres Alters war diese Engine schlecht mit den Leistungen aktueller Engines zu vergleichen. Sie bot aber einige wichtige Vorzüge gegenüber anderer Engines.

„Half-Life“ ist als eines der erfolgreichsten Spiele der letzten Jahre stark verbreitet, es gibt daher unzählige Modifikationen, entwickelt von enthusiastischen Liebhabern. Die Entwicklerszene ist mit Newsforen, Infoseiten und Tutorials im Internet dementsprechend groß. Es besteht kostenloser Zugriff auf ein extrem umfangreiches Wissen.

Wird die Engine in der Umgebung eines Spieles wie zum Beispiel Half-Life „Half-Life“ eingesetzt, so sind auch keine Probleme bei der Datenübertragung zwischen verschiedenen Rechnern zu erwarten, da der Netcode bereits relativ ausgereift und durch die Anwendung in zahlreichen Onlinespielen (z. B. Counter-Strike) erprobt ist. Über eine selbstentwickelte Modifikation könnte das Spiel „Half-Life“ an die Anforderungen im Projekt angepasst werden.

Diesen Vorzügen gegenüber steht die nach heutigen Maßstäben eher bescheidene grafische Leistung der Engine. Dennoch wurde die Half-Life Engine als Alternative betrachtet.

## 4.3 Die Quake III Engine

Mit der Quake III Engine kann unter bestimmten Voraussetzungen eine einigermaßen gute Darstellung erzielt werden (bei Computerspielen wie „Return to Castle Wolfenstein“ oder „Medal of Honor Allied - Assault“).

Allerdings wird in diesen Spielen nicht die ursprüngliche Version der Quake III Engine verwendet, vielmehr wurde sie als Basis benutzt, weiterentwickelt und teilweise stark modifiziert. Der Quellcode dieser Weiterentwicklungen und Modifikationen ist jedoch nicht öffentlich zugänglich. Die reine Quake III Engine gleicht in der Darstellungsqualität der Madness Engine und liegt damit ebenso wie die Madness Engine in ihrer Grafikleistung weit hinter allen anderen hier betrachteten Engines zurück. Somit war klar, dass die Quake III Engine nicht als Alternative in Frage kommt.

## 4.4 Die NetImmerse Engine

Die NetImmerse Engine bietet eine gute Grafikleistung, Shader werden unterstützt und es gibt verschiedene Entwicklerprogramme, die sich für diese Engine verwenden lassen.

Die Entwicklung von Anwendungen für die Engine ist jedoch problematisch. Zugriff auf den Sour-

cecode der Engine selbst besteht nicht und es gibt keine Anwendung für diese Engine, deren Sourcecode verfügbar ist. Ebenso ist es nicht möglich, eine bestehende Anwendung über das Entwickeln einer Modifikation so anzupassen, dass die Anforderungen des Projektes erfüllt werden. Daher scheidet die NetImmerse Engine als Alternative aus.

#### 4.5 Die Vulpine-Engine

Diese Engine wurde in Deutschland entwickelt. Zum Zeitpunkt des Auswahlverfahrens liefen einige Demoanwendungen der Firma Vulpine mit dieser Engine.

Auf Grund der dort gesehenen Grafikleistung wurde die Vulpine-Engine für die Anwendung im Cave favorisiert. Der größte Vorteil dieser Engine wäre der direkte Support durch eine deutsche Firma, ähnlich wie bei der Madness Engine. Entwicklertools sind genügend zu bekommen und alle weiteren Informationen waren der Dokumentation zu entnehmen. Zwischenzeitlich wurde dem Projekt unter bestimmten Bedingungen sogar die Verwendung des Sourcecodes der Engine in Aussicht gestellt.

Gleichzeitig wären sämtliche Arbeitsergebnisse des Projektes und Modifikationen an der Vulpine-Engine unter eine zuvor von jedem Projektmitglied zu unterzeichnende Geheimhaltungsverein-

barung gefallen. Aus diesem Grund wurde die Vulpine-Engine nicht für die weitere Projektarbeit ausgewählt.

#### 4.6 Die Unreal Engine 2

Die Unreal Engine 2 wurde von der Firma Epic Games Inc. entwickelt. Auch diese Engine zählt aufgrund ihrer guten Grafikleistung zu den Favoriten. Sie erfüllt ebenso wie die NetImmerse Engine und die Vulpine-Engine in dieser Hinsicht alle Anforderungen.

Als Anwendungen gibt es Spiele wie „Unreal Tournament 2003“ oder „Unreal 2“. Der Sourcecode des Spiels „Unreal Tournament 2003“ ist verfügbar, Entwicklertools werden in großem Umfang zusammen mit dem Spiel und der Lizenz geliefert. Auch für „Unreal Tournament 2003“ gibt es eine große Entwicklerszene für Themen wie Modifikationen und Maps. Es sind umfassende Tutorials und Newsforen zu finden, sodass bei Fragen und Problemen schnelle Hilfe erwartet werden kann. Zusätzlich gibt es auch seitens Epic Unterstützung für die Entwicklerszene. Nach jedem Patch veröffentlicht Epic auch den neuen Sourcecode für „Unreal Tournament 2003“. Dadurch wird das Erstellen von Mods wesentlich vereinfacht.

Die Nutzung der Unreal Engine 2 ist für nicht-

kommerzielle und für Zwecke der Lehre kostenfrei nutzbar und einige lizenzierte Versionen des Spiels samt Entwicklersoftware waren im Projekt bereits vorhanden. Daher ist diese Engine eine echte Alternative zur Madness Engine.

Als Alternativen zur Madness Engine haben sich die Half-Life Engine in Verbindung mit dem Spiel „Half-Life“ und die Unreal Engine 2 mit dem Spiel

„Unreal Tournament 2003“ qualifiziert.

Die beiden Letzteren unterscheiden sich im Prinzip nur in ihrer grafischen Leistung. Sowohl Entwicklertools als auch genügend Hilfsquellen sind für beide Engines vorhanden, Probleme mit eventuell zu erwerbenden Lizenzen gibt es nicht.

Die Wahl fiel aufgrund der überzeugenden grafischen Leistung auf die Unreal Engine 2.

## 5 Fazit

Bei der Präsentation des Prototypes zeigte sich, dass der Rahmen in der gewählten Bauweise nicht sehr gut zu transportieren war. Die Höhe der einzelnen Wände bereitete Probleme beim Durchqueren von Türen und das fest angebrachte Projektionsmaterial erwies sich beim Transport als hinderlich und im Fall des Zeitungspapiers auch als sehr empfindlich.

Bei der Nutzung des Caves wurde deutlich, dass die gesamte Konstruktion etwas instabil war. Als eine einigermaßen praktikable, dennoch verbesserungswürdige Lösung erwies sich der Ein- und Ausgang, beim dem eine Seite des Caves auf- und zugeklappt wurde.

Die vier Materialien, die als Projektionsoberfläche gewählt worden waren, zeigten ebenfalls verschiedene Vor- und Nachteile.

Die Folie und das Papier waren schwer zu befestigen und reagierten sehr empfindlich auf Bewegungen jeder Art. Die Folie war zudem sehr lichtdurchlässig, das Bild nicht klar zu erkennen. Das Papier zeigte sehr gute Projektionsergebnisse.

Die Leinwand zeigte insgesamt das beste Ergebnis. Zwar war die Struktur des Stoffs bei der Projektion zu erkennen, dennoch entstand ein klares und ausreichend helles Bild. Das Material war widerstandsfähig und ließ sich gut spannen.

Bei Papier und Stoff entstanden jedoch Klebe-

streifen beziehungsweise Nähte, da die Rohmaterialien nicht ausreichend groß verfügbar waren und somit ungewollte Schatten bei der Projektion.

Die Methode, mit der ein Rahmen mit Stoff gespannt wurde, war kompliziert und langwierig und musste für das Endprodukt neu überdacht werden.

Die sperrigen Rahmen ließen sich ohne Bespannung viel einfacher transportieren. Es ist demnach sinnvoll, das Material erst am Präsentationsstandort des Caves zu befestigen und beim Abbau des Caves wieder abzunehmen.

Für das Endprodukt ergaben sich also folgende Anforderungen: Zum Einen muss der Rahmen transportabel gestaltet werden, d.h. er muss in seine Einzelteile zerlegt werden können, damit auch kleinere Durchgänge kein Hindernis darstellen. Des Weiteren muss er stabiler konstruiert werden als der Prototyp, es muß also stärkeres Holz verwendet werden.

Die Suche nach dem besten Projektionsmaterial war mit der Präsentation des Prototypes noch nicht abgeschlossen, ein ideales Material war noch nicht gefunden. Dies würde weiterer Versuche bedürfen.

Auch für die Bespannung gab es noch keine Lösung, die alle Anforderungen erfüllte, es wurde

aber deutlich, dass die Seiten flexibel und schnell zu bespannen sein müssen.

Ebenso muss sich eine Arbeitsgruppe noch um eine vernünftige Eintrittsmöglichkeit bemühen, durch die der Cave betreten werden kann.

Insgesamt erwies es sich als sehr hilfreich, zuerst einen Prototyp innerhalb weniger Wochen zu entwickeln. So bekam jedes Projektmitglied eine Vorstellung, was in den nächsten zwei Jahren Schwerpunkt der Projektarbeit sein wird.

# Buch II

---

THE REAL DEAL

---

## 1 Konstruktion des Caves

### 1.1 Erfahrungen aus der Konstruktion des Prototypen

Erste Überlegungen zur Konstruktion eines Caves sollten an einem Prototypen überprüft werden. Dieser ist innerhalb der ersten Wochen des Projektes gebaut und am 22. November 2002 präsentiert worden.

In den wesentlichen Grundzügen hat sich der Prototyp als sehr brauchbar erwiesen. Die Abmessungen sind geeignet, die Vision einer virtuellen Welt für mehrere Besucher gleichzeitig abzubilden. Das Verhältnis von Breite und Höhe (4:3) entspricht den Projektionsbedingungen der LCD-Projektoren.

Die Konstruktion des Holzrahmens hat ihre Aufgabe gut erfüllt, besonders im Hinblick auf den befürchteten Schattenwurf der Rahmenbalken auf die Projektionsfläche. Durch die diagonale Durchtrennung der vertikalen Rahmenbalken und die Durchführung der Projektionsfolie durch diese Diagonale wurde der Schattenwurf fast vollständig vermieden (siehe Abbildung II.1).

Ein kritischer Punkt ist die gewählte Dicke der Rahmenbalken des Prototyps. Die Belastungen beim Auf- und Abbau und beim häufigen Rein- und Rausgehen der Cavebesucher haben gezeigt, dass die Gesamtkonstruktion zu instabil ist. Au-

ßerdem ist der Cave in der Prototypfassung nicht ausreichend mobil. Der Prototyp besteht aus vier fest verschraubten Wandelementen, die nur zusammengestellt und untereinander fixiert werden müssen.

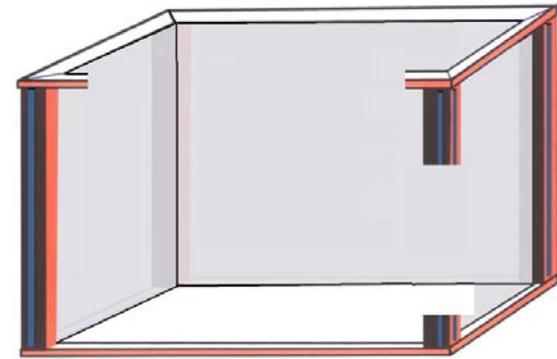


Abbildung II.1: Grundlegender Aufbau des Cave-Rahmens.

Der Zeitaufwand für den Aufbau ist dadurch nicht sonderlich groß, jedoch müssen die vier Wandelemente jeweils als Ganzes einzeln und mit äußerster Umsicht transportiert werden. Gerade beim Transport innerhalb des GW 2 ist dies nicht einfach, da die Türen dort nur wenig höher sind als die Rahmen des Caves. Des Weiteren ist zu berücksichtigen, dass bei künftigen Präsentationen an anderen Orten unter Umständen mit noch niedrigeren Türen und engeren Fluren zu rech-

nen ist. Außerdem wäre ein Transport mit einem Fahrzeug mit der gegenwärtigen Konstruktion nicht ohne weiteres möglich. Damit ist klar geworden, dass Konstruktionsänderungen erforderlich sind, um eine höhere Mobilität zu erreichen.

Unzureichend hat sich auch der Zugang in das Innere des Caves erwiesen. Um in den Prototypen zu gelangen, muss ein ganzes Wandelement an einer Seite geöffnet und anschließend wieder zugeschoben werden. Dieser Mechanismus hat sich bei einer größeren Besucherzahl als untauglich erwiesen.

Beim Einsatz des Prototypen sind relativ willkürlich vier verschiedene Projektionsmaterialien getestet worden. Die besten Resultate sind bei der Projektion auf Papier und Stoff zu erzielen, doch auch bei diesen Materialien sind nicht alle Eigenschaften zufriedenstellend. Das Papier liefert ein sehr klares Bild, lässt jedoch die Lichtquelle des Beamers stark durchscheinen. Außerdem entstehen durch das Zusammensetzen der Gesamtfläche aus kleinen Einzelstücken Nähte, die lichtundurchlässig sind. Daher kann dieses Material nicht überzeugen. Der weiße Stoff lässt die Lichtquelle des Beamers ebenfalls zu stark durchscheinen und verfälscht zudem das Bild durch seine grobe Struktur. Da auch hier das Rohmaterial nicht in genügend großen Maßen zur Verfügung

stand, wird das Bild auch wieder durch Nahtansätze verfälscht.

Unbefriedigend ist auch die gewählte Spannvorrichtung des Projektionsmaterials. Beim Prototypen wird durch einen Hohlraum an beiden Seiten des Projektionsmaterials ein Metallhohlstab durchgeschoben (auf Abbildung II.1 blau markiert) und die Stäbe werden durch eine Spannvorrichtung untereinander gespannt. Diese Konstruktion hat sich als sehr umständlich und wenig praxistauglich erwiesen.

Zusammenfassend sind bei der definitiven Konstruktion also folgende Verbesserungen erforderlich: größere Stabilität, ein verbesserter Zugang, ein neues Projektionsmaterial und eine andere Aufhängung des Projektionsmaterials.

Für den endgültigen Cave sind die Abmessungen des Prototypen (2,66 m x 2,00 m) übernommen worden. Auch der diagonale Zuschnitt der Rahmenbalken zur Vermeidung von Schattenwurf hat sich als funktional erwiesen und ist im Endprodukt zu finden. Die Dicke der vertikalen Ständer ist zur Verbesserung der Stabilität auf zwölf Zentimeter erhöht worden.

Zur Verbesserung der Mobilität sind die Verbindungen der einzelnen Rahmenstücke unter Verwendung von Holzdübeln steck- und schraubbar gestaltet. Der Zeitaufwand für den Auf- und Abbau des Caves ist dadurch gegenüber dem Zeit-

aufwand für den Auf- und Abbau beim Prototypen nur unwesentlich höher, Transport und Lagerung sind nun jedoch wesentlich einfacher zu bewerkstelligen. Nachdem im Plenum nachträglich auch eine Dachprojektionsfläche beschlossen wurde, um eine verbesserte Illusion einer virtuellen Welt zu erzielen, hat sich die Konstruktion wie in Abbildung II.2 ergeben. Durch die diversen Anschrägungen sind teilweise recht komplexe Zuschnittmuster entstanden (siehe Abbildung II.3).

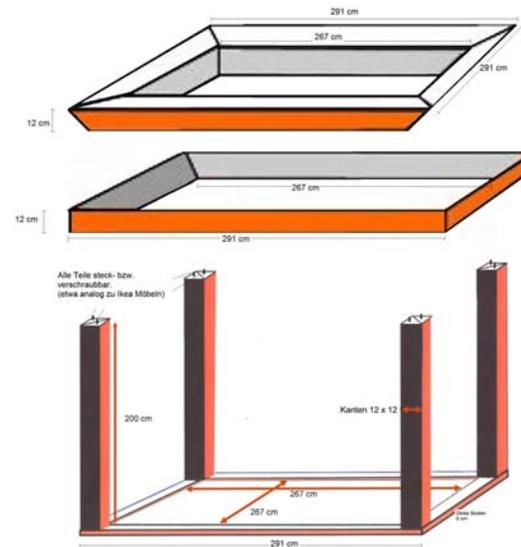


Abbildung II.2: Die Abmessungen des Cave-Rahmens.

Um eine möglichst kostengünstige Herstellung zu ermöglichen, ist die Konstruktion mit der Tischlerei des universitätseigenen Bauhofes abgestimmt worden. Von dort ist auch eine Zusage für den Zuschnitt der erforderlichen Holzteile gekommen. Leider hat dieser Betrieb jedoch sechs Monate später seine Zusage unvermittelt wieder zurückgezogen, sodass aus Zeitmangel eine externe Tischlerei beauftragt werden musste. Die kalkulierten Kosten haben sich dadurch verdreifacht.

## 1.2 Das Projektionsmaterial

Bei den Überlegungen zu einem geeigneten Projektionsmaterial ist Segeltuch die erste Idee gewesen. Dies war nach Meinung der Arbeitsgruppe nahtfrei in der geeigneten Größe zu bekommen sei. Nach den Anfragen bei einigen Segelmachereien stellte sich jedoch heraus, dass Segeltuch nicht breit genug produziert wird, die maximal verfügbare Breite beträgt 1,20 m. Auch Segeltuch hätte mit einer Überlappung von 1 cm zusammengenäht werden müssen, mit einem für ein Projektionsmaterial sicherlich unbefriedigenden Endergebnis.

Durch Zufall und persönliche Bekanntschaft sind wir auf Messebauunternehmen aufmerksam geworden, bei denen auch häufig großflächige Projektionen erforderlich sind. Diese speziellen Projektionsfolien bestehen aus Kunststoff und weisen eine Licht-

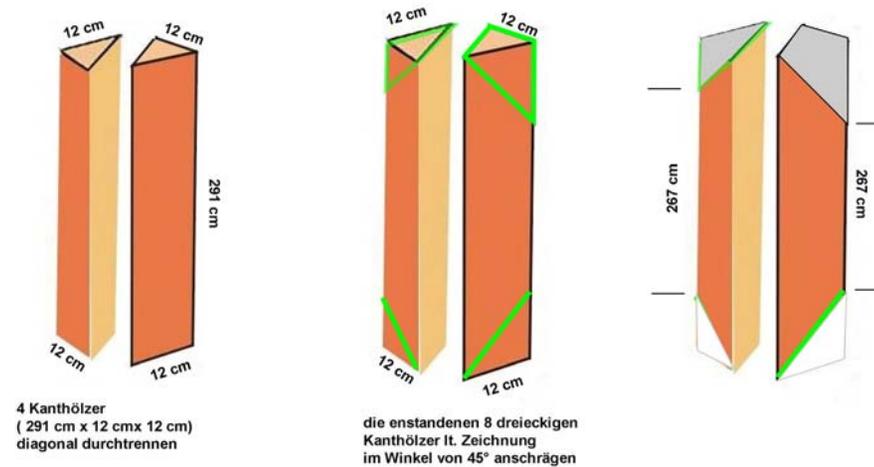


Abbildung II.3: Die Bestandteile des Rahmens für das Dach des Caves.

durchlässigkeit von 90% auf. Eine einzelne Folie wird maximal 1,20 m breit geliefert und die einzelnen Bahnen werden dann mit einem Spezialklebstoff verschweißt. Nachdem eine Probe für Tests bestellt worden war, zeigte sich, dass die Schweißnaht bei der Projektion praktisch unsichtbar blieb. Zudem hat sich durch persönliche Beziehungen die Möglichkeit ergeben, das Material günstiger zu beziehen. Der Einsatz dieser Folie als Projektionsfläche wurde beschlossen.

Zur Aufhängung sind an den Rändern der zugeschnittenen Projektionsfolie im Abstand von etwa 10 cm Metallösen eingietet. An entsprechender Stelle der Rahmenbalken sind Metallstifte an-

gebracht. Daran wird die Folie dann mit Gummibändern aufgehängt. An der Bodenseite ist keine besondere Befestigung erforderlich, da die Folie durch die Befestigung oben und an den Seiten völlig ruhig hängt. Insgesamt sind etwa 600 Gummibänder zu spannen.

Der Cave kann durch eine Seitenwand betreten werden. An dieser Seitenwand wird die Folie nicht vollständig an der Seite fixiert, stattdessen wird unten an der Folie ein Gewicht angebracht. Dadurch hängt die Folie glatt und ruhig herunter und kann zum Betreten oder Verlassen des Caves jederzeit seitwärts geklappt werden.

### 1.3 Überlegungen zu einer Dachprojektionsfläche

Um die Dachfläche von 2,67 m x 2,67 m auszu-leuchten, ist ein Abstand zwischen Projektor und Projektionsfläche von etwa fünf Metern erforderlich, bei einer Höhe des Caves von etwa zwei Metern. Um diesen Abstand zu verkürzen, muss eine Spiegelkonstruktion entwickelt werden, die die Projektion mehrfach umlenkt. Um zu einem akzeptablen Abstand des Beamers vom Cave von maximal fünf Metern zu kommen (bei einer vertretbaren Gesamthöhe von etwa sechs Metern für den Cave inklusive der Spiegelhöhe), ist ein Spiegel in den Abmessungen von ca. 2,70 m x 3,70 m erforderlich.

Ein Spiegel von 2,70 m x 3,70 m ist jedoch als Glasspiegel schon wegen des Gewichtes nicht zu

realisieren. Als Alternative bietet die Firma Seco Sign GmbH in München auf Polyurethanschaum aufgezugene Spiegelfolie an. Doch auch dort ist ein Spiegel dieser Abmessung eine Sonderanfertigung und hätte mehrere tausend Euro gekostet. In der Konsequenz aller dieser Überlegungen ist auf eine Dachprojektionsfläche aus finanziellen Gründen verzichtet worden.

Auch für die Aufstellung der LCD-Projektoren für die Seitenwände des Caves wird ein Abstand von vier bis fünf Metern benötigt (abhängig vom Öffnungswinkel des benutzten Projektors) benötigt. Für eine Präsentation des Caves ist also ein Raum von dreizehn Metern in Länge und Breite (2 x 5 m Abstand der Beamer, 3 m Breite des Caves) und mindestens 2,50 m Höhe benötigt. Auf dem Universitätsgelände bietet das Studierhaus die geeignete Umgebung.

## 2 Die Engine

### 2.1 Entscheidung für eine Grafikengine

Die Suche nach der richtigen Grafik-Engine zur Umsetzung des Caves hat sich über mehrere 3D-Echtzeit-Grafik-Engines erstreckt, die einzeln anhand der gestellten Anforderungen untersucht wurden. Überzeugt hat uns nach Abschluss des Auswahlverfahrens die Unreal Engine 2.

Durch den Einsatz dieser bereits voll entwickelten Engine bestand keine Notwendigkeit mehr, eine eigene Engine zu programmieren. Daher konnte sich von Anfang an auf die eigentliche Umsetzung der virtuellen Welten unseres Caves konzentriert werden.

Als Softwaregrundlage wird das Computerspiel Unreal Tournament 2003 benutzt. Dieses Computerspiel stellt neben dem Spiel selbst auch alle nötigen Werkzeuge zur Verfügung, um eine Modifikation (unsere Anwendung für den Cave) zu Unreal Tournament 2003 zu entwickeln. Neben dem Unreal Editor, dem Entwicklungstool zur Erstellung eigener virtueller Welten, wird auch der Sourcecode der Spieleroutinen in der zur Unreal Engine gehörenden Scriptsprache, Unreal Script, mitgeliefert.

Jeder Besitzer einer Ausgabe von Unreal Tournament 2003 ist daher eingeladen, selbst kreativ

zu werden und neue Spielmodi zu gestalten. Wird aus dem entstandenen Werk keinen Profit erzielt, ist die Nutzung und Veränderung der Spieleroutinen, also das Schreiben eines Mod, ausdrücklich erlaubt.

Neben der hohen Darstellungsqualität und der Vielzahl möglicher Effekte, die mit der Unreal Engine 2 realisiert werden können, gibt es weitere Features, die für die Wahl dieser Engine ausschlaggebend waren. Zu diesen Features gehören der 3D Sound, der es ermöglicht, live 5.1 Dolby Surround Sound abzumischen, das Vertex Animation Modell, das es erlaubt jede, beliebige Geometrie nach bestimmten Algorithmen zu animieren, sowie das Bone Character Animation Modell, mit dessen Hilfe sich Charaktere mit bis zu dreißig Knochen beliebig animieren lassen. Um die Animationsmöglichkeiten abzurunden, besteht die Möglichkeit, verschiedene Animationsbereiche festzulegen und beliebig zu kombinieren. Zum Beispiel werden der Oberkörper und der Unterkörper getrennt voneinander animiert, ergeben im Gesamtbild aber eine einzige Animation des Charakters. So kann ein Charakter rennen, während sich der Oberkörper in verschiedene Richtungen dreht.

Eine weitere wichtige Funktion ist die bereits eingebaute Karma Physik Engine, die es ermöglicht

beliebige Objekte mit physikalischen Eigenschaften zu versehen, die hinterher interaktiv in der Welt benutzbar bzw. manipulierbar sind.

## 2.2 Szenengraph der Unreal Engine 2

Die Unreal Engine 2 benutzt als Grundlage für jedes Level einen BSP-Baum (Binary Space Partitioning). Ein BSP-Baum ist eine Datenstruktur, die benutzt wird, um Objekte in einem Raum zu organisieren. Ein BSP-Baum ist eine rekursive Zerlegung des Raumes, die jedes Liniensegment (bzw. jedes Polygon in 3D) wie eine Zerlegungsebene ansieht, welche benutzt wird, um alle übrigen Objekte des Raumes als vor oder hinter dieser Ebene liegend zu kategorisieren. Wenn demnach ein neues Objekt in den Baum eingefügt wird, wird es zunächst in Relation zur Wurzel zugeordnet und dann rekursiv unter Betrachtung jedes folgenden Knotens eingeordnet.

Diese BSP-Geometrie ist ausschlaggebend, um verdeckte Polygone vor dem Rendern eines Frames automatisch zu entfernen. Jedes Polygon, das nicht gerendert werden muss, spart Rechenzeit, die bei sehr großen und komplexen 3D-Welten oft ausschlaggebend dafür ist, dass eine Simulation flüssig dargestellt werden kann.

Die BSP-Geometrie wird im Unreal Editor mit

Hilfe von Constructive Solid Geometry (CSG) erstellt. Im Gegensatz zu anderen 3D-Engines wird ein Raum vom Weltspace subtrahiert und nicht zu diesem hinzuaddiert.

Um detaillierte Objekte in den von BSP-Geometrie erschaffenen Räumen zu platzieren, wird ein externes Modellprogramm benötigt. In diesem Programm wird ein komplexes 3D-Modell erstellt und texturiert. Danach wird es in die Unreal Engine importiert und kann dann als sogenanntes Static Mesh platziert werden.

Ein Static Mesh wird auch in dem BSP-Baum kategorisiert. allerdings wird das gesamte Objekt, das oftmals aus tausenden von Polygonen besteht, als ein einziger Knoten in diesem BSP-Baum kategorisiert. Dadurch lässt sich komplexe Geometrie darstellen, ohne dass der BSP-Baum so groß wird, dass sich der Geschwindigkeitsvorteil der BSP-Geometrie in das Gegenteil umkehrt. Würde ein derart detailliertes Objekt als BSP-Geometrie kategorisiert, würde es tausende von neuen Knoten in dem BSP-Baum verursachen.

Ein weiterer Vorteil von Objekten als Static Meshes ist der geringe Speicherbedarf im Gegensatz zu geometrischen Objekten als BSP-Geometrie. Darüberhinaus sind sie beliebig oft platzierbar und werden trotzdem nur einmal in den Speicher geladen. Static Meshes sind also beliebig oft referenzierbar. Dadurch ist es möglich, sehr detail-

lierte Level zu erstellen, ohne dabei hohe Leistungseinbußen in Kauf nehmen zu müssen.

Um Geometrie zu animieren, stehen zwei Möglichkeiten zur Verfügung. Zum einen gibt es Animated Skeletal Meshes. Das sind Objekte, die sogenannte Knochen besitzen und über die Bewegung der Knochen animiert werden. An jedem Knochen sind verschiedene Vertices des Objekts angebunden, die sich mit dem Knochen mitbewegen. Auf diese Weise lässt sich sehr einfach eine Charakteranimation erstellen.

Meshes können auch mit Hilfe von Keyframe Animationen erstellt werden. Dabei wird nur in den sogenannten Keyframes das Aussehen des Objekts abgespeichert. Zwischen den einzelnen Keyframes interpoliert die 3D-Engine die Schritte, so dass eine flüssige Animation entsteht.

Für die Simulation von Außenarealen kommen der Terraineditor und die Skybox zum Einsatz. Terrain wird mit Hilfe einer Highmap erstellt. Eine Highmap ist eine Textur im Graustufenformat. Jeder einzelne Pixel der Textur steht für ein Vertex des Terrains und je heller ein Pixel ist, desto höher liegt der Vertex. Aus dieser Textur wird dann die geometrische Struktur des eigentlichen Terrains errechnet. Für die Simulation von Himmel und Sonne gibt es die Skybox. Durch diese entsteht der Eindruck eines unendlich weit entfernten Horizonts. Dabei kann die Skybox auch

geometrische Objekte beinhalten, wie zum Beispiel Häuser oder Felsen. So kann der Eindruck eines sehr großen Außenareals vermittelt werden, obwohl das eigentliche Level sehr klein und kompakt ist. Durch den Einsatz der Skybox wird der Speicherplatzbedarf verringert.

Die Beleuchtung in der Unreal Engine 2 ist etwas komplexer. Es gibt verschiedene Möglichkeiten, ein Level zu beleuchten. Zum einen gibt es die klassische Vorberechnung der Map im Editor. Dort wird der Lichtfall berechnet und in einer Lightmap gespeichert. Diese wird dann über die einzelnen Oberflächen gelegt. Lightmaps funktionieren allerdings nur auf BSP-Geometrie. Auf Static Meshes wird keine Lightmap gelegt, obwohl der Schattenfall von Static Meshes in die Lightmapberechnung mit einbezogen wird.

Durch die Berechnung des Lichtfalls im Voraus entsteht der Nachteil, dass eine dynamische Änderung der Lichtverhältnisse zur Laufzeit nicht mehr möglich ist. Abhilfe schafft das Vertex Lighting. Dadurch lassen sich Licht und Schatten auf einzelnen Objekten zur Laufzeit berechnen. Allerdings empfangen Objekte mit Vertex Lighting keine Schatten von anderen Objekten und werfen auch keine Schatten auf andere Objekte.

Um dieses Problem zu lösen, kann der Projektor verwendet werden. Mit ihm können Schatten und Lichtschein auf alle Objekte geworfen wer-

den. Der Nachteil des Projektors ist allerdings, dass seine Funktionen nicht von der Engine selbst berechnet wird, sondern von dem Designer eine sogenannte Projektive Texture erstellt werden muss, die dann der Projektor-Funktion zugewiesen wird und so den gewünschten Schattenwurf generiert.

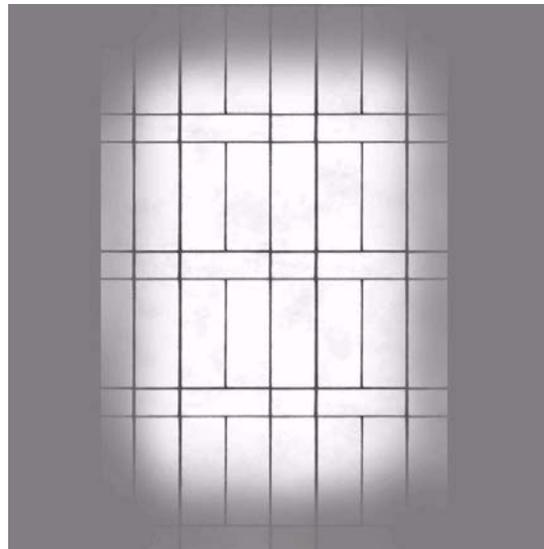


Abbildung II.4: Die Textur für den Projektor.

In Abbildung II.4 ist eine Projektive Textur zu sehen. Eine projektive Textur ist eine Graustufentextur. Der RGB Wert (127,127,127) ist der Mittelwert einer projektiven Textur. Dieser Wert markiert die Stellen, die von der Unreal Engine als unsichtbar berechnet werden sollen. Hel-

lere Pixel als dieser Mittelwert werden als Lichtschein in der Unreal Engine dargestellt, dunklere als Schatten. Anschließend wird diese Textur einem Projektor zugewiesen (siehe Abbildung II.5).

Mit dieser Technik wird beispielsweise in der Kunsthalle der Eindruck erweckt, dass die Sonne durch das Glasdach scheint.

Die Unreal Engine 2 bietet auch zahlreiche Spezialeffekte wie Partikeleffekte oder Wasseroberflächen. Da keine dieser Effekte in den virtuellen Umgebungeneingesetzt werden, wird auf eine nähere Beschreibung verzichtet.

Viele Materialeffekte, wie zum Beispiel Reflektionen oder Transparenz von Oberflächen, werden mit dem Materialeditor, der im Unreal Editor implementiert ist, erstellt. Da dieser Materialeditor sehr wichtig ist, um die verschiedenen Shader zu erzeugen, wird dieser in Folgenden genauer beschrieben.

## 2.3 Materialeditor

Um realitätsnahe, dreidimensionale Umgebungen und Gegenstände zu kreieren, werden eine hochauflösende Textur und exakt gebaute Modelle mit möglichst wenigen Oberflächen benötigt, damit bei der Darstellung der Szene keine durch ungenügende Hardwareleistung entstehenden Verzögerungen die Illusion zerstören. Unreal Tour-



Abbildung II.5: Die Platzierung des Projektors im Level.

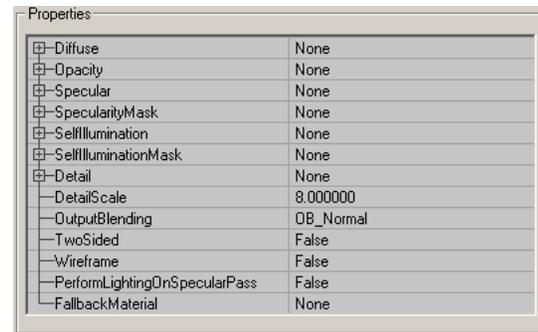


Abbildung II.6: Die Einstellungsmöglichkeiten eines Shaders.

nament 2003 bietet dem Leveldesigner eine Vielzahl an Möglichkeiten für die Texturerstellung. Dabei wurde auf eine möglichst modulare Umgebung gesetzt.

Es gibt sechs Rohmaterialklassen in Unreal Tournament 2003, wobei die erste Klasse, die Texturen (normale Texturen, mit oder ohne Alphakanal, sowie prozedurale Texturen), Ausgangspunkt für die anderen Klassen ist. Shader dienen der Effektgenerierung einer Textur. Sie sind ein wichtiges Werkzeug bei der Erzeugung realistischer Texturen.

Zunächst wird die Haupttextur (Diffuse) ausge-



Abbildung II.7: Die Löcher sind mittels einer SpecularityMask (schwarzer Kreis auf weißem Grund) erzeugt worden.

wählt, auf die der Shader angewendet werden soll. Sie dient als Grundlage für die weiteren Effekteinstellungen. Wenn die Textur durchsichtig werden oder an einigen Stellen lichtdurchlässig erscheinen soll, kann man dies durch eine weitere Textur erreichen, welche bei Opacity angegeben werden kann. Dazu werden die Informationen des Alphakanals dieser Textur genutzt.

Die Spiegeleinstellungen (Specular und SpecularityMask) erzeugen mit Hilfe von Umgebungstexturen (Environmental Maps oder TexEnvMap) den Eindruck einer Spiegelung der Umgebung auf einer Textur. Die Spiegelmaske (SpecularityMask) gibt dem Leveldesigner die Möglichkeit,

⊕ Diffuse	FireTexture'MaterialTest.Procedural.Fire'
⊕ Opacity	Texture'MaterialTest.Alpha.platecaution'
⊕ Specular	None
⊕ SpecularityMask	None
⊕ SelfIllumination	None
⊕ SelfIlluminationMask	Texture'MaterialTest.Alpha.platecaution'
⊕ Detail	None
OutputBlending	OB_Normal
TwoSided	False
Wireframe	False
⊕ FallbackMaterial	Texture'Engine.DefaultTexture'

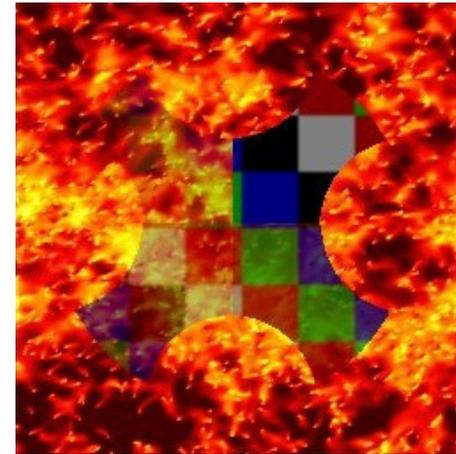


Abbildung II.8: Die Auswirkung der Opacity-Einstellungen.

nur bestimmte Bereiche reflektierend zu machen oder den Grad der Spiegelung zu bestimmen. Dies geschieht wieder über den Alphakanal.

Das Eigenleuchten (SelfIllumination) wird ähnlich wie bei der Spiegelung auch über eine zusätzliche Textur geregelt, die keine Lichtquelle für die Darstellung benötigt und somit im Dunkeln sichtbar ist. Durch die Maske wird hier ebenfalls der Bereich des Effektes sowie dessen Stärke festgelegt.

Für eine weitere Verbesserung des visuellen Eindrucks gibt es Detailtexturen, welche erst bei Annäherung an die Oberfläche sichtbar werden und so den Eindruck schaffen, dass die eigentliche

Textur sehr detailreich ist. Über die Detailskalierung (DetailScale) kann die Größe der überlagernden Detailtextur verändert werden.

Mittels des OutputBlending können verschiedene Methoden der Texturdarstellung gewählt werden. Normales OutputBlending verändert nichts an den Eigenschaften des erstellten Shaders. Wird Masked gewählt, wird die erste Farbe der Palette einer Textur unsichtbar. Existiert ein Alphakanal, werden die Bereiche mit Werten unter 128 unsichtbar.

Modulated macht Bereiche der Textur unsichtbar, die den RGB-Wert (127,127,127) besitzen, die anderen Bereiche werden halbtransparent. Translucent lässt Bereiche mit niedrigen RGB-



Abbildung II.9: Eine holzvertäfelte Wand, mit und ohne Detailtextur.



Abbildung II.10: Eine rote Lichtquelle illuminiert die Szene, der Rest ist Eigenleuchten und Maske.

Werten transparent erscheinen. Je höher die RGB-Werte, desto lichtundurchlässiger werden sie. Invisible veranlasst den Shader dazu, vollständig durchsichtig zu werden, wobei die Umgebungstextur bei eingestellter Opacity angezeigt wird.

Zweiseitig (TwoSided) sollten Texturen sein, die auf Objekte gelegt werden, dessen Aufbau es zulässt, beide Seiten einer Oberfläche zu sehen. Ansonsten wäre die Oberfläche von der einen Seite her gesehen nicht durchsichtig, wohingegen von der anderen Seite betrachtet die Oberfläche einfach nicht gezeichnet wird. Ein Beispiel hier-

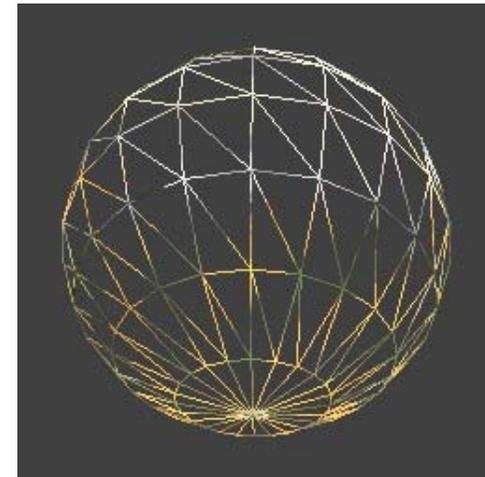


Abbildung II.11: Die Möglichkeiten der Darstellung als Drahtgittermodell im Level.

für sind teiltransparente Gitter, sowie Pflanzen, die nur aus zweidimensionalen Flächen aufgebaut sind.

Die Option Wireframe lässt die Textur nur auf dem geometrischen Netz des Objektes sichtbar zu werden, der Rest wird durchsichtig. Für einen Vergleich eines texturierten Modells und seinem Drahtgittermodell innerhalb des Levels ist dies vorteilhaft, so kann beispielsweise das Innere eines Zylinders gut verdeutlicht werden.

Die letzten Optionen bei der Shadergenerierung sind enginespezifisch und ändern am Aussehen des Shaders nichts.

Als nächste Klasse gibt es die Modifier, welche zum Verändern der Textureigenschaften dienen. Dazu gehört vor allem die für die Shader häufig gebrauchte `TexEnvMap`. Sie regelt die Berechnung der Spiegelung, entweder in Relation zur Welt oder zur Kamera.

Interessant ist auch der `ColorModifier`, welcher die Farbe einer gewählten Textur verändern kann. Verschiedene Farbnuancen einer Textur müssen nicht extra erstellt werden und sparen Texturspeicher. Der Gedanke der Performance findet sich auch in dem `ConstantColor` Modifier wieder, der ohne Texturspeicher eine farbliche Textur erzeugt. Auch beim `TexScaler` (skaliert die Textur) sowie in den für Animationen verwendbare Modifier, wie `TexOscillator` (Oszillierende Bewegung, Schrumpfen, Vergrößern einer Textur), `TexPanner` (Textur bewegt sich konstant in eine Richtung), und `TexRotator` (Textur wird um einen Punkt rotiert) wird auf die Performance geachtet. Animationen sind dadurch um einiges leichter und ohne hohe Texturkosten erstellbar, insbesondere in Kombination mit Shadern, Masken und anderen Modifiern.

Mit Hilfe der Klasse `Combiner` lassen sich Klassen über verschiedene Methoden miteinander kombinieren. Zum einen wird damit die Mischung der Farben (`Use_Color_From_Material / 2`, `Multiply`, `Add`, `Substract`, `AlphaBlend_With_Mask`, `Add_With_Mask_Modulation`) anhand ei-

ner Maske oder direkt geregelt, zum anderen wird die Erstellung des neuen Alpha-Kanals (mittels `Use_Mask`, `Multiply`, `Add`, `Use_Alpha_From_Material1 / 2`) gesteuert.

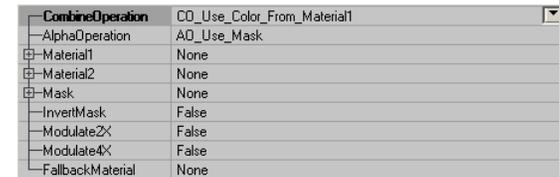


Abbildung II.12: Die Einstellungsmöglichkeiten für den Combiner.

Um Umgebungstexturen mit Leben zu füllen, wird die Klasse `Cubemap` benötigt. In einer `Cubemap` befinden sich sechs Texturen. Je nachdem, wo sich die spiegelnde Oberfläche im Level befindet, wird von ihrer Position aus eine `Cubemap` erstellt. Der Ziffernblock dient im Spiel mit den Standardeinstellungen der Einstellung der Blickrichtungen für den zu erstellenden Würfel. Das Level sollte ohne Vollbild aus dem Unreal Editor gestartet werden. Dann kann über die Konsole die Größe der Textur sowie die Art der Fortbewegung ('Ghost' schaltet die Gravitation für den Spieler aus) eingestellt werden. Werden nun das HUD, das Zielkreuz und die Waffe ausgeblendet, können die benötigten Texturschnappschüsse mittels der Taste „F9“ genommen werden. Die so erstellten Schnappschüsse werden als Texturen importiert

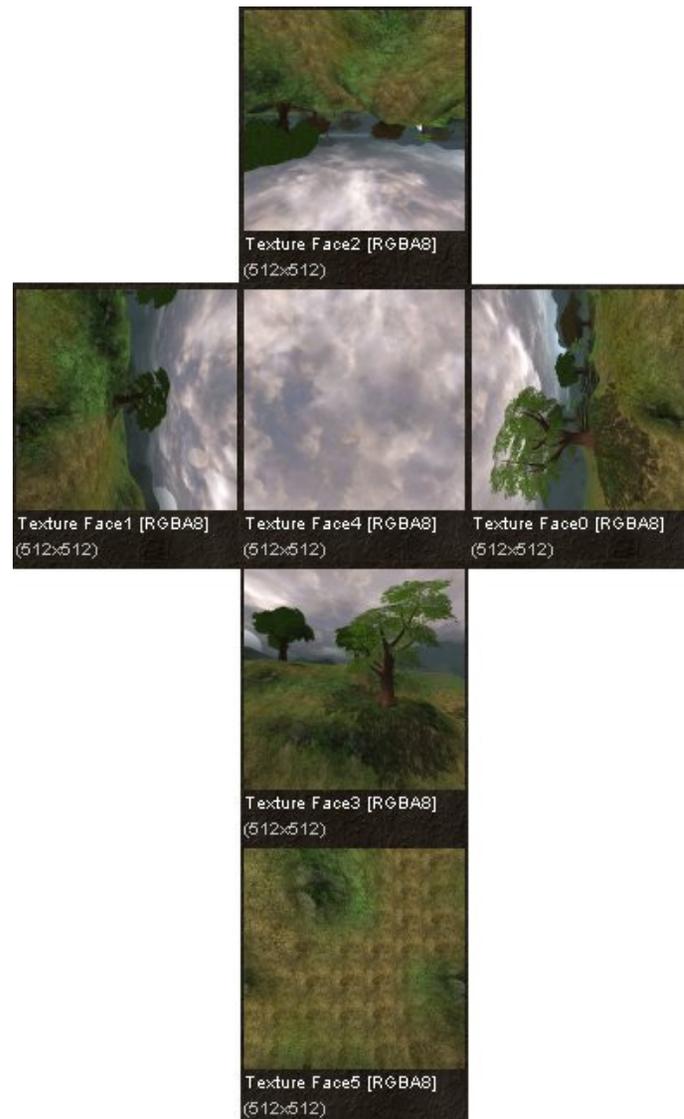


Abbildung II.13: Cubemap für eine Umgebungstextur.

und in der Cubemap angegeben (siehe Abbildung II.13).

Die letzte Rohmaterialklasse ist die Klasse FinalBlend. Hier wird die Mischmethode festgelegt, die das Material endgültig erhält.

Zu den Methoden gehören Overwrite (Die Textur wird ohne Effekte dargestellt), Modulate (Bereiche mit RGB-Werten (127,127,127) werden transparent, Werte unter 127 dunkeln sie ab, Werte über 127 hellen sie auf), AlphaBlend (dasselbe wie innerhalb der Textur bAlphaTexture auf true zu setzen, die Textur wird anhand des Alphakanals transparent), AlphaModulate\_MightNotFogCorrectly (Mischung aus Modulate und AlphaBlend, mit Problemen bei Nebelberechnungen), Translucent (Je nach RGB-Wert Transparenz ohne Alphakanal), Darken (schwärzt je nach RGB-Wert dahinterliegende Objekte), Brighten (hellt je nach RGB-Wert dahinterliegende Objekte auf) und Invisible (macht die Textur unsichtbar).

Neben der Option 'Zweiseitig' können die Eigenschaften des Alphakanals mittels eingeschaltetem AlphaTest und der Variable AlphaRef verändert werden. AlphaRef definiert die Referenzschwelle für unsichtbare Bereiche der Textur. Werte, die unter dem Referenzwert liegen, werden unsichtbar, alles darüber wird gemäß des Alphakanalwertes halbdurchsichtig bis lichtundurchlässig.

Bei Fehlern in der Berechnung wird wie bei allen Klassen auf das FallbackMaterial zurückgegriffen.

Da so ziemlich alle Klassen miteinander kombinierbar sind und der Editor durch die Explorerstruktur die Zusammensetzung eines Materials in einfachen Schritten ermöglicht, ist es dem Leveldesigner möglich, schnell die benötigten Materialien zu entwerfen. Darüber hinaus schafft der modulare Aufbau die Möglichkeit, Texturen, die auf den gleichen Quelltexturen aufbauen, durch die Änderung der Quelltextur alle auf einmal zu bearbeiten. Durch die Kombinationsvielfalt gibt es eine große Anzahl an Möglichkeiten für Performancegewinne.

Die visuelle Qualität nimmt erheblich zu, Reflektionen beispielsweise sind für eine natürliche Darstellung einer dreidimensionalen Umgebung wichtig und tragen dem Erlebnis der Virtualität bei. Einzig die Cubemaps sind noch nicht benutzerfreundlich: Die Erstellung nimmt viel Zeit in Anspruch. Die Cubemap-Erstellung könnte bei der Berechnung des fertigen Levels über im Level plazierbare Cubemap-Kameras gelöst werden. So müssten nicht bei jeder Änderung im Level erneut die Cubemaps von Hand erstellt werden. In Zukunft könnte das Problem der Spiegelung ebenfalls durch Echtzeitberechnung gelöst werden, allerdings ist die hier vorgeschlagene Methode sparsamer.

## 2.4 Programmierung der Unreal Engine

Um das verteilte Rendern der einzelnen Wände realisieren zu können, muss die eingesetzte Engine über Netzwerkfähigkeiten verfügen.

Der Netzwerkcode der Unreal Engine 2 ist für Spiele über das Internet ausgelegt und benötigt deshalb eine sehr geringe Bandbreite. Um die verteilte Berechnung für den Cave als Mod zu realisieren, wird die zu der Unreal Engine 2 gehörende Scriptsprache Unreal Script benutzt. Mit dieser Scriptsprache kann jeder Aspekt des Spiels an die jeweiligen Wünsche angepasst werden.

Die Unreal Script Sourcen werden mit dem Spiel geliefert. Unreal Script ist eine Mischung aus C++ und Java. Es ist völlig objektorientiert und bietet ähnlich wie Java eine GarbageCollection. Die Sourcen der Unreal Engine 2, die dieses Unreal Script und die API dazu zur Verfügung stellt, sind in C geschrieben und nur verfügbar, wenn eine Lizenz für die Unreal Engine erworben wird. Daher gibt es für das Projekt keine Möglichkeit, Code in der Engine direkt zu ändern. Durch die Nutzung von UnrealScript besteht daran jedoch auch kein Bedarf.

Eine genauere Betrachtung des Cubecam Mutators und des SahnGametype gewähren einen detaillierteren Einblick in UnrealSprict.

## 2.5 Unreal Mutator „Cubecams“

Mutatoren sind eine Art von Modifikation für die Unreal Engine. Sie werden wie ein Plugin benutzt und können das Verhalten der Engine ändern. Vor dem Start der Map lassen sich beliebig viele Mutatoren aktivieren. Solange ein Merkmal der Engine nicht von mehreren Mutatoren gleichzeitig verändert wird, wirken alle Effekte gleichzeitig.

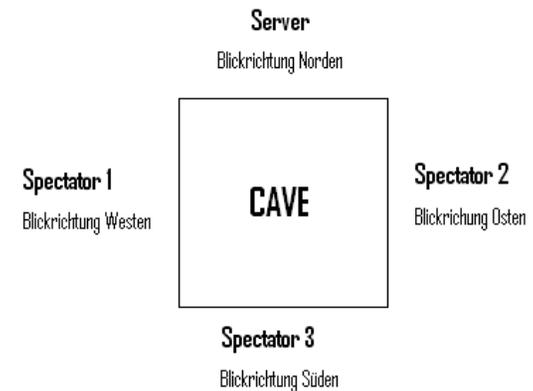


Abbildung II.14: Jedem teilnehmenden Client und dem Server lässt sich mit dem Cubecam Mutator im Spiel eine Blickrichtung zuweisen.

Mutatoren können von Benutzern in der Scriptsprache Unreal Script geschrieben und mit dem mitgelieferten Hilfsprogramm UCC.EXE compiliert werden. Unreal Tournament 2003 wird mit

einigen vorhandenen Mutatoren ausgeliefert, wie dem LoGrav Mutator, der die Gravitation der virtuellen Welt verringert. Für die Spielergemeinde sind Mutatoren ein wertvolles Werkzeug um Abwechslung in das Spiel zu bringen, für das Projekt bieten Mutatoren eine einfache Möglichkeit, um z. B. die Blickrichtung der Clients zu ändern.

Des Weiteren werden Mutatoren genutzt, um auf die Beschleunigung des virtuellen Charakters direkt zuzugreifen und so Probleme mit der Steuerung zu beheben.

Bei der Programmierung eines Mutators muss darauf geachtet werden, dass die Unreal Engine 2 objektorientiert programmiert ist und aus einer sehr großen Klassenstruktur besteht, deren Verständnis einen ebenso großen Aufwand erfordert. Jeder virtuellen Person wird in der Engine eine xPlayer-Klasse zugeordnet, die die Kontrolle ermöglicht und die Translationen und Rotationen für diese Person berechnet. Durch den Mutator wird diese Klasse mit einer neuen, eigenen Klasse ersetzt.

Nun können die Funktionen in der Klasse xPlayer überschrieben und um neue Funktionen erweitert werden. Es sind vier Vektoren eingeführt, die die jeweilige Blickrichtung repräsentieren.

```
const north    = Vect(1,0,0);  
const east     = Vect(0,1,0);
```

```
const south    = Vect(-1,0,0);  
const west     = Vect(0,-1,0);
```

Abhängig von der Konfiguration der Blickrichtung für den jeweiligen Client oder Server wird in der Funktion `PlayerCalcView()`, die bei der Berechnung jedes einzelnen Frames aufgerufen wird, mit diesen Vektoren die Blickrichtung angegeben. Der Vorteil bei dieser Lösung des Problems mit den vier Blickrichtungen ist die einfache Handhabung dieser und eine schnelle manuelle Konfiguration zur Laufzeit. So ist es in kurzer Zeit möglich, die virtuelle Welt beliebig in 90 Grad Schritten zu drehen. Um die Installation zu beschleunigen, ist von der Arbeitsgruppe festgelegt worden, dass der Server immer die Blickrichtung Norden übernimmt.

## 2.6 Unreal Gametype „SahneGametype“

Ein Gametype ist ähnlich wie ein Mutator eine Modifikation von Unreal Tournament 2003. Allerdings bietet ein Gametype sehr viel mehr Möglichkeiten als ein Mutator, denn anstatt vorhandene Klassen und Variablen zu ändern, kann hier ein komplett neuer Spieltyp erzeugt werden. Zu den vorhandenen Spieltypen gehören beispielsweise „Capture The Flag“ oder „Deathmatch“, die zwar vieles gemeinsam haben, aber sich jeweils

```

class Cubecams extends Mutator Config(User); // Mutator wird deklariert
function PostBeginPlay() { // Diese Funktion wird nach dem Start eines Spieles
    // aufgerufen und hier von uns "überschrieben"
    Level.Game.PlayerControllerClassName = "Micasa.CamController";
    // die PlayerController-Klasse wird durch unsere eigene ersetzt
}

```

Abbildung II.15: Die Erstellung eines eigenen Mutators.

durch besondere Merkmale auszeichnen. Sie lassen sich ebenso wie die Mutatoren vor dem Start der Map festlegen, wobei allerdings nur ein Gametype ausgewählt werden kann.

Für das Projekt ist ein eigener Gametype programmiert worden, um damit virtuelle Charaktere mit künstlicher Intelligenz zu versehen. Diese sollen mit dem Cavebenutzer interagieren.

Es wurden ein Charakter erstellt und erfolgreich mit einer selbsterstellten Laufanimation sowie einer Winkanimation versehen. Ein normaler Unreal Tournament 2003 Standardcharakter hat allerdings etwa 85 Animation für Gehen, Rennen, Springen, Schwimmen, Ducken, etc.

Um Charaktere während der Laufzeit über die Konsole der Engine zu erzeugen, zu löschen oder zu verändern bietet der SahneGametype folgende selbstgeschriebene Funktionen:

**SahneAddBots(int num)**

fügt num zufällige Charaktere hinzu, die von der künstlichen Intelligenz gesteuert werden.

**SahneBotChangeName(optional string botName)**

Gibt dem zuerst hinzugefügten Charakter den Namen botName.

**SahneAddBot(optional string botName)**

Fügt einen Charakter mit dem Namen botName hinzu.

**SahneBotMoveTo(int x, int y, int z)**

Lässt alle Charakter zum Punkt (x, y, z) laufen.

**t()**

Shortcut für das Hinzufügen des einzigen fertiggestellten Charakters.

Die Klasse, die sich um die Kontrolle der Charaktere kümmert und die die künstliche Intelligenz darstellt, wird in den Voreinstellung der Klasse SahneGametype festgelegt (siehe Abbildung II.16).

Ähnlich wie bei den Mutatoren werden hier

```

defaultproperties { // Beschreibt Standardwerte f"ur Klassenvariablen
    BotControllerClass=class'SahneGameType.SahneBot'
                    // Als neue BotControllerClass wird unsere eigene
                    // Klasse SahneBot zugewiesen
}

```

Abbildung II.16: Eine eigene Bot-Klasse erstellen.

vorhandene Klassen mit eigenen Klassen ausgetauscht, die veränderte oder erweiterte Funktionalitäten besitzen.

Der computergesteuerte virtuelle Charakter, auch Bot genannt, befindet sich zu jeder Zeit in einem festgelegten Zustand.

Er verlässt diesen nur, wenn ein `gotostate()`-Aufruf ihn zum Wechsel zwingt. Die hier programmierte künstliche Intelligenz gibt dem Charakter vor, so lange untätig herumzustehen, bis er den virtuellen Charakter des Cavebenutzers gesehen hat. Dann soll der Bot solange hinter dem Spieler herlaufen, bis er auf eine Distanz von etwa 156,6 UE herangekommen ist, um dann zu Winken und einen festgelegten Kommentar abzugeben.

Das Sehen eines Benutzers löst in der Engine den Event `SeePlayer()` aus, welcher dazu führt, dass der folgende Befehl ausgeführt wird.

```

// Zustandswechsel
gotostate('SahneBotSiehtPlayer');

```

Im neuen Zustand führt der Bot alle 1500 Millisekunden einen Bewegungsbefehl auf die aktuelle Position des Cavebenutzers aus, bis er sich in einer Reichweite von 200 Unrealeinheiten befindet. Danach kommt wieder ein `gotostate()`-Aufruf und der Bot wechselt in den Zustand `SahneBotBeiPlayer`. Hier werden dann die Animation und der Audioclip abgespielt. Zehn Sekunden später wechselt der Bot wieder in seinen Ursprungszustand und wartet auf ein erneutes `SeePlayerEvent`.

## 2.7 MiCasa Stick

In Unreal Tournament 2003 gibt es Objekte, die sich durch physikalische Einwirkungen beeinflussen lassen, sogenannte Karmaobjekte. Diese Karmaobjekte lassen sich jedoch nur ungerichtet beispielsweise durch Waffenfeuer bewegen. Um Objekte gerichtet bewegen zu können, ist die Idee

entstanden, eine Waffe zu modifizieren, um damit Objekte gezielt zu bewegen. Diese neue Waffe heißt MiCasa Stick.

Dazu wird die Klasse einer bestehenden Waffe überschrieben, das Tracing-Verhalten der Waffe allerdings nur unwesentlich verändert. Beim MiCasa Stick wird nur einmal am Anfang des Dauerfeuers gezielt und nicht wie bei anderen Waffen

bei jedem Schuss. Wird das Objekt gefunden, so wird es bei jeder Bewegung der Waffe bewegt, solange die Feuerfunktion aktiviert bleibt.

Um Rotationen bzw. Tiefenbewegungen zu erzeugen, ist ein weiteres Gerät (vgl. Kapitel 3.2) entwickelt worden. Dieser erfasst über verschiedene Achsen die Bewegung eines echten Arms, um diese in der virtuellen Welt zu simulieren.

## 3 Sensorik und Aktorik

### 3.1 Die Tanzmatte

Um sich in der virtuellen Welt innerhalb des Caves fortzubewegen, wurden sich verschiedene Möglichkeiten der Interaktion überlegt. So gab es die Idee, mittels eines Laufbandes die natürlichen Bewegungsabläufe beim Laufen in der realen Welt in die virtuelle Welt zu übertragen. Daraus ergaben sich bei genauerer Betrachtung allerdings Probleme.

Ein Laufband bietet z. B. nur die Möglichkeit nach vorne und nach hinten zu laufen und somit hätte die virtuelle Welt wahrscheinlich so auf die Wände projiziert werden müssen, dass die Blickrichtung änderbar ist damit man auch nach links oder rechts laufen kann. Dies ist für den Betrachter nicht wirklich von Vorteil, da der Eindruck entstand, rotierende Wände wären für die Orientierung und das Wohlbefinden des Besuchers nicht zuträglich. Die Idee, das Laufband auf einen Drehteller zu stellen, würde aus Sicherheitsgründen nur eine Person innerhalb des Caves gestatten. Dieses Problem warf auch die Idee von der Bewegung einer großen Kugel um. Der Akteur befände sich auf einem Gestell, welches er zum Abstützen oder Festhalten nutzen kann. Dieses Gestell befände sich seinerseits über einer großen Kugel, die er mit Laufbewegungen auf der Stel-

le in Rotation versetzen könnte. Diese Rotation würde ähnlich eines Trackballs in Bewegung in der virtuellen Welt umgesetzt. Ein zusätzlicher Haken an dieser Idee war der enorme Aufwand und der fehlende Platz (entweder würde ein tiefes Loch im Boden benötigt, oder aber ein recht hoher Raum, wobei der Cave aufgehängt oder aufwendig mit Gerüsten auf die Blickhöhe des Akteurs gebracht werden müsste).

Schlussendlich wurde sich für die Lösung der Tanzmatte entschieden, die zwar eine etwas andere Art der Bewegung (örtlich begrenzt) für den Besucher darstellte, allerdings viele Vorteile, wie z. B. niedrige Baukosten und hohe Kompatibilität dank der technischen Basis auf einem Numpad, mit sich brachte.

#### 3.1.1 Von der Idee zum Konzept

Nachdem die Entscheidung, eine Tanzmatte als Eingabegerät zu bauen, gefallen war, wurden zunächst bestehende kommerzielle Lösungen untersucht. Zu dieser Zeit gab es noch keine große Auswahl an Tanzmatten und die folgenden Extreme waren die grundlegenden Konzepte der Tanzmattenkonstruktion.

Taugliche Tanzmatten mussten erst für viel Geld besorgt werden. Da eine möglichst kostengüns-



Abbildung II.17: Elastische PVC-Tanzmatte. Günstig, aber eher unpraktisch.

tige Lösung angestrebt wurde, fiel die Entscheidung leicht, die Tanzmatte selbst zu bauen. Trotz des Sparaspekts bei der Findung des Designkonzepts sollte die Tanzmatte eine stabile und robuste Bauweise aufweisen können. Vollmetallgehäuse waren zu teuer, wie an den Preisen im Baumarkt schnell zu erkennen war. Darüber hinaus wäre die Konstruktion viel zu schwer geworden, und somit nicht mehr leicht transportabel. Leichtere Materialien mit gleicher Festigkeit, zum Beispiel Aluminium, sind dementsprechend noch teurer.

*“Den eingefleischten Freaks sollten diese Tanzmatten jedoch auch nicht ausreichen. Die wabblige Oberfläche der Plastikteppiche hielt den geballten Steps der Freaks nicht aus, rutschte und*



Abbildung II.18: Vollmetalltanzmatte. Robust, aber zu teuer (ca. 1850 \$).

*faltete sich recht unangenehm zusammen. Entsprechende Hardware aus Holz und Metall gab es nicht. Die handwerklich geschickten Tänzer waren jetzt gefragt, die bereits existierenden Tanzmatten in eine Hardcase Variante umzubauen.“*

(Quelle: <http://www.giga.de/>  
4.12.2002 18:07 Uhr)

Angeregt von dieser Idee auf der Internetseite von NBC-Giga, einer bekannten täglichen Computersendung, auf der eine Alternative zur käuflichen Tanzmatte der Firma Sony dargestellt wurde, wurde der Plan gefasst, eine Holzkonstruktion zu bauen. Diese Holzkonstruktion sollte neun bewegliche Kontaktfelder besitzen, die die Gehbewegung innerhalb des Caves simulieren sollten.



Abbildung II.19: GIGA-Tanzmattenlayout für die PlayStation.

Auf der Sony PlayStation gibt es Spiele, wie Dance Dance Revolution, die als Eingabemedium eine Tanzmatte vorsehen. Aus der Bauanleitung, die man bei [www.giga.de](http://www.giga.de) herunterladen konnte, konnten viele Anregungen für den Bau der eigenen Matte bezogen werden.

### 3.1.2 Der Aufbau der Tanzmatte

Die Tanzmatte besteht aus einer Rahmenkonstruktion aus Holz. Die Abmessungen betragen circa 109 x 109 Zentimeter. 1 cm breite Querstreben unterteilen den Rahmen in neun Teilfelder.



Abbildung II.20: Alle Felder der Tanzmatte sind bereit und die LEDs leuchten rot.

Jedes Teilfeld hat am Rand abgesetzte, 1 cm breite Trägerleisten auf denen Gummipuffer (gesponsert von der Firma Viborg, Verden) angebracht waren; es handelt sich hierbei um alte, ausgediente Gummiaufhängungen von Auspuffanlagen.

Diese Gummipuffer bewerkstelligen die Rückfederung nach Betätigung einer ruckplatte. Es wurde auch über eine Lösung mit Federn nachgedacht. Beim Kauf der Federn für die Kontakte wurde jedoch festgestellt, dass diese Option zu kostspielig werden würde.

Den Abschluss der Konstruktion bilden exakt zugeschnittene Holzquadrate, die mit Metallplatten

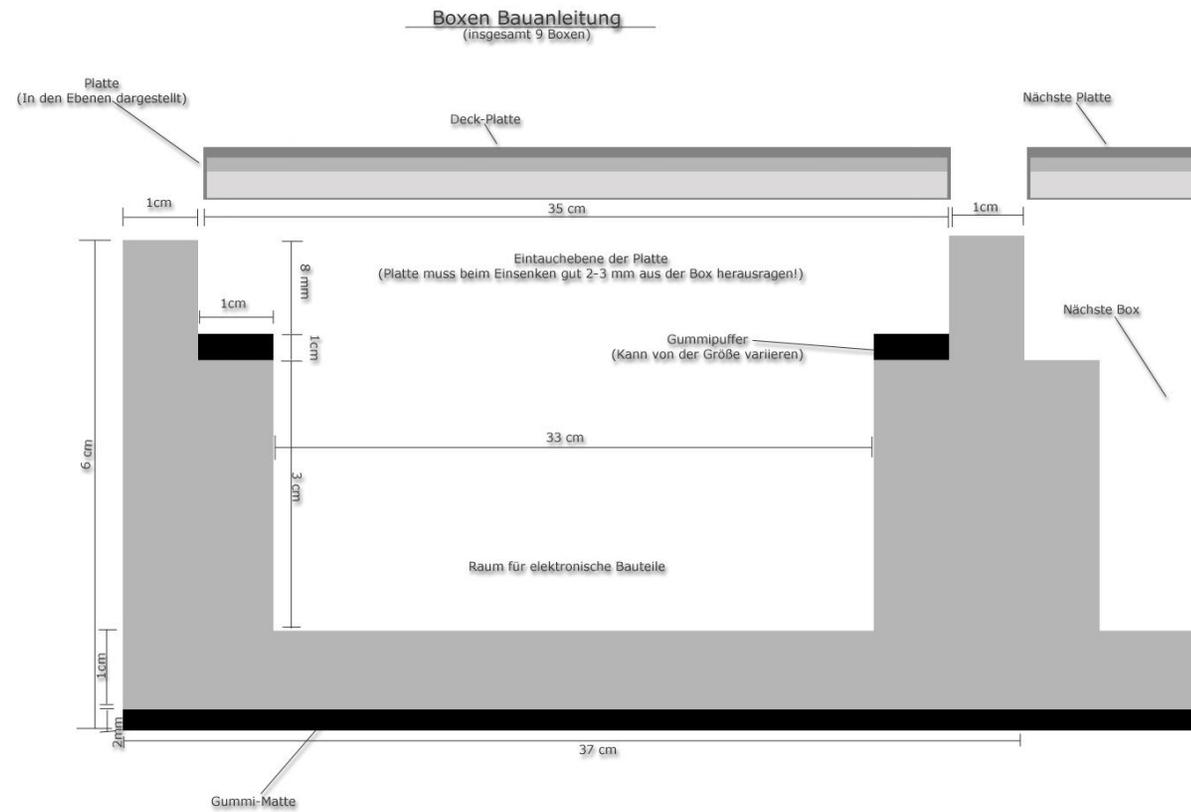


Abbildung II.21: Querschnitt durch eine Box der Tanzmatte.



Abbildung II.22: Anbindung der Tanzmatte an den PC per Tastaturkontroller.

(als Trittschutz) abgedeckt wurden. Durch die Gummipufferung ist jede einzelne Teilplatte in einem Schwankungsbereich von zwei Millimetern gegen die Grundkonstruktion beweglich.

### 3.1.3 Die Elektronik

Damit bei einer Belastung einer Teilplatte ein Effekt ausgelöst wird, wurden in jedem Teilstück vier Kontaktfedern angebracht, die bei Bewegung der Deckplatte zwei separate Stromkreise schließen. Der erste Kontaktpuls wird über einen Tastaturkontroller zum PC gesendet, mit dem zweiten Stromkreis werden die Leuchtdioden der jeweiligen Platte geschaltet.



Abbildung II.23: Feedback durch LED-Umschaltung.

Der Tastaturkontroller stammt aus einer defekten Tastatur. Er war so montiert, dass eine Reihe von Kontakten an die Tastaturfolie drückte und so eine große Anzahl von Stromkreisen (für die einzelnen Tasten) mit diesem Kontroller verbunden waren. Um jetzt genau die Kontakte zu finden, die benötigt wurden, wurde ein Tastaturtestprogramm benutzt.

Die Ergebnisse wurden notiert und danach die Kabel für die Tasten an den Kontroller gelötet. Da dies eine etwas wackelige Sache war, wurde der Tastaturkontroller innerhalb der Tanzmatte mit Heißkleber verklebt. Ein Spannungsregler half, die benötigten 5 Volt für die Relais bereitzustellen. Die Abbildung X.Y zeigt den elektrischen Stromkreis, der die LEDs schaltet.

Die zweifarbigen LEDs werden über das Relais geschaltet, im deaktivierten Zustand fließt der Strom durch die rote LED. Wird nun das Relais mittels der zweiten Kontaktfläche und somit des Transistors (und Widerstand) durch Kontakt mit einer der beiden für die Relaisaktivierung verantwortlichen Federn aktiviert, fließt der Strom durch die grüne LED. Der Nutzer bekommt ein Feedback, ob und welches Feld geschaltet ist. Außerdem hört man das Umschalten des Relais durch einen kurzen Klick.

Der Einsatz des Tastaturkontrollers war die einfachste Schnittstellenlösung, die 18 Informationsangaben der neun Felder (neun Mal geschaltet oder nicht geschaltet) direkt nutzbar machte. Die Aufteilung der Matte wurde vom PC analog zum Nummernfeld der Tastatur interpretiert, sodass als Eingabealphabet die Zahlen 1 bis 9 zur Verfügung standen. Tastatureingaben sind für Unreal eine Selbstverständlichkeit, und über die User.ini wurden die Bewegungen den entsprechenden Tasten zugeteilt.

```
NumPad1=MoveForward | MoveSouthWest
NumPad2=MoveForward | MoveSouth
NumPad3=MoveForward | MoveSouthEast
NumPad4=MoveForward | MoveWest
NumPad5=Walking
NumPad6=MoveForward | MoveEast
NumPad7=MoveForward | MoveNorthWest
NumPad8=MoveForward | MoveNorth
NumPad9=MoveForward | MoveNorthEast
```

MoveNorth, -West, -South, -East sind Funktionen aus dem CamController.uc Script, welches wiederum folgendes definiert:

```
exec function MoveNorth() {
    MoveVector = north;
}
exec function MoveNorthEast() {
    MoveVector = northeast;
}
exec function MoveEast() {
    MoveVector = east;
}
[...]
```

Diese Vektoren (die zuvor ebenfalls in diesem Script als Konstante definiert wurden, so zum Beispiel (1,0,0) für 'north') legen die Bewegungsrichtung fest. MoveForward bewegt den Spieler dann in diese in MoveVector festgelegte Richtung.

### 3.1.4 Entwicklung und Verbesserung des Eingabegeräts

In einer ersten Bauversion wurden, anstelle der Federn, mit Alufolie umwickelte Styroporplättchen verwendet. Bei Belastungstests ergab sich, dass die Elastizität der Plättchen unzureichend war. So wurde der Kontakt bei Belastung der

Deckplatte zwar beim ersten Mal zuverlässig getätigt, bei wiederholter Belastung war die Schaltung aber unzuverlässig. Im Übrigen hat sich das Gerät bewährt, die Bedienung ist für den Benutzer sehr intuitiv, die einzelnen Schaltungsvorgänge erfolgen zuverlässig. Die Federn waren eine gute Investition, und wenn mehr Geld vorhanden gewesen wäre, hätten die Gummipuffer ebenfalls durch Federn ersetzt werden können.

Die auf die Holzplatten geklebten Metallplatten schützen die Holzplatten vor Beschädigungen, jedoch waren die Kanten sehr scharf. Nach ein paar kleineren Schnittwunden wurden die Kanten kurzerhand abgeschliffen.

Die so abgerundeten Kanten konnten nun ohne Risiko angefasst werden. Der feine Metallstaub ist bis heute allerdings mangels Staubsauger in der Tanzmatte verblieben.

### 3.1.5 Materialliste

- Massive Grundplatte: Holzplatte 109x109x1
- Bau für Rahmen, innen und außen: 6x Langhölzer, a 107x1,0x5,8 sowie 2x Langhölzer, a 109x1,0x5,8
- Innenboxen (9 Stück): 2x Kurzholz, a 35x33x3 und 2x Kurzholz, a 33x33x3
- Pads (9 Stück): Eisenplatte 34,8x34,8x0,1 (gesponsert von Herrn Fischer); Holzplatte 34,8x34,8x5
- Säulenelemente für Kontakte (36 Stück): Holzsäule 2,7x2,7x2,7
- Gummipuffer (56 Stück) (gesponsert von der Firma Viborg, Verden)
- Alufolie
- Leitungen aus Netzkabeln (gesponsert von Herrn Fischer)
- Lüsterklemmen
- 36 Stahlfedern
- 36 Unterlegscheiben für die Federn
- Schleifutensilien
- 1 Unmenge an Schrauben, Heißkleber, 2-Komponentenkleber und Geduld
- 9 5V Relais
- 9 Transistoren
- 9 Widerstände 100 kOhm
- viele 470 Ohm-Widerstände
- 48 Dual - LEDs, rot/grün, matt, 3mm Durchmesser
- Netzteil (9 V / 1 A)

## 3.2 Der A.R.M.

### 3.2.1 Motivation

Für das Erleben einer vermischten Realität in einem Cave reicht es nicht aus, den Benutzer mit einer virtuellen Welt zu umgeben. Es muss die Möglichkeit zum Agieren geben. Andernfalls besteht die Gefahr, dass man zu einem passiven Beobachter wird und die Grenze zwischen realer und virtueller Welt bestehen bleibt.

Möchte man nun beide Realitäten ineinander übergehen lassen, ist es entscheidend, dem Benutzer die Möglichkeit zu geben, Einfluss auf das virtuelle Geschehen um ihn herum zu nehmen. Er muss in der Lage sein, virtuelle Objekte zu manipulieren und mit ihnen zu interagieren.

Voraussetzung hierfür ist, dass der Benutzer mit seinem ganzen Körper (oder zumindest Teilen davon) und nicht nur mit seinen audiovisuellen Sinnen in die Cave-Anwendung integriert wird. Beispielsweise könnte der Benutzer die Position von Objekten im virtuellen Raum verändern. Für den Benutzer bedeutet dies, dass er die virtuelle Welt um sich herum ein Stück weit selbst gestalten kann. Diese Interaktion soll durch Bewegungsabläufe aus dem realen Leben ausgeführt werden können. Im Sinne einer Mixed Reality sollte sich die Kommunikation mit der virtuellen Umwelt nicht von der Kommunikation mit der realen

Umwelt unterscheiden. Für den Benutzer bedeutet dies, dass er intuitiv mit der Cave-Umgebung interagieren kann.

Um diese Vorstellung von Interaktion in einer Cave-Umgebung umzusetzen, wird ein Real-Time Motion Capturing-System benutzt. Unter geringstmöglichem Kostenaufwand und unter Verwendung vorhandener Komponenten wurde ein System entwickelt, das die intuitive Interaktion mit der Mixed-Reality-Cave-Umgebung realisiert.

### 3.2.2 Der Entwurf

Bei den Interaktionen des Menschen mit realen Gegenständen spielen die Arme meist eine zentrale Rolle. Deshalb sollte das Motion Capturing-System die Bewegung der Arme in die Cave-Anwendung integrieren und so die intuitive Interaktion mit virtuellen Objekten realisieren.

Somit stellte sich die Schwierigkeit, wie die realen Bewegungen in Form von geeigneten Informationen erfasst und in der Cave-Anwendung auf einen virtuellen Arm abbildet werden konnten.

Beobachtet man die Bewegungsabläufe beim Greifen eines Objektes, stellt man fest, dass neben den Muskeln die Hand-, Ellenbogen- und Schultergelenke die Hauptakteure sind. Durch die Ausrichtung dieser Gelenke kann man relativ zum

Körper des Benutzers die Position eines gegriffenen Objektes im Raum bestimmen.

Es stand also fest, dass die Gelenke des menschlichen Armes eine Schlüsselrolle spielten und eine virtuelle Entsprechung dieser Gelenke entwickelt werden musste. Um die Stellung der Gelenke am realen Arm zu erfassen, war eine Vorrichtung nötig, die während der Bewegung des realen Arms die Winkel zwischen Unterarm und Oberarm und zwischen Oberarm und Oberkörper erfassen konnte. Mittels dieser Messwerte würde dann die Position eines Gegenstandes in der Hand des Benutzers errechnet werden. Die Idee des A.R.M (A Real-Time Motion Capturing Device) war geboren.

Zur Repräsentation der Gelenke sind Potentiometer, die eine gewisse Ähnlichkeit mit den menschlichen Gelenken haben, am besten geeignet. Dabei handelt es sich um Drehwiderstände, die je nach Stellung mehr oder weniger Strom fließen lassen und so interpretierbare Signale erzeugen. Es gibt keinen Moment, in dem kein Strom durch einen Potentiometer fließt; es wird immer ein Signal geliefert.

Zu finden sind Potentiometer beispielsweise in Joysticks oder in Stereoanlagen, wo sie zum Einstellen der Lautstärke dienen. Ein Joystick besitzt meist einen oder mehrere Feuerknöpfe, besteht aber hauptsächlich aus einem Steuerknüppel, der

auf zwei Achsen in die vier Himmelsrichtungen bewegt werden kann. Die Rolle der Potentiometer besteht darin, die Stellung des Joysticks an der X- bzw. Y-Achse zu messen. Diese Achsenbewegung könnte in eine Bewegung eines Punktes in einer X/Y-Ebene abgebildet werden.



Abbildung II.24: Ein Potentiometer.

Nach dem feststand, wie die Gelenke des menschlichen Armes nachgebildet werden sollten, musste der Rest des A.R.M. realisiert werden. Die Kon-

struktions musste so ausgelegt sein, dass sich der Kopf des Potentiometer und seine Drehachse gegeneinander drehen ließen.

Zunächst wurde ein Ellenbogengelenk konstruiert, an dem der Unterarm und der Oberarm jeweils durch Aluminiumrohre mit geringem Durchmesser repräsentiert wurden. Ein Potentiometer wurde dazu benutzt, eine Verbindung zwischen den beiden Rohren herzustellen. Durch Scherenbewegungen konnte der Winkel zwischen den beiden Rohren verändert und von dem Potentiometer gemessen werden.

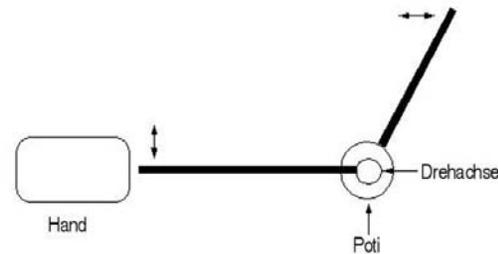


Abbildung II.25: Entwurf eines Ellenbogengelenks.

War diese einfache Konstruktion mit Klettverschlüssen am Arm eines Benutzers befestigt, ließen sich die Bewegungen des Ellenbogengelenks bereits gut auslesen und weiterverwerten. Dennoch wurde schnell klar, dass sich die anderen Gelenke des realen Arms nicht so einfach realisieren ließen. Ein Ellenbogengelenk mochte sich mit

seinem einem Freiheitsgrad relativ einfach simulieren lassen, jedoch besitzt das Schultergelenk, ebenso wie das Handgelenk, zwei Freiheitsgrade. Dies bedeutet, dass hierfür zwei Potentiometer in einem Gelenk nötig sind.

Die Überlegungen zu einer erweiterten Konstruktion führten zu einer Art Exoskelett, einem vereinfachten Nachbau der Unter- und Oberarmknochen. Diese Vorrichtung kann am Rücken des Benutzers befestigt werden, der über einen Haltegriff, der am künstlichen Unterarm angebracht ist, das Exoskelett bei seinen Armbewegungen mitführen kann.

Um die Freiheitsgrade realer Gelenke nachahmen zu können, wurden künstliche Gelenke entwickelt, die ein Achsenkreuz besaßen. An jeder dieser Achsen wurde jeweils ein Potentiometer angebracht. Die Achsen werden jede für sich mit einem Metallbügel umschlossen. Zwei dieser Gelenke mit einem Aluminiumrohr verbunden bildeten den vereinfachten Nachbau des Ober- und Unterarms eines Menschen.

Im folgenden Abschnitt wird dieser Entwurf genauer skizziert und es wird erläutert wie durch eine Designstudie die Machbarkeit des Vorhabens geprüft wurde.

### 3.2.3 Visualisierung des Eingabegerätes

Zwar gab es eine klare Vorstellung von der Funktionsweise des A.R.M., jedoch Unsicherheiten, ob sich die Vorstellungen umsetzen ließen ohne einen entsprechenden Prototypen zum Testen zu entwickeln. Die Konstruktion eines solchen Prototypen wäre sehr aufwendig gewesen und bei einer Fehlkonstruktion wäre für Design und Herstellung des endgültigen Gerätes wahrscheinlich nicht mehr ausreichend Zeit übrig geblieben.

Um dieses Dilemma zumindest teilweise zu umgehen, wurde eine Designstudie des A.R.M. am Computer erstellt. Dazu wurde zunächst ein vereinfachtes 3D-Modell des Eingabegerätes modelliert. Anschließend entstand mit Unterstützung einer Animationssoftware eine Kinematik, mit der die Eigenschaften des A.R.M. bezüglich der Freiheitsgrade des Modells dargestellt werden konnten.

Zuerst wurden die Gelenke konstruiert, die für das Eingabegerät verwendet werden sollten. Daraus ließen sich die Freiheitsgrade der einzelnen Gelenke ableiten. Diese groben Werte konnten dann in der Kinemattkette verwendet werden. In Abbildung II.26 zeigen die von den einzelnen Gliedern der Kette abstehenden Markierungen die eingegebenen Beschränkungen an. Nach Abschluss dieser Arbeit wurde ein sogenanntes Bi-

ped erstellt; eine Gliederpuppe, die die Anatomie des menschlichen Bewegungsapparates nachempfunden. Nun konnte der Entwurf des A.R.M. an der Gliederpuppe getestet werden. Hierzu wurde die Kinemattkette des Eingabegerätes so mit dem Biped befestigt, wie später das echte Gerät vom Benutzer getragen werden sollte. Der Anfangspunkt der Kinemattkette des virtuellen A.R.M. wurde auf der Rückseite des Torsos des Biped befestigt und das Ende am der Hand des Biped (siehe Abbildung II.27).



Abbildung II.26: Kette mit dem virtuellen Modell des A.R.M.

Sowohl die Kinematik des Eingabegerätes, als auch die des Biped ist invers. Inverse Kinematik bedeutet, dass die Kettenmanipulation und das Verhalten der einzelnen Kettenglieder vom Blatt (in diesem Fall: das Ende am Handgelenk des Biped) zum Stamm (die Wurzel der Kinemattkette auf dem Rücken des Biped) berechnet wird.

Mit diesem virtuellen Versuchsaufbau war es möglich, das Verhalten des A.R.M.-Entwurfs am Biped bei den unterschiedlichsten Bewegungen zu studieren, ohne einen funktionsfähigen Prototypen zu bauen.

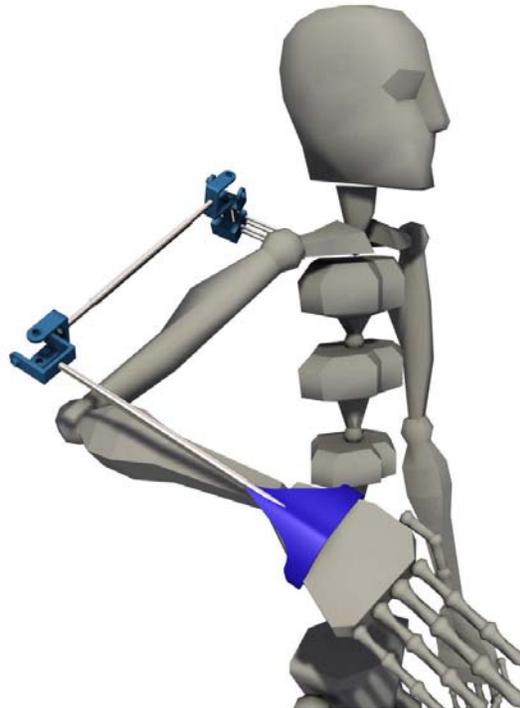


Abbildung II.27: Der virtuelle A.R.M. am Biped.

Dabei stellte sich heraus, dass die Anordnung der Achsen im Ellenbogengelenk für das Erfassen der Bewegung eines menschlichen Arms ungeeignet

war. Das Konzept für das Ellenbogengelenk wurde überarbeitet und damit hatte sich der Schritt der Visualisierung des A.R.M.-Entwurfs bereits gelohnt.

### 3.2.4 Anbindung und Konstruktion

Bevor mit der Konstruktion des A.R.M. begonnen werden konnte, mussten Möglichkeiten der Kommunikation zwischen dem A.R.M. und der Cave-Anwendung untersucht werden. Konzeptionell sollte der A.R.M. wie ein Joystick am Computer angeschlossen werden.

Um letztlich mit der Cave-Anwendung zu kommunizieren, musste zunächst die Anbindung des A.R.M. an das Betriebssystem (hier Windows XP) erfolgen. Ab Windows 95 wird die Unterstützung für zahlreiche verschiedene Joystick-Modelle bereits vom Betriebssystem bereitgestellt. Damit nun eine Kommunikation zwischen dem A.R.M. und dem Betriebssystem ermöglicht werden konnte, mussten die Bedingungen, unter denen ein Joystick als solches erkannt wird, untersucht werden. Dazu soll hier zunächst auf die Schnittstelle eingegangen werden, über die sich solche Eingabemöglichkeiten an einen PC anschließen lassen.

Diese Schnittstelle ist der sogenannte Gameport. Diese Schnittstelle ist in ihrer ursprünglichen

Form eigens dafür vorgesehen, Joysticks zur Benutzung in Computerspielen freizugeben. Spätere Versionen waren dahingehend erweitert, dass auch die Kommunikation mit MIDI-Geräten, wie zum Beispiel elektronische Klaviaturen, unterstützt werden konnte.

Ein Gameport besteht aus 15 Pins, die zur Signalerzeugung zum Teil Strom abgeben und auch aufnehmen können. Die folgende Tabelle liefert eine genaue Übersicht der Pin-Belegung eines Standard-Gameports.

Pin Nr.	Belegung / Funktion
1	5 V
2	Joystick 1, Feuerknopf 1
3	Joystick 1, X-Achse
4	GND
5	GND
6	Joystick 1, Y-Achse
7	Joystick 1, Feuerknopf 1
8	5 V
9	5 V
10	Joystick 2, Feuerknopf 1
11	Joystick 2, X-Achse
12	GND
13	Joystick 2, Y-Achse
14	Joystick 2, Feuerknopf 2
15	5 V

An den Gameport können vier Achsen (Pins 3, 6, 11, 13) und vier Feuerknöpfe (Pins 2, 7, 10,

14) angeschlossen werden. Die restlichen Pins dienen zur Stromversorgung (5 V) und zur Erdung (GND). Es könnten also zwei Joysticks über einen Gameport angeschlossen werden. Joysticks mit hohem Funktionsumfang nutzen alle zur Verfügung stehenden Anschlüsse aus und belegen den Gameport somit alleine.



Abbildung II.28: Ein Standard-Gameport.

Damit ein Joystick als solcher vom Betriebssystem erkannt wird, müssen mindestens zwei Potentiometer mit den Pins für das jeweilige Achsenpaar verbunden werden, da hier zur Signalerzeugung ein Dauerstrom fließt. Wie oben bereits erwähnt, fließt durch einen Potentiometer immer Strom und so liefert er auch immer ein Signal. Verbindet man zwei Potentiometer mit den entsprechenden Pins am Gameport, so werden die Signale vom Betriebssystem erkannt und als Joystick interpretiert. Wird kein Signal geliefert, so ist (zumindest aus Sicht des Betriebssystems) kein Joystick angeschlossen.

Für das Konzept des A.R.M. bedeutete dies, dass zwei Gelenke an einen Gameport angeschlossen

werden konnten. Die Entscheidung fiel auf das Schulter- und das Ellenbogengelenk, da diese zentral für die Erfassung der Bewegung des menschlichen Arms sind. Das Schaltbild (siehe Abbildung II.30) beschreibt, wie die vier Potentiometer der beiden Gelenke mit dem Gameport verbunden sind.

Zwei Potentiometer teilen sich hierbei eine Stromversorgung und liefern dann am Eingang für die jeweilige Achse ein entsprechendes Signal. Diese Schaltung kann von Windows auf zwei unterschiedliche Weisen interpretiert werden. Zum einen kann es sich dabei um zwei Joysticks handeln, die sich einen Gameport teilen. Zum anderen kann es sich für das Betriebssystem um einen Joystick handeln, der über zusätzliche Funktionen verfügt (beispielsweise eine Ruderfunktion und eine Schubkontrolle). Diese Variationen können über die Windows Joystick-Konfiguration frei gewählt werden.

Für den A.R.M. spielt es keine Rolle, wie die angeschlossenen Potentiometer interpretiert werden. Es ist jedoch von Bedeutung für die Weiterverarbeitung der gelieferten Daten, wie sich später zeigen sollte.

Im nächsten Schritt musste nun das Auslesen der Informationen der Potentiometer realisiert werden. Es sollte eine Möglichkeit entstehen, die Hardware anzusprechen.

Das „Microsoft Windows application programming interface (API)“ bietet die Möglichkeit, Applikationen für Windows zu schreiben und vorhandene Konstrukte in eigene Programme zu verwenden. Unter anderem besteht die Möglichkeit, Informationen aus einem von Windows erkannten Joystick auszulesen. So konnte die Kommunikation zwischen der Hardware und dem Betriebssystem relativ einfach realisiert werden.

```

*** JOY-O-MAT-X ***
PROGRAM FOR CHECKING UP TO 16 CONNECTED
JOYSTICKS UNDER WINDOWSxp.
-----
DEVICE ID: 0 FOUND...
PointOfView: 65535
-----
X VALUE : 32767
Y VALUE : 511
Z VALUE : 58773
RUDDER : 4351
-----
BUTTON 1: 1
BUTTON 2: 0
BUTTON 3: 1
BUTTON 4: 0
-----
PRESS ANY KEY TO EXIT

```

Abbildung II.29: Screenshot Joy-O-Mat.

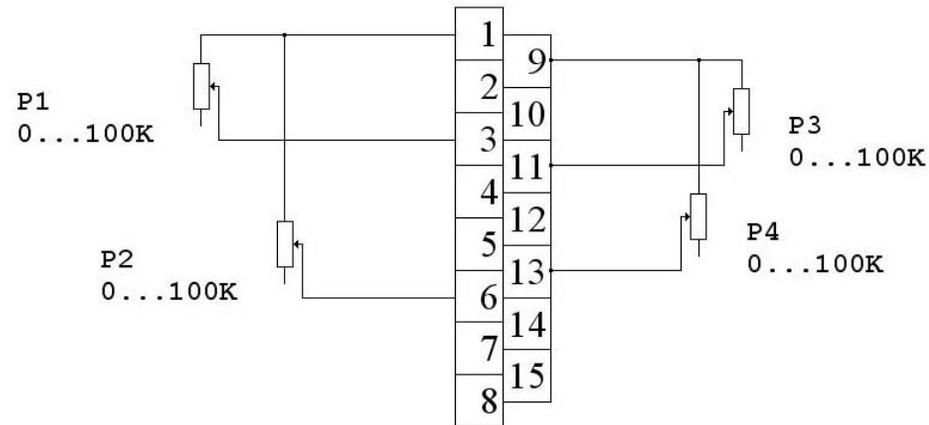


Abbildung II.30: A.R.M. Schaltbild.

Es wurde ein Programm entwickelt, das korrekt angeschlossene Joysticks bzw. Potentiometer und Feuerknöpfe auslesen konnte. Für „Joy-O-Mat-X“ stellte sich der A.R.M. als Joystick mit Ruder- und Schubfunktion dar, damit alle Informationen, die über den A.R.M. ermittelt werden können, kompakt ausgegeben werden können.

„Joy-O-Mat-X“ liefert hierbei Informationen über den Status der vier Achsen und der ersten vier Knöpfe eines jeden erkannten Joysticks (bis zu 16 können vom Betriebssystem erkannt werden). Die Variablen X VALUE und Y VALUE geben hierbei Auskunft über die Stellung des X- und des Y-Potentiometer. Die Variablen Z VALUE (müsste korrekter Weise eigentlich Throttle heißen) und RUDDER repräsentieren den Status der Schub- und

der Ruderkontrolle, die beim A.R.M. Informationen über das zweite Gelenk liefern.

Jede der Variablen nimmt dabei einen Wert von 0 bis 65535 für die Achsen und ein Wert zwischen 0 und 1 für die Feuerknöpfe an. So konnten in Echtzeit Informationen über die Ausrichtung der einzelnen Gelenke ausgelesen werden.

Es wurde mit der Konstruktion des A.R.M. begonnen, da die Kommunikation zwischen dem Betriebssystem und der Hardware hergestellt war.

Der A.R.M. sollte leicht und robust sein. Neben den Potentiometern, Kabeln und Steckern für die Erfassung der Gelenksstellungen sind die Gelenke selbst und Knochen des A.R.M. aus Aluminium gebaut. Abbildung II.33 zeigt welche Teile insgesamt verbaut sind.

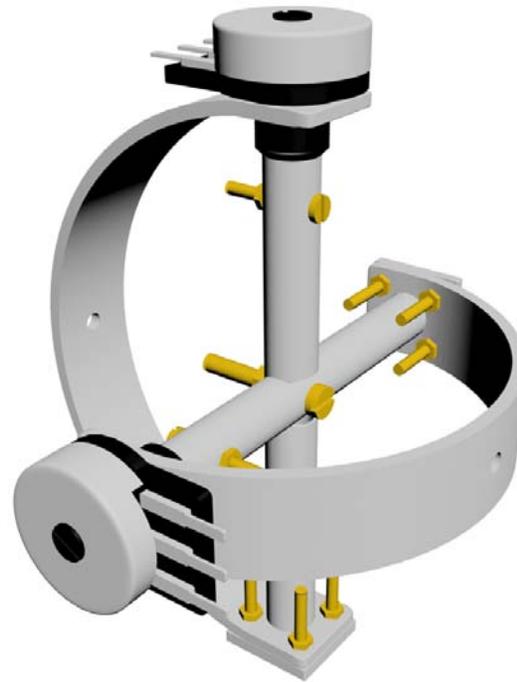


Abbildung II.31: Schematische Darstellung eines Gelenks.

Da die beiden Gelenkhälften, wie oben beschrieben, durch ein Achsenkreuz verbunden werden sollten, mussten sie so konstruiert sein, dass sie sich nicht gegenseitig behinderten. Würde man die Gelenke gleich groß bauen, so führte es zu einem eingeschränkten Bewegungsspielraum. Eine der Gelenkhälften ist somit kleiner als die andere.

Die Gelenke sind miteinander verstrebt, sodass sie, ähnlich wie beim menschlichen Arm, durch Knochen verbunden sind. Um zu verhindern, dass die Gelenke durch zu harte Bewegungen vom Benutzer auseinanderbrechen, sind die Knochen des künstlichen Ober- und Unterarms so konzipiert, dass sie sich der Länge nach ausdehnen konnten. Diese Konstruktion besteht aus drei Aluminiumrohren, wobei zwei Rohre mit kleinerem Durchmesser jeweils in ein Ende des Größeren geschoben wird. Die kleineren Alurohre sind mit einem Gummizug verbunden, der durch das Innere des äußeren Alurohrs verläuft (siehe Abbildung II.32).

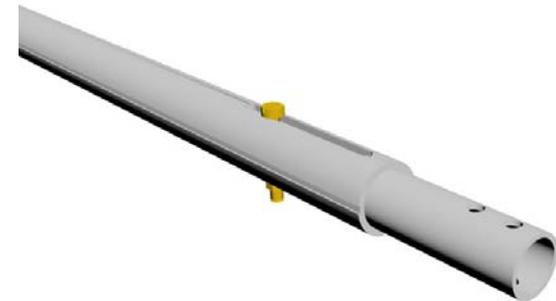


Abbildung II.32: Armknochen.

Wird der A.R.M. durch den Benutzer gestreckt, so fängt diese Konstruktion einen Teil der Kraft ab, die sonst auf die Gelenke gewirkt und so im schlimmsten Fall zu Beschädigungen geführt hätte.

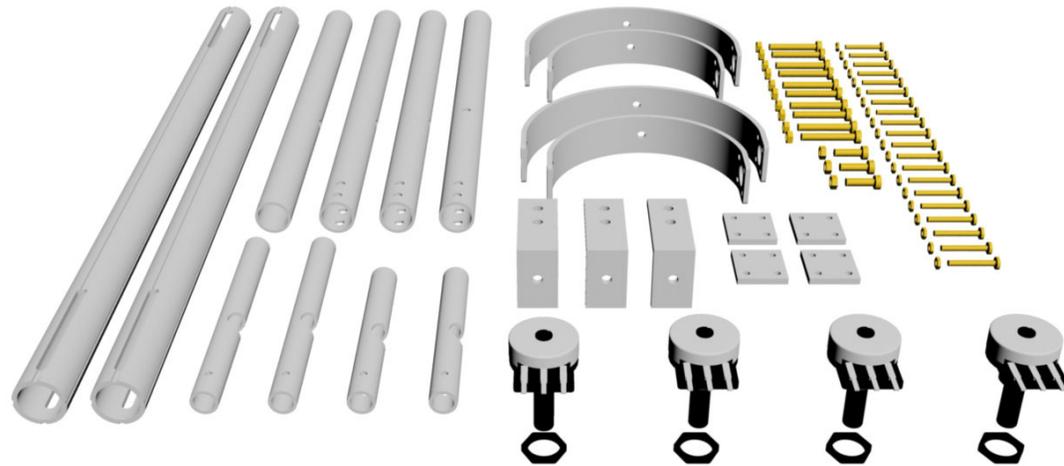


Abbildung II.33: Verwendete Materialien für den Bau des A.R.M.

te. Abbildung II.34 zeigt den fertig zusammengesetzten A.R.M.

Damit der A.R.M. vom Benutzer komfortabel getragen werden kann, wird eine Koppel zur Befestigung benutzt, die sich wie ein Rucksack anlegen lässt. An dieser Koppel ist ein Ausleger befestigt, an dem der A.R.M. mit etwas Abstand zum Benutzer angebracht wird. Dieser Abstand kann variiert werden, um unterschiedliche Schulterbreiten auszugleichen. Der Haltegriff besteht aus einem kleinen Holzstab und einem Aluminiumbügel.

### 3.2.5 Die Software

Die rohen Werte, die der angeschlossene A.R.M. liefert, sind eigentlich unbrauchbar. Schließlich sollte ein Punkt im Raum relativ zum Standpunkt des Benutzers berechnet werden. Die Werte vom A.R.M. mussten also noch interpretiert und umgerechnet werden.

Der Hardwaretreiber liefert für jeden Potentiometer Werte im Bereich von 0 bis 65535 der natürlichen Zahlen, im Folgenden als  $S$  bezeichnet. Da der Wertebereich des Potentiometers nahezu linear ist, kann eine einfache Abbildungsfunktion



Abbildung II.34: Schematische Darstellung des A.R.M.

verwendet werden. Der Bereich, in den abgebildet werden soll, wird im Folgenden mit  $D$  bezeichnet. Er liegt für unsere Abbildung im Bereich von 0 bis 360, also entsprechend den Gradangaben für die Auslenkung des jeweiligen Gelenks unseres Arms.

Die allgemeine Form der Formel für die Abbildung lautet wie folgt. Die Variablen  $h$  und  $l$  stellen die obere und untere Grenze des Wertebereichs in den abgebildet werden soll dar,  $x_{max}$  den maximalen Wert des Wertebereichs aus dem abgebildet werden soll.

$$\exists x_{max} \in S \ni x_{max} \geq x_n \in S$$

$$\exists h \in D \ni h \geq h_n \in D$$

$$\exists l \in D \ni l \geq l_n \in D$$

$$x \in S$$

$$f(x) = \frac{(h-l)x}{x_{max}} + l$$

Mit dieser Abbildung ist es möglich, die Ausrichtung jedes Gelenkes zu bestimmen. An jedem Gelenk ist eine Verstrebung montiert, die einer Translation entspricht. Das zweite Gelenk, also das Ellenbogengelenk, ist am Ende der Verstrebung des ersten Gelenkes, dem Schultergelenk, befestigt. Der Endpunkt der zweiten Verstrebung ist an der Hand fixiert und bildet damit den Punkt, der berechnet werden soll (siehe Abbildung II.35).

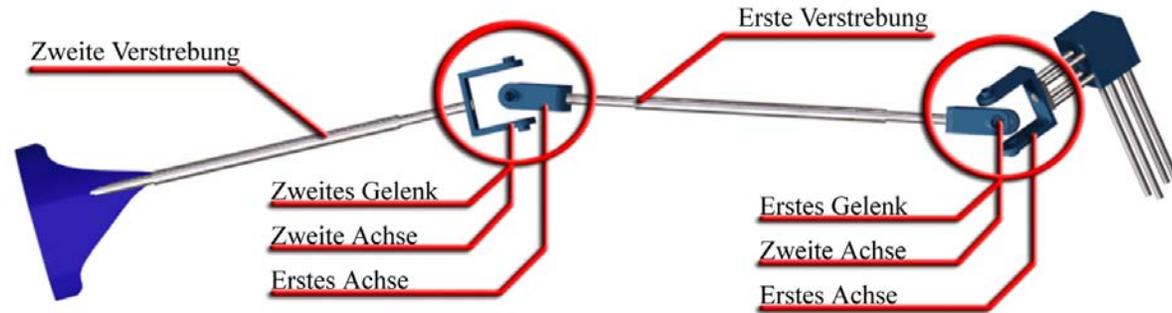


Abbildung II.35: Achsen der Gelenke.

Zur Vereinfachung der Berechnung wird davon ausgegangen, dass der Abstand zwischen den Gelenken konstant bleibt. Um aus den gemessenen Werten eine Position relativ zur Position der Schulter zu errechnen, werden Transformationsmatrizen und homogene Koordinaten benutzt. Alle nötigen Transformationen können so über Matrizenmultiplikationen effizient realisiert werden.

Im Folgenden sind die Matrizen für die Rotation und Translation angegeben. Der Vollständigkeit halber ist auch die Rotationsmatrize für die Rotation um die Z-Achse angegeben.

Rotation um die X-Achse um den Winkel Alpha:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation um die X-Achse um den Winkel Alpha:

$$\begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation um die X-Achse um den Winkel Alpha:

$$\begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Translation um den Vector [x y z]:

$$\begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Durch das Multiplizieren aller Rotations- und Translationsmatrizen entsteht die vollständige

Transformationsmatrize. Aus dieser Matrix kann die relative Position der Hand zur Schulter extrahiert werden. Um einige Probleme bei der Berechnung auszuschließen, wurde beim Multiplizieren das Inverse der Matrizen benutzt. Als Folge steht die zu berechnende Position nicht in der letzten Spalte, sondern in der letzten Zeile.

Ursprungsmatrize:

$$\begin{bmatrix} 11 & 12 & 13 & \underbrace{14} \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \\ 41 & 42 & 43 & \underbrace{44} \end{bmatrix}$$

Transformierte Matrize:

$$\begin{bmatrix} 11 & 21 & 31 & 41 \\ 12 & 22 & 32 & 42 \\ 13 & 23 & 33 & 43 \\ \{14 & 24 & 34 & 44\} \end{bmatrix}$$

### 3.2.6 Fazit

An der Konstruktion des A.R.M. gibt es noch manches zu verbessern. Beispielsweise muss an den Dimensionen der Konstruktion noch gearbeitet werden, denn für manchen Benutzer war der A.R.M. einfach zu groß.

Dennoch ist ein funktionierender Prototyp eines Real-Time Motion Capturing-Systems entstanden, das bei sehr geringem Kostenaufwand die intuitive Kommunikation zwischen einem Benutzer und einer Mixed Reality Anwendung realisiert (vgl. Kapitel 6.2.3).

### 3.3 MiCasa Stick und Laserpointer

#### 3.3.1 Anbindung der Webcams und des A.R.M. an die Engine

Die Daten der Eingabegeräte müssen zur Weiterverarbeitung an die Engine gesendet werden. Hierzu musste zunächst eine Möglichkeit gesucht werden, um Daten in die Engine einzuspeisen. Dies ist bei einer Spiele-Engine nicht so einfach, da sie nicht so konzipiert ist, dass sie Daten anderer Eingabegeräte als Maus, Tastatur oder Joystick verarbeiten kann.

Jedoch kann man die UDP-Schnittstelle, die eigentlich zur Konfiguration der Spieleinstellungen durch den Administrator gedacht ist, erweitern. Diese Funktionalität befindet sich in der Klasse UDPBeaconEx und überprüft in jedem Durchlauf, ob Pakete zur Bearbeitung vorliegen. Diese werden dann nach bestimmten Schlagwörtern durchsucht und entsprechende nachfolgende Daten werden so aufbereitet, dass sie als Variablen in der Engine verwendet werden können.

Im Folgenden ist der Inhalt eines extern gesendeten Paketes an die Spiele-Engine dargestellt:

```
POS100,200
```

Die verarbeitende Klasse findet das Schlagwort POS und erkennt somit, dass es sich um Daten handelt, die die Waffe im Spiel neu ausrichten sol-

len. Die beiden Werte 100 und 200 werden nun am Komma getrennt und die Werte können von einer anderen Klasse (cubecam) genutzt werden um die Rotation der Waffenhand auf 100 in x- und 200 in y-Richtung zu setzen.

Genauso funktioniert auch die Kommunikation mit der Software, die die Armbewegungen empfängt.

#### 3.3.2 Vom Zauberstab zum Laserpointer

Als die Idee zum Einsatz eines Zauberstabes zur Eingabe von Befehlen in der Cave-Umgebung aufkam, musste ein Weg gefunden werden, diesen mit der Spiele-Engine zu verbinden. Dies sollte jedoch ohne jeden Einsatz von Leitungen geschehen, denn wer sah jemals einen Zauberer durch eine Fantasiewelt gehen, der seinen Zauberstab an eine sehr lange Kabeltrommel anschließen muss, um Beschwörungen machen zu können?

Eine erste Idee war, dies durch eine im artec vorhandene Sensorik zu verwirklichen, die auf Funkpeilung basiert. Allerdings wurde dies schnell wieder verworfen, da der Sender eine Leitungsverbindung benötigt, um dessen Lage im Raum zu übertragen. Eine Ausführung mit leitungslosem Sender war zu teuer, sodass auch die eventuelle Anschaffung eines solchen Senders nicht in Frage kam.

So musste eine Selbstbau-Lösung entwickelt werden, die das Tracking eines Zauberstabes in einem Raum ermöglicht.

Es wurde beschlossen, dieses durch den Einsatz von Webcams zu realisieren. Die Webcams erkennen die Zauberstabposition und -lage unter Zuhilfenahme einer Bilderkennungssoftware.

Hierzu sollte der Zauberstab mit Infrarotdioden ausgestattet werden, die in einem bestimmten Muster am Stab befestigt werden sollten, um dann eine eindeutige Lageerkennung zu gewährleisten.

### 3.3.3 Bilderkennung I

Als Vorbild zur Lageerkennung diente das AR-Toolkit, welches bestimmte Muster auf einer Musterkarte erkennen kann, einem bestimmten Objekt zuweist und dieses schließlich mit OpenGL auf das Realbild projiziert.

Das AR-Toolkit wurde jedoch nicht auf die gewünschten Bedürfnisse angepasst, da es sehr komplex aufgebaut ist und sehr viele Ressourcen verbraucht. Eine geeignete Anwendung muss sehr schlank und schnell gehalten werden, sodass sie nicht zu stark mit der Cave-Engine um die Prozessorleistung konkurriert. So wird bei der aktuellen Version der Bilderkennung eine von Microsoft frei zum Download zur Verfügung gestellte Software-Bibliothek genutzt, die VisSDK.

Diese Bibliothek stellt eine einfache Schnittstelle zur Ansteuerung von Webcams zur Verfügung, mit der die Bilder mit wenig Aufwand direkt von der Kamera in ein Programm geladen und dort weiterverarbeitet werden können. Die Bilddaten müssen dann nur noch nach dem gesuchten Objekt durchsucht werden und entsprechend über UDP an die Engine weitergegeben werden.

### 3.3.4 Webcamposition

Zunächst sollte der Zauberstab als Objekt im Cave erkannt werden. Da sich der Anwender des Stabes jedoch frei im Raum drehen können sollte, musste der gesamte Raum durch mehrere Kameras abgedeckt werden, die dabei in den Cave hineinsehen, so dass sich die Zahl der Kameras nun auf vier erhöhte. Die Berechnung der Position ist durch die perspektivische Verzerrung bei der Nutzung einer Kamera bereits enorm hoch. Deshalb musste der Gesamtaufbau noch einmal überdacht werden.

Der Zauberstab wurde durch einen Laserpointer ersetzt, mit dem auf die Cavewände gezielt werden muss. Die Kameras wurden nun außerhalb des Caves positioniert, um den vom Laserpointer auf der Cavewand projizierten Laserpunkt zu erkennen. Dadurch wird die vorher ausserordentlich komplizierte Bilderkennung sehr vereinfacht, da nun nur noch der Punkt auf einer zweidimen-

sionalen Leinwand gefunden werden musste. Dies konnte mit einer einfachen Funktion ermöglicht werden, so dass sich nur noch Probleme der korrekten Ausrichtung der Kameras zur Leinwand ergaben und damit der korrekte Wert der Punktposition ermittelt werden konnte.

### 3.3.5 Bilderkennung II

Die Punktposition konnte nun einfach in eine Rotation der Waffenhand umgesetzt werden. Jede Kamera nimmt Bilder von einer der vier Leinwände auf. Damit deckt sie in der Horizontalen einen 90 Grad großen Winkel im Raum ab. Die Punktposition in x-Richtung wird also einfach auf 90 Grad skaliert und je nach Projektionswand mit

einem Ausgleichswert addiert, um die Rotation an die Engine zu übergeben.

Zur Berechnung der y-Rotation wird die gleiche Funktionalität benutzt, hier wird die Position jedoch auf den vertikalen Öffnungswinkel des von der Engine berechneten Bildes skaliert.

### 3.3.6 Polarisisation

Die Kameras und Beamer wurden mit Polarisationsfiltern, die senkrecht zueinander ausgerichtet wurden, ausgerüstet. Dadurch konnte das von der Engine berechnete und auf die Leinwände projizierte Bild herausgefiltert werden. Somit können die Kameras dieses Bild nicht sehen und nicht an der Punktfindung gehindert.

## 4 Umgebungen und Weltenbau

### 4.1 Das Mehrzweckhochhaus

#### 4.1.1 Die Ebene 1 des MZH

Die Idee, eine originalgetreue Nachbildung einer Ebene des Mehrzweckhochhauses (MZH) der Universität Bremen zu erstellen, entstand bei der ersten Zielfindung am 15. November 2002. Erst beinhaltete die Idee nicht die Ebene 1 des MZH, sondern eine fiktive Ebene 9, in der bei der Gestaltung der Fantasie freien Lauf gelassen werden sollte. Der gedankliche Schritt und die Entscheidung, einen Teil des Gebäudes nachzubauen, war damit bereits getan.

Auf dem ersten Projektwochenende in Nienburg fing eine Teilgruppe des Projekts mit einer ersten Umsetzung an. Zu Beginn handelte es sich um eine Art Übung, bei der die Aufgabe darin bestand, virtuelle Umgebungen für die Cave-Anwendung zu bauen. Verschiedene Programme zur Erstellung von Spiele-Maps wurden ausprobiert, um das für unser Projekt am besten geeignete zu ermitteln.

Auf dem ersten Projektwochenende in Nienburg (Januar 2003) legten einige Mitglieder des Projekts den Grundstein für die Ebene 1 des MZH, allerdings noch für die Madness Engine.

Zuerst wurde mit GtkRadiant (<http://www.q3radiant.com/>) für die Quake III Engine begonnen. Es entstand ein erstes grobes Gerüst für die Ebene 1. Später, nachdem die Entscheidung gefallen war, auf die Unreal Engine umzusteigen, wurde nur noch der im Lieferumfang enthaltene Unreal Editor benutzt. Auch die finale Version der Ebene 1, die auf dem Projekttag am 09. Juli 2004 präsentiert wurde, ist mit dem Unreal Editor entstanden.

Ursprünglich wurde vom artec die Madness Engine als Grundlage für alle Anwendungen im Cave vorgeschlagen. Diese Engine wurde von Martin Faust, einem Mitglied der Forschungsgruppe, entwickelt und orientiert sich an der Quake III Engine. Mit der Quake III Engine als Grundlage wurden erste Versuche zum Bau einer virtuellen Umgebung unternommen.

Als Map-Editor diente „GtkRadiant“, ein Map-Editor für Quake III und andere Spiele, die auf dieser Engine basieren. Mit diesem Editor wurde an einem Abend eine allererste Version der Ebene 1 erstellt. Dabei standen keine Maße, keine Skizzen, keine Modellierungsfotos zur Verfügung - lediglich Erinnerungen und ein bisschen Fantasie.

Es wurde eine Grundlage zum Experimentieren mit dem Editor und anderen Tools benötigt. Im

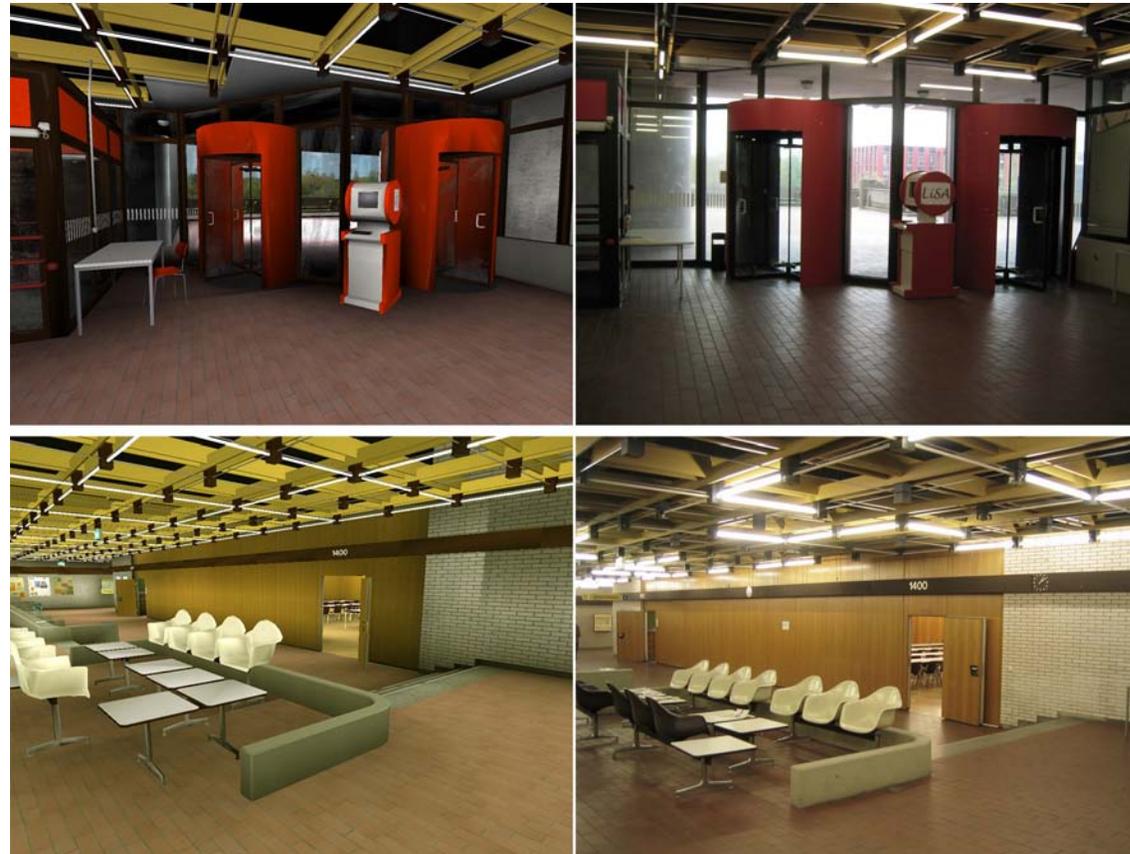


Abbildung II.36: Das virtuelle MZH und sein reales Vorbild.

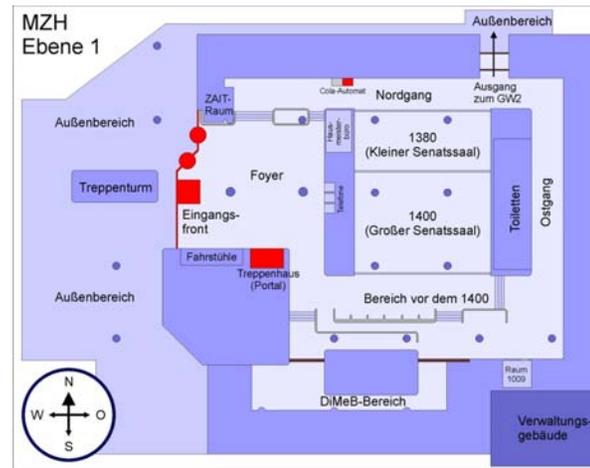


Abbildung II.37: Skizze der Ebene 1 des MZH.

Laufe des ersten Abends und des kommenden Tages entstand so eine rudimentäre Nachbildung der Ebene 1. Die Karte umfasste den großen Senatsaal (MZH 1400), den Bereich davor mit den Bänken, sowie einige Gangabschnitte zum Foyer, zum DiMeB-Bereich und zu den Toiletten.

Es war faszinierend, dass fast alle Projektmitglieder nach einem kurzen Blick zur Beamerleinwand sofort wussten, was modelliert wurde, auch wenn sie bis dahin noch nicht direkt daran mitgewirkt hatten. Die Ebene 1 besaß also selbst in groben Formen einen hohen Wiedererkennungswert. Die Arbeit an der Karte verlief in den kommenden Wochen eher zögerlich, da ziemlich bald nach dem Projektwochenende Zweifel aufkamen, ob die

Quake III Engine überhaupt verwendet werden sollte. Das hinderte aber nicht daran, Texturen zu sammeln und erste Objekte zu modellieren, und sei es auch nur als Übung.

Gegen Ende des Semesters wurde sich gegen die Quake III und für die Unreal Engine entschieden. Da die Quake-Karte nicht in ein Unreal2-Format konvertiert werden konnte, wurde von Neuem mit dem Bau der Ebene 1, diesmal für die Unreal Engine, begonnen.

Zwar steckte bereits viel Arbeit in der Quake III-Karte, jedoch wurde sie mit der Zeit mehr und mehr zu einer Experimentierumgebung, die sich von einer Cave-Anwendung immer weiter entfernte. Zudem war sie teilweise ausschließlich aus der

Erinnerung entstanden und folgte keinem einheitlichen Maßstab, was das Einfügen neuer Elemente sehr schwierig machte.

Die Arbeit an der ersten Version der MZH-Map für Unreal Tournament 2003 begann im Laufe des zweiten Projektsemesters (Sommersemester 2003). Recht bald stand auch der erste Meilenstein fest: eine öffentliche Präsentation des Caves Ende Oktober 2003.

Als Vorlage für den virtuellen Nachbau der Ebene 1 wurde sich bereits im ersten Projektsemester um Baupläne des MZH bemüht. Diese Bemühungen scheiterten jedoch an der Haustechnik des MZH, welche die Pläne wahlweise nicht herausgeben konnte, durfte oder wollte.

Eine Alternative war allerdings bald gefunden. Ein Kommilitone besorgte ein Set Feuerwehrpläne des MZH in digitaler Form (siehe Abbildung II.38). Diese Feuerwehrpläne lieferten zwar den Grundriss, ähnlich wie die ersehnten Baupläne, es fehlten jedoch jegliche Längenmaße. Es waren lediglich Flächenmaße für die Räume eingetragen.

Ein weiteres Problem war die Qualität der Feuerwehrpläne, deren Mängel allerdings erst mit der Zeit entdeckt wurden. Teilweise fehlten auf den Plänen Wände oder gleich komplette Räume und diverse Türen waren an anderer Stelle eingezeichnet, als sie in der realen Ebene 1 zu finden waren (vermutlich durch Umbauarbeiten bedingt). Zu-

dem waren manche Abstände sehr ungenau oder gleich komplett falsch eingezeichnet.

Im Gegensatz zur vorherigen Quake III-Version sollte die Karte von der Ebene 1 des MZH diesmal von Anfang an auf einem einheitlichen Maßstab und exakten Messungen beruhen. Die Feuerwehrpläne lieferten keinen Maßstab und keine verwertbaren Maßangaben, somit wurden zunächst einige Referenzmaße benötigt.

Zu diesem Zweck wurden einige Wände und bauliche Strukturen (Mauern und Säulen) der Ebene 1 von Hand vermessen. Mit diesen Referenzmaßen konnten viele der fehlenden Maße der Ebene 1 errechnet werden. Dieses Vorgehen stieß jedoch häufig auf Unstimmigkeiten. So gab es in der realen Ebene 1 plötzlich eine Tür, die im Plan entweder nicht eingezeichnet oder mit völlig anderen Maßen ausgestattet war. Manche Gänge waren breiter oder schmaler als auf dem Plan und so mußten alle Dimensionen in der Ebene 1 mehrmals nachgemessen und die Modelle mehrmals korrigiert werden.

Zum Modellieren der Karte in einem korrekten Maßstab war es erforderlich, die metrischen Maßangaben in Unrealeinheiten umzuwandeln. Im Unreal Developer Network wurde dafür der Umrechnungsfaktor 52,5 vorgeschlagen (1 Meter = 52,5 Unrealeinheiten). Er bildete zukünftig den Maßstab für sämtliche Arbeiten an der Ebene 1.

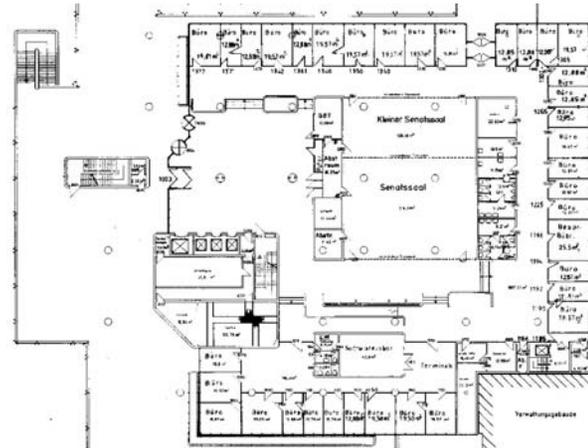


Abbildung II.38: Der Feuerwehrplan der Ebene 1 des MZH.

Zu den enormen Mengen an Skizzen und Maßen kam eine weitere Modellierungshilfe hinzu: Fotos. Objekte und Baustrukturen wurden aus allen erdenklichen Blickwinkeln fotografiert, damit beim Modellieren nicht der Bezug zu den tatsächlichen Dimensionen verloren ging. Gleichzeitig wurden praktisch von jeder Wand, jedem Plakat, jedem Schild und jedem Schalter ein Foto benötigt, das sich als Textur für den virtuellen Nachbau aufbereiten ließ.

Mit dem Maßnehmen und Anfertigen von Skizzen wurde der komplexe Aufbau der Ebene 1 erst klar. Zwar wirkt die Ebene auf den ersten Blick einheitlich und geometrisch strukturiert, da sie hauptsächlich aus rechteckigen Räumen, Gän-

gen und Strukturen besteht und es kaum schräge Wände oder gar runde Formen gibt. Dennoch macht sich auf den zweiten Blick eine gewisse Komplexität bemerkbar.

Ein großer Teil der Ebene 1 (der nördliche und östliche Teil, der DiMeB-Bereich und der kleine Senatssaal) ist leicht erhöht und wird durch Treppen, Mauern und Wände vom Rest abgegrenzt. Treppen benötigen andere Texturen als der normale Fliesen-Boden und jede Mauer muß extra modelliert werden.

Viele Mauern gehen in einer Rundung in die nächste Mauer über und verfügen außerdem über einen konkaven Mauerfuß. Die Regelmäßigkeit der Mauern wird häufig durch Wechsel der Bo-

denhöhe oder eingelassene andere Konstruktionen unterbrochen.

Die Beleuchtungskonstruktion erzeugt nur die Illusion einer relativ niedrigen Decke. Darüber folgen noch ein bis zwei Meter Raum, in dem unter anderem das Klimasystem installiert ist. Zu allen Wänden hin und in manchen Bereichen ist eine weiße Zwischendecke eingelassen, die exakt an die Beleuchtungskonstruktion angepasst sein muss. Insbesondere vom Foyer aus ist der Außenbereich gut zu sehen und muss entsprechend modelliert werden.

Die Modellierung der BSP-Geometrie der Ebene ging Hand in Hand mit dem Maßnehmen und dem Anfertigen von Skizzen. Zumeist wurden die genommenen Maße und angefertigten Skizzen sofort in BSP-Geometrie umgesetzt.

Der Aufbau der BSP-Geometrie orientierte sich teilweise an der Standardvorgehensweise zum Bau einer Indoor-Map. Die Standardvorgehensweise sieht vor, zunächstmal grobe Räume und Gänge zu erstellen und dann Details herauszuarbeiten.

Für das MZH-Level wurden zunächst ein großer, fast die gesamte Ebene umfassender Raum geschaffen. In dessen Mitte wurde anschließend der Block mit den Senatssälen und den an sie angrenzenden Räumen (Hausmeisterbüro, Toiletten) gebaut. Nun wurden die erhöhten Böden wieder ad-

tiert (aufgrund der komplexen Baustruktur wurden sie nicht gleich bei der Subtraktion zu Beginn berücksichtigt). Weitere Bereiche wie der DiMeB-Bereich wurden erzeugt und einige grundlegende Baustrukturen wie die geraden Abschnitte der Mauern eingesetzt (die abgerundeten Teile wurden später als Static Meshes modelliert).

Nachdem nun die grobe Struktur geschaffen war, konnten darauf aufbauende Arbeiten beginnen, wie beispielsweise das Einbauen von Bürotüren, Säulen oder Tafelschienen. Mit der Zeit entstanden so immer mehr Details.

In der ersten Version war vieles noch BSP-Geometrie, was aus Performance-Gründen eher Static Meshes hätte werden sollen. Dazu gehören unter anderem die Türen oder die braunen Bänder, die oben entlang vieler Wände verläuft.

Die Static Meshes für die Objekte der ersten Version der MZH-Karte wurden noch mit dem Unreal Editor gebaut. Der Bau eines Static Meshes im Editor verläuft ähnlich dem Bau der BSP-Geometrie. Das fertig modellierte und texturierte Objekt wird dann mittels einer bestimmten Export-Methodik in ein Static Mesh verwandelt und kann als solches nun in ein Level importiert werden.

Diese Methode zur Static Mesh-Erstellung hat jedoch Nachteile, wie im Laufe der Zeit bemerkt wurde. So sind die Modellierungs-Möglichkeiten

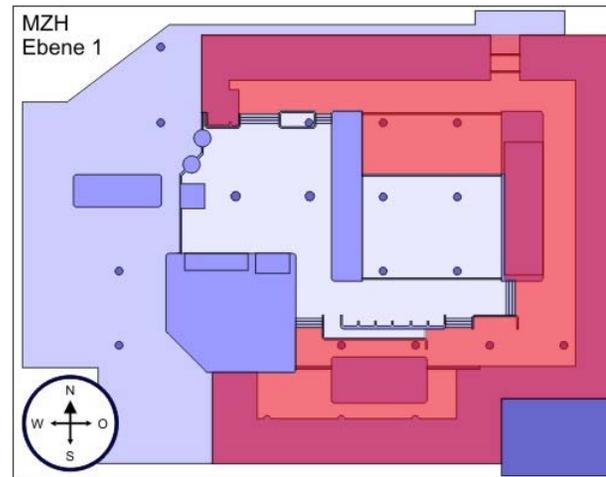


Abbildung II.39: Karte von den Bereichen der Ebene 1 mit erhöhtem Fußboden.

des Editors auf die BSP-Tools beschränkt, maßgenaues Arbeiten oder das Modellieren komplexer Elemente ist entweder unmöglich oder zu kompliziert und zeitaufwendig. Ferner werden die Static Meshes teilweise inkorrekt exportiert (Fehler beim Splitten komplexer Polygone) und beim Exportieren mit optisch stark gerundeten Kanten versehen (was bei vielen Objekten nicht wünschenswert war).

Ende September war die Ebene 1 bereits soweit fertig, wie es für die Präsentation angedacht war. Zur weiteren Entwicklung der Ebene war der Eindruck der Karte im Cave nötig. Daher haben wir uns die Wochen bis zur Präsentation auf Expe-

rimente mit der Beleuchtung und der Karma-Engine konzentriert und diverse Texturen verbessert und geändert.

Am Projektwochenende auf Norderney (kurz vor der Präsentation) nutzen wir die Karte der Ebene 1 noch zu verschiedenen Experimenten mit der Beleuchtung und der Karma-Engine (vgl. Kapitel 2.1).

In der Sitzecke des DiMeB-Bereichs wurden einige Deckenlampen installiert, die durch physische Einwirkungen bewegt werden konnten und deren Lichtschein sich dann dynamisch mit der umherspringenden Lampe bewegte. Da die Waffe zum Manipulieren von Karmaobjekten damals

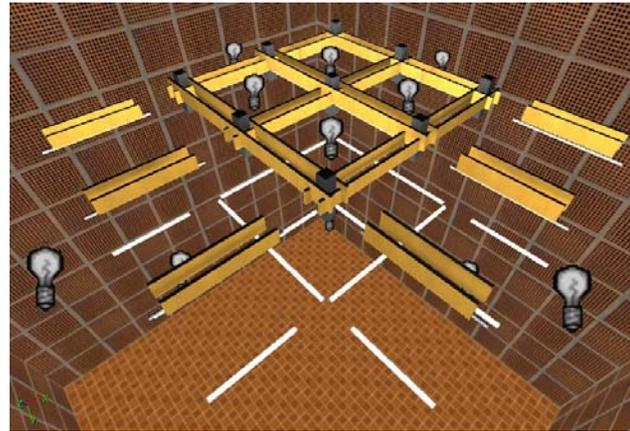


Abbildung II.40: Der erste Bausatz der Beleuchtungskonstruktion der Ebene 1. Später wurden noch wesentlich mehr benötigt.

noch nicht existierte, musste die physische Einwirkung durch die Granaten einer Waffe aus Unreal Tournament 2003 erzeugt werden. Zwei weitere Experimente waren eine große Kugel vor dem Raum 1400, die nur mühsam bewegt werden konnte und eine ebenfalls bewegliche Lampe mit dynamischem Lichtschein im Hausmeisterbüro.

#### 4.1.2 Die erste Präsentation und ihre Auswertung

Bei der bereits seit langem geplanten Präsentation hatten alle MiCa-Projekte die Gelegenheit, ihre Caves der interessierten Öffentlichkeit zu präsentieren. Im MiCasa-Cave konnte man unter an-

derem die Ebene 1 des MZH erkunden. Diese war bereits soweit modelliert, dass die Entwickler einen guten Eindruck von der Wirkung der Karte im Cave bekommen konnten und die Besucher einiges zu entdecken hatten.

Die Gänge der Ebene 1 waren detailgetreu modelliert: Neben einer installierten Deckenkonstruktion gab es Notausgangsschilder, passende Beleuchtung und einige Objekte, wie verteilt aufgestellte Mülleimer oder den Cola-Automaten, die etwas Leben in die Gänge brachten. Die Türen entlang der Gänge waren in der Map noch sehr einheitliche BSP-Objekte, immerhin schon mit Türgriffen.

Das Foyer war nur teilweise modelliert, haupt-

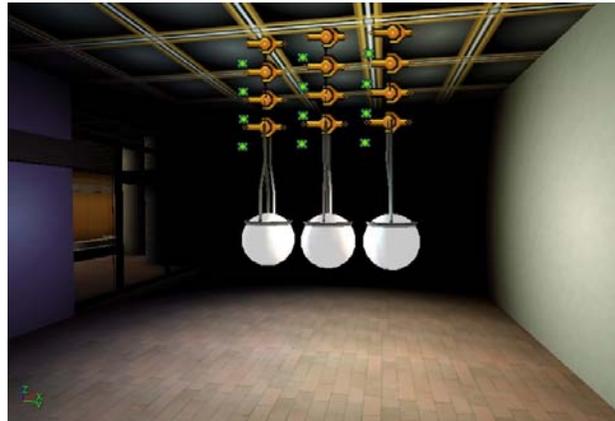


Abbildung II.41: Lampen und Ketten sind verknüpfte Karmaobjekte. Werden die Lampen in Bewegung versetzt bewegt sich der Lichtschein dynamisch mit.

sächlich zur Ostseite hin (u.a. die Wand, wo die Telefone und das Hausmeisterbüro zu finden sind). Es fehlte aber die gesamte Westwand, statt der Eingangsfront und einem Blick in die Außenwelt existierte nur ein mysteriöser, dunkler Bereich. Die Fahrstühle waren simple Blöcke in der Wand, der ZAIT-Besprechungsraum ein brauner Klotz. Es gab bereits die meisten der baulichen Strukturen, wie etwa Säulen, Mauern und Treppen zu den erhöhten Bereichen des MZH.

Detailliert waren auch die beiden Senatssäle: Zwar gab es nur Tische und keine Stühle, aber verschiedene andere Objekte wie der Beamer, das Wandbild im Raum 1380, Tafeln oder die Graffiti auf den Säulen im Raum 1400 sorgten für eine

vertraute Atmosphäre im Cave.

Es gab bereits eine Deckenkonstruktion, damals noch ein im Unreal Editor modellierter und komplexer Static-Mesh-Bausatz.

Der DiMeB-Bereich mit den Testlampen als Karmaobjekte wurde unzugänglich gemacht, die Lampe im Hausmeisterbüro und die riesige Karma-Kugel wurden vor der Präsentation entfernt (in Ermangelung von Möglichkeiten, diese Karma-Objekte zu bewegen).

Die Präsentation gab den Map-Entwicklern und den anderen Projektmitgliedern erstmals die Möglichkeit, die virtuelle Ebene 1 im Cave zu erleben. Wie gut oder schlecht Objekte und Tex-

turen wirkten oder wie Beleuchtung und Atmosphäre passten.

Hauptkritikpunkt war die Größe des Levels - die Decke wirkte zu niedrig, die Türen und Durchgänge zu schmal. Dieses Problem sollte jedoch durch das seit längerem angedachte Rekonfigurieren der Maße der Spielerfigur (dem PlayerActor) behoben werden, weswegen sich dazu entschlossen wurde, den Modellierungsmaßstab beizubehalten.

Wenige weitere Kritikpunkte waren die nicht modellierten Bereiche des Levels, teils noch vorhandene Beta-Texturen oder das Fehlen vieler Objekte. Diese Kritikpunkte würden sich aber mit der weiteren Entwicklung des Levels von selbst erledigen.

Das Feedback der Besucher war zumeist sehr positiv. Jene, die mal im MZH waren, erkannten die Ebene 1 auf Anhieb und hatten meist auch viel Spaß bei der Erkundung der Ebene und dem Entdecken verschiedener bereits modellierter Details.

Der große Besucherandrang warf gleich zwei Probleme auf: Einerseits befand sich selten nur eine einzelne Person im Cave (wie es das Konzept vorsieht), zumeist waren es Besuchergruppen von 4 bis 7 Personen. Andererseits konnten die meisten Besucher aufgrund des großen Andrangs nur wenige Minuten im Cave bleiben.

Die Tanzmatte war für die meisten Besucher in

den ersten paar Minuten eine sehr gewöhnungsbedürftige Methode zur Bewegung durch die virtuelle Umgebung im Cave, die intuitive Benutzung entstand nach einiger Übung. Ein aktionsreiches Spiel wie PacMan könnte Besucher, die sich mit der Steuerung noch nicht gut auskennen, schnell frustrieren.

Viele Besucher waren bereits mit der Entdeckung und der Erfahrung einer rundum projizierten virtuellen Umgebung immens beschäftigt. Ein Spiel hätte (zumindest in den ersten Minuten) vermutlich als störend empfunden werden können.

Bei der weiteren Entwicklung einer Spielidee mussten diese Fakten also nunmehr ebenfalls berücksichtigt werden.

Das Spiel musste mit größeren Besuchermassen zurechtkommen, d. h. es musste schnell erklärt und intuitiv sein und durfte nicht zulange dauern bzw. musste von einer anderen Person relativ problemlos übernommen werden können.

Die Eingewöhnung bei der Bedienung musste berücksichtigt werden, insbesondere für actionbasierte Spiele.

Das Spiel musste gegebenenfalls auch so konzipiert sein, das mehrere Besucher gleichzeitig im Cave sein konnten, ohne das Spiel oder den Spieler zu stören.

Nach dem Präsentationstag wurden zunächst nur Detailverbesserungen durchgeführt, wie etwa das

Erstellen neuer Texturen. außerdem begann die Konzeptarbeit an der Ebene 0, d.h. es wurden verschiedene Maße genommen und Skizzen angefertigt und eine Konzeptversion einer Map für Unreal Tournament 2003 entstand. Im laufenden Wintersemester 03/04 kamen die Arbeiten allerdings nur schleppend voran.

Die Modellierung der Ebene 0 wurde im Semester vorläufig zurückgestellt, dafür wurden Anfang 2004 die Arbeiten an der Ebene 1 wieder aufgenommen. In der Zeit entstand der DiMeB-Bereich und ein Teil der Außenwelt und des Eingangsbereiches als BSP-Geometrie, zusammen mit diversen Objekten.

#### 4.1.3 Eine neue Karte

Grundsätzlich gab es zwei Möglichkeiten, die Map der Ebene 1 des MZH weiterzuentwickeln: Entweder basierend auf der letzten Version der Ebene 1 die Arbeit fortzuführen oder eine komplett neue Karte aufzusetzen. Die Wahl fiel auf eine neue Version, denn insbesondere die Arbeiten an der alten Karte seit der Präsentation im Oktober 2003 hatten einige Probleme deutlich gemacht und es kamen auch neue Gründe auf, die eine neue Version bzw. auch ein neues Herangehen an die Karte notwendig machten.

##### 1.

Die Maße der alten Karte waren teilweise zu ungenau (was maßgeblich an der Qualität der Feuerwehrläne lag). Das Resultat war, dass jene Static Meshes und BSP-Geometrien, die auf diese ungenauen Maße aufbauten, auch nicht mehr passten und daher ihrerseits korrigiert werden mussten. Die weiter getragenen Korrekturen wurden jedoch allmählich unüberschaubar und machten die Modellierung nach genommenen Maßen und Vorlagen schwierig, da alles immer wieder so korrigiert werden musste, dass es mit dem bereits Modellierten passte.

##### 2.

Die BSP-Geometrie sollte neu aufgesetzt werden, um einigen (teils neuen, teils wieder ausgegrabenen) Ideen eine Basis zu liefern. Zu den Ideen zählten unter anderem die Ebene 0, Ebene 9 und das Dach, sowie ein möglicher Zugang zu den Außenbereichen und einigen Büros. Ferner sollten einige Details herausgearbeitet werden, wie die eingelassenen Mauerfüße bei manchen Wänden oder einige Elemente der Tafelwände in den Senatssälen.

##### 3.

Bislang wurden die Static Meshes direkt im Unreal Editor modelliert, was verschiedene Probleme mit sich brachte.

- die Static Meshes, die sich in Unreal Editor erzeugen ließen, überzeugten von der Qualität her nicht, der Export wies qualitative Mängel auf.
- verschiedene Objekte, die bislang BSP-Geometrie waren, sollten zur besseren Handhabung und auch aus Performance-Gründen künftig als Static Mesh vorliegen.
- es waren noch zahlreiche weitere Static Meshes gewünscht.
- die gesamte Modellierungs-Arbeit lag bislang bei einer Person, nämlich dem Map-Modellierer selbst.

Die Baustruktur (also Räume und Gänge) der Ebene 1 sollte dieses Mal mehr an den Feuerwehrplänen ausgerichtet sein. Zwar waren diese Pläne in einigen Punkten fehlerhaft und allgemein zu ungenau, jedoch boten sie als einzige Quelle einen mehr oder weniger vollständigen Überblick über die Baustruktur der Ebene. Allerdings musste eine Möglichkeit gefunden werden, die besagten Mängel der Feuerwehrpläne zu korrigieren und die Ungenauigkeiten in den Planmaßen mit den genommenen Objektmaßen in Einklang zu bringen.

Außerdem verlangte die Komplexität der Ebene 1 ein strukturiertes Vorgehen, da unter anderem die Erstellung zahlreicher Static Meshes von der Ge-

nauigkeit des Levels abhängig war. Beispielsweise mussten die nunmehr als Static Mesh geplanten Türen oder Fahrstühle in die vorgesehenen Aussparungen passen, die Deckenkonstruktion - ein Bausatz aus verschiedenen Static Meshes - erforderte die exakte Modellierung aller Gänge und Decken.

Um die Einheitlichkeit zwischen den verschiedenen Maßen und Vorlagen zu gewährleisten und gleichzeitig den Aufbau der Geometrie planen zu können, wurde eine Modellierungsvorlage erstellt.

Die Modellierungsvorlage war eine in Corel Draw erstellte Vektorgrafik. Die Grundlage für bildete der Feuerwehrplan der Ebene 1, dessen Baustruktur (Wände, Säulen, Türaussparungen) in eine neue einheitliche Skizze übertragen wurde. Dabei wurden Ungenauigkeiten und falsche Maße in den Feuerwehrplänen genauso wie Änderungen der Baustruktur (beispielsweise neue Räume, Wände und verschobene Türen) berücksichtigt und korrigiert. Da die Skizze als Vektorgrafik angelegt wurde, konnte die Baustruktur auch gleich in modellierbare Elemente zerlegt und der BSP-Aufbau so teilweise im voraus geplant werden. Außerdem wurde die Skizze gleich in Unrealeinheiten erstellt, was das weitere Planen der Map und Modellieren der BSP-Geometrie einfacher und übersichtlicher machte.

Diese Modellierungsvorlage erlaubte nun auch das exakte Konzeptionieren und Bauen von Static Meshes, die von der Baustruktur abhängig waren (Deckenkonstruktion, Borte, Ziermauern u.a.).

Das Erstellen des Plans war zeitlich aufwendig, bedingt durch Neuvermessungen und Abgleichen der Maße und verschiedenen Vorlagen. Das weitere Entwickeln und Planen wurde dadurch aber bedeutend einfacher.

So wurden aus der Modellierung von Static Meshes und dem Erstellen von Texturen zwei neue Arbeitsbereiche. Für beide Bereiche wurden Listen mit benötigten Static Meshes bzw. Texturen erstellt, anhand derer sich die Arbeit planen und auf einzelne Projektmitglieder verteilen ließ. Die Static Meshes wurden mit Programmen zum Modellieren von dreidimensionalen Objekten erstellt, die weit mehr Werkzeuge zum Bearbeiten dieser Objekte anboten als der Unreal Editor. Zusätzlich sollten verstärkt Shader zum Einsatz kommen, um bestimmten Oberflächen (z. B. Glas und Chrom) realistischere Reflektionen zu verleihen.

Die zugänglichen Bereiche der Ebene 1 sind das Foyer, die Gänge, der DiMeB-Bereich und die Senatssäle. Weitere Räume und Bereiche sind das Treppenhaus (Zugang zum Portal), die Fahrstühle, zahlreiche Büros, die Toiletten und diverse Technik- und Hauswirtschaftsräume. Im Gegen-

satz zur Modellierung der ersten Version der Map wurden nun noch einige Details herausgearbeitet. Diese Kleinigkeiten fielen kaum jemandem explizit auf, ließen aber auf ihre subtile Weise die Baustruktur und damit den Gesamteindruck realistischer wirken.

Die Ebene 1 hat zwei Seiten mit großzügigen Fenstern, durch die die Umgebung des MZH gesehen werden kann. Um diese Umgebung anzudeuten, aber dennoch nicht in allen Details nachmodellieren zu müssen, wurden mehrere Hintergrundtexturen erstellt, die in einiger Entfernung auf eine planare Wand plaziert wurden. Diese Umgebungstexturen werden auch Skybox genannt. Bei weit entfernten Teilen der Umgebung, wie z. B. dem Himmel oder weit entfernten Gebäuden reicht eine Skybox völlig aus, um den Eindruck einer vollständigen Umgebung beim Betrachter zu erzielen.

Schwieriger wird es bei näher am Fenster gelegenen Objekten wie Blumenkübeln oder Treppen. Hier werden die fehlenden perspektivischen Verschiebungen schnell deutlich, wenn der Betrachter sich vor dem Fenster bewegt und auf die Hintergrundtextur achtet.

Der an der nördlichen Seite der Ebene 1 gelegene Ausgang zum GW 2 ist ein relativ schmaler Durchgang mit zwei hintereinander liegenden Türen. Die Außenwelt auf dieser Seite verfügt nur



Abbildung II.42: Überblick über die finale Version der MZH-Map von Nordosten aus gesehen.

über wenig nahe Objekte, und bei denen wären Fehler in der Perspektive nur schwer auszumachen, da der Blickwinkel des Betrachters durch die Tür recht klein ist. So wurde hier lediglich der Teil bis zur ersten Treppe modelliert. Die sichtbaren Pflanzen und die Treppenmauern, der GW 2-Gebäudeteil und diverse Hintergründe existieren nur auf der Skybox.

Die westliche Seite der Ebene 1 (die Eingangsfront) verfügt über zahlreiche Fenster, die somit auch einen breiten Blickwinkel auf die Außenwelt. Zudem kann der Betrachter nah an diese Fenster herantreten und damit den Blickwinkel auf die Umgebung noch vergrößern. Es gibt daher zahlreiche Objekte (Säulen, Boden, Decke und Pflan-

zenkübel), bei denen die fehlende perspektivische Verschiebung auffallen würde. Diese nahegelegenen Objekte wurden modelliert und texturiert, die Skyline (u. a. mit Wald und Mensa) ist eine Skybox. Bei den verwendeten Texturen für die Skybox handelt es sich um zusammengeschnittene Fotos der Umgebung.

Obwohl diese neue Version der virtuellen Ebene 1 des MZH an sich vollständig ist, gibt es Ansätze für die zukünftige Arbeit an dieser Karte.

- Fehlende Blocking Volumes

Es fehlten Blocking Volumes für die Kollisionsberechnungen unter anderem für die Tischgruppen im 1400 und im 1380 sowie diverse Papierkörbe.

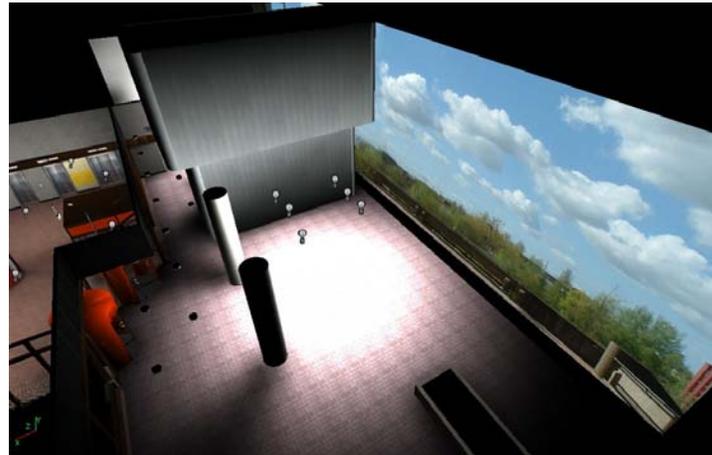


Abbildung II.43: Fast alles Illusion: Lediglich nahe Objekte wie die Säulen wirklich modelliert. Eine Skybox zeigt die Peripherie und den Horizont.

Durch das Fehlen der Blocking Volumes war es für die Besucher des Caves möglich, diese Objekte zu besteigen. Durch die Größe des PlayerActors hing man damit allerdings in der Deckenkonstruktion fest und konnte sich nur durch geschickte Bewegungsmanöver oder Sprünge wieder aus der vermeintlichen Falle befreien.

- Beleuchtung

Die Beleuchtung hätte insgesamt etwas heller ausfallen können. Teilweise fehlte noch Beleuchtung oder sie war nicht präzise genug ausgerichtet.

- Antiportals

Zum Schluss fehlte noch diverse BSP-Geometrie, unter anderem die Decken. Das machte das Einrichten von Antiportals schwierig, da abgeschottete Schlüsselpunkte fehlten. Die befürchteten Performance-Einbußen traten beim Einsatz im Cave jedoch nicht ein, da durch die starren Ansichten fast die gesamte benötigte BSP-Geometrie beim Betreten der Karte auf einmal geladen werden konnte und weiteres Nachladen von Informationen selten notwendig war.

- Größe der Karte

Bei der Entwicklung der virtuellen Ebene 1 des MZH wurde vom Umrechnungsfaktor 52,5 (siehe

oben) zwischen Meter und Unrealeinheiten ausgegangen. Um das die Größenverhältnisse zu der virtuellen Figur zu korrigieren, sollten die Abmessungen des PlayerActor rekonfiguriert werden. Diese Konfiguration fehlte bis zum Schluß, womit das Größenverhältnis des PlayerActors zur MZH-Karte auch nicht ganz passte.

- Kollisionsberechnung

Bei einigen Objekten fehlte die Kollisionsberechnung, unter anderem bei sämtlichen Türen und den Fahrstühlen. Dadurch konnte man problemlos hinter die Kulissen wandern und damit auch unter anderem auf den Boden des großen, alles umgebenden Kubus' fallen, von wo es kein Zurück mehr gab.

In levelbasierten Spielen ist es üblich, sogenannte Eastereggs einzubauen. Eastereggs sind kleine Räume, die von den Leveldesignern versteckt eingebaut werden und von den Spielern gefunden werden sollen (daher auch Easteregg = Osterei). Oft enthalten sie allerlei Grüße oder den Kontakt des Künstlers.

Auch die virtuelle Ebene 1 des MZH enthält Eastereggs, allerdings in der Version für den Projekttag 2004 weniger als ursprünglich geplant.

- Fliegende Kakaotüten

Im Eingangsbereich, bei der Sitzgruppe direkt vor dem Hausmeister-Büro, kann man zwei fliegende

Kakaotüten bewundern. Eine weitere Kakaotüte schwebt im Senatssaal über einem Tisch.

- Der geheime Raum

Die zweite Tür links vom linken Eingang zum DiMeB-Bereich ist der Raum 1009. Bei diesem Raum fehlt die Tür und das Licht darin ist fast immer ausgeschaltet. Normalerweise stehen dort verschiedene Schränke und ein Kopierer, aber das weiß fast keiner. Also bekamen die Projekttagbesucher eine etwas andere Realität vorgesetzt. In der virtuellen Ebene 1 befinden sich in diesem Raum ein überdimensionaler Stuhl, ein Sitzplatz an der Decke und ein Gagfoto des Projektteams.

- Peez fetter taktischer Plan

... kann auf einer Tafel vor dem 1380 bewundert werden. Die (reale) Tafel stand mehr als ein Jahr auf dem Gang vor dem 1380, ehe sich jemand erbarmt hat und sie spurlos verschwinden ließ. Der fette taktische Plan demonstriert ... einen Plan. Öhm. Plan, ja.

- Einblicke von außen

Eigentlich kein Easteregg, sondern ein Fehler in der Kollisionsberechnung. Bei sämtlichen der türkisfarbenen Türen fehlten die Blocking Volumes und so konnte man ungehindert durch sie hindurch gehen. Das erlaubte ungewohnte (und eigentlich auch ungewollte) Einblicke von außen, z.B. durch die Eingangsfront ins Foyer.

#### 4.1.4 Verworfen oder nicht realisierte Ansätze

In den zahlreichen Entwicklungsphasen des virtuellen MZH gab es noch viele Ideen und Ansätze, die im folgenden kurz besprochen werden sollen.

- Das komplette MZH als virtuelle Welt

Diese Idee wurde aus verschiedenen Gründen recht bald verworfen. So bieten die meisten Ebenen des MZH zu wenig interessantere Aspekte, um den Besucher damit zu fesseln. Zudem wäre die Konstruktion ein zu hoher Arbeitsaufwand gewesen.

- Das Dach des MZH

Ein Blick vom Dach des MZH auf den Campus und die Bremer Skyline, eventuell in Verbindung mit einer Windmaschine, wäre ein interessanter Effekt gewesen. Dazu wäre ein funktionierendes Fahrstuhlsystem (Map-Teleporter) nötig gewesen. Und für die Windmaschine hätte die Konstruktion des Caves nochmal an einigen Stellen überarbeitet werden müssen.

- Zu lösende Aufgaben im MZH & PacMan

Ein immer wieder aufkommender Gedanke war, dass der Besucher im Cave etwas zu tun haben müsste, daß ihm eine Aufgabe gestellt werden sollte. Für das virtuelle Ebene 1 des MZH gab es zwei Hauptgedanken. Zum einen sollte ein Action-Spiel wie PacMan realisiert werden

(mit verschiedenen Profs als Special Ghost Stars). Zum anderen sollten dem Besucher bestimmte Aufgaben gestellt werden, wie etwa das Finden von Gegenständen (z. B. Crimsons Tasse, SBLNs oder der Maps'n'Robots-Teddy).

- Die mystische Ebene 9

Der große Fahrstuhl erlaubt als Fahrziel-Anwahl auch die Ebene 9. Dafür ist jedoch ein Zugangsschlüssel nötig, den nur die Techniker besitzen. Und so wissen viele von der Existenz der Ebene 9, kaum einer weiß jedoch, was sich dort befindet. Grund genug, hier ein wenig die Fantasie spielen zu lassen und sie in unserer virtuellen Welt nachzubauen. Ideen waren unter anderem eine paradiesische Idylle mit Bäumen und einem Fluß unterzubringen oder ein PacMan-/Spiel-Level.

- Fahrstühle

Der große Fahrstuhl sollte ursprünglich zugänglich sein und als Level-Teleporter zu Ebene 0, Ebene 9 oder dem Dach dienen. Da all diese Level jedoch nicht existent waren, bestand auch nur geringer Anlass, das Interior des Fahrstuhls zu modellieren.

- Ebene 0

Ursprünglich ebenfalls geplant, fiel die Ebene 0 erst Anfang des letzten Projektsemesters einem realistischen Zeitplan zum Opfer. So wären für die Ebene 0 zahlreiche weitere Static Meshes er-

forderlich gewesen, weitere BSP-Modellierung sowie ein funktionierender Fahrstuhl. Ferner gibt es in der Ebene Fenster zu drei Seiten, darunter die umfangreiche Fensterfront der Rechnerräume in der Ebene 0. Insbesondere letztere hätten eine präzise Modellierung der Außenwelt erfordert.

Die Arbeit an einer ersten Version der Ebene 0 wurde im Anschluß an die Präsentation im Oktober 2003 begonnen, gelangte aber nicht über den Konzeptstatus hinaus.

- Taschenlampe

Eine weitere Spielidee war, die virtuelle Ebene 1 stark abzdunkeln und den MiCasa Stick als Taschenlampe fungieren zu lassen. Die Besucher müssten dann in einer dunklen Ebene 1 umherirren und sich nur mit Hilfe der Taschenlampe orientieren. Dieser Effekt hätte sich einerseits gut mit Soundeffekten verbinden lassen, andererseits mit der Spielidee, etwas im MZH suchen zu müssen. Ein entsprechender Mod dafür existierte auch ziemlich bald, die Idee wurde jedoch nicht weiter verfolgt.

- Geräusche im MZH

Um die Atmosphäre im MZH-Level noch intensiver zu gestalten war geplant, ambiente Geräusche und Musik an bestimmten Stellen in das Level einzubauen. Beispielsweise Laufgeräusche oder ein Laptop, dessen Lüfter anspringt.

- Zugang zu einigen Außenbereichen der Ebene 1

Im Laufe des 3. Projektsemesters kam die Idee auf, wenigstens einen Teil des Außenbereichs begehbar zu machen. Voraussetzungen wären eine komplette Skybox und eine modellierte Außenwelt gewesen (sowie natürlich offene Türen).

- Bewegliche Türen und manipulierbare Objekte  
Um dem MiCasa Stick für das MZH einen Zweck zu geben, sollte man ursprünglich Objekte damit manipulieren können. Beispiele dafür wären bewegliche Türen, Lichtschalter, Cola-Automat oder Fahrstuhl-Knöpfe.

#### 4.1.5 Static Meshes und Shader

Nachdem die Entscheidung gefallen ist, welcher Gegenstand als Static Mesh benötigt wird, ist der erste Schritt das Sammeln der relevanten Maße. Dazu wird eine Skizze erstellt und mit den benötigten Maßen versehen (siehe Abbildung II.45). Meistens ist eine Genauigkeit von 0,5 bis 1 cm ausreichend. Dabei wird der geometrische Aufbau beim Modellieren des Static Meshes bereits durchdacht.

Anhand der genommenen Maße kann das maßstabsgetreue Drahtgittermodell erstellt werden. Zuerst müssen dazu die Maße in die Maßeinheit unserer virtuellen Umgebung (Unrealeinheiten) umgerechnet werden. Ein Meter entspricht 52,5 Unrealeinheiten. Durch diese Skalierung ließen

sich die realen Größenverhältnisse in der virtuellen Ebene 1 nachbilden. Anschließend wurde das Static Mesh aus mehreren geometrischen Grundformen zusammengesetzt und modelliert.

Das so entstandene Drahtgittermodell ist gemessen an seinem realen Vorbild sehr grob. Winzige Rillen, Abschürfungen und Spalten werden nicht modelliert, sondern mit Hilfe von Texturen angedeutet. Texturen entstehen meistens aus hochauflösenden Fotos, die aufbereitet und auf das Objekt mit Hilfe von Texturkoordinaten abgebildet werden. So erhält das Static Mesh gleichzeitig mehr Detail und eine realistische Farbgebung, die schließlich vom realen Vorbild abstammt.

Prinzipiell kann das Static Mesh zu diesem Zeitpunkt bereits in die virtuelle Welt integriert werden. Je nach Komplexität des Modells ist es zur Vereinfachung der Kollisionsberechnungen sinnvoll, Kollisionsgeometrie hinzuzufügen bevor das Static Mesh verwendet wird. Dabei werden einfache, konvexe geometrische Körper um die komplexe Geometrie des eigentlichen Drahtgittermodells platziert, sodass diese komplett umschlossen wird. Die Grafik-Engine kann nun alle Abfragen an diesen einfachen Kollisionsgeometrien abarbeiten, was diese Kollisionsberechnung extrem vereinfacht.

Nachdem die Geometrie erstellt, die Texturkoordinaten zugewiesen und die Kollisionsgeometrie

hinzugefügt ist wird das Static Mesh aus der jeweiligen Modellier-Software exportiert und mit dem Unreal Editor in der virtuellen Welt platziert.



Abbildung II.44: Ein modellierter Stuhl mit Kollisionsobjekten.

Nachdem sämtliche Texturen des MZH mittels Digitalkameras fotografiert, nachbearbeitet und kachelbar gemacht wurden, wirkte das MZH-Level schon recht realistisch mit den fotorealis-





Abbildung II.46: Durch die Verwendung von Texturen erhalten Objekte gleichzeitig mehr Detail und eine realistische Farbgebung.

tischen Texturen. Es fehlten jedoch noch jegliche Reflektionen, die wesentlich zum realistischen Gesamteindruck der virtuellen Welt beitragen würden.

Natürlich wirkende Objekte sehen von allen Seiten etwas anders aus, weil bei allen Oberflächen Reflektionen eine Rolle spielen. Bei der Texturerzeugung aus Fotografien kämpften wir noch mit den Tücken der ungleichmäßigen Beleuchtung. Da Reflektionen dynamisch sind, kann man sie auch nicht einfach in eine Textur einbauen und so simulieren. Es müssen sog. Shader erstellt werden, die über eine zusätzliche Umgebungstextur

diese Spiegelung dynamisch nachahmen.

Um ein Objekt mit Reflektionen zu versehen wurde zunächst eine Cubemap erstellt, die an dem Ort im Level ihren Ursprung hat, an dem sich das Objekt befindet. Für alle verwendeten Glasflächen wurden solche Cubemaps erzeugt, beispielsweise die Eingangsfront, sämtliche Glasvitritten und das Glas des Tageslichtprojektors. Insgesamt bedeutete dies eine große Anzahl von über 50 Bildern. Die Auflösung wurde auf 256x256 Pixel reduziert, da wir Geschwindigkeitseinbußen bei höheren Auflösungen befürchteten. Mit größeren Texturen (bis zu 1024x1024 möglich) würde die Qualität der Reflektionen demnach noch steigen.



Abbildung II.47: Türgriffe, -motor, und Schalter glänzen.

Nachdem alle Cubemaps erstellt waren, wurden daraus Umgebungstexturen erzeugt und schlussendlich im Unreal Editor auf die entsprechenden Objekte gelegt. Die Textur für den Projektor war etwas komplizierter, da nur ein bestimmter Teil der Projektoroberfläche (nämlich das Glas) spiegelnd sein sollte. Dies wurde über einen Alphakanal erzielt, der in genau diesem Bereich die Anwendung der Reflektionstextur zuließ.

Um den Grad der Spiegelung zu regulieren, wurde eine leicht graue (95% Weißanteil) Textur als SpecularityMask verwendet, die den Effekt noch realistischer wirken ließ. Insgesamt ließen diese



Abbildung II.48: In einer Glasvitrine im Eingangsbereich spiegelt sich die Eingangsfront.

Spiegeleffekte auf sämtlichen Glasflächen das Level viel realistischer wirken.

Neben Reflektionen, die die Umgebung, ähnlich einem Spiegel, recht exakt wiedergeben, gibt es Oberflächen, deren Reflektionseigenschaften zwischen Vollspiegelung und matter Oberfläche liegen. Diese unvollkommene Spiegelung bescherte uns ebenfalls Probleme bei der Texturgewinnung. Je rechter der Winkel zu einer solchen glänzenden Oberfläche wird, desto stärker werden die einfallenden Lichtstrahlen des Blitzes reflektiert und desto unbrauchbarer wurden die Fotos. In vielen Fällen wurden die Oberflächen dann von der Seite



Abbildung II.49: In der großen Glastür vor dem DiMeB-Bereich spiegelt sich die virtuelle Umgebung.



Abbildung II.50: Die vertäfelten Holzwände mit Reflektionens-Shader.

fotografiert und die optische Verzerrung anschließend in der Bildbearbeitung herausgerechnet. Dennoch sollte das fotografierte Material in der virtuellen Welt ähnliche Reflektionseigenschaften aufweisen, da es ein wesentlicher Bestandteil der Materialeigenschaften ist. Kleine Veränderungen in der Helligkeit und Farbe eines Materials bei der Betrachtung aus unterschiedlichen Blickwinkeln sind immer vorhanden, und werden meist nur unterbewusst wahrgenommen. Kunststoffoberflächen (rauh wie glatte) sind ein Beispiel dafür, wie in Abbildung II.47 zu sehen ist.

Der Glanzpunkt auf der Oberfläche wandert entsprechend dem Blickwinkel über das Objekt.

Beim Feinschliff des Levels mittels Shader mussten alle Texturen, die einen Glanz erhalten sollten, einen Shader erhalten. Prinzipiell war es immer der gleiche Shader, nur die Diffuse-Texturen änderten sich. Die so gewonnenen Shader mussten nur noch als Textur für das Objekt eingetragen werden.

Dieser Shadertyp diente ebenfalls als Grundlage für einen weiteren, etwas raffinierteren Effekt. Da jede Menge Holzoberflächen in der Ebene 1 zu texturieren waren, aber der Glanz das Holz zu neu erschienen ließ, wurde über die Specularity-Mask des neuen Holzshaders der Spiegelung eine Struktur gegeben. Dazu wurde die Holztextur

modifiziert, und ein Farbkanal (RGB) in den noch leeren Alphakanal (A) kopiert, die Kontraste zwischen Hell und Dunkel vergrößert und schlussendlich die neue Textur abgespeichert. Das Ergebnis sieht Holz ähnlicher, als die normale, shaderlose Version.

#### 4.1.6 Fazit

Die Verwendung von visuellen Effekten, die der Vergrößerung der Realitätsanmutung dienen, ist sehr zu empfehlen, besonders wenn es sich um möglichst detailgetreue Nachbildungen der Wirklichkeit in der Virtualität handelt. Nuancen bei der Darstellung virtueller Welten entscheiden hierbei meist über den Grad des empfundenen Realismus. Neben dem geschickten Einsatz von

Shadern spielen Licht und Schatten eine große Rolle bei der Wahrnehmung der virtuellen Umgebung.

Beim Bau der ersten Ebene des MZH wurde sich auf den Bau einer großen Anzahl von Objekten mit fotorealistischen Texturen konzentriert. Deshalb konnten erst gegen Fertigstellung des Levels Umgebungstexturen und andere Shader erzeugt werden. Leider ist deren Generierung statisch, und muss manuell vorgenommen werden. Durch große wie kleine Änderungen am Level mussten somit auch die Texturen für die Reflektionen überarbeitet werden. Wäre die Erstellung dynamisch und mit der Levelgenerierung im Mappeditor integriert, könnte man schon früh Umgebungstexturen einbauen und testen, was eine große Arbeitserleichterung bedeuten würde.

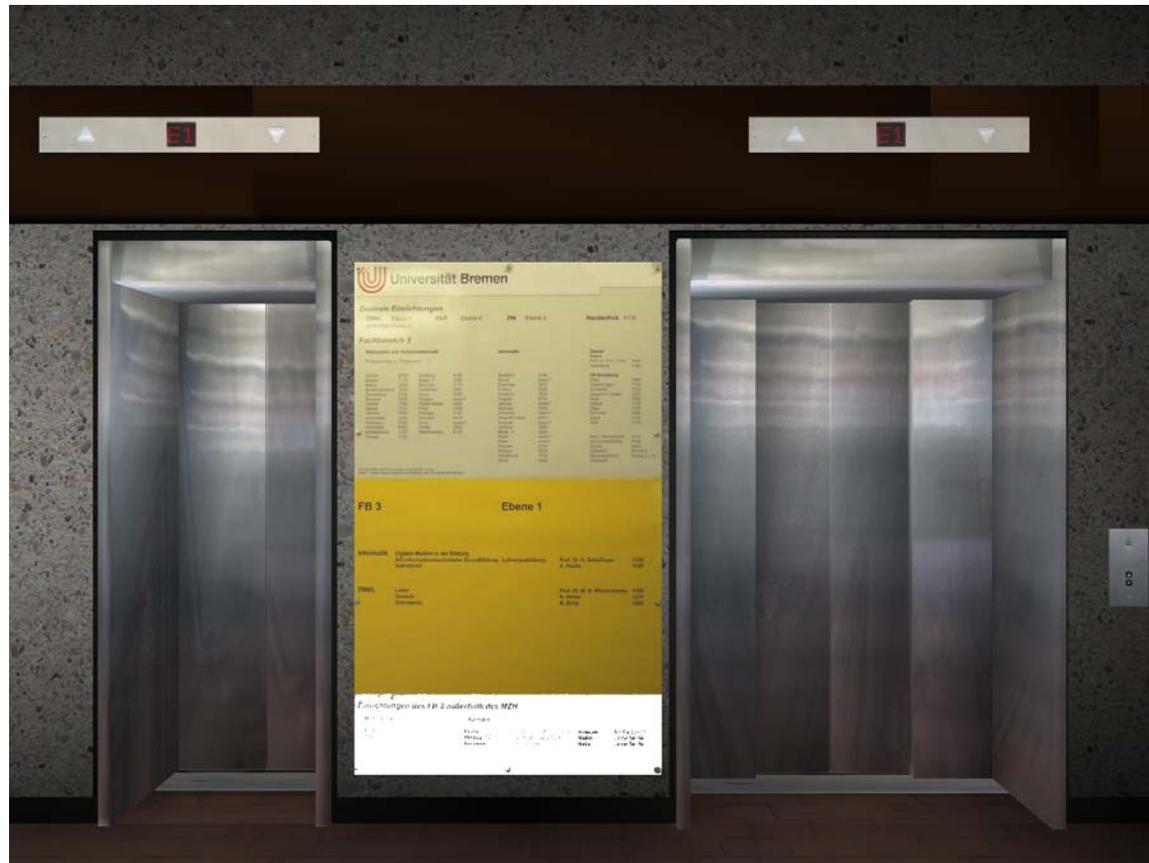


Abbildung II.51: Die Fahrstühle im Eingangsbereich.



Abbildung II.52: Der Flur vor dem kleinen Senatssaal.

## 4.2 Die Kunsthalle

Als erstes Umsetzungsobjekt wurde die Kunsthalle Bremen ausgewählt. Sie sollte später als Ausgangs- und Übertrittspunkt für Exkursionen an anderen Orten der Stadt, also in andere Teile der virtuellen Welt dienen.

Um die Kunsthalle Bremen möglichst detailgetreu nachbauen zu können, wurden zunächst möglichst viele Informationen über die Kunsthalle benötigt. Diese Informationen beinhalten Bilder und Maße der einzelnen Räume.

Im September 2003 war die zuständige Teilgruppe in die Kunsthalle eingeladen. Es wurde eine Ausnahmegenehmigung zum Fotografieren erteilt und eine freie Bewegung, in Begleitung einer Aufsichtsperson der Kunsthalle, zugesichert. Bei diesem Besuch entstanden sehr viele Fotos, die zum einen als Vorlage für viele geometrische Objekte, wie die Holztüren und die Fußleisten, dienten. Zum anderen dienten sie als Vorlage für die fotorealistischen Texturen, die für die virtuelle Umgebung der Kunsthalle erstellt wurden.

Um die Räume in den richtigen Größen nachbauen zu können, wurden möglichst genaue Maße der einzelnen Wände benötigt. Beim ersten Besuch in der Kunsthalle wurde nur eine Wand vermessen und gehofft, dass die restlichen Maße anhand der Grundrisssskizze der Kunsthalle aus dem Inter-

net rekonstruiert werden könnten. Bei diesem Besuch entstanden auch verschiedene Videosequenzen, die als Orientierung bei der Nachbildung der räumlichen Gegebenheiten dienten.

Nun wurde festgelegt welcher Teil der Kunsthalle nachgebaut werden sollte. Der Entschluss fiel auf den Nachbau des Eingangsbereiches bis in den ersten Flur. Zusätzlich wurde der Raum mit der offenen Decke bis zur nächsten Etage sowie der langgezogenen Raum mit der Fensterfront gegenüber des Eingangsbereichs nachgebaut. Zwei Nebenräume links vom Eingangsbereich und links ausgehend vom mittleren Raum sollten ebenfalls nachempfunden werden. Die Treppe im ersten Flur sollte zwar vorhanden, aber nicht benutzbar sein.

Wie sich später herausstellte, waren diese Einschränkungen richtig. Zum einen wurden nicht mehr Räume für unsere Portraits benötigt und zum anderen waren diese Teile gerade rechtzeitig zum Projekttag fertiggestellt.

### 4.2.1 Realisierung

Die größte Schwierigkeit war zunächst die Umsetzung der Größenverhältnisse der einzelnen Räume. Anhand der Skizze aus dem Internet und einem vorhandenen Wandmaß, wurden die Maße der restlichen Wände ausgerechnet. Der nächste Arbeitsschritt bestand in der Umrechnung dieser

Maße in Unrealeinheiten. Wie der Internetseite des Herstellers für Entwickler, dem sogenannten UDN ([www.udn.epicgames.com](http://www.udn.epicgames.com)), zu entnehmen war, entspricht einem Meter genau 52.2 Unrealeinheiten. Nachdem alle Maße übersetzt waren, konnte die Arbeit an der ersten Version der virtuellen Kunsthalle im Unreal Editor begonnen werden.

Es stellte sich heraus, dass die verwendete Skizze zu ungenau war, da die errechneten Maße unrealistisch wirkten. Diese erste Version der Kunsthalle hatte zu kleine Räume, wobei die Decken oft zu hoch waren.

Zunächst wurde versucht, die Räume, die zu klein oder zu groß waren anhand von Erinnerungen an den Kunsthallenbesuch passend zu editieren. Diese Vorgehensweise brachte jedoch nicht den gewünschten Erfolg. Es wurde ein wesentlich genauerer Bauplan der Kunsthalle benötigt. Auch die Fotos, die beim ersten Besuch in der Kunsthalle geschossen wurden, waren nicht zufriedenstellend. Viele Fotos, die als Vorlage für Texturen dienen sollten, waren qualitativ nicht geeignet, sondern unscharf oder mit unerwünschten Reflexionen. Also wurde ein weiterer Termin mit dem Ansprechpartner der Kunsthalle Herrn Jenß vereinbart, da erneut Fotos für Texturen und ein genauerer Bauplan benötigt wurden. Zunächst wurde einen Termin vereinbart, um die Fotos zu schießen. Ein paar Wochen später wurden zwei

Vertreter der Teilgruppe von Herrn Jenß empfangen, der einen genauen Grundrissplan der benötigten Etage zur Verfügung stellte. Auf diesem Plan waren fast alle Maße notiert, die für den Nachbau notwendig waren.

Während der Korrekturarbeiten erwies es sich als effizienter, mit dem Bau der virtuellen Umgebung der Kunsthalle erneut zu beginnen. Nun konnten die einzelnen Räume maßstabsgetreu mit Hilfe der CSG Funktionen des Unreal Editor erstellt werden.

Nachdem nun die Größenverhältnisse der einzelnen Räume der Wirklichkeit ähnlich waren, konnte mit dem Erstellen der Texturen unter Vorlage der Fotos und dem Nachgestalten der einzelnen geometrischen Objekte begonnen werden.

#### 4.2.2 Texturen

Das Erstellen der verschiedenen Boden- und Wandtexturen stellte sich als zeitaufwendig und kompliziert heraus. Es wurde für den Parkettboden beispielsweise eine kachelbare Textur benötigt, das heißt sie muss mehrmals neben- und übereinander gelegt werden können, ohne dass die Ränder des eigentlichen Texturbilds sichtbar sind.

Aus dem Foto, das nur einen Teil des Parkettbodens abbildete, musste eine Textur erstellt wer-

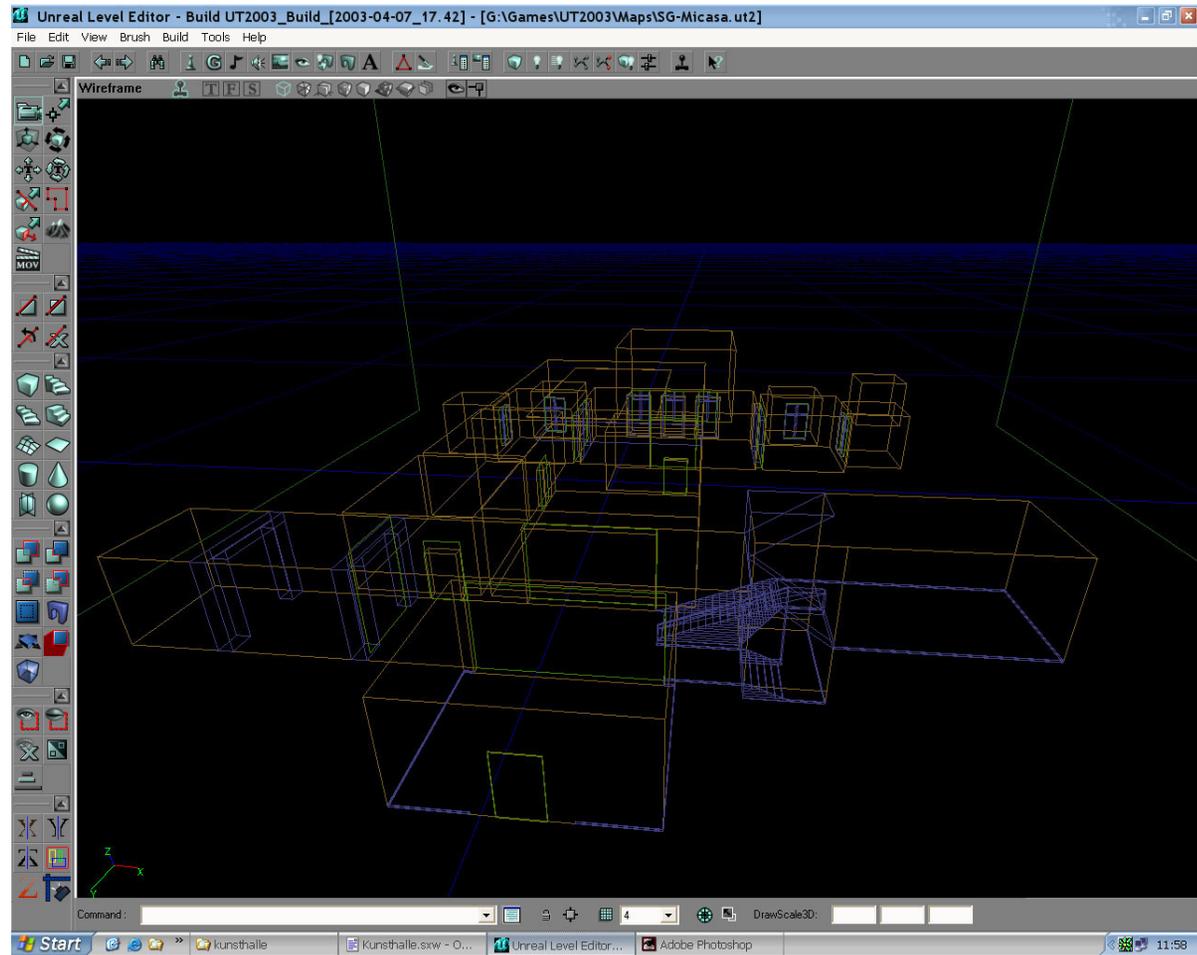


Abbildung II.53: Drahtgittermodell des Erdgeschosses.

den, die den gesamten Raum ausfüllen kann. Durch das Kopieren und Aneinandersetzen verschiedener Bildausschnitte des Parketts wurde die gewünschte Eigenschaft der Kachelbarkeit erreicht.

Ein weiteres großes Problem bestand in der ungleichmäßigen Ausleuchtung der fotografierten Objekte. Generell wird die Farbe und die Helligkeit eines Gegenstandes von vielen Faktoren, wie z. B. der Beleuchtung und der Farbe umgebender Objekte, beeinflusst. Diese Zusammenhänge werden in der virtuellen Umgebung mit Hilfe von unterschiedlichen Beleuchtungsmodellen simuliert. Ist eine Textur in Beleuchtung und Farbgebung inhomogen, so addieren sich die Beleuchtungsverhältnisse der realen, fotografierten Umgebung mit denen der virtuellen. So mussten z. B. Helligkeitsunterschiede entweder durch das Kopieren anderer Stellen oder durch die Anwendung von Masken retouchiert werden. Diese Arbeit kann bei einer einzigen großen kachelbaren Textur oft mehrere Stunden dauern.

In der virtuellen Kunsthalle wurden drei solcher großen Texturen verwendet; der Steinboden im Eingangsbereich, das Parkett und der Mosaikboden in dem hinteren Galeriebereich.

Die Wandtextur, die einer Rauhfaser tapete gleicht, war recht einfach herzustellen. Mit Hilfe eines Programms, das speziell zur Erstellung von

kachelbaren Texturen dient, konnte diese Textur generiert werden. Durch verschiedene Filter und Shadereinstellungen entstand der Eindruck einer Rauhfaser tapete. Das Programm berechnete selber eine Textur, die kachelbar ist und keine Helligkeits- oder Farbunterschiede aufweist.

Eine weitere Schwierigkeit bestand in den Randverzerrungen des Mosaikbodens in einigen Räumen. Da der Mosaikboden bereits als kachelbare Textur erstellt wurde, war es nicht möglich diese Randstreifen direkt in diese Textur einzuarbeiten.

Für die Randstreifen wurden eigene Texturbilder erstellt. Dazu wurde in einer Modellingssoftware ein sehr flaches Objekt mit lediglich einem Polygon erzeugt und mit den Randstreifen als weitere Textur belegt. Nun konnte das Objekt an der entsprechenden Position in der Map und lediglich eine halbe Unrealeinheit über dem eigentlichen Boden mit der Mosaiktextur platziert werden. So entstand der Eindruck, dieser Randstreifen befindet sich direkt auf dem Boden. Damit der Benutzer des Caves hinterher keine automatische Bewegung nach oben macht, um den kleinen Höhenunterschied zwischen Mosaikboden und Randstreifen auszugleichen, wurde die Kollisionsberechnung für die Randstreifen deaktiviert. Mit dieser Methode entstanden auch die Risse auf dem Mosaikboden, sowie die Hinweisschilder an der Treppe und die schwarzen Streifen auf dem



Abbildung II.54: Kachelbarkeit des Parketts.

Parkettboden in den Nebenräumen. Hiermit war die Arbeit am Fußboden der Kunsthalle weitestgehend abgeschlossen.

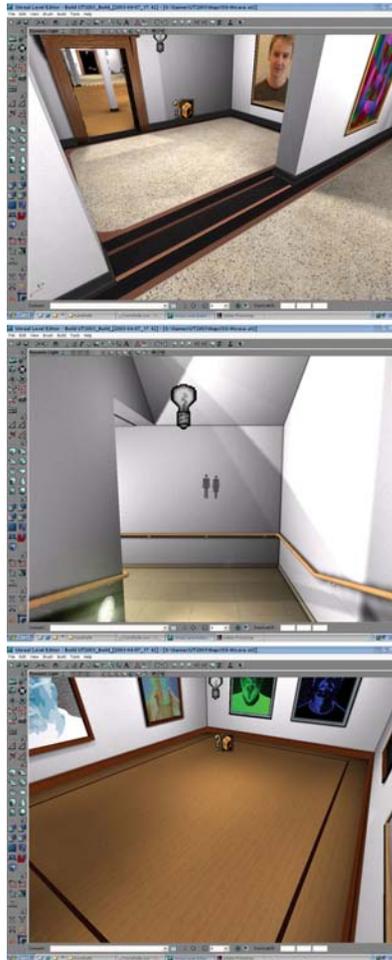


Abbildung II.55: Zierelemente am Fußboden.

#### 4.2.3 Geometrische Objekte

Die verwendete Grafik-Engine verfügt nicht über Effekte wie Bumpmapping oder Displacement Mapping, deshalb mussten die Details der großen Holztüren in der Kunsthalle geometrisch herausgearbeitet werden und konnten nicht von der Grafik-Engine anhand der Texturen berechnet werden. Unter Vorlage der Fotos aus der Kunsthalle wurden sämtliche Türen in einer Modellingssoftware erstellt. Nachdem die Geometrie herausgearbeitet war, mussten die Türen texturiert werden. Da einzelne Fotos nicht aussagekräftig genug waren und nicht den gesamten benötigten Bereich abbildeten, wurden verschiedene einzelne Oberflächen unterschiedlicher Fotos extrahiert und auf die entsprechende Stelle der Tür gelegt. Damit wurden detaillierte Modelle der Holztüren geschaffen, die sehr viele verschiedene Texturen benötigten. Durch die vielen Details enthielten die Modelle jedoch zu viele Polygone, um in der Unreal Engine 2 dargestellt werden zu können. Die Anzahl der Polygone musste wieder reduziert werden, ohne dabei das Aussehen der Türen zu verändern. Es entstand eine Version mit 4000 Polygonen, die aus fast 20.000 Polygonen errechnet wurde. Diese reduzierte Version war geeignet, um in der Unreal Engine in Echtzeit zum Einsatz zu kommen.

Weiterhin musste, ebenfalls aus Performancegründen die Anzahl der verwendeten Texturen für diese Objekte verringert werden. Dies geschah mit dem Verfahren des Texture Baking, bei dem die indexModellingsoftware Modellingsoftware alle, auf den einzelnen Oberflächen des Objekts verwendeten, Texturen in eine Textur zusammen rendert. Dadurch reduziert sich allerdings auch die Auflösung der einzelnen Texturen. Da aber drei solcher Texturen für dieses eine Objekt erstellt wurden, ist ein Qualitätsverlust bei den Texturen kaum zu erkennen.

Des Weiteren musste die Größe der Türen genau auf die virtuelle Umgebung der Kunsthalle abstimmt werden. Hierzu wurde der im Unreal Editor erstellte Grundriss der Kunsthalle in die Modellingsoftware importiert und die Türen dementsprechend angepasst. So konnte sichergestellt werden, dass das dreidimensionale Objekt beim Importieren in den Unreal Editor genau in das Level passt.

Die Holztüren wurden in drei unabhängigen Objekten in die Unreal-Engine importiert; der Rahmen und beide Türen. So war es möglich einige Türen geöffnet und andere geschlossen in der Umgebung zu präsentieren. Unter dieser Voraussetzung wäre es auch möglich gewesen, die Türen durch den Benutzer im Cave öffnen und schließen zu lassen. Diese Interaktionsmöglichkeit hätte allerdings separat implementiert werden müssen.



Abbildung II.56: Beispiel für TextureBaking.

Für die Fußbodenleisten wurde zunächst davon ausgegangen, es könnten Grundformen der Fußbodenleisten erstellt und diese dann in Unreal Editor aneinandergesetzt werden. Demnach würde ein gerades Segment, ein Eckstück nach innen und ein Eckstück nach außen von den Fußbodenleisten benötigt werden. Diese könnten dann kombiniert werden, dass sie um die Innenwände aller Räume in der Kunsthalle laufen.

Es stellte sich als Problem heraus, dass die einzelnen Stücke der Leisten an verschiedenen Stellen skaliert mussten. Die Geometrie der Objekte wurde dadurch lediglich im Maßstab verändert,

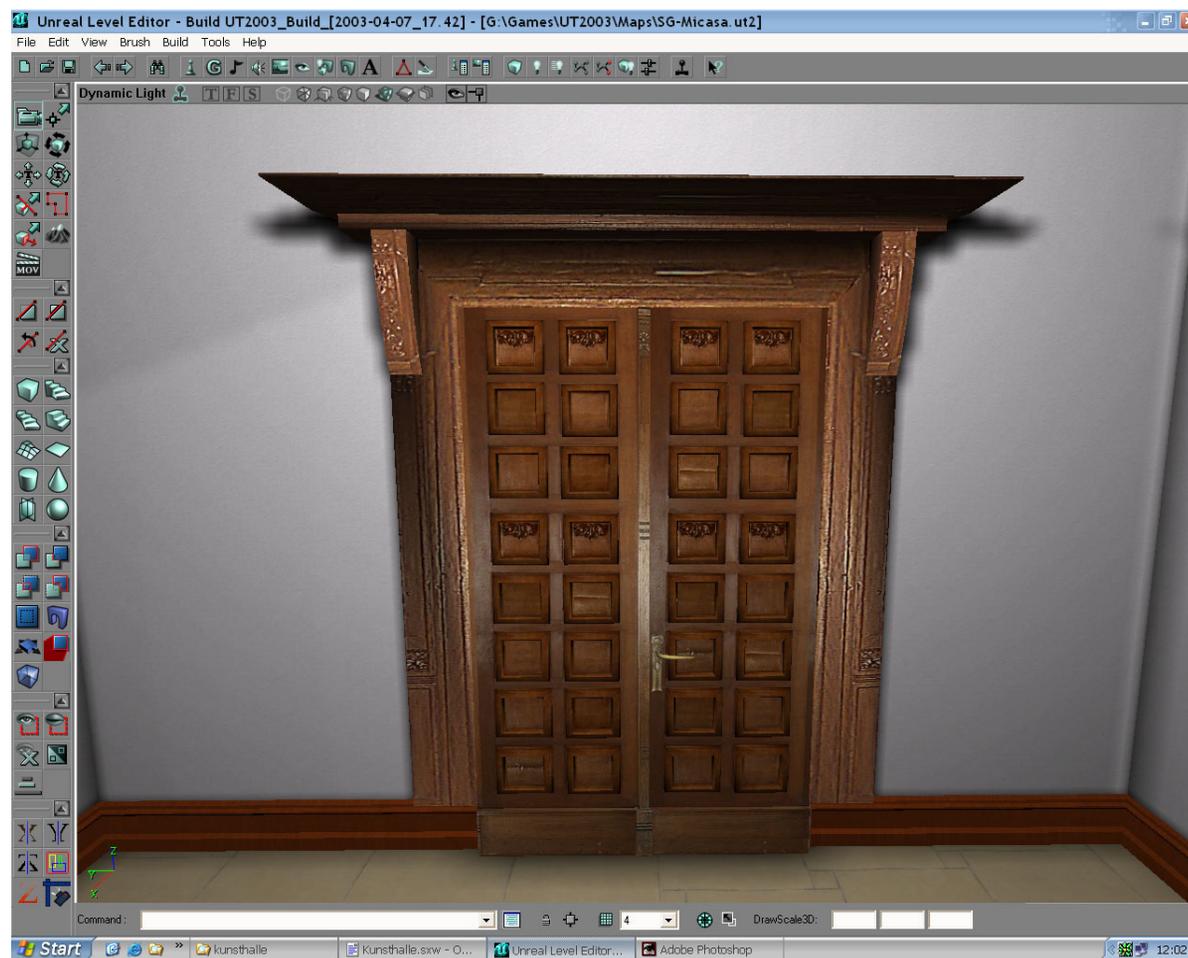


Abbildung II.57: Eine Holztür in der Kunsthalle.

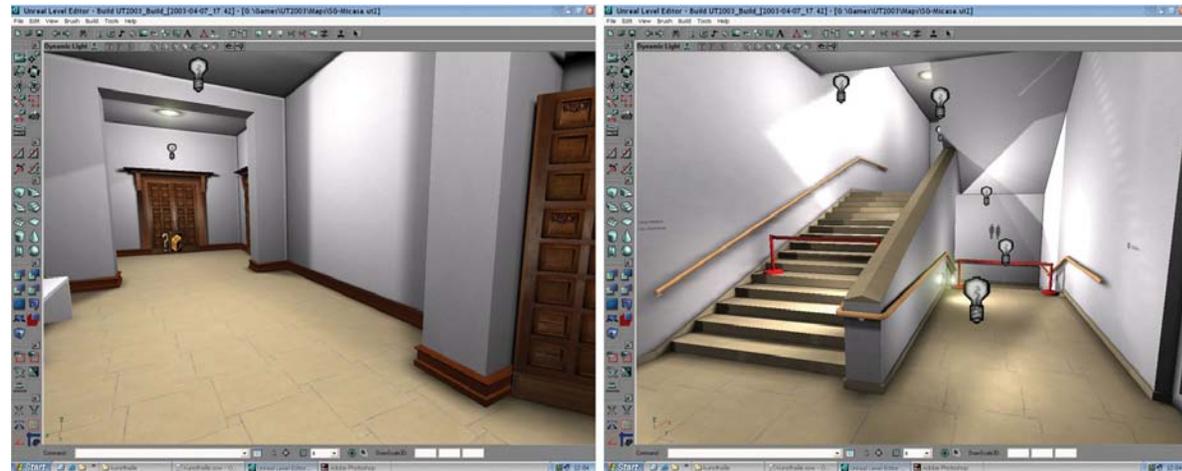


Abbildung II.58: Fußbodenleisten und Handläufe.

die Textur würde jedoch ebenfalls skaliert werden und so würden sichtbare Texturunterschiede auf den Fußbodenleisten zurückbleiben. Um dies zu verhindern, wurde jeder Teil der Fußbodenleisten individuell nachgebaut und in der Modellingssoftware an den importierten Grundriss angepasst. Schließlich wurden die zusammenhängenden Fußbodenleisten als ein Objekt exportiert und im Unreal Editor in das Level gesetzt.

So entstanden auch die Handläufe der Treppen.

Nun war das Erdgeschoss der Kunsthalle quasi schon nachgebaut worden. Die Teilgruppe hatte sich dazu entschlossen, die virtuelle Kunsthalle zur Vorstellung des Projekts zu nutzen. Es wur-

den Fotos von jedem Projektmitglied gemacht, die dieses dann nach Belieben verfremden konnte. Auf diese Weise konnte sich jedes Projektmitglied darstellen und seiner Fantasie freien Lauf lassen. Zum Aufhängen der verfremdeten Portraits wurden entsprechende Bilderrahmen benötigt. Diese wurden wie alle anderen Objekte auch mit einer Modellingssoftware erstellt und in Unreal Editor importiert, wo dann die einzelnen verfremdeten Bilder eingefügt wurden.

Der nächste Schritt bestand in der Ausleuchtung der virtuellen Kunsthalle. Um den Lichtquellen, die im Unreal Editor in die Map eingefügt werden, einen sinnvollen Ursprung zu geben, wur-

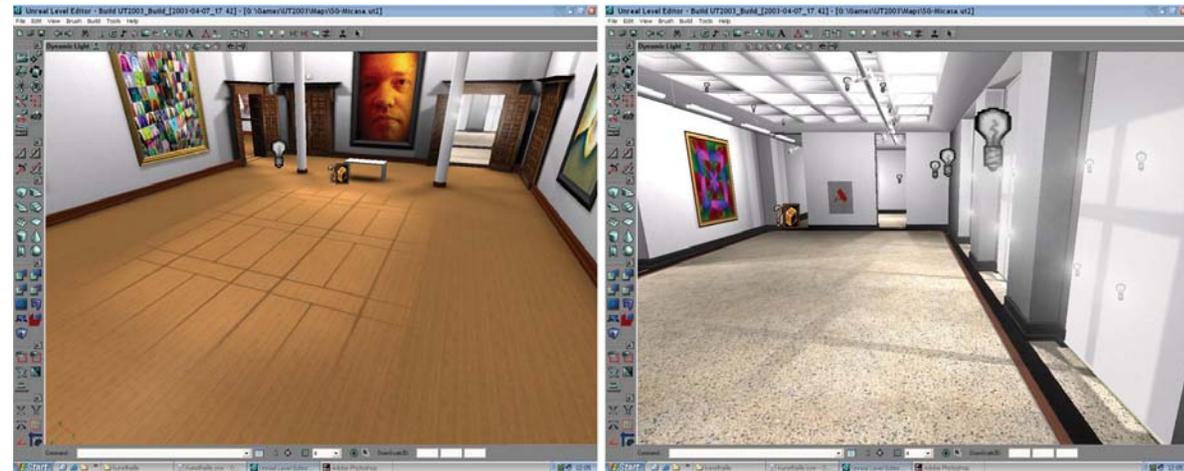


Abbildung II.59: Projektoren für realistischeres Licht und dazugehörigen Schattenwurf.

den verschiedene Lampen gebaut. Dabei wurde teilweise von den Originalen aus der Kunsthalle abgewichen, um die Räume im Level besser ausleuchten zu können. Die Lichtquellen in Unreal Editor sehen nicht genau wie reales Licht aus. Es mussten also verschiedene Lichtquellen kombiniert werden, damit der Eindruck entsteht, es handle sich um eine Lichtquelle, die in der Wirklichkeit existiert.

Um einige Effekte bei Schattenwurf und Licht realistischer gestalten zu können, wurden beispielsweise zwei Projektoren eingesetzt.

Mit Projektoren können Licht und Schatten in der Engine wesentlich genauer und detaillierter

erstellt werden als mit einer gewöhnlichen Lichtquelle des Unreal Editor.

Mit den beiden verwendeten Projektoren wird der Lichteinfall und der Schattenwurf der Sonne simuliert. Dieser Bereich der virtuellen Kunsthalle kann nur unter Verwendung von Projektoren realistisch ausgeleuchtet werden.

Um die Umgebung noch realistischer wirken zu lassen, wurden verschiedenen Oberflächen Reflektionen zugewiesen. Reflektionen werden in der Engine nicht in Echtzeit berechnet und müssen deshalb manuell eingefügt werden.

Sie wurden auf den verwendeten Glasmaterialien am Eingangsbereich und der Glastafel benutzt.

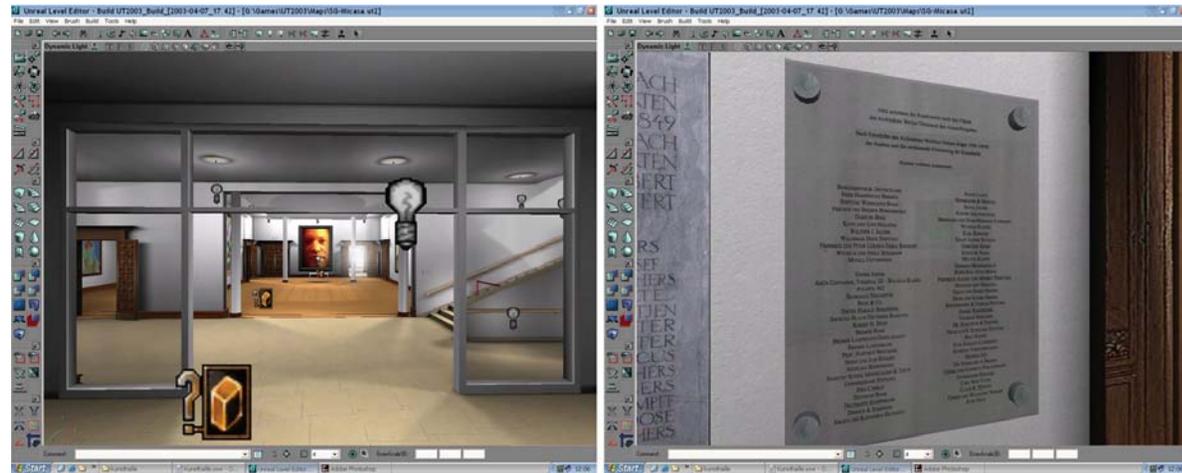


Abbildung II.60: Reflektionen für Glasoberflächen.

Einfache Lichtreflexe wurden den Fußböden zugewiesen, sodass die Person im Cave den Eindruck bekommt, dass das ausgesendete Licht der Lampen auf dem Boden reflektiert. Es wurden keine Spiegelungen eingebaut, da der reale Boden der Kunsthalle wenig reflektiert und im Cave nur kleine Teile des Fußbodens sichtbar sind, sodass diese

geringe Reflektion nicht zu erkennen wäre.

Abschließend wurde die virtuelle Kunsthalle in das Portal eingefügt und mit Hilfe einer Warpzone und Teleportern an das Portal angeschlossen, damit die Person im Cave die alle vorhandenen Umgebungen samt der Kunsthalle über das Portal erreichen kann.

### 4.3 Playground

Der Playground ist aus dem Bedarf einer Testumgebung für die Ein-/Ausgabegeräte wie beispielsweise der A.R.M. (vgl. Kapitel 3.2) entstanden. Außerdem mussten die erstellten Funktionen zum Bewegen von Objekten in einer virtuellen Umgebung (vgl. Kapitel 3.3) überprüft werden.

Diese virtuelle Umgebung sollte dann am Projekttag präsentiert werden, damit Personen im Cave neben der Tanzmatte noch auf andere Weise mit der virtuellen Welt interagieren können.

#### 4.3.1 Die erste Version

Zur Entstehung des Levels wurden im Unreal Editor drei Räume erstellt, von denen zwei direkt übereinander liegen und durch eine Treppe verbunden wurden. In dieser Map wurden ursprünglich nur zwei Texturen (Fußboden/Decke und Wände) verwendet, wodurch das Level sehr eintönig wirkte. Zusätzlich wurden Gegenstände in einer Modellingsoftware erstellt, die sich in Form und Farbe unterscheiden. Es handelt sich hierbei um gelbe, blaue, grüne und rote Zylinder, Kegel, Quader und gestreckte Halbkugeln, die als bewegliche Objekte anscheinend zufällig in beiden Räumen anhand des Unreal Editor verteilt wurden. Somit entsprach die Playground Map den Anforderungen an eine Testumgebung.

Um das Level noch interessanter zu gestalten, wurde ein Zielort für die beweglichen Gegenstände erschaffen. So konnte innerhalb der virtuellen Umgebung und während einer Testphase ein kleines Spiel absolviert werden. Das äußere Erscheinungsbild des Ortes, an den die Gegenstände getragen werden sollten, ist der Tanzmatte (vgl. Kap 3.1) sehr ähnlich. Er unterscheidet sich von ihr nur durch die unterschiedlichen zu den beweglichen Gegenständen passenden Farben. Aufgabe ist es demnach jeden Gegenstand auf ein Feld der Plattform zu stellen, das seiner eigenen Farbe entspricht.

In dieser virtuellen Umgebung mit spielerischer Komponente konnten die benötigten Tests durchgeführt werden. So wurde eine feine Abstimmung der Ein- und Ausgabegeräte für die korrekte Positionierung der Gegenstände im virtuellen Raum erreicht.

#### 4.3.2 Die finale Umgebung

Die virtuelle Umgebung sollte nun interessanter und abwechslungsreicher gestaltet werden. Es wurden hauptsächlich Texturen und Static Meshes verwendet, die bereits für das MZH- und Kunsthallenlevel erstellt wurden. So erhielt das Testlevel eine Universitätsatmosphäre, deren konkrete Entsprechung jedoch nicht existiert.

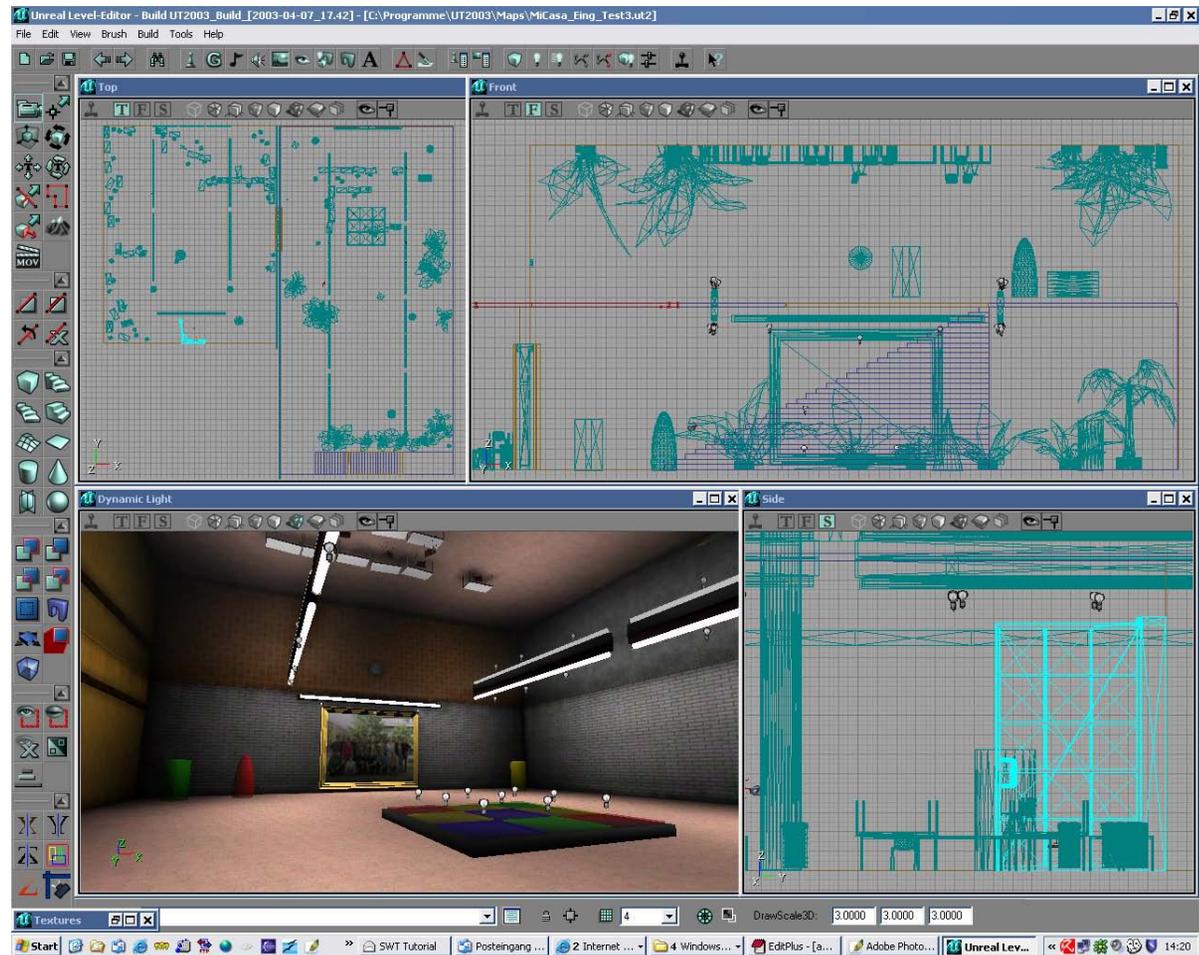


Abbildung II.61: Die Playground-Welt im Unreal Editor.



Abbildung II.62: Die Zielplattform für die beweglichen Gegenstände.

Wände, Decken und Fußböden erinnern stark an die erste Ebene des MZH. Nach dem Texturieren der BSP-Geometrie wurden der Map neben den beweglichen Gegenständen und der Zielplattform weitere Karmaobjekte hinzugefügt. Dabei handelte es sich um eine große Anzahl von Tischen und Stühlen, Mülleimer und Pflanzen wie sie im MZH vorkommen. Dadurch wirkte die virtuelle Umgebung bereits realistischer. Einige von diesen Gegenständen sind, wie die Gegenstände für das Testspiel, beweglich und können umgeworfen bzw. durch die Räume bewegt werden. Welche Gegenstände tatsächlich beweglich sind, muss der Benutzer des Caves herausfinden. Um unser Projekt in dieser Umgebung zu präsentieren, wurde ein Kunstwerk bestehend aus drei MiCasa-Logos in unterschiedlichen Stilen erstellt und ein Gruppenfoto aufgehängt.

Schließlich wurde dieses Level über das Portal zugänglich gemacht. Hierzu wurde, wie zuvor für die MZH- und die Kunsthallenmap, eine Tür in einer Modellingssoftware erstellt, die den Ein- und Ausgang bildete. Sie sollte wie die beiden anderen Türen den Charakter der dahinter verborgenen Umgebung widerspiegeln. Das Design der Tür wurde an das Aussehen der Zielplattform angepasst und so ein Wiedererkennungswert geschaffen. Die Tür wurde dann sowohl in das Portal als auch in den Playground eingefügt. Sie wurde in dem Raum in der unteren Etage positioniert, der keinen wei-

teren Raum über sich hat, sodass der Blick des Betrachters zuerst auf die MiCasa-Logo-Bildserie fällt.

Der Playground ist keine Nachbildung existenter Räumlichkeiten. Obwohl sie realistisch aussehen soll, können Gegebenheiten gezeigt werden, die es in der Realität nicht gibt. Das Ziel war es, die Person im Cave mit einer Situation zu konfrontieren, die allen täglichen Erfahrungen und damit Erwartungen an die Realität widerspricht. Im Cave besteht die Möglichkeit, Teile der Realität extrem genau nachzubilden, für den Betrachter eine Realität zu erzeugen, ihn aber trotzdem mit unmöglichen Gegebenheiten zu irritieren.

Um dies zu erreichen, wurden zunächst die Funktionen von Decke und Fußboden in dem Raum in der ersten Etage umgekehrt. Die Decke enthielt Texturen und Gegenstände, die in der Realität nur auf Fußböden zu sehen sind wie Stühle, Tische, Mülleimer. Auf dem Fußboden wurden hingegen Lampen angebracht. Auch Wanddekorationen wie eine Wanduhr wurden umgedreht und schienen auf dem Kopf zu stehen. Als einzige Ausnahme und zur noch größeren Verwirrung des Betrachters sind die Spielgegenstände auf der Fläche geblieben, die erfahrungsgemäß einen Fußboden darstellt. Die Konsequenz war ein Eindruck der Person im Cave, als wäre der gesamte Raum gedreht worden bzw. man selber an der Decke laufe. Ein Widerspruch besteht jedoch in der Po-



Abbildung II.63: Die MiCasa-Logo-Bildserie.

sition der Spielfiguren, die Ratlosigkeit bewirkt. Es wurde noch ein weiterer Aspekt eingearbeitet, um diese Unlogik zu unterstreichen. Der ursprüngliche Fußboden, auf dem die Lampen montiert sind und auf dem der Benutzer sich durch diesen Raum bewegt, wurde durch eine Glasfläche ersetzt. Betritt nun eine Person im Cave diesen virtuellen Raum, sieht er einen vermeidlichen Fußboden, mit Fliesen, Stühlen, Tischen etc. weit über sich und Lampen auf seiner Fußhöhe. Außerdem kann er durch das Glas hindurch den Raum unter ihm und eine enorme Tiefe bis zu dessen Boden sehen. Jeder Mensch hat die Erfahrung gemacht, dass Glas - vor allem bei großen Flächen - zerbrechlich ist und dieser Glasboden demnach nicht in der Lage ist, einen Menschen zu tragen. Dennoch muss die Person im Cave diese virtuelle Glasfläche betreten, um die dort liegenden Gegenstände für das Spiel zu holen, deren Zielplattform sich eine Etage unter diesem Raum befindet. Der Raum, in dem die Zielplattform liegt, hat dementsprechend eine Glasdecke. Für den Betrachter erscheint diese jedoch zunächst wie eine spiegelnde Oberfläche, da dort auf dem Kopf stehende Tische, Stühle etc. zu sehen sind. Erst nachdem sich die Person im Cave im Raum umgeschaut hat, stellt sie fest, dass es sich nicht um eine Spiegelung handeln kann, da beispielsweise die Zielplattform für das kleine Spiel wie auch das Gruppenfoto an der Wand nicht gespiegelt

werden. Daraus schließt der Betrachter, es könne keine Spiegelung und müsse eine durchsichtige Glasscheibe sein. Dies widerspricht jedoch seinen Erfahrungen, weil dann die Einrichtungsgegenstände alle verkehrt platziert wären. Um diese Irritation aufzuklären, muss die Person im Cave die Treppe zu dem oberen Raum benutzen, um sich dort mit weiteren scheinbaren Widersprüchen konfrontiert zu sehen.

Letztendlich war es nicht das Ziel, den Betrachter in Konflikte durch die erwähnten Widersprüche zu bringen. Vielmehr war die Intention, dem Betrachter zu zeigen, dass sämtliche Umgebungen trotz allem Realismus aufgrund neuester Beleuchtungstechniken, guter Texturqualität und maßstabsgetreuer BSP-Geometrie Virtualität bleiben. Es ist Willkür der Erschaffer dieser Level, welche Informationen gezeigt und welche verborgen bleiben wie bei den bereits vorherrschenden Massenmedien.

## 4.4 Das Portal

### 4.4.1 Entstehungsgeschichte

Nachdem sich die zwei großen Teilprojekte Kunsthalle und MZH manifestiert hatten, fehlte eine passende Methode diese beiden virtuellen Welten miteinander zu verbinden. Es existierte

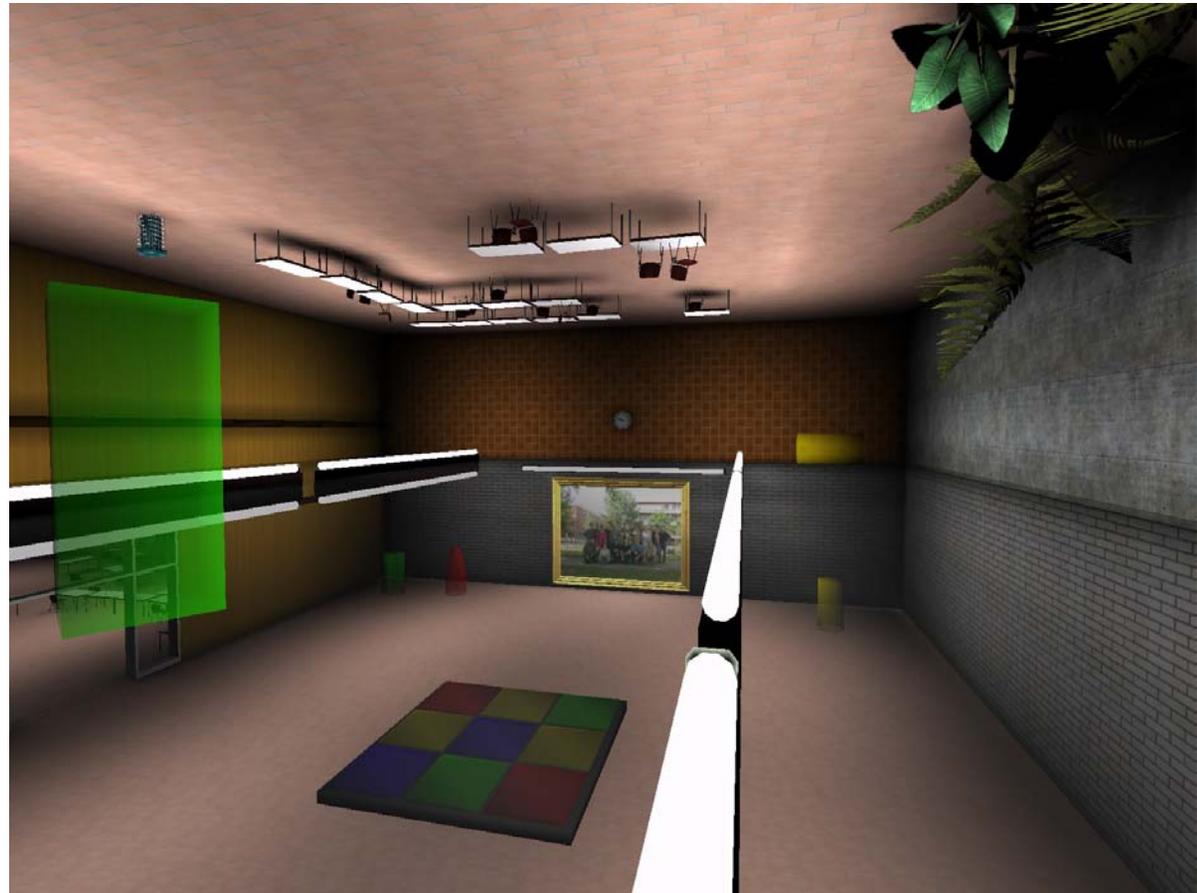


Abbildung II.64: Der Blick durch die Glasdecke.

die Idee eines relativ einfach zu realisierenden automatischen Wechsels der Map, der stattfinden sollte, sobald die Person im Cave sich an einen bestimmten Ort der Map begibt. Allerdings hatte diese Idee den Nachteil, dass während der Ladezeit ein Standardbildschirm der Engine mit dem Namen der Map angezeigt wird, der überhaupt nicht zu den erstellten virtuellen Umgebungen passt.

Der erste Ansatz der folgte, war ein Umschreiben des Quellcodes der Unreal Engine 2, um den Ladebildschirm an unsere Bedürfnisse anzupassen. Leider war dies nicht möglich, da der Quellcode der Engine für diesen Zweck nicht zur Verfügung stand. Auch eine freundliche Anfrage an die Hersteller der Engine, das Universitätsprojekt mit einer kostenlosen Lizenz zu unterstützen, war nicht erfolgreich.

Ein weiteres technisches Problem war, dass es keine sinnvolle direkte Verbindung zwischen dem MZH und der Kunsthalle gab. Es fehlte ein weiteres Element, welches die beiden Level miteinander verbinden sollte. So entstand das Portal. Bei dem Portal handelt es sich um ein drittes Level, welches sehr stark an das Konstrukt aus dem Film „Matrix“ erinnert. In unserer Version bietet dieses Konstrukt allerdings die Möglichkeit durch türartige Portale die damit verbundenen Level zu betreten.

Das technische Problem des Mapwechsels wur-

de dann dadurch gelöst, dass wir alle Level in dieser Portalmap vereinten und so einen direkten Sprung von einer virtuellen Umgebung zur anderen innerhalb der Map mit Hilfe von Teleportern durchführen konnten. Dadurch hatte die Map eine sehr viel höhere Ladezeit, musste allerdings nicht wieder gewechselt werden. Für die Präsentation am Projekttag erwies sich dies als sehr praktikabel.

In den letzten Monaten vor dem Projekttag wurde das Portal dann durch eine weitere Tür erweitert, die zum Playground führte (vgl. Kapitel 4.3).

#### 4.4.2 Umsetzung

Um die Illusion der weißen Unendlichkeit innerhalb des Portals zu schaffen, wurde ein relativ zur Spielergröße gigantischen würfelförmigen Raum erstellt, dessen vier Wände, sowie Decke und Boden mit einer unbeleuchteten, absolut weißen Texturen belegt wurden. Inmitten dieses unendlich scheinenden weißen Würfels sieht man drei Türen (siehe Abbildung II.67). Rechts daneben, nur durch Warpzones mit dem eigentlichen Portal verbunden, liegen die drei virtuellen Welten; die erste Ebene des MZH, die Kunsthalle und der Playground (von oben nach unten).

Um eine weitere Illusion zu schaffen, wurden Warpzones verwendet, die dem Benutzer das Ge-



Abbildung II.65: Anordnung der Portaltüren von oben.

fühl geben, die Geometrie der Level läge direkt hinter der Tür. Gleichzeitig ist es möglich, sich um die Tür herumzubewegen und von der anderen Seite hindurchzublicken.

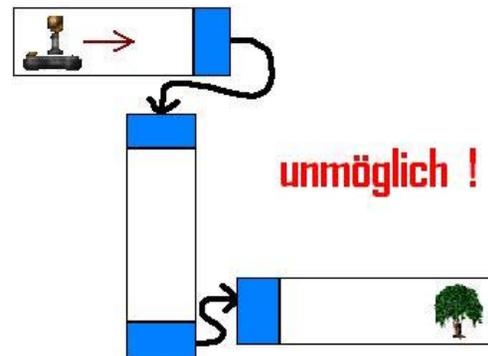


Abbildung II.66: Der Spieler oben links kann den Baum unten rechts nicht durch zwei Warpzones hindurch sehen.

#### 4.4.3 Die Anordnung der Portaltüren

Abbildung II.66 zeigt, dass es der Unreal Engine 2 nicht möglich ist, einen Blick durch zwei Warpzones hindurch zu berechnen. Deshalb mussten die Portaltüren in einer Reihe angeordnet werden.

Es gab zudem das Problem, dass einige Static Meshes durch die Warpzone gesehen werden konnten. Diese Schwierigkeit wurde mit Anordnung der Türen in einer Reihe ebenfalls gelöst.

#### 4.4.4 Warpzones

Eine Warpzone ist ein spezielles Merkmal der Unreal Engine 2, welches es dem Spieler ermöglicht, die Geometrie des Raumes zu verknüpfen. So entsteht die Illusion, dass ein Teil des Levels in einen anderen Teil der virtuellen Umgebung übergeht (siehe Abbildung II.69).

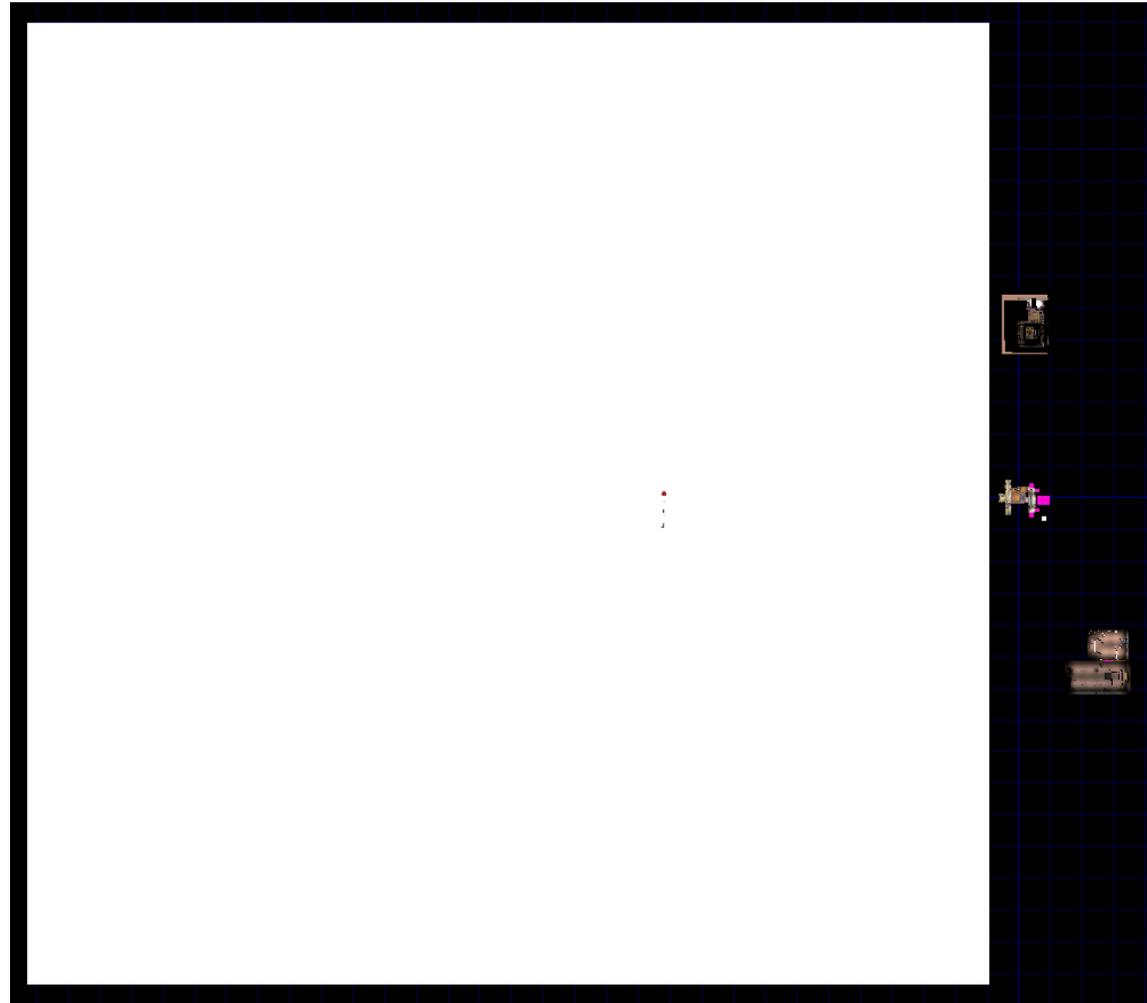


Abbildung II.67: Das Portal-Level aus der Vogelperspektive.



Abbildung II.68: Zwei Beispiele für Türen, die aus dem Portal in andere Welten führen.

Bei der Erstellung von zwei zusammenhängenden Warpzones ist zu beachten, dass diese an beiden Orten in der Map die gleichen Ausmaße haben müssen, damit der Effekt der Warpzone glaubhaft wird. Ist dies nicht gewährleistet, kommt es zum sogenannten HOM (Hall of Mirror Effekt). Dabei können Teile des Bildes nicht gerendert werden und dadurch sehen sie so aus, als befände sich der Betrachter in einer Halle von Spiegeln. Die Warpzones, die im Portal verwendet wurden, haben nur eine Tiefe von einer Unrealeinheit. Breite und Höhe sind an die Türen angepasst, da diese jeweils unterschiedliche Maße haben.

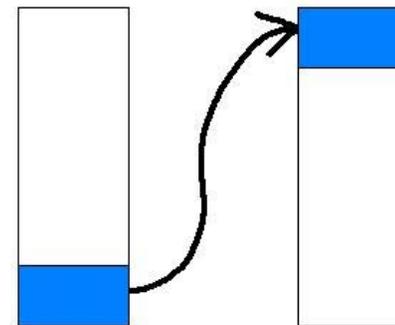


Abbildung II.69: Der blaue Bereich ist eine Warpzone.

Hier handelt es sich im Prinzip um zwei Warpzones, die in entgegengesetzte Richtungen ausgerichtet sind, da beim Herumgehen um eine Portaltür auch von hinten in das dazugehörige Level gesehen werden sollte.

Das Portal hat mehrere Technik- und Designprobleme gelöst und die interessante Technologie der Warpzones zur Geltung gebracht, was auf dem Projekttag positiv aufgefallen ist.



Abbildung II.70: Zur besseren Orientierung wurde zwischen die Portaltür des MZH und der Kunsthalle ein Wegweiser aufgestellt.

## 5 Finanzierung

### 5.1 Marketing und Sponsorensuche

Die Idee, die Entwicklung des Caves von Sponsoren unterstützen zu lassen, war tatsächlich leider eine Notwendigkeit. Nachdem bereits der erste Prototyp des Caves eine ganze Menge Geld verschlungen hatte und die Planungen für den Bau des eigentlichen Cave Ähnliches vermuten ließen, musste versucht werden, Sponsoren für das Projekt zu finden.

Beispielsweise unterstützte ein Grafikdesigner das Projekt, indem er wichtige Tipps für die weiteren Pläne bezüglich des Marketings und der Gestaltung der Präsentationsmaterialien gab.

Für das Marketing des Cave und die erste Zwischenpräsentation in Zusammenarbeit mit dem Projekt MiCaDo ein Flyer erstellt, in dem die grundsätzlichen Kriterien und noch benötigten Materialien, sowie Kontaktinformationen aufgeführt waren. Dieser Flyer wurde sowohl bei der ersten Präsentation als auch an andere private sowie berufliche Kontakte verteilt.

Im Laufe des folgenden Jahres wurde außerdem eine Mappe erstellt, in der sich die beiden Teilprojekte MiCasa und MiCaDo vorstellten. Der Druck der Mappen konnte mit freundlicher Unterstützung der Werbeagentur Alves realisiert werden.

Durch das Engagement eines Kommilitonen aus dem Projekt MiCaDo erklärte sich die Firma Beck's bereit, das Projekt mit Rechnern und Getränken für den Projekttag zu unterstützen.

Dank zoom, einem Beamerverleih aus Bremen, stand eine ausreichende Anzahl Beamer für die Präsentation zur Verfügung. So konnte der Cave planmäßig im Studierhaus vorgestellt werden.

Gegen Ende des dritten Projektsemesters war der Druck der Mappen abgeschlossen und es konnte damit begonnen werden, Unternehmen anzuschreiben und die Mappen zu verteilen.

Leider war das Interesse, ein studentisches Universitätsprojekt zu unterstützen, gering bis nicht vorhanden. So beschränkten sich alle Unterstützungen auf Sachmittel, wie etwa die geliehenen Beamer oder der Druck der Mappe.

### 5.2 Kosten des Caves

Der Holzrahmen und dessen Anfertigung haben mit über 1 200 Euro den größten Teil der Ausgaben ausgemacht. Dazu kommt die Bespannung der Seitenwände mit Projektionsfolie, die circa 400 Euro gekostet hat. Des Weiteren kamen Kleinbeträge für Schrauben, Gummiringe, Niete und Ähnliches an Aufwendungen dazu, die in der

Gesamtbetrachtung etwa 70 Euro ausmachen.

Den kleineren Teil der Ausgaben machten die Eingabegeräte aus. Für den Prototypen des Armes waren circa 60 Euro Entwicklungskosten notwendig. Die Produktion eines zweiten Armes würde nur noch etwa 15 Euro kosten. Die Tanzmatte hat in der Herstellung 150 Euro gekostet, die Funkmaus 30 Euro.

Das artec stellte einen Switch im Wert von 100 Euro zur Verfügung. Somit ergibt sich ein Gesamtbetrag rund 2000 Euro, sofern die Entwicklungs- und Baukosten für den Prototyp unberücksichtigt bleiben.

Ebenso unberücksichtigt bleiben in dieser Aufstellung die Kosten für die Präsentation des Cave auf dem Projekttag und die Unkosten aus dem

Bereich Marketing. Diese belaufen sich auf 58 Euro zuzüglich der Druckkosten der Mappe.

Hinzu kamen diverse Kosten, die durch den Einsatz privater Hard- und Software nicht vom Projekt getragen werden mussten. Bei einem eventuellen Neu- beziehungsweise Nachbau des Cave sollten diese aber in der Kalkulation berücksichtigt werden.

Vier leistungsfähige Rechner mit aktuellen 3D Grafikkarten der oberen Leistungsklasse, Netzkabel, ein Switch, Verlängerungskabel und Mehrfachsteckdosen wurden aus privaten Beständen gestellt. An Software sind Windows-Lizenzen und das Spiel Unreal Tournament 2003 zu finanzieren. Zu berücksichtigen sind auch die Arbeitskosten für Herstellung, Rechnereinrichtung, Transport, Aufbau etc.

## 6 Der Projekttag

### 6.1 Die Vorbereitungen

#### 6.1.1 Das Studierhaus

Im Studierhaus wurde der Cave während des Projekttag präsentiert, sodass Interessenten den Cave betreten und sich in einer virtuellen Umgebung bewegen und mit ihr interagieren können.

Auf der Suche nach einem geeigneten Raum für die Präsentation des Caves am Projekttag stehen nicht viele Möglichkeiten zur Verfügung. Der Raum muss groß genug sein, um den Cave mit den Beamern unterzubringen, es wird demnach eine quadratische Fläche von 13 Metern Seitenlänge benötigt. Ebenfalls muß eine schnelle Erreichbarkeit vom Stand in der ersten Ebene des MZH gegeben sein, damit die Projekttagbesucher sich nicht zu weit vom Geschehen des Projekttag entfernen müssen. Ein weiterer Faktor für die Raumwahl ist die Möglichkeit der Verdunklung. Für eine ausreichend zufriedenstellende Projektion wird eine gute Abschirmung des Tageslichts benötigt.

Als Optionen sind die Räume 5210 im MZH und B1400 im GW2 in Betracht gezogen worden, außerdem das Studierhaus. Nach ausgiebigen Messungsdurchläufen in den verschiedenen Lokalitä-

ten entschied sich die zuständige Gruppe für das Studierhaus, da hier das Optimum an Platz und Lage zu finden war.



Abbildung II.71: Letzte Arbeiten in der Nacht vor dem Projekttag



Abbildung II.72: Schrittweiser Aufbau des Caves.

In mehreren Gruppensitzungen wurde der Bedarf an Ausrüstung für den Projekttag im Studierhaus ermittelt. Wichtige Kriterien hierbei waren unter anderem die ausreichende Stromversorgung, ein Internetzugang zur Übermittlung von Webcambildern in das MZH und genügend Kabel zur Vernetzung und Stromversorgung der Rechner. Fehlende Kleinteile wurden schnell beschafft. Zum Transport der Ausrüstung standen jeweils ein Transportwagen aus dem artec und der Studierwerkstatt zur Verfügung.

Glücklicherweise konnte der Schlüssel zum Studierhaus bereits am Tag vor dem Projekttag abgeholt werden. So konnten der Aufbau des Caves, der Rechner und einiger Beamer bereits ein Tag früher beginnen. Außerdem konnte das Projekt unerwartete Probleme weitgehend ohne Zeitdruck lösen.

Der Aufbau des Caves hat keine Probleme aufgeworfen. Zuerst wurden zwei Seitenwände aufge-

baut, die dann mit jeweils zwei weiteren Trägern zum Cave verbunden werden. Hiernach erfolgte die Bespannung der Seiten, sowie die Auswahl der Eingangsseite.

Anschließend geschah der Aufbau der Beamer, Webcams und Computer. Der Aufbau der Infrastruktur hat kaum Probleme bereitet, da diese vorsorglich bereits einen Monat vorher getestet wurden. Die softwaretechnischen Installationen haben sich als wesentlich problematischer erwiesen. In einer Nachtschicht einiger Projektmitglieder konnten jedoch alle Probleme beseitigt werden, sodass zu Beginn des Präsentationstages ein voll funktionsfähiger Cave zur Verfügung stand.

Die Navigation innerhalb des Cave erfolgte anhand der Tanzmatte (vgl. Kapitel 3.1). Am Vorabend des Projekttag wurden die Gummipuffer, die zur Rückfederung einer Platte nach einem Kontakt dienen, mittels Paketband aufgepolstert. Dadurch sollten sie der bevorstehenden Belas-



Abbildung II.73: Aufpolsterung der Puffer mit Paketband kurz vor dem Projekttag.

tung durch den Besucheransturm besser standhalten (siehe Abbildung II.73).

Wie sich herausstellte, war diese Lösung bestens geeignet, denn während des Projekttag gab es keinen Dauerkontakt eines Feldes durch Überbelastung der Gummipuffer. Nach Verlassen eines Feldes wurde dies problemlos in die Ausgangsposition zurückgehoben.

Die Verdunklung der Räumlichkeiten verlief erfreulich problemlos. Die Außenjalousie und der Vorhang an der Hauptfensterwand, sowie die manuell angebrachte Verdunklung an den verbleibenden Fenstern erwies sich als sehr gute Lösung.

Desweiteren waren im Studierhaus ebenso wie am Stand Informationsplakate in Form einer Wandzeitung angebracht. Eine Webcam übertrug im Abstand einiger Sekunden Bilder ins Internet, die im MZH präsentiert werden sollten.

### 6.1.2 Der Stand

Jedes Informatikprojekt, das bereits vier Semester existiert, erhält am Projekttag einen Stand in ersten Ebene des MZH. Im Gegensatz zu den meisten anderen Projekten diente der Stand nur zur Information über unser Projekt. Die eigentli-

che Präsentation konnte aus Platzgründen nicht am Stand erfolgen, weshalb ein weiterer Ort für den Aufbau des Caves gesucht werden musste (vgl. Kapitel 6.1.1). Für die Gestaltung des Standes wurde deshalb geplant, nur generelle Informationen für Besucher bereitzustellen und das Interesse für den Cave zu wecken.

Monate vor dem Projekttag wurde ein Organisationsteam mit Vertretern aus jedem Informatikprojekt dieses Jahrgangs gebildet. Die Verteilung der Standplätze wurde durch dieses Organisationsteam vorgenommen. MiCasa erhielt einen Stand mit vier Metern Breite vor dem Raum 1400 mit Blick auf die Eingangsfront aus Glas.

Zunächst sollten Besucher sich am Stand über die Kernthematiken der Projektarbeit informieren können. Es wurden im TZI Wandzeitungen im Format DIN A 1 gedruckt und an Stellwänden fixiert, die Texte und Bilder zu den Themen Rahmenkonstruktion, Projektion, Interaktion, Engine und Weltenbau, Modellierung und Charakterdesign bereitstellten.

Damit Interessente sich Informationen über das Projekt mitnehmen konnten, wurden Mappen verteilt, die bereits zur Sponsorsuche verwendet wurden. Zusätzlich wurden Flyer ausgelegt, die generelle Informationen zum Mixed Reality Cave enthielten (vgl. Kapitel 5.1).

Um einen ersten Eindruck vom Cave zu vermit-

teln, wurde ein Rechner aufgestellt, dessen Monitor eine Liveübertragung aus dem Studierhaus zeigen sollte.

Außerdem wurde auf einem weiteren Monitor eine Filmsequenz gezeigt, die die Arbeit von MiCasa während der letzten zwei Jahre widerspiegelt. Musikalisch untermalte Filmausschnitte von Projektwochenenden, Probeaufbauten des Caves und Bildern der Webcam im Projektraum verdeutlichten die Entwicklung des Projektes.

Darüberhinaus war geplant, die Aufmerksamkeit von Besuchern mit etwas Besonderem auf unseren Stand zu lenken. Ein großes Plakat eignete sich gut dafür, wobei das Motiv zur Arbeit von MiCasa passen musste. Eine Anwendung im Cave war die Ebene 1 des MZH, in deren realem Vorbild sich ebenfalls der Stand für den Projekttag befand. Das Motiv sollte Wirklichkeit mit Virtualität verschmelzen lassen und so einen auf den ersten Blick verwirrende Wirkung auf den Betrachter haben. Es wurde ein Plakat gestaltet, das einen nahtlosen Übergang zur Rückwand unseres Standes zeigt. Im Inneren des Plakates wird dieses Mauerwerk aufgebrochen und der Betrachter kann einen Blick in das Innere des dahinterliegenden Raumes 1400 werfen. Durch den Übergang der Mauer neben dem Stand, zunächst in eine fotorealistic Darstellung und dann in die Abbildung der modellierten Welt, ist ein sehr realistischer Eindruck entstanden. Teile der Darstellung

des Raumes 1400 wurden als Drahtgittermodell gedruckt, um die virtuelle Grundlage des Plakatinhaltes zu verdeutlichen.

Damit das Plakat Aufmerksamkeit auf sich ziehen konnte, durfte der Blick von Projekttagbesuchern auf das Plakat jedoch nicht durch Gegenstände wie Tische und Wandzeitungen beeinträchtigt werden. Deshalb wurde der gesamte Stand auf dieses Plakat, das eine Größe von zwei Meter Höhe und vier Meter Breite aufwies, ausgelegt. Die Stellwände mit den Wandzeitungen wurden seitlich des Standes als dessen Begrenzung aufgestellt. Davor wurden ebenfalls seitlich wenige Tische aufgestellt, damit die Monitore dort platziert werden konnten.

Die Präsentation des Caves sollte im Studierhaus stattfinden, einem Gebäude, dessen genaue Lage nicht jedem Besucher des Projekttages bekannt ist. Eine mündliche Wegbeschreibung erschien, wie eine Skizze am Stand im MZH, unpraktisch. Am besten wäre eine Zeichnung, die der Interessent mitnehmen könnte, wobei dem Projekt ein einfacher Zettel zu langweilig war. Es wurden Papierwürfel entwickelt, die aus zwei Hälften bestehen und dadurch geöffnet werden können (vgl. Kapitel 3). In diesem Würfel wurden auf zwei Flächen die Wegbeschreibung zum Studierhaus, sowie ein Foto des Studierhauses gedruckt. Die restlichen Flächen wurden für den Druck

von Sponsorenlogos, Projektfotos und einer technischen Abbildung von einer Cavekonstruktion genutzt. Vier Innenwände einer Würfelhälfte dienten zur Veranschaulichung des Prinzips eines Caves. Es wurden vier Momentaufnahmen der virtuellen Ebene 1 des MZH jeweils im 90 Grad Winkel gemacht, wobei der Standpunkt genau dem Ort des Projektstandes entspricht.



Abbildung II.74: Papierwürfel für Interessenten

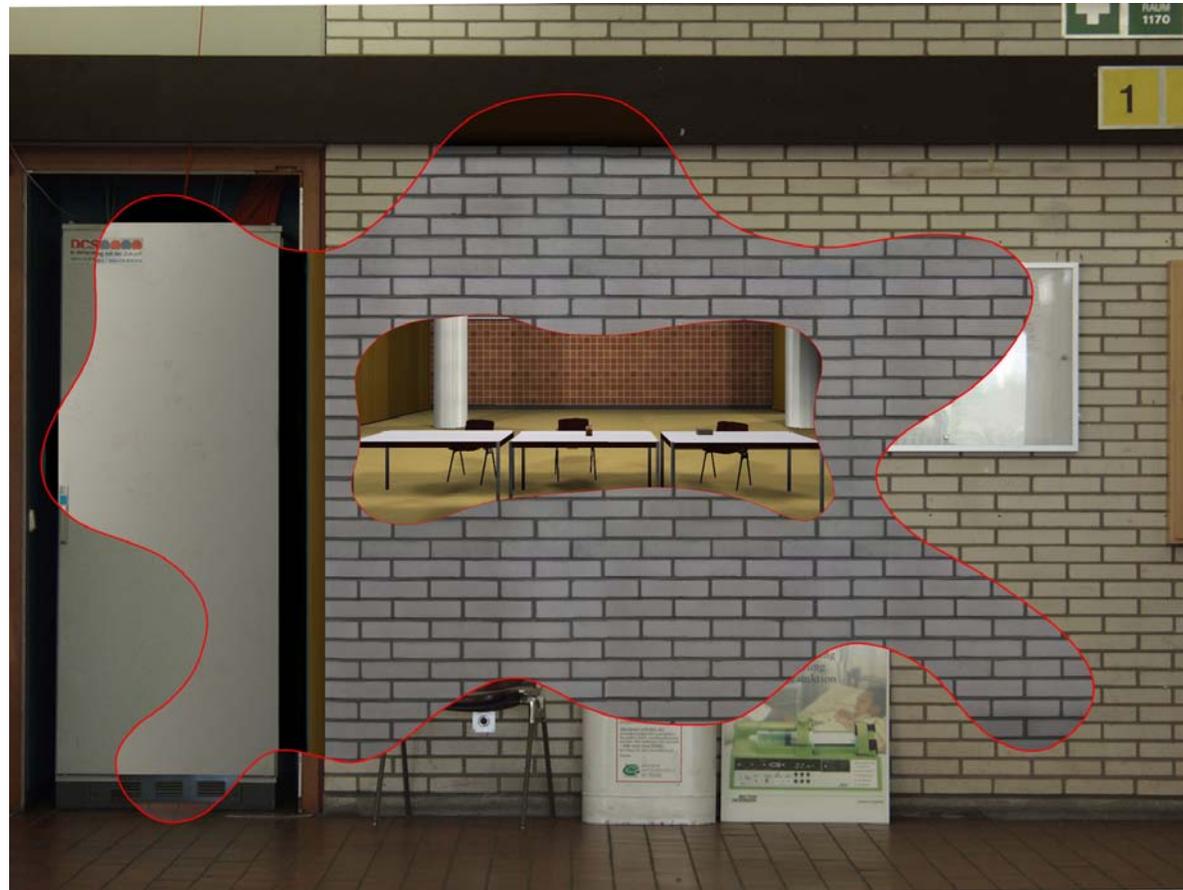


Abbildung II.75: Das Plakat für den MiCasa-Stand.

Diese vier Bilder wurden so in den Würfel gedruckt, dass sich der Betrachter einen Eindruck einer Rundumsicht virtueller Nachbauten realer Schauplätze verschaffen konnte. Dieses Phänomen konnte der Interessent direkt anschließend im Cave erleben. Jeder Würfel erhielt zudem ei-

ne kleine Tüte mit Weingummi zur Wegzehrung bis zum Studierhaus. Die Würfel wurden auf einem weiteren Tisch am Stand pyramidenförmig aufgestellt und gelegentlich sogar über den Luftweg an Passanten verteilt.

### 6.1.3 Die Bühnenpräsentation

2047 - Das Jahr, in dem wir uns wiedersehen

Jedes Projekt hatte am Projekttag 30 Minuten lang Gelegenheit, die grundlegenden Inhalte der vergangenen Projektarbeit auf einer Bühne vorzustellen. Die Entscheidung bei MiCasa fiel schnell auf ein Theaterstück, das natürlich und nicht gespielt wirken sollte. Gleichzeitig musste es bestimmte Informationen über die eigene Projektarbeit enthalten und außerdem darauf hinweisen, dass die eigentliche Präsentation der Arbeit nicht wie üblich an einem Stand im MZH, sondern im Studierhaus stattfinden würde. Darüber hinaus sollte das Stück amüsant sein und nicht langatmig wirken. Es fanden sich im Projekt wenig freiwillige Schauspieler, so dass das Stück außerdem mit wenig Sprechrollen auskommen muss-

te. Die Entscheidung fiel auf die beliebte Zeitreise, das Stück sollte in der Zukunft spielen. Zwei ehemalige Projektmitglieder sollten sich nach vielen Jahren zufällig auf der Straße begegnen und über ihre Erfahrungen im Informatikprojekt zu Studienzeiten sprechen. So konnten lustige Anekdoten über die Universität und Lehrende, wie über Projektmitglieder und -arbeit eingebracht werden. Außerdem konnte die Zeitreise genutzt werden, um mögliche technische Neuerungen vorzustellen, hierbei waren der Fantasie kaum Grenzen gesetzt.

Es entstand das folgende Drehbuch, das nach einigen Verbesserungen erst zwei Tage vor dem Projekttag fertiggestellt wurde. Die Schauspieler probten mit den Autoren erstmals einen Tag vor Aufführung. Die dritte und letzte Probe fand zwei Stunden vor dem Auftritt statt.

Inmitten einer Metropole in einem pulsierenden Fluss von Menschen, die hektisch von A nach B laufen oder anders herum, treffen sich zwei Blicke. Augenblicklich bleiben die beiden Träger dieser Blicke stehen, und als sie endgültig realisieren, wen sie da getroffen haben, verwandeln sich zwei alltagsgeplagte Gesichter in Gesichter, die sich viel zu sagen haben scheinen.

*REGIE 1: Die Bühne wird überrannt in einer Schleife bis Peez und Sefi gleichzeitig (und immer noch mit allen Anderen) die Bühne betreten und sich dann die Blicke treffen (Scheinwerfer zur Verdeutlichung). Nachdem sich Peez und Sefi „erkannt“ haben, gehen sie aufeinander zu und. Ab hier betritt bis auf weiteres keiner die Bühne.*

Wir schreiben das Erdenjahr

**2047**

in irgendeinem Land,  
in irgendeiner Stadt...

Peez: Hey Sebastian, das gibt's doch nicht. Das wir uns hier über den Weg laufen ist ja ein Wunder. Wir haben uns doch seit der Uni nicht mehr gesehen. Wie geht's Dir? Was machst Du so?? Was machst Du vor allem hier???

Sefi: Der Peez! Immer locker bleiben. Ja ...

*REGIE 2: Pause für ein kurzes Erholen*

... gut gehts mir, gut. Naja, Maren und ich haben geheiratet und es wird endlich wieder mehr getanzt. Es hat wieder zum Titel gereicht - *[umreißt mit den Händen das Schild mit der Leuchtschrift, die sagt:]* deutscher Meister im Discofox.

Peez: Hey! Herzlichen Glückwunsch. Nicht schlecht.

Sefi: Danke. Deswegen sind wir auch gerade hier in der Stadt. Showtanzen und so. So ein Meistertitel bringt auch ne Menge Verantwortung mit sich. Jedenfalls haben wir in meinem Erziehungsurlaub eine Tanzschule eröffnet. Ich brauchte einfach mal Abwechslung, ich wollte nicht den ganzen Tag vorm PC sitzen. Meine Älteste wird die Tanzschule wohl mal übernehmen.

Peez: Krass. Eine Älteste? Verstehe. *[grinst]* Wie viele gibt's denn?

Sefi: Ähm ... inzwischen sind es drei. Das reicht dann auch. Ich hab ja auch noch andere Hobbys ... Arbeit zum Beispiel. Was machst Du eigentlich?

Peez: Ich hab mich auch selbstständig gemacht. Vor ein paar Jahren habe ich meinen Job bei Pixar aufgegeben und hier in Deutschland die erste Holothek für Filme aufgemacht. Das ist wie die Holodecks bei Star Trek, Alter. Aber hast Du sicher schon gehört ... in den USA ist das ja der letzte Schrei. Es gibt zwar noch nicht viele Filme, die so produziert werden, aber das ist groß im Kommen.

Sefi: Ja, Filme waren ja schon immer Dein Ding. Das wissen wir spätestens seitdem Du mir bei Jauch als Telefon-Joker mein neues Auto finanziert hast. Ich erinnere mich noch genau an die Frage: Welche Figur spielt Quentin Tarantino in seinem eigenen Film „Reservoir Dogs“?

- a) Mr. White
- b) Mr. Pink
- c) Mr. Brown
- d) Mr. Blonde

Peez: *[lacht]* ... die ist ja nun wirklich nicht so schwer.

Sefi: Ja, wir hatten viel Spaß damals.

Peez: Sag mal, ich war ja neulich in L.A. im Quark's in einer Holo-suite und weißt du, an was mich das erinnert hat? Erinnerst du dich noch an unser Projekt damals im Hauptstudium?

Sefi: Hm... Das war doch das Projekt „MiCasa“, bei Willi Bruns.

### REGIE 3: Lexikon-Einblende: „MiCasa“



Unser *[Werbestimme]* Mixed Reality Cave. Das war doch die Grundlage für die Holo-suite-Forschung ein paar Jahre später. Daraus haben sie einen der achtzehn neuen Bachelor/Master-Studiengänge gebastelt. Den Titel krieg ich noch zusammen, warte ... *[zögerlich und der Rest wie ein Wasserfall:]* „Holographic ... Digital ... Systems ... Media Engineering Economics“. Das wars.

Peez: *[lacht]* ... „Holographic Digital Systems Media Engineering Economics“ ... da hat Jürgen Friedrich doch wieder in seinen Karteikasten mit Mode-Wörtern gegriffen, oder?

Sefi: „Fun“ oder „E-Learning“ hätte er auch noch unterkriegen können.

Peez: Allerdings. Jedenfalls war unser Projekt ja völlig überlaufen, da mussten wir doch drei Teilprojekte draus machen ...

*REGIE 4: zückt ein eingerahmtes Streichholz*

... erinnerst Du Dich?

Sefi: Ist das DAS Streichholz? Damit hat alles angefangen. Das hat uns vor einer Abstellkammer als Projektraum gerettet. War doch ne gute Idee von Willi, Streichhölzer ziehen zu lassen, oder?

*REGIE 5: Lexikon: „Streichholz“*

Peez: *[lacht]* ... Und dann haben wir zwei Jahre lang an diesem Cave rumentwickelt und gebaut ...

*REGIE 6: Lexikon-Einblende: „Cave“, Fotos vom Cave-Aufbau*

Sefi: Genau. Ein Rahmen aus Holz, ein bisschen Projektionsfolie und fertig war das Mixed Reality Erlebnis ...

Peez: Genau. ... Hat Kay eigentlich jemals sein Geld für den Rahmen wiedergekriegt?

Sefi: Ich weiß nicht, ich glaube schon. Naja, ... auf jeden Fall haben wir mit Beamern auf alle vier Seiten unseres Caves einer virtuelle Welt projiziert.

Peez: Beamer!





### Beamer

**Bea|mer;** der -;  
[sprich: „Bima“, engl.]

Teures Ding, mit dem man irgendwas irgendwo draufprojizieren kann.

© 2002 – 2004 MiCasa

REGIE 7: *Lexikon-Einblende: „Beamer“*

Ich erinnere mich dunkel ... So ein uraltes Ding haben die jetzt im Deutschen Museum in München stehen, gleich neben einem IBM Notebook mit TCPA-Chip fürs digitale Rechte Management ... das war auch ne Schnaps-Idee!

Wenn man sich das heute vorstellt ... in den Cave konnte man sich reinstellen und dann in unserer künstlichen Welt herumlaufen. Sowas Ähnliches wie die Holofilme von heute.

Sefi: Richtig! Als Caveman ...



### Caveman

**Cave|man;** der -;  
[sprich: „Käufmänn“, engl.]

Person, die im Cave steht und sich – mit Hilfe von Eingabegeräten – in der virtuellen Welt bewegt und mit ihr interagiert.

© 2002 – 2004 MiCasa

REGIE 8: *Lexikon-Einblende: „Caveman“*

konnte man auf unserer Tanzmatte



### Tanzmatte

**Tanz|mat|te;** die -;

Smarter Fachbegriff für ein Eingabegerät mit 9 Tastenfeldern.

Die T. wird zur Bewegung in der virtuellen Umgebung benötigt, die auf den Cave projiziert wird.



© 2002 – 2004 MiCasa

REGIE 9: *Lexikon-Einblende: „Tanzmatte“*

durch die Gegend laufen und sich durch das Level bewegen. Lustig fand ich auch, wie man in Anjas Testwelt Dinge durch die Gegend bewegen konnte.

Peez: Ach ja, dafür hatten wir den Laser-Pointer und den Metallarm.

REGIE 10: *Lexikon-Einblende „Arm“*

Sefi: Ja der Zauberstab.

**Arm****Arm;** der -;

Floskel für ein – ja, noch eins – Eingabegerät zur Navigation von Objekten.

Siehe auch: „Ergonomie“, Spitzname: *Army, Navy*.

© 2002 – 2004 MiCasa

*REGIE 11: Lexikon-Eintrag „Zauberstab“*

mit dem Laserpointer war wirklich gut. Was hatten wir da eigentlich noch als virtuelle Welt?

Peez: Hm, wir hatten noch Teile der Kunsthalle Bremen nachgebaut und dann war da ja noch das MZH.

*REGIE 12: Lexikon-Eintrag „MZH“***Zauberstab****Zau|ber|stab;** der -;

Fachbegriff für ein Eingabegerät mit Infrarotsender.

Mit dem Z. und Polarisationsfiltern kann man Gegenstände in der virtuellen Realität bewegen.

Siehe auch: „Gute Fee“, „Gandalf“.

© 2002 – 2004 MiCasa

Sefi: Genau, die Kunsthalle ... da hingen Portraits von uns. Naja und das MZH gibt's ja schon lange nicht mehr.

Peez: Ja! Das hab ich auch gehört ... das ist vor 40 Jahren eingestürzt. Oli und Thorsten haben sich damals schon ständig darüber beschwert, wie krumm und schief die Ebene 1 gebaut ist.

*REGIE 13: Lexikon-Eintrag „Ebene 1“***MZH****MZH;** das -;Abk. für: Mehrzweck-Hochhaus.

Gebäude, in dem u.a. Studenten der Informatik und Mathematik einige Zeit ihres Lebens verbringen.

Gegenteil: *gleichmäßig gebaut, genormt*

© 2002 – 2004 MiCasa

Irgendwann hat die Plakat-Polizei angefangen, an der einzigen tragenden Säule die Plakate zu entfernen, das musste ja schief gehen.

Sefi: [lacht] Was ist aus denen beiden eigentlich geworden?

Peez: Hab gehört, die haben jetzt Jobs, wo sie neue Maßstäbe setzen können...

*REGIE 14: im Hintergrund wandelt zwei Gestalten mit Block und Zollstock entlang und vermessen die*

*Leinwände usw.*



Sefi: Achja ... War eigentlich eine spannende Sache, so einen Cave mal wirklich zu bauen. Steht der eigentlich noch immer in unserem Projektraum im GW 2?

Peez: Die Webcam läuft noch ... da hab ich ihn neulich noch stehen sehen. Ich mach übrigens demnächst mal wieder n Film aus Bildern und leg den dann aufn Thunder...

*REGIE 15: Lexikon: „Thunder“*



Sefi: Super! Hast Du in letzter Zeit eigentlich mal wieder den Informatikverteiler von Ralf gelesen? Gibt es den eigentlich überhaupt noch? Kay hat erzählt, Ralf hätte jetzt ein Programm geschrieben, das seine Mails automatisch aus dem Spamfilter der Studenten zurück in den Posteingang verschiebt ...

Peez: Ich habe tatsächlich wieder angefangen Ralfs Mails zu lesen - und nachdem Kay sein viertes Studium angefangen hat, spammt er ja wieder fleißig rum Ü manchmal ist das allerdings gar nicht schlecht - bei dem Gebührenschungel ...

Sefi: Wem sagst Du das... ich bin bloß froh, dass die Uni angefangen hat, das Geld für was Nützliches auszugeben - ohne die neue Klimaanlage hätte man in der Keksdose bei neuerdings 4096 Erstsemestern doch einige Atemprobleme ...

Peez: Das ist aber trotzdem immer noch ein schöner Hörsaal... Ich habe mir da letztens den 4 Stunden Vortrag von Frieder angehört ...

Sefi: Nike? Ich dachte der wäre schon vor 20 Jahren in Rente gegangen?

- Peez: Er wollte nicht so einfach aufhören und hat jetzt noch ein paar Diplomanden angenommen. 2005 hat er sich mit ein paar Piercings und vier Stahlseilen an seinem Schreibtisch festgekettet - bis sein Lehrauftrag dann doch noch verlängert wurde. Ich kann mir die Uni ohne Frieder auch schwer vorstellen ...
- Sefi: Stimmt. Ohne Datenstau im Netz allerdings auch nicht. Ich erinnere mich immer noch sehr gerne an die Zeit als wir zum „Mobilen Campus“ wurden. So mit Vorlesungen überall und allen Materialien online und so. Wir haben doch tatsächlich gedacht, die Sache mit Utes Klappfolien wäre vorbei.
- Peez: Ehrlich gesagt würde ich momentan sonst was für normale Klappfolien geben. Diese weiterentwickelten In-Eye-Displays verursachen bei mir immer Kopfschmerzen.
- Sefi: Apropos Kopfschmerzen... erinnerst Du Dich noch an unser erstes Projektwochenende? Als Daniel, unser Projektbetreuer, und Tim unbedingt auf der zugefrorenen Weser spazieren gehen wollten? Irgendwann abends um zehn haben wir doch die Polizei angerufen und ne Vermisstenmeldung aufgegeben ... die hatten den Rettungshubschrauber schon losgeschickt, als die beiden seelenruhig zurückkamen.
- Peez: Ein Hoch auf die neuen subdermalen GPS-Displays! Damit wären die beiden nicht verloren gegangen. Inzwischen sind die Teile ja auch nicht mehr so teuer ...
- Sefi: Ich hab jetzt auch so eins. Dieses ganz Neue, das Willi Bruns für sein Motorrad entwickelt hat ... das fährt jetzt übrigens nicht mehr nur im Cave, sondern in der Realität und erzeugt quasi ein halb transparentes 3D-Feld um den Fahrer... damit weiß man immer, was wie wo ist. Ist ganz angenehm. Das Coolste aber ist, dass das elektrische Feld so gepolt ist, dass Mücken und Fliegen draußen bleiben.

- Peez: *[gedankenverloren]* Unser alter Cave. Kaum zu glauben, dass wir sowas damals virtuelle Welt genannt haben ... und wie lahm die Rechner damals waren. 1 GB RAM und 2,7 GHz ... deine Kinder würden dich auslachen, oder?
- Sefi: Allerdings. Oh, apropos ... die muss ich jetzt dringend vom Hoverball abholen.
- Peez: Hey! Wenn Du den Chronometer in deinem DeLorean zurückdrehst, bist Du vielleicht noch pünktlich ...
- Sefi: Ich versuchs. Grüß alle schön. Gibt es noch etwas? Sonstiges?
- Peez: Nee, Daniel - oder Dace - hat schon lange aufgehört mit seinem Sonstiges.

*REGIE 16: Lexikon-Eintrag „Dace“*



**Dace**

**Dace**; *das* -;

Betreuer des Projektes MiCasa.

Siehe auch:  
*Sonstiges, Sonstiges Sonstiges.*

© 2002 – 2004 MiCasa

- hat schon lange aufgehört mit seinem Sonstiges.
- Sefi: Warum?
- Peez: Die späteren Projekte haben immer angefangen ihn auszuzählen, wenn er zu lange gebraucht hat. Mittlerweile nennt er das als Tarnung Top 1.
- Sefi: Okay, dann bringe ich hier jetzt mal TOP 1 ein - ich hab gehört, dass der Cave gerade jetzt im Studierhaus aufgebaut ist. Wir sollten da mal vorbeitigern, meinst Du nicht?
- Peez: Joa, lass mal machen. *[Aufforderung an das Publikum:]* Und ihr könnt alle mitkommen!

## 6.2 Der Ablauf des Projekttag

Eine Stunde vor dem offiziellen Beginn des Projekttag traf sich die Projektmitglieder für letzte Vorbereitungen.

### 6.2.1 Im Studierhaus

Außen am Studierhaus wurden Plakate des Formates DIN A 1 aufgehängt, die das Projektlogo und den Projektnamen MiCasa zeigten. Diese wurden am Nachmittag vor dem Projekttag bereits mit wasserresistentem Klarlack eingesprüht, damit die Papierplakate einem Regenschauer standhalten. Darüber hinaus waren auf dem Boulevard von MZH zum Studierhaus, einer Entfernung von ca. 130 Metern, zum einen Kreidefüße aufgezichnet und zum anderen Flatterband angebracht worden. Diese Ideen erwiesen sich neben der Wegbeschreibung in jedem Würfel als praktikabel und erfolgreich. Dadurch wurden auch Personen, die keinen Würfel erhielten auf das Studierhaus aufmerksam.

Im Studierhaus wurde ein zweites Exemplar der Wandzeitung an den Wänden angebracht. Das Licht von dem Beamern wurde von den Projektionsflächen des Caves reflektiert, so dass ausreichend Licht zum Lesen vorhanden war.

Die vorbereiteten Rechner wurden hochgefahren und die Anwendung gestartet. Anschließend wurde ein Besuch der virtuellen Umgebungen durch Projektmitglieder erfolgreich getestet.

Das System samt Anwendungen funktioniert den ganzen Tag relativ problemlos. Genügend anwesende Projektmitglieder erklärten den zahlreichen Besuchern die Funktionalität sowie die Bedienung des Caves und der Eingabemethoden. Ein zuvor erstellter Anwesenheitsplan zeigte sich als gute Idee, darüber hinaus waren allerdings auch immer weit mehr Projektmitglieder im Studierhaus anwesend als laut Plan eingeteilt waren. Der Cave stand von 9:00 Uhr bis 17:00 Uhr einem breiten Publikum für die Besichtigung und das Ausprobieren zur Verfügung.

### 6.2.2 Die virtuelle Ebene 1 des MZH

Der Projekttag 2004 war die zweite Gelegenheit, zu der unser Cave und seine Anwendungen der Öffentlichkeit vorgestellt wurden. Die Ebene 1 des MZH war eine der virtuellen Umgebungen, die die Besucher erkunden konnten. Ihr Eindruck war dabei vielfältig, zumeist jedoch positiv oder sogar begeistert.

Was die MZH-Karte wohl insbesondere so interessant gemacht hat, war der Wiedererkennungsg-



Abbildung II.76: Vorbereitungen beim Studierhaus.



Abbildung II.77: Interessenten im Cave am Projekttag.

wert: Die meisten Besucher kamen ja aus der Ebene 1 des MZH und hatten das Original daher erst kurz zuvor gesehen und waren darin herumgelaufen. Diese Gänge und Räume nun nochmal virtuell begehen zu können, war für viele eine interessante Erfahrung.

Die ungewohnte Steuerung mit der Tanzmatte machte die Bewegung im teilweise recht engen MZH-Level anfangs immer etwas schwierig. So brauchte man schon einige Anläufe, um durch eine Tür hindurch zu gelangen.

Besonders die Besucher, die mit der Ebene 1 vertraut waren, freuten sich über viele Details, wie etwa den Stuhl aus den Senatssälen, den Cola-Automat, den Schwamm, den komischen Computer mit den fehlenden Tasten (das Info-Terminal

im Eingangsbereich) oder die Graffitis an den Säulen. Sogar die Lautsprecher in der Deckenkonstruktion und die beiden fehlenden Steine in der 1380-Frontwand wurden von aufmerksamen Besuchern bemerkt. Und erstaunlich war auch, wieviele der langjährigen Studenten von der Existenz des (realen) Raums 1009 nichts wussten.

Einige Besucher hätten sich etwas mehr Interaktion in der Ebene 1 gewünscht, als nur durch sie hindurch zu laufen. Dabei wurde so etwas wie ein Spiel nicht unbedingt erwartet, sondern vielmehr ein paar kleinere Interaktionen oder Aufgaben. Es wurden verschiedenste Vorschläge gemacht.

- Öffnen und Schließen von Türen
- Rufen eines Fahrstuhls



Abbildung II.78: Der funktionierende Cave am Projekttag.

- Bewegen von Stühlen und Tischen beim Dagegenlaufen, Umwerfen der Kakaotüte oder eines Papierkorbs
- Bedienen des Colaautomaten
- Lichtschalter

Die Erwähnung der angedachten Interaktion mit dem Laserpointer brachten einen Besucher auf eine besonders interessante Idee. Wenn ein Besucher mit dem Laserpointer etwas auf die Tafel schreiben könnte, dann könnte man auch den Schwamm dazu verwenden, die Tafel wieder sauber zu wischen.

Ein Spiel wie PacMan hielten erstaunlich viele der Besucher, mit denen ich gesprochen habe, für übertrieben. Die Steuerung fiel den meisten am Anfang ohnehin schon schwer und zunächst gäbe es in dem MZH-Level so viel zu entdecken, daß man sich auf das Spiel gar nicht konzentrieren könnte.

### 6.2.3 Der A.R.M

Der Projekttag stellte die erste richtige Testphase für den A.R.M. dar. Leider offenbarten sich im laufenden Betrieb nach einiger Zeit ein paar Fehler beim A.R.M (vgl. Kapitel 3.2) inklusive Lasererkennung, sodass das Gerät aus dem Ca-

ve genommen werden musste und kurzfristig an einen separaten Stand innerhalb des Studierhauses verlegt wurde. Dies zeigte sich als gute Lösung, da dadurch ungestört die Funktionalität getestet werden konnte, während im Cave weiterhin Leute agieren konnten.

Es war spannend zu sehen, inwiefern sich das Eingabegerät bewährt und das Interesse der Projekttagsbesucher weckt. Nach anfänglichen Schwierigkeiten am Morgen gelang es dann doch noch, den A.R.M. zusammen mit dem Tonnenspiel außerhalb des Cave zu präsentieren. Das Tonnenspiel stellte die Besucher vor die Aufgabe, durch Benutzung des A.R.M. virtuelle Tonnen in einer kleinen 3D-Welt aufeinander zu stapeln. Die Tonnen reagierten dabei mit entsprechender Physik. Die Aufgabe war nicht unbedingt leicht, denn die Tonnen neigten leicht zum Kippen. Die Besucher ließen sich jedoch nicht davon abbringen, es immer und immer wieder zu versuchen. Vielleicht war es hauptsächlich das etwas futuristische Äußere des A.R.M., welches das Interesse vieler Besucher geweckt hatte, denn es galt an diesem Tag doch viele Fragen zu beantworten und Besucher zu betreuen. Viele der Besucher wollten den A.R.M. ausprobieren, um zu sehen, was sich damit bewerkstelligen lässt.

Nach einer kurzen Einweisung und der Erklärung des Tonnenspiels, schienen die Benutzer sich mit der Benutzung des A.R.M. gut zurechtzufinden.



Abbildung II.79: Präsentation der virtuellen Ebene 1 auf dem Projekttag 2004.



Abbildung II.80: Zwei Interessenten testen den A.R.M.

Das war die Bestätigung, dass die Bedienung des A.R.M. für die Benutzer tatsächlich intuitiv war. Die Benutzung schien allen Benutzern leicht verständlich zu sein und bedurfte fast keinerlei Erklärung. Die aufkommenden Fragen richteten sich

zum größten Teil an die Technik, die hinter dem Eingabegerät steckt.

Auch die kleinsten Besucher des Projekttag ließen es sich nicht nehmen, unseren für manche Personen vielleicht doch etwas überdimensionierten A.R.M. zu benutzen. Die Testpersonen hatten am Projekttag wertvolle Erkenntnisse über unsere Interaktionsmöglichkeit geliefert, die auf Schwächen und Verbesserungsmöglichkeiten hinwiesen. Dies hat auch die Motivation gegeben, sich hierüber weitere Gedanken zu machen. Der A.R.M. ist und war vielleicht noch nicht ganz perfekt, aber zumindest schienen die Besucher Spaß mit ihm zu haben.

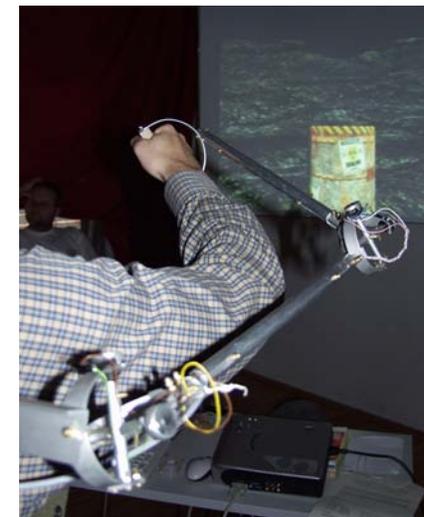


Abbildung II.81: Der A.R.M. in Aktion.

#### 6.2.4 Am Stand im MZH

Die Stellwände, Tische und Wandzeitungen wurden bereits am vergangenen Nachmittag zum Stand gebracht und platziert. Die beiden Rechner für den Film und die Liveübertragung wurden aufgestellt. Das Organisationsteam hatte sich um einen Netzwerk- und Stromanschluss für jedes Projekt gekümmert und verlegte meterweise Kabel. Einige der 250 vorhandenen Würfel wurden aufgestellt und schließlich wurde das Plakat, eines der wichtigsten Bestandteile des Standes, aufgehängt. Damit war der Stand aufgebaut.

Der Projekttag verlief am Stand reibungslos. Die nach Plan stündlich wechselnde Standbelegschaft sorgte für eine gute Information der Besucher, sowie die Umleitung in das Studierhaus, wo die eigentliche Projektpräsentation erfolgte. Einzig die Übertragung der Geschehnisse aus dem Studierhaus war mangels Netzkapazität nicht möglich. Daher waren leider nur Momentaufnahmen aus dem Studierhaus zu sehen. Die in Handarbeit gedruckten und gefalteten Würfel trafen nicht nur bei den kleinen Besuchern des Projektstandes auf Anklang. Ein Projektmitglied wurde einige Wochen nach dem Projekttag noch von einem Bekannten angesprochen, der nur auf Umwegen in den Besitz eines Würfels gekommen war. Entgegen den Erwartungen hatten einige Würfel demnach eine recht hohe Lebensdauer.

#### 6.2.5 Bei der Bühnenpräsentation

Auch die Bühnenpräsentation konnte das Publikum begeistern. Die Entscheidung für eine Souffleuse stellte sich als sehr sinnvoll heraus, da die beiden Hauptdarsteller unter anderem durch Aufregung und sehr kurzer Lernphase (ca 24 Stunden) den Text nicht komplett ohne Hilfe vorbringen konnten. Die während des Stücks gezeigten Lexikoneinträge zur Erklärung von projektinternen Ausdrücken sorgten für regelmäßiges Gelächter im Publikum. Beide Auftritte der ca. zehn Statisten am Anfang und Ende des Stücks liefen ebenfalls überraschend unproblematisch ab, obwohl sie erst direkt vor dem Auftritt eingewiesen wurden und das Drehbuch nicht kannten.

#### 6.2.6 Der letzte Ausklang

Abends erfolgte der Abbau von Stand und Cave sehr geordnet innerhalb von einer Stunde. Anschließend lud Willi alle drei Teilprojekte in einen nahegelegenen Biergarten zu erfrischenden Getränken ein, wobei der Dauerregen die gute Stimmung unter den nahezu wasserundurchlässigen Sonnenschirmen nicht trübte. So fand der erfolgreiche und mindestens genauso anstrengende Projekttag seinen Abschluss.



Abbildung II.82: Der Stand von MiCasa im MZH.

# Buch III

---

DAS PROJEKTLEBEN

---

## 1 Projektwochenenden und Exkursionen

### 1.1 Projektwochenende Nienburg

Nach der ersten Eingewöhnungsphase wurde im Projekt der Wunsch nach einer gemeinsamen Unternehmung laut.

Bei den Planungen zu einem ersten gemeinsamen Projektwochenende sollten die Hauptkriterien der Projektmitglieder nach einer preiswerten Unterkunft, die über einen eigenen Aufenthaltsraum verfügt, mit dem Semesterticket zu erreichen ist und außerdem einen ansprechenden Gemütlichkeitsfaktor besitzt, beachtet werden.

Mit knapper Mehrheit fiel die Wahl auf Nienburg an der Weser. Das nett bebilderte Faltblatt des Naturfreundehauses hatte uns überzeugt. Wohl auch, weil zu dieser Jahreszeit kaum eine Jugendherberge in unseren Breitengraden geöffnet hatte. Vielleicht würde bis dahin sogar ein wenig Schnee und unser Ausflug würde zum Winterurlaub fallen.

Des weiteren sollte eines unser größten Abenteuer gestartet werden: wir wollten selber kochen. Hierzu wurde bereits in Bremen fleißig eingekauft und Autos von einigen Projektmitgliedern beladen, denen von der gesamten Gruppe freundlich nahelegt wurde, doch mit dem Auto nach Nienburg zu fahren.



Abbildung III.1: Gemeinsames Kochen.

#### Freitag

Nach einer Zugfahrt der meisten MiCasianer von einer guten Stunde und einem kleinen Fußmarsch erreichten wir das Naturfreundehaus. Zu unse-



Abbildung III.2: Geschäftiges Treiben am Freitag.

rer Überraschung lag Schnee und der anliegende Parkplatz glich mehr einer Schlittschuhbahn als einer Asphaltfläche. Anwohner hatten Schlitten und Schlittschuhe mitgebracht, um sich auf dieser doch recht großen Fläche - die während der letzten Tage mit original Weserwasser getränkt worden war und überfrozen war - zu verlustieren.

Im Naturfreundehaus sollten wir die einzigen Gäste dieses Wochenende sein und richteten uns zunächst häuslich ein. Anschließend wurde diskutiert, gearbeitet, Fußball im Schnee gespielt, Doppelkopf gekloppt und Lego gespielt.

Zwischendurch quälte die Projektmitglieder der Hunger, sodass die für das Abendessen am Freitag zuständige Gruppe sich ans Werk machte. Für die Informatiker relativ überraschend war die Tatsache, dass das Kochen mit transportablen Kochplatten nicht nur weniger komportabel sondern vielmehr auch extrem langwieriger ist, als mit dem heimischen Einbauherd. Schließlich

mundeten die traditionellen Spaghetti mit roter Soße und die salatigen Vitamine dann aber doch und die für das Abendessen unterbrochenen Tätigkeiten konnten wieder bis tief in die Nacht aufgenommen werden.

### Samstag

Nach einer nahezu durchzechten Nacht, traf sich die müde Gruppe am Samstagmorgen zum stärkenden Frühstück. Danach fiel sie bis zum Mittag in einen produktiven Arbeitswahn. Die Konstruktionsgruppe schmiedete erste Pläne für den Bau des Caves, die MZH-Gruppe realisierte voller Enthusiasmus erste virtuelle Räume der ersten Ebene. Ebenfalls mangelte es nicht an lustigen und mehr oder minder praktikablen Anwendungsideen für weitere virtuelle Umgebungen. Weiterhin kamen wir der Entscheidung, welche Grafik-Engine für unsere Zwecke und frisch gewonnenen Ideen am sinnvollsten erscheint, ein großes Stück näher.



Abbildung III.3: Gemeinsames Essen.

Später am Tag entschied sich die gesamte Gruppe von Informatikern doch tatsächlich, an das Tageslicht und die frische Luft zu gehen. Wir verbrachten den Nachmittag mit vielen Nienburgern auf dem zugefrorenen Parkplatz neben unserer Unterkunft. Nebenwirkungen gab es abgesehen von einigen Stürzen, eingefrorenen Schuhen und Eisdeckendurchbrüchen nicht und so wärmte sich der geschaffte Trupp im Aufenthaltsraum mit heißem Tee, Kaffee und Kakao wieder auf. Anschließend entschieden sich zwei Projektmitglieder, deren Namen hier nicht genannt werden sollen, zu einem ganz besonderen Spaziergang mit unerwarteten Folgen.

Vom beinahe verzehrten Abendessen bekamen die beiden Zurückgekehrten unter den sorgen- und vorwurfsvollen Blicken der Anderen natürlich etwas ab. Der Samstagabend schloss sich für die Einen mit gemeinsamen Filmsehen und Kartenspielen. Für die Anderen erreichte das Projektwochenende mit einem Ausflug in die einzige Discothek Nienburgs einen unerreichbaren Höhepunkt.

### Sonntag

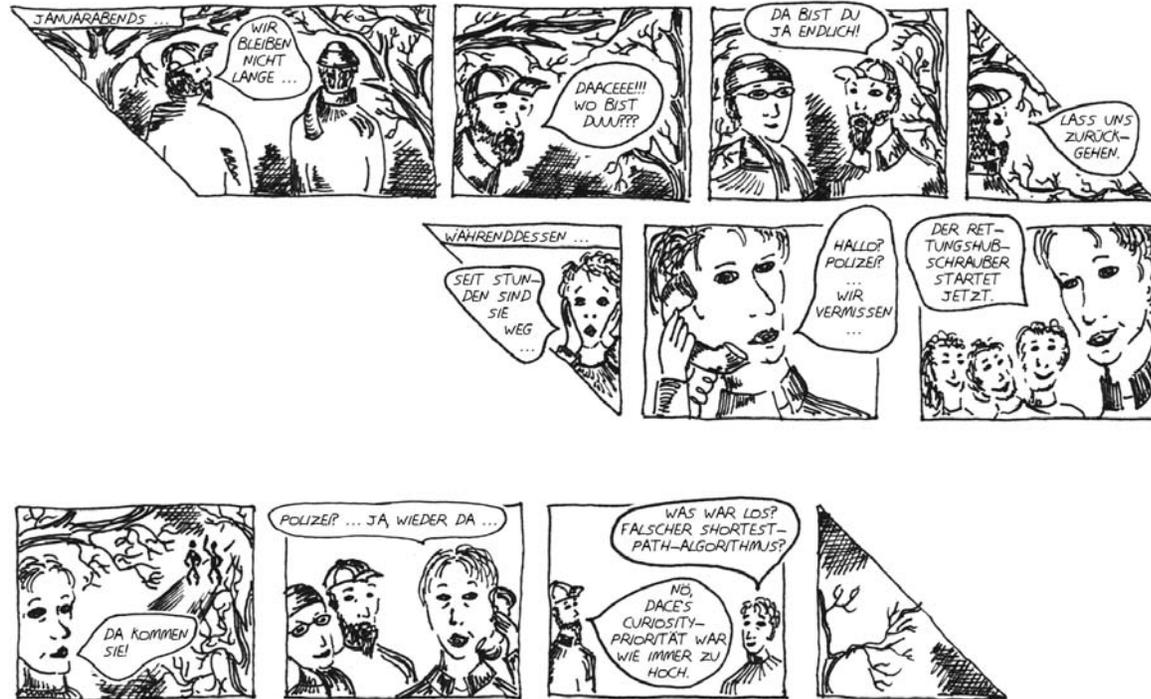
Am nächsten Morgen reisten die MiCasianer nach ihrem ersten gemeinsamen Projektwochenende müde wieder ab und ließen das Naturfreundehaus Nienburg im Eis zurück.



Abbildung III.4: Projektleben in Nienburg.

# ABENTEUERLUST

by Anja Hasagen



## 1.2 Projektwochenende Norderney

Das Ziel für das zweite Projektwochenende stand schnell fest. Im Herzen von Norderney, als Gast einer Jugendherberge, wollte MiCasa sich in die Produktivität und Partylaune stürzen.

Informatiker haben, nicht nur auf Norderney, oft unter den Vorurteilen von Nichtinformatikern in freier Wildbahn zu leiden. Es gibt vorherrschende Meinungen über die Gattung der Informatiker, die einzelne, besonders sensible Exemplare geradezu verletzen und aus dem emotionalen Gleichgewicht bringen. Dieser Artikel soll in erster Linie dazu dienen, derartige Vorurteile abzubauen und diese Gattung Mensch besser zu verstehen, denn auch Informatiker haben Gefühle. Während des gesamten Projektwochenendes auf Norderney wurden die MiCasianer von einem Forscherteam begleitet, das erstaunliche Fotos geschossen hat. Viele Informatiker scheinen sozialen Anschluss und zum Teil sogar die verbale Kommunikation zu anderen Menschen zu suchen.

Als Vorbereitung auf die Arbeit des Forscherteams ist es auf die Straße gegangen und hat Passanten nach ihren Meinungen über Informatiker gefragt. Die häufigsten Antworten werden in diesem Beitrag zitiert. Anschließend wird das Team, welches eine Gruppe von knapp zwanzig Informatikern, die sich selbst MiCasa nennen, und diese ein Wochenende beobachtet hat, sich

dazu äußern und von ihren Erfahrungen sprechen. Zur Verdeutlichung werden Fotos gezeigt, die bisher noch keinem Forscherteam zuvor gelungen sind.

### Vorurteil 1:

„Informatiker sind tragen Hornbrillen und sind Genies in Mathematik.“



Abbildung III.5: Der Informatiker in seiner natürlichen Umgebung.

Ganz offensichtlich trägt dieses Exemplar keine Hornbrille und wer es besser kennenlernen durfte, wird bestätigen, dass seine Talente nicht in der Mathematik schlummern.

**Vorurteil 2:**

„Informatik ist nur etwas für Jungen.“



Abbildung III.6: Es gibt Informatikerinnen.

Unter den abgebildeten InformatikerInnen sind zwei weibliche Exemplare auszumachen. Das entspricht einer Frauenquote von zwanzig Prozent. Darüberhinaus konnte das Forscherteam die Entdeckung machen, dass die beiden Informatikerinnen zumindest äußerlich angepasst an andere weibliche Menschen aussehen. Die Mehrheit der Frauen, wie diese beiden auch, ziehen es vor lange Haare zu tragen. Außerdem konnte durch neueste Technik und durch den mutigen Einsatz eines Forschers festgestellt werden, dass diese beiden Exemplare sogar Schmuck wie Ohrringe, Ringe und Halsketten tragen. Mit noch nie dagewesenen Fotos kann das Forscherteam belegen, dass es tatsächlich Informatikerinnen gibt, und diese unter anderem typisch weibliche Verhaltensweisen und Interessen zeigen.

**Vorurteil 3:**

„Informatiker hocken den ganzen Tag vor dem Rechner und hacken und spielen.“



Abbildung III.7: Informatiker vorm Rechner.

Tja, diese drei männlichen Exemplare von Informatikern sitzen tatsächlich (wenn auch gemeinsam) vor einem Rechner. Das Beobachtungsteam konnte diese Informatiker observieren, wie sie viele Stunden ununterbrochen am Rechner programmiert und auch gespielt haben. Natürliche menschliche Bedürfnisse wie Nahrung und Verdauung schienen nebensächlich zu sein.

Als einziges Lebensmittel diente Kaffee, wie die ausgetrunkene Kaffeetasse im Vordergrund verrät. Das Forscherteam sah sich gezwungen, die Beobachtung dieser drei Exemplare aus Gründen wie Erschöpfung und starkem Schlafdefizit abbrechen.

**Vorurteil 4:**

„Inzwischen gibt es natürlich auch angepasste Informatiker, die aber dann genau so langweilig sind wie Maschinenbauer und E-Techniker.“



Abbildung III.8: Informatiker suchen die Gefahr.

Dieses Bild entstand als sich ein Fotograf außergewöhnlich nah an zwei männliche, scheinbar angepasste, Informatiker herangeschlichen hat. Man findet sie beim Genussmittelkonsum an einer Stelle, die durch ein entsprechendes Verbot gekennzeichnet ist. Menschen, auch Informatiker, die derartige Verbote überschreiten, sind prinzipiell nicht langweilig. Sie haben vielmehr den Ruf draufgängerisch, verr(a)ucht und geheimnisvoll interessant zu sein. Dies entspricht dem Image eines Helden, der sich ständig am Rande der Illegalität aufhält und viel Erfolg in der (schnellen) Liebe hat. Eine solche eine Person kann nicht

langweilig sein.

Für die Gattung der Maschinenbauer und E-Techniker kann in diesem Rahmen nicht gesprochen werden, es müssten sich entsprechende Entdeckerteams finden, die sich näher mit dieser Gattung beschäftigt.

Außerdem konnte das Forscherteam noch weitere überraschenden Verhaltensweisen von InformatikerInnen dokumentieren. Beispielsweise gibt es unter den Informatikern durchaus einen beachtlichen Prozentsatz an Frischluftliebhabern. Es wurden an diesem Wochenende mehrere kilometerlange Wanderungen durch die friesischen Dünen und entlang der Küste unternommen und sogar Landschaftsfotos zu Erinnerungszwecken geschossen.

Weiterhin wurde beobachtet, dass Informatiker durchaus ein ausgeprägtes Sozialleben haben können und auch ohne Einsatz eines Computer miteinander kommunizieren. Die verwendete Sprache erschien dem Forscherteam durch die Verwendung von Fachvokabular und einigen Witzen, die offenbar nur von Exemplaren der gleichen Gattung verstanden werden können, teilweise etwas fremd. Überraschenderweise war es den Forschern jedoch möglich einer Konversation zwischen Informatikern sinngemäß zu folgen. Daraus schließt das Team, dass eine Vermischung von Informatikern und anderen Gattungen durchaus denkbar ist oder sogar bereits vorkommt.



Abbildung III.9: Die Landschaft Norderneys.

Einige Informatiker ließen sich neben den erwähnten Wanderungen zu sportlichen Aktivitäten hinreißen, die sich im traditionellen Frisbee-, Basket- und auch Fußballspiel widerspiegelten. Die Abendgestaltung erschien dem Forscherteam ebenfalls geradezu durchschnittlich. Neben einigen Exemplaren, die weiterhin gearbeitet haben, fanden sich auch Kleingruppen zum Anschauen eines Filmes, Doppelkopfspiel und sogar zum Dis-

kogang. Gelegentlicher Alkoholkonsum schien bei dieser Gattung ebenfalls üblich zu sein, wobei die Wirkung sich mit der anderer Gattungen in jeder Hinsicht zu decken scheint.

Das Forscherteam ist überwältigt von den Entdeckungen dieses Wochenendes. Es plant bereits seine nächsten Beobachtungen der Gattung InformatikerInnen.



Abbildung III.10: Ausgeprägtes Sozialleben von InformatikerInnen.

### 1.3 Projektwochenende Dorum

Die Anfahrt zum nahe Cuxhaven gelegenen Dorum gestaltete sich dank guter Zugverbindung und Taxidienst der Firma Cuxland-Ferienparks sehr komfortabel. Wir hatten drei Häuser angemietet, die einigermaßen dicht zusammenstanden.

Zunächst wurden natürlich die Rechner aufgestellt, da sich herausstellte, dass sich die Häuser nicht mit einer Funkstrecke verbinden ließen (wir hatten extra einen WLAN Router für diese Aufgabe mitgenommen). Also wurde ein Haus oder besser gesagt das Wohnzimmer, auserkoren, um einer grossen Menge informatischen Equipments eine Herberge zu bieten. Ein Netzwerk wurde aufgebaut, das natürlich gleich auf korrekte Funktion getestet werden musste. Deshalb wurden grosse Datenmengen von einem Rechner auf den anderen verschoben, um die Stabilität der Übertragungsrates zu überprüfen. Um die Zugriffsqualität festzustellen, wurde anschließend mehrere Stunden ein extremer Stresstest mit Programmen wie UT2004, COD sowie Homeworld2 gefahren. Aber es wurde natürlich auch ein Plan geschmiedet, welche Arbeiten nun in den nächsten Tagen bewältigt werden sollten und von einer Minderheit wurden sogar „Real-life-events“ geplant.



Abbildung III.11: Watt?

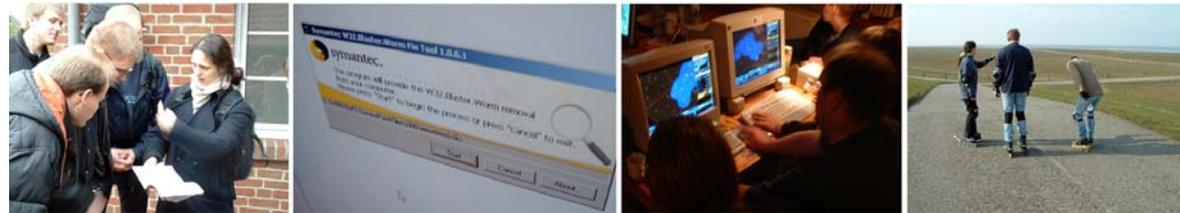


Abbildung III.12: Ankunft und erste Beschäftigungen in Dorum.

Der größte Teil der nachfolgenden Tage und Nächte wurde dazu genutzt, gemeinsam Texturen und Meshes für die Level zu erstellen, sowie den Levelbau voranzutreiben. Die späten Nachtstunden bzw. die frühen Morgenstunden wurden meist mit der Weiterentwicklung der Eingabegeräteanbindung verbracht.

So ist in Dorum der Zauberstab (vgl. Kapitel 3.3.2) in die Engine implementiert worden, der fast unverändert weiterhin im Projekt benutzt wurde. Auch die Bots machten in dieser Zeit ihre ersten Schritte durch ein Testlevel. Hier wurden die Möglichkeiten zur Beeinflussung ihres Verhaltens in der MiCasa-Welt ergründet.

Es gab sogar kurze Exkursionen in die Außenwelt, die oben genannten „Real-life-events“. So setzten wir uns beispielsweise bei Deichwanderungen der stark mit Sauerstoff durchsetzten frischen Seeluft aus und einige Projektmitglieder steigerten ihren Lungendurchsatz sogar noch bei der morgendli-

chen, sportlichen Betätigung mit Inlineskates.

Ein sehr schönes und besonderes Happening war auch das „gemeinsame“ Kochen, das gleichzeitig in mehreren Häusern stattfand, da die Kochgelegenheiten eines Hauses nicht für eine derart grosse Personenanzahl ausgelegt war - unsere Definition von „remote cooking“. Es wurden also einige Menü-Module auf andere Häuser outgesourced (wie z. B. das Kochen der Kartoffeln).

Die Fahrt nach Dorum hat uns auf unserem Weg zum Projekt einen weiteren grossen Schritt vorgebracht, wie es auch an den anderen Projektwochenenden war. Auch und vielleicht sogar vor allem durch den persönlichen Kontakt. Wir waren und sind eine grosse Gruppe mit sehr viel Know-How, die sich gruppenübergreifend gegenseitig in den Teilaufgaben weiterhelfen konnte, somit noch stärker zusammenwuchs und auch die schrägsten und hirnrissigsten Ideen umsetzen konnte und kann.



Abbildung III.13: Kochen, Fußball und Zocken.



Abbildung III.14: Geschäftiges und sportliches Treiben.

## 1.4 MiCaSa in Berlin (30. Juli - 1. Aug 2004)

Von den neun MiCaSianern, die auszogen, das unbekannte Land Berlandien zu erkunden und dort Willi zu besuchen.

### Freitag

Wir unerschrockenen MiCasianer Anja, Dace, Daniel P., Kay, Oliver, Peez, Sabine, Thorsten F. und Till segeln heute in das sagenhafte Land Berlandien. Dazu haben Till und Kay große Schiffe besorgt, eine silberne Golfiatic und eine grüne Golf Kombiatic 2.0. Bei Sonnenaufgang und in aller Frühe treffen wir uns um neun Uhr vor dem SFG, wo sich der artec-Heimathafen befindet. Anja und Peez werden wir an einem einsamen Strand der Insel Braunschweigien unterwegs an Bord nehmen. Alle sind etwas aufgeregt, denn nur Wochen vor uns sind bereits MiCarpet und MiCaDo auf derselben Route aufgebrochen und haben uns nur einige rätselhafte Nachrichten hinterlassen.

Um 9.05 Uhr tauchen die ersten Expeditionsteilnehmer auf und gehen an Bord. Bei ruhiger See heißt es um 09:10 Uhr Leinen los. Trotz Karte segeln wir mit Navigationssystem auf etwas verschlungenen (See-)Wegen über Braunschweigien nach Osten.

Kurz vor ein Uhr laufen wir nach insgesamt knapp vier Stunden Fahrt in den Hafen von Berlandien ein, machen in einer ruhigen Nebenbucht fest, löschen die Ladung und gehen von Bord. Die Berlandianer Willi und Brigitte, seine Frau, erwarten uns Abgesandte der MiCasianer bereits und nehmen uns sehr freundlich auf ihrer Insel auf.



Abbildung III.15: Ein Picknick auf der Insel.

Zum Glück haben sie für uns in ihrem Tipi jede Menge Platz.

Wir erkunden nach der fremden Küste auch gleich das Landesinnere, wo ein sehr warmes Klima zu herrschen scheint; als Fortbewegungsmittel benutzen wir eine Art verkappter Straßenbahn, deren An- und Abfahrtsstationen an der charakteristischen weißen Schlange auf grünem Grund schon von weitem zu erkennen sind.



Abbildung III.16: Unsere Bleibe für das Wochenende.



Abbildung III.17: Der nahegelegene Bahnhof.

Während wir in MiCasani eine etwas andere Herangehensweise haben, scheint dieses Transportmittel in Berlandien hauptsächlich zum Transport von Fahrrädern zu dienen.

Um die Ureinwohner nicht zu verschrecken und dennoch viel über die Gegend zu erfahren, sehen wir uns unauffällig auf dem Alexanderplatz um, versorgen uns mit einheimischer, landestypischer Verpflegung und halten Ausschau vom Fernsehturm.



Abbildung III.18: Der Fernsehturm.

Später schlagen wir uns weiter nach Westen durch und erkunden den Dom, den Vorplatz der Humboldt-Universität, den Innenhof der Staatsbibliothek und das Gebiet „Unter den Linden“. Peez versucht derweil, als Schupo anzuheuern.



Abbildung III.19: Peez will zur Schupo.

Caves sehen wir in diesem unbekanntem Land erstmal keine, die scheint es dort nicht in freier Wildbahn und unter freiem Himmel zu geben.

Um sechs Uhr abends sind wir wieder bei Willi und grillen, ein selbstkühlendes 20l-Faß Bier und 185 Wespen sind auch gerade dort. Als Gastgeschenke an Willi übergeben wir ein MiCasa-T-Shirt und „Robots and Empire“ von Asimov; bis ein Uhr sitzen wir noch in seinem Garten.

### Samstag

Morgens um acht Uhr ziehen Till, Willi, Peez und Anja an die Wannseeküste; Till und Willi zum Schwimmen und Peez und Anja um die morgendliche Binnenseeluft zu genießen. Zuerst versucht Willi sich in kurzer Hose bekleidet mit seinem Bademantel und Badesachen über dem Arm auf den kostenpflichtigen Abschnitt des Wannseestrandes mit den Worten „Hab Besuch aus Bremen - können wir kurz mal reinschauen?“ an der Kassiererin vorbeizuschmuggeln, die jedoch nur herzlich lacht. Kurz darauf springen die beiden an einem abgelegeneren Teil ins Wasser und genießen die Erfrischung des 21Grad C warmen Wassers.

Auf dem Rückweg zu Willis Haus wird noch schnell ein Stop beim benachbarten Bäcker eingelegt, um Brötchen zu besorgen, mit denen wir uns für den Tag stärken können.

In die Innenstadt fahren wir um halb elf Uhr. Unter den Linden bilden wir zwei Stoßtrupps: Kay, Oliver, Sabine, Thorsten und Till wollen das Land zunächst von einer Touristenkutsche aus erforschen, ehe sie sich per pedes interessanten Stellen widmen. Anja, Dace, Daniel und Peez planen unterdessen, das Gebiet gleich zu Fuß zu durchqueren.



Abbildung III.20: Ein kleiner Temperaturtest.

Der erste Forschertrupp entdeckt eine der besagten Touristenkutschen gleich hinter dem Brandenburger Tor. Hier disponieren wir jedoch um und beschließen, uns zunächst mal dem Regierungsviertel der Berlindianer zu widmen. Ein urzerhand erworbenes Logbuch, das frühere Forscher hinterlassen haben, liefert uns Hintergrund-

informationen über den Reichstag und klärt auch die Fragen nach den modernen Bauten am Spreeufer (dem Paul-Löbe-Haus und dem Marie-Elisabeth-Lüders-Haus). Nach der abschließenden Besichtigung des Bundeshauptlingamts genehmigen wir uns eine Erfrischung am sogenannten Pressestrand an der Spree. Dabei kommt uns der Gedanke, daß eine Fahrt mit einem 250-Personen-Touri-Kanu möglicherweise viel spannender sein könnte als mit der Tourikutsche.

Den Gedanken beibehaltend, machen wir uns also auf den Weg zu einer Anlegestelle ganz in der Nähe. Die anderthalbstündige Flußfahrt führt uns zunächst durch das Regierungsviertel gen Osten am Reichstag vorbei. Wir passieren dann diverse Sehenswürdigkeiten, wie beispielsweise das ARD-Gebäude, den Dom und die Museumsinsel, bevor wir kehrt machen und nochmal den westlich der Anlegestelle gelegenden Teil der Spree erkunden (u. a. das Kanzleramt und den Tiergarten). Kapitän Der-auf-dem-stinkenden-Wasserbüffel-fährt erzählt dabei Interessantes zu den Sehenswürdigkeiten.

Nachdem wir wieder festen Boden unter den Füßen haben, machen wir uns auf den Weg zurück zur S-Bahn und steuern unser letztes Tagesziel an; einen Dschungel, den die Berlindianer den Botanischen Garten nennen. Wir erkunden ihn auf befestigten Pfaden. Dabei entdecken wir auch ei-

ne müllfressende Pflanze, die praktischerweise bereits mit passenden lateinischen Namen beschriftet ist.



Abbildung III.21: Eine Müllfressende Pflanze.

Mitten im Dschungel machen wir schließlich einen sonderbaren Fund: riesige Tempelanlagen aus Glas, in denen die Flora (und teilweise auch Fau-

na) der gesamten Welt gefangen gehalten wird. Und so durchqueren wir die heißen Wüsten von Spanien und Chile, das feuchte Klima von Brasilien, das Feuer Afrikas sowie die Würze Australiens und werden dabei unerwartet von riesigen und gefährlichen Tieren (Wachteln) fast gefressen. Im Tempelshop verkaufen uns Eingeborene getrocknete Früchte und Wurzeln, die wir, erschöpft von den Strapazen des Tages, mit auf den Heimweg nehmen.

Um zahlreiche Erfahrungen und leicht gerötete Haut reicher kommen wir gegen fünf Uhr bei Willi an. Da der andere Forschertrupp noch nicht zurückgekehrt ist und möglicherweise gefangengenommen wurde, beschließen wir kurzerhand noch einen Kurztrip zur „Krummen Lanke“ zu unternehmen, wo sich einige von uns zur Erfrischung auch prompt ins Wasser stürzen.



Abbildung III.22: Der Pariser Platz und das Brandenburger Tor.

Der zweite Zug setzt die Erkundung dort fort, wo sie am Vortag unterbrochen wurde, und wandert über den Pariser Platz, durch das Brandenburger Tor, am Reichstag und Kanzleramt vorbei, an der Spree entlang und am vernagelten Schloß Bellevue.

Die Siegestsäule mit goldenem Funkenmariechen oben drauf bietet einen anstrengendem Aufstieg und imposanten Ausblick.



Abbildung III.23: Die Siegestsäule.

Weiter zieht der Erkundungstrupp durch den Dschungel des Tiergartens (in dem offenbar eine MiCaninchenplage herrscht, mit der wir aber nichts zu tun hatten), über die geheimnisvolle Luiseninsel. Er geht an der Philharmonie vorbei, sieht die Nationalgalerie mit den Bildern des Mu-

seum of Modern Art in New York und entsprechendem Besucherandrang von weitem, geht weiter zum Potsdamer Platz und Sony Center, wo er um halb drei Uhr Rast macht.

Anschließend reisen die Forscher mit der S-Bahn zur Gedächtniskirche, flanieren den Kuhdamm entlang und durch einen Supermarkt, in dem sie nach langer Beratung Weißwein für Brigitte besorgen. Weiter geht es per S-Bahn und durch das Hallesche Tor, die Friedrichstraße hinauf zum ehemaligen Checkpoint Charlie.



Abbildung III.25: Ceckpoint Charlie.

Anja und Peez gehen ins dortige Museum und informieren sich grundlegend über die (ost)deutsche Geschichte der vergangenen 50 Jahre, Daniel und Dace schlendern über den Bethlehemkirchplatz weiter in Richtung des alten Ost-Berlindiens bis zum Museum der Kommunikati-

on, in dem ein paar halbstarke Roboter Ball spielen.

Auf der anschließenden Rückfahrt spielen zweimal Straßenmusikanten in der S-Bahn, was ein ausbaufähiges Geschäftsmodell zu sein scheint. Um kurz nach sieben Uhr kommt demnach auch der zweite Zug in Willis verlassenen Garten an, nur um zu erfahren, daß sich die Abenteurer des Bustrupps darauf beschränkten, in der „Krummen Lanke“ zu baden und sich anschließend an deren Ufer entlang in den MiCasianischen Garten zu schleppen, nachdem sie auf einem MiCahn auf der Spree die Beine hochgelegt hatten.

Nach dem Spaghettinessen um acht Uhr übergibt die versammelte Forschergemeinde Gastgeschenke (die Flasche Wein und die getrockneten Früchte) an Brigitte.

Um halb zehn Uhr abends legen wir wieder ab in Richtung Oranienburger Str., kehren im Hof ein, besichtigen die Kunstkommune im Tacheles, machen Performance-Photos und ziehen weiter in eine Anti-George-W.-Bush Themenkneipe.

Die ersten MiCameraden begeben sich dann vor Mitternachts wieder auf den Rückweg zu Willis Insel, andere unternehmen noch Kundschaftergänge in die nähere Umgebung. Dabei beobachten sie das mitunter wunderliche Verhalten von einheimischen Berlindianern und Berlindianerinnen (Besoffene, Prostituierte, Obdachlose, Punks,



Abbildung III.24: Der Potsdamer Platz und das Sony Center.



Abbildung III.26: Das Tacheles.

schlipstragende Pinguine, vergnügungssüchtiges Jungvolk) und von Nachtwachen, die ein goldenes Gebäude (Synagoge) beschützen.

### **Sonntag**

Zwischen acht und elf Uhr stehen wir mehr oder minder freiwillig auf, verabschieden uns um halb zwölf Uhr von Willi und Brigitte und setzen Segel in Richtung Potsdam bzw. Sanssouci. Nachdem wir die Schiffe fest vertäut haben bekommen wir von Kapitän Kay ca. eine Stunde Zeit im Park herumzulaufen.

Ein Erkundungsteam mit Anja, Daniel P., Patrick und Sabine suchte Schatten und Stille in der Friedenskirche und taumelte anschließend durch den Lustgarten vorbei am grünen Gitter und der Villa Illaire und fand über die historische Mühle

schließlich wieder den Weg zum Anleger. Das andere Team erforschte die Gärten um das Schloß, das japanisches Haus, das Neues Palais, den Botanischen Garten und die Orangerie.

Um ca. 13.30 Uhr legen wir wieder in Richtung Bremen ab, wo wir um ca. 16.30 Uhr an der Universität Bremen in den artec-Heimathafen einlaufen.

Berlindien war sehr schön (und warm), wir haben viel gesehen und gelernt, z.B. daß eine Ananas viel Säure enthält, Informatiker schnell Sonnenbrand bekommen, die Schallgeschwindigkeit (bei 0 Grad C) ungefähr 330 m/s beträgt, in Patricks Rucksack ab und zu Flut herrscht, der Aufzugführer im Fernsehturm eher genervt, der Aufpasser am Schloß Bellevue eher mitteilungsbedürftig ist und die Siegestsäule keinen Aufzug hat.



Abbildung III.27: Die unerschrockenen MiCasianer.

## 1.5 MiCasa im Heidepark 15. Oktober 2004

Ein Reisebericht von der Erde

Die MiCaSianer vom Planeten MiCa waren vor einiger Zeit mit ihrem Raumschiff in der Garage der Uni Bremen notgelandet und hatten sich seitdem im Projekt MiCasa versteckt. Nun wollen sie nicht nur nach Hause telefonieren, sondern eine geeignete Raumschiffkonstruktion ausfindig machen, die sie von der Erde wieder auf ihren Planeten bringen kann. Dazu fährt der kleine Erkundungstrupp aN~jA, dA'cE, pAt\*rick, sEb%ast#ian und t!m in den Heidepark bei Soltau, in dem interessante Fahrgeräte stehen sollen, um sich nach Konstruktionen umzuschauen, die sich vielleicht zu einem Umbau zum Raumschiff eignen.

Gut gepflegt mit Erdennahrung machen wir uns früh mit dem Zug auf den Weg. Das Reisen auf diesem Planeten geht offenbar furchtbar langsam, in der Zeit hätten wir schon längst in der Milchstraße sein können. Sol stand schon hoch über dem terrainischen Horizont, als wir um elf Uhr westeuroäischer Erdenzeit den Zielbahnhof erreichen. Es sieht dort nicht gerade wie das Zentrum der Galaxis aus. Als wir uns auf den Weg machen, ist die Temperatur mittlerweile auf gefühlte 00010010 Grad MiCa gestiegen, also auf etwa 18 Grad Celsius.

Nachdem erste Kommunikationsschwierigkeiten überwunden waren („Erdenmensch, wo ist das Raumschiff? Wir sind gekommen, um die Menschheit zu unterwerfen. Habt ihr Kryptonit?“), sehen wir uns im Heidepark um und testen einige der Fahrgeräte.



Abbildung III.28: Eingang des Heideparks.

Da die Erde der blaue Panet ist, beginnen wir mit der Schiffschaukel, die zwar genug Platz für einen Rückflug aller MiCasianer bieten würde, aber einen relativ primitiven Reifenantrieb besitzt und demnach für Reisen mit Überlichtgeschwindigkeit ausscheidet. Die kleine Wildwasserbahn ist offenbar nicht ausreichend gegen den Einfluß der Elementen geschützt, pAt\*rick hat aber dennoch eine Probe genommen, die sich allerdings bei der anschließenden Analyse als zu wässrich herausstellte. Beim Mountain Rafting



Abbildung III.29: Erdennahrung und provinzielle Erdenbahnhöfe.

erreichen wir zwar eine angenehme Geschwindigkeit, die Wendigkeit und Spurstabilität waren aber insgesamt nicht ausreichend. Von den Erdenwachen am Eingang werden  $t!m$  und  $da'cE$  nicht zu den sehr interessant aussehenden Wasserscootern gelassen, da die Wachen eine Überlastung des lokalen Raum-Zeit-Gewichtsgefüges befürchten. Die Krake liegt nicht besonders gut in den Kurven und der Grottenexpress, den wir anschließend testen, bleibt konsequent nicht in der Spur. Dadurch sind beide Konstruktionen für lange Flüge durchs Weltall nicht geeignet.

Langsam ist unsere Geduld am Ende, und so ziehen wir weiter zur steilsten Holzachterbahn der Erde, der Colossos. Nach einer umständlichen take-off Prozedur mit Schlangestehen, Sicherheitsgurt und -bügel, nimmt die Bahn auf der Startrampe auch ganz passabel Schwung auf, den sie jedoch in einer Serie von kleinen Hügeln wieder verliert. Außerdem macht sie ein paar merk-

würdige Geräusche, die uns schließlich weiterziehen lassen. Dabei stellen wir fest, dass es nur das Knurren unserer außerirdischen Mägen war, sodass wir uns am Colossos-Grill mit der mitgebrachten Verpflegung versorgen und auch die örtlichen Spezialitäten probieren. Für die muss man ziemlich viele Erden-Euro auf die Theke legen, außerdem gibt es sie nur in den Größen „mini“ und „sehr klein“.

Danach lassen wir es erstmal ruhig angehen und fahren mit dem Flipper, der immerhin etwas angenehmer in den Kurven liegt als die Krake. Der Rainbow bewegt sich nur seitwärts, was uns ziemlich komisch vorkommt. Der Condor gewinnt zwar kurz nach dem Start an Höhe, kann aber die Erderziehungskraft nicht ganz überwinden und kommt nicht aus den Atmosphäre heraus. Um den Topspin, der die Teilnehmer auf den Kopf stellt, machen wir einen Bogen denn das erscheint uns ziemlich unpraktische für unser Ziel.



Abbildung III.30: Örtliche Spezialitäten mit irdisch kleinen Portionen.

Auf dem Dach des benachbarten Gebäudes, in dem einige Automaten stehen, steht zwar ein rotes Flugzeug, das man aber leider nicht besteigen und auf seine Weltraumtauglichkeit testen kann. In der Hängeachterbahn Limit verlieren wir fast unsere Moonboots, sind aber auch nicht begeistert vom Piloten, der sehr willkürlich die Geschwindigkeit variiert. Auf der Go-Kart Bahn gibt es leider nur elektrische Maschinen, die gänzlich ungeeignet für interstellaren Fernverkehr sind. In der Bobbahn lassen wir uns kräftig durchschütteln, während kräftige Gravitationsfelder die Gondeln merklich verformen. In der großen Wildwasserbahn, die es recht gemächlich angehen lässt, nach ungefähr 20 Erdenminuten aber eine schöne Abfahrt macht, wären wir wahrscheinlich Lichtjahre zu unserem Planeten unterwegs. Der drehende Raumbahnsteig der Wildwasserbahn stellt sich allerdings als sehr praktische Konstruktion heraus. Im Scream fallen wir vielleicht 100m im freien Fall, was uns momentan aber nicht weiterhilft.



Abbildung III.31: Scream bzw. „ich weiß, was du letzten Sommer auf der Erde getan hast“

Da uns die schnellen Fahrgeräte bis jetzt noch nicht für die interstellare Personenbeförderung zusagten, probieren wir nun nach einem weiteren Imbiss die langsamen aus. Auf dem See fahren wir eine Runde mit einem Raddampfer, auf vorinstallierte Schienen angewiesen zu sein, scheint uns allerdings denkbar unvorteilhaft. Danach werden wir in der Monorail- und der Bimmelbahn mit Propaganda überschüttet. Das halten wir noch nicht mal bis zum Mond aus. Bei der Oldtimer-Ralley können wir immerhin etwas lenken, was aber am Kurs insgesamt nicht viel ändert. Abschließend fahren wir Karussell mit großen Teetassen, die ein großes Lenkrad in der Mitte haben, gut gegen eindringendes Wasser geschützt sind und nicht übermäßig wackeln. Diese erscheinen uns noch am geeignetsten von allen Geräten, um für den Heimflug nach MiCa umgebaut und ausgerüstet zu werden. May the force be with us.



Abbildung III.32: Colossos, die Analyse, Mountain Rafting und Testpilotin Anja.

## 2 Das Streichholz

Wie zum Beginn jedes Wintersemesters, wurde auch zu Anfang unseres ersten Projektsemesters fieberhaft nach passenden Räumen für die schier überquellenden Studentenmassen gesucht. Dem Projekt MiCasa wurden zwei Räume im Inneren des sagenumwobenen GW 2 bewilligt, in das sich kaum ein Informatiker je hingetraut hat und noch weniger danach wieder wohlauf gesichtet wurden. Zwei Räume für knapp 60 Studenten, die zudem, aufgrund dieser enormen Interessentenzahl, auf drei Teilprojekte aufgeteilt wurden. Ja, ganz recht ... nein, kein Tippfehler ... zwei Räume für drei Teilprojekte. Genau um diese offensichtliche Problematik zu lösen, wurde in den geringfügig größeren der beiden Räume kurzerhand eine Trennwand eingefügt. Dieses Aus-ein-mach-zwei-Prinzip ist in der Theorie auch durchaus plausibel, nur hatten drei Teilprojekte jetzt zwei, ganz und gar praktisch sehr kleine, und einen weiteren, größeren Raum zur Verfügung. Jedes Projekt hoffte auf den größten Raum und spekulierte wild, warum genau dieses Projekt es nun verdiene, in den großen Raum einziehen zu dürfen. Letztendlich fanden sich jedoch keine triftigen Argumente und so sollte der Zufall entscheiden, welches Teilprojekt die kommenden zwei Jahre in welchem Raum verbringen darf. Ja, das mit dem Zufall ist aber so eine Sache ... es gibt zwar eine berechenbare Wahrscheinlichkeit, mit der einen

der gewünschte Fall treffen kann, aber eine Garantie gibt es nicht. Trotz der Behauptungen einiger bekannter Werbeträger, man könne sein Schicksal mit ein wenig System in die gewünschten Bahnen leiten, ist und bleibt es nunmal doch reine Glückssache.



Abbildung III.33: Das Streichholz

An einem Freitag während des Ausklanges sollte die schicksalshafte Entscheidung dann fallen.

Willi Bruns höchstpersönlich verstand sich gut darin, die unparteiische (Glücks)Spielbank zu mimen und streckte drei scheinbar gleiche Streichholzenden in die Runde. Sei es aus Unentschlossenheit, Höflichkeit oder dem verbreiteten Angstvor-dem-Nichtglück-Syndrom zögerten die Teilprojekte MiCarpet und MiCaDo einen drei hölzernen Stifte zu ziehen. Nicht so ein Abgesandter des Teilprojektes MiCado, der mit offensiver Naivität oder auch Trotz vor den möglichen Konsequenzen auf Antrieb das längste Streichholz zog und unter Jubelschreien und Enttäuschungsschluchzen aller Anwesenden triumphierend in die Höhe hielt.

Das Streichholz wurde in einen goldenen Rahmen eingebettet und bekam über der Tür zum großen Projektraum einen Ehrenplatz.

MiCasa dankt dem artec für die Entscheidung, keinen Studenten eines anderen Projektes zu verweisen. Es wurde ein Weg gefunden, jedem Studenten die Möglichkeit zu geben, an seinem Wunschprojekt teilzunehmen. Um dies mit den vorhandenen Mitteln realisieren zu können, musste manchmal schlicht der Zufall entscheiden, welches Projekt ein grösseres Stück vom Kuchen abgekommen. Die MiCasianer haben bei der Raumverteilung einfach Schwein gehabt und freuen sich darüber!

### 3 Faltanleitung für den MiCasa-Würfel

Der Würfel, wie er am Stand auf dem Projekttag verteilt wurde (vgl. Kapitel 6.1.2), besteht aus Papier und kann ohne jegliche Hilfsmittel konstruiert werden. Durch den Einsatz von Papier und den Verzicht auf Hilfsmittel wie beispielsweise Klebstoff konnten die Kosten und der Aufwand gering gehalten werden.

Damit der Leser die Möglichkeit hat, diesen Würfel auch einmal in gefalteter Form zu sehen, ist hier der Weg von einem Blatt Papier zu einem fertigen Würfel dokumentiert. Die Bilder zeigen die Entstehung einer Außenhälfte.

In der Anleitung werden zuerst alle benötigten

Knicke erzeugt, danach wird der Würfel zusammengefoldet. Durch das vorangestellte Erzeugen der Knicke funktioniert das Falten nachher unproblematisch.

Die Vorlage für den Aufdruck befindet sich auf dem beiliegenden Datenträger. Es ist zu beachten, dass der Würfel aus zwei Hälften besteht, d. h. für einen Würfel müssen zwei Hälften gefaltet werden. Wichtig hierbei ist, dass es sich um einen Innen- und einen Außenwürfel handelt, da diese geringfügig unterschiedlicher Größe sind.

Das Projekt MiCasa wünscht viel Spaß beim Basteln!



Das Blatt ist auf einer Seite zur Hälfte nicht bedruckt, diese Seite gehört nach unten.



Nun wird eine kurze Kanten an der Linie zwischen Bildreihe zwei und drei angelegt und das Blatt gefalzt.



Die Kanten der kurzen Seite werden aufeinandergelegt und das Blatt bekommt einen weiteren Knick.



Die noch nicht auf der Mitte angelegte, kurze Seite dort anlegen und ein dritter Knick entsteht.



Jetzt werden die langen Seiten jeweils an der Linie ...



... zwischen der äußeren und der mittleren Bildreihe angelegt.



Das Blatt sollte jetzt fünf Knicke und zwölf Quadrate haben.



Jetzt werden wieder die beiden kurzen Kanten aufeinander gelegt. Die am Knick liegende, linke Ecke wird soweit zur Mitte geführt, bis der erste Schnittpunkt zweier Knicke erreicht ist.



Das Gleiche wird auch mit der rechten Ecke gemacht, dreieckige Ohren entstehen.



Auf der gegenüberliegenden Seite wird nur das obere der beiden aufeinander liegenden Papiere in gleicher Weise zu einem Dreieck gefaltet.



Dies geschieht auch wieder auf beiden Seiten. Danach wird das Blatt wieder auseinandergefaltet.



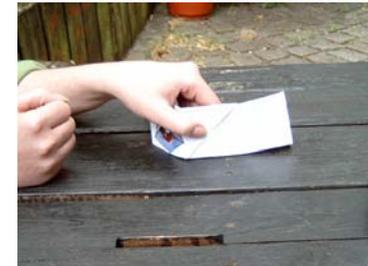
An einer der beiden kurzen Kanten gibt es zwei Quadrate ohne diagonalen Knick. Diese so falten, dass die Kante oben abschließt und angrenzende Knicke übereinander liegen. Den neuen Knick bis zum ersten Knickschnittpunkt falten.



Das Blatt wird so hingelegt, dass sich die zuletzt entstandenen Diagonalen am oberen Ende befinden. Nun werden die Dreiecke im unteren Viertel unter das Blatt gefaltet und hochgehoben.



Danach wird das Blatt an der Mittelkante und die anderen Dreiecke gefaltet und dazwischen gelegt.



Nun noch vorsichtig an der Mitte anfassen und auseinander ziehen ...



... und etwas in Form drücken.



Zu guter letzt noch die beiden verbleibenden Diagonalen falten

...



... und die so entstehenden Laschen ...



... an den Seiten unterschieben.



Fertig ist die Würfelhälfte.

## 4 LAN-Wochenende

### 4.1 Steckbrief/Einladung

**Titel:**

LAN-Wochenende

**Definition:**

Ein Wochenende mit dem Zocken von PC-Spielen verbringen

**Intention:**

Primär: Absolute n00bs in die Welt der Egoshooter einweihen.

Sekundär: Zocken bis die Schwarte kracht!!!

**Zielgruppe:**

Sämtliche Projektmitglieder und interessierte Bekannte / Freunde / Tanten / Cousins / Großeltern / Ehegatten / Mitbewohner

**Räumlichkeiten:**

Projektraum von MiCasa

**Zeitpunkt:**

22./23. März 2003

**Vorhanden:**

Getränkeautomat in unmittelbarer Umgebung, hauseigene Minibar in Form eines Snackautomaten, sanitäre Anlagen, Energie/Strom, Netzwerkzubehör, Überwachungskamera zur Gewährleistung der Sicherheit aller Hardware

und deren rechtmäßigen Besitzern, Breitband-Internetanbindung

**Mitbringen:**

Rechner inkl. Headset und jeder Menge Plattenplatz, PC-Spiele, ggf. stark koffeinhaltige Getränke, je nach Bedarf und Geschmack Nahrungsmittel aller Art

### 4.2 Erfahrungsberichte

*Samstagmittag haben die Ersten erstmal aufgebaut. Hat ein wenig gedauert bis die Rechner standen und alle Spieler eine anständige Verbindung zum Host hatten, aber das ist ja immer so eine Sache. Als erstes wurde Counter-Strike gespielt. Da musste ich erstmal wieder warm werden, ich bin auch eher der Unreal Tournament Spieler und nicht unbedingt für Spiele wie CS zu haben, bei denen man nächtelang irgendwelche Taktiken trainieren muss, um überhaupt mitmachen zu können. Aber dafür lief es ganz gut, waren ein paar hübsche Runden bei, vor allem, wenn man das Dreamteam NTB und Kontrolleur in seiner Mannschaft hatte. Nach ein paar Matches zum Anfang hatten wir alle ziemlich Kohldampf und es wurde was bei Crêpes Suzette geordert und abgeholt. Leider haben die Leute dort ausgerechnet meine Knoblauchsauce vergessen. Toll, Ba-*



Abbildung III.34: Die Räumlichkeiten.

*quette ohne Knobisauce ... naja, satt gemacht hat es ja trotzdem. Dann ging es auch schon weiter, diesmal mit UT 2003. Da habe ich dann ordentlich mit KillaBee gerockt. Hat wirklich Spaß gemacht mit Leuten zu spielen, die man eigentlich jeden Tag sieht, aber noch nie in dieser speziellen Situation des Spiels erlebt hat. Wir haben so ungefähr bis morgens um vier gespielt. Es wurde demnach nicht durchgespielt wie normalerweise. War aber auch mal ganz nett, früher als sonst ins Bett zu gehen. Nach dem Aufstehen am Sonntag haben wir uns dann wieder im Projektraum getroffen und gemeinsam abgebaut. Thunderbird hat ja sogar versucht, in diesem völlig sauerstofffreien Raum zu schlafen, nachdem dort vorher stundenlang gezockt wurde und meh-*



Abbildung III.35: „als exitussi weggestrafed is, slide ich so um die 3cke, und dann derbe weggebashed mit nem dicken luck heady“

*rere PCs und Menschen ihre Transpiration verbreitet haben. Als wir mittags unsere Sachen abgeholt haben, war er aber schon samt seinen Klammotten verschwunden.*

DarklordWarlock

*Ja, das LAN-Wochenende... war ja eigentlich dafür geplant worden, dass MiCasianer, die vorher noch keine Erfahrung mit Egoshootern haben, das mal ausprobieren sollen. Kurz davor hatten wir uns ja für die Unreal Engine und Unreal Tournament 2003 als Basis für unsere virtuelle Umgebungen im Cave entschieden. Und die Prinzipien sollten halt mit diesem Wochenende an die blutigen Anfänger spielend vermittelt werden. Das Bewegen in einer Map beispielsweise,*

oder dass der Spieler sich gar nicht bewegt, sondern sich die Map um den Spieler dreht und kippt. War schon interessant. Leider war ich der einzige sogenannte n00b, der sich getraut hat, mitzumischen. Dementsprechend verliefen die Matches auch.

Bei Counter-Strike wurde mein Charakter regelmäßig erschossen, bevor ich den Gegner überhaupt gesehen hatte. Und dann dieses Kaufen von Waffen und anderem Zubehör am Anfang ... hat eine Weile gedauert, bis ich das kapiert hatte. Bei mir hat es immer ewig gedauert, bis ich diese zig Tasten gedrückt hatte und endlich loslaufen konnte. Nach ein paar Matches hat NTB dann lachend erzählt, dass alle anderen Spieler sich Kaufskripte geschrieben haben, die die gewünschten Gegenstände alle mit einem Tastendruck kaufen. Unter den wahrscheinlich gut gemeinten *Sie-ist-ein-n00b-und-weisses-doch-nicht-besser-Blicken* sämtlicher anderer Spieler habe ich mich dann spätestens wie ein kompletter Neuling in der eigenen Welt der CS-Zocker gefühlt.

Bei „Bombing Run“, das ist ein Spielmodus bei UT2003, hatte ich dann eine geringfügig höhere Lebenserwartung. Dort hat man beliebig viele Leben und das Spiel ist wesentlich offensiver. Es versteckt sich eigentlich keiner das ganze Spiel über in einer dunklen Ecke und erschießt alle Charaktere und besonders n00bs hinterrücks, die

mehr oder weniger ahnungslos an ihm vorbeilaufen. Alle Spieler begegnen sich normalerweise direkt und ziemlich unstrategisch, wer mehr geübt oder auch mal mehr Glück hat, erschießt den anderen, der dann unverzüglich irgendwo in der Nähe wie durch ein Wunder aufersteht und Gelegenheit zu einer Revanche hat. Als ich dann grundlegende Orientierungsprobleme halbwegs gelöst hatte, war und blieb mein größtes Problem das Zielen auf andere Spieler. Damit man nicht sofort wieder - vor allem von KillaBee - ins Jenseits befördert wird, muss man wie ein Verrückter hin- und herrennen und dabei möglichst auch noch springen. So weit so gut ... das ging nach ein wenig Übung auch schon besser, aber wie soll ich zur selben Zeit auch noch andere Spieler ins Visier nehmen und treffen können? Trotz aller guten Tipps von DarklordWarlock blieb mir das völlig schleierhaft.

Auch wenn ich bei den Statistiken konsequent immer ganz hinten lag, habe ich viel gelernt und Spaß gehabt. Wann steigt die nächste LAN, ich bin dabei?!  
exitussi

Ich hatte echt lange nicht mehr gespielt und dachte eigentlich meine Zockerzeiten seien vorbei, bis dieses LAN-Wochenende vom Projekt geplant wurde. Mit achtzehn Person kann es auch schon richtig zur Sache gehen. Letztendlich haben dann sechs Studis gespielt und vier andere den Spielern



Abbildung III.36: Eindrücke.

*zwischenzeitlich einen Besuch abgestattet. Deshalb konnten wir zum Beispiel UT 2003 nur mit zusätzlichen BOTs spielen und bei CS hat es auch nicht für ein 5on5 (zwei Mannschaften mit je fünf Spielern treten gegeneinander an) gereicht. Aber trotzdem hat es viel Spaß gebracht, bei CS haben sogar einige Taktiken wieder geklappt. Zum Glück hatte ich noch mein altes Skript fürs Einkaufen am Anfang des Spiels, sonst braucht man eigentlich gar nicht erst mitspielen - bei dem enormen Zeitverlust gleich zu Beginn der Runde.*

*Ich sollte wirklich mal wieder häufiger spielen, hatte sich irgendwie ein wenig im Sande verlaufen. Meine Lust am CS-Zocken wurde quasi wieder entfacht. UT ist ja eigentlich nicht so meine Baustelle, ist mir zu direkt, ich brauche das Teamplay und die Strategie von CS. Ganz anders als Thunderbird, der alles durch die Bank*

*spielt. Der konnte hinterher in dem Mief im Projektraum nicht pennen und hat dann noch online weiter gezockt, als das offizielle LAN-Wochenende eigentlich schon beendet war. Kein Wunder, hätte ich auch gemacht bei dem Wahnsinnsping. Aber auch UT 2003 hat gerockt, dabei kann man sich einfach mal ein wenig abreagieren und den Könnern wie DarklordWarlock und Kill-aBee in dunklen Ecken auflauern. Insgesamt war die LAN schon ganz cool! Danach habe ich mich erstmal ausgeschlafen, abends hatte ich schließlich noch nen wichtigen Clanwar.*

*Drei Wochen später habe ich auf einmal ein Paket Knoblauchsauce von Crêpes Suzette unter dem Beifahrersitz von meinem Auto gefunden, stammt vermutlich noch von dem Wochenende ... hoffentlich hat die keinem gefehlt.*

NTB



The image shows a screenshot from a game during a DeathMatch in Antalus. A player's character is lying on the ground, indicating they have been killed. A statistics overlay is displayed on the screen, listing the names of the players, their frags, deaths, and network status. The background shows a dark, rocky environment with some green patches.

SPIELER	FRAGS	TODE	NETZ
DarklordWarlock	20	6	PING 0 FPSTD. 113 10:34
KillaBee	20	9	PING 0 FPSTD. 113 10:34
NTB	13	15	PING 0 FPSTD. 73 10:34
Kontrolleur	11	20	PING 0 FPSTD. 82 10:34
Thunderbird	10	19	PING 0 FPSTD. 56 10:34
exitussi	10	23	PING 0 FPSTD. 56 10:34

**Sie sind tot. FEUER drücken für Neueinstieg!**

Abbildung III.37: Die Statistik.

Anhang

## 1 Ausarbeitungen für den Freitags-Ausklang

### 1.1 Sushi-Meister für die Informatiker

Anfangs waren wir am Überlegen, wie wir unsere Idee von einem Eingabegerät, dessen Vorstellung von seiner Art her doch sehr technikbelastet ist, für das geschätzte Auditorium möglichst ansprechend gestalten konnten. Wir entschlossen uns dann nach endlosen Disputen dazu, es mit etwas sehr Elementarem des menschlichen Seins zu versuchen, der Nahrungsaufnahme. Doch wie sollte man die Benutzung eines Motion Capturing-Systems mit etwas wie Kochen in Verbindung bringen?

Betrachtet man den Prozess des Kochens etwas genauer, dann fällt auf, dass die Bewegung des Kochs bisweilen der Choreographie einer Ballettaufführung in Komplexität und Ästhetik nicht unähnlich ist. Insbesondere trifft dieses auf japanische Sushi-Meister zu, die mit Messer und jahrelanger Erfahrung gerüstet aus den giftigsten Tieren, die der Ozean zu bieten hat, genießbare Gerichte zaubern. Dieses komplexe Zusammenspiel in den Rechner zu übertragen wäre wohl jedem Animator, der nicht eben auch Sushi-Meister ist, unmöglich.

An diesem Punkt könnte er sich allerdings der Technik des Motion Capturing bedienen, um die

Komplexität der Bewegung in all ihren Facetten einzufangen. So wäre es möglich einen virtuellen Sushi-Meister zu entwickeln, der uns die Kunst seines Handwerks, ohne eine doch recht beschwerliche Reise in das Land der aufgehenden Sonne, näher bringen könnte.

Auch wir hatten den Wunsch, oder vielmehr die Aufgabe, unseren sehr zahlreichen Zuhörern unsere Vorstellung eines Benutzerinterfaces nahe zu bringen. Da wir nicht sicher waren, auf wie viel Gegenliebe eine technische Beschreibung toßen würde, beschlossen wir, die Zungen des angetretenen Auditoriums mit diversen Spirituosen zu lockern. Vom Standpunkt der Didaktik eine zugegeben etwas bedenkliche Methodik, die sich jedoch als sehr effektiv herausstellte. Um eine gewisse  $\text{\$}$ Teilnahmebereitschaft zu erzeugen, wurde ein alkoholischer Cocktail nur an diejenigen ausgeschenkt, welche mit präzisen Fragen unseren Vortragenden in Bedrängnis brachten. Einerseits stachelte dies die übrigen Zuhörer an, sich zwecks Alkoholkonsums ebenfalls mit dem Thema auseinanderzusetzen.

Andererseits milderte es die Präzision der Fragen und somit den Druck auf den Vortragenden. Da jedes anständige Besäufnis nicht ohne eine gewisse Grundlage an Gesprächsstoff als auch an Nahrung abgehalten werden sollte, kommt hier wieder die Allegorie des Sushi-Meisters zum Zuge. In un-

serem Fall in Form von Sabine, die mit dem hier beschriebenen Rezept für eine Nahrungsgrundlage sorgte. Damit auch der Gesprächstoff nicht ausblieb, beschrieb der jeweils Vortragende einige Aspekte unseres Eingabegerätes. In Ermangelung eines vorzeigbaren Prototyps beschränkte sich unsere Vorführung auf einige virtuellen Modelle.

Während Ersin und Serpil damit beschäftigt waren, die Versorgung der Beteiligungswilligen mit Alkohol sicherzustellen, übernahmen Tim und Markus die Beantwortung der allmählich zunehmenden Fragen. Es fiel ihnen angesichts der schwindenden Reserven an Cocktails nicht leicht.

Selbst die nicht anders zu erwartende Skepsis des Betreuerstabes gegenüber der frevelhaften Laudatio an ein ausschweifendes Gelage wurde vom warmen Essen erstickt.

Abschließend lässt sich sagen, dass zumindest von unserer Warte aus der Vortrag als ein kleiner Erfolg zu betrachten ist. Auch als die Zeit, die wir zu füllen hatten, schon längst verstrichen war, durften wir uns noch über die Anwesenheit heiter diskutierender Teilnehmer freuen. Und so hoffen wir, dass dieser Eulenspiegelstreich sowohl in Taten, als auch in Worten, noch Platz hat an dieser unserer freien Universität Bremen.

## 1.2 MiCasas Argumente für die Unreal Engine

Dieser Ausklang hatte die Intention, den anderen Teilprojekten unsere Wahl für die Unreal Engine 2 zu erläutern. Dabei sollten die Vorteile, die MiCasa von der Benutzung der Unreal Engine 2 überzeugt hatten, zusammengefasst und vorgestellt werden.

Dies geschah in einer herkömmlichen Präsentationsform, bei der die Redner sich zur Illustration ihrer Argumente digitaler Folien behelfen. Der Ausklang wurde mit unserer Präsentation am 09. Januar 2004 abgehalten.

Inhaltlich wurden zunächst die oft synonym verwendeten Begriffe Unreal Engine 2 und Warfare definiert. Warfare meint ein Computerspiel, das die Unreal Engine 2 als Basis benutzen soll. Es befand sich zu diesem Zeitpunkt noch in der Entwicklung und konnte somit nicht käuflich erworben werden.

Anschließend wurde das Lizenzmodell der Grafiken-Engine vorgestellt und auf deren Leistungsmerkmale eingegangen. Die Lizenz der Engine wird offiziell Unreal Engine 2 Runtime genannt. Der Kern der Engine, der sogenannte Engine Core, wurde in C++ programmiert und ist nicht einsehbar. Änderungen der Spiellogik können über die dafür entwickelte Programmierschnittstelle Unre-

al Script vorgenommen werden und als Modul in die Grafik-Engine geladen werden. Die Unreal Engine 2 verfügt über eine integrierte Physiken-Engine, eigene Lichteffekte und ein Partikelsystem, das für die Berechnung von Effekten wie Schnee, Wasser und Nebel zuständig ist. Der Terraineditor dient zum Nachbauen von Außenlevels, beispielsweise mit Bergen (siehe Abbildung IV.1).

Mit dem Materialeditor können Materialien wie Shader erstellt und bearbeitet werden. Integrierte künstliche Intelligenz steuert Spielfiguren durch die virtuellen Umgebungen und haucht ihnen ein wenig Leben durch bestimmte Verhaltensweisen ein.



Abbildung IV.1: Erstellung von Bergen bzw. Hügeln mit dem Terraineditor.

Danach wurde durch einen Einblick in den Unreal Editor 2 die Erstellung von virtuellen Umgebungen prinzipiell erläutert. Es wurde auf die Features und Interaktionsmöglichkeiten verwiesen. Abschließend wurde ein achtminütiger Film gezeigt, der die Leistungsmerkmale anschaulich darstellt.

Nach dieser Präsentation blieb Zeit für die Beantwortung der Fragen aus dem Auditorium.

Nachtrag:

Bei Redaktionsschluss des Projektberichtes im September 2004 war das Verkaufsdatum vom Warfare immer noch nicht bekannt. Nun wird spekuliert, dass - falls die Arbeit an diesem Computerspiel noch abgeschlossen wird - es auf der sich in der Entwicklung befindlichen Unreal Engine 3 basiert und nicht mehr auf der vorgestellten Unreal Engine 2.

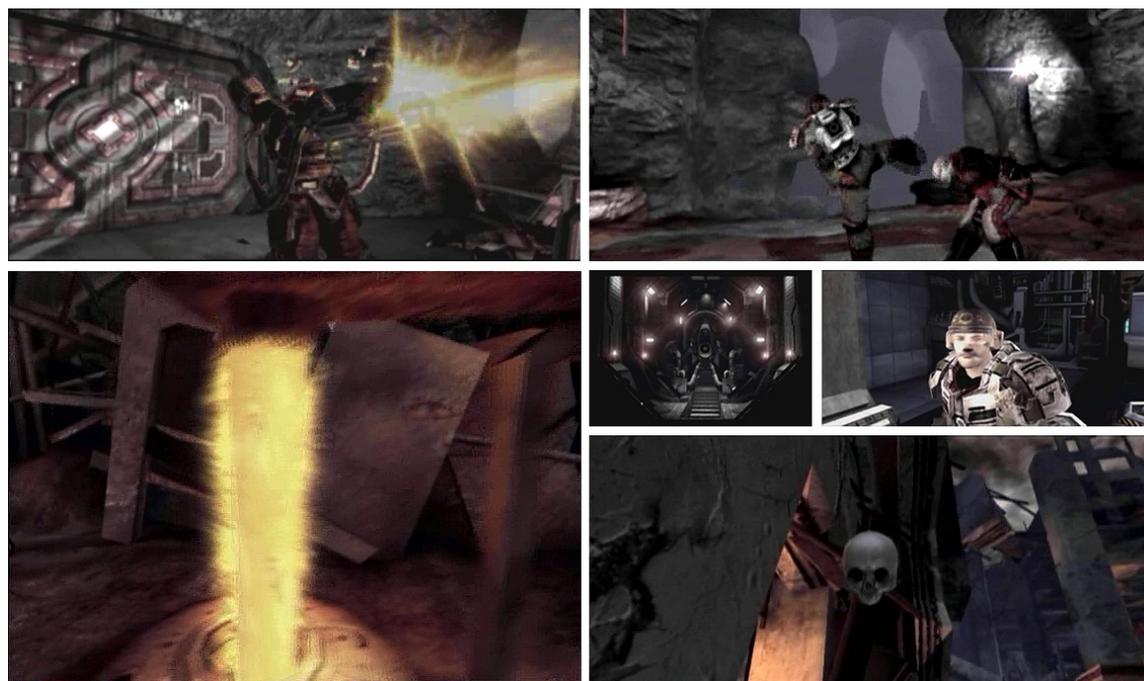


Abbildung IV.2: Eindrücke der Unreal Engine.

### 1.3 Die Tanzmatte

Beschreibung eines alternativen Eingabemediums

Im Rahmen unseres Hauptstudiums Projektes MiCasa war uns die Aufgabe gestellt eine Höhle zu konstruieren, deren Wände durch Computerprojektion den Eindruck einer virtuellen Welt erzeugen sollten. Der Besucher dieser Welt sollte ũ als Verbindung zwischen Realität und Illusion ũ Interaktionsmöglichkeiten mit dem die Projektion steuernden Rechner haben. Wir benötigten unter anderem ein Medium welches die Fußbewegung des Besuchers an den Rechner weitergab.

Angeregt von einer Idee auf der Netzseite von NBC-Giga, auf der eine Alternative zur käuflichen Tanzmatte der Firma Sony dargestellt wurde, haben wir den Plan gefasst, eine Holzkonstruktion zu bauen mit neun beweglichen Kontaktfeldern, mit der ein Besucher Gehbewegungen simulieren konnte.

#### Baubeschreibung

Die Tanzmatte besteht aus einer Rahmenkonstruktion aus Holz. Die Abmessungen betragen  $x$  mal  $y$ . Querstreben unterteilen den Rahmen in neun Teilfelder. Jedes Teilfeld hat am Rand abgesetzte Trägerleisten auf denen Gummipuffer (gesponst von der Firma Viborg, Verden) an-

gebracht waren; es handelt sich hierbei um alte, ausgediente Gummi-aufhängungen von Auspuffanlagen.

Den Abschluß der Konstruktion bilden exakt zugeschnittene Holzquadrate, die mit Metallplatten (als Schutz) abgedeckt wurden. Durch die Gummipufferung ist jede einzelne Teilplatte in einem Schwankungsbereich von zwei Millimetern gegen die Grundkonstruktion beweglich. Soweit die Mechanik der Tanzmatte.

Damit bei einer Belastung einer Teilplatte ein Effekt ausgelöst wird, sind in jedem Teilstück vier Kontaktfedern angebracht, die bei Bewegung der Deckplatte zwei separate Stromkreise schließen. Der erste Kontaktimpuls wird über einen Tastaturkontroller zum PC gesendet, mit dem zweiten Stromkreis werden Leuchtdioden geschaltet.

Durch den Einsatz des Tastaturkontrollers können die 18 Informationen der neun Felder (9 mal geschaltet oder nicht geschaltet) über eine Leitung übermittelt werden.

Die Aufteilung der Matte wird vom PC analog zum Nummernfeld der Tastatur interpretiert, so daß als Eingabealphabet die Zahlen 1 bis 9 zur Verfügung stehen.

Die Dioden leuchten bei nicht geschlossenem Kontaktkreis rot, bei geschlossenem Kontaktkreis



Abbildung IV.3: Die Tanzmatte in Aktion.

grün. Durch die Umschaltung bekommt der Nutzer ein Feedback, ob und welches Feld geschaltet ist.

#### **Erfahrungen mit dem Eingabegerät**

In einer ersten Bauversion haben wir, anstelle der Federn, mit Alufolie umwickelte Styroporplättchen verwendet. Bei Belastungstests ergab sich, daß die Elastizität der Plättchen unzureichend

war. Es wurde der Kontakt bei Belastung der Deckplatte zwar beim ersten Mal zuverlässig getätigt, bei wiederholter Belastung war die Schaltung aber unzuverlässig.

Im übrigen hat sich das Gerät sehr bewährt, die Bedienung ist für den Benutzer sehr intuitiv, die einzelnen Schaltungsvorgänge erfolgen zuverlässig.

## 2 Die Präsentation des Prototypen

### Pleiten, Pech und Pannen - und doch eine gelungene Präsentation

Der erste Meilenstein im Projekt war die Präsentation des Prototyps. Am Freitag, den 22.11.2002 sollten alle drei Mica-Projekte ihren Prototypen präsentieren. Das bedeutete eine Menge Arbeit für jeden Teilnehmer. Neben dem Bau des Caves musste auch die Engine auf den einzelnen Rechnern zum Laufen gebracht werden. Da uns zu diesem Zeitpunkt nur private Ressourcen zur Verfügung standen, war viel Einsatz von jedem erforderlich.

Nach und nach entstanden die Wände und deren Bespannung. Auch die Engine lief auf mehreren Notebooks mehr oder weniger gut. Ein Hub stand uns zur Verfügung und wir waren zuversichtlich, den Cave mit funktionstüchtiger Engine präsentieren zu können. Kurz vor dem festgesetzten Tag kam es dann aber doch zu einigen Problemen. Nach einem Update der Engine aus dem CVS-Verzeichnis gab es plötzlich nur noch zwei Notebooks, auf denen die Engine länger als ein paar Sekunden lief. Es blieb uns als Lösung nur die Möglichkeit, einige Rechner von zu Hause zu organisieren. Alexander, der schon sein Notebook zur Verfügung gestellt hatte, und Daniel P. stellten ihre privaten Rechner zur Verfügung. Die vierte Wand wurde von Sebastians Notebook beleuchtet.

Am Abend vor der Präsentation wurde die Engine noch einmal im Netzwerk auf diesen vier Rechnern erfolgreich getestet und wir fühlten uns nun bereit für die Präsentation. Der Tag begann mit einem Plenum, in dem die letzten Details geklärt wurden und wir noch einmal den Ablauf der Präsentation durchsprachen. Wir entschieden auch, zur Vorlesung nur einige wenige Projektmitglieder zu schicken, damit die anderen schon mit den Vorbereitungen für die Präsentation beginnen konnten.

Das nächste Hindernis erwartete uns dann im Projektraum. Wie geplant sollte nach dem Plenum mit dem Aufbau der Rechner und des Caves im GW2 begonnen werden. Das Hindernis zeigte sich, als wir den benötigten Switch aus dem Projektraum holen wollten, denn dieser war nicht aufzufinden. Nach einer kurzen, jedoch erfolglosen Suche mussten wir betrübt feststellen, dass der Switch über Nacht vermutlich entwendet worden war. Einem glücklichen Zufall ist es zu verdanken, dass genau an diesem Morgen der vom artec ersprochene Switch eingetroffen war und die Präsentation trotzdem durchgeführt werden konnte. Wir beschlossen, den Diebstahl und seine Folgen in einer späteren Diskussion zu erörtern, denn im Augenblick waren wir vollauf mit den

Vorbereitungen der Präsentation beschäftigt.

Damit kamen wir gut voran, so dass nach etwa zwei Stunden der Cave stand, die Rechner angeschlossen waren und das Netzwerk installiert werden konnte. Auch zwei der drei Beamer, die Anja bei verschiedenen Arbeitsgruppen der Uni ausgeliehen hatte, waren schon vor Ort und konnten angeschlossen werden. Der dritte Beamer sollte später abgeholt werden und den 4. bekamen wir vom rtec. An dieser Stelle richten wir einen besonderen Dank an die Arbeitsgruppen Rechnernetze und Rechnerarchitektur sowie an den Technikpool der Angewandten Informatik für ihre Leihgaben. Schon jetzt wurden viele Vor- und Nachteile der Konstruktion unseres Caves deutlich. Die Idee, eine Wand als Tür zu benutzen, erwies sich als äußerst praktikabel. Die Folie war dagegen nicht zu gebrauchen. Auch in Hinsicht auf die Konstruktion konnten viele Ideen bewertet werden, der Prototyp hatte seinen Zweck erfüllt.

Als wir jedoch alle Rechner an das Netzwerk angeschlossen hatten und den ersten Testlauf starten wollten, traten weitere Komplikationen auf. Die einzelnen Clients konnten keine Verbindung mit dem Server aufbauen. Es begann eine fiebrige Fehlersuche, die Konfigurationen der Clients und des Servers wurden überprüft, doch die Einstellungen erwiesen sich alle als korrekt. Den

Ursprung der Verbindungsprobleme fanden wir jedoch erst nach einiger Zeit, denn niemand hatte zu Anfang auch nur im Entferntesten vermutet, dass der nagelneue Switch vom artec, gerade von seiner Originalverpackung befreit, schon seinen Dienst versagte. Auf diese Idee kamen wir erst, als auch die Versuche, einzelne Rechner über ein Ping zu erreichen, fehlschlagen. Wir erhielten schnell Hilfe von Micado, die uns ihr neues Switch zur Verfügung stellten.

Nun funktionierte das System, die Präsentation konnte beginnen. Die Moderation hatte Patrick übernommen und erklärte den Zuschauern nun in einigen Sätzen, warum wir verschiedene Stoffe für die Bespannung verwendet und welche Gedanken wir uns zur Konstruktion gemacht haben. Er erläuterte unsere Ideen zur Sensorik und sprach über unsere Vorstellungen zum Thema Anwendung für den Cave. Danach konnte dann jeder den Cave selbst genau unter die Lupe nehmen. Die zur Verfügung gestellte Madness Enginge lief und die Präsentation war sowohl für unsere Besucher als auch für uns selbst ein voller Erfolg.

Vor allem für uns selbst war diese Präsentation von großer Bedeutung, denn erst durch diese Präsentation wurde vielen von uns klar, auf welches Abenteuer sich jeder einzelne eingelassen hatte. Viele konnten sich nur schwer vorstellen, wie der Cave und die dargestellten Bilder auf einen wir-

ken würden. Erst nach der Präsentation wurde vielen von uns klar, was genau wir in den nächsten zwei Jahren zu konstruieren versuchen würden.

Als Nachtrag sei hier noch erwähnt, dass der Switch tatsächlich entwendet worden war. Nach

einer Anzeige, einer Einstellung des Verfahrens und der Erklärung seitens der Uni und der privaten Versicherung Sebastians, man könne den Schaden nicht bezahlen, wurde Herr Bruns angesprochen. Dieser stand zu seinem Wort und war nach der Absage aller anderen Stellen bereit, den Schaden von seinem Sonderkonto zu bezahlen.

### 3 Die Präsentation zur Halbzeit

(24. Oktober 2003)

Es ist soweit, das Hauptstudiumsprojekt MiCasa hat Halbzeit. Am 24. Oktober 2003 zwischen 12 bis 16 Uhr wurde der Cave im GW2 unter den Treppen des GW 2-Foyers aufgebaut und präsentiert, da dort angemessene Licht- und Raumverhältnisse zu finden waren. Diese Präsentation war ein Einblick in den Stand des Projektes und sollte einen Vorgeschmack auf den kommenden Projekttag im Juli 2004 geben. Alle Interessierten waren herzlich eingeladen, sich den aktuellen Stand des Projektes anzuschauen. Professor Willi Bruns hat extra für die Präsentation sein Forschungssemester unterbrochen, um sich den Cave anzusehen.

Hauptaugenmerk sollte dieses Mal auf dem Cave selbst liegen, d. h. auf dem Rahmen und der Projektionsfolie sowie auf den Eingabegeräten. Für den Cave galten Stabilität, Transportierbarkeit und leichte Handhabbarkeit sowie eine gute Bildqualität als wichtigste Kriterien.

Auch musste getestet werden, inwieweit die neuen Ideen und Veränderungen gegenüber dem Prototypen eine Verbesserung waren. Der Prototyp, welcher in den ersten vier Wochen des Projektes gebaut worden war, diente dem Ausprobieren vieler Möglichkeiten für den Bau des Rahmens und

der Darstellungsqualität verschiedener Projektionsmaterialien (vgl. Buch I, Kapitel 1.4).

Die Unreal Engine 2 (vgl. Buch II, Kapitel 2.1) lief auf vier privaten Rechnern, Beamer wurden von verschiedenen Arbeitsgruppen der Uni zur Verfügung gestellt. Auch das MZH-Level (vgl. Buch II, Kapitel 4.1.1) wurde getestet. Im Cave war die Tanzmatte (vgl. Buch II, Kapitel 3.1) angeschlossen. Über diese Tanzmatte konnte sich der Besucher durch das Level bewegen.

Um möglichst viele Besucher während der Präsentation über das Projekt zu informieren, haben wir eine Wandzeitung entworfen. Hier wurden Themen wie Rahmen, Bespannung, Engine, Eingabegeräte und Level-Bau behandelt.

Außerdem war ein Flyer erstellt worden, der einen kurzen Überblick über das Projekt gab. Um Besucher anzulocken waren Werbeplakate aufgehängt und über den Email-Verteiler von Ralf Streibl eine Einladung an alle Studenten des Fachbereichs Informatik verschickt worden. Diese Anstrengungen haben sich ausgezahlt, es kamen viele Interessierte vorbei, die durch die Email oder die Plakate auf die Präsentation aufmerksam geworden waren. Diese Zuschauer probierten den Cave aus und informierten sich ausführlich

über den technischen Aufbau und die Funktion des Caves.

Zusätzlich wurde der Webcam-Film gezeigt, der das Schaffen im Projektraum in den letzten beiden Semestern in Momentaufnahmen (Ein Foto alle fünf Sekunden, wenn jemand im Raum ist) dokumentierte.

Im Allgemeinen war diese Halbzeitpräsentation eine sehr hilfreiche Erfahrung. Es wurden Fehler gefunden, die es in der verbleibenden Zeit zu beheben galt. Auch für die Vorbereitung des Projekttagess war diese Halbzeitpräsentation eine große Hilfe, da hier schon einmal viele Vorgänge getestet werden konnten. Alles in Allem war das Ganze als eine sehr erfolgreiche Aktion.

# Abbildungen

I.1	Kassenbon Tresor . . . . .	5
I.2	Der Prototyp-Text. . . . .	6
I.3	Bespannung der Rahmen (Seitenansicht). . . . .	8
I.4	Bespannung der Rahmen (Draufsicht). . . . .	8
I.5	Bespannung der Projektionswand mit Folie . . . . .	10
I.6	Skizze des MiCaDo-Caves. . . . .	12
I.7	Das Eingabegerät von MiCaDo. . . . .	14
I.8	Die Cave-Wände von MiCasa. . . . .	15
I.9	Im MiCasa-Cave war reichlich Platz für viele Teilnehmer. . . . .	15
I.10	Die Bespannungstechnik von MiCarpet. . . . .	16
II.1	Aufbau des Cave-Rahmens . . . . .	29
II.2	Die Abmessungen des Cave-Rahmens. . . . .	31
II.3	Die Bestandteile der Dachkonstruktion . . . . .	32
II.4	Die Textur für den Projektor. . . . .	37

---

II.5	Die Platzierung des Projektors im Level. . . . .	38
II.6	Die Einstellungsmöglichkeiten eines Shaders. . . . .	39
II.7	Beispiel für eine SpecularityMask . . . . .	39
II.8	Die Auswirkung der Opacity-Einstellungen. . . . .	40
II.9	Beispiel für eine Detailtextur . . . . .	41
II.10	Beispiel für Eigenleuchten . . . . .	42
II.11	Die Möglichkeiten der Darstellung als Drahtgittermodell im Level. . . . .	42
II.12	Die Einstellungsmöglichkeiten für den Combiner. . . . .	43
II.13	Cubemap für eine Umgebungstextur. . . . .	44
II.14	Die Blickrichtung der Clients. . . . .	46
II.15	Die Erstellung eines eigenen Mutators. . . . .	48
II.16	Eine eigene Bot-Klasse erstellen. . . . .	49
II.17	Elastische PVC-Tanzmatte. . . . .	52
II.18	Vollmetalltanzmatte. . . . .	52
II.19	GIGA-Tanzmattenlayout. . . . .	53
II.20	LEDs an den Tanzmattenfeldern. . . . .	53
II.21	Querschnitt durch eine Box der Tanzmatte. . . . .	54
II.22	Anbindung der Tanzmatte. . . . .	55
II.23	Feedback durch LED-Umschaltung. . . . .	55

---

II.24	Ein Potentiometer. . . . .	59
II.25	Entwurf eines Ellenbogengelenks. . . . .	60
II.26	Kette mit dem virtuellen Modell des A.R.M. . . . .	61
II.27	Der virtuelle A.R.M. am Biped. . . . .	62
II.28	Ein Standard-Gameport. . . . .	63
II.29	Screenshot Joy-O-Mat. . . . .	64
II.30	A.R.M. Schaltbild. . . . .	65
II.31	Schematische Darstellung eines Gelenks. . . . .	66
II.32	Armknochen. . . . .	66
II.33	Verwendete Materialien für den Bau des A.R.M. . . . .	67
II.34	Schematische Darstellung des A.R.M. . . . .	68
II.35	Achsen der Gelenke. . . . .	69
II.36	Das virtuelle MZH und sein reales Vorbild. . . . .	75
II.37	Skizze der Ebene 1 des MZH. . . . .	76
II.38	Der Feuerwehrplan der Ebene 1 des MZH. . . . .	78
II.39	Erhöhter Fußboden in der Ebene 1. . . . .	80
II.40	Modell der Beleuchtungskonstruktion in der Ebene 1. . . . .	81
II.41	Lampen als Beispiel für Karmaobjekte. . . . .	82
II.42	Überblick über die MZH-Map. . . . .	87

---

II.43	Skybox für die Aussenbereiche der Ebene 1. . . . .	88
II.44	Ein modellierter Stuhl mit Kollisionsobjekten. . . . .	92
II.45	Skizzen für die Modellierung. . . . .	93
II.46	Texturierung von Objekten. . . . .	94
II.47	Türgriffe, -motor, und Schalter glänzen. . . . .	95
II.48	Spiegelungen auf der Glasvitrine. . . . .	95
II.49	Spiegelungen auf der DiMeB-Tür. . . . .	96
II.50	Die vertäfelten Holzwände mit Reflektionen. . . . .	97
II.51	Die Fahrstühle im Eingangsbereich. . . . .	99
II.52	Der Flur vor dem kleinen Senatssaal. . . . .	100
II.53	Drahtgittermodell des Erdgeschosses. . . . .	103
II.54	Kachelbarkeit des Parketts. . . . .	105
II.55	Zierelemente am Fußboden. . . . .	106
II.56	Beispiel für TextureBaking. . . . .	107
II.57	Eine Holztür in der Kunsthalle. . . . .	108
II.58	Fußbodenleisten und Handläufe. . . . .	109
II.59	Die Projektoren in der Kunsthalle. . . . .	110
II.60	Reflektionen für Glasoberflächen. . . . .	111
II.61	Die Playground-Welt im Unreal Editor. . . . .	113

---

II.62	Die Zielplattform für die beweglichen Gegenstände. . . . .	114
II.63	Die MiCasa-Logo-Bildserie. . . . .	116
II.64	Der Blick durch die Glasdecke. . . . .	118
II.65	Anordnung der Portaltüren von oben. . . . .	120
II.66	Das Warpzoneproblem. . . . .	120
II.67	Das Portal-Level aus der Vogelperspektive. . . . .	121
II.68	Zwei Portaltüren. . . . .	122
II.69	Warpzoneskizze. . . . .	122
II.70	Der Wegweiser im Portal. . . . .	123
II.71	Letzte Arbeiten in der Nacht vor dem Projekttag . . . . .	126
II.72	Schrittweiser Aufbau des Caves. . . . .	127
II.73	Aufpolsterung der Puffer mit Paketband kurz vor dem Projekttag. . . . .	128
II.74	Papierwürfel für Interessenten . . . . .	130
II.75	Das Plakat für den MiCasa-Stand. . . . .	131
II.76	Vorbereitungen beim Studierhaus. . . . .	143
II.77	Interessenten im Cave am Projekttag. . . . .	144
II.78	Der funktionierende Cave am Projekttag. . . . .	145
II.79	Präsentation der virtuellen Ebene 1 auf dem Projekttag 2004. . . . .	147
II.80	Zwei Interessenten testen den A.R.M. . . . .	148

---

II.81	Der A.R.M. in Aktion. . . . .	148
II.82	Der Stand von MiCasa im MZH. . . . .	150
III.1	Gemeinsames Kochen. . . . .	152
III.2	Geschäftiges Treiben am Freitag. . . . .	153
III.3	Gemeinsames Essen. . . . .	154
III.4	Projektleben in Nienburg. . . . .	155
III.5	Der Informatiker in seiner natürlichen Umgebung. . . . .	157
III.6	Es gibt Informatikerinnen. . . . .	158
III.7	Informatiker vorm Rechner. . . . .	158
III.8	Informatiker suchen die Gefahr. . . . .	159
III.9	Die Landschaft Norderneys. . . . .	160
III.10	Ausgeprägtes Sozialleben von InformatikerInnen. . . . .	161
III.11	Watt? . . . . .	162
III.12	Ankunft und erste Beschäftigungen in Dorum. . . . .	163
III.13	Kochen, Fußball und Zocken. . . . .	164
III.14	Geschäftiges und sportliches Treiben. . . . .	165
III.15	Ein Picknick auf der Insel. . . . .	166
III.16	Unsere Bleibe für das Wochenende. . . . .	167
III.17	Der nahegelegene Bahnhof. . . . .	167

---

III.18	Der Fernsehturm. . . . .	168
III.19	Peez will zur Schupo. . . . .	168
III.20	Ein kleiner Temperaturtest. . . . .	169
III.21	Eine Müllfressende Pflanze. . . . .	170
III.22	Der Pariser Platz und das Brandenburger Tor. . . . .	171
III.23	Die Siegessäule. . . . .	171
III.25	Ceckpoint Charlie. . . . .	172
III.24	Der Potsdamer Platz und das Sony Center. . . . .	173
III.26	Das Tacheles. . . . .	174
III.27	Die unerschrockenen MiCasianer. . . . .	176
III.28	Eingang des Heideparks. . . . .	177
III.29	Erdennahrung und provinzielle Erdenbahnhöfe. . . . .	178
III.30	Örtliche Spezialitäten mit irdisch kleinen Portionen. . . . .	179
III.31	Scream bzw. „ich weiß, was du letzten Sommer auf der Erde getan hast“ . . . . .	180
III.32	Colossos, die Analyse, Mountain Rafting und Testpilotin Anja. . . . .	181
III.33	Das Streichholz . . . . .	182
III.34	Die Räumlichkeiten. . . . .	189
III.35	Zitat eines CS-Spielers. . . . .	189
III.36	Eindrücke. . . . .	191

---

III.37	Die Statistik. . . . .	192
IV.1	Erstellung von Bergen mit dem Terraineditor. . . . .	196
IV.2	Eindrücke der Unreal Engine. . . . .	198
IV.3	Die Tanzmatte in Aktion. . . . .	200

# Glossar

## **3D Sound.**

3D Sound ist die Bezeichnung für eine den Hörer komplett umgebende Geräuschkulisse.

## **Alphakanal.**

Ein Maske, oder Alphakanal, wird verwendet um verschiedene Materialeffekte zu generieren. Ein Alphakanal ist mit in ein Bild eingebettet. Mit Hilfe eines Alphakanals kann man bestimmen, welche Teile eines Bildes hinterher sichtbar sein sollen und welche nicht. Je nachdem, welche Funktion der Unreal Engine 2 mit diesem Alphakanal gefüttert wird, kann ein Alphakanal dazu beitragen, Specular Highlights oder Transparency zu erstellen.

## **Animated Skeletal Meshes.**

Siehe auch Bone Charakter Animation Modell.

## **Antiportals.**

Antiportals werden in der Unreal Engine 2 benutzt, um zu bestimmen welche Bereiche beim Rendern weggeschnitten werden sollen. Hierbei

wird stupide weggeschnitten, ohne darauf zu achten, ob die Polygone hinter dem Antiportal im Blickfeld des Users liegen oder nicht. Der Mapper muss also darauf achten, dass durch eine falsche Positionierung keine Hall of Mirror Effekte entstehen.

## **API.**

Ist die Kurzform für Application Programming Interface. Über die API hat der User Zugriff auf alle Klassen und Funktionen und weitere nützliche Hilfen um sein Programm umzusetzen.

## **AR-Toolkit.**

Bei dem AR-Toolkit handelt es sich um eine Open Source Blickrichtungsverfolgungs-Bibliothek, die eine einfache Entwicklung vielfältiger Augmented Reality (eine Kombination einer realen und einer virtuellen Szene) Anwendungen ermöglicht.

## **Ausklang.**

Projektinterne Veranstaltung jeden Freitag zur regen Diskussion von teilprojektübergreifenden Themen.

**Biped.**

Grundsätzliche Bezeichnung für ein zweibeiniges (Lebe-) Wesen.

**Blocking Volumes.**

Da die Kollisionsberechnung für 3D - Modelle mit extrem vielen Polygonen sehr aufwendig ist, wird die Kollisionsberechnung in der Unreal Engine 2 oft über Blocking Volumes geregelt. Diese sind geometrisch sehr einfach gehalten und umschließen den zu blockenden Bereich, so dass der Spieler dort nicht hineinlaufen kann.

**Bone Character Animation Modell.**

Hierbei handelt es sich um ein Animationsmodell für Charaktere. Um einen Charakter zu animieren, legt man in das 3D Modell des Charakters sozusagen Knochen (Bones). Diese Knochen werden mit verschiedenen Vertices des 3D Modells verknüpft. Ein Oberarmknochen wird logischerweise mit den Oberarm Vertices des Modells verknüpft. Nun kann der Knochen animiert werden und das 3D Modell folgt dem Knochen. Das eigentlich Interessante dabei ist, dass die einzelnen Knochen über Gelenke miteinander verbunden sind und sich so gegenseitig beeinflussen können. Wird also das Handgelenk angehoben, hebt sich gleichzeitig der ganze Arm.

**Bot.**

Ein Bot ist ein computergesteuerter Gegner, der mit Hilfe von künstlicher Intelligenz, Wegpunk-

ten und teilweise gescripteten Ereignissen dem menschlichen Spieler das Leben schwer machen soll.

**BSP - Baum Geometrie.**

Ein BSP - Baum ist eine Datenstruktur, die benutzt wird, um Objekte in einem Raum zu organisieren. Ein BSP - Baum ist eine rekursive Zerlegung des Raumes, die jedes Liniensegment (bzw. jedes Polygon in 3D) wie eine Zerlegungsebene ansieht, die benutzt wird, um alle übrigen Objekte des Raumes als vor oder hinter dieser Ebene liegend zu kategorisieren. Mit anderen Worten, wenn ein neues Objekt in den Baum eingefügt wird, wird es zunächst in Relation zur Wurzel zugeordnet und dann rekursiv unter Betrachtung jedes folgenden Knoten eingeordnet. Diese Zuordnungen werden in einem Baum gespeichert. Mit Hilfe dieser BSP - Baum Geometrie lässt sich nun leicht entscheiden, welche Polygone von anderen verdeckt werden, sodass nicht sichtbare Polygone sofort weggeschnitten werden können.

**Bumpmapping.**

Ein Verfahren, dass durch geschickte Licht-/Schattenberechnung den Eindruck einer rauen Oberfläche erweckt, obwohl die Geometrie absolut plan ist.

**Capture The Flag.**

Dies ist ein beliebter Spielmodus in Online Multi-

player Spielen. Hier treten zwei Teams gegeneinander an und versuchen die gegnerische Flagge zu stehlen und in die eigene Basis zu bringen, um einen Punkt zu bekommen. Gleichzeitig versucht man seine eigene Flagge zu beschützen. Das Team dass nach Ablauf der Zeit die gegnerische Flagge öfter gestohlen hat, gewinnt.

**Client.**

Ein Server / Client Modell wird häufig in der Netzwerkkommunikation gefunden. Ein Server stellt einen Dienst zur Verfügung, den ein Client mit einem Verbindungsaufbau zum Server nutzen kann.

**Color Modifier.**

Eine Funktion des Material Editors der Unreal Engine 2. Mit dem Color Modifier lässt sich die Grundfarbe einer Textur leicht verändern. So kann man mit einer Textur mehrere stimmungs-volle Einfärbungen der Textur erstellen und spart so Texturspeicher.

**Combiner.**

Eine Funktion des Material Editors der Unreal Engine 2. Mit dem Combiner lassen sich zwei verschiedene Materialien, mit Hilfe von verschiedenen Überblendungsfunktionen, zu einem zusammenfassen.

**ConstantColor Modifier.**

Siehe Color Modifier.

**Constructive Solid Geometrie (CSG) Funktionen.**

Konstruktive Solid Geometrie bietet die Möglichkeit, verschiedene Grundformen miteinander zu kombinieren, um komplexere 3D Modelle zu erstellen. Um nun die Grundgeometrie zu kombinieren, benötigt man einige Funktionen. Diese sind zum Beispiel Add und Subtract. Dabei werden zwei Grundgeometrieformen entweder miteinander addiert oder voneinander subtrahiert. Das Ergebnis ist dann eine Kombination aus den beiden eingegebenen Geometrieformen. Bsp: Ein Quader mit einem Loch in der Mitte entsteht durch Subtraktion eines Zylinders aus der Mitte des Quaders.

**Corel Draw.**

Proprietäres Programm zum Erstellen von Vektorgrafiken.

**Cubecam.**

Die Cubecam ist unsere Lösung des verteilten Rechnens. Über diesen Mutator kann eingestellt werden, welcher PC welche Wand berechnen soll. Dies passiert über ein kleines Menü. Die korrekte Kameraposition für die entsprechende Wand wird dann automatisch eingestellt.

**Cubemap.**

Eine Cubemap spiegelt die Umgebung um einen ganz bestimmten Punkt im virtuellen Raum wider. Wie der Name schon sagt, besteht die Cu-

bemap aus 6 einzelnen Texturen. Norden, Süden, Osten, Westen, Top, Bottom. Setzt man die Texturen an ihren Kanten zusammen, entsteht ein Würfel.

**Deathmatch.**

Ein beliebter Spielmodus in Online Multiplayer First Person Shootern, wie Unreal Tournament es einer ist. Hierbei geht es darum, jeden Gegner auszuschalten und selbst, wenn möglich, zu überleben. Der Spieler mit den meisten Kills gewinnt nach Ablauf der Zeit.

**DetailScale.**

Gibt die Skalierung der Detailtextur an. Je größer die Detailtextur auf einer Textur skaliert wird, desto eher wird sie auch eingeblendet, wenn man auf eine Wand mit Detailtextur zuläuft.

**Detailskalierung.**

Siehe DetailScale.

**Detailtextur.**

Dies ist eine graustufen Textur, die einer Textur auf einem Objekt mehr Details verleiht, je näher man der eigentlichen Textur kommt. So können zum Beispiel auf einer Steintextur auch noch feinere Strukturen dargestellt werden je näher man herankommt. Dadurch wirkt die eigentliche Textur auch nicht verwaschen oder pixelig wenn man sie von Nahem betrachtet. Ab einer bestimmten

Entfernung wird die Detailtextur komplett ausgeblendet.

**DiMeB.**

Die Arbeitsgruppe von Professorin Dr. Heidi Schelhowe: Digitale Medien in der Bildung.

**Displacement Mapping.**

Hierbei wird durch eine Effekttextr die Geometrie eines Objektes verändert. Es entsteht so eine rauhe Oberfläche, die, anders als beim Bumpmapping, kein Fake ist, sondern wirklich in der Geometrie existiert.

**Drahtgittermodell.**

Das Drahtgittermodell ist eine einfache, dreidimensionale Darstellung in der Computergrafik. Die Eckpunkte eines Objektes werden durch Linien miteinander verbunden. Das entstehende Modell wird auf eine zweidimensionale Fläche projiziert. Viele Programme in der 3D Computergrafik besitzen Modi, um die Objekte als Drahtgittermodell (engl. Wireframe) darzustellen.

**Eigenleuchten.**

Siehe Selfillumination.

**Engine.**

Das Softwaregrundgerüst mit dessen Hilfe die virtuellen Welten überhaupt erst sichtbar werden. Heutzutage ist eine Grafik Engine sogenannte lizenzierte Middleware für Spieleentwickler. Die-

se Engine umfasst nicht nur die Grafik, sondern auch Sound, KI, Netzwerkfähigkeit etc.

**Environmental Map.**

Mit der Environmental Map wird eine Cubemap dazu verwendet, Reflektionen der Umgebung auf eine spiegelnde Oberfläche zu bringen. In der Unreal Engine 2 wird die Environmental Map und die dazugehörige Cubemap nicht zur Laufzeit generiert, wodurch Echtzeitreflexionen ausgeschlossen sind.

**Exoskelett.**

Ein Exoskelett ist eine Stützstruktur für einen Organismus, die eine stabile äußere Hülle um diesen bildet.

**Fallback Material.**

Einige Shader benötigen bestimmte Hardware- und Software- Voraussetzungen um korrekt dargestellt werden zu können. Ist dies nicht der Fall, wird der komplexe Shader durch einen Einfacheren ersetzt, damit auch auf älteren Computersystemen eine möglichst hohe Darstellungsqualität erhalten bleibt.

**Finalblend.**

Eine Funktion des Material Editors der Unreal Engine 2. Die Finalblend Funktion fasst die verschiedenen Facetten eines Materials zusammen und setzt noch eine globale Eigenschaft, wie zum

Beispiel Translucency, auf das gesamte Material.

**Frame.**

Ein Frame ist ein einzelnes Bild in einer Folge von Bildern. Im Kino wird eine Sekunde Film mit 24 einzelnen Bildern erstellt. Also 24 Frames/Sekunde. Auch bei Computerspielen werden die Bewegungen und Animationen mit einer Folge von einzelnen Frames erstellt. Allerdings reichen den meisten Computerspielern 24 Frames nicht aus. Heutige Grafikkarten berechnen die Computerspiele meist mit mehr als 50 Frames/Sekunde um auch bei schnellen Bewegungen ein Flimmern oder Ruckeln zu verhindern.

**Freiheitsgrad.**

Unter einem Freiheitsgrad eines physikalischen Systems versteht man eine (verallgemeinerte) Koordinate, mit der das System beschrieben werden kann. Die Zahl der Freiheitsgrade ist eine Eigenschaft des Systems.

**Funkpeilung.**

Unter Funkpeilung versteht man das Verfahren, durch physikalische Messung die Richtung, aus der ein Funksignal stammt, zu bestimmen.

**Gameport.**

Der Gameport ist ein Port an PC-artigen Computern, an dem man analoge Joysticks und Gamepads anschließen kann. Am Computer befin-

det sich eine (heute meist gelbe) 15-polige Sub-D-Buchse.

**Gametype.**

Ein Gametype ist ein Spielmodus wie Deathmatch oder Capture The Flag. In jedem Gametyp gelten eigene Spielregeln.

**Garbage Collection.**

Garbage Collection oder (automatische) Speicherbereinigung (auch Freispeichersammlung) ist ein Fachbegriff aus der Softwaretechnik. Garbage Collection steht für ein Verfahren zur regelmäßigen automatischen Wiederverfügbarmachung von nicht mehr benötigtem Speicher und anderen Betriebsmitteln, indem nicht mehr erreichbare Objekte im Speicher automatisch freigegeben werden. Das Wort Garbage Collection kommt aus dem Englischen und heißt wörtlich übersetzt Müllabfuhr.

**Grafik-Engine.**

siehe Engine.

**GW2.**

Beim GW2 handelt es sich um das (sagenumwucherte) Gebäude Geisteswissenschaften 2, in dem sich der Projektraum aufgrund des Platzmangels im MZH befand.

**Hall of Mirror Effekt.**

Der Hall of Mirror Effekt ist ein Rendering - Feh-

ler, der auftritt, wenn sozusagen ein Loch in der Geometrie ist und man durch das Loch in das Nichts blickt. Im Gegensatz zur Realität hört die virtuelle Welt irgendwo einfach auf. Blickt man nun in dieses Nichts wird der Framebuffer nicht mehr korrekt geupdatet und es werden veraltete Daten aus dem Framebuffer einfach erneut geschrieben. Durch Bewegen des Blickwinkels wird dann ein veraltetes Frame so oft geschrieben, dass man den Eindruck einer Endlosspiegelung bekommt - wie in einer Spiegelhalle - bekommt.

**Highmap.**

Eine Highmap ist die Grundlage zur Erstellung eines Terrains. Eine Highmap ist eine Graustufen - Textur. Jeder einzelne Pixel bestimmt über seinen Helligkeitswert die Höhe des Terrains. Je heller, desto höher das Terrain an dieser Stelle.

**Homogene Koordinaten.**

In der Computergrafik werden Transformationen in einem um eine Dimension erweiterten Koordinatensystem durchgeführt. Damit ist es möglich, mehrere Transformationen nacheinander in einer gemeinsamen Matrix zu kombinieren.

**HUD.**

Kurzform für Heads up Display. In Computerspielen zeigt das HUD dem Spieler Informationen über seine gewählte Waffe, Lebenspunkte oder einen Wegweiser zum nächsten Ziel.

**Indoor-Map.**

Eine virtuelle Welt, die nur das Innere eines Gebäudes simuliert, ohne einen Blick auf die Außenwelt zu gewähren.

**Karmaobjekt.**

Werden einem Objekt, in der Unreal Engine 2, eine oder mehrere physikalische Eigenschaften zugewiesen, wird es zum Karmaobjekt. Nun werden diese zugewiesenen physikalischen Eigenschaften von der Karma-Engine simuliert.

**Karma-Engine.**

Die Karma-Engine ist eine Physik-Engine. Sie dient zur Berechnung und Simulation von physikalischen Abläufen in virtuellen Welten. Epic Games hat die Karma-Engine vom Entwickler Mathengine eingekauft und in die Unreal Engine 2 implementiert.

**Keyframe.**

Siehe Keyframe - Animation Modell.

**Keyframe - Animation Modell.**

Das Vertex Animation Modell animiert 3D - Objekte, wie der Name schon sagt, mit Hilfe der Vertizes des 3D - Modells. Hierbei wird das Modell anhand von Keyframes animiert. In jedem Keyframe werden die entsprechenden Positionen der einzelnen Vertizes festgesetzt. Zwischen den einzelnen Keyframes interpoliert der Computer in

einer bestimmten Anzahl von Schritten. Dadurch entsteht eine Animation. Je mehr Zwischenschritte zwischen den Keyframes liegen, desto flüssiger wird die Animation.

**Kollisionsabfrage.**

Die Kollisionsabfrage stellt in 3D Grafik Engines sicher, dass Objekte miteinander kollidieren und daraufhin interagieren können. So wird auch sichergestellt, dass der Spieler nicht plötzlich aus dem Level ins Nichts fällt.

**Koppel.**

Bei der Koppel handelt es sich um einen Gürtel, der für den A.R.M. als Halterung benutzt wurde.

**Laserpointer.**

Ein Gerät, das einen schwachen Laserstrahl generiert.

**Level.**

Ein Level ist in Computerspielen ein Abschnitt des Spiels, den der Spieler bewältigen muss, um in den nächsten Abschnitt zu gelangen. Die Aufteilung von Spielen in Levels ist meist technisch begründet, da nicht eine beliebig große Anzahl von Ressourcen (d.h. Texturen, Umgebungsobjekte oder Effekte) gleichzeitig im Speicher gehalten werden kann. Daher kommt es beim Wechsel zu einem neuen Level üblicherweise zu massiven

Ladevorgängen, bei denen die Ressourcen geladen werden, die für den nächsten Abschnitt benötigt werden und veraltete Ressourcen freigegeben werden. Siehe auch Map.

**Lightmap.**

Eine Lightmap enthält Beleuchtungsinformationen für ein 3D Objekt. Damit auf einem Objekt entsprechend der gesetzten Lichter auch Schattenverläufe zu sehen sind, muss diese Information in der Textur des Objektes gespeichert werden. Dies passiert über die Lightmap, die später mit der eigentlichen Textur multipliziert wird. Sie gibt an, wo die Textur heller und wo dunkler dargestellt wird. Die Erstellung der Lightmap kann nicht zur Laufzeit passieren. Sie muss vorberechnet werden. Darum sind dynamische Lichteffekte mit Lightmaps verständlicherweise nicht möglich.

**Map.**

Eine virtuelle Welt. Die Begriffe kommen aus dem Bereich Computerspiele und beschreiben dort einen Abschnitt (Level) des gesamten Spiels. Üblicherweise kommt eine mehr oder weniger lange Ladezeit vor dem Betreten eines neuen Levels vor.

**Material Editor.**

Mit Hilfe des Material Editors kann man aus einer einfachen Textur ein Material erstellen, das reflektiert, Specularity besitzt und vielleicht noch

dazu animiert ist. Es ist sozusagen das Tool mit dessen Hilfe man sich seine eigenen Shader zusammenschließen kann.

**MiCasa Stick.**

Modifizierte Unrealwaffe zum gezielten Bewegen von Karmaobjekten (siehe Karmaobjekt) innerhalb einer virtuellen Umgebung.

**Mod.**

Siehe Modifikation.

**Modellingssoftware.**

Eine Software zum Erstellen der 3D - Objekte, die in einer virtuellen Welt platziert werden können. Die bekanntesten Vertreter sind:  
3D Studio Max, Maya, SoftImage.

**Modifier.**

Eine Funktion des Material Editors der Unreal Engine 2. Über den Modifier lassen sich verschiedene Funktionen auf ein Material anwenden, um so aus einer Textur mehrere verschiedene Materialien zu fertigen.

**Modifikation.**

Eine Modifikation, kurz Mod, ist eine Erweiterung eines käuflich zu erwerbenden Hauptprogrammes, meist ein Computerspiel. Ein Mod kann das gesamte ursprüngliche Spiel in allen Punkten (Grafik, Sound, Level, Modelle) ändern, oder aber auch nur Teile davon. Mods werden

normalerweise von Fans entwickelt und unentgeltlich über das Internet veröffentlicht.

**Mutator.**

Mit einem Mutator kann man in Unreal Tournament die Spielregeln eines Gametypes leicht beeinflussen oder verändern, ohne dabei gleich einen komplett neuen Gametype zu kreieren.

**MZH.**

Mehrzweckhochhaus. Gebäude der Universität Bremen, in dem Informatikstudenten einen Großteil ihrer Veranstaltungen haben.

**Opacity.**

Eine Funktion des Material Editors der Unreal Engine 2. Mit Hilfe des Alphakanals einer Textur lässt sich mit der Opacity Funktion festlegen, welche Teile der Textur sichtbar sein sollen und welche nicht.

**OpenGL.**

OpenGL ist eine Spezifikation für ein plattform- und programmiersprachenunabhängiges API (Application Programming Interface) zur Entwicklung von 3D Computergrafik. Der OpenGL-Standard beschreibt etwa 250 Befehle, die die Darstellung komplexer 3D Szenen in Echtzeit erlauben. Hersteller können jedoch auch eigene Erweiterungen definieren.

**Output Blending.**

Eine Funktion des Material Editors der Unreal Engine 2. Über das OutputBlending lässt sich bestimmen, wie ein Material, das aus verschiedenen Materialeigenschaften besteht, überblendet werden soll. Zur Auswahl stehen hier additiv, multiplikativ, Alphakanal und noch einige weitere. Je nach gewähltem Überblendungsverfahren ändert sich die tatsächliche Darstellung des Materials.

**Partikelsystem.**

Ein System, um Effekte zu erstellen, die aus verschiedenen Partikeln bestehen und keine klare geometrische Form haben. Beispiele hierfür sind: Rauch, Feuer, Wasserdampf / Wasserspritzer.

**Physikengine.**

Simuliert physikalische Eigenschaften in einer virtuellen Welt. Dadurch fallen Objekte so wie in der Wirklichkeit, wenn man z.B. dagegen läuft.

**Plugin.**

Ein Plugin oder Ergänzungs- oder Zusatzmodul ist eine gängige Bezeichnung für ein Softwareprogramm, das in ein anderes Softwareprodukt eingeklinkt wird. Softwarehersteller definieren Schnittstellen zu ihren Produkten, mit deren Hilfe Dritte Erweiterungen  $\ddot{U}$  Plugins genannt  $\ddot{U}$  für diese Softwareprodukte programmieren können. Das Plugin erweitert dann die Funktionalität dieses Softwareprodukts. Weit verbreitete Beispiele für Plugins sind etwa das Acrobat Reader-

Plugin oder der Macromedia Flash Player für die verschiedenen Webbrowser.

**Player Actor.**

Der Player Actor ist ein wichtiges Objekt im Unreal Script. Über den Player Actor lassen sich viele Eigenschaften des Spielers bestimmen, festlegen oder abrufen.

**Polarisationsfilter.**

Lichtwellen breiten sich in verschiedenen Winkeln aus. Wenn man nun das Licht durch einen Polarisationsfilter leitet, werden dort nur Lichtwellen durchgelassen, die horizontal oder vertikal ausgerichtet sind. Benutzt man zwei Polarisationsfilter, einen der horizontale Lichtwellen durchlässt und einen der vertikale Lichtwellen durchlässt, kann man so das gesamte Licht ausfiltern.

**Polygon.**

Ein Polygon ist in der geometrischen Mathematik, wie der Name schon sagt, eine Form mit mehreren Ecken. In der 3D Computergrafik werden auch Polygone verwendet. Allerdings sind sie dort in der Regel dreieckig. Das liegt daran, dass der Computer Dreiecke einfacher und schneller berechnen kann als Vielecke. In der Computergrafik ist ein Polygon das kleinste Element eines 3D Modells.

**Projektive Textur.**

In der Unreal Engine 2 gibt es sogenannte Pro-

jektoren, die dazu benutzt werden, dynamische Licht- und Schatteneffekte zu generieren. Damit ein Projektor seine Arbeit tun kann, benötigt er eine Projektive Textur. Eine solche Textur ist in Graustufen gehalten. Der Graustufenwert  $RGB=127,127,127$  wird hinterher transparent dargestellt. Niedrigere Werte werfen Schatten, höhere werfen einen Lichtschein.

Man kann aber auch ganz normale Texturen oder Bilder als Projektive Textur verwenden. Dann wird das Bild über all das projiziert, welches im Frustrum der Projektionsfunktion liegt.

**Rendern.**

Rendern bezeichnet den Vorgang aus Rohdaten durch Anwendung geeigneter Verfahren, neue Daten zu generieren. In der Computergrafik, besonders der 3D Computergrafik, wird durch das Rendern aus der Beschreibung einer Szene (Objekte, ihre Position und Eigenschaften) ein Bild erzeugt.

**Selfillumination.**

Eine Funktion des Material Editors der Unreal Engine 2. Wie der Name schon sagt sorgt diese Funktion dafür, dass die Textur den Eindruck eines Leuchtens erweckt.

**Shader.**

Ein kleines Programm, das der Grafik Engine mitteilt, welche Eigenschaften das Material ei-

ner Oberfläche besitzt. Einige programmierbare Eigenschaften sind: Textur, Reflektionen, Specularity, Partikeleffekt für ein Einschussloch, Laufsound (z.B. metallisch oder hölzern).

**Skybox.**

Die Skybox ist ein Trick der Unreal Engine 2 um Außenwelten mit einem nahezu unendlich entfernt wirkenden Hintergrund und Sonnenlicht zu simulieren. Eine Skybox besteht aus einem Würfel, in dessen Inneren verschiedene Hintergrundobjekte und ein 360 Grad Hintergrundbild mit Boden und Himmel auf die einzelnen Würfelseiten platziert werden. Als nächstes werden die Begrenzungswände des Außenareals mit der Skybox verknüpft. Der Inhalt der Skybox wird nun auf die Begrenzungswände des Außenareals projiziert. So entsteht der Eindruck eines Außenlevels.

**Specularity.**

Eine Funktion des Material Editors der Unreal Engine 2. Mit Hilfe des Alphakanals einer Textur lässt sich mit der Specularity Funktion ein Highlight auf einer Textur erzeugen.

**Specularity Mask.**

Mit Hilfe der Specularity Mask wird bestimmt, an welchen Stellen eines Materials Spiegelungen vorhanden sind und wie stark diese sind. Die Specularity Mask wird aus dem Alphakanal der Textur gewonnen. Je heller ein Bildpunkt, desto sichtbarer ist an dieser Stelle die Spiegelung.

**Spiegelmaske.**

Siehe Specularity Mask.

**Static Mesh.**

Um 3D Dekorationsobjekte in einem Unreal Level platzieren zu können, müssen diese in irgendeinem 3D Datenformat vorliegen. Bei der Unreal Engine 2 heisst dieses Static Mesh. Das Besondere an dem Static Mesh ist, dass es als ein Objekt in den BSP - Baum eingebaut wird. So darf ein Static Mesh aus tausenden von Polygonen bestehen, ohne dass die Systemperformance zu stark einbricht. Es wird auch nur einmal Speicherplatz für das Static Mesh benötigt, sollte man es mehrfach platzieren.

**Teleporter.**

Ein Teleporter verbindet zwei Punkte einer Map miteinander. Man kann die andere Gegenstelle durch einen Teleporter hindurch nicht sehen, betritt man aber einen Teleporter, wird man auf den anderen Endpunkt teleportiert.

**Terraineditor.**

Ein Tool mit dessen Hilfe man Außenlandschaften mit Bergen und Täler praktisch malen kann. Als Grundlage dient hier eine Graustufen-Textur. Jeder Pixel der Textur entspricht einem Vertex in der 3D Welt. Je heller der Pixel ist, desto höher liegt der Vertex hinterher. Diese Methode ist mit dem Displacementmapping vergleichbar. Al-

lerdings wird sie vor dem Laden des Levels berechnet und nicht zur Laufzeit.

**TexEnvMap.**

Eine Funktion des Material Editors der Unreal Engine 2. Mit der TexEnvMap Funktion wird eine Cubemap als Environment Map auf den Specularity Kanal eines Materials projiziert. Es entsteht der Eindruck einer Spiegelung. Diese ist allerdings vorberechnet und findet nicht in Echtzeit statt. Dieses Verfahren spart so aber enorm viel Rechenleistung zur Laufzeit.

**TexOscillator.**

Eine Funktion des Material Editors der Unreal Engine 2. Mit der TexOscillator Funktion kann man eine Textur animieren. Wie der Name schon sagt, wird die Textur über eine oszillierende Funktion (z.B. eine Sinuskurve) animiert.

**TexPanner.**

Eine Funktion des Material Editors der Unreal Engine 2. Mit dem TexPanner kann eine Textur vertikal oder horizontal verschoben und animiert werden.

**TexRotator.**

Eine Funktion des Material Editors der Unreal Engine 2. Mit dem TexRotator kann eine Textur rotiert und so animiert werden.

**TexScaler.**

Eine Funktion des Material Editors der Unreal Engine 2. Mit dem TexScaler lässt sich eine Textur nachträglich skalieren und kann so auch animiert werden.

**Textur.**

Ein 2D Bild, das auf die Polygone eines 3D Objektes gelegt wird und so den Eindruck einer Oberfläche erzeugt.

**Texture Baking.**

Ein Verfahren, um die vielen verwendeten Texturen eines Objektes in einer Texturdatei zusammenzufassen. Dabei werden nur die Teile der verwendeten Originaltextur in die neue Texturdatei geschrieben, die auch wirklich auf einer Oberfläche des Objektes sichtbar sind.

**Texturspeicher.**

Der Texturspeicher ist ein Speicherbereich auf einer modernen Grafikkarte mit Hardware 3D Beschleunigung. Dieser Bereich wird benötigt, um die Texturen der darzustellenden virtuellen Welt auf Abruf bereit zu halten. Sollte der Texturspeicher auf der Grafikkarte nicht ausreichen, wird ein Teil des Hauptspeichers für diese Arbeit mit einbezogen.

**TwoSided Textur.**

Normalerweise hat ein Material nur eine Seite. Nämlich die, die dem Betrachter zugewandt ist. Um Rechenzeit zu sparen, wird die unsichtbare

Rückseite sofort ausgeblendet. Hat man aber ein Material mit Transparenz, kann es sein, dass man die Rückseite durch die Transparenz hindurch sieht. Stellt man jetzt das Material auf TwoSided wird auch die Rückseite dargestellt.

**Umgebungstextur.**

siehe Environmental Map.

**Unreal Developer Network.**

Das UDN ist eine von Epic gepflegte Webseite, die alle erdenklichen Informationen rund um die Unreal Engine bietet. Dort kann man viele Anleitungen finden, um bestimmte Funktionen der Unreal Engine richtig zu nutzen.

**Unreal Editor.**

Entwicklungsumgebung für die Unreal Engine. Hauptsächlich zur Erstellung von virtuellen Umgebungen, aber auch zum Erstellen und Bearbeiten von Unreal Script-Klassen.

**Unreal Einheiten.**

Eine Unreal Einheit ist die kleinste darstellbare Größe der Unreal Engine. Ein Meter entspricht ungefähr 52,2 UE (Unreal Einheiten).

**Unreal Script.**

Interne Programmierschnittstelle in der Unreal Engine. Mit Hilfe des Unreal Scripts lassen sich alle spielelogischen Abläufe beliebig ändern und festsetzen. Das Unreal Script ist eine Mischung

aus C++ und Java. Übersetzt wird eine Unreal Script Datei in einen Bytecode, der zur Laufzeit von einem Interpreter ausgeführt wird. Interessant ist hierbei, dass sich der Bytecode jederzeit wieder in den Sourcecode zurückübersetzen lässt und man so wieder Änderungen vornehmen kann.

**Vertex Animation Modell.**

Siehe Keyframe Animation Modell.

**Vertex Lighting.**

Vertex Lighting wird verwendet, um dynamische Lichteffekte zu generieren. Dabei werden die Lichtwerte an die einzelnen Vertices eines 3D Modells gebunden und dann mit der Textur, die auf den Vertices liegt, multipliziert. Diese Lichtberechnung kann zur Laufzeit ausgeführt werden, ist aber nicht sehr genau und kann auch keine Schatten von Objekten berechnen.

**Vertices.**

In der 3D Computergrafik ist ein Vertex (aus dem Lateinischen, plural: Vertices) ein Scheitelpunkt einer Primitive und enthält, neben einer Positionsangabe in Form eines Vektors, meistens noch einige andere Angaben wie zum Beispiel eine Farbe, Transparenz oder eine zweite Positionsangabe, die für andere Zwecke verwendet werden kann (z.B. Texturkoordinaten). Der Vertex ist ein fundamentales Prinzip bei der polygonalen Model-

lierung: Zwei Vertizes zusammen genommen ergeben die Endpunkte einer Linie, drei Vertizes definieren ein Dreieck, etc.

**Warpzone.**

Mit Hilfe der Warpzone lassen sich in der Unreal Engine verschiedene Punkte einer Map miteinander verbinden. Dabei müssen diese Punkte nicht unbedingt direkt nebeneinander liegen. Man kann durch eine Warpzone in die andere Zone hineinsehen und auch hindurchgehen.

**Wireframe.**

Siehe Drahtgittermodell.

**xPlayer-Klasse.**

Dies ist die Hauptklasse des Spielers in der Unreal Engine 2. Hier wird zum Beispiel bestimmt, wie die Bewegung berechnet wird, welche Perspektive der Spieler hat und noch sehr vieles mehr.

**ZAIT.**

Zentrum für angewandte Informationstechnologien. Aufgaben des ZAIT sind die technisch - wissenschaftliche Beratung und Betreuung bei der Durchführung von EDV - Anwendungen in Forschung und Lehre.

# Index

## A

A.R.M.	59, 62, 65, 66, 146
Aktoren	10, 21
Aktorik	18, 51
Alphakanal	39, 40, 45, 95, 98
Alphatest	45
Antiportal	88
Anwendung	<i>siehe</i> Cave
AR-Toolkit	72
artec	11, 22, 71, 74, 125, 127, 166, 175, 183, 201, 202
Ausklang	3, 149, 182, 196

## B

Beamer	7, 9, 11, 13–15, 30, 33, 73, 82, 124, 126, 127, 136, 137, 142, 202, 204
Befestigungstechnik	6, 10
Beleuchtung	36, 80, 81, 83, 88, 94, 104
Beleuchtungsmodell	104
Bespannung	2, 8–10, 16
Biped	61, 62
Blickwinkel	78, 87, 97
Blocking Volume	87, 89
Bot	49, 163, 191
BSP-Baum	35
BSP-Geometrie	35, 36, 79, 84, 85, 88, 115, 117

---

Bumpmapping .....	106
<b>C</b>	
Cave .....	2, 29, 51, 58, 77, 80–83, 88, 90, 189
Anwendung .....	58, 62, 74, 76
Charakteranimation .....	36
Clanwar .....	191
Combiner .....	43
Computerspiel .....	23, 34, 63, 196, 197
Constructive Solid Geometrie .....	35
Counter-Strike .....	23, 188, 190
Cubecam Mutator .....	46
Cubemap .....	43–45, 94, 95
<b>D</b>	
Dachprojektionsfläche .....	31, 33
Dance Dance Revolution .....	53
Detailskalierung .....	40
Displacement Mapping .....	106
Dorum .....	<i>siehe</i> Projektwochenende
Drahtgittermodell .....	42, 91, 92, 103, 130
Drucksensoren .....	15
<b>E</b>	
Easteregg .....	89
Ebene 1 .....	<i>siehe</i> MZH
Egoshooter .....	189
Eigenleuchten .....	40, 42
Eingabegeräte .....	2, 13, 51, 194, 195
Exoskelett .....	60

**F**

Feuerwehrpläne ..... *siehe* MZH  
FinalBlend ..... 45  
Folie ..... *siehe* Projektionsmaterial  
Freiheitsgrade ..... 60, 61

**G**

Gameport ..... 62–64  
Gametype ..... 47, 48  
Grafik-Engine ..... 2, 3, 5, 22, 34, 92, 106, 153, 196  
GW 2 ..... 6, 11, 15, 29, 86, 182, 204

**H**

Half-Life ..... 23, 25  
Half-Life Engine ..... 22, 23, 25  
Highmap ..... 36  
Holzrahmen ..... 7, 29, 124  
homogene Koordinaten ..... 69  
Host ..... 188

**I**

Indoor-Map ..... 79  
Interaktion ..... 2, 3, 51, 58, 197, 199

**J**

Joystick ..... 59, 62–65, 71

**K**

Karma-Engine ..... 34, 80  
    Karmaobjekte ..... 49, 80, 82  
Keyframe Animation ..... 36  
Kinematik ..... 61

Inverse Kinematik .....	61
Kinemattkette .....	61
Klappfolien .....	140
Kollisionsberechnung .....	87, 89, 92, 104
künstliche Intelligenz .....	48, 49
Kunsthalle Bremen .....	3, 18, 37, 101–123, 138
<b>L</b>	
LAN-Wochenende .....	3, 188–191
Laserpointer .....	71, 72, 138, 146
LED .....	53, 55–57
Leinwand .....	6, 8, 9, 16, 26, 73, 76
Lichtquelle .....	7, 30, 40, 42, 109, 110
Lightmap .....	36
<b>M</b>	
Madness Engine .....	5, 12, 22–25, 74
Map .....	36, 46, 48, 77, 81, 84–87, 104, 109, 112, 115, 119, 122, 189
Match .....	188, 190
Materialeditor .....	37, 196
MiCaDo .....	3, 11–13, 16, 124, 166, 183, 202
MiCarpet .....	3, 16, 166, 183
MiCasa .. 1–3, 5, 11, 14, 15, 81, 115, 116, 124, 129, 133, 135, 142, 150, 157, 163, 166–168, 177, 182–184, 188, 196, 199, 204	
MiCasa Stick .....	49, 50, 71, 91
Mixed Reality .....	58, 70
Mixed Reality Cave .....	129, 135
Mod .....	<i>siehe</i> Modifikation
Modellierung .....	74, 78, 79, 83–86, 91, 93, 129
Modellingsoftware .....	22, 104, 106, 107, 109, 112, 115
Modifikation .....	23, 24, 34, 46, 47, 91

- 
- Motion Capturing ..... 2, 58, 59, 70, 194  
Mutator ..... 46–48  
MZH ..... 3, 20, 74–98, 112, 115, 117, 119, 123, 126–128, 130, 138, 142, 144  
    Ebene 0 ..... 84, 90, 91  
    Ebene 1 ..... 3, 74, 76–78, 80–89, 91, 97, 98, 142, 144, 147  
    Ebene 9 ..... 20, 74, 84, 90  
    Feuerwehrpläne ..... 77, 84, 85  
    Senatssaal ..... 76, 78, 82, 84, 89, 100
- N**
- NBC-Giga ..... 52, 199  
Netcode ..... 22, 23  
NetImmerse Engine ..... 22–24  
Nienburg ..... *siehe* Projektwochenende  
Norderney ..... *siehe* Projektwochenende  
Numpad ..... 51
- O**
- Opacity ..... 39, 40, 42
- P**
- PacMan ..... 83, 90, 146  
Papier ..... *siehe* Projektionsmaterial  
Partikeleffekt ..... 37  
Partikelsystem ..... 196  
Perspektive ..... 87  
PlayerActor ..... 83, 88, 89  
Playground ..... 3, 112, 113, 115, 119  
PlayStation ..... 53  
Polarisationsfilter ..... 73  
Polygon ..... 35, 80, 104, 106  
Portal ..... 3, 86, 115, 119

- 
- Potentiometer ..... 59, 60, 63–65, 67  
Präsentation ..... 9  
Projekt1–3, 6, 9, 14, 18, 22–25, 46–48, 74, 115, 119, 124, 125, 127, 129, 130, 133, 135, 136, 149, 152, 163,  
177, 182–184, 190, 199, 201, 204  
    Projektraum ..... 3, 5, 6, 129, 136, 139, 183, 188, 189, 191, 201, 205  
Projektbericht ..... 2, 3, 197  
Projektion ..... 7, 9, 10, 15, 16, 26, 29–31, 33, 73, 126, 129, 199  
Projektionsfläche ..... 7–10, 29, 33  
Projektionsfolie ..... 29, 31, 32, 124, 136, 204  
Projektionsmaterial ..... 26, 30, 31, 204  
    Folie ..... 6, 9, 10, 14, 16, 26, 32, 33, 196, 202  
    Papier ..... 6, 7, 9, 11, 14, 15, 26, 30, 184  
    Segeltuch ..... 31  
    Stoff ..... 6, 7, 9, 10, 12, 14, 15, 26, 30, 202  
Projektive Textur ..... 37  
Projektor ..... 33, 36–38, 95, 110  
Projekttag ..... 3, 74, 89, 101, 112, 119, 123–130, 133, 142, 145–149, 184, 204, 205  
Projektwochenende ..... 74, 76, 80, 129, 140, 152, 154, 157, 162, 163  
    Dorum ..... 162  
    Nienburg ..... 74  
    Norderney ..... 80, 157  
Prototyp ..... 2, 5, 6, 11, 12, 14, 22, 26, 27, 29–31, 61, 62, 70, 124, 125, 195, 201, 202, 204
- Q**  
Quake III ..... 74, 76, 77  
Quake III Engine ..... 22, 23, 74, 76
- R**  
Rahmen ..... 2, 5, 7–10, 14, 16, 18, 26, 29–32, 53, 57, 107, 129, 136, 159, 204  
Reflektionen ..... 37, 45, 86, 94, 95, 98, 110, 111  
Relais ..... 55–57

---

Rendern ..... 35, 46  
RGB-Wert ..... 40, 42, 45  
Rohmaterialklasse ..... 39, 45

**S**

SahneGametype ..... 46–48  
Segeltuch ..... *siehe* Projektionsmaterial  
Selfillumination ..... 40  
Senatssaal ..... *siehe* MZH  
Sensoren ..... 10, 15, 21  
Sensorik ..... 18, 51, 71, 202  
SFG ..... 14, 166  
Shader ..... 23, 37, 39, 40, 42, 43, 86, 91, 94, 97, 98, 104, 196  
Simulation ..... 16, 21, 35, 36  
Skizzen ..... 74, 78, 79, 84, 93  
Skybox ..... 36, 86–88, 91  
Sourcecode ..... 22–24, 34  
SpecularityMask ..... 39, 95, 97  
Spiegel ..... 33, 95, 122  
Sponsoren ..... 3, 124, 129, 130  
Static Meshes ..... 35, 79, 80, 82, 84–86, 90–93, 112, 120  
Stoff ..... *siehe* Projektionsmaterial  
Streichholz ..... 136, 182, 183

**T**

Tanzmatte ..... 2, 16, 51–55, 57, 83, 112, 125, 127, 137, 144, 199, 200, 204  
Taschenlampe ..... 21, 91  
Tastatur ..... 13, 55, 56, 71  
Tastaturkontroller ..... 55, 56, 199  
Teleporter ..... 90, 111, 119  
Terraineditor ..... 36, 196

---

Texture Baking .....	107
Texturen . 39, 42, 45, 76, 78, 80, 82–84, 86, 87, 92, 94, 97, 98, 101, 102, 104, 106, 107, 112, 115, 119, 163	
Detailtextur .....	40
Detailtexturen .....	40, 41
Texturkoordinaten.....	92
Umgebungstexturen.....	39, 43, 86, 95, 98
Texturspeicher .....	43
Transparenz .....	37, 45
<b>U</b>	
UDP .....	71, 72
Universität Bremen .....	1–3, 74, 175, 195
Unreal Developer Network .....	77
Unreal Editor .....	34, 35, 37, 43, 74, 79, 82, 84, 86, 92, 95, 102, 107, 109, 110, 112, 113, 197
Unreal Engine 2 .....	22, 24, 34, 35, 37, 47, 74, 76, 106, 119, 120, 196, 204
Unreal Script .....	34, 46, 196
Unreal Tournament 2003 .....	24, 25, 34, 37, 39, 46–49, 77, 81, 84, 125, 188, 189
Unrealeinheiten .....	49, 77, 85, 89, 91, 102, 104
<b>V</b>	
Vektoren .....	47, 56
Vertex .....	36
Vertex Animation Modell .....	34
Vertex Lighting .....	36
virtuelle Umgebung2, 3, 37, 74, 83, 91, 96, 101, 102, 104, 107, 112, 115, 119, 120, 126, 142, 189, 196, 197	
virtuelle Welt .....	3, 12, 16, 21, 22, 29, 31, 34, 47, 50, 51, 58, 90, 92, 94, 97, 112, 117, 119, 136, 138, 141
virtuellen Welt .....	199
Vulpine-Engine .....	22, 24
<b>W</b>	
Warfare .....	196, 197
Warpzone .....	111, 119, 120, 122, 123

---

Würfel ..... 43, 119, 130, 142, 149, 184, 187

**Z**

Zauberstab ..... 71, 72, 137, 163

Zielfindung ..... 2, 5, 74

## Das artecLab:

bildet eine experimentelle Gruppe von Wissenschaftlern, Ingenieuren und Künstlern.  
Wir analysieren und erproben formale und nicht-formale Methoden der Modellierung, Produktion und Simulation.

Wir konstruieren sensorisierte Computer-Umgebungen und erforschen neue Formen der Mensch-Maschine-Interaktion.

Wir experimentieren mit der Vermischung von realen und virtuellen Welten an der Grenze zwischen maschineller Funktion und menschlicher Phantasie.

Wir sind Grenzgänger auf den Gebieten Kunst, Arbeit und Technik: Art, Work and Technology.

Der Mixed Reality Ansatz eröffnet neue Sichtweisen.

Wir modellieren mit realen Gegenständen, die eine reiche sinnliche Erfahrung mit der Widerspenstigkeit realer Phänomene vermitteln.

Wir formen virtuelle Gegenstände, die vielfältige Übersetzungen zwischen konkreten und abstrakten Sichtweisen realisieren.

Wir bauen Schnittstellen und Interfaces, die komplexe Verhältnisse zwischen der realen Welt der physischen Gegenstände und der virtuellen Informationswelt erfahrbar machen.

Die Computer-Wissenschaften und ihre mathematischen Grundlagen haben eine eigene Ästhetik.

Wir verstehen Ästhetik als Balance zwischen sinnlicher Erfahrung und verstandesmäßiger Durchdringung der uns umgebenden Phänomene.

Wir haben das Ziel, eine spielerische Erfahrung der Mensch-Maschine-Beziehung zu ermöglichen - auch jenseits der Grenzen von Rationalität, Nützlichkeit oder Effizienz.

Wir verfolgen gleichzeitig einen partizipatorischen und sozial verpflichteten Ansatz.

## artecLab Paper

1. Jörg Richard, F. Wilhelm Bruns, Mensch und Maschine im Spielraum - Technische Praxis und Ästhetische Erfahrung
2. F. Wilhelm Bruns, Hyperbonds - Applications and Challenges
3. Yong-Ho Yoo, Energy Interface for Mixed Reality Design
4. Micado, Projektbericht des studentischen Projekts micado zu Mixed Reality Caves
5. Micarpet, Projektbericht des studentischen Projekts micarpet zu Mixed Reality Caves
6. Micasa, Projektbericht des studentischen Projekts micasa zu Mixed Reality Caves
7. Bernd Robben, Ralf Streibl, Alfred Tews, Mixed Reality Adventures, Bericht vom Symposium im Kino 46