

Virtuelle Präsentation des Instituts für Kartografie

Lokalisierung und 3D-Rundgang



Diplomarbeit, Sommersemester 2007

Autorin: Leila Dedual

Leitung: Prof. Lorenz Hurni, Institut für Kartografie, ETH Zürich

Betreuung: Ionut Iosifescu Enescu, Institut für Kartografie, ETH Zürich
Claudia Matthys, Institut für Kartografie, ETH Zürich

VORWORT

Dieser Bericht beschreibt das Resultat meiner Diplomarbeit am Institut für Kartografie der Eidgenössischen Technischen Hochschule Zürich. Die Diplomarbeit wurde während des Sommersemesters 2007 unter der Betreuung von Ionut Iosifescu Enescu und Claudia Matthys erarbeitet.

Ich möchte mich bei meinen beiden Betreuern, die mich stets unterstützten und von deren Hilfe ich viel profitieren konnte, herzlich bedanken. Ebenso möchte ich Prof. Lorenz Hurni dafür danken, dass er diese Diplomarbeit am Institut für Kartografie ermöglichte.

Ein weiterer Dank geht an Dr. Christian Häberling und Juliane Cron, welche durch die Bereitstellung von Daten ebenfalls zum Gelingen dieser Diplomarbeit beitrugen.

Zürich, Juli 2007

Die Autorin

Leila Dedual

Bachstrasse 2

CH - 8600 Dübendorf

Telefon: +41 43 355 96 67

ldedual@student.ethz.ch

ZUSAMMENFASSUNG

Die Virtuelle Realität ist eine Technik, die es ihren Benutzern erlaubt, in eine dreidimensionale Welt einzutauchen und mit dieser Welt zu interagieren. Diese Pseudo-Wirklichkeit und ihre physikalischen Eigenschaften werden von einem Computer in Echtzeit generiert.

Diese Diplomarbeit befasst sich mit der Erstellung einer solchen virtuellen Welt. Das Ziel besteht im Aufbau einer Applikation zur Lokalisierung des Instituts für Kartografie der ETH Zürich sowie in der Erstellung einer virtuellen 3D-Präsenz dieses Instituts.

Das gesetzte Ziel wird in zwei Schritten erreicht. Zuerst wird die Animation einer drehenden und zoomenden Weltkugel erstellt. Sie ist auf der Support-Website des Instituts platziert und dient als Blickfang mit dem Zweck, eine grobe Lokalisierung des IKA auf der Landes-Ebene aufzuzeigen.

Der virtuelle Rundgang stellt den zweiten Schritt und den grösseren Teil der Diplomarbeit dar. Als Grundlage wird mit einer Modellierungssoftware ein texturiertes dreidimensionales Modell des Instituts erstellt. Dieses Modell wird anschliessend – in verkleinerter Form – als VRML-Datei exportiert.

Der Rundgang wird als einfaches 3D-Informationssystem für Gebäude-Innenräume konzipiert. Er wird in der Beschreibungssprache VRML erstellt und bietet die Möglichkeit, sich den Weg zu den Büros des Instituts anzeigen zu lassen. Links zu externen Webseiten bieten weitere Informationen über einzelne Projekte, die in den Büros bearbeitet werden. Die in den Fluren hängenden Poster können als Pdfs angesehen werden.

Das Modell lässt verschiedene Möglichkeiten der Weiterbearbeitung zu. Sowohl eine Veränderung der Textur und Geometrie als auch die Ausgabe in einem anderen Datenformat ist möglich. Das Modell lässt sich beliebig erweitern und beispielsweise als Ebene in ein Computerspiel einfügen. Dadurch kann ein erweiterter Rundgang mit zusätzlichen Funktionen erstellt werden; es besteht die Möglichkeit, ein interaktives Informationssystem zu entwickeln, weitere Navigationsfunktionen einzufügen sowie das Modell beispielsweise mit den Institutsbüros der Etage E zu erweitern. Mit der vorliegenden Arbeit ist das Grundgerüst für diese Weiterbearbeitungen gegeben.

Als Beispiel zur Aufzeigung der Anpassungsmöglichkeiten des Modells wird der Rundgang mit einer anderen Textur versehen und leicht abgeändert. Das Resultat stellt das Institut so dar, wie es nach der Realisierung des neuen Gestaltungskonzeptes aussehen wird.

Am Schluss der Arbeit wird der für die Erfüllung des Ziels gewählte Lösungsweg kritisch beurteilt. Ausserdem werden die Vor- und Nachteile der benutzten Programme aufgezeigt und als Ausblick mögliche Weiterverwendungen des Modells aufgezeigt.

INHALTSVERZEICHNIS

ABBILDUNGSVERZEICHNIS	VI
TABELLENVERZEICHNIS	VII
ABKÜRZUNGSVERZEICHNIS	VII
1 EINLEITUNG	1
1.1 Motivation	1
1.2 Zielsetzung	1
1.3 Inhaltsübersicht	2
2 KONZEPTION DER VIRTUELLEN PRÄSENTATION	3
2.1 Geplantes Vorgehen	3
2.1.1 Die Animation der Weltkugel	3
2.1.2 Der Rundgang durch das Institut	4
2.2 Änderungen zum geplanten Vorgehen	5
2.2.1 Schritt 2: Vereinfachter Rundgang	6
2.2.2 Schritt 3: Rundgang mit kompletter Funktionalität	6
2.3 Bestehende Arbeiten	6
2.3.1 Realisierung eines Geoinformationssystems	6
2.3.2 Modellierung der Osnabrücker Altstadt in SketchUp	7
2.3.3 Virtueller Rundgang durch ein Büro	7
2.3.4 3D-Benutzerschnittstelle zur Konfiguration und Visualisierung von Digitalen Fabriken	8
3 TECHNISCHE GRUNDLAGEN	9
3.1 Animationen fürs Web	9
3.1.1 SVG	9
3.1.2 Macromedia Flash	9
3.2 3D-Grafik-Software	10
3.2.1 3D Studio MAX	10
3.2.2 Maja	11
3.2.3 Google SketchUp und SketchUp Pro 5	11
3.3 3D-Modellierungssprachen – VRML	11
3.4 Game-Engines	12

3.4.1	Definition und Aufbau	13
3.4.2	Ausgewählte Game-Engines	14
4	AUSFÜHRUNG TEIL 1: DIE FLASH-ANIMATION	16
4.1	Vorgehen	16
4.1.1	Rotation	16
4.1.2	Zoom Teil 1: Welt – Europa	17
4.1.3	Zoom Teil 2: Europa – Schweiz	18
4.1.4	Überblendung zur ETH Hönggerberg	19
4.1.5	Ticker	19
4.2	Verwendete Daten	20
4.3	Ergebnis	21
5	AUSFÜHRUNG TEIL 2: DER VIRTUELLE RUNDGANG	22
5.1	Vorgehen	22
5.1.1	Erstellung des 3D-Modells	22
5.1.2	Vereinfachung des Modells	26
5.1.3	Neues Gestaltungskonzept	27
5.1.4	Implementierung des Rundganges	27
5.1.5	Rundgang mit kompletter Funktionalität	30
5.2	Verwendete Daten	30
5.3	Ergebnisse	31
5.4	Anleitung zur Weiterbearbeitung	34
5.4.1	Änderungen an den Modellen	34
5.4.2	Änderungen an den VRML-Funktionen	35
6	AUFLISTUNG UND KRITISCHE BEURTEILUNG DER ERGEBNISSE	36
6.1	Erstellte Produkte	36
6.1.1	Flash-Animation	36
6.1.2	(Vollständiges) Modell des IKA	36
6.1.3	Schlankes Modell für VRML / Web	37
6.1.4	Schlankes Modell mit neuem Gestaltungskonzept	37
6.1.5	Rundgänge in VRML	37
6.2	Beurteilungen und Erkenntnisse	37
6.2.1	Beurteilung der Produkte	37
6.2.2	Der Lösungsweg und die verwendete Software	38

7	AUSBLICK	42
7.1	Ausbau der Flash-Animation	42
7.2	Erweiterung der 3D-Modelle	42
7.2.1	Vervollständigung des grossen Modells	42
7.2.2	Optimierung der kleinen Modelle	43
7.3	Weitere Funktionen für die VRML-Rundgänge	44
7.4	Einbindung des Modells in ein Computerspiel	45
8	QUELLENVERZEICHNIS	46
8.1	Literatur	46
8.2	Internet	46
8.3	Daten	48
A	ANHANG.....	A-1
A 1	Macromedia Flash.....	A-1
A 1.1	Aufbau des Dokumentes	A-1
A 1.2	Animationen mit Hilfe von Tweens.....	A-2
A 1.3	Zeitleisten-Effekte	A-3
A 1.4	Masken	A-3
A 2	Google SketchUp	A-4
A 2.1	Das Koordinatensystem von SketchUp.....	A-4
A 2.2	Masseinheiten einstellen	A-4
A 2.3	Point Inference.....	A-5
A 2.4	Line Tool.....	A-5
A 2.5	Push/Pull-Tool.....	A-6
A 2.6	Tape Measure Tool	A-7
A 2.7	Texturierung	A-8
A 2.8	Komponenten und Instanzen	A-10
A 2.9	Referenzieren eines Modells	A-11
A 2.10	Import- und Exportformate von SketchUp.....	A-11
A 3	VRML	A-13
A 3.1	Definitionen und Funktionen.....	A-13
A 3.2	VRML-Viewer	A-20
A 4	In SketchUp verwendete Materialien	A-23

ABBILDUNGSVERZEICHNIS

Abbildung 1: Ursprünglich geplantes Vorgehen in zwei Schritten	3
Abbildung 2: Neues Konzept mit drei Schritten.....	6
Abbildung 3: Die beiden Welt-Bänder	16
Abbildung 4: Die optische Täuschung.....	17
Abbildung 5: Die Bühnenansicht nach Ende der 1. Zoomstufe	18
Abbildung 6: Die Schweiz nach dem Ende der zweiten Zoomstufe.....	19
Abbildung 7: Zustand am Ende der Animation.....	20
Abbildung 8: Kachelung des Bodens.....	24
Abbildung 9: Kacheleffekt	25
Abbildung 10: Struktur des VRML-Codes	28
Abbildung 11: Die Start-Ansicht des Rundgangs.....	31
Abbildung 12: Das IKA im neuen Kleid.....	32
Abbildung 13: Liste der Viewpoints.....	33
Abbildung 14: Leeres Flash-Dokument	A-2
Abbildung 15: Darstellungsarten in der Zeitleiste.....	A-3
Abbildung 16: Das Koordinatensystem von SketchUp.....	A-4
Abbildung 17: Die wichtigsten Fälle des Point Inference.....	A-5
Abbildung 18: Value Control Box.....	A-5
Abbildung 19: Anwendung des Push/Pull-Tools.....	A-7
Abbildung 20: Anpassung einer Textur	A-9
Abbildung 21: Koordinatensystem von VRML	A-14
Abbildung 22: Cortona-Player Version 5.1 von ParallelGraphics.....	A-22

TABELLENVERZEICHNIS

Tabelle 1: Die Import- und Exportmöglichkeiten von Google SketchUp und SketchUp Pro 5A-12

Tabelle 2: In SketchUp verwendete MaterialienA-26

ABKÜRZUNGSVERZEICHNIS

DLL	Dynamic Link Library
ETH	Eidgenössische Technische Hochschule
FLA	Flash (Format der Flash-Quelldateien)
HIL	Lehrgebäude Bauwesen (Hönggerberg Ingenieurwissenschaften Lehrgebäude)
IKA	Institut für Kartografie
SVG	Scalable Vector Graphics
SWF	Small Web Format / ShockWave Flash (kompilierte Flash-Dateien)
VCB	Value Control Box
VR	Virtual Reality (Virtuelle Realität)
VRML	Virtual Reality Modeling Language
XML	Extensible Markup Language

1 EINLEITUNG

Die Virtuelle Realität (engl.: Virtual Reality, VR) ist eine Technik, die es ihren Benutzern erlaubt, in eine dreidimensionale Welt einzutauchen und mit dieser Welt zu interagieren. Diese Pseudo-Wirklichkeit und ihre physikalischen Eigenschaften werden von einem Computer in Echtzeit generiert.

Die Wurzeln der Virtuellen Realität liegen in den (mechanischen) Simulatoren zur Ausbildung von Kampffliegern im 2. Weltkrieg sowie in den Computergrafik-Forschungen der frühen 60-er Jahre [1]. Heute kommen Virtuelle Realitäten in vielen Bereichen zum Einsatz. Bekannte Einsatzgebiete sind beispielsweise die Ausbildung von Piloten (Flugsimulatoren) und Ärzten (virtuelle Operationen), die Erstellung von virtuellen Prototypen in der Industrie sowie Visualisierungen in der Architektur. Virtuelle Welten werden ausserdem in Form von Computerspielen stark kommerziell genutzt.

Die vorliegende Diplomarbeit befasst sich mit einem weiteren Einsatzgebiet für VR. Es wird eine virtuelle Welt erstellt, mithilfe welcher das Institut für Kartografie (IKA) präsentiert werden kann.

1.1 MOTIVATION

Als Motivation zur Ausführung dieser Diplomarbeit steht der Wunsch im Vordergrund, die virtuelle Präsenz des Instituts für Kartografie zu erweitern. Das Institut soll auf eine moderne, unkonventionelle Art vorgestellt werden. Das Kennenlernen und der Umgang mit der dafür benötigten Software ist eine interessante und anspruchsvolle Aufgabe und ein weiterer Faktor bei der Wahl dieses Diplomarbeits-Themas.

1.2 ZIELSETZUNG

Das Ziel der Diplomarbeit besteht darin, das Institut für Kartografie virtuell vorzustellen. Die Zielgruppe setzt sich vor allem aus Personen zusammen, welchen das IKA nicht vertraut ist, die sich aber für das Institut interessieren. Diese Interessenten sollen als Erstes einen groben Überblick darüber erhalten, wo sich das Institut befindet; als Zweites sollen sie sich innerhalb des IKA zurechtfinden können. Beispielsweise sollen sie erkennen, wo welches Büro zu finden ist.

Es gibt eine Vielzahl von Möglichkeiten und Instrumenten, wie eine solche virtuelle Präsentation erstellt werden kann. In der vorliegenden Arbeit soll der von der Autorin gewählte Weg beschritten werden. Anschliessend soll beurteilt werden, wie sinnvoll der gewählte Weg war. Die Vor- und Nachteile der verwendeten Software sollen ausserdem kritisch beurteilt werden.

In einem ersten Teil soll die Animation einer Weltkugel erstellt werden. Die Weltkugel soll sich drehen und schliesslich so weit hineinzoomen, dass der Standort des IKA lokalisiert werden kann. Die Animation soll übers Web mit den gängigen Browsern abgespielt werden können. Sie soll primär als Blickfang dienen, um die Aufmerksamkeit der Besucher der Website auf den virtuellen Rundgang durch das Institut zu richten, welcher mittels eines Hyperlinks gestartet werden kann.

Der virtuelle Rundgang durch das Institut für Kartografie stellt den zweiten Teil der Arbeit dar. Als Grundlage für den Rundgang muss ein 3D-Modell des IKA erstellt und mit Fotos texturiert werden. Das Modell muss dabei so erstellt werden, dass es auch über die Diplomarbeit hinaus weiterverwendet werden kann. Spätere Änderungen oder Nachführungen müssen deshalb möglichst einfach und schnell vorgenommen werden können. Dazu soll eine passende Software gefunden werden.

Anschliessend soll aus dem Modell der virtuelle Rundgang erstellt werden. Dieser soll so beschaffen sein, dass sich der interessierte Benutzer innerhalb des virtuellen Instituts frei bewegen kann. An geeigneten Stellen sollen Informationen zum Institut sowie zu den einzelnen Büros angezeigt werden. Der fertige Rundgang soll ausserdem im Web bereitgestellt werden. Es soll getestet werden, mit welcher Software er am zweckmässigsten erstellt werden kann.

1.3 INHALTSÜBERSICHT

In Kapitel 2 wird das Konzept der Diplomarbeit erläutert. Zuerst wird das geplante Vorgehen beschrieben und anschliessend erklärt, wieso dieses ursprüngliche Konzept angepasst werden musste.

Kapitel 3 beschreibt die technischen Grundlagen, welche zur Erreichung der Ziele der Diplomarbeit erarbeitet werden.

In den Kapiteln 4 und 5 werden die Ausführung der einzelnen Arbeitsschritte sowie die verwendeten Daten und die erzielten Ergebnisse detailliert beschrieben.

Kapitel 6 liefert eine kritische Beurteilung der Ergebnisse sowie der verwendeten Software. Der Ausblick in Kapitel 7 rundet die Arbeit ab.

Im Anhang werden die wichtigsten Funktionen der in der Diplomarbeit verwendeten Programme näher beschrieben.

2 KONZEPTION DER VIRTUELLEN PRÄSENTATION

2.1 GEPLANTES VORGEHEN

Wie in Kapitel 1.2 erwähnt, besteht die virtuelle Präsentation des Instituts für Kartografie aus zwei Teilen: der drehenden Weltkugel und dem Instituts-Rundgang. Das Konzept sah ursprünglich vor, diese beiden Teile in zwei Schritten anzugehen (Abbildung 1). Im Folgenden wird erörtert, wie das Vorgehen zur Erstellung dieser zwei Schritte ursprünglich gedacht war. Im Kapitel 2.2 wird auf die im Laufe der Arbeit vorgenommenen Änderungen – insbesondere auf die Aufteilung der ursprünglich zwei Schritte auf drei Schritte – eingegangen.

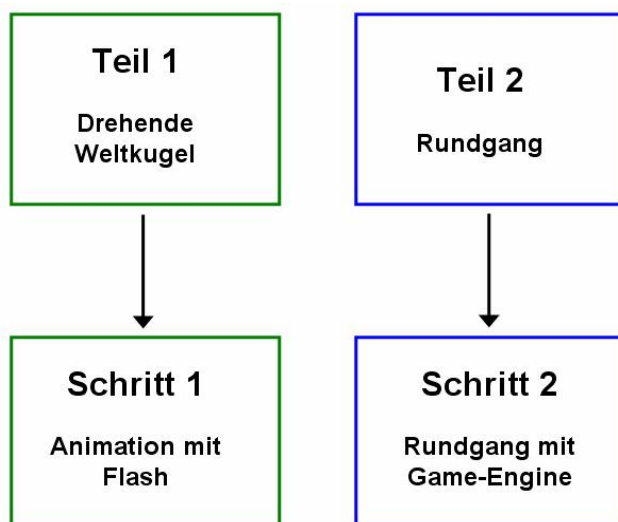


Abbildung 1: Ursprünglich geplantes Vorgehen in zwei Schritten

2.1.1 DIE ANIMATION DER WELTKUGEL

Die Animation soll mit Macromedia Flash Professional 8 erstellt werden. Ausgangslage der Animation ist eine Weltkugel, auf welcher die Kontinente sichtbar sind. Sie soll automatisch um ihre Achse rotieren. Der Drehsinn ist dabei derselbe wie in Natura.

Nach einer gewissen Zeit soll die Rotation stoppen. Es ist geplant, ein möglichst stufenloses Hineinzoomen von der Ausgangsstufe Welt über die Stufen Europa, Schweiz und Zürich bis zur Stufe ETH Höggerberg zu erreichen. Während des Zoomens soll durch einen Pfeil oder einen Kreis die jeweils nächste Stufe angezeigt werden:

- Anzeigen von Europa auf Stufe Welt
- Anzeigen der Schweiz auf Stufe Europa

- Anzeigen von Zürich auf Stufe Schweiz
- Anzeigen der ETH Hönggerberg auf Stufe Zürich
- Anzeigen des IKA auf Stufe ETH Hönggerberg

Während der gesamten Animation sollen die Koordinaten der ETH Hönggerberg (oder andere Informationen) als Ticker¹ unter der Animation durchlaufen.

Unterhalb der Animation soll ein Link eingefügt werden, welcher beim Anklicken ein neues Fenster mit dem Startpunkt des Rundgangs öffnet.

Damit die Animation übers Web problemlos abgespielt werden kann, darf die Datei nicht grösser als 1 MB sein. Ausserdem muss der Betrachter Flash installiert haben. Ein Link auf die Website von Adobe sorgt dafür, dass die aktuelle Flash-Version heruntergeladen werden kann.

Zur Implementierung der Weltkugel sowie der einzelnen Zoomstufen sollen geeignete Ausgangsdaten (Raster- oder Vektordaten) gesucht werden. Diese sollen gegebenenfalls in Adobe Illustrator² bearbeitet werden.

2.1.2 DER RUNDGANG DURCH DAS INSTITUT

Da kein bereits vorhandenes 3D-Modell des IKA gefunden wird, muss es neu erstellt und texturiert werden. Dabei sollen nur die Räumlichkeiten im G-Stock modelliert werden; die ebenfalls zum Institut gehörenden Büros E 24.3 und E 25 (zwei Etagen tiefer im E-Stock gelegen) können weggelassen werden. Jedoch müssen auch Teile des D-Stockes modelliert werden, da sich hier der Haupteingang des HIL-Gebäudes befindet. Der Weg vom Haupteingang des HIL bis zum Fahrstuhl und hinauf zu den Räumlichkeiten des IKA soll nämlich ebenfalls begangen werden können.

Es müssen primär nur die Flure und die wichtigsten Büros modelliert werden. Die restlichen Büros und Räume können zu einem späteren Zeitpunkt ergänzt werden. Ausserdem soll die Datei möglichst klein gehalten werden, damit bei der Darstellung später keine Probleme auftreten.

Nach der Erstellung des Modells muss der Rundgang erstellt werden. Es soll möglich sein, sich mithilfe der Computermaus frei durch das Modell zu bewegen. Die Türen sollen geöffnet und geschlossen werden können. Steht man vor der Tür eines Raumes bzw. betritt man ihn, sollen in Form von Fotos oder Text Informationen über den Raum sowie über allfällige darin laufende Projekte dargestellt werden. Ausserdem sollen allgemeine Informationen über das Institut verfügbar sein.

Zur Unterstützung des realitätsnahen Empfindens ist geplant, Töne (Sound) einzubetten. Dadurch können Geräusche wie Schritte, Anklopfen, das Öffnen einer Tür und das Klicken mit der Maustaste simuliert werden. Der Rundgang soll jedoch ohne Sound genau gleich informativ sein. Es sollen also keine Informationen in Sound verpackt werden.

¹ Ein Ticker ist ein Laufband, das aktuelle Informationen zeigt.

² Adobe Illustrator ist ein Vektorbasiertes Zeichen- und Grafikprogramm.

Der Rundgang soll in einer Game-Engine (siehe Kapitel 3.4) erstellt werden. Dazu soll das 3D-Modell exportiert und als Ebene ins Spiel eingefügt werden. Danach können Manipulationen am Spiel vorgenommen werden, wie beispielsweise das Entfernen allfällig vorhandener Waffen.

Jedermann, der sich den Rundgang im Web anschauen möchte, muss vorgängig diese Game-Engine auf seinem Rechner installieren. Deshalb muss eine Game-Engine ausgewählt werden, welche gratis heruntergeladen werden kann und möglichst auf allen drei Betriebssystemen Windows, Mac OS und Linux läuft. Ausserdem sollte die Grösse der Download-Datei möglichst klein sein, damit der Download nicht zu lange dauert.

Es ist geplant, eine Liste mit den nötigen und sinnvollen Anforderungen an die zu wählende Game-Engine aufzustellen. Anhand dieser Liste sollen aktuelle Game-Engines untersucht und die geeignetste ausgewählt werden. Es sollen ausserdem auch einfache Testmodelle erzeugt werden, anhand derer die in Frage kommenden Game-Engines direkt miteinander verglichen werden können.

Als Ergänzung zum virtuellen Rundgang können diverse Zusatz-Informationen und -Tools eingefügt werden, wie beispielsweise:

- Möglichkeiten zum Ausdruck verschiedener Pläne im Druckerraum
- aktuelle Mensa-Menükarten im Flur
- Busfahrpläne
- neues Gestaltungskonzept des IKA
- automatische Animation des Rundganges
- Wegbeschreibung ab Zürich Hauptbahnhof und ab den Autobahnausfahrten; evtl. mit Animation
- Wegbeschreibung / genaue Lokalisierung des IKA innerhalb der ETH Hönggerberg

2.2 ÄNDERUNGEN ZUM GEPLANTEN VORGEHEN

Im Laufe der Diplomarbeit wurde untersucht, wie der virtuelle Rundgang nicht nur ins Web gestellt, sondern auch direkt übers Web angesehen werden kann. Mit einer Game-Engine ist dies nicht möglich; der Interessent müsste sich den Rundgang herunterladen, um ihn ansehen zu können. Deshalb führte dieses erweiterte Ziel zu einer Änderung des Konzepts. Das neue Konzept besteht nun nicht mehr aus zwei, sondern aus drei Schritten. Der erste Schritt – die Animation der Weltkugel – wurde dabei so belassen, wie im ersten Konzept (Kapitel 2.1.1) vorgesehen.

Der Rundgang soll nun jedoch in die Schritte 2 und 3 aufgeteilt werden (Abbildung 2). Im Folgenden diese beiden Schritte genauer erläutert.

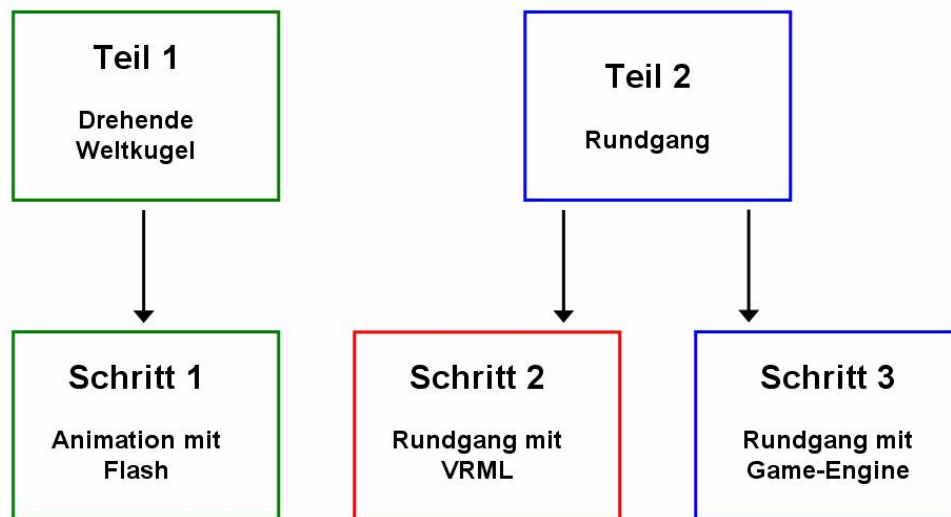


Abbildung 2: Neues Konzept mit drei Schritten

2.2.1 SCHRITT 2: VEREINFACHTER RUNDGANG

Der vereinfachte Rundgang soll so erstellt werden, dass er direkt übers Web abgespielt werden kann. Dazu wird die Beschreibungssprache VRML (siehe Kapitel 3.3) verwendet. Mit VRML können nicht alle Funktionen erstellt werden, die mit einer Game-Engine realisiert werden könnten. Aber VRML bietet genügend Möglichkeiten, um den Rundgang so zu erstellen, dass Besucher sich innerhalb des Instituts zurechtfinden können und einen ersten Eindruck vom IKA erhalten.

Der grosse Vorteil gegenüber einer Game-Engine ist – wie bereits erwähnt – die Möglichkeit, den VRML-Rundgang direkt übers Web abzuspielen. Es braucht dazu lediglich ein Plugin für den Browser (siehe Kapitel A 3.2). Ausserdem ist ein VRML-Rundgang deutlich einfacher und schneller zu implementieren. Genügt die Zeit nicht mehr, um mithilfe einer Game-Engine ein ganzes Computerspiel zu erstellen, ist eine VRML-Anwendung deshalb eine zweckmässige Alternative.

2.2.2 SCHRITT 3: RUNDGANG MIT KOMPLETTER FUNKTIONALITÄT

Im Schritt 3 soll schliesslich der ursprünglich geplante Rundgang mit der kompletten Funktionalität einer Game-Engine erstellt werden. Er soll so erstellt werden, wie in Kapitel 2.1.2 beschrieben.

2.3 BESTEHENDE ARBEITEN

2.3.1 REALISIERUNG EINES GEOINFORMATIONSSYSTEMS

Mit der Diplomarbeit „Realisierung eines Geoinformationssystems – Visualisierung und Analysefunktionalität mit einer 3D Engine“ von Michal Beck (Beck, 2002) existiert bereits eine Arbeit, die der vorliegenden Aufgabenstellung sehr ähnlich sieht. Frau Beck benutzte das

Computerspiel „Quake III Arena“ der Firma id Software, um das Modell des Instituts für Photogrammetrie der Universität Stuttgart darzustellen. Auf S. 23 Ihres Berichtes schreibt sie: „Unter den Computerspielen Freaks ist es sehr beliebt, Veränderungen oder Erweiterungen an ihrem Spiel anzubringen. [...]

Um den Wünschen der Benutzer gerecht zu werden, entschied sich id Software für eine Zweiteilung des Programmcodes. Der eine Teil des Programms besteht aus der eigentlichen Engine. Diese ist unter anderem für die Graphikberechnung zuständig und wird streng geheim gehalten. [...] Der andere Teil des Programms beinhaltet die Spiel-Logik und UI (User Interface), also die zugrundeliegenden Regeln für das Verhalten von Spielern, Monstern und Entitäten (Entities). Dieser Teil ist in drei externen Dateien, den DLL-Dateien, [...] ausgelagert. [...] Das besondere an diesen Dateien ist, dass ihr Quellcode freigegeben wurde. Dadurch können alle Anwender Änderungen am Programmcode vornehmen.“

Frau Beck tauschte die drei DLL-Files³ aus, wodurch eine Echtzeitführung durch das Institut realisiert werden konnte. Gemäss [2, S. 18] wurden ausserdem die im Spiel integrierten Waffen-Tools umfunktioniert und für andere Aufgaben benutzt.

2.3.2 MODELLIERUNG DER OSNABRÜCKER ALTSTADT IN SKETCHUP

Im Rahmen eines dreiwöchigen Praktikums modellierten zehn Studenten die Altstadt von Osnabrück. Zur Modellierung und Texturierung verwendeten sie Google SketchUp. Die Texturen erstellten sie aus Fotos, welche sie grösstenteils mit dem Bildbearbeitungsprogramm Adobe Photoshop 6 nachbearbeiteten. Das fertige Modell exportierten sie nach Google Earth und VRML.

In Google Earth ist es nun möglich, die einzelnen Gebäude zu betrachten oder sich eine Tour durch die Altstadt anzusehen.

In VRML fügten die Studenten eine Beleuchtung ein und verlinkten mehrere VRML-Dateien miteinander. Als Resultat entstanden mehrere VRML-Präsentationen, in welchen frei navigiert werden kann.

Als drittes Produkt boten sie einen virtuellen Rundgang in Form von kurzen Flash-Filmsequenzen an. An bestimmten Stellen kann der Betrachter selber wählen, in welche Richtung er weitergehen möchte, indem er auf den entsprechenden Pfeil klickt. Alle drei Präsentationsarten können auf der Homepage betrachtet werden. [3]

2.3.3 VIRTUELLER RUNDGANG DURCH EIN BÜRO

Shiratuddin und Thabet (2002) beschreiben in ihrer Arbeit den Versuch, mithilfe einer Game-Engine eine erschwingliche und realistische dreidimensionale Applikation zu erstellen, welche die Visualisierung und virtuelle Beschreitung eines Büros ermöglicht. Sie wählten dazu die 3D-Game-

³ DLL = Dynamic Link Library. Dies ist eine Codebibliothek, welche jeweils zum Spiel hinzu geladen wird, wenn sie gebraucht wird.

Engine „Unreal Tournament“ aus. Sie importierten dafür den vorhandenen 2D-Grundriss des Büros in 3D Studio VIZ⁴, um dort das 3D-Modell zu erstellen. Anschliessend mussten sie das fertige Modell ins AutoCAD-Format *.dxf exportieren, da der Editor von Unreal Tournament (UnrealEd 2.0) dieses Format unterstützt.

Nach dem Import fügten Shiratuddin und Thabet in UnrealEd 2.0 u. a. Texturen, Audioeffekte und Beleuchtungen hinzu, um den virtuellen Rundgang aufzuwerten.

2.3.4 3D-BENUTZERSCHNITTSTELLE ZUR KONFIGURATION UND VISUALISIERUNG VON DIGITALEN FABRIKEN

Die Diplomarbeit I von Nina Aschenbacher behandelt die „Konzeption und Realisierung einer 3D-Benutzerschnittstelle zur Konfiguration und Visualisierung von Digitalen Fabriken“ [4]. Im ersten Teil führt sie eine Evaluierung verschiedener 3D-Modellierungswerkzeuge (3D Studio MAX, Cinema 4d 8, Movie 3D und Blender) durch. Der zweite Teil ihrer Diplomarbeit „beschäftigt sich mit der Umsetzung der in Teil I beschriebenen Ergebnisse zu einem möglichst intuitiven Editor für virtuelle Fabriken.

Der Fabriкеditor soll dreidimensionale Modelle zu einer Fabrik zusammensetzen können, welche anschliessend abgespeichert und in einer in Java programmierten Laufzeitumgebung wieder geladen werden können“ [4]. Frau Aschenbacher verwendet die jMonkey Engine als Laufzeitumgebung.

⁴ 3D Studio VIZ ist eine schlanke Version von 3D Studio MAX (siehe Kapitel 3.2.1).

3 TECHNISCHE GRUNDLAGEN

3.1 ANIMATIONEN FÜRS WEB

Laut Wikipedia [5] ist Animation „im engeren Sinne jede Technik, mit der einzelbildweise Bewegung im Film geschaffen wird. Die Einzelbilder können gezeichnet sein, im Computer berechnet, oder sie können fotografische Aufnahmen sein. Bei der Wiedergabe mit 24 Bildern pro Sekunde entsteht beim Betrachter die Illusion von Bewegung.“

Die Animation mithilfe von Computern ist somit ein Teilgebiet der verschiedenen Animationsformen. Die Einsatzgebiete von Computeranimationen sind vielfältig. Solche Animationen werden sowohl in Computerprogrammen (wie beispielsweise Computerspielen) als auch in anderen Medien wie Filmen eingesetzt [6]. Die Verwendung von Animationen ist ausserdem auch im Internet verbreitet – zur Animation von Logos, Planeten, Wetter- und Strassenkarten, um nur einige Beispiele zu nennen.

Soll eine Computeranimation übers Web angezeigt werden, muss die Dateigrösse möglichst klein gehalten werden. Ansonsten dauert das Laden zu lange, oder die Animation wird nicht flüssig abgespielt. Zur Erstellung von Web-Animationen muss somit ein Programm verwendet werden, welches kleine Dateien erzeugt. SVG und Flash sind zwei geeignete Programme für diese Aufgabe.

3.1.1 SVG

SVG ist die Abkürzung für „Scalable Vector Graphics“. SVG wird als Standard zur Beschreibung von zweidimensionalen Vektorgrafiken verwendet [7]. Es wurde vom World Wide Web Consortium als Empfehlung herausgegeben. Die meisten Elemente von SVG können von den gängigen Webbrowsern (mit Ausnahme des Internet Explorers) ohne Zusatzprogramme angezeigt werden.

SVG basiert auf XML⁵. Deshalb können SVG-Dateien in jedem Texteditor bearbeitet werden. Mit SVG unterstützt neben Vektor- auch Rastergrafiken und Text. Ausserdem können mit SVG Animationen erstellt werden.

3.1.2 MACROMEDIA FLASH

Macromedia Flash⁶ ist eine proprietäre Entwicklungsumgebung, mit welcher eine Vielzahl unterschiedlicher Projekte erstellt werden kann. Die Möglichkeiten reichen von einfachen Animationen bis zu komplexen Anwendungen, die eine Interaktion mit dem Benutzer erlauben. In Flash-Projekten können Bilder, Videos und Geräusche (Sound) eingebunden werden.

⁵ XML = Extensible Markup Language. Es ist eine Auszeichnungssprache, welche Daten in Form von Textdateien darstellt.

⁶ Macromedia wurde im Jahr 2005 von Adobe übernommen. Die nachfolgenden Flash-Versionen werden deshalb als „Adobe Flash“ bezeichnet.

Ein neues Projekt wird in einer Flash-Dokumentdatei erstellt. Flash-Dokumente liegen im FLA-Format vor. Sie können Dank der Unterstützung von Vektorgrafiken⁷ sehr klein gehalten werden. Deshalb ist Flash besonders gut geeignet, um Projekte fürs Internet zu erstellen.

Die fertig erstellten Flash-Dokumente werden ins SWF-Format kompiliert („veröffentlicht“). Diese webtauglichen, komprimierten SWF-Dateien werden als „Flash-Filme“ bezeichnet.

Flash wird heute auf vielen Webseiten benutzt. Um Flash-Dateien im Webbrowser abspielen zu können, ist ein so genannter Flash-Player erforderlich, welcher inzwischen auf den meisten internetfähigen Rechnern installiert ist.

Der Teil 1 der vorliegenden Arbeit wird mit Macromedia Flash Professional 8 ausgeführt. Für weitere Informationen über die bei der Ausarbeitung der Diplomarbeit gebrauchten Begriffe und Funktionen sei auf Kapitel A 1 im Anhang sowie auf die Hilfe innerhalb der Flash-Umgebung verwiesen.

3.2 3D-GRAFIK-SOFTWARE

Gemässe [8] versteht man unter dem Begriff „3D-Grafik-Software“ Computerprogramme, die es mithilfe der 3D-Computergrafik ermöglichen, dreidimensionale Szenen zu erstellen und/oder daraus ein Bild bzw. eine Computeranimation zu berechnen. Die auf diese Weise erzeugten Szenen können anschliessend beispielsweise für Anwendungen der virtuellen Realität oder für CAD-Anwendungen genutzt werden.

Auf dem Markt existieren unzählige 3D-Grafikprogramme. Drei davon – 3D Studio MAX, Maja und SketchUp – sollen im Folgenden vorgestellt werden.

3.2.1 3D STUDIO MAX

3D Studio Max der Firma Autodesk ist ein 3D-Computergrafik- und Animationsprogramm. Es wird sowohl in den Bereichen Computerspiele, Comic, Animationen und Film als auch in gestalterischen Berufen wie Design oder Architektur eingesetzt [9]. Das Softwarepaket ist sehr umfangreich. Es bietet neben weiteren Funktionen auch Partikelsysteme, Plugins für physikalische Simulationen und Charakteranimationen.

3D Studio MAX wird sowohl von Filmstudios eingesetzt als auch von Computer- und Videospieleentwicklern. So wurde es unter anderem für den Actionfilm „Jurassic Park“ und den Science-Fiction-Film „Planet der Affen“ verwendet.

⁷ Flash kann sowohl Bitmapbilder (Rasterbilder) als auch Vektorgrafiken bearbeiten. Bitmapbilder benötigen jedoch deutlich mehr Speicherplatz als Vektorgrafiken.

3.2.2 MAJA

„Die Software Maya ist eine professionelle und sehr verbreitete 3D-Visualisierungs- und Animationssoftware. Sie wird hauptsächlich in der Film- und Fernsehindustrie und bei der Erstellung von Grafiken für Computer- und Videospiele eingesetzt“ [10]. Weitere Einsatzgebiete sind die Industrielle Fertigung, Architekturvisualisierungen, Entwicklung und Forschung. Maya wird ebenfalls von der Firma Autodesk vertrieben.

3.2.3 GOOGLE SKETCHUP UND SKETCHUP PRO 5

SketchUp ist eine Modellierungssoftware zur Erstellung und Texturierung von 3D-Modellen. Sie ist schnell erlernbar und intuitiv zu bedienen. Besonders geeignet ist sie, um technische Skizzen und Architekturentwürfe zu erstellen.

SketchUp ist in zwei Versionen erhältlich. Das für den privaten Gebrauch kostenlose Google SketchUp ist gemäss [11] „ein einfach zu erlernendes 3D-Modellierprogramm“. Es bietet weniger Funktionen als die kommerzielle Version SketchUp Pro 5. Diese ist „ein leistungsstarkes Werkzeug für das Modellieren in 3D“ [11].

Die Gratisversion bietet alle Funktionen, die für die Modellierung des Instituts für Kartografie erforderlich sind. Jedoch sind die Exportmöglichkeiten sehr beschränkt. Für den Export als VRML-Datei muss deshalb SketchUp Pro 5 verwendet werden, welches als achtstündige Demo-Version gratis heruntergeladen werden kann.

Aus Tabelle 1 im Anhang ist ersichtlich, dass aus SketchUp diverse Formate importiert und exportiert werden können. Dies garantiert eine relativ grosse Weiterverwendbarkeit von SketchUp-Dateien.

Die Dateien von Google SketchUp und SketchUp Pro 5 liegen im Format *.skp vor. Speichert man eine neu erstellte SketchUp-Datei zum ersten Mal, stellt man fest, dass eine zweite Datei mit der Endung *.skb erstellt wird. Dies ist eine Backup-Datei, welche bei jedem Speichern der *.skp-Datei aktualisiert wird.

Das Modell des IKA wird mit Google SketchUp erstellt. In Kapitel A 2 im Anhang werden die wichtigsten Funktionen von Google SketchUp beschrieben, welche für die Erstellung des Modells verwendet werden.

3.3 3D-MODELLIERUNGSSPRACHEN – VRML

Modellierungssprachen (auch Beschreibungs- oder Auszeichnungssprachen genannt) beschreiben die Daten und teilweise auch die Verfahren, welche zur Bearbeiten der Daten nötig sind. „Bei einer Auszeichnungssprache werden die Eigenschaften, Zugehörigkeiten und Verfahren von bestimmten Wörtern, Sätzen und Abschnitten eines Textes beschrieben bzw. zugeteilt“ [12]. Der Übergang zu

Programmiersprachen ist fließend. Die Modellierungssprachen besitzen wie die Programmiersprachen ebenfalls eine Syntax, Semantik und Grammatik.

In diesem Kapitel soll anhand von VRML eine 3D-Modellierungssprache vorgestellt werden.

VRML ist die Abkürzung für Virtual Reality Modeling Language. Es ist eine Modellierungs- oder Beschreibungssprache zur Beschreibung von dreidimensionalen Szenen. Weil viele Programme den Import und Export von VRML-Formaten unterstützen, wird VRML oft auch als Austauschformat von 3D-Modellen verwendet. VRML-Dateien tragen die Endung *.wrl (für „World“) bzw. als komprimierte Variante die Endungen *.wrz und *.wrl.gz.

Mit VRML lassen sich Geometrien und ihre Ausleuchtungen beschreiben sowie Animationen und Interaktionsmöglichkeiten erstellen. Ausserdem lassen sich Scripts (z. B. Java- oder VRML-Scripts) einbinden, wodurch weitere Funktionen programmiert werden können.

VRML ist auf den Einsatz im Internet ausgerichtet [13]. VRML-Dateien werden von den Web-Browsern in Echtzeit dargestellt. Eine Vielzahl von Browser-Plugins stellt dabei sicher, dass die VRML-Dateien auf jeder Plattform betrachtet werden können.

Der VRML-Code liegt in einem Textformat vor. Eine VRML-Datei kann somit direkt in einem Texteditor oder einem syntaxunterstützten Editor (wie VrmIPad) erstellt und bearbeitet werden. Die komfortablere Möglichkeit zur Erstellung von VRML-Code besteht jedoch darin, in einer 3D-Modellierungs-Software mit VRML-Exportfunktion (wie 3D Studio Max, Maya oder SketchUp) das gewünschte Modell zu erstellen und anschliessend als VRML-Datei zu exportieren.

In der Diplomarbeit wird die VRML-Version 2.0 verwendet. In Kapitel A 3 im Anhang wird auf die wichtigsten Grundlagen und Funktionen dieser Modellierungssprache eingegangen. Anschliessend wird am Beispiel des Cortona VRML-Clients die eine Auswahl aktueller kostenloser VRML-Viewer vorgestellt.

3.4 GAME-ENGINES

Dank den Game-Engines muss ein neues Computerspiel nicht von Grund auf neu programmiert werden; die benötigten Bausteine können aus einer Game-Engine entnommen werden. Die Game-Engine stellt somit das Grundgerüst der meisten Computerspiele dar. Da sie unabhängig von einem bestimmten Spiel ist, kann sie für immer wieder neue Spiele verwendet werden.

Die meisten 3D-Computerspiele setzen sich aus einer solchen Game-Engine, den zum Spiel gehörenden Daten sowie den Spielregeln zusammen. Es gibt unzählige Game-Engines, welche oft für

einen bestimmten Typ von Computerspielen (z. B. First-Person-Shooter-Games, Simulationen oder Denkspiele) ausgelegt sind.

Im Gegensatz zu VRML lassen sich in mit einer Game-Engine erstellten Computerspielen auch grosse Modelle integrieren und sind deshalb für virtuelle Präsentationen gut geeignet.

3.4.1 DEFINITION UND AUFBAU

Eine Game-Engine ist eine ganzheitliche Sammlung der für ein Computerspiel benötigten Grundlagen. Sie besteht im Wesentlichen aus einer Programmbibliothek, welche die häufig benutzten Werkzeuge für die Entwicklung von Computerspielen bereitstellt.

Die folgenden (und oft noch weitere) Module sind in den meisten Game-Engines vorhanden (Beck, 2002 und [14]), wobei sie je nach Spieltyp unterschiedlich stark ausgebaut sind:

- Physik-Engine: Sie ist für die plausible Simulation der Naturgesetze (Schwerkraft, Dynamik, Kinematik, Kollisionsbehandlung) zuständig. Bestimmte Algorithmen sorgen dafür, dass sich die Gegenstände und Personen im Spiel physikalisch richtig verhalten. Die Physiksysteme werden in neueren Spielen immer realistischer. Wikipedia [14] schreibt dazu: „Neben höherem visuellen Realismus kann das Physiksystem auch Auswirkung auf das Spielgeschehen haben, wenn beispielsweise ein fallendes Objekt (z.B.: eine geworfene Handgranate) einen Hügel herunterkullert, anstatt darauf liegen zu bleiben.“
- Sound-Engine: Die Sound-Engine ist für die Geräusche eines Spiels (Hintergrundmusik und spielspezifische Geräusche wie Schüsse, Schritte oder Klopfen) zuständig. Heute wird oft 3D-Sound unterstützt, wodurch der räumliche Eindruck verstärkt wird.
- Event-Engine: „Die Event-Engine definiert Ereignisse, die unter bestimmten Bedingungen ausgeführt werden. Kommt z. B. ein Spieler in die Nähe einer Türe, so soll sie automatisch aufgehen.“ (Beck, 2002)
- künstliche Intelligenz: Dank künstlicher Intelligenz verhalten sich virtuelle, vom Computer erzeugte Mitspieler oder Gegner wie wirkliche Menschen. Dies verleiht einem Spiel mehr Realitätsnähe und macht es interessanter. Laut Wikipedia [14] geht künstliche Intelligenz heutzutage neben der klassischen Pfadsuche bereits so weit, dass sich Teams untereinander verständigen, Deckung suchen, Spielfiguren einkreisen und teilweise sogar lernfähig sind und sich an die Spielweise der Spieler anpassen.
- (grafische) Benutzerschnittstelle: Hier wird bestimmt, wie ein Spieler mit dem Programm interagieren kann. Die Steuerung ist ein wichtiger Teil jedes Spieles. Wurden früher vor allem die Pfeiltasten der Tastatur als Eingabegeräte gebraucht, so werden heute – abhängig vom Spieltyp – Tastatur, Maus, Joystick, Gamepad und Lenkrad verwendet.
- Netzwerkunterstützung: Mit ihrer Hilfe können mehrere Spieler gleichzeitig dasselbe Spiel spielen und sind für ihre Mitspieler gegenseitig sichtbar.

- Rendering- oder Grafik-Engine: Sie ist für die grafische Darstellung am Bildschirm zuständig. Heutzutage ist meist eine 3D-Darstellung verlangt, weshalb die Grafik-Engine dann oft auch 3D-Engine genannt wird. Damit eine Visualisierung in Echtzeit möglich ist, muss die Rendering-Engine mind. 15 bis 20 Bilder pro Sekunde bearbeiten können.

Viele dieser Module beinhalten hochentwickelte Algorithmen, welche teilweise für militärische Anwendungen, Wissenschaft, Medizin oder Special Effects in der Filmindustrie entwickelt wurden [15, S. 7].

3.4.2 AUSGEWÄHLTE GAME-ENGINES

Im Internet sind gute Zusammenstellungen aktueller Game-Engines zu finden. Devmaster.net [16] hat eine sehr gute Seite mit 259 Game Engines (Stand 18.04.07). Eine differenzierte Suchmaske erlaubt das schnelle Auffinden von Engines mit den gewünschten Eigenschaften. Auf „The Game Programming Wiki“ [17] sind rund 100 Game Engines vertreten (Stand 20.04.07). Leider gibt es hier keine Suchfunktionen, so dass die passenden Engines von Hand herausgelesen werden müssen. Es gibt jedoch zu jeder Engine eine kurze Beschreibung und Auflistung der wichtigsten Kriterien. Weitere nützliche Listen sind in Wikipedia [18], auf [19] und [20] zu finden. Wie alle diese Zusammenstellungen zeigen, ist heute eine Vielzahl kommerzieller und kostenloser Game-Engines erhältlich. Eine kleine Auswahl davon soll an dieser Stelle kurz vorgestellt werden.

Unreal-Engine

Die Unreal-Engine 1.0 wurde vom US-amerikanischen Spieleentwickler „Epic Games“ erstellt und 1998 auf den Markt gebracht. Seither wurde die Engine in vielen Spielen (u. a. in den Ego-Shootern „Unreal“ und „Unreal Tournament“) verwendet, stetig weiterentwickelt und für zahlreiche Betriebssysteme und Spielkonsolen lauffähig gemacht. Als aktuellste Version wurde im November 2006 die Unreal-Engine 3.0 herausgegeben.

Die Unreal-Engine ist eine der beliebtesten Game-Engines für Actionspiele und gehört zu den wichtigsten 3D-Game-Engines auf dem Markt. Sie wurde in C++ geschrieben.

Irrlicht

Irrlicht ist eine plattformunabhängige 3D-Game-Engine; sie läuft auf Linux, Mac OS und diversen Windows-Versionen. Sie ist in C++ geschrieben. Sie wird vom Österreicher Nikolaus Gebhardt und seinem Team entwickelt.

Irrlicht weist eine hohe Performance auf und unterstützt viele Spezialeffekte wie transparente Objekte und dynamische Schatten.

Irrlicht wurde unter anderem benutzt, um das Weltraum-Strategiespiel „Galactic dream“ und die Bowling-Simulation „Bowling Evolution“ zu erstellen.

Irrlicht wird laufend weiterentwickelt. Im Juni 2007 wurde die neueste Version von Irrlicht, Irrlicht 1.3.1, veröffentlicht.

Quake-Engine

Die Quake-Engine wurde 1996 von des US-Unternehmens „id Software“ entwickelt. Sie ist ein Ego-Shooter und die erste echte 3D-Game-Engine, denn sie benutzt als erste echte 3D-Modelle.

Die Engine wurde in der Sprache C erstellt. Sie läuft auf Windows, Linux und Mac OS.

Mit der Quake-Game-Engine und ihren Weiterentwicklungen wurde beispielsweise die Ego-Shooter-Spielerreihe Quake (Quake, Quake II und Quake III Arena) erstellt. Im Zusammenhang mit dem Spiel „Quake III Arena“, welches 1999 auf den Markt kam, fand die letzte grosse Verbesserung der Quake-Engine statt. Mit Doom 3 ist im Jahr 2004 eine neue Engine erschienen. Sie ist eine Weiterentwicklung der Quake-Engine, wurde aber vollständig neu in C++ programmiert.

Microsoft XNA Framework

XNA ist eine Technologie für die Entwicklung von Computerspielen unter Microsoft Windows und Xbox 360⁸. Sie vereint viele Schnittstellen für die Spiele-Entwicklung unter einem Dach. Dadurch soll ein schneller Entwicklungsprozess ermöglicht werden.

Das Ziel von XNA Framework und der dazugehörigen Entwicklungsumgebung „XNA Game Studio Express“ besteht darin, den Neulingen den Einstieg in die Programmierung von Computerspielen zu erleichtern.

XNA Game Studio Express wurde in auf Visual C# (Express) geschrieben. Die aktuelle Version 1.0 Refresh erschien im April 2007.

⁸ Die Xbox 360 ist eine Videospielekonsole. Sie wurde von Microsoft entwickelt.

4 AUSFÜHRUNG TEIL 1: DIE FLASH-ANIMATION

Die Animation der Weltkugel wird in Macromedia Flash Professional 8 erstellt. In Kapitel 4.1 werden die einzelnen Implementierungsschritte genauer erläutert. Anschliessend werden die verwendeten Datengrundlagen vorgestellt, und in Kapitel 4.3 wird das Endprodukt beschrieben.

4.1 VORGEHEN

Die Implementierung der Flash-Animation setzt sich aus mehreren Teilen zusammen. Im Folgenden werden diese Teile genauer erörtert.

4.1.1 ROTATION

Als Erstes wird die Rotation implementiert. Dazu wird eine optische Täuschung zu Hilfe genommen. Die Idee dieser optischen Täuschung stammt vom Beispiel einer drehenden Weltkugel, welche auf [21] zu finden ist. Es wird eine auf die Ebene projizierte Abbildung der Erde verwendet, welche im Vektorformat vorliegt [6.04 World (Gall Projection), D1]. Nach der Vorbearbeitung in Adobe Illustrator (siehe Kapitel 4.2) wird sie in die Flash-Bühne importiert. Aus dieser Abbildung werden anschliessend zwei Bänder zusammengesetzt: *Welt_vorne* und *Welt_hinten*. Pro Band wird die Abbildung der Erde mehrmals aneinandergereiht, wobei sie für die *Welt_hinten* horizontal gespiegelt wird.

Die beiden Bänder werden anschliessend hintereinander gelegt (Abbildung 3). Mithilfe eines Bewegungs-Tweens werden sie nun aufeinander zu bewegt. Eine Maske sorgt dafür, dass jeweils nur ein kreisförmiger Ausschnitt der Bänder zu sehen ist. Dadurch wird der Eindruck erweckt, es handle sich bei der vorliegenden Animation um eine rotierende Kugel. Effektiv handelt es sich aber lediglich um zwei Translationen.

Das Einfügen der Ebene *Abdeckung* zur Abdunklung der *Welt_hinten* sowie zweier verschieden grosser Kreise (*Kugel* und *Kugel_aussen*) verstärkt den Effekt der Rotation (Abbildung 4).

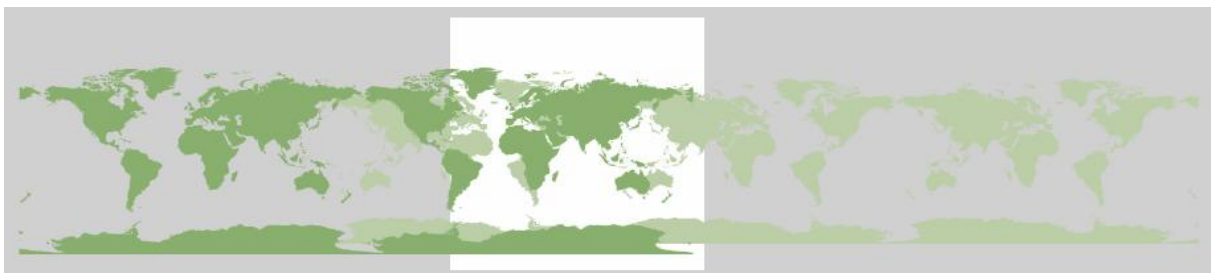


Abbildung 3: Die beiden Welt-Bänder

Das hellere Band (rechts) ist spiegelvekehrt und bewegt sich nach links. Das dunklere Band bewegt sich nach rechts. Das weisse Rechteck ist die Bühne.



Abbildung 4: Die optische Täuschung

Die Endposition der beiden Bänder wird so gewählt, dass nach etwas mehr als einer vollständigen „Rotation“ Europa – horizontal betrachtet – im Zentrum liegt. Dann stoppt die „Rotation“. Jetzt verschwinden *Welt_vorne* und *Welt_hinten*, an ihre Stelle tritt die Ebene *Europa*, welche ebenfalls die Abbildung der Erde enthält – jedoch als Einzel-Abbildung, nicht als Band.

4.1.2 ZOOM TEIL 1: WELT – EUROPA

Das Zoomen stellt den zweiten Teil der Flash-Animation dar. Es wird in zwei Schritten implementiert: Zoom Welt – Europa und Zoom Europa – Schweiz.

Beim ersten Schritt wird die Ebene *Europa* mittels der Funktion „Transformieren“ bis zur Stufe Europa gezoomt. Gleichzeitig wird ihre (zunächst kreisrunde) Maske mit einem Form-Tweening versehen, damit sich ihre Form zu einem Quadrat wandelt.

Die Ebenen *Abdeckung*, *Kugel* und *Kugel_aussen* sollen allmählich ausgeblendet werden. Dazu werden sie mithilfe von Form-Tweens so weit abgedunkelt, bis sie vor dem schwarzen Hintergrund nicht mehr sichtbar sind.

Sobald diese erste Zoomstufe beendet ist, wird mit der Ebene *CH_Grenze* die Schweizer Landesgrenze sichtbar, welche durch ein „CH“ (Ebene *CH*) gekennzeichnet ist (Abbildung 5).



Abbildung 5: Die Bühnensicht nach Ende der 1. Zoomstufe

4.1.3 ZOOM TEIL 2: EUROPA – SCHWEIZ

Als nächster Schritt erfolgt das zweite Zoomen: Das Hineinzoomen bis auf Stufe Schweiz. Auch hier wird die Ebene *Europa* mittels der Funktion „Transformieren“ gezoomt. Ausserdem werden nun allmählich Teile der Schweiz sichtbar. Da sich diese aus mehreren Schichten (Landesgrenze, Gewässernetz, wichtige Städte) zusammensetzen, welche als Vektoren [5.12 Germany, D1] vorliegen und nicht zur selben Zeit eingeblendet werden sollen, werden sie auf mehrere Ebenen aufgeteilt. Die Landesgrenze (Ebene *CH_Grenze*) wird mittels eines Form-Tweens gezoomt. Die Ebene *CH* wird nicht gezoomt, sondern durch ein Bewegungs-Tween lediglich verschoben. Das Gewässernetz (Ebene *Gewaesser*) und die Städte (Ebene *StaedteI*) werden mithilfe der Funktion „Transformieren“ gezoomt. Mit einem Trick wird erreicht, dass diese beiden Ebenen erst nach einer Weile eingeblendet werden: Zu beiden Ebenen wird je eine Maske erstellt, die sich ausserhalb des sichtbaren Bereiches befindet. Solange die Maske existiert, wird somit nichts angezeigt. Erst zu dem Zeitpunkt, wenn die Maske wegfällt, werden die Ebenen sichtbar.

Als nächstes soll der Standort der ETH Hönnggerberg angezeigt werden. Dazu wird am Schluss dieser Zoomstufe mit einem Pfeil (Ebene *Pfeil*) und dem Text „ETH Hönnggerberg“ (Ebene *ETH_Schrift*) der Standort der ETH Hönnggerberg innerhalb der Schweiz grob markiert (Abbildung 6).

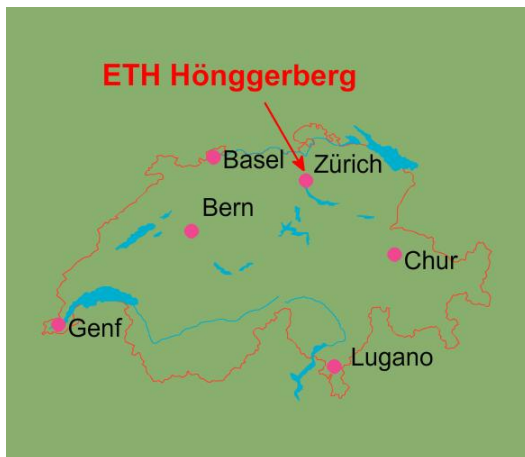


Abbildung 6: Die Schweiz nach dem Ende der zweiten Zoomstufe

Als Alternative zum zweiten Zoomen wird ein Zoomen der Schweiz als Rasterbild getestet. Das Ergebnis ist jedoch nicht zufrieden stellend (zu schlechte Auflösung und zu grosse Strichdicken bei den höheren Zoomstufen).

4.1.4 ÜBERBLENDUNG ZUR ETH HÖNGGERBERG

Nun soll eine Überblendung zur Abbildung der ETH Hönggerberg erfolgen. Dazu wird die Schweiz ausgeblendet, indem eine Ebene *Ausblenden* erstellt wird. Die Transparenz dieser Ebene (zu Beginn bei 0%) wird mittels eines Form-Tweens allmählich auf 100% erhöht, so dass die darunter liegenden Ebenen der Schweiz „verschwinden“. Der Pfeil wird ebenfalls ausgeblendet, und die Bezeichnung „ETH Hönggerberg“ der Ebene *ETH_Schrift* wird mittels eines Bewegungs-Tweens an seine endgültige Position verschoben.

Während die Schweiz ausgeblendet wird, wird die Abbildung der ETH Hönggerberg (Ebene *ETH*) eingeblendet – dies mithilfe der Funktion „Übergang“. Anschliessend wird innerhalb der ETH Hönggerberg der Standort des Instituts für Kartografie mittels der Ebene *IKA* markiert.

4.1.5 TICKER

Während der ganzen Animation läuft am unteren Rand ein Ticker von rechts nach links. Darin ist der Text „Rundgang durch das Institut für Kartografie“ zu lesen. Der Ticker wird mit einem Bewegungs-Tween erstellt.

Abbildung 7 zeigt den Endzustand nach Ablauf der Animation.



Abbildung 7: Zustand am Ende der Animation

4.2 VERWENDETE DATEN

Für die Erstellung der Flash-Animation werden ausschliesslich Vektordaten verwendet. Die Gründe liegen darin, dass die Resultate schöner aussehen als mit Rasterbildern und die Datei durch konsequente Verwendung von Vektordaten kleiner gehalten werden kann.

Für die Welt-Bänder sowie das Zoomen wird die Karte 6.04 World (Gall Projection) von Digital Wisdom, Inc. [D1] verwendet. Diese Karte liegt als Datei „604GAL3I.AI“ vor und wird in Adobe Illustrator vorbearbeitet. Alle Ebenen ausser der Ebene *Land fill* werden gelöscht. In dieser Ebene werden die kleinsten Inseln gelöscht. Anschliessend werden zwei Kopien erstellt: „Welt_vorne.ai“ und „Welt_hinten.ai“. Die Ebene *Land fill* wird in diesen beiden Dateien unterschiedlich eingefärbt, wobei sie in der Datei „Welt_hinten.ai“ blasser dargestellt wird.

Nach dem Import in Flash wird schliesslich entschieden, auch alle Seen, welche als Löcher in der Ebene *Land fill* ausgespart sind, aufzufüllen. Nur noch die Binnenmeere werden als Löcher beibehalten.

Die einzelnen Ebenen der Schweiz stammen aus der Karte 5.12 Germany (Datei „512GER3I.AI“), ebenfalls von Digital Wisdom, Inc. [D1]. Auch diese Datei wird in Adobe Illustrator vorbereitet. Es werden die Ebenen *City/town locations*, *City/town names*, *Rivers*, *Ocean/lakes* und *Borders* verwendet. Bei der Ebene *Ocean/lakes* werden kleine Seen gelöscht. Alles, was ausserhalb der Schweizer Grenze liegt, wird ausserdem weggelöscht. Lediglich die Seen, welche über die Grenze hinausragen, werden ganz beibehalten.

Generell wird die Farbgebung und Strichdicke geändert.

Die Vektordaten der ETH Hönggerberg entstammen der Website des IKA [D2]. Von dieser Datei „ETH-Hoenggerberg_color.ai“ werden die Ebenen *Innenhof*, *allg. Gebäude*, *ETH-Gebäude*, *Wasser*, *Wald*, *Wiese* und *Rahmen/Strassenfüllung* verwendet. Sie werden in eine neue Datei („ETH-Hoenggerbert_aktuell.ai“) gespeichert. Die Farbgebung wird in Adobe Illustrator nicht verändert. Erst nach dem Import in Flash wird sie leicht angepasst.

4.3 ERGEBNIS

Das Ergebnis dieses 1. Schrittes auf dem Weg zu einer virtuellen Präsentation des IKA besteht aus einem 18 Sekunden langen Flash-Film. Wie er aussieht, ist im Wesentlichen bereits in Kapitel 4.1 erklärt und soll hier nochmals zusammengefasst werden.

Der Film beginnt mit der Weltkugel. Sie dreht sich um ihre Achse. Nach etwas mehr als einer vollständigen Umdrehung liegt Europa auf der dem Betrachter zugewandten Seite. Die Rotation stoppt.

Nun wird allmählich hineingezoomt. Dabei füllt nach einer Weile Europa den gesamten Bereich aus. Das Zoomen setzt sich jedoch bis zur Stufe Schweiz fort. Die Schweizer Grenze sowie die wichtigsten Teile des Gewässernetzes und ein paar zur Orientierung wichtige Städte sind nun sichtbar.

Es erscheint ein Pfeil, der zur Stadt Zürich zeigt und mit „ETH Hönggerberg“ beschriftet ist.

Nun verschwindet dieses Bild der Schweiz, und an seine Stelle tritt ein Situationsplan der ETH Hönggerberg. Ein neuer Pfeil wird eingeblendet, welcher zum IKA zeigt.

Während der gesamten Rotation läuft am unteren Bildrand ein Ticker mit dem Text „Rundgang durch das Institut für Kartografie“ durch.

Der Flash-Film kann im Internet unter <http://support.cartography.ch/> betrachtet werden.

5 AUSFÜHRUNG TEIL 2: DER VIRTUELLE RUNDGANG

In den folgenden Kapiteln wird beschrieben, wie der virtuelle Rundgang erstellt wird. Dieser virtuelle Rundgang stellt den Hauptteil der Diplomarbeit dar – sowohl was die dafür gebrauchte Zeit betrifft als auch im Hinblick auf die Erfüllung des Ziels, das IKA virtuell vorzustellen.

5.1 VORGEHEN

Zuerst wird ein möglichst vollständiges 3D-Modell des Instituts für Kartografie erstellt. Die Erstellung dieses Modells wird in Kapitel 5.1.1 erörtert. Wie Tests zeigen, ist dieses Modell jedoch zu gross, um als VRML-Datei in einem Browser befriedigend dargestellt zu werden. Das Laden dauert zu lange, und die Navigation ist viel zu langsam. Deshalb wird das Modell vereinfacht und verkleinert (Kapitel 5.1.2). Anschliessend wird dieses vereinfachte Modell als VRML-Datei exportiert. Schliesslich werden die gewünschten Funktionen für den Rundgang in VRML implementiert (Kapitel 5.1.4).

5.1.1 ERSTELLUNG DES 3D-MODELLS

Das dreidimensionale Modell des IKA soll möglichst vollständig sein. Es ist jedoch niemals möglich, die Wirklichkeit vollständig abzubilden. Ausserdem stellt sich rasch heraus, dass die Modellierung viel Zeit in Anspruch nehmen wird. Deshalb wird das Schwergewicht der Modellierung auf die Flure des Instituts gelegt, oder besser gesagt auf den Raum, der ausserhalb der Büros und anderen Räumen öffentlich zugänglich ist. Für das erstrebte Ziel (die Präsentation des IKA) genügt es, wenn die einzelnen Zimmer von aussen betrachtet werden können. Es ist nicht zwingend erforderlich, dass sie auch betreten werden können.

Die Flure werden möglichst realitätsnah und genau modelliert; die Räume dagegen werden unmöbliert dargestellt und nur rudimentär texturiert. Als Software für die Modellierung und Texturierung wird Google SketchUp verwendet.

Modell

Als Ausgangspunkt für die Erstellung des Modells liegen die Grundrisse des D- und G-Stockes des HIL vor. Sie können in Google SketchUp importiert werden. Die Raumhöhe sowie alle weiteren Masse werden mit einem Massband ermittelt. Da das Modell nicht hochgenau werden muss, wird je nach Situation auf 1 - 5 cm genau gemessen. Zur Konstruktion des Modells werden anschliessend die im Anhang in Kapitel A 2 beschriebenen Verfahren verwendet. Dabei wird darauf geachtet, dass ein späteres Bearbeiten eines oder mehrerer Elemente möglichst einfach ist. Dies wird durch häufige Verwendung von Komponenten erreicht. So wird beispielsweise die Komponente *Schrank* erstellt, welche für sämtliche braunen Einzelschränke verwendet werden kann. Die einzelnen Instanzen dieser Komponente werden so rotiert, dass sie an die jeweilige Wand passen. Die Doppelschränke wiederum

werden aus zwei Einzelschränken und einem Zwischenteil (Komponente *Schrank_Zwischenwand*) zusammengesetzt und als neue Komponente *Schrank_gross* abgelegt. Sollen nun alle Instanzen der Komponente *Schrank* bearbeitet werden, muss dies nur einmal an einer Instanz geschehen. Alle anderen Einzel- und Doppelschränke werden automatisch angepasst.

Dasselbe Vorgehen wird auch für die braunen und weissen Türen verwendet, wobei hier wegen unterschiedlicher Breiten jeweils mehrere Komponenten angelegt werden müssen.

Bei der Erstellung des Modells stellt sich heraus, dass der verwendete Grundriss des G-Stockes nicht mit der Realität übereinstimmt. Mehrere Türöffnungen sind an einem anderen Ort eingezeichnet, als sie sich in Wirklichkeit befinden. Die Abweichungen betragen zum Teil über einen halben Meter, dürfen also nicht vernachlässigt werden.

Um das Modell näher an die realen Verhältnisse anzupassen, müssen mehrere Türen in SketchUp verschoben werden. Da die Fehler im Grundriss jedoch erst nach fortgeschrittener Modellierung entdeckt werden, wird darauf verzichtet, sämtliche Ungenauigkeiten zu beheben, was einen grossen Mehraufwand bedeutet hätte. Es werden lediglich diejenigen Türöffnungen verschoben, welche stark von ihrer wirklichen Position abweichen.

Die Entdeckung dieses Fehlers im Grundriss zeigt einmal mehr, dass grundsätzlich alles hinterfragt werden soll. Es darf nicht einfach davon ausgegangen werden, dass die erhaltenen Daten korrekt und auf dem neusten Stand sind.

Texturierung

Um ein realistisch aussehendes Abbild der Wirklichkeit zu erstellen, sind Texturen von grosser Wichtigkeit. Eine gute Texturierung lässt ein Modell so gut aussehen, wie wenn es mit viel mehr Polygonen modelliert worden wäre. Die Reduzierung der Anzahl Polygone erhöht ausserdem die Performance.

Der Zeitbedarf für die Texturierung darf aber nicht unterschätzt werden. Er ist um ein Vielfaches grösser als der Aufwand, der für das blosse Erstellen des Modells benötigt wird. Die Erstellung, Auswahl und Bearbeitung der Fotografien und Grafiken beansprucht dabei mit Abstand die meiste Zeit, während das eigentliche Texturieren deutlich schneller erledigt werden kann.

Für die Texturierung des Modells des IKA werden drei grundsätzlich verschiedene Texturtypen verwendet. Dies sind einerseits Farbtöne, welche direkt in SketchUp erstellt werden können. Unter anderem werden die glatten Wände und sämtliche Pfeile auf diese Weise texturiert. Zum zweiten sind es flächenhafte Texturen, die sich innerhalb der texturierten Ebene mehrmals wiederholen (so genannte Kachelung; siehe Abbildung 8). Beispiele dafür sind die Texturen für die Böden und Decken. Schliesslich werden aber vor allem objektbezogene Texturen verwendet, welche nur einmal pro texturierte Fläche vorkommen. Sie müssen auf der Fläche so platziert werden, dass sie sie ganz bedecken. Hier können Türen, Schrank-Frontseiten und Poster als Beispiele aufgeführt werden.

In SketchUp sind bereits standardmässig einige Texturen („Materialien“ genannt, vgl. Kapitel A 2.7) vorhanden, welche verwendet werden können. Eigene Texturen können auf verschiedene Arten erstellt werden. Farbtöne können – wie bereits erwähnt – direkt in SketchUp erstellt werden. Einfache Grafiken lassen sich beispielsweise mit dem Zeichenprogramm Paint erstellen. Die in dieser Diplomarbeit weitaus am häufigsten verwendete Methode, um Texturen zu bekommen, ist jedoch das Abfotografieren der realen Objekte. Tabelle 2 im Anhang gibt Aufschluss über die Erstellungsart der einzelnen Materialien.

Für die Aufnahme der Fotografien wird die Digitalkamera Sony Cyber-shot DSC-W30 verwendet. Die Fotos liegen anschliessend im JPEG-Format vor.

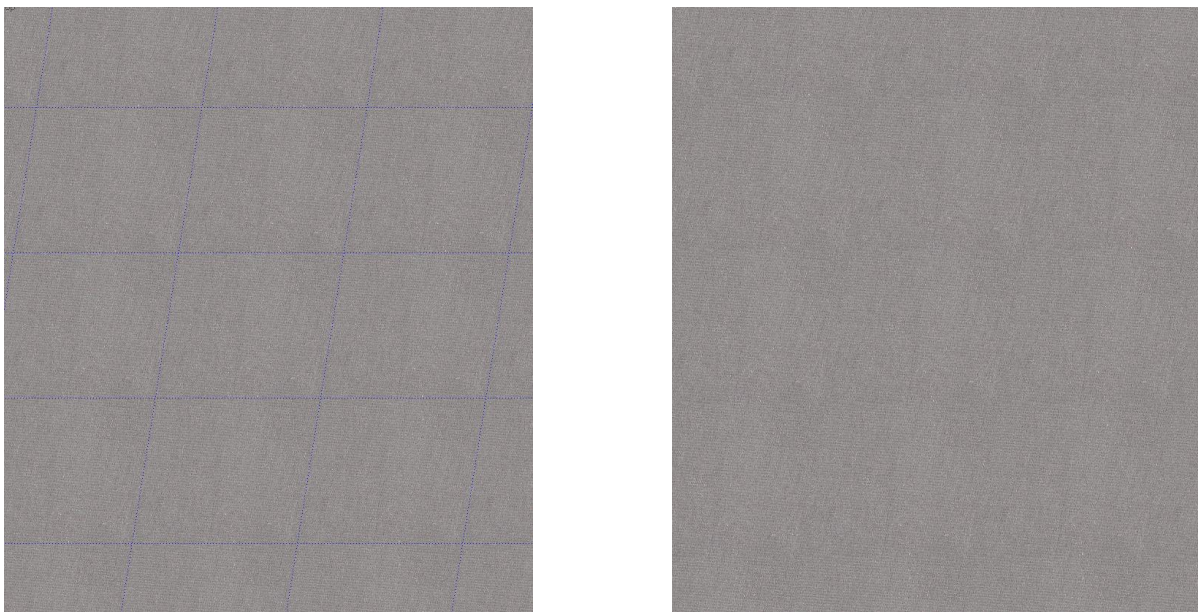


Abbildung 8: Kachelung des Bodens

Links: Boden mit eingezeichneten Kachelgrenzen. Rechts: derselbe Ausschnitt ohne Kachelgrenzen.

Michal Beck erwähnt in ihrer Diplomarbeit (Beck, 2002, S. 54), was bei der Aufnahme von Fotos beachtet werden muss, wenn sie zum Texturieren eines Modells gebraucht werden sollen. Sie empfiehlt, spiegelnde Oberflächen (wie in Glas gerahmte Bilder, zu welchen die im IKA aufgehängten Poster zählen) ohne Blitz zu fotografieren. Durch die Reflexion des Blitzes an der Glasoberfläche sei ein heller Fleck auf dem Bild zu sehen. Diese Beobachtung kann nach der Aufnahme der IKA-Poster sowie weiterer spiegelnder Objekte bestätigt werden. Jedoch sind die Flure des IKA nur schwach ausgeleuchtet. Das Fotografieren der dort hängenden Poster ohne Blitz führt somit zu ungeeigneten, dunklen und unscharfen Bildern. Auch eine Aufhellung oder anderweitige Bearbeitung der Aufnahme in einem Bildbearbeitungsprogramm (z. B. das was ich verwendet habe) vermag das Foto kaum auf die Qualität eines mit mehr Licht aufgenommenen Bildes zu setzen. Als Abhilfe könnte das (aufwändige) Aufnehmen mittels Stativ und langen Belichtungszeiten dienen. In der vorliegenden Arbeit wurden jedoch der Blitz verwendet und die Fotos nicht direkt von vorne aufgenommen. Durch das Aufnehmen schräg von der Seite können die Reflexionen eliminiert oder zumindest auf ein

akzeptables Minimum reduziert werden, so dass befriedigende Resultate entstehen. Darüber hinaus überdeckt das Blitzlicht allfällige Spiegelungen auf den Postern. Wenn sich zum Beispiel eine eingeschaltete Deckenlampe auf dem Poster spiegelt, ist sie auf einer mit Blitz getätigten Aufnahme nicht sichtbar. Würde man statt des Blitzes ein Stativ verwenden wollen, welches lange Belichtungszeiten ohne Verwackeln ermöglicht, müsste man die störenden Deckenbeleuchtungen entweder selektiv ausschalten oder zudecken. Auch ein Ausleuchten mittels spezieller Scheinwerfer käme in Frage. Dies sind aber unverhältnismässig aufwändige Methoden.

Neben den Postern werden auch die meisten anderen Elemente in den Fluren mit Blitz aufgenommen.

Bei der Aufnahme von Texturen, welche gekachelt werden sollen, muss darauf geachtet werden, dass ein homogenes Foto entsteht. Praktischerweise geht man folgendermassen vor: Man erstellt das Foto und texturiert damit die gewünschte Fläche. Beim Betrachten dieser Fläche stellt man sofort fest, ob die Textur homogen ist oder ein Kacheleffekt (Abbildung 9) auftritt. Falls ein solcher Effekt auftritt, wird das Foto in einem Bildbearbeitungsprogramm solange zurechtgeschnitten, bis eine homogene Fläche entsteht und die Kachelung auf der texturierten Fläche nicht mehr ersichtlich ist.

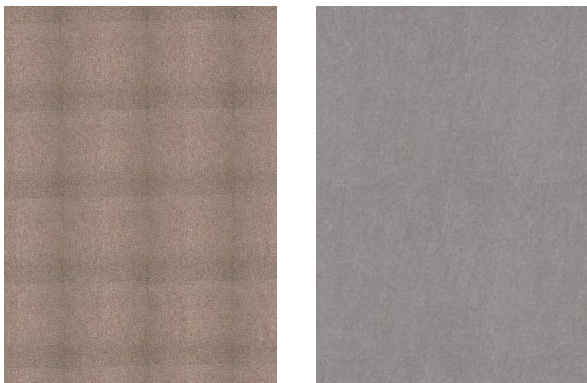


Abbildung 9: Kacheleffekt

Links: Boden mit schlechter Textur (Kacheleffekt); rechts: Boden ohne Kacheleffekt.

Für eine Texturierung in SketchUp müssen die aufgenommenen Fotos nicht entzerrt werden. Schräg aufgenommene Bilder können problemlos so gezerrt werden, dass sie auf die zu texturierende Oberfläche passen. Trotzdem ist eine orthogonale Aufnahme wünschenswert. Gerade bei Objekten wie Türen sieht die Textur schöner aus, wenn die Türfalle orthogonal von vorne aufgenommen wurde. Jedoch ist es im IKA wegen der oben erwähnten Reflexionen bei Aufnahmen mit Blitz sowie wegen der teilweise engen Platzverhältnisse (Winkel des Kameraobjektivs zu klein) oft nicht möglich, eine orthogonale Aufnahme zu erstellen.

Zeitaufwand

SketchUp lässt sich sehr einfach und intuitiv bedienen. Der Modellierungsaufwand darf aber trotzdem nicht unterschätzt werden. Ein Grossteil der gesamten Ausarbeitungszeit der Diplomarbeit wird für die Erstellung und Texturierung des Modells benötigt.

Bei der reinen Modellierung (ohne Texturierung) ist die Menge das zeitlich relevante Element. Einzelne Flächen oder primitive Körper können sehr schnell erstellt werden. Müssen jedoch – wie im vorliegenden Fall – hunderte Einzelteile modelliert werden, summiert sich die Zeit rasch auf.

Noch mehr Zeit als die Modellierung nimmt aber die Texturierung in Anspruch. Die Erstellung, Auswahl und Bearbeitung der Fotos und anderen Grafiken ist sehr zeitaufwändig. Da der reale Zustand des IKA abgebildet werden soll, kann jedoch nicht auf dieses Erfassen der wirklichen Texturen verzichtet und stattdessen vordefinierte Materialien verwendet werden.

Sind die Texturen einmal vorbereitet, dauert das eigentliche Texturieren in SketchUp nicht mehr lange.

Das fertige Modell wird schliesslich als „Modell_IKA.skp“ abgespeichert.

5.1.2 VEREINFACHUNG DES MODELLS

Damit das Modell problemlos im Web angezeigt werden kann, muss die Dateigrösse reduziert werden.

Nach Fertigstellung des Modells „Modell_IKA.skp“ wird deshalb unter dem Namen „Modell_IKA_klein.skp“ eine Kopie der Datei erstellt. Diese wird – ebenfalls in Google SketchUp – vereinfacht und verkleinert.

Als Erstes werden sämtliche Räume aus der Datei gelöscht. Für eine Orientierung genügt es, wenn die einzelnen Zimmertüren von aussen betrachtet werden können. Das Betreten der Büros und anderen Räume ist nicht zwingend erforderlich. Ausserdem sind die Zimmer sowieso nur rudimentär texturiert und nicht möbliert; ein Betreten wäre also ohnehin nicht sinnvoll.

Wo möglich, werden die Geometrien vereinfacht. So wird das Dach, welches ursprünglich aus mehreren Einzelteilen besteht, zu einem Ganzen zusammengefügt. Dadurch muss die Datei nun nur die äussersten Eckpunkte dieses einen Polygons speichern, und nicht mehr auch die Begrenzungspunkte der einzelnen Teilstücke.

Eine weitere Möglichkeit, um das Modell kleiner zu machen, ist das Ersetzen komplexer Geometrien durch Fotos.

Und schliesslich bewirkt die Verkleinerung der Textur-Fotos eine starke Reduzierung der Dateigrösse. Allerdings tritt hier der Nachteil auf, dass die Texturen nicht mehr so scharf sind, was bei näherem Heranzoomen störend wirken kann.

5.1.3 NEUES GESTALTUNGSKONZEPT

Das fertig vereinfachte Modell „Modell_IKA_klein.skp“ wird ebenfalls kopiert. Die Kopie („Modell_IKA_Konzept.skp“) wird wiederum abgeändert, um das neue Gestaltungskonzept des IKA zu modellieren. Dieses Konzept wurde von Juliane Cron, René Sieber und Elisabeth Kuth entwickelt. Es sieht folgende Änderungen gegenüber dem heutigen Zustand vor:

- Institutsfarben: Grün und Silber/Alu
- Panorama auf Schränken
- Namensschilder in Grün
- Empfangsbereich u. a. mit historischer Ecke (Bietschhorn-Relief etc.) am heutigen Standort der Sitzgruppe, Verkleidung der beiden Säulen und Meeting Point
- „Werkstatt“ auf Flur-Innenseite (Ausstellung von Arbeiten, Projekten, Diplomarbeiten etc. auf grün-weissen Metalltafeln)
- magnetische Anschlagtafel(n)
- Studi-Lounge am heutigen Standort des Bietschhorn-Reliefs

Das neue Gestaltungskonzept wird so gut wie möglich umgesetzt:

- Die Textur der Schränke wird mit dem Panorama ergänzt.
- Die Türschilder werden ausgetauscht.
- Die Sitzgruppe und das Bietschhorn-Relief tauschen ihren Platz.
- Die beiden Säulen werden verkleidet.
- Für die „Werkstatt“ wird die Komponente „Poster“ erstellt. Deren Instanzen werden an den Wänden der Flur-Innenseite platziert.

5.1.4 IMPLEMENTIERUNG DES RUNDGANGES

Die beiden kleineren Modelle („Modell_IKA_klein.skp“ und „Modell_IKA_Konzept.skp“) werden mithilfe von SketchUp Pro 5 als VRML-Dateien („rundgang.wrl“ und „konzept.wrl“) exportiert. Die Datei „rundgang.wrl“ wird danach in einem einfachen Texteditor weiterbearbeitet. Da die beiden Modelle in Bezug auf die implementierten Funktionen nur wenige Unterschiede aufweisen, können die neu erstellten Code-Fragmente anschliessend mit wenigen Änderungen auch in den Code von „konzept.wrl“ integriert werden.

Um das Übernehmen der selber erstellten Code-Teile in die Datei „konzept.wrl“ sowie in neu exportierte Versionen von „rundgang.wrl“ und „konzept.wrl“ zu vereinfachen, werden diese Teile an den Anfang bzw. ans Ende der Export-Datei gestellt. Die Struktur des fertig bearbeiteten VRML-Codes sieht somit wie in Abbildung 10 aus. Die einzelner selber erstellten Fragmente werden mit den Zeichen ##### gekennzeichnet.

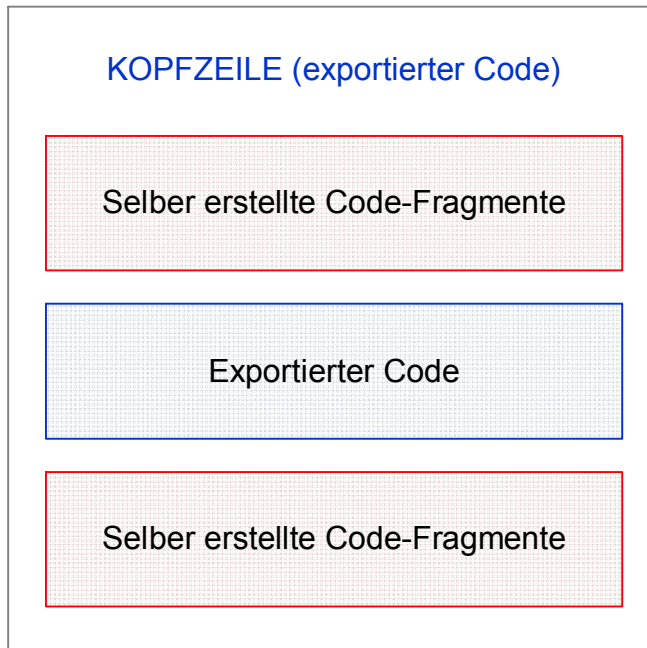


Abbildung 10: Struktur des VRML-Codes

Im Folgenden werden die Funktionen beschrieben, welche in die Abschnitte 2 und 4 des VRML-Codes (siehe Abbildung 10) eingefügt werden. Ausserdem werden die Schwierigkeiten beschrieben, welche beim Umgang mit dem Koordinatensystem von VRML auftreten.

Für nähere Beschreibungen der einzelnen Begriffe siehe Kapitel A 3 im Anhang.

Animierte Wege zu den Büros

Um die Wege zu den Büros automatisch beschreiten zu können, müssen die entsprechenden Viewpoints animiert werden. Dies geschieht mittels der beiden Knoten „OrientationInterpolator“ und „PositionInterpolator“. Sie besitzen vordefinierte Orientierungen bzw. Positionen, welche bei Aktivierung des Viewpoints laufend an diesen übergeben werden. Durch Interpolationen entsteht eine kontinuierliche Animation.

Sobald der Benutzer im Browser-Menü auf einen bestimmten Viewpoint klickt, wird dessen Knoten aktiviert und dadurch die entsprechende Animation gestartet.

Zoom zu den Türschildern

Die Türschilder werden mit von transparenten Quadern überdeckt. Diese Quader dienen als Angriffsflächen für die Touch-Sensoren. Bei einem Mausklick auf ein Türschild wird in Wirklichkeit nicht dieses Türschild, sondern der darüber liegende Quader berührt. Durch diese Berührung wird der entsprechende Touch-Sensor aktiviert. Ein VmlScript erkennt dies und aktiviert seinerseits den Viewpoint des betreffenden Türschildes. Durch den Wechsel der aktuellen Ansicht hin zur Ansicht des Türschild-Viewpoints entsteht der Eindruck des Zoomens.

Links zu Websites

Bei gewissen Büros werden Hyperlinks zu bestimmten Websites angebracht. Es handelt sich dabei um die Websites der Projekte, die im entsprechenden Büro bearbeitet werden.

Zur Erstellung der Hyperlinks wird der Anchor-Knoten verwendet. Die in diesem Knoten vorhandenen Objekte reagieren – ähnlich wie bei einem Touch-Sensor – auf Mausklicks.

Für die Aktivierung der Hyperlinks würde es grundsätzlich genügen, lediglich einen Text in den Anchor-Knoten einzufügen. Versuche zeigen jedoch, dass die Flächen zwischen den einzelnen Buchstaben nicht auf Mausklicks reagieren – nur die Buchstaben selber öffnen den Link. Um diesen Mangel zu beheben, wird in jeden Anchor-Knoten zusätzlich ein transparenter Quader eingefügt, welcher in der Folge ebenfalls auf Mausklicks reagiert.

Poster als Pdf

Die in den Fluren hängenden Poster können wegen ihrer Grösse und Auflösung direkt im VRML-Rundgang nicht gelesen werden. Mittels eines Hyperlinks (siehe oben) wird deshalb die Möglichkeit geboten, die einzelnen Poster als Pdf zu öffnen.

Zur Erstellung dieser Hyperlinks wird dasselbe Verfahren verwendet wie bei den Hyperlinks zu den Websites.

Weitere Hyperlinks

Damit unerfahrene Benutzer sich schnell mit den Funktionen des VRML-Viewers zurechtfinden, wird ein Link zu einer Anleitungs-Seite erstellt. Auf dieser Seite sind die wichtigsten Funktionen des Viewers sowie die vom Rundgang gebotenen Funktionen erklärt.

Ein weiterer Link wird erstellt, um die Fotos der Etagen-Informationentafeln anzuzeigen, welche sich bei den Fahrstühlen befinden.

In der Datei „rundgang.wrl“ wird ausserdem ein Link eingefügt, welcher den VRML-Rundgang mit dem neuen Farbkonzept öffnet. Ein weiterer Hyperlink in der Datei „rundgang.wrl“ führt von der Etage D (dem Startpunkt beim Öffnen der Datei) zum IKA in der Etage G.

Beleuchtung

Mittels des Knotens „PointLight“ werden mehrere punktförmige Lichtquellen in den Code eingefügt. Sie sorgen dafür, dass die Szene gut ausgeleuchtet ist.

Kollisionserkennung

Der Collision-Knoten sorgt dafür, dass nicht durch Wände oder andere Objekte hindurch gegangen werden kann. Er wird um den Export-Teil des Codes gelegt, d. h. sämtliche exportierten Elemente werden durch ihn als undurchdringbare Widerstände erfasst.

Schwierigkeiten mit dem Koordinatensystem von VRML

Beim Export aus SketchUp werden die Koordinaten nicht unverändert übernommen. Neben der Verkleinerung um den Faktor 2.54 (vgl. Kapitel A 3.1 im Anhang) findet ausserdem eine Translation und Rotation statt.

Aus diesem Grund wird die Excel-Datei „Koordinaten.xls“ erstellt, welche diese Skalierung, Translation und Rotation vornimmt. Die Koordinaten der in VRML zu platzierenden Quader, Texte und Lichtquellen können somit aus SketchUp herausgelesen und in die Excel-Datei eingegeben werden. Die von Excel transformierten Werte können anschliessend in den VRML-Code eingetragen werden. Gegebenenfalls müssen nach einer Sichtung der VRML-Datei im Browser noch kleine Anpassungen an den Koordinaten vorgenommen werden.

Damit später nachvollzogen werden kann, welche Werte aus SketchUp herausgelesen worden sind, werden diese in die SketchUp-Datei „Pfade.skp“ eingezeichnet.

5.1.5 RUNDGANG MIT KOMPLETTER FUNKTIONALITÄT

Diese im neuen Konzept als Schritt 3 vorgesehene Arbeit – die Erstellung des Rundganges in einer Game-Engine – kann aus Zeitgründen leider nicht mehr durchgeführt werden. Erste Versuche mit Irrlicht und XNA Framework zeigen jedoch, dass das grosse Modell für eine Weiterbearbeitung mit einer Game-Engine geeignet ist. Es steht somit für allfällige spätere (Studenten-)Arbeiten bereit.

Die Arbeit in einer Game-Engine ist jedoch viel komplexer und schwieriger als in VRML. In einer Game-Engine müssen die Animationen und Bewegungen, die Kollisionserkennung sowie viele weitere Funktionen manuell programmiert werden.

5.2 VERWENDETE DATEN

Es stehen nur wenige Daten zur Verfügung, welche verwendet werden können. Lediglich die Grundrisse des D- und G-Stockes können benutzt werden.

Der Grundriss des D-Stockes wird in zwei Dateien geliefert: „HIL_D_20070316.dwg“ und „HIL_D_20070316.dxf“ [D3]. Die Datei „HIL_D.ai“ wird anschliessend daraus abgeleitet.

Der Grundriss des G-Stockes wird ebenfalls als DWG- und DXF-Datei bereit gestellt: „HIL_G.dwg“ und „HIL_G.dxf“ [D3]. Er wird ebenfalls zusätzlich in eine Illustrator-Datei („HIL_G.ai“) konvertiert.

5.3 ERGEBNISSE

Das Ergebnis dieses 2. Schrittes besteht aus den beiden Dateien „rundgang.wrl“ und „konzept.wrl“.

Wird der Rundgang gestartet, öffnet sich die Datei „rundgang.wrl“. Beim Start befindet man sich in der Eingangshalle des HIL, mit Blick zu den Fahrstühlen (Abbildung 11). Durch einen Mausklick auf den Link „Info“ öffnet sich das Foto der Etagen-Informationstafel, so dass die Informationen über die einzelnen Stockwerke gelesen werden können.

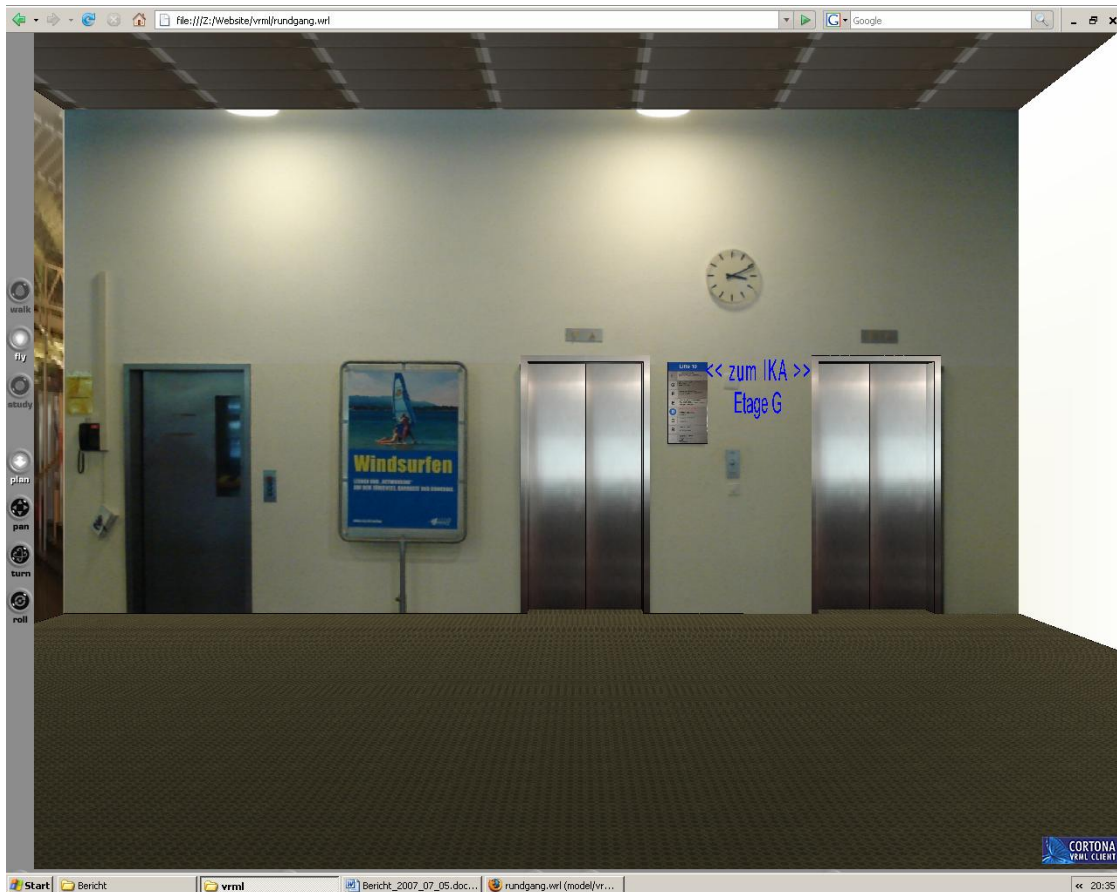


Abbildung 11: Die Start-Ansicht des Rundgangs

Beim Klicken auf den Hyperlink „zum IKA“ gelangt man zu den Räumlichkeiten des Instituts für Kartografie in der Etage G. Die Ansicht ist wiederum auf die Fahrstühle gerichtet. Auch hier öffnet sich mit dem Link „Info“ das Foto der Etagen-Informationstafel.

Klickt man auf „Anleitung“, wird ein Dokument geöffnet, welches die wichtigsten Funktionen des VRML-Viewers sowie die vom Rundgang gebotenen Funktionen erklärt.

Und schliesslich wird beim Klick auf den Link „Neues Gestaltungskonzept“ die Datei „konzept.wrl“ geöffnet, welche das IKA im neuen Kleid präsentiert (Abbildung 12). Die Start-Ansicht dieses zweiten Rundganges befindet sich direkt in der Etage G, ebenfalls mit Sicht zu den Fahrstühlen. Die beiden Links „Info“ und „Anleitung“ sind hier ebenfalls verfügbar.

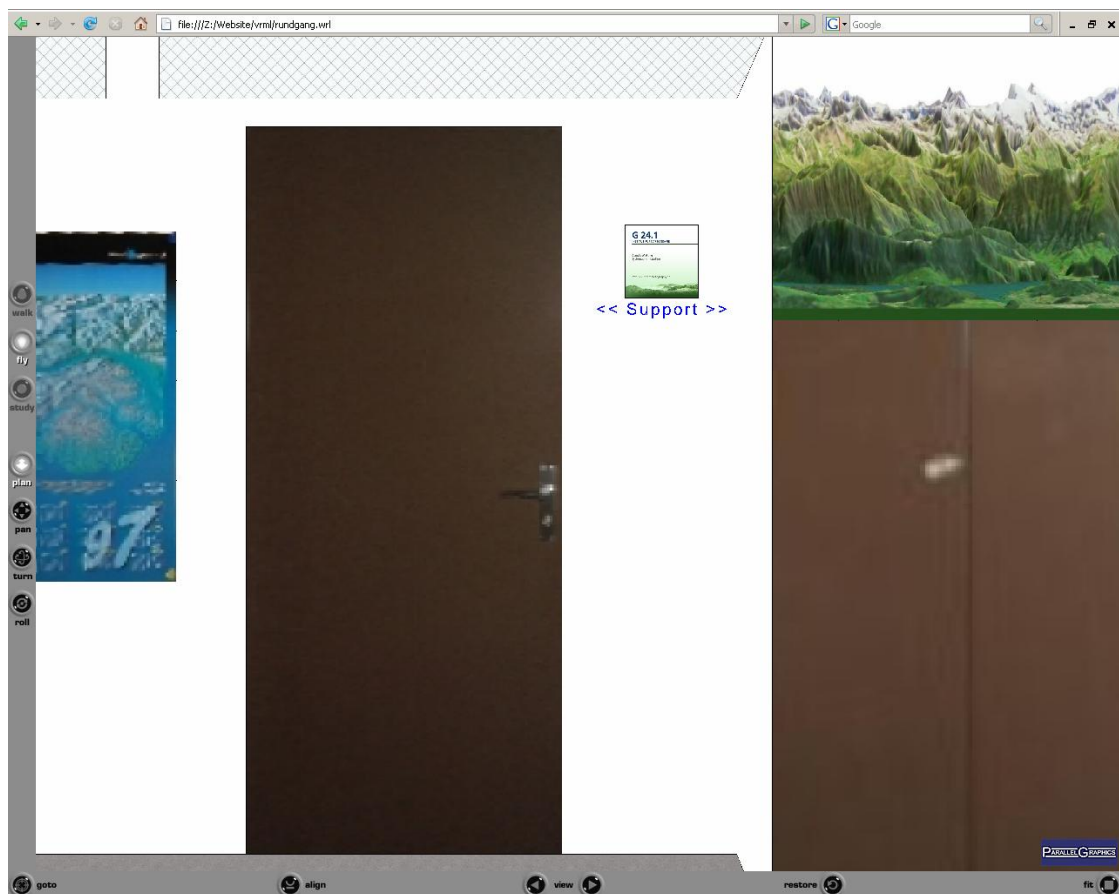


Abbildung 12: Das IKA im neuen Kleid

In beiden Rundgängen lassen sich durch Selektion der betreffenden Viewpoints die Wege zu den einzelnen Büros sowie weiteren Räumen und interessanten Objekten des IKA anzeigen. Diese Wege beginnen jeweils beim Fahrstuhl.

Durch einen Mausklick auf die einzelnen Türschilder werden diese herangezoomt, so dass sie besser gelesen werden können. Mehrere Links zu externen Webseiten bieten ausserdem weitere Informationen über einzelne Projekte, die in den Büros bearbeitet werden.

Es ist jederzeit möglich, durch Aktivierung eines anderen Viewpoints (vgl. Abbildung 13) zu einem anderen Zimmer bzw. Objekt oder zurück zum Fahrstuhl zu gehen.

Viewpoints	
✓ Start	
Fahrstuhl	
Bietschhorn-Relief	
G 10.1	- Serverraum
G 10.3	- Bibliothek / Kartensammlung
G 10.5-6	- Praktikumsraum
G 10.6	- Praktikumsraum
G 10.7	- Digitizer- und Plotter-Raum
G 12.1	- R. Geisthoevel, St. Huber, P. Jeller
G 12.2	- R. Sieber, A. Wipf, M. Rueegsegger
G 13.2	- M. Hugentobler, I. Iosifescu, D. Isenegger
G 13.3	- B. Jenny, Ch. Lienert
G 14.1	- F. He, H. Jenny, J. Trau
G 14.2	- S. Ahrer, T. Dahinden, E. Hutzler, R. Stopper
G 21	- Praktikumsraum
G 22	- Seminarraum
G 23	- Kartensammlung
G 24.1	- Claudia Matthys
G 24.2	- Ph. Foret, B. Piatti, M. Werner
G 24.3	- L. Hurri
G 25.1	- Sekretariat
G 25.2	- Besprechungszimmer / Zeitschriftensammlung
G 27.1	- A. Neumann, St. Raeber, O. Schnabel
G 27.2	- H.-R. Baer, Ch. Haeberling
G 27.3	- J. Cron, Ph. Marty, A. Weber

Abbildung 13: Liste der Viewpoints

Der Rundgang „rundgang.wrl“ bietet zusätzlich die Möglichkeit, die in den Fluren hängenden Poster als Pdfs anzusehen.

Die Rundgänge können unter <http://support.cartography.ch/> mittels des Links unterhalb der Flash-Animation geöffnet werden. Sie stellen eine nützliche und interessante Orientierungshilfe für potenzielle Besucher des IKA dar, welche sich bereits im Vorfeld über die genaue Position eines bestimmten Büros informieren möchten. Auch alle weiteren interessierten Personen erhalten durch die virtuellen Rundgänge einen ersten Eindruck des Instituts.

Zukünftige Änderungen am Aussehen des IKA können mit relativ wenig Aufwand zuerst im virtuellen Modell vorgenommen werden. Dadurch können die Änderungen noch vor der realen Umsetzung visualisiert und allenfalls angepasst werden.

5.4 ANLEITUNG ZUR WEITERBEARBEITUNG

Dieser Abschnitt richtet sich an Personen, welche die vorliegenden Modelle und Rundgänge weiterbearbeiten möchten. Zuerst wird aufgezeigt, wie Änderungen an den Modellen vorgenommen werden können – dies geschieht am zweckmässigsten in SketchUp. Anschliessend wird erklärt, was nach dem erneuten Export als VRML-Datei unternommen werden muss, damit sämtliche Funktionen der Rundgänge übernommen werden können.

5.4.1 ÄNDERUNGEN AN DEN MODELLEN

Sollen die Modelle geändert werden, wird empfohlen, dies in SketchUp auszuführen. Von einer direkten Manipulation des VRML-Codes wird abgeraten, da dies deutlich umständlicher ist und ein tieferes Verständnis der Codedatei voraussetzt. In SketchUp dagegen können Anpassungen sehr einfach vorgenommen und die veränderten Modelle anschliessend als neue VRML-Dateien exportiert werden.

Bearbeitung der Geometrien

Zur Bearbeitung der Geometrie in SketchUp sei auf Kapitel A 2 im Anhang verwiesen. Dort werden die bei der Erstellung der Modelle verwendeten wurden.

Bei der Veränderung der Geometrien muss beachtet werden, dass die nach dem Export in VRML eingefügten Quader, Schriften und Lichtquellen (siehe Kapitel 5.1.4) absolute Koordinaten besitzen, welche sich nach den Koordinaten der SketchUp-Modelle richten. Werden die Modelle oder bestimmte Komponenten in SketchUp verschoben, müssen in den selber erstellten Code-Fragmenten ebenfalls Änderungen vorgenommen werden.

Austausch der Texturfotos

Soll ein Texturfoto ausgetauscht werden, kann entweder das neue Foto neu importiert (siehe Kapitel A 2.7, Abschnitt „Import einer 2D-Grafik“) oder das alte Foto ausgewählt und ersetzt werden. Um ein Foto direkt zu ersetzen, ist folgendes Vorgehen notwendig:

- Öffnen der Komponente, zu welcher die betreffende Fläche gehört (durch Doppelklick auf die Komponente oder Selektion der Komponente → Rechtsklick → Edit Component).
- Selektion der betreffenden Fläche → Rechtsklick → Entity Info.
- Im Fenster „Entity Info“ bei „Face“ Klick auf das auszutauschende Foto.
- Im Fenster „Choose Paint“ Klick auf „Edit“.
- Im Fenster „Edit Material“ evtl. oben links einen neuen Namen eingeben. Dann das Häkchen bei „Use texture image“ deaktivieren und wieder aktivieren.
- Im Fenster „Choose Image“ das gewünschte Foto auswählen.

- Evtl. Textur zurechtziehen (siehe Kapitel A 2.7).

Falls das neue Foto genau gleich zurechtgezerrt werden soll wie das alte Foto, ist der letzte Schritt nicht mehr notwendig. Das ausgetauschte Foto bekommt nämlich automatisch dieselbe Zerrung wie das erste Foto.

Falls das neue Foto anders gezerrt werden soll, empfiehlt sich das in Kapitel A 2.7 beschriebene Vorgehen, da es schneller ist.

Export als VRML-Datei

Um das veränderte Modell als VRML-Datei exportieren zu können, muss SketchUp Pro 5 verwendet werden (siehe Kapitel 3.2.3).

Nach dem Export müssen die von der Autorin selber erstellten Code-Fragmente neu eingefügt werden. Dies geschieht am zweckmässigsten, indem eine ältere Version des VRML-Codes geöffnet und die benötigten Code-Teile in den neuen Export hineinkopiert werden.

Um deutlich zu machen, welche Code-Fragmente aus dem Export von SketchUp stammen und welche im Nachhinein von Hand hinzugefügt wurden, sind sämtliche händisch erstellten Teile durch die Zeichen „#####“ vom restlichen Code abgegrenzt. Die „#####“ stehen dabei jeweils am Anfang und am Ende der betreffenden Code-Teile.

Sämtliche händisch hinzugefügte Code-Fragmente stehen ausserdem entweder vor oder nach dem Export-Code. Es müssen also nach einem neuen Export der SketchUp-Datei nur zwei kompakte Code-Portionen kopiert werden.

5.4.2 ÄNDERUNGEN AN DEN VRML-FUNKTIONEN

Änderungen an den VRML-Funktionen können direkt im Code vorgenommen werden. Dazu muss die VRML-Datei in einem Texteditor oder einem syntaxunterstützten Editor (wie VrmIpad) geöffnet werden.

Sollen Koordinaten aus SketchUp abgegriffen werden, müssen diese ins Koordinatensystem von VRML umgerechnet werden. Dazu kann die Excel-Datei „Koordinaten.xls“ (siehe Kapitel 5.1.4) verwendet werden.

Die in der Excel-Datei aufgelisteten Punkte beziehen sich auf die in der SketchUp-Datei „Pfade.skp“ eingetragenen Positionen.

Soll auf Elemente im Export-Teil des Codes zugegriffen werden (um beispielsweise Objekte zu animieren), müssen ihre Namen aus den SketchUp-Modellen in Erfahrung gebracht werden. In Kapitel 7.3 ist ein Beispiel dazu aufgeführt.

6 AUFLISTUNG UND KRITISCHE BEURTEILUNG DER ERGEBNISSE

Das primäre Ziel der Diplomarbeit, die virtuelle Präsentation des Instituts für Kartografie, wurde erreicht. Wie verlangt, wird als Erstes ein grober Überblick über den Standort des IKA vermittelt. Mithilfe einer zweiten Applikation wird den Besuchern der Website anschliessend die Möglichkeit geboten, sich innerhalb des Instituts umzusehen, sich den Weg zu den einzelnen Büros und wichtigen Objekten anzeigen zu lassen sowie weitergehende Informationen zu erhalten.

Im Folgenden werden die während der Diplomarbeit erstellten Produkte nochmals zusammenfassend aufgelistet. Danach werden diese Produkte, der eingeschlagene Lösungsweg und die verwendeten Programme kritisch beurteilt.

6.1 ERSTELLTE PRODUKTE

6.1.1 FLASH-ANIMATION

Die Flash-Animation wurde mit Macromedia Flash Professional 8 erstellt. Sie zeigt eine rotierende Weltkugel, welche bis auf Stufe Schweiz hineinzoomt. Anschliessend erfolgt eine Überblendung zu einem Situationsplan der ETH Höggerberg.

Inhaltlich zu den anderen erstellten Produkten gehörend, ist die Animation aus technischer Sicht gänzlich unabhängig von diesen. Sie liefert einen groben Überblick über den Standort des Instituts für Kartografie. Vor allem aber dient sie als Blickfang, um die Besucher der Website auf den virtuellen Rundgang durch das Institut für Kartografie aufmerksam zu machen.

Zu finden ist die Animation unter <http://support.cartography.ch/>.

6.1.2 (VOLLSTÄNDIGES) MODELL DES IKA

Das mit Google SketchUp erstellte 3D-Modell beinhaltet sämtliche sich auf der Etage G befindenden Räume des IKA sowie den Eingangsbereiches des HIL (Etage D). Die Flure des IKA wurden dabei möglichst vollständig und realitätsnah modelliert und texturiert. Die Zimmer sind jedoch nur rudimentär texturiert und nicht möbliert.

Dieses grosse Modell kann beispielsweise für die Erstellung eines vollständigen Rundganges mit einer Game-Engine verwendet werden. Ausserdem kann das IKA offline direkt in SketchUp betrachtet werden.

6.1.3 SCHLANKES MODELL FÜR VRML / WEB

Ausgehend von dem grossen Modell des IKA wurde eine schlanke Version erstellt, welche klein genug ist, um für eine direkte Betrachtung im Web aufbereitet zu werden. Dazu wurden die Räume gelöscht, Geometrien vereinfacht und die Texturfotos verkleinert.

Die Dateigrösse dieses verkleinerten Modells beträgt nur einen Bruchteil derjenigen des ursprünglichen Modells.

6.1.4 SCHLANKES MODELL MIT NEUEM GESTALTUNGSKONZEPT

Durch wenige Änderungen des verkleinerten Modells wurde anschliessend das neue Gestaltungskonzept des IKA bestmöglich integriert. Die Änderungen liegen im Wesentlichen in einer Neugestaltung der Schränke und Türschilder, in einer Umplatzierung des Bietschhorn-Reliefs sowie in einem Austauschen der in den Fluren hängenden Poster.

6.1.5 RUNDGÄNGE IN VRML

Die beiden schlanken Modelle wurden schliesslich als VRML-Dateien exportiert. Die beiden daraus in VRML erstellten Rundgänge bieten mehrere Funktionen, um sich innerhalb der virtuellen Modelle des IKA zurechtzufinden und zusätzliche Informationen zu erhalten. Beispielsweise können die Wege zu den einzelnen Büros angezeigt werden, und über Links werden die Websites verschiedener mit einzelnen Büros in Verbindung stehender Projekte angezeigt.

Die Rundgänge werden durch den Link unterhalb der Animation (auf <http://support.cartography.ch/>) gestartet.

6.2 BEURTEILUNGEN UND ERKENNTNISSE

6.2.1 BEURTEILUNG DER PRODUKTE

Flash-Animation

Die Flash-Animation lässt sich wie gewünscht übers Web auf allen mit einem Flash-Player ausgestatteten Rechnern betrachten. Sie erfüllt die in der Aufgabenstellung geforderten Eigenschaften. Der Übergang von der Stufe Schweiz zur ETH Höggerberg ist jedoch etwas unsanft. Die Einbindung eines weiteren Zooms von der Stufe Schweiz über Zürich zur ETH Höggerberg würde die Animation bestimmt aufwerten.

Modelle des IKA

Die erstellten 3D-Modelle erheben den Anspruch, das Institut für Kartografie möglichst realitätsnah abzubilden. Ein Modell wird jedoch immer eine Vereinfachung der Wirklichkeit darstellen; es ist

niemals möglich, alle Aspekte und Details einzubringen. Sowohl aus Zeit- oder Kostengründen als auch aus technischen Gründen (für eine Darstellung im Web darf das Modell nicht zu gross sein) müssen Einschränkungen in Kauf genommen werden.

Unter Berücksichtigung dieser Punkte können die beiden kleineren Modelle als genügend detailliert und exakt beurteilt werden.

Beim grossen Modell muss bemängelt werden, dass die Zimmer nur sehr grob texturiert sind und ausserdem keine Möblierung beinhalten. Es hätte jedoch den Rahmen dieser Diplomarbeit gesprengt, wären die Räume ebenfalls so detailliert abgebildet worden wie die Flure.

Generell ist die Texturierung der Modelle nicht optimal. So konnten die beiden Modelle des Bietschhorns und des Zervreila-Staudammes wegen ihrer Höhe und Ausdehnung nicht von oben fotografiert werden. Als Folge davon sind sie nur auf den Seiten texturiert. Da die Türen nicht genau von vorne fotografiert werden konnten, sind die Türfallen leicht seitlich dargestellt. Beim grossen Modell werden ausserdem die Deckenlichter nicht regelmässig dargestellt, weil sie aus verschiedenen Fotos zusammengesetzt sind.

Rundgänge in VRML

Die erstellten VRML-Rundgänge weisen nicht alle gewünschten Eigenschaften auf. Die ursprünglich geplante Einbindung von Sound zur Unterstützung des realitätsnahen Empfindens wurde nicht realisiert. Und die freie Navigation ist zwar möglich, jedoch relativ mühsam und unhandlich zu bedienen. Alle wesentlichen Elemente zur Erfüllung der in der Diplomarbeit gesetzten Ziele sind jedoch in genügender Qualität vorhanden.

Die Rundgänge können im Web in Echtzeit beschritten werden, sofern die Datenübertragungsrate nicht zu klein ist. Allerdings bestehen Unterschiede zwischen den Darstellungen in einzelnen Browsern. Die Rundgänge wurden mit dem Cortona VRML-Client und Mozilla Firefox erstellt; das Resultat sieht dort am besten aus.

Dank der Tatsache, der fast aller eigenhändig erstellte Code entweder vor oder nach dem Export-Code eingefügt wurde (und nicht dazwischen), lässt er sich sehr einfach in ein neu exportiertes, abgeändertes Modell übernehmen. Die Forderung nach einfacher Nachführung und Weiterbearbeitung ist somit erfüllt.

6.2.2 DER LÖSUNGSWEG UND DIE VERWENDETE SOFTWARE

Flash-Animation

Der Einsatz von Flash für die Animation der Weltkugel hat sich als zweckmässig erwiesen. Die Arbeit mit Flash bestätigt, dass diese Software für Web-Anwendungen sehr geeignet ist. Auch nach dem Einfügen mehrerer Vektorgrafiken bleibt die Datei klein genug, um übers Web problemlos angezeigt zu werden.

Flash ist ausserdem relativ benutzerfreundlich. Bereits mit kleinem Aufwand lassen sich anschauliche Resultate erzielen.

Dieser Teil des Lösungsweges kann somit nicht bemängelt werden.

Modelle des IKA

Die Auswahl von SketchUp für die Modellierung ist ebenfalls sinnvoll. SketchUp besticht durch seine einfache und intuitive Bedienung. Dank den umfangreichen Hilfe-Texten und den praktischen Tutorials ist der SketchUp-Neuling schnell mit den wichtigsten Funktionen der Software vertraut und kann effizient arbeiten.

Ein grosser Pluspunkt sind die Funktionen zur Texturierung von Oberflächen. Im Vergleich zu anderer 3D-Modellierungs-Software wie 3D Studio Max oder Blender ist das Texturieren in SketchUp extrem einfach.

Die einfache Bedienbarkeit und das rasche Arbeiten verleitet dazu, den Modellierungsaufwand von grossen oder komplexen Modellen zu unterschätzen. Dazu soll gesagt sein, dass die einzelnen Elemente mit SketchUp zwar sehr schnell erstellt werden können; müssen jedoch hunderte solcher Elemente modelliert werden, summieren sich die Arbeitszeiten aber natürlich dennoch beträchtlich.

Ein Nachteil von SketchUp besteht darin, dass die Dateien schnell sehr gross werden. Insbesondere der Import von Textur-Fotos sowie von vorgefertigten, oft komplexen Modellen aus der 3D-Galerie lässt eine Datei schnell anwachsen. Dateigrössen von mehreren Dutzend MBytes sind die Folge.

Besonders tückisch in Bezug auf die Dateigrösse ist der Umstand, dass kleine, aber komplexe Modelle (wozu viele von der 3D-Galerie importierte Objekte gehören) die SketchUp-Datei zwar nicht wesentlich grösser machen, beim VRML-Export die VRML-Datei aber deutlich vergrössern. Das Verhältnis zwischen SketchUp- und VRML-Dateigrösse ist also nicht linear, sondern hängt von der Beschaffenheit der jeweiligen Modelle ab.

Die Nachteile von SketchUp wiegen aber deutlich weniger als seine Vorteile; deshalb kann die Verwendung von SketchUp für Arbeiten wie die Modellierung des IKA empfohlen werden.

Bei der Ausführung gibt es jedoch ein paar Punkte, die hätten optimiert werden können. Für die Texturierung der einzelnen Türschilder (heutiger Zustand) wurden Fotos erstellt. Für das neue Gestaltungskonzept lagen sie jedoch als digitale Dateien vor, was ihre Einbindung in das Modell deutlich vereinfachte und zu besseren Ergebnissen führte. Es hätte deshalb versucht werden sollen, auch für die aktuellen Türschilder digitale Daten zu erhalten.

Bei den Postern sieht die Situation ähnlich aus. Zuerst wurden die Poster fotografiert; erst später wurde versucht, ihre digitalen Dateien zu erhalten, was auch mehrheitlich gelang. Wäre direkt zu Beginn nach digitalen Daten gesucht worden, hätte damit einige mühsame Fotografierarbeit erspart werden können.

Generell hätten mit dem Einsatz einer Kamera mit Weitwinkelobjektiv sowie einem Stativ qualitativ bessere Texturfotos erstellt werden können. Hier muss jedoch das Verhältnis von Aufwand und Ertrag berücksichtigt werden.

Die Verwendung eines Messbandes zur Bestimmung der Ausmasse der einzelnen Elemente genügt für die zu erreichende Genauigkeit sowie die Anzahl der auszumessenden Teile. Es lässt sich schnell und einfach verwenden und ist somit für die vorliegende Arbeit gut geeignet. Im Hinblick auf eine spätere Weiterbearbeitung (insbesondere eine Ergänzung des grossen Modells mit der Büromöblierung, was eine grosse Anzahl zu vermessender Elemente bedeutet) sollte jedoch untersucht werden, ob andere Techniken wie beispielsweise ein Laserscanner besser geeignet sind.

Rundgänge in VRML

VRML ist dank der Menge an verfügbaren Beispielen und Anleitungen (sowohl im Internet als auch in Buchform) schnell erlernbar. Da der VRML-Code textbasiert ist, kann er problemlos in jedem Texteditor bearbeitet werden. Ein spezieller, die VRML-Syntax unterstützender Editor erleichtert zwar die Arbeit, stellt aber – wie die Erfahrung mit der Bearbeitung des Rundganges in einem gewöhnlichen Editor zeigt – absolut kein Muss dar.

Die Textbasiertheit des VRML-Codes ist jedoch auch der Grund dafür, dass das VRML-Format, verglichen mit binären Formaten, sehr umfangreich ist. Die VRML-Datei des kleineren IKA-Modells hatte nach dem Export aus SketchUp beispielsweise bereits über 60'000 Zeilen Code. Dies erschwert das Programmieren; und zwar vor allem dann, wenn auf einzelne Teile innerhalb des Codes zugegriffen werden muss.

Ein weiterer Nachteil von VRML liegt darin, dass sich diese Beschreibungssprache nicht für grosse Modelle eignet (siehe Kapitel xy (Beschreibung, was warum mit grossem modell nicht geht)). Bei grossen Modellen ist die Performance ungenügend oder sogar unbrauchbar langsam.

Die Navigationsfunktionen sind ein weiterer Schwachpunkt von VRML. Mögen sie für Landschaften und Aussenansichten genügen, so sind sie aber für das freie Navigieren in Innenräumen mühsam und unhandlich.

Schliesslich stellt auch die Tatsache ein Problem dar, dass nicht jeder VRML-Viewer eine VRML-Datei gleich darstellt. Dies führt dazu, dass eine für einen bestimmten Viewer erstellte VRML-Datei in einem anderen Viewer unter Umständen nur unbefriedigend gerendert wird.

Abschliessend kann gesagt werden, dass VRML wegen der Performance bei grossen Modellen und der unpraktischen Navigation in Innenräumen nicht geeignet ist, um eine komplexe Anwendung zu erstellen. Für die Darstellung kleinerer Modelle mit einfacheren Funktionen und ohne Notwendigkeit des freien Navigierens eignet sich VRML aber durchaus. Es bleibt somit zu prüfen, ob VRML für eine Echtzeit-Darstellung und -Navigation in Innenräumen übers Web die beste Lösung ist. Die mit VRML

erreichten Resultate sind nicht gänzlich befriedigend, und die Erstellung neuer Codefragmente ist unter Umständen mit viel Fleissarbeit verbunden.

7 AUSBLICK

Die in der vorliegenden Diplomarbeit erstellten Produkte können auf vielfältige Weise weiterverwendet werden. In den folgenden Kapiteln werden einige Anregungen gegeben, wie die vorhandenen Erzeugnisse ausgebaut oder als Grundlagen für neue Arbeiten übernommen werden können.

7.1 AUSBAU DER FLASH-ANIMATION

Der Übergang zwischen der Stufe Schweiz und dem Situationsplan der ETH Höggerberg ist relativ abrupt. Es wäre möglich, weitere Vektorgrafiken einzufügen. Mit diesen Grafiken könnte ein lückenloses Zoomen von der Stufe Schweiz über Zürich bis zur Stufe ETH Höggerberg erzeugt werden.

Als Ergänzung zu diesem neuen Zoomvorgang liesse sich auf Stufe Zürich ein Link zu einer zweiten Flash-Animation einfügen. In dieser zweiten Animation könnten die wichtigsten Verkehrsverbindungen zur ETH Höggerberg angezeigt werden; beispielsweise mit animierten Verkehrsmitteln des öffentlichen Verkehrs (Bahn, Tram, Bus) ab Zürich Hauptbahnhof sowie der Anzeige der vom Individualverkehr zu benutzenden Strassen ab den Autobahnausfahrten.

Als weiterer Zusatz könnte am derzeitigen Ende der Animation ein Zoom angehängt werden. Von der Stufe ETH Höggerberg könnte so weit hineingezoomt werden, bis das HIL deutlich zu erkennen ist. Ein neuer Pfeil würde anschliessend den Haupteingang anzeigen. Dadurch würde gewissermassen ein lückenloser Übergang zwischen der Flash-Animation und dem VRML-Rundgang entstehen.

7.2 ERWEITERUNG DER 3D-MODELLE

Die drei Modelle des IKA können jederzeit verändert, verbessert oder aktualisiert werden. Sie können in verschiedene Formate exportiert werden (siehe Tabelle 1). Dadurch besteht die Möglichkeit einer vielseitigen Weiterverwendung für unterschiedlichste Zwecke.

7.2.1 VERVOLLSTÄNDIGUNG DES GROSSEN MODELLS

Da das grosse Modell nicht die Bedingung erfüllen muss, eine möglichst kleine Dateigrösse aufzuweisen, lässt es sich stark ausbauen. Um ein wirklich vollständiges Modell zu erhalten, müssen vor allem die Büros besser texturieren und mit Möbeln versehen werden. Dadurch könnten sie – wird das Modell beispielsweise mit einer Game-Engine weiterbearbeitet – betreten werden.

Darüber hinaus lassen sich in den Fluren beliebige Details ergänzen, um das Modell noch realitätsnaher zu gestalten.

Dieses Modell kann natürlich jederzeit an den aktuellen Zustand des Instituts angepasst werden. Besonders im Hinblick auf die Umsetzung des Gestaltungskonzepts soll das Modell aktualisiert werden, sobald dieses Konzept im IKA realisiert worden ist.

Das Modell kann auch dahingehend erweitert werden, dass die auf anderen Stockwerken liegenden Büros des IKA ebenfalls erreicht werden können. Die Einbindung der Cafeteria sowie des ganzen HIL-Gebäudes ist denkbar.

7.2.2 OPTIMIERUNG DER KLEINEN MODELLE

Bei den verkleinerten Modellen ist darauf zu achten, dass die Datei nicht wesentlich grösser wird. Ansonsten wird ein allfälliger Neuexport in VRML nicht mehr in Echtzeit dargestellt.

Diese Einschränkung der Dateigrösse führt dazu, dass die Modelle nur noch wenig erweitert werden können. Insbesondere ist es nicht möglich, die weggelassenen Räume wieder zu integrieren. Auch können keine grossen Textur-Dateien importiert werden.

Es ist wünschenswert, die aus den kleinen Modellen gelöschten Pflanzen wieder zu integrieren. Sie stammen ursprünglich aus der 3D-Galerie von Google. Diese Google-Modelle sind in der Regel relativ komplex aufgebaut und führen zu einer erheblichen Vergrösserung der Datei. Sie dürfen deshalb nicht ohne Abklärung ihrer Auswirkungen in die IKA-Modelle eingefügt werden. Dabei ist zu berücksichtigen, dass sich ein solches Google-Modell unter Umständen die Grösse der SketchUp-Datei nur unwesentlich verändert, die aus SketchUp exportierte VRML-Datei jedoch beträchtlich vergrössern kann.

Da die Modelle möglichst klein gehalten werden sollen, können Anstrengungen zur Minimierung der Dateigrösse unternommen werden. Der Austausch der Fotos der Türschilder durch die digitalen Daten (beim Modell des aktuellen Zustandes) würde die Datei verkleinern und zusätzlich ein besseres Resultat erzielen. Jedoch dürfte es sich als schwierig erweisen, diese digitalen Daten zu erhalten.

Es ist zu prüfen, ob das Ersetzen bestimmter Foto-Texturen durch selbst erstellte Grafiken (z. B. bei Türen und Böden) eine Verringerung der Dateigrösse mit sich bringt, ohne den realitätsnahen Eindruck wesentlich zu schmälern.

Falls die beiden Modelle für nicht für VRML, sondern für eine andere Applikation gebraucht werden sollen, können sie natürlich – je nach den Möglichkeiten dieser Applikation – weiter ausgebaut werden.

Grundsätzlich stellt sich die Frage, ob es längerfristig Sinn macht, beide Modelle weiterzuführen. Die reale Umsetzung des neuen Gestaltungskonzepts hat soeben begonnen. Somit ist es am sinnvollsten, sich bei der Aktualisierung der Modelle auf das Modell des neuen Gestaltungskonzeptes zu konzentrieren und dieses an den neuen Zustand des IKA anzupassen. Das momentan noch aktuelle, aber in Kürze veraltete Modell des jetzigen Zustandes kann unverändert erhalten werden, um zu späteren Zeiten das „frühere“ Aussehen des IKA betrachten zu können.

7.3 WEITERE FUNKTIONEN FÜR DIE VRML-RUNDGÄNGE

Die Modelle für den VRML-Rundgang können nicht mehr wesentlich erweitert werden (siehe Kapitel 7.2.2). Jedoch besteht noch viel Spielraum, um neue Funktionen in den Code einzufügen.

Von der Eingangshalle des HIL (Etage D) gelangt man momentan mittels eines Hyperlinks zu den Räumlichkeiten des IKA in der Etage G. Denkbar wäre, diesen Übergang durch Animation des Fahrstuhls interessanter zu gestalten. Der Weg zum IKA könnte dann wie in Wirklichkeit beschriftet werden: Die Fahrstuhltüren öffnen sich, und man betritt den Fahrstuhl. Danach schliessen sich die Türen, der Fahrstuhl fährt hinauf zur Etage G, die Türen öffnen sich erneut, und man tritt auf den Flur des IKA hinaus.

Um einzelne Elemente wie Türen animieren zu können, müssen gezielte Veränderungen im Export-Teil des Codes vorgenommen werden. Da der Code sehr lang und unübersichtlich ist, müssen die Namen der interessierenden Elemente bekannt sein, damit man den Code danach absuchen kann. Dazu kann SketchUp zu Hilfe genommen werden. Die Komponenten der Fahrstuhltüren beispielsweise heissen in SketchUp „Tuere_Lift_links“, „Tuere_Lift_rechts“ und „Tuere_Lift_innen“. Die Namen werden beim Export nach VRML übernommen und können deshalb im Code wiedergefunden werden.

Mit einem ähnlichen Prinzip könnte auch eine Tour durch die Flure des IKA erstellt werden. Dazu müssten die Türen im Treppenhaus animiert werden. Die Tour würde beim Fahrstuhl starten und durch einen der beiden seitlichen Flure zum Treppenhaus führen. Die Türen des Treppenhauses würden sich anschliessend öffnen, so dass man durch sie hindurch zum Flur auf der anderen Seite gehen kann. Von dort aus würde die Tour durch diesen Flur zurück zu den Fahrstühlen führen. Damit wäre die Tour ein geschlossener Kreis.

Mittels Hyperlinks kann ein VRML-Modell mit einem anderen verlinkt werden. Eine solche Funktion ist bereits im Code des aktuellen Modells integriert, wodurch das Modell des neuen Gestaltungskonzept geöffnet werden kann.

Mit derselben Methode könnte das Problem der Dateigrösse eliminiert werden. Dazu müssten die einzelnen Zimmer separat in SketchUp erstellt und als eigene VRML-Modelle exportiert werden. Anschliessend könnten im Flur-Modell bei den einzelnen Türen Hyperlinks angebracht werden. Durch diese Hyperlinks liessen sich vom Flur aus die entsprechenden Zimmer aufrufen.

7.4 EINBINDUNG DES MODELLS IN EIN COMPUTERSPIEL

Aus dem grossen Modell kann mittels einer Game-Engine ein eigenes Computerspiel erstellt werden. In diesem Spiel könnten sämtliche in Kapitel 2.1.2 aufgeführten Funktionen – insbesondere die Möglichkeit eines bequemen freien Navigierens sowie die Einbettung von Sound – umgesetzt werden.

Dank der in Kapitel 3.4.1 aufgeführten Module lassen sich mit einer Game-Engine vielfältige weitere Anwendungen realisieren. Es besteht die Möglichkeit, ein interaktives Informationssystem zu entwickeln, weitere Navigationsfunktionen einzufügen sowie das Modell beispielsweise mit den Institutsbüros der Etage E zu erweitern. Mit der vorliegenden Arbeit ist das Grundgerüst für diese Weiterbearbeitungen gegeben.

Darüber hinaus liesse sich bei Verwendung einer geeigneten Game-Engine ein für den Mehrbenutzerbetrieb optimiertes Computerspiel entwickeln. Dadurch könnten sich mehrerer Personen gleichzeitig im Modell aufhalten und sich gegenseitig sehen sowie miteinander interagieren. Die Möglichkeiten reichen von einfachen Chat-Funktionen bis zu virtuellen Konferenzen.

8 QUELLENVERZEICHNIS

8.1 LITERATUR

Ammann, E. (1997): Programmierung animierter Welten – Java, JavaScript und VRML. Bonn: International Thomson Publishing GmbH, 310 S.

Beck, M. (2002): Realisierung eines Geoinformationssystems – Visualisierung und Analysefunktionalität mit einer 3D Engine. Diplomarbeit, Institut für Photogrammetrie, Universität Stuttgart, 99 S.

Marrin, Ch.; Campbell, B. (1997): Teach Yourself VRML 2 in 21 days. Indianapolis: sams net, 479 S.

Shiratuddin, M. F.; Thabet, W. (2002): Virtual Office Walkthrough Using a 3D Game Engine. Department of Building Construction, Virginia Polytechnic Institute and State University, Blacksburg, USA, 25 S. http://faculty.arch.usyd.edu.au/kcdc/journal/vol4/shira/ijdc_final_journal.pdf

8.2 INTERNET

- [1] Internet Dienstleistungen, Vienna: Virtual Reality – Die Welt aus dem Computer. http://www.vienna.cc/d/artikel/gratis_report_virtual_reality.htm (Zugriff 25. Juni 2007).
- [2] Fritsch, D. (2005): Die Landkarte im Spannungsfeld von Internet und Multimedia. Festvortrag, Universität Stuttgart. http://www.ifp.uni-stuttgart.de/publications/2006/051112%20Festvortrag%20Fritsch%20final_web.pdf (Zugriff 20. März 2007)
- [3] Blohm, Ph.; Golubchik, R. et al. (2006): Computergrafikpraktikum 2006. <http://project.informatik.uos.de/cgp/> (Zugriff 5. Juli 2007)
- [4] Aschenbacher, N. (2006): Konzeption und Realisierung einer 3D-Benutzerschnittstelle zur Konfiguration und Visualisierung von Digitalen Fabriken – Evaluierung von Modellierungswerkzeugen und Implementierung mit OpenGL und jMonkey Engine in Java. Diplomarbeit, Universität Kassel, Kassel, 60 S. http://www.se.eecs.uni-kassel.de/se/fileadmin/se/publications/Diplomarbeit_NinaAschenbrenner.pdf (Zugriff 5. Juli 2007)
- [5] Wikipedia – Die freie Enzyklopädie: Animation. <http://de.wikipedia.org/wiki/Animation> (Zugriff 5. Juli 2007)
- [6] Wikipedia – Die freie Enzyklopädie: Computeranimation. <http://de.wikipedia.org/wiki/Computeranimation> (Zugriff 5. Juli 2007)
- [7] Wikipedia – Die freie Enzyklopädie: Scalable Vector Graphics. http://de.wikipedia.org/wiki/Scalable_Vector_Graphics (Zugriff 5. Juli 2007)
- [8] Wikipedia – Die freie Enzyklopädie: 3D-Grafik-Software. <http://de.wikipedia.org/wiki/3D-Grafik-Software> (Zugriff 5. Juli 2007)

- [9] Wikipedia – Die freie Enzyklopädie: 3ds Max. http://de.wikipedia.org/wiki/Scalable_Vector_Graphics (Zugriff 5. Juli 2007)
- [10] Wikipedia – Die freie Enzyklopädie: Maya (Software). http://de.wikipedia.org/wiki/Maya_%28Software%29 (Zugriff 5. Juli 2007)
- [11] Google: Deutsche SketchUp-Seite. <http://sketchup.google.de/> (Zugriff 29. Juni 2007)
- [12] Wikipedia – Die freie Enzyklopädie: Auszeichnungssprache. <http://de.wikipedia.org/wiki/Auszeichnungssprache> (Zugriff 5. Juli 2007)
- [13] Butz, A.: Computergrafik und Virtuelle Realität. Vorlesung Digitale Medien, Ludwig-Maximilians-Universität München, Wintersemester 2006/2007. <http://www.medien.ifi.lmu.de/lehre/ws0607/dm/vorlesung/dm11b.pdf> (Zugriff 25. Juni 2007)
- [14] Wikipedia – Die freie Enzyklopädie: Spiel-Engine. <http://de.wikipedia.org/wiki/Spiel-Engine> (Zugriff April 2007)
- [15] Stang, B. (2003): Game Engines – Features and possibilities. <http://www.garagegames.com/uploaded/GameEngines2.pdf> (Zugriff 18. April 2007)
- [16] DevMaster's Game and Graphics Engines Database. <http://www.devmaster.net/engines/> (Zugriff 29. Juni 2007)
- [17] The Game Programming Wiki. http://gpwiki.org/index.php/Game_Engines (Zugriff 29 Juni 2007)
- [18] Wikipedia – The free Encyclopedia: List of game engines. http://en.wikipedia.org/wiki/List_of_game_engines (Zugriff April 2007)
- [19] 3D Engines List. <http://cg.cs.tu-berlin.de/~ki/engines.html> (Zugriff April 2000)
- [20] 3D and Game Engine. <http://fraktali.849pm.com/graphengines.html> (Zugriff 29. Juni 2007)
- [21] Flash4all – Tutorials und Effekte. <http://www.flash4all.de/tutorialsneu/tutorialsneu.htm> (Zugriff 20. März 2007)
- [22] Google: Deutsche Google Earth-Seite. <http://earth.google.de/> (Zugriff 2. Juli 2007)
- [23] Wikipedia – Die freie Enzyklopädie: VRML. <http://de.wikipedia.org/wiki/VRML> (Zugriff Juni 2007)
- [24] Parallel Graphics: Website von Parallel Graphics. <http://www.cortona3d.com/> (Zugriff 29. Juni 2007)

8.3 DATEN

- [D1] Digital Wisdom, Inc. (1997): Mountain High Maps Version 2.5. 4 CDs, Tappahannock (USA).
- [D2] Institut für Kartografie der ETH Zürich (2007): ETH Situationspläne.
http://www.karto.ethz.ch/services/downloads/eth_maps → Hönggerberg → ZIP-Datei:
Illustrator & Font → Datei „ETH-Hoenggerberg_color.ai“ (Stand Januar 2007)
- [D3] ETH Zürich, Infrastrukturbereich Immobilien: Grundriss HIL D- und G-Stock, letztmals revidiert am 16. März 2007 bzw. 28. Oktober 2004. (Kontaktperson: ETH Zürich, Ines Dutli, IMMOBILIEN, Hochstrasse 60, HCH, CH-8092 Zürich)

A ANHANG

A 1 MACROMEDIA FLASH

In diesem Kapitel werden einige wichtige Begriffe und Funktionen aufgeführt, welche bei der Ausarbeitung der Diplomarbeit gebraucht werden. Für ausführlichere Informationen und Beschreibungen sei auf die Hilfe innerhalb der Flash-Umgebung verwiesen.

A 1.1 AUFBAU DES DOKUMENTES

Ein Flash-Dokument besteht aus vier Hauptteilen: Bühne, Zeitleiste, Bibliothek und ActionScript (siehe Abbildung 14).

In der Bühne (weisses Rechteck) werden die Inhalte wie Vektor- und Bitmapgrafiken, Textfelder und Schaltflächen platziert. Es ist ausserdem möglich, mit den Zeichen-Werkzeugen eine eigene Grafik direkt in der Bühne zu erstellen. Die Bühne zeigt das an, was beim Abspielen des Flash-Filmes zu sehen sein wird.

In der Zeitleiste wird festgelegt, zu welcher Zeit diese Inhalte angezeigt werden sollen. Hier kann ausserdem die Reihenfolge der einzelnen Ebenen auf der Bühne bestimmt werden. Elemente niedrigerer Ebenen werden von Inhalten höherer Ebenen überdeckt.

Die Bibliothek zeigt die im aktuellen Flash-Dokument enthaltenen Elemente an.

Im ActionScript-Fenster wird der Code eingegeben. Mit ActionScript-Code können verschiedene Anweisungen und Benutzer-Interaktionen programmiert werden. Soll beispielsweise eine Animation bei einem bestimmten Bild stoppen, wird diesem Bild der ActionScript-Code

```
stop();
```

hinzugefügt.

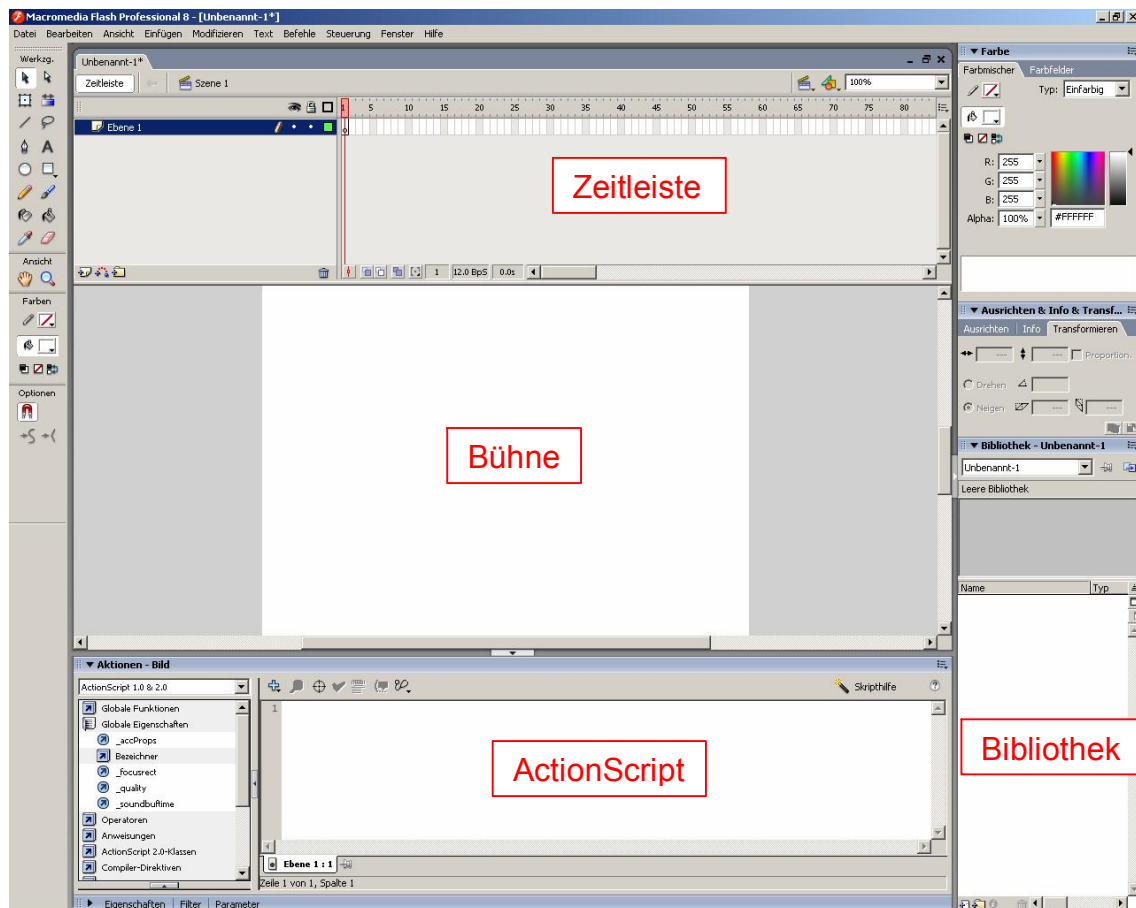


Abbildung 14: Leeres Flash-Dokument

A 1.2 ANIMATIONEN MIT HILFE VON TWEENS

Will man eine Tween-Animation erstellen, müssen zuerst zwei Schlüsselbilder⁹ erzeugt werden, die Beginn und Ende der Animation anzeigen. Flash fügt dann die zwischen den beiden Schlüsselbildern liegenden Bilder automatisch ein, wodurch eine fließende Animation erzeugt wird. Dies ist eine schnelle und unkomplizierte Methode, um Animationen zu erstellen.

Es gibt zwei verschiedene Tweens: Bewegungs-Tweens und Form-Tweens.

Bei einem Bewegungs-Tween wird ein Objekt von einer Position zu einer anderen bewegt. Die beiden Schlüsselbilder stellen dabei die Anfangs- bzw. Endposition des Objektes dar.

Ein Form-Tween erlaubt als Änderung vom ersten zum zweiten Schlüsselbild sowohl Form- als auch Farb-, Positions-, Grössen- und Transparenzveränderungen. Bei Formveränderungen entsteht ein morphing-ähnlicher Effekt, bei welchem eine Form schrittweise in eine andere übergeht.

⁹ Ein Schlüsselbild ist ein Bild, in welches Änderungen von Objekteigenschaften für eine Animation definiert werden können und in welches ActionScript-Code eingefügt werden kann.

In der Zeitleiste von Flash werden die verschiedenen Einzelbilder und Animationen entsprechend kenntlich dargestellt. Abbildung 15 zeigt die wichtigsten Darstellungsarten.

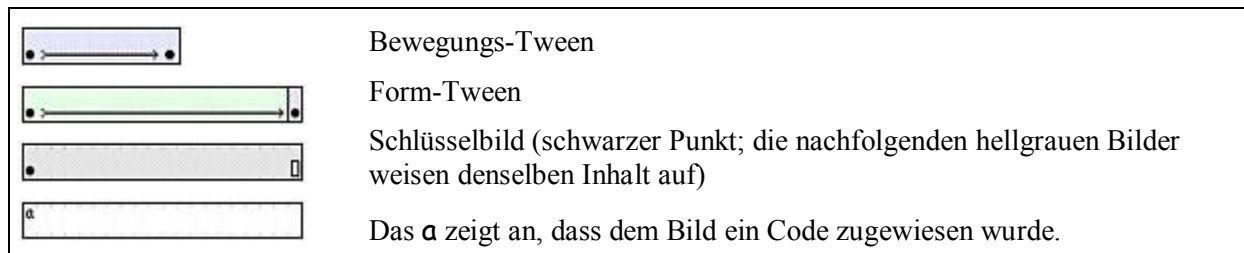


Abbildung 15: Darstellungsarten in der Zeitleiste

Quelle: Flash-Hilfe.

A 1.3 ZEITLEISTEN-EFFEKTE

In Flash sind so genannte Zeitleisten-Effekte bereits vordefiniert. Jeder Zeitleisten-Effekt manipuliert eine Grafik oder ein Symbol auf eine bestimmte Art. Durch Anpassung der Parameter können mit kleinstem Zeitaufwand komplexe Animationen erstellt werden.

In der Diplomarbeit werden die beiden Effekte „Transformieren“ und „Übergang“ verwendet. Mit ihnen wird bestimmt, wie sich ein ausgewähltes Element über einen gewissen Zeitraum hinweg verändern soll. Die Effekte beginnen jeweils beim ausgewählten Element und erstrecken sich über eine frei bestimmbare Anzahl Bilder.

Mit „Transformieren“ kann das ausgewählte Element verschoben, skaliert und gedreht werden. Ausserdem können sein Farbton und seine Transparenz (Alpha-Wert) verändert werden. Kombinationen zwischen den einzelnen Manipulationen sind ebenfalls möglich.

„Übergang“ wird verwendet, um das ausgewählte Element auf eine bestimmte Art und Weise ein- oder auszublenden.

A 1.4 MASKEN

Eine Maske erlaubt das selektive Anzeigen von Teilen der zu ihr gehörenden, darunter liegenden Ebenen. Wenn auch nicht ganz korrekt, so lässt sich die Funktion einer Maske am intuitivsten dadurch erklären, dass man sich eine „Öffnung“ vorstellt, durch welche die darunter liegenden Ebenen sichtbar sind.

A 2 GOOGLE SKETCHUP

In diesem Kapitel werden die wichtigsten Funktionen von Google SketchUp beschrieben, welche für die Erstellung der IKA-Modelle verwendet werden. Ziel dieses Kapitels ist in erster Linie, einem späteren Benutzer von Google SketchUp, welcher noch keine Vorkenntnisse hat, einen leichteren und schnelleren Einstieg zu ermöglichen. Die einzelnen Unterkapitel erheben deshalb keinen Anspruch auf Vollständigkeit. Im Anhang dieses Berichtes (Kapitel A 1) finden sich Anweisungen zur Benützung einzelner Tools. Für detailliertere und weiterführende Angaben sei auf die ausführliche Hilfe in Google SketchUp (in Englisch) sowie auf die hilfreichen Tutorials und Videobeispiele von SketchUp verwiesen.

A 2.1 DAS KOORDINATENSYSTEM VON SKETCHUP

SketchUp benutzt ein 3D-Koordinatensystem mit den drei Achsen X (grün), Y (rot) und Z (blau). Z repräsentiert dabei die Höhe über oder unter der horizontalen Grundfläche, welche von der X- und Y-Achse aufgespannt wird. Die positiven Achsrichtungen werden als ausgezogene, die negativen Achsrichtungen als gepunktete Linien dargestellt. Abbildung 16 zeigt das Koordinatensystem. Mit dem *Change Axes*-Tool kann die Position der drei Achsen geändert werden.

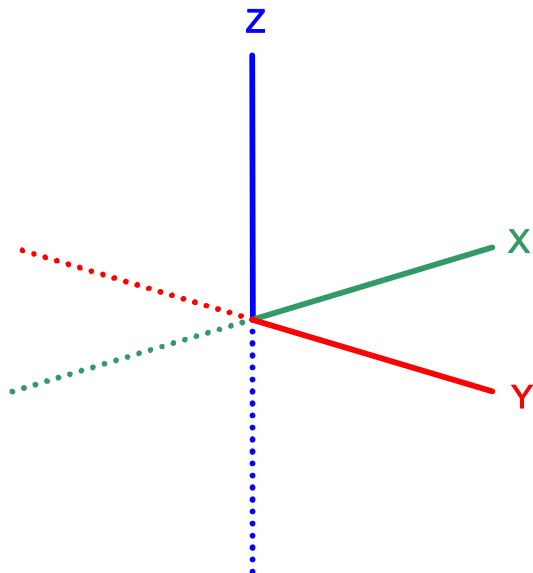


Abbildung 16: Das Koordinatensystem von SketchUp

A 2.2 MASSEINHEITEN EINSTELLEN

Unter dem Menü *Window* → *Model Info* lassen sich verschiedene Einstellungen vornehmen. Besonders wichtig für das Konstruieren eines Modells anhand eines realen Vorbildes ist es, sinnvolle Masseinheiten einzustellen. Unter der Registerkarte *Units* kann zwischen mehreren Formaten ausgewählt werden. Für die vorliegende Arbeit wird die Einheit *Centimeters* (Zentimeter) eingestellt.

A 2.3 POINT INFERENCE

Für exakte und schnelle Konstruktionen und Messungen ist das in SketchUp automatisch ausgeführte Snappen eine grosse Hilfe. Will man beispielsweise eine Linie zeichnen, welche bei einem ganz bestimmten Punkt eines bereits erstellten Gebildes beginnt, braucht man mit dem Mauszeiger lediglich in die Nähe dieses Punktes zu fahren – es wird dann auf den gewünschten Punkt gesnappt. Ein Point Inference gibt dabei Auskunft darüber, auf welcher Art von Punkt sich der Mauszeiger befindet. Die fünf wichtigsten Fälle sind in Abbildung 17 aufgezeigt:

- Endpoint (grün): Endpunkt einer Linie oder eines Bogens
- Midpoint (blaugrün): Mittelpunkt einer Linie oder Kante
- Intersection (schwarz): Stelle, wo eine Linie sich mit einer zweiten Linie oder einer Fläche schneidet
- On Face (blau): Punkt, welcher auf einer Fläche liegt
- On Edge (rot): Punkt, welcher auf einer Kante liegt

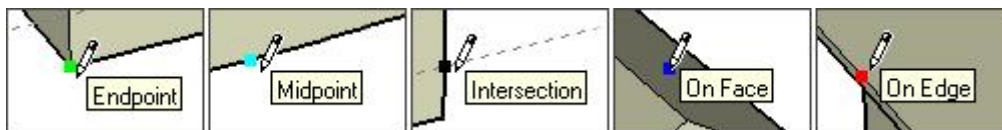


Abbildung 17: Die wichtigsten Fälle des Point Inference

Quelle: SketchUp-Hilfe.

A 2.4 LINE TOOL

Mit dem Line Tool werden Linien und Kanten konstruiert. Dabei können die exakte Länge einer Linie oder sogar die Koordinaten ihrer Endpunkte eingegeben werden.

Beim Zeichnen einer Linie zeigt die Value Control Box (VCB. Sie befindet sich am rechten unteren Rand des SketchUp-Fensters. Siehe Abbildung 18) ihre aktuelle Länge an. Um eine Linie mit einer bestimmten Länge zu erhalten, kann während der Konstruktion die gewünschte Länge direkt in die VCB eingegeben werden.

Der Endpunkt einer Linie ist gleichzeitig automatisch der Startpunkt einer allenfalls zu zeichnenden nächsten Linie. Für diese nächste Linie muss also nur noch ihr Endpunkt gesetzt werden. Wird auf diese Weise ein geschlossenes, koplanares Polygon erzeugt, wird dieses automatisch als Fläche dargestellt. Um das Polygon zu schliessen und somit die Fläche zu erzeugen, ist es wichtig, dass für den Endpunkt der letzten Linie auf den Startpunkt der ersten Linie gesnappt wird.

Für die Erzeugung von primitiven Flächen können auch – einfacher – die drei Flächen-Tools *Rectangle*, *Circle* und *Polygon* verwendet werden.



Abbildung 18: Value Control Box

Erstellung von Linien

Folgende Schritte sind zur Erstellung von Linien nötig:

- Line Tool selektieren. Der Mauszeiger ändert sein Aussehen zu einem Bleistift.
- Auf den gewünschten Ort klicken, um den Startpunkt zu setzen.
- Den Mauszeiger zum gewünschten Endpunkt bewegen.
- Klicken, um den Endpunkt zu setzen und somit die Linie abzuschliessen.

Definieren exakter Masse

Um eine Linie mit einer bestimmten Länge zu erhalten, ist folgendes Vorgehen am zweckmässigsten:

- Line Tool selektieren.
- Startpunkt setzen.
- Den Mauszeiger in Richtung des gewünschten Endpunktes bewegen, so dass die Linie in die gewünschte Richtung zeigt.
- Mit den Zifferntasten der Tastatur die gewünschte Länge eingeben. Die Eingabe wird in der VCB angezeigt. Der eingegebene Wert bezieht sich dabei – sofern er nur numerisch, also ohne Masseinheit eingegeben wurde – auf die im aktuellen Dokument eingestellte Masseinheit.
- Die Enter-Taste drücken, um den Endpunkt zu setzen und somit die Linie abzuschliessen.

A 2.5 PUSH/PULL-TOOL

Mit diesem Tool lässt sich eine Fläche zu einem Körper „aufziehen“. Ausserdem ist es möglich, bereits vorhandene Körper weiter zu vergrössern oder zu verkleinern. Auch bei diesem Tool kann in die VCB ein exakter Wert eingetragen werden.

In SketchUp werden die Körper innen hohl dargestellt. Dies ist von grossem Vorteil, wenn man zum Beispiel Innenräume darstellen will. Man erspart sich das „Aushöhlen“ der Körper.

Um eine Fläche zu einem Körper „aufzuziehen“, sind folgende Schritte erforderlich:

- Push/Pull Tool selektieren.
- Mauszeiger zu derjenigen Fläche bewegen, welche gezogen werden soll.
- Bei gedrückter Maustaste die Fläche ziehen.
- Gegebenenfalls nach dem Loslassen der Maustaste den exakten Wert eingeben, um welchen die Fläche aufgezogen werden soll.

Abbildung 19 zeigt die Anwendung des Push/Pull-Tools.

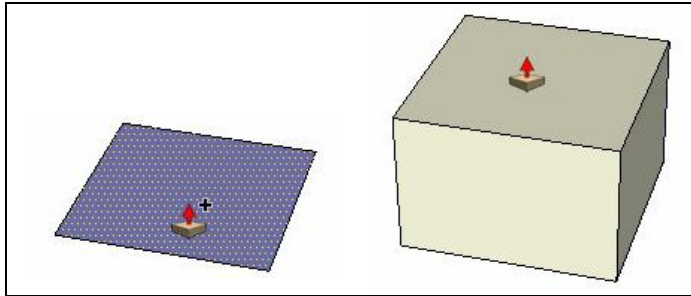


Abbildung 19: Anwendung des Push/Pull-Tools

Links: Ausgangsfläche; rechts: aufgezogener Körper. Quelle: SketchUp-Hilfe.

A 2.6 TAPE MEASURE TOOL

Mit dem *Tape Measure*-Tool können Distanzen gemessen, Konstruktionslinien und -punkte erstellt sowie das Modell skaliert werden.

Um ein Modell zu skalieren, misst man eine Referenzstrecke und gibt anschliessend in der VCB ihre neue, gewünschte Länge ein. Der Umrechnungsfaktor zwischen der ursprünglichen und der neuen Länge der Referenzstrecke wird anschliessend von SketchUp verwendet, um das ganze Modell entsprechend zu skalieren.

Distanzmessung

Um die Distanz zwischen zwei Punkten zu messen, ist folgendes Vorgehen notwendig:

- Das Tape Measure Tool selektieren. Der Mauszeiger ändert sein Aussehen zu einem Massband.
- Auf den ersten Punkt klicken, um die Messung zu starten.
- Auf den zweiten Punkt klicken. Die gemessene Distanz wird als Tooltip sowie in der VCB angezeigt.

Skalierung

Um das ganze Modell oder einzelne Komponenten zu skalieren, ist folgendes Vorgehen nötig:

- Das Tape Measure Tool selektieren.
- Die Messung durchführen, wie oben unter “Distanz messen” beschrieben.
- Anschliessend die gewünschte neue Länge eintippen, die das gemessene Stück erhalten soll. Die neue Länge wird in der VCB angezeigt.
- Die Enter-Taste drücken.

- In der folgenden Dialogbox („Do you want to resize the model?“) auf “Yes” klicken.

Die eingegebene neue Länge wird als Referenzlänge für ein proportionales Skalieren des Modells verwendet.

A 2.7 TEXTURIERUNG

Oft ist es sinnvoll, eine Fläche mit beispielsweise einem Foto zu texturieren, um die Konstruktion einer komplexen Geometrie zu vermeiden und dem Modell ein fotorealistisches Aussehen zu verleihen. So erspart beispielsweise das Bild einer Tür die Konstruktion der Türfalle.

In SketchUp wird im Zusammenhang mit Texturierungen der Begriff „Material“ verwendet. Dies können Farbtöne sein, mit welchen eine Fläche eingefärbt wird, sowie Muster, Zeichnungen oder Fotos, welche als Texturen auf Flächen aufgebracht werden.

In Google SketchUp ist unter dem Menüpunkt *Window* → *Material Browser* eine Vielzahl verschiedener Materialien bereits vorgegeben. Es können jedoch auch andere Materialien hinzugefügt werden. SketchUp erlaubt nämlich den Import von 2D-Rasterbildern verschiedener Formate. Nach dem Import sind die neuen Materialien im *Material Browser* gespeichert und können von dort aus jederzeit wieder aufgerufen werden. Durch Klicken auf die zu texturierende Fläche wird das Material darauf platziert. Falls nötig, wiederholt es sich mehrmals, um die gesamte Fläche zu füllen.

Um ein neu platziertes Material auf der Fläche zu verschieben, seine Grösse zu ändern oder es wie ein Gummituch in die passende Form zu ziehen, werden Stecknadeln (Pins) verwendet. Mit ihrer Hilfe kann das Material auf der Fläche zurechtgezerrt werden. Abbildung 20 zeigt den Prozess des Anpassens einer Textur auf eine Fläche.

Wie bereits angedeutet, werden neue Materialien importiert; es wird nicht nur eine Verknüpfung zum Speicherort des Materials hergestellt. Dies hat gewisse Nachteile. Zum einen wird die SketchUp-Datei bei vielen Importen schnell sehr gross, was während der Arbeit zu einem langsameren Reagieren der Datei führen kann. Zum anderen kann nicht einfach extern ein Material geändert werden, indem zum Beispiel ein Foto ausgetauscht oder bearbeitet wird. Es ist immer auch nötig, das geänderte Material neu zu importieren.

Der Vorteil der importierten Materialien liegt jedoch darin, dass beim Verschieben der SketchUp-Datei auf einen anderen Computer nicht darauf geachtet werden muss, dass alle verknüpften Materialien mitgeliefert werden.

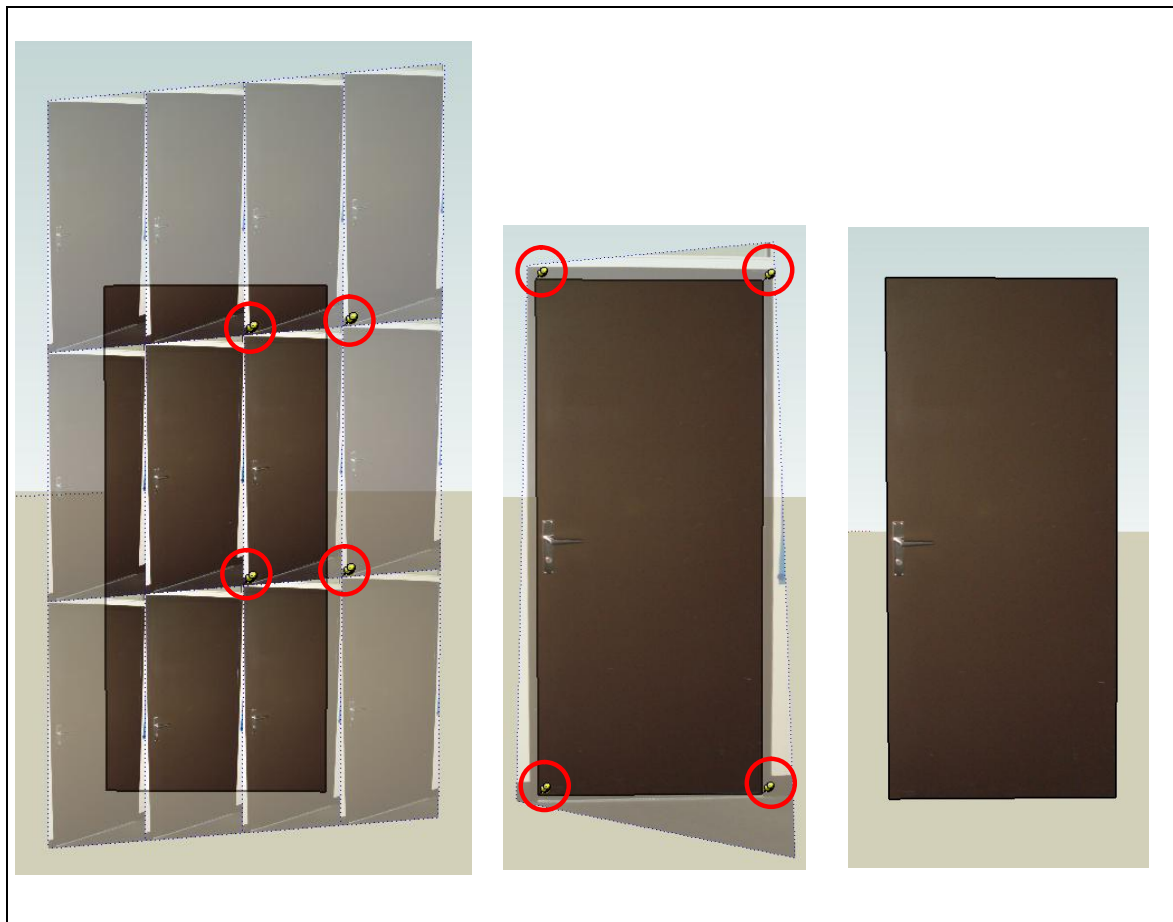


Abbildung 20: Anpassung einer Textur

Links: Ausgangszustand; die Textur-Wiederholung ist deutlich zu erkennen. Mitte: Zurechtgezogene Textur. Rechts: Aussehen nach Beendigung des Anpassungsprozesses. (Zur besseren Sichtbarkeit sind die Pins rot eingekreist.)

Import einer 2D-Grafik

Um ein Foto (oder eine andere 2D-Grafik) als Material in SketchUp zu verwenden, muss es mit folgenden Schritten importiert werden:

- Im Menu „File“ den Punkt „Import“ wählen.
- Im Fenster „Öffnen“ das Kontrollkästchen „Use as Texture“ aktivieren.
- Die gewünschte Datei importieren. Der Mauszeiger ändert sein Aussehen zu einem Farbeimer.
- Das neue Material auf der gewünschten Fläche platzieren. Falls nötig, wird es sich mehrmals wiederholen, um die gesamte Fläche zu füllen.

Das neue Material ist nun im Material Browser gespeichert und kann von dort aus (Menü Window → Material Browser) jederzeit wieder aufgerufen werden.

Position Texture Tool

Um ein Material auf einer Fläche zu verschieben, seine Grösse zu ändern oder es wie ein Gummituch in die passende Form zu zerren, werden Stecknadeln (Pins) verwendet. Dabei gibt es zwei Modi: Fixed Pin Mode und Free Pin Mode. Im Folgenden wird nur auf das Verzerren eines Materials im Free Pin Mode eingegangen, da dies die einzige benötigte Aktion ist, um ein Material auf eine Oberfläche anzupassen.

Im Free Pin Modus wird ein Material mit folgenden Schritten auf eine Fläche zurechtgezerrt:

- Mit der rechten Maustaste auf das Material (d. h. auf die Fläche, auf welcher das Material liegt) klicken.
- Vom erscheinenden Menu *Texture* → *Position* selektieren. Ein Gitter aus gepunkteten Linien erscheint auf dem Material, um dessen einzelne Teile (Kacheln) anzuzeigen. Der Mauszeiger ändert sein Aussehen zu einer Hand, und vier Stecknadeln werden angezeigt.
- Mit der rechten Maustaste auf das Material klicken.
- Darauf achten, dass bei „Fixed Pins“ kein Häkchen ersichtlich ist.
- Die einzelnen Stecknadeln an die gewünschte Position auf dem Material bringen. (Durch Anklicken der Stecknadeln können sie vom Material losgelöst werden. Ein zweites Klicken an der neuen Position verbindet sie wieder mit dem Material.) Abb. zeigt diesen Schritt.
- Das Material durch ziehen der Stecknadeln bei gedrückter Maustaste so zerren, dass es die gewünschte Position und Form einnimmt.
- Mit der rechten Maustaste klicken und im erscheinenden Menu „Done“ auswählen. Damit ist die Bearbeitung abgeschlossen. Abb. zeigt das Resultat.

A 2.8 KOMPONENTEN UND INSTANZEN

Ein Modell kann in einzelne Komponenten (engl.: Components) aufgeteilt werden. Um eine Komponente zu erstellen, werden alle zugehörigen Konstruktionen markiert und dann im Kontext-Menü (erscheint beim Drücken der rechten Maustaste) *Make Component* ausgewählt. Die erstellte Komponente ist nun im Components-Browser (zu finden unter *Windows* → *Components*) gespeichert und kann von dort aus beliebig viele Male als so genannte Instanz (engl.: Instance) in die Zeichenfläche eingefügt werden. Dabei sind sämtliche Instanzen dieser Komponente miteinander verbunden, d. h. wenn eine Instanz bearbeitet wird (dazu muss im Kontext-Menü *Edit Component* gewählt werden), verändern sich alle anderen Instanzen ebenso. Dies ist sehr praktisch, wenn beispielsweise in einem Flur mehrere identische Schränke stehen. Man muss nur einmal die Komponente *Schrank* konstruieren und kann dann so viele Instanzen von *Schrank* in die Zeichenfläche einfügen, wie man braucht. Will man später alle Schränke verändern, muss man dies ebenfalls nur einmal bei einer Instanz tun – alle anderen Instanzen werden automatisch ebenfalls geändert.

Alle Änderungen, die jedoch an einer „geschlossenen“ Instanz vorgenommen werden – also ohne im Kontext-Menü *Edit Component* ausgewählt zu haben – wirken sich nicht auf die anderen Instanzen

derselben Komponente aus. So können Instanzen beispielsweise unabhängig voneinander rotiert werden.

Mit *Make Unique* (ebenfalls im Kontext-Menü zu finden) wird aus einer Instanz eine neue, unabhängige Komponente.

Unter dem Menüpunkt *Google → Get Models* wird die 3D-Galerie von Google aufgerufen. Hier ist eine grosse Anzahl von 3D-Modellen (Möbel, Pflanzen, Menschen und vieles weitere) vorhanden. Die Modelle können nach Belieben ins eigene Modell importiert werden und werden dann als eigene Komponenten im aktuellen Dokument abgelegt. Sie können genau gleich bearbeitet werden wie die selber erstellten Komponenten.

A 2.9 REFERENZIEREN EINES MODELLS

Ein in Google SketchUp erstelltes Modell kann mithilfe der *Get Current View*-Funktion georeferenziert werden. Dazu muss zuerst Google Earth¹⁰ gestartet werden. In Google Earth wird nun zu derjenigen Gegend navigiert, in welcher das Modell zu liegen kommen soll. Anschliessend wird in Google SketchUp der *Get Current View*-Button gedrückt, um die aktuelle Ansicht von Google Earth zu importieren. Jetzt kann das Modell so verschoben werden, dass es an der gewünschten Stelle der Google-Earth-Ansicht zu liegen kommt. Diese Art der Georeferenzierung ist natürlich nicht exakt, weil die Platzierung ausschliesslich anhand visueller Kriterien erfolgt.

A 2.10 IMPORT- UND EXPORTFORMATE VON SKETCHUP

	Google SketchUp	SketchUp Pro 5
Importformate	3DS (*.3ds) DEM (*.dem) (*.ddf) ACAD Files (*.dwg, *.dxf) SketchUp-Dateien (*.skp) JPEG Image (*.jpg) Portable Network Graphics (*.png) Tagged Image-Datei (*.tif) Targa-Datei (*.tga) Windows Bitmap (*.bmp)	3DS (*.3ds) DEM (*.dem) (*.ddf) ACAD Files (*.dwg, *.dxf) SketchUp-Dateien (*.skp) JPEG Image (*.jpg) Portable Network Graphics (*.png) Tagged Image-Datei (*.tif) Targa-Datei (*.tga) Windows Bitmap (*.bmp)

¹⁰ Google Earth ist eine Gratissoftware zur Darstellung eines virtuellen Globus. Siehe [22].

2D-Exportformate	Windows Bitmap (*.bmp) JPEG Image (*.jpg) Tagged Image File (*.tif) Portable Network Graphics (*.png)	Portable document Format (*.pdf) Encapsulated PostScript Format (*.eps) Windows Bitmap (*.bmp) JPEG Image (*.jpg) Tagged Image File (*.tif) Portable Network Graphics (*.png) Epix (*.epx) AutoCAD DWG (*.dwg) AutoCAD DXF (*.dxf)
3D-Exportformate	Google Earth (*.kmz) Google Earth 4 (*.kmz)	3DS (*.3ds) AutoCAD DWG (*.dwg) AutoCAD DXF (*.dxf) FBX (*.fbx) OBJ (*.obj) XSI (*.xsi) VRML (*.wrl)

Tabelle 1: Die Import- und Exportmöglichkeiten von Google SketchUp und SketchUp Pro 5

A 3 VRML

A 3.1 DEFINITIONEN UND FUNKTIONEN

In diesem Kapitel werden die in der Diplomarbeit verwendeten Elemente und Funktionen erklärt. Sämtliche eigenhändig erstellte Code-Fragmente sind ausserdem direkt im Code kommentiert.

Kopfzeile

```
#VRML V2.0 utf8
```

Das Zeichen „#“ wird normalerweise gebraucht, um Kommentare in den Code einzufügen, die vom Browser nicht interpretiert werden sollen. Eine Ausnahme bildet jedoch die erste Zeile des Codes, die so genannte Kopfzeile. Hier wird das Sonderzeichen nicht als Kommentar, sondern als Teil des Codes interpretiert.

Die Kopfzeile sagt dem Browser, dass es sich um eine VRML-Datei der Version 2.0 handelt. „utf8“ ist ein Standard zur Darstellung von Zeichen; ähnlich wie ASCII.

Koordinatensystem und Masseinheiten

In (Ammann, 1997, S. 154) wird das Koordinatensystem von VRML folgendermassen beschrieben: „VRML arbeitet mit dem üblichen dreidimensionalen Raum mit (x, y, z)-Koordinaten, wobei die sogenannte Rechte-Hand-Regel für die Achsen gilt (x-Achse in Richtung des ausgestreckten Daumens der rechten Hand, y-Achse in Richtung des abgespreizten Zeigefingers und z-Achse in Richtung des abgespreizten Mittelfingers dieser Hand). Die Standarddarstellung im WWW-Browser ist dabei so, dass die x-Achse nach rechts zeigt, die y-Achse nach oben und die z-Achse aus der Darstellungsebene heraus zum Betrachter hin.“ Die Rotationen folgen der Rechten-Hand-Regel für Rotationen, „wo der abgespreizte Daumen der rechten Hand in Richtung der Rotationsachse zeigt und das Heranbewegen der übrigen vier Finger an die Handinnenfläche die positive Drehrichtung anzeigt“ (Ammann, 1997, S. 154). Der positive Drehsinn entspricht somit – zum Koordinatenursprung hin gesehen – dem Gegenuhrzeigersinn.

Abbildung 21 zeigt das Koordinatensystem von VRML.

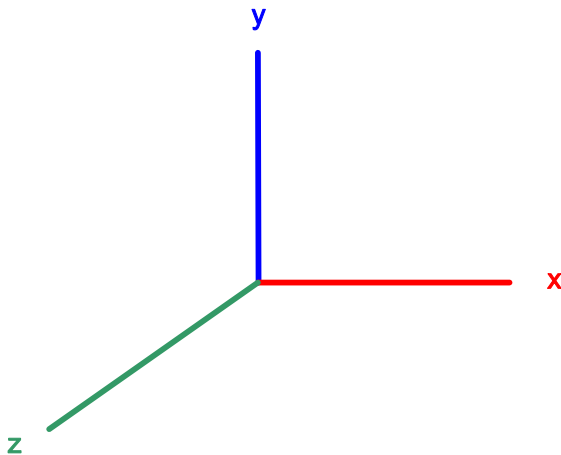


Abbildung 21: Koordinatensystem von VRML

Die Masseinheiten von VRML sind nicht an reale Werte gebunden. Um die Arbeiten mehrerer Autoren miteinander verbinden zu können, gibt es jedoch Vereinbarungen:

- Winkelmasse werden in VRML in Radiant angegeben. 90° entsprechen dabei $\pi/2$ bzw. 1.5712 rad.
- Als Zeiteinheit werden Sekunden verwendet.
- Längeneinheiten werden grundsätzlich in Metern angegeben. Der Vergleich zwischen den Massen in SketchUp und VRML zeigt jedoch, dass die Meterangaben von SketchUp beim Export nach VRML um einen Faktor 2.54 verkleinert werden. Die Metermasse werden also gewissermassen in Zoll-Werte umgerechnet.

Knoten-Prinzip

Eine VRML-Szene ist aus einer Folge von Knoten zusammengesetzt. Knoten (engl.: nodes) sind dreidimensionale Grafikobjekte (Elemente) bzw. grafische Eigenschaften solcher Elemente (Geometrien, Materialien, Lichter, Farben, Sound, Viewpoints, Sensoren, Animationen usw.). Durch so genannte Gruppierungsknoten können ausserdem mehrere Knoten zu Gruppen zusammengefasst werden.

VRML-Knoten werden folgendermassen definiert:

```
Knotentyp { Parameter }
```

In den geschweiften Klammern werden entweder die Parameter des Knotens aufgeführt oder (bei Gruppierungsknoten) die zu gruppierenden Kindknoten.

Ein Gruppierungsknoten sieht dann beispielsweise folgendermassen aus:

```
Group { Children [ ] }
```

Eine sehr praktische Zusammenstellung über die in VRML verwendbaren Knoten findet sich in Anhang A des Buches von Marrin und Campbell (1997).

Im Umgang mit Knoten sowie generell mit VRML-Code muss beachtet werden, dass die Syntax von VRML auf Gross- und Kleinschreibung empfindlich ist.

DEF und USE

Dank den Schlüsselwörtern „DEF“ und „USE“ müssen mehrfach gebrauchte Elemente nur einmal definiert werden. Die Syntax zur Definition des im letzten Abschnitt erwähnten Knotens sieht folgendermassen aus:

```
DEF Knotenname Knotentyp { Parameter }
```

Um diesen Knoten später wieder aufzurufen, muss lediglich

```
USE Knotenname
```

einggegeben werden.

Transformation

```
Transform {  
    translation 10 20 30  
    rotation 1 0 0 -1.570796  
    scale 2 2 2  
    # evtl. weitere Parameter  
    children [  
        # Hier stehen die zu transformierenden Elemente.  
    ]  
}
```

Dieser Code zeigt die Parameter des Knotens „Transform“. Mit seiner Hilfe können die üblichen drei Transformationen vorgenommen werden: Translation, Rotation und Skalierung.

Es werden hauptsächlich drei Parameter gebraucht:

- translation: Translation. Die drei Werte geben die Grösse der Verschiebung in x-, y- und z-Richtung an.
- rotation: Rotation. Die ersten drei Werte stehen für die Koordinaten eines Punktes. Die Gerade zwischen diesem Punkt und dem Nullpunkt ist die Rotationsachse. Der vierte Wert bestimmt

die Grösse der Rotation. Im obigen Beispiel steht die Zahl -1.570796 für eine Rotation um die x-Achse um 90° im Uhrzeigersinn.

- scale: Skalierung. Die drei Werte geben die Grösse der Skalierung der x-, y- und z-Werte an.

Viewpoints

Mit dem Knoten „Viewpoint“ wird die Betrachterposition definiert. Ein Viewpoint weist die folgenden Parameter auf:

- position: Position. Die drei Werte stehen für die Koordinaten des Standpunktes (x, y, z).
- orientation: Orientierung. Die ersten drei Werte stehen für die Koordinaten eines Punktes. Die Gerade zwischen diesem Punkt und dem Nullpunkt ist die Rotationsachse. Der vierte Wert bestimmt die Grösse der Rotation.
- fieldOfView: Sichtfeld. Dieser Parameter ist vergleichbar mit dem Winkel eines Kameraobjektivs. Der Wert 0.785398 kommt dem menschlichen Auge am nächsten.
- description: Benennung des Viewpoints. Dieser Name erscheint im Viewpoint-Menü des Browsers. Wird dieser Parameter nicht gesetzt, erscheint der betroffene Viewpoint nicht im Browser-Menü.
- jump: Sprung. Ein „FALSE“ bedeutet, dass nicht direkt zum Viewpoint gesprungen wird, sondern dass der Betrachter langsam dort hin geführt wird.

Der folgende Code ist ein Beispiel für einen solchen Viewpoint. Hier wird der Viewpoint mit dem Namen „Start“ definiert.

```
DEF Start Viewpoint {  
    position 23597 -343 -4890  
    orientation 0 1 0 1.5712  
    fieldOfView 0.785398  
    description "Start"  
    jump TRUE  
}
```

Animation der Viewpoints

Am Beispiel des Weges zum Serverraum soll an dieser Stelle erklärt werden, wie ein Viewpoint animiert werden kann.

```
DEF G10_1 Viewpoint {  
    fieldOfView 1
```



```
description "G 10.1      -      Serverraum"

jump TRUE

}

DEF O_G10_1 OrientationInterpolator {

    key      [ 0, 0.13, 0.27, 0.47, 0.57, 0.76, 1 ]

    keyValue [ 0      1      0      1.5712,
               0      1      0      1.5712,
               0      1      0      2.0949,
               0      1      0      2.0949,
               0      1      0      1.5712,
               0      1      0      1.5712,
               0      1      0      0 ]

}

DEF P_G10_1 PositionInterpolator {

    key      [ 0, 0.13, 0.27, 0.47, 0.57, 0.76, 1 ]

    keyValue [ 23440      51      -4935,
               23428      51      -4935,
               23390      51      -4935,
               23312      51      -4732,
               23221      51      -4665,
               23093      51      -4665,
               22961      45      -4626 ]

}

DEF Time_G10_1 TimeSensor {

    cycleInterval 15

}

ROUTE G10_1.bindTime TO Time_G10_1.startTime
ROUTE Time_G10_1.fraction_changed TO P_G10_1.set_fraction
ROUTE Time_G10_1.fraction_changed TO O_G10_1.set_fraction
ROUTE P_G10_1.value_changed TO G10_1.set_position
```

```
ROUTE O_G10_1.value_changed TO G10_1.set_orientation
```

Hier wird der Weg zum Raum G 10.1 definiert. Dies geschieht, indem bestimmte Werte für die Positionen und Orientierungen während des Abschreitens des Weges definiert werden. Der Computer interpoliert dann zwischen diesen Werten, wodurch fließende Übergänge entstehen. Dabei spielen die beiden Knoten „OrientationInterpolator“ und „PositionInterpolator“ eine wichtige Rolle. Sie besitzen je zwei Parameter

- key: Die Aufteilung unter „cycleInterval“ (Knoten „TimeSensor“) angegebenen Gesamtzeit des Weges in einzelne Intervalle.
- keyValue: Die Werte der Orientierung bzw. Position, die am Ende der einzelnen Zeit-Intervalle erreicht sein sollen.

Die wichtigste Rolle wird jedoch von den „ROUTE“ übernommen. Ohne sie wären Animationen und Benutzer-Interaktionen nicht möglich. Mit „ROUTE“ werden Werte eines Knotens (so genannte Ereignisse) einem anderen Knoten übergeben. Im obigen Code-Fragment wird zum Beispiel der Wert des Parameters „bindTime“ des Knotens „G10_1“ an den Parameter „startTime“ des Knotens „Time_G10_1“ übergeben.

Die „ROUTE“ bewirken in obigem Code-Fragment Folgendes: Sobald der „Viewpoint“-Knoten „G10_1“ aktiviert wird, wird der Zeitpunkt der Aktivierung dem Knoten „Time_G10_1“ übergeben. Die „cycleInterval“-Zeit dieses Knotens beginnt. Während sie vergeht, sendet der Knoten kontinuierlich die vergangene Zeit an die Knoten „OrientationInterpolator“ und „PositionInterpolator“. Diese beiden Knoten senden ihrerseits kontinuierlich die interpolierten Werte der Orientierung bzw. Position an den Knoten „G10_1“, wodurch dieser stetig seine Orientierung und Position verändert. Die Animation wird sichtbar.

Touch-Sensor

Der Touch-Sensor wird dazu benutzt, um bestimmte Elemente auf Mausklick empfindlich zu machen. Wird mit der Maus auf eines der im Touch-Sensor eingebetteten Elemente geklickt, wird der entsprechende Touch-Sensor aktiviert.

Im unten stehenden Beispiel wird nach Aktivierung des Touch-Sensors „T_IKA“ der Wert TRUE an das VmlScript „Heranzoomen_IKA“ übergeben. Das Skript gibt darauf den Wert TRUE an den Viewpoint „Fahrstuhl“ weiter (siehe dazu auch unten beim Kapitel „Scripting“), worauf dieser aktiviert wird. Die aktuelle Ansicht wechselt zur Ansicht dieses Viewpoints.

```
Group {
  children [
    DEF T_IKA TouchSensor { }
    Shape {
```

```
# An dieser Stelle wird das Touch-Sensor-
Element eingefügt, also das Element, das auf
die Berührung reagieren soll.

    }

]

}

ROUTE T_IKA.isActive TO Heranzoomen_IKA.angeklickt
    # Der Hyperlink wurde angeklickt, und deshalb...
ROUTE Heranzoomen_IKA.aktivieren TO Fahrstuhl.set_bind
    # ... wird dieser Viewpoint aktiviert.
```

Hyperlinks

Ein Hyperlink besteht primär aus folgenden Parametern (siehe unten):

- url: Uniform Resource Locator. Gibt den Pfad an, welcher durch Betätigung des Hyperlinks geöffnet werden soll.
- description: Beschreibung. Dieser Text erscheint als Quickinfo-Text, wenn sich die Maus über dem Hyperlinkfeld befindet.
- parameter: Parameter. Der Eintrag „target=DESC“ bedeutet, dass der Link in einem neuen Fenster geöffnet wird.

Das folgende Beispiel zeigt den Hyperlink zu der Orchestra-Website.

```
Anchor {
    url "http://www.eu-orchestra.org"
    description "Website Orchestra"
    parameter ["target=DESC"]
    # evtl. weitere Parameter
    children [
        # an dieser Stelle wird der eigentliche
        Hyperlink (also z. B. ein Text) eingefügt.
    ]
}
```

Beleuchtung

Der Knoten „PointLight“ definiert eine punktförmige Lichtquelle, die ihr Licht gleichmässig in alle Richtungen aussendet. Er beinhaltet folgende Parameter:

- location: Standpunkt der Lichtquelle
- color: Farbe des Lichtes (hier: weisses Licht)
- radius: Radius des Lichteinflusses
- ambientIntensity: Die Intensität, die auf alle von diesem Licht beschienenen Objekte angewandt wird.

Dies ist ein Beispiel einer solchen Lichtquelle:

```
PointLight {  
    location 21979 59 -4711  
    color 1 1 1  
    radius 500  
    ambientIntensity 0  
    # evtl. weitere Parameter  
}
```

Scripting

Mit dem „Script“-Knoten lässt sich ein Script (z. B. ein Java- oder VRML-Script) einbinden. Im unten stehenden Beispiel wird die Variable „angeklickt“ als Input an das Script übergeben. Ist diese Variable aktiv, wird die Variable „aktivieren“ auf „TRUE“ gesetzt und als Output weitergegeben.

```
DEF Heranzoomen_G10_3 Script {  
    eventIn SFBool angeklickt  
    eventOut SFBool aktivieren  
    url "vrmlscript:  
        function angeklickt(active) {  
            aktivieren = TRUE;  
        }  
    }  
}
```

A 3.2 VRML-VIEWER

Damit eine VRML-Datei im Internet angezeigt werden kann, muss ein Plugin für den Browser vorhanden sein. Auf dem Markt sind zahlreiche solche Ergänzungsmodule für verschiedene Plattformen und Browser erhältlich. Bekannte Produkte sind Cortona, CosmoPlayer und FreeWRL. Auf [23] finden sich nützliche Auflistungen von aktuellen Plugins und Stand-Alone-Programmen.

Am Beispiel des Cortona VRML Clients von ParallelGraphics (siehe

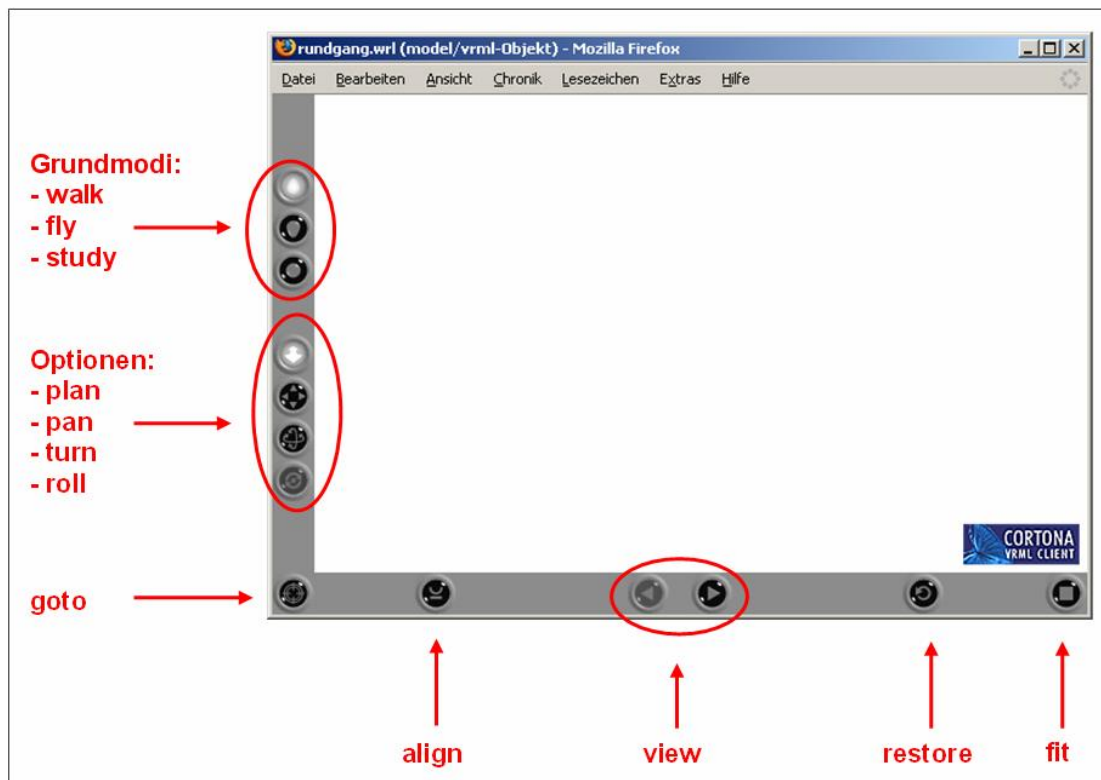


Abbildung 22 und [24]) sollen die verschiedenen Navigationsfunktionen eines VRML-Plugins vorgestellt werden. Die Bewegungen können dabei sowohl mit der Maus als auch mit den Pfeiltasten gesteuert werden.

- walk: Der Betrachter kann sich nur in der horizontalen Ebene bewegen.
- fly: Der Betrachter kann sich in der horizontalen und vertikalen Ebene bewegen.
- study: Die Szene wird um das Zentrum des Objekts bewegt.
- plan: Verschiebung in der horizontalen Ebene
- pan: Verschiebung in der vertikalen Ebene
- turn: Drehung in der horizontalen Ebene
- roll: Drehung in der vertikalen Ebene
- goto: Zoom zum angeklickten Objekt
- align: Setzt die Kamera-Achsen parallel zu der horizontalen Ebene der Szene.
- view: Navigation zwischen den Viewpoints

- restore: Verschiebung der Position zurück zum zuletzt aktivierten Viewpoint
- fit: Bewirkt, dass die ganze Szene im Cortona-Fenster sichtbar ist.

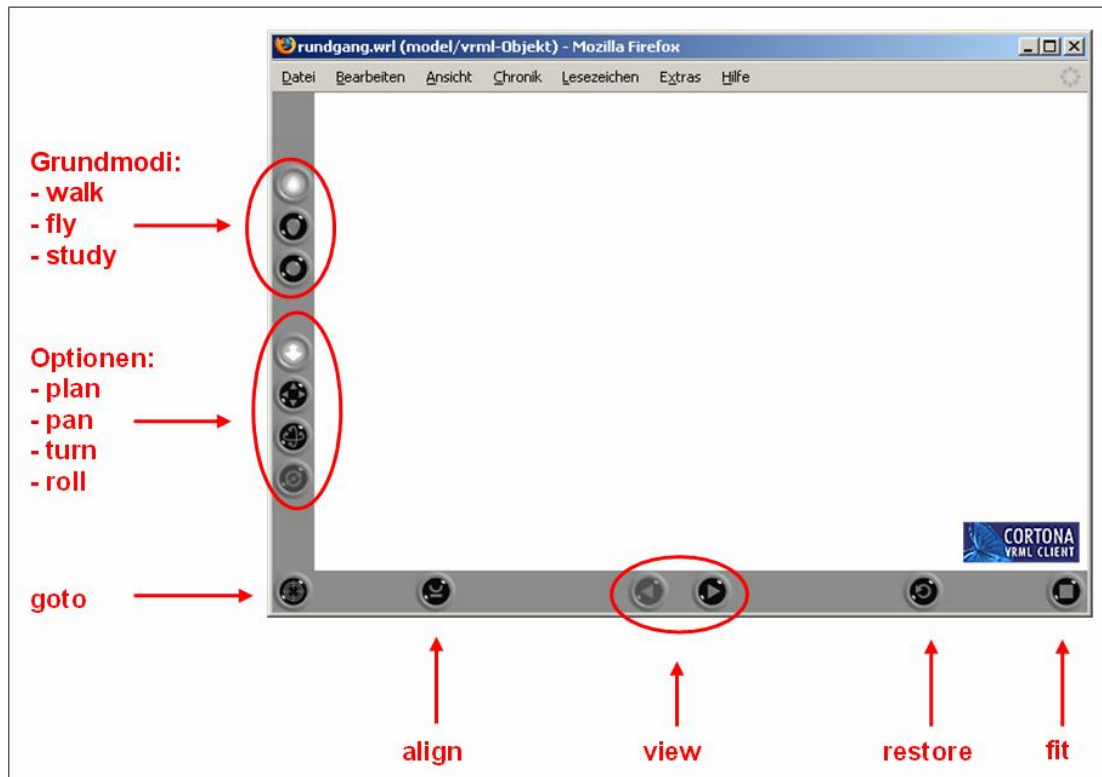


Abbildung 22: Cortona-Player Version 5.1 von ParallelGraphics

A 4 IN SKETCHUP VERWENDETE MATERIALIEN

Tabelle 2 gibt Auskunft über die für die Erstellung der 3D-Modelle verwendeten Materialien. Neben diesen Materialien wurden manche Flächen mit einfarbigen Farbtönen texturiert. Diese Farbtöne wurden direkt in SketchUp erstellt und sind hier nicht aufgeführt.

Es sind nicht sämtliche Materialien bzw. Komponenten in allen Modellen vorhanden.

Materialien¹¹	Beschreibung	Verwendung in den Komponenten	Art der Erstellung
Aussicht	Aussicht aus den Fenstern der Flure sowie des Zimmers G21	Waende	Aufnahme mit Digitalkamera
Badekacheln	Kacheln an den Wänden bzw. am Boden der Toiletten	Waende	Erstellung im Zeichenprogramm Paint
Boden	Spannteppich im G-Stock	Boden	Aufnahme mit Digitalkamera, danach Zuschneiden, bis der übrig bleibende Bereich eine etwa homogene Farbe aufweist
Bodenplatten	im Treppenhaus	Boden	Aufnahme mit Digitalkamera
Decke	Decke im G-Stock	Decke	orthogonale Aufnahme mit Digitalkamera und Zuschneiden auf eine sinnvolle Einheit
D-Stock	Ansichten der Eingangshalle sowie des Haupteinganges	D_Stock, Tuere_D_aussen	Aufnahme mit Digitalkamera

¹¹ Die Bezeichnungen der einzelnen Materialien stimmen nicht mit denjenigen des SketchUp-Dokumentes überein. Da in dieser Tabelle teilweise mehrere Materialien in einer Zeile zusammengefasst werden, wird aus Gründen der Einfachheit und Übersichtlichkeit auf das Auflisten sämtlicher Bezeichnungen verzichtet.

Materialien	Beschreibung	Verwendung in den Komponenten	Art der Erstellung
Fahrstuhl	Aussenansicht des geschlossenen Fahrstuhles	Lift_aussen, Tuere_Lift_links, Tuere_Lift_rechts	Aufnahme mit Digitalkamera
Glasscheiben	transparente Scheiben mit Gitternetz	Waende, Tuere_D_aussen, Tuere_D_innen, Tuer_Treppenhaus1, Tuer_Treppenhaus2	Material ist standardmässig in Google SketchUp vorhanden
Himmel	„Aussicht“ aus den Büros des IKA (ausser G21)	Waende	Material ist standardmässig in Google SketchUp vorhanden
Installation „Atlas der Schweiz“	Aussenansichten der beiden Seiten	AdS	Aufnahme mit Digitalkamera
Kopierer	Kopierer beim Büro G 14.1	Kopierer	grosses Modell: Import aus 3D-Galerie; kleine Modelle: Aufnahme mit Digitalkamera
Lavabo	Ansicht des ganzen Bereiches beim Lavabo	Waende	Aufnahme mit Digitalkamera
Pflanzen	zwei Pflanzen beim Raum G 25.2	Pflanze_gross, Pflanze_klein	Import aus der 3D-Galerie
Poster	gerahmte und ungerahmte Poster	Waende, Poster_AdS1, Poster_AdS2, Poster_eLML, Poster_Intergraph	Aufnahme mit Digitalkamera oder bereits digital vorhanden
Schrank Gestaltungskonzept	grünes Relief	Schrank	bereits digital vorhanden

Materialien	Beschreibung	Verwendung in den Komponenten	Art der Erstellung
Schränke	Vorderansichten der braunen und grauen Schränke	GUV, Schrank, Schrank_G21_gross, Schrank_G21_mittel, Schrank_G21_visavis, Schrank_gross, Schrank_Kopierer	Aufnahme mit Digitalkamera; braune Schränke: zusätzlich Retouchierung mit Paint (Überdecken der Schranknummer)
Sitzecke	Sessel und Tischchen	Sessel, Tischchen	grosses Modell: Import aus 3D-Galerie; kleine Modelle: selbst erstellt und mit einfarbigem Farbton texturiert
Türen (innerhalb des IKA)	Vorderansichten von braunen und weissen Türen, mit Türfalle auf linker oder rechter Seite bzw. ohne Türfalle, in verschiedenen Breiten	Tuere_braun_110, Tuere_braun_64, Tuere_braun_65, Tuere_braun_75, Tuere_braun_85, Tuere_braun_88, Tuere_braun_90, Tuere_weiss_103, Tuere_weiss_60, Tuere_weiss_88, Tuere_weiss_90, Tuerflaeche_links_64, Tuerflaeche_links_90, Tuerflaeche_rechts_64, Tuerflaeche_rechts_90	Aufnahme mit Digitalkamera
Türen (zum G15 sowie zu den Treppen)	Ansichten von den Fluren des IKA aus, für jede Tür separat	Waende, Tuer_Treppenhaus1, Tuer_Treppenhaus2	Aufnahme mit Digitalkamera
Türschilder	Namensschilder neben den Bürotüren	Waende	Aufnahme mit Digitalkamera
Türschilder Gestaltungskonzept	Namensschilder neben den Bürotüren	Waende	bereits digital vorhanden
Vitrinen	Bietschhorn-Relief, Infomaterial zum Atlas der Schweiz, Zervreila-Staudamm	AdS_Vitrine, Bietschhorn, Zervreila	Aufnahme mit Digitalkamera

Materialien	Beschreibung	Verwendung in den Komponenten	Art der Erstellung
Wand (rau)	--	Waende	Aufnahme mit Digitalkamera, danach Zuschneiden, bis der übrig bleibende Bereich eine etwa homogene Farbe aufweist; anschliessend Aufhellung des Bildes und Verstärkung des Kontrastes
Wand beim Büro G28.3	--	Waende	Aufnahme mit Digitalkamera

Tabelle 2: In SketchUp verwendete Materialien