

Entwicklung eines webbasierten Designwerkzeugs für Landschaftsarchitekten

Diplomarbeit

zur Erlangung des Grades einer Diplom-Informatikerin
im Studiengang Computervisualistik

vorgelegt von
Theresa Henze

Erstgutachter: Prof. Dr. Stefan Müller
Institut für Computervisualistik, AG Computergraphik
Zweitgutachter: Thorsten Grosch
Institut für Computervisualistik, AG Computergraphik

Koblenz, im Dezember 2006

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

	Ja	Nein
Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.	<input type="checkbox"/>	<input type="checkbox"/>
Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.	<input type="checkbox"/>	<input type="checkbox"/>

.....
(Ort, Datum)

.....
(Unterschrift)

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Ziele	1
1.3	Aufbau der Arbeit	2
2	Grundlagen	3
2.1	Die Arbeit eines Landschaftsarchitekten	3
2.1.1	Programme zur Gartengestaltung	6
2.1.2	Anforderungsanalyse	8
2.2	Technologien	11
2.2.1	Webserver	11
2.2.2	Datenbank	11
2.2.3	Zeichenoberfläche	12
2.2.4	Dreidimensionale Darstellung	14
2.3	X3D	15
2.3.1	Navigation	19
2.3.2	Interaktion mit ECMAScript	20
2.3.3	X3D-Player	22
3	Entwicklung und Implementation	24
3.1	Gelände	24
3.1.1	Webseite	25
3.1.2	Flash	26
3.1.3	X3D	28
3.2	Wege/Plätze	32
3.2.1	Webseite	32
3.2.2	Flash	33
3.2.3	X3D	33
3.3	Objekte: Pflanzen und Gegenstände	37
3.3.1	Webseite	37
3.3.2	Flash	38
3.3.3	X3D	38
3.4	Das Projekt	41
3.5	Was fehlt	41
4	Bewertung	43
4.1	Die Benutzertests	43
4.1.1	Eigenversuch	45
4.1.2	1. Proband	46
4.1.3	2. Proband	47
4.1.4	3. Proband	47
4.1.5	Fragen an die Probanden	49

4.1.6	Ergebnisse der Benutzertests	51
5	Ergebnisse	52
5.1	Ausblick	52
5.1.1	XML als Datenbank	53
5.1.2	Gewässer und Wasserläufe	53
5.1.3	Jahreszeiten	54
5.1.4	Tageszeiten	55
5.1.5	Pflanzenalter	55
5.1.6	Duftkarten	55
5.1.7	Online-Service	55
5.2	Fazit	56
	Anhang A - Datenbank	57
	Anhang B - Auf der CD	61
	Literatur	62

Abbildungsverzeichnis

1	Ein typischer Ausführungsplan für die Gestaltung eines Gartens [1]	4
2	Google SketchUp [2]	7
3	X3D Basis-Profil [3]	16
4	“Hello World“ im Octaga Player	19
5	“Hello World“ im Flux Player	23
6	Beispielkarten mit Höhenlinien (l) und Höhenpunkten (r) . .	24
7	Beispiel einer Heightmap (links) und deren dreidimensionale Ansicht (rechts) aus dem Programm Bryce. [4]	25
8	Die Übersicht zur Geländeerstellung. Man hat die Optionen: neues Gelände erstellen, Arborata Projekt laden, fertiges Gelände laden.	26
9	Zeichnen der Höhenwerte in der Flashapplikation gelaende.swf	27
10	Verzerrung der Wegform bei einem Gelände der Größe 12x4 Meter. Das rote Raster markiert die Quadratmeter.	28
11	ElevationGrid: zeigt die Zusammenhänge von Spacing und Dimension mit den Höhenwerten.	29
12	Das entstandene Gelände im X3D-Browser Octaga.	31
13	Auflistung aller platzierten Wege mit ihrer Textur.	33
14	Die Flashapplikation flaechen.swf, wobei auf das Höhenbild des Geländes die Form des Weges gemalt wird.	34
15	Die Wegtextur liegt 10 cm über dem eigentlichen Gelände .	35
16	Vorschau des Gartens mit den gewählten Wegen.	36
17	Auflistung aller gewählten und platzierten Pflanzen. Eine ähnliche Übersicht gibt es auch für die Gegenstände.	37
18	Die Platzierung der Pflanzen werden mit einem grünen Punkt auf der Höhenkarte des Geländes markiert. Gegenstände erhalten ein blaues Quadrat.	38
19	Beispiel für ein dreidimensionales Modell eines Baums: Texturierte Darstellung (links oben), Gitternetz (rechts oben), Drahtgitter mit verdeckten Kanten einmal von vorne und vorn oben (unten)	39
20	Das Gelände mit Wegen und den platzierten Pflanzen. . . .	40
21	Eine Ansicht auf das fertige Gelände mit Wegen, Pflanzen und Gegenständen.	41
22	1. Referenzbild [5]	44
23	2. Referenzbild [5]	44
24	Ergebnis nach dem Erstellen des Geländes und dem Platzieren alle Objekte	45
25	Ergebnis nach dem Versuch, die Mauern nachzubilden . . .	45
26	Ansichten des von Proband 1 erstellten Gartens	46

27	Ansichten des von Proband 2 erstellten Gartens	47
28	Ansichten des von Proband 3 erstellten Gartens	48
29	von links nach rechts: Frühling, Sommer Herbst, Winter. Aus einem Eifelpanorama von Steffen Heinz	54

Tabellenverzeichnis

1	Beispielwerte für die Auflösung in Flash	27
2	Beispielwerte für die Auflösung des Geländes	30
3	Größe und Speicherbedarf einiger quadratischer Beispielge- lände	30

1 Einleitung

In Zusammenarbeit mit einer angehenden Landschaftsarchitektin der Hochschule Anhalt¹ entstand die Idee von einer Webapplikation, der die Zusammenarbeit von Gartendesignern mit ihren Kunden vereinfacht. Der Versuch, dass der Kunde selbst seine Gartenwünsche gestalten und diese auch dreidimensional betrachten kann, wurde bisher nicht zufriedenstellend umgesetzt. Gartendesignprogramme für den Laien gibt es zwar genug, aber die gewonnenen Informationen liegen meist in proprietären Dateiformaten vor und können vom Architekten nicht weiter genutzt werden.

Im Rahmen dieser Diplomarbeit ist nun ein Prototyp für ein Programm entstanden, womit der Kunde selbst Konzepte für seinen Garten entwickeln und durch die gestaltete dreidimensionale Landschaft navigieren kann. Die so erhaltenen Daten werden als ISO-Standard vorliegen und der Architekt wird sie auch für weitere Entwürfe nutzen können.

1.1 Motivation

Obwohl es viele Gartenbesitzer und auch zahlreiche Gartenratgeber gibt, ob in Buchform oder in TV und Computer-Software, so ist die Hemmschwelle doch recht hoch für den eigenen Garten einen Architekten zu beauftragen. Vor allem der Kostenfaktor schreckt mögliche Interessenten davon ab, einen Gartenarchitekten aufzusuchen und/oder ein Computerprogramm (oft ab 30 Euro) zu kaufen, da ja noch die Kosten für die baulichen Maßnahmen und die Bepflanzung hinzu kommen. Viele Gartenbesitzer werden so wahrscheinlich von größeren Projekten absehen. Dies ist besonders an der aktuellen Popularität von Designer- und Heimwerker-sendungen zu erkennen.

Tatsächlich ändern könnte das ein kostenloses Programm, welches durch Einfachheit überzeugt und das Ausprobieren von Ideen ermöglicht. Das Wichtigste an so einem Programm aber wäre die Möglichkeit alle Versuche auch dreidimensional betrachten zu können, um wirklich eine Entscheidung fällen zu können.

1.2 Ziele

Es ist das Ziel dieser Arbeit, zu prüfen, ob man ein Designwerkzeug für Landschaftsarchitekten und auch Laien erstellen kann, welches eine intuitive Erstellung von einem Gelände ermöglicht, worauf Wege und Pflanzen, sowie beliebige weitere Objekte platziert werden können. Dabei ist eine dreidimensionale Darstellung der Auswahl von großer Bedeutung, damit

¹<http://www.hs-anhalt.de/>

sich Benutzer einen realistischen Eindruck von ihrem Traumgarten machen können.

Um auch die Aktualität der Daten zu gewährleisten und hohe Flexibilität zu ermöglichen, sollte das Programm später auch um weitere Funktionen erweiterbar sein und trotzdem weiterhin auf allen Plattformen laufen. Idealerweise ist hier eine webbasierte Lösung anzustreben, was den Kauf eines Programmes unnötig macht.

Es soll gezeigt werden, ob die gesetzten Ziele auch tatsächlich in diesem Rahmen umsetzbar sind und welche Funktionen mit den gewählten Technologien nicht ermöglicht werden können.

1.3 Aufbau der Arbeit

Das in dieser Diplomarbeit entstandene Programm *Arborata* ist nach dem lateinischen Wort für Baum *arbor* benannt. Im weiteren Verlauf wird auch mit diesem Namen auf das Programm verwiesen.

Das 2. Kapitel beschreibt die bisherige Arbeit eines Landschaftsarchitekten und wie dieses Projekt helfen kann, nicht nur die Arbeit zu beschleunigen, sondern auch Missverständnisse mit dem Kunden zu vermeiden. Dabei wird auch auf Gartendesignprogramme eingegangen, die derzeit erhältlich sind und welche Technologien es zur Umsetzung dieses Projekts gibt. Eine Anforderungsanalyse zeigt dabei die Herausforderungen des Programms.

In Kapitel 3 werden die einzelnen Komponenten der Implementierung erklärt und wie diese funktionieren. Dabei wird auf die verwendeten Technologien mit ihren Vor- und Nachteilen eingegangen.

Zum Test der Applikation wird in Kapitel 4 der Nachbau eines Gartenprojekts beschrieben und mit einigen Abbildungen untermauert. In einem Benutzertest werden die Stärken und Schwächen des Programms aufgezeigt. Ausserdem wird auf die Anforderungsanalyse eingegangen, welche der gesetzten Ziele umgesetzt werden konnten.

Das letzte Kapitel ist eine Zusammenfassung der Ergebnisse dieser Diplomarbeit. Es werden Möglichkeiten gezeigt, wie das Programm erweitert werden kann und was mit den vorgestellten Technologien noch machbar ist.

2 Grundlagen

Um den Nutzen des Programms deutlich zu machen, wird kurz die Arbeitsweise eines Garten- oder Landschaftsarchitekten vorgestellt. Es soll vor allem gezeigt werden, welche Programme derzeit benutzt werden und welche Daten die Architekten mit ihren Kunden austauschen.

2.1 Die Arbeit eines Landschaftsarchitekten

Jeder Garten- und Landschaftsarchitekt verfolgt unterschiedliche Methoden, um Entwürfe für Kunden zu erstellen. Dabei haben sie aber meist folgenden Arbeitsablauf gemein:

1. Der Kunde erklärt dem Landschaftsarchitekt seine Gartenwünsche
2. Der Landschaftsarchitekt macht sich ein Bild von dem Gelände (Plan oder eigene Vermessungen)
3. Der Landschaftsarchitekt erstellt mehrere Entwürfe (mindestens drei)
4. Durch Rücksprache werden die Entwürfe verfeinert, wobei weitere Kundenwünsche aufgenommen werden können.
5. Kunde entscheidet, welcher Entwurf realisiert werden soll
6. Landschaftsarchitekt wählt die genaue Pflanzenliste aus
7. Baumschulen und/oder Gärtnereien werden mit Aufbau beauftragt

Bis zu Punkt 4 weiß der Kunde nicht, wie viel ihn der Garten kosten könnte. Je unsicherer der Kunde mit seinen Wünschen ist, desto öfter wird Punkt 4 und 5 wiederholt. Die Dauer dieses Vorgangs hängt davon ab, wie gut der Landschaftsarchitekt die Wünsche und Bedürfnisse seines Kunden erkennt.

Arborata soll diese Problematik überbrücken und es auch Nicht-Architekten ermöglichen einen Garten zu erstellen, dafür muss es übersichtlich und möglichst selbsterklärend sein. Wichtig wäre eine Möglichkeit sich, wie in den bisherigen Programmen, ein Bild von der erstellten Auswahl zu machen, also auch eine dreidimensionale Darstellung anzubieten. Im Gegensatz zu bereits erhältlichen Gartendesignern wird nicht nur auf die dreidimensionale Darstellung Wert gelegt, sondern ebenso auf eine Pflanzendatenbank, die nur tatsächlich käufliche Pflanzen anbietet.

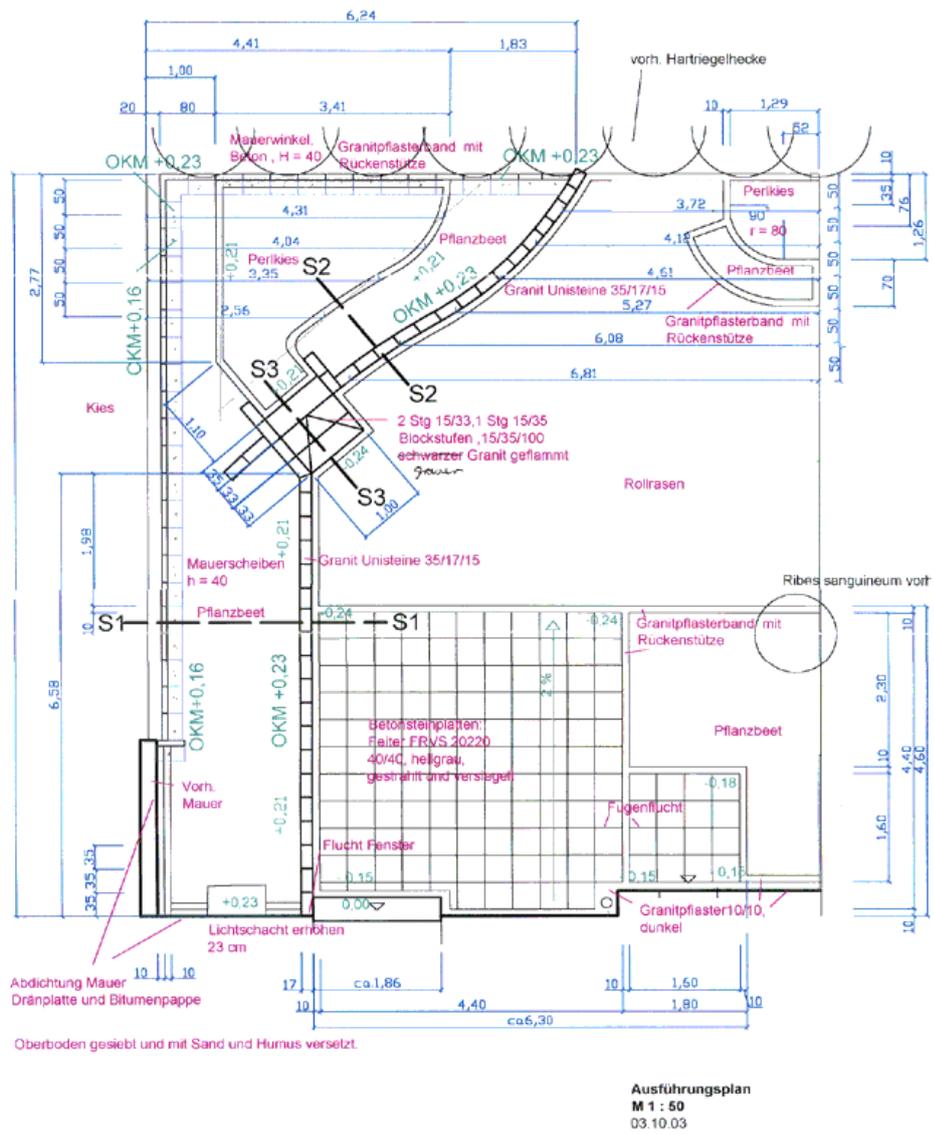


Abbildung 1: Ein typischer Ausführungsplan für die Gestaltung eines Gartens [1]

Das System sollte auch ein für den Architekten angepasste Übersicht bieten: Welche Pflanzen sollen in den Garten; Welche Geländeformation ist vorhanden oder soll erstellt werden; Welche Objekte (Lampen/Dekoration/etc.) möchte der Kunde. Diese Informationen bieten dem Architekten schon wichtige Hinweise, worauf der Kunde hinaus möchte und welchen Geschmack er treffen sollte.

Aus der Sicht des Kunden hingegen wird es möglich eine Kosteneinschätzung zu treffen. Man kann beispielsweise bereits einschätzen, wie viel Quadratmeter Kies für den Weg benötigt werden oder wie viel eine zwei Meter große Blau-Tanne kostet. Dabei soll das Angebot der Pflanzen und Objekte stets realistisch bleiben: Zu jedem Objekt soll auch bekannt sein, wer das Objekt herstellt oder die Pflanzen züchtet. Also nur tatsächlich in Baumärkten und Baumschulen erhältliche Produkte.

Ein Programm auf Basis von X3D (siehe Kapitel 2.3) kann diese Anforderungen bieten, da bei jeder Vorschau eine X3D-Datei erstellt wird, die schon alle nötigen Informationen für den Landschaftsarchitekten beinhaltet: Geländeinformationen sind darin ebenso gespeichert, wie Aussehen der Wege und Plätze und welche Pflanzen benutzt werden. Alles was der Landschaftsarchitekt für weitere Entwürfe benötigt, ist also schon durch die X3D Datei gegeben.

Nun könnte der Weg zum Gartenentwurf so aussehen:

1. Der Kunde erstellt nach eigenen Wünschen und Vorstellungen einen Gartenentwurf mit Hilfe von Arborata
2. Der Kunde speichert die X3D-Datei aus der Vorschau und bespricht diese mit dem Landschaftsarchitekten
3. Zusammen können beide Änderungen an dem Kundenentwurf machen
4. Der Landschaftsarchitekt erstellt einen fertigen Bepflanzungsplan zur Realisierung
5. Baumschulen und/oder Gärtnereien werden mit Aufbau beauftragt

Obwohl Arborata ursprünglich nicht dazu gedacht war, den Landschaftsarchitekten zu ersetzen, wird bei dem gezeigten Arbeitsablauf klar, dass auch der Gartenbesitzer selbst die erforderlichen Arbeiten (Kauf und Setzen der gewünschten Produkte) erledigen kann, ohne einen Architekten beauftragen zu müssen. Allerdings sollte dabei beachtet werden, dass gerade bei größerem baulichen Maßnahmen, wie Teiche ausheben o.ä., die entsprechenden Kenntnisse vorhanden sein sollten. Ausserdem kann das Programm keinesfalls die Erfahrungswerte und ästhetisches Verständnis eines Architekten widerspiegeln.

2.1.1 Programme zur Gartengestaltung

Es gibt zahlreiche Programme, die als Gartendesigner verkauft werden. Die Zielgruppe dieser Programme sind vor allem Heimgärtner mit wenig Erfahrungen zur Gartengestaltung und Pflege. Dabei kann man zwischen zwei Gruppen von Programmen unterscheiden, einmal mit umfangreicher Pflanzendatenbank und zum anderen Applikationen, die dreidimensionale Ansichten ermöglichen. Beides in einem Programm zu finden ist dagegen sehr selten.

Programme mit dreidimensionaler Ansicht: Dreidimensionale Darstellungen von Pflanzen werden im Allgemeinen nicht speziell für die jeweilige Pflanze erstellt, sondern nur beispielhaft als Baum, Busch oder Blume implementiert, so dass Charakteristika, wie Farbe, Duft und Lichtempfindlichkeit nicht weiter verarbeitet werden können. Die entstandenen Daten haben dadurch keinen Bezug zu den tatsächlich gewünschten Einstellungen des Benutzers.

Ein anderes Problem ist die Geländeerstellung, die über das Zeichnen der Umrisse und das Setzen einzelner Höhenpunkte funktioniert. Für ungeübte Benutzer ist diese Arbeitsweise leider schwer zu verstehen und die benötigten Höhenwerte sind für den eigenen Garten oft nicht bekannt.

Die meisten 3D-Programme haben aber auch den Vorteil, dass die Möglichkeit besteht das eigene Haus zu konstruieren, so dass beispielsweise eine realistische Schattenberechnung durchgeführt werden kann.

Getestet wurde das aktuelle Programm *3D Garten* von RTL2, welches durch die gute Bewertung auffiel. Das Gelände wird hier mit Hilfe des Setzens einzelner Höhenpunkte generiert, was jedoch nicht intuitiv ist, um schnell ein Gelände zu erstellen. Das Programm umfasst Möglichkeiten Häuser, Wintergärten und Teiche zu konstruieren. Sehr praktisch dagegen ist eine Renderansicht, die auch den Schattenwurf zu beliebigen Tageszeiten darstellen kann.

Programme mit Pflanzendatenbank bieten einen großen Umfang an Pflanzen, die in Deutschland erhältlich sind, zusammen mit wichtigen Charakteristika, wie Wuchseigenschaften und Pflegehinweise sowie typischen Bildern. Einige sind auch auf bestimmte Gebiete spezialisiert und umfassen beispielsweise nur Nadelgehölze oder nur Rosengewächse. Heute gibt es zu diesen Pflanzendatenbanken auch Designwerkzeuge, die bei der Planung helfen sollen, und Hinweise auf Standortprobleme (z.B. Lichtmangel oder zu nah am nächsten Baum) geben.

Trotz der Planungshilfen hat der Benutzer hier keine Möglichkeit sich einen realistischen Eindruck seiner Entwürfe zu machen, da eine dreidimensionale Darstellung nicht verfügbar ist.

Programme für Landschafts- oder Gartenarchitekten: Für professionelle Gartengestaltung werden hauptsächlich Vektorprogramme genutzt, wie beispielsweise Adobe Illustrator². Aus diesen Programmen entstehen zweidimensionale Ansichten, die zur Planung der Bebauung/Bepflanzung verwendet werden. Auf Wunsch des Kunden können aber auch dreidimensionale Darstellungen erzeugt werden, indem Höhenlinien aus 2D-Plan in CAD-Anwendungen, wie Auto-CAD³, in die gewünschten Höhen gezogen werden und eine Fläche darüber erzeugt wird. Dann werden vorgefertigte Modelle für Pflanzen und Gegenstände auf das Terrain gesetzt und die Daten gerendert. Leider erfordern derartige Programme eine entsprechende Ausbildung oder viel Übung und sie stellen oft keine speziellen Funktionen für Garten- und Landschaftsbau zur Verfügung. Einige der 3D-Programme bieten allerdings kostenpflichtige Bibliotheken an, die fertige Modelle von verschiedenen Pflanzen und Objekten enthalten.

Ein Beispiel für die direkte Modellierung in einer dreidimensionalen Ansicht stellt Google SketchUp⁴ dar. Es ist ein einfach zu erlernendes 3D-Modellierprogramm mit Echtzeitschattenwurf [2]. Gerade bei diesem Programm, wofür auch eine kostenlose Version existiert, ist jedoch keine Gartenmodellierung vorgesehen. Wäre eine solche Software jedoch OpenSource würde sich eine Entwicklung von Arborata auf Basis von SketchUp wirklich lohnen, da das Programm einfach zu erlernen ist und die Grundwerkzeuge für die Arbeit im dreidimensionalen Raum bietet. Die kostenlose Version erlaubt nur den Export in das eigene Datenformat.

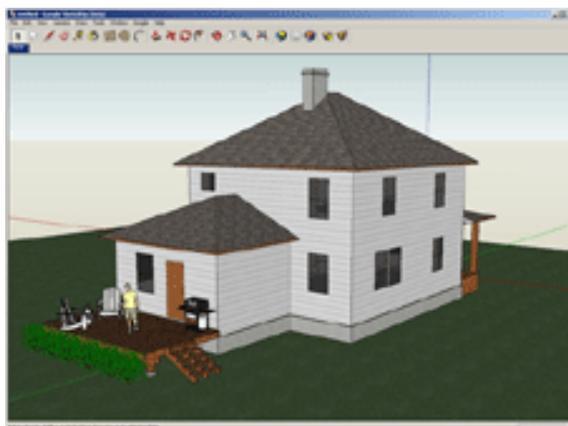


Abbildung 2: Google SketchUp [2]

²<http://www.adobe.com/de/products/illustrator/>

³<http://www.autodesk.de/>

⁴<http://sketchup.google.com/>

Eine weitere 3D-Modellierungssoftware, die hier Erwähnung finden sollte, ist Quest 3D⁵. Auch wenn es nur sehr selten von Garten- oder Landschaftsarchitekten genutzt wird, bietet es eine Physiksimulation in Echtzeit und Datenbankanbindung [6]. Auf Grund der hohen Kosten ist es für das Arborata-Projekt jedoch ungeeignet.

2.1.2 Anforderungsanalyse

Alle folgenden Anforderungen entstanden auf Basis verschiedener Gespräche mit angehenden Landschaftsarchitekten, aber auch interessierten Laien. Dabei stand die einfache Bedienbarkeit und Umsetzbarkeit im Vordergrund, dennoch wurden auch Gedanken zu schwierigeren Themen zusammengetragen. Die Ergebnisse der Gespräche werden im Folgenden zusammengefasst.

Die benötigten Funktionen haben unterschiedliche Prioritäten. So ist zum Beispiel das Erstellen des Geländes von größter Wichtigkeit, weil alle anderen Funktionen des Programms davon abhängen, da man ohne Gelände weder Wege noch Objekte platzieren kann. Es wird zwischen 3 Prioritätsklassen unterschieden:

- **Priorität 1:** Grundfunktionen des Programms, die umgesetzt werden müssen, um eine lauffähige Applikation zu erhalten.
- **Priorität 2:** Funktionen, deren Implementierung dem Benutzer zwar mehr Möglichkeiten eröffnen, aber deren Fehlen nicht die Funktionalität des Programms beeinträchtigt.
- **Priorität 3:** Umfangreichere Problemstellungen, deren Umsetzung nicht mehr in dieser Diplomarbeit vorgesehen sind, die aber für eine eventuelle Weiterentwicklung des Programms wichtig sind.

Gelände: Um die Pflanzen/Gegenstände zu platzieren und auch einen echten Eindruck ihrer Wirkung geben zu können, sollte der Anwender sein eigenes Gartenterrain möglichst realitätsnah nachbilden können. Mit Hilfe einer Heightmap (siehe Kapitel 3.1) soll der Benutzer die Geländehöhen "malen" können.

- Erstellen (Priorität 1) der Heightmap in gewünschter Größe in Flash.
- Bearbeiten (Priorität 2) der Geländehöhen, aber nicht der Geländegröße.
- Löschen (Priorität 1) des Terrains und aller vorhandenen Positionen der Pflanzen/Gegenstände und der Wege/Plätze.

⁵<http://www.quest3d.com/>

Wege/Plätze: Ein Gelände besteht nicht nur aus einer Rasenfläche, sondern auch aus verschiedenen Oberflächen, die sich in Aussehen und Material unterscheiden. Der Benutzer soll diese Oberflächen beliebig auf das Terrain legen können.

- Erstellen (Priorität 1) eines oder mehrerer Wege oder Plätze.
- Bearbeiten (Priorität 2) der Lage auf dem Gelände.
- Löschen (Priorität 1) löscht den erstellten Weg bzw. Platz.

Objekte: Als Objekte fasse ich Pflanzen und Gegenstände zusammen, beide haben die Funktionen Suche, Platzieren und Löschen gemeinsam. Unter Gegenständen sind nicht-lebende Elemente, wie beispielsweise Parkbänke oder Zierbrunnen zu verstehen. Da Pflanzen aber ganz andere Eigenschaften besitzen werden sie in der Arbeit oft getrennt von den Gegenständen betrachtet.

- Suche (Priorität 1) nach Objekten über den Namen oder Volltext. Für Pflanzen steht zusätzlich die Suche über den deutschen und lateinischen Namen, sowie der Suche über Eingabe von beliebigen Pflanzeigenschaften zur Verfügung.
- Merktzettel (Priorität 1) zum Merken von Pflanzen, auch wenn sie nicht im Gelände platziert sind.
- Platzieren (Priorität 1) der Objekte auf einer 2D-Heightmap des Geländes.
- 2D-Darstellung (Priorität 1) der Pflanzen in der X3D-Vorschau als Billboard. Gegenstände dagegen sollen immer dreidimensional dargestellt werden.
- 3D-Darstellung (Priorität 2) der Pflanzen in der X3D-Vorschau.
- Löschen (Priorität 1) der einzelnen Objekte aus dem Gelände oder vom Merktzettel.

Gewässer: Ein weiteres Element in Gärten oder Landschaften sind Gewässer. Dazu gehören glatte Wasserflächen, wie Teiche oder Pools, aber auch Wasserläufe und -fälle.

- Wasserfläche (Priorität 2) für Pools oder Teiche.
- Bach (Priorität 3) für alle fließenden Gewässer.
- Wasserfall (Priorität 3)

Treppen: (Priorität 3) Eine Implementierung von Treppen könnte sehr schwierig werden, auf Grund der vielen Möglichkeiten beim Treppenbau, die von einer einläufig geraden Treppe bis hin zu mehrläufigen, gewendelten Treppen reichen⁶. Diese verschiedenen Treppenformen unterscheiden sich teilweise sehr stark von einander, auch in ihrer Bauweise.

Es wäre dennoch sinnvoll, Treppen modellhaft als einläufige gerade Treppe zu implementieren, da viele Gärten mindestens ein paar Stufen beinhalten und der Benutzer einen realistischeren Eindruck von seinem Garten bekommt.

Bodenbeschaffenheit: Nicht jedes Gelände hat eine gleichmäßige Rasendecke und es werden normalerweise Beete gelegt, in denen die Pflanzen wachsen. Die Oberfläche des Terrains kann aus Fels, Sand oder Kies beschaffen sein, aber auch aus Pflanzen, wie Moose, Efeu oder Wacholder⁷. Also wird eine Funktion benötigt, die unterschiedliche Bodenarten darstellen kann, sowie eine Flächenbepflanzung.

Diese Anforderung ist sehr ähnlich mit dem Wege/Plätze-Problem und kann analog umgesetzt werden. Da die Flächenbepflanzung allerdings dreidimensional ist und nicht wie die Wege mit einer Textur dargestellt werden kann, ist zu sehen, ob diese Funktion nicht anders implementiert werden muss.

- Bodenarten (Priorität 2) zur Darstellung unterschiedlicher Bodenbeschaffenheit.
- Flächenbepflanzung (Priorität 2) zur Realisierung von flächigem Pflanzenwuchs.

Steingärten: (Priorität 3) Eine Unterhaltung mit einem Landschaftsarchitekten über die Umsetzung eines aktuellen Projekts zeigte eine weitere Anforderung an das Programm: Steingärten. Es handelt sich dabei um Anpflanzungen auf Böden, die je nach Art mehr oder weniger Steinanteil enthalten. Das kann von einer Fläche mit einzeln gesetzten Steinen bis zu einer Mauer mit bepflanzten Fugen reichen⁸.

Die Umsetzung von einzelnen Steinen kann im Rahmen von platzierbaren Objekten erfolgen, die auf das Gelände gesetzt werden. Mauern oder alpine Geröllhänge dagegen würden eine extra Funktion benötigen, die nicht mehr im Rahmen der Diplomarbeit umgesetzt wird.

⁶Eine Auflistung der Treppenformen finden sie unter: <http://www.bauwerkverlag.de/baulexikon/index.shtml?TREPPENFORMEN.HTM>

⁷Einige Beispiele für sogenannte Bodendecker finden sie unter: http://www.gartenatelier.de/Pflanzen/bodendecker_pflanzen.htm

⁸Beispiele zum Thema Steingarten: <http://www.derkleinegarten.de/steingarten.htm>

2.2 Technologien

In diesem Kapitel werden die Technologien beschrieben, die für Arborata benutzt wurden.

2.2.1 Webserver

Für Arborata wird der Apache Webserver⁹, PHP 4¹⁰ und das Datenbankmanagementsystem (DBMS) MySQL 5.0¹¹ genutzt, alles aus XAMPP 1.5¹².

- Apache 2.2.0
- MySQL 5.0.18
- PHP 5.1.1, PHP 4.4.1-pl1 und PEAR

2.2.2 Datenbank

Zur Verwaltung der Datenbestände und ihrer Zusammenhänge (z.B. Pflanzen und ihre Position auf dem Gelände) wird eine relationale Datenbank benutzt. Es gibt sehr verschiedene Datenbankmanagementsysteme (DBMS) mit unterschiedlichen Vor- und Nachteilen. Da ein freies DBMS für diese Arbeit benötigt wird, wurde das OpenSource DBMS MySQL 5.0 benutzt. Zum einen, da alle zur Verfügung stehenden Testserver dieses System unterstützen und zum anderen weil die SQL-Konformität von MySQL recht hoch ist und damit die spätere Lesbarkeit und Verständlichkeit des Projekts erhöht.

Zur Administration der Datenbank wurde MySQL-Front¹³ benutzt, da es das Erstellen der Tabellen vereinfacht und dadurch beschleunigt.

Hier eine kurze Beschreibung zu den einzelnen Tabellen:

- **plants** enthält alle Pflanzen und ihre Eigenschaften, sowie ihre Notation in X3D.
- **items** umfasst die Gegenstände und ihre Größenangaben und die X3D-Notation.
- **surfaces**: Oberflächenmaterialien für Wege und Plätze.
- **themes** beschreibt Themenkomplexe, die als Einschränkungen auf die Objektsuche benutzt werden können.

⁹<http://www.apache.org/>

¹⁰<http://www.php.net/>

¹¹<http://www.mysql.com/>

¹²<http://www.apachefriends.org/xampp.html>

¹³<http://www.mysql-front.de/>

- **themes.objects** ordnet den Objekten und Oberflächen ein oder mehrere Themen zu.
- **colors** enthält Farbgruppen zur Beschreibung der Pflanzen.
- **item_type**: Objektgruppen für die Gegenstände.
- **default_terrains** beinhaltet vorgefertigte Gelände.
- **session** speichert das Gelände und die Geländegröße für den jeweiligen Benutzer.
- **session.objects** enthält pro Benutzer alle ausgewählten Objekte, wie Gegenstände und Pflanzen und ihre Positionen auf dem Gelände.
- **session_surfaces** definiert die Wege und Plätze eines Geländes.

`plants` beinhaltet sämtliche Pflanzeigenschaften, die zum größten Teil aus dem BRUNS Pflanzen¹⁴ Katalog 2005/6 übernommen worden, einer Bestellsoftware für Garten- und Landschaftsbau, Landschaftsarchitekten, Grünflächenämter und Gartencenter. Der Katalog umfasst 6.800 Pflanzen mit 350 möglichen Eigenschaften.

Die Tabellen `themes` und `themes.objects` sind zwar schon vorbereitet, ihre Funktion jedoch nicht mehr in Arborata implementiert worden. Eine vollständige Liste der Tabellenfelder und wofür sie benutzt werden, befindet sich in Anhang A.

2.2.3 Zeichenoberfläche

Arborata benötigt eine Zeichenoberfläche für die Geländeerstellung. Das Terrain soll auf einer zweidimensionalen Draufsicht als Heightmap (siehe Kapitel 3.1) erstellt werden. Um das Malen/Zeichnen zu ermöglichen, wurden die folgenden Technologien betrachtet:

Flash von Adobe¹⁵ (*früher von Macromedia*) ist ein Programm zur Erstellung vektorbasierter Grafiken und Animationen. Mit dem Autorenwerkzeug für Flash können Animationen erzeugt werden, die als SWF-Datei auf Webseiten mit dem Plugin Flash Player abgespielt werden können. Eine Kombination von Animation und Programmierung mit Hilfe von ActionScript erlaubt es komplexe Inhalte auf Webseiten anschaulich darzustellen. [7]

Flash eignet sich besonders für Benutzeroberflächen, die auf Webseiten eingebunden werden, da alle Elemente verlustfrei auf beliebige

¹⁴<http://www.bruns.de/>

¹⁵<http://www.adobe.com>

Größen skaliert werden können. Zwar kennen die meisten Internetnutzer nur die Flashwerbeposter, allerdings wissen auch Spieleprogrammierer den Funktionsreichtum von Flash und ActionScript zu schätzen. Flash hat sich im Webbereich zur Standardlösung für Animation und Interaktion entwickelt, so dass für die gebräuchlichsten Internetbrowser auch Plugins existieren. [8]

Scalable Vector Graphics (SVG) ist ein XML-Format zur Definition von zweidimensionalen Vektorgrafiken. Auch Animationen lassen sich erzeugen mit der Synchronized Multimedia Integration Language (SMIL). Da es rein für die Darstellung gedacht ist, unterstützt es jedoch keine Interaktion. SVG-Grafiken können auch nicht rasterisiert (pixelweise) ausgelesen werden und sind für Arborata ungeeignet.

Windows Presentation Foundation

Windows Presentation Foundation (WPF) ist ein Framework zur Ausgabe von 2D-/3D-Grafiken, Video, Audio und Bildern. Im Gegensatz zu seinen Vorgängern GDI und GDI+ setzt WPF auf DirectX auf und bietet dadurch volle Hardwarebeschleunigung. Effekte, die bisher vor allem in Spielen vorkamen, können mit WPF einfach auf den Desktop gebracht werden. Dabei erkennt WPF automatisch, welche Performance das System bietet und passt die Effektivität der Anwendung an die gegebenen Ressourcen an. Schatten-, Blur-, Glow und Transparenzeffekte werden, wenn möglich, direkt von der Grafikkarte gerendert. [9]

Jedoch ist WPF nicht für Webanwendungen gedacht und auf Grund der Abhängigkeit von DirectX auch plattformabhängig.

Eine sehr funktionsreiche Zeichenoberfläche bietet der **Oekaki Shi-Painter**¹⁶, ein JavaApplet, welches zahlreiche Optionen zur Verfügung stellt, wie z.B. verschiedene Pinselstärken und Bezierkurven. Die gezeichneten Bilder können in den Grafikformaten PNG, JPG gespeichert werden. [10]

Der Oekaki Shi-Painter ist zwar sehr umfangreich und sehr bequem zu nutzen, allerdings müssen alle erstellten Grafiken erst als Bilder auf den Server gelegt werden und können erst hinterher in die Datenbank gelesen werden. Leider ist auch hier kein Airbrush-Pinsel implementiert, um weiche Geländeübergänge zu erzeugen.

Für Arborata wurde eine Umsetzung mit Flash gewählt, da es an die speziellen Ansprüche der Diplomarbeit angepasst werden kann und - bis

¹⁶Ein Beispiel zum Testen ist unter folgender URI zu finden: <http://mchan.vanillae.de/cgi-bin/draw/bbsnote.pl?fc=paint.sii&anime=true&width=300&height=300>

auf einen Airbrush-Pinsel - alle benötigten Funktionen programmiert werden konnten.

2.2.4 Dreidimensionale Darstellung

Das Projekt benötigt eine dreidimensionale Darstellung des Gartens, die mit Hilfe von Texturen einen groben Eindruck von der Wirkung aller benutzten Elemente zulässt. Eine ideale Lösung hätte selbst in der webbasierten Programmierung kein Plugin benötigt, allerdings gibt es nur wenige Möglichkeiten eine Echtzeitdarstellung auf Clientrechnern zu erzeugen, ohne ein extra Programm installieren zu müssen.

Die 3D-Ansicht mit **JavaScript** (ECMAScript) wäre zwar vorteilhaft, weil die meisten Browser JavaScript bereits unterstützen und keine zusätzlichen Applikationen nötig sind, jedoch ist die Darstellung von Materialien/Texturen und die Beleuchtung der Modelle nicht ohne weiteres möglich. Da keine Rechenoperationen von der Grafik-Hardware übernommen werden, belastet dies den Web-Client sehr¹⁷.

Java3D ist eine Klassenbibliothek zur Erzeugung, Manipulation und Darstellung dreidimensionaler Grafiken auf Basis von OpenGL oder DirectX [11]. Mit sogenannten Java Applets können auch webbasierte Anwendungen programmiert werden. Jedoch hätte ein kompletter 3D-Browser mit entsprechendem Framework geschrieben werden müssen, um Java3D nutzen zu können. Das Nachprogrammieren der Funktionalitäten vorhandener 3D-Browser hätte den Rahmen der Diplomarbeit gesprengt.

Adobe Flash kann zwar zur Darstellung dreidimensionaler Objekte benutzt werden, beherrscht dies jedoch nicht von sich aus. Mit Hilfe von ActionScript-Klassen kann aber eine 3D-Umgebung berechnet werden. Es gibt auch Autorenprogramme, die Flash für 3D-Darstellung nutzen. Aber auch hier würde der CPU des Clientrechners zu stark belastet werden, wenn ein Beleuchtungsmodell, Kollisionserkennung, u.s.w. hinzukommen.

Auch von Adobe ist **Shockwave**, das Internetformat von Director, welches auf Grund von sehr hoher Kompression und dynamischem Nachladen der Inhalte (Streaming) auch umfangreiche Applikationen zulässt. Unter Shockwave werden derzeit testweise 3D Spiele entwickelt¹⁸ Im Gegensatz zu Flash gibt es für Shockwave kein Plugin für die Linux-Plattform.

¹⁷Beispiele für 3D mit JavaScript: <http://www.zed0.getdata.be/J-Plotter.php> und <http://www.abrahamjoffe.com.au/ben/canvascape/>

¹⁸Beta Version eines 3D-Shooters: <http://www.rasterwerks.com/game/phosphor/beta1.asp>

VRML (Virtual Reality Modelling Language) ist eine 3D Modellierungssprache, die 1995 mit der Version 1.0 eingeführt wurde. Es entstand aus Open Inventor von Silicon Graphics, Inc., welches zum Erstellen realistischer 3D-Umgebungen mit minimalen Kenntnissen der Grafikprogrammierung entwickelt wurde. Diese Version erstellte rein statische Szenen, die zwar mit einer Kamerafahrt animiert werden können, aber die Geometrien konnten nicht mehr verändert werden.

Erst die spätere Version VRML97 konnte auch mit Benutzereingaben umgehen und umfasst den Script-Knoten, der das Einbinden externer Skripte erlaubt, sowie einige Multimedia-Komponenten für die Animation. Seit dem wird es von den meisten Modellierungsprogrammen als Exportformat unterstützt. [12]

X3D, Extensible 3D, ist der Nachfolger von VRML 1.0 und VRML97 und der ISO-Standard für 3D Grafik.

Für Arborata wurde X3D zur Definition und Darstellung der dreidimensionalen Objekte gewählt. Warum es gewählt wurde und wie es funktioniert, folgt im nächsten Kapitel 2.3.

2.3 X3D

Die 3D-Modellierungssprache X3D ist nicht nur ein offenes ISO-Dateiformat, sondern auch eine Laufzeitarchitektur. Es kombiniert sowohl die Beschreibung der Geometrien, als auch des Laufzeitverhaltens für die Echtzeitdarstellung. [3]

Es kann in VRML-Syntax (Classic-Encoding, Datei-Endung .x3dv) oder auf XML-Syntax (XML-Encoding, Datei-Endung .x3d) aufgebaut werden. Es ermöglicht dynamische 2D/3D Grafik mit gebietsabhängigem Audio, Video, Animation, Navigation, Interaktion und Netzwerkunterstützung, ausserdem können eigene Skripte zur Manipulation der XML-Knoten implementiert werden, beispielsweise mit ECMAScript oder Java. [13]

Zur Anpassung an die spezifischen Anforderungen des Inhalts wird X3D in **Komponenten** unterteilt, die eine Sammlung von X3D-Knoten für einen bestimmten Zweck enthalten (z.B. die Texturkomponente mit den Knoten ImageTexture, TexturTransform, MovieTexture, etc.). Es gibt für jede Komponente mehrere Level, die auch steigende Implementierungsanforderungen definieren. Mit jedem Level steigt auch die Anzahl und/oder Komplexität der zu unterstützenden Knoten oder Attribute. [3]

Ein **Profil** fasst Komponenten bestimmter Levels zusammen. Für jede X3D-Datei muss angegeben werden, welches Profil benötigt wird. [3]

X3D Basis-Profile [3]:

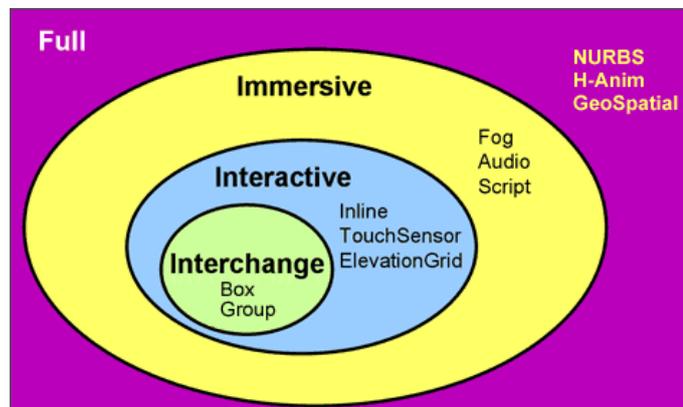


Abbildung 3: X3D Basis-Profile [3]

- **Interchange** ist das Basis-Profil zur Kommunikation zwischen Anwendungen. Es unterstützt Geometrie, Textur, Basisbeleuchtung und Animation.
- **Interactive** aktiviert grundlegende Interaktion mit der 3D-Umgebung mit Sensor-Knoten für Navigation und Interaktion (z.B. PlaneSensor, TouchSensor, etc.)
- **Immersive** bringt volle 3D Grafik und Interaktion mit Audio, Kollision, Nebel und Skripten.
- **Full** beinhaltet alle definierten Knoten inklusive NURBS-, H-Anim- und GeoSpatial-Komponenten.

Weitere X3D Profile [3]:

- **MPEG-4 Interactive** ist eine kleine Version des Interactive Profils extra für Broadcast und mobile Geräte.
- **CDF (CAD Distillation Format)** ist in Entwicklung, um CAD-Daten in ein offenes Format zu übersetzen.

Für Arborata werden die Komponenten Geometrie, Textur, Beleuchtung, Animation, Navigation und Interaktion benötigt. Es kommt aber auch der Nebel und Audio zum Einsatz. Daher ist in jeder X3D-Datei von Arborata das X3D-Profil *Immersive* angegeben.

Es folgen Beispiele zum Aufbau einer X3D-Datei in VRML-Syntax und in XML-Syntax, die eine Kugel mit Erdtextur definieren.

```

#X3D V3.0 utf8
PROFILE IMMERSIVE
META "description" "Hello World"

Transform {
  children [
    Viewpoint {
      description "The World"
      orientation [ 0 1 0 1.57 ]
      position [ 6 0 0 ]
    }
    DEF TheWorldTurns Transform {
      rotation 0 1 0 1.57
      children [
        Shape {
          geometry Sphere {
          }
          appearance Appearance {
            texture ImageTexture {
              url "earth-topo.jpg"
            }
          }
        }
      ]
    }
  ]
}

```

Beispiel: Die VRML-Notation der "Hello World"-Grafik. Die Syntax erinnert durch die Klammerungen noch stark an C.

Innerhalb des Gruppierungsknoten Transform wird einmal der Viewpoint `The World` definiert, der die Kameraposition sechs Meter entlang der X-Achse verschiebt und die Blickrichtung um 90° (1,57 rad) um die Y-Achse rotiert.

Danach wird ein weiterer Transform-Knoten `TheWorldTurns` angegeben, der alle Kindelemente ebenfalls um 90° um die Y-Achse rotiert. Bei dem Kindelement handelt es sich um den Geometrieknoten `Sphere`, der eine Kugel mit dem voreingestellten Radius 1m definiert. Der Kugel wird durch Angabe einer URL noch eine Bildtextur zugewiesen: `url "earth-topo.jpg"`.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN"
    "http://www.web3d.org/specifications/x3d-3.0.dtd">
<X3D profile="Immersive" >
  <head>
    <meta content="Hello World" name="description"/>
  </head>
  <Scene>
    <Viewpoint description="The World"
      orientation="0 1 0 1.57" position="6 0 0"/>

    <Transform rotation="0 1 0 1.57"
      DEF='TheWorldTurns' >
      <Shape>
        <Sphere/>
        <Appearance>
          <ImageTexture url="earth-topo.jpg"/>
        </Appearance>
      </Shape>
    </Transform>
  </Scene>
</X3D>

```

Beispiel: Die XML-Notation der "Hello World"-Grafik [14]. Nach der XML-Deklaration folgt die DOCTYPE-Deklaration, die die Document Type Definition (DTD) angibt, um das XML-Dokument zu validieren. Die XML-Notation erstellt genau das gleiche Bild, wie die VRML-Notation. Der X3D-Browser wird beides interpretieren können.



Abbildung 4: "Hello World" im Octaga Player

2.3.1 Navigation

Die Navigation-Komponente von X3D stellt verschiedene Navigationsweisen zur Verfügung:

- EXAMINE lässt die Kamera um einen Rotationsmittelpunkt kreisen und ist vor allem zum Betrachten einzelner Objekte geeignet.
- WALK wird zum natürlichen Laufen in der Welt benutzt, wobei ein Avatar definiert wird, der angibt, in welcher Höhe die Augen (Kamera) liegen und damit auch welche Höhenunterschiede überwunden werden können. Dabei wird Gravitation und Kollision benutzt, um das Gefühl von Laufen oder Fahren zu vermitteln.
- FLY ist ähnlich dem WALK, es wirkt keine Gravitation auf den Avatar des Betrachters.
- LOOKAT bewirkt das Betrachten eines angewählten Objekts, so dass das Zentrum der Kamera-Rotation in das Objekt gesetzt wird.
- ANY lässt den X3D Browser eine Navigationsart wählen, die in den Kontext passt und der Benutzer kann jederzeit eine andere wählen.
- NONE: Es steht keine der genannten browserspezifischen Navigationsarten zur Verfügung, der Benutzer kann nur über definierte Ankerpunkte in der Szene navigieren.

Für Arborata wurde WALK gewählt mit einer Geschwindigkeit von 2 m/s, da dies am ehesten der natürlichen Betrachtungsweise auf ein Gelände entspricht. Außerdem kann der Benutzer jederzeit eine andere Navigationsart (ANY) wählen.

```
<NavigationInfo type=' "WALK" "ANY"'
  headlight=' false'
  speed=' 2' />
<Viewpoint description='Grundposition'
  position='0 {erster Höhenwert im Gelände} 0'
  orientation='0 1 0 3.93' />
<Viewpoint description='Vogelperspektive'
  position='4 10 5'
  orientation='1 0 0 -1.5' />
```

Es können mehrere "Aussichtspunkte" definiert werden unter Angabe der Kameraposition und der Blickrichtung. Für Arborata werden zwei Punkte definiert. Zum einen der Startpunkt, der auf der nordwestlichen Ecke des Geländes liegt und das Laufen auf dem Gelände erlaubt. Die Blickrichtung wird um die Z-Achse zum Gelände rotiert und für die Position der Kamera wird die Z-Koordinate auf den ersten Höhenwert im Gelände plus der Avatarhöhe gesetzt. Der Avatar hat standardmäßig eine Größe von 0,25x1,6x0,75 Meter. Diese Größe ist hier beibehalten worden.

Der zweite Aussichtspunkt muss über den Browser aufgerufen werden und liegt oberhalb des Geländes, so dass man auf die Mitte des Terrains schaut. Die Positionsangabe ' 4 10 5 ' gilt für ein Gelände der Größe 8x10 Meter und die Kamera schaut entlang der negativen Y-Achse.

2.3.2 Interaktion mit ECMAScript

Da bei den Benutzertests (siehe Kapitel 4.1) aufgefallen ist, dass im X3D-Browser keine echte Interaktion möglich ist, wurde noch eine Funktion eingebaut, die zum Test der Interaktionsmöglichkeiten dient. Da alle Probanden des Tests eine Anzeige eines Größenrasters auf dem Gelände für sinnvoll hielten, soll diese Funktion nachträglich eingebaut werden. Die Gitterlinien werden einen Abstand von einem Meter zueinander haben und können mit der Taste R (für Raster) und G (für Gitter oder Grid) ein- bzw. ausgeblendet werden.

Das Gitter liegt als Bildtextur auf dem Gelände und erhält den eindeutigen Bezeichner `Raster`. Das Bild "raster_aus.png" ist ein einfarbig weißes Bild, welches durch das Bild "raster_ein.png" ersetzt werden soll, in welchem rote Linien zu sehen sind.

```
<ImageTexture DEF='Raster' url='raster_aus.png'
  repeatS='true' repeatT='true' />
```

Um Tastenereignisse verarbeiten zu können, wird ein `KeySensor`-Node benötigt, der alle Tastatureingaben aufnimmt. Er bekommt den Bezeichner `keySensor`.

```
<KeySensor DEF='keySensor' />
```

X3D kann jedoch nicht einfach so die Eingaben auf das Raster umleiten, sondern es erfolgt in einzelnen Schritten über sogenannte `Routes`.

1. Der `KeySensor` empfängt ein Ereignis mit dem Namen `keyPress`, es hat den Typ `SFString`
2. Die erste Route leitet das Ereignis von dort zum Script-Knoten namens `Script` in das Feld `set_input`
3. Das Skript erlaubt den Input von `set_input` und gibt den `String` `string_changed` zurück (da man pro URL-Eintrag mehrere Quellen definieren kann, wird hier `string_changed[0]` gesetzt)
4. Die zweite Route übernimmt das Feld `string_changed` und gibt es an das Feld `url` von `Raster` weiter

Das ECMAScript und die `Routes` sehen dann wie folgt aus:

```
<Script DEF='Script'>
  <field name="set_input" type="SFString"
    accessType="inputOnly"/>
  <field name="string_changed" type="MFString"
    accessType="outputOnly"/>

  <![CDATA[
    ecmaScript:
    i = 0;

    function set_input( value, ts )
    {
      if( value=='G' || value=='R' )
      {
        if( i%2 ) string_changed[0] = 'raster_ein.png';
        else string_changed[0] = 'raster_aus.png';
        i = i+1;
      }
    }
  ]>
```

```
</Script>
```

```
<ROUTE fromNode="keySensor" fromField="keyPress"  
      toNode="Script" toField="set_input"/>
```

```
<ROUTE fromNode="Script" fromField="string_changed"  
      toNode="Raster" toField="url"/>
```

Beim Tastendruck auf R oder G wird das Gitter ein- oder ausgeblendet.

Über ECMAScript, aber auch Java, kann man den gesamten DOM (Document Object Model) der X3D-Datei manipulieren und X3D bietet nicht nur den KeySensor, sondern z.B. auch den TouchSensor (beim Anklicken einer Geometrie). Auf diese Weise lassen sich vielfältige Interaktionsmöglichkeiten schaffen.

2.3.3 X3D-Player

Es gibt mehrere Player zur Darstellung der X3D-Dateien. Diese rendern die jeweilige Ansicht und führen auch Skripte aus, falls welche vorhanden sind.

Auf der Webseite des Web3D Consortiums ¹⁹ werden einige Browser vorgestellt, von denen hier nur drei erwähnt werden, weil sie frei erhältlich sind und auch höheren Bekanntheitsgrad haben:

- Xj3D Player vom Web3D Consortium ²⁰
- Flux Player von Media Machines ²¹
- Octaga Player von Octaga ²²

Der **Xj3D Browser** ist ein Projekt des Web3D Consortium. Er ist vollständig in Java geschrieben und dient zur Darstellung von VRML97 und X3D-Dateien. Allerdings konnten nach der Installation des Browsers keine X3D-Datei tatsächlich geladen werden. Selbst Beispiele von der Web3D Webseite riefen immer Parsing Fehler hervor.

Der **Flux Player** von Media Machines, ein kostenfreies Plugin für Internet Explorer und Firefox, fällt besonders auf durch die kurzen Ladezeiten der X3D Szenen, macht aber Probleme bei der Navigation, da Flux den Navigationsmodus "WALK" nicht unterstützt. Ein zweites Problem, welches die Nutzung von Flux in diesem Projekt erschwert: Zwar erfüllt Flux die

¹⁹http://www.web3d.org/tools/viewers.and_browsers

²⁰<http://www.xj3d.org>

²¹<http://www.mediamachines.com>

²²<http://www.octaga.com>

Mindestanforderungen für das Immersive Profil mit 16000 Höhenangaben pro Gelände (ElevationGrid), aber das ist für größere Gelände unzureichend.

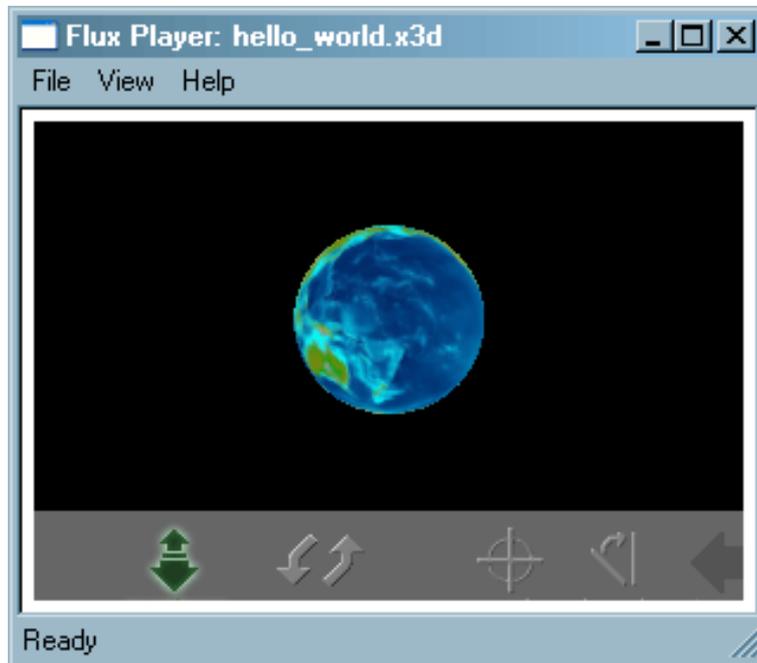


Abbildung 5: "Hello World" im Flux Player

Die besten Ergebnisse wurden mit dem **Octaga Player** erzielt. Er braucht zwar meist beim Start einige Sekunden, um die Szene vorzuladen, stellt dann aber auch mehr als 16000 Höhenangaben pro Gelände dar und setzt auch den Navigationsmodus "WALK" sehr gut um. Insgesamt ist es einfacher mit dem Octaga Player zu arbeiten, da er Parsing Fehler und andere Warnungen direkt beim Laden der Szene darstellt, im Gegensatz zum Flux Player, der leider keinerlei Warnung zeigt, sondern die Szene oder das Objekt mit dem Fehler einfach nicht anzeigt.

Der Octaga Player ist in Abbildung 4 zu sehen.

3 Entwicklung und Implementation

Es soll auf einer übersichtlichen Weboberfläche gearbeitet werden, die nicht durch zu viele Optionen verwirrt. Es wurde versucht, sich auf einfache Funktionen zu beschränken, die aber dem Benutzer größt mögliche Flexibilität bieten sollen. Dabei wird zwischen dem Gelände, den Wegen/Plätzen und Objekten (Pflanzen und Gegenstände) unterschieden.

Die folgenden Unterkapitel sind in die Hauptbestandteile des Programms gegliedert, dazu gehört Gelände, Wege/Plätze und Objekte. Um die einzelnen Aspekte dieser Teile zu beschreiben, wird jeder in die Abschnitte Webseite, Flash und X3D geteilt. "Webseite" erklärt den Aufbau der Internetansicht, "Flash" erläutert die Funktionsweise der programmierten Flashapplikation und "X3D" umfasst die speziellen Probleme bei der dreidimensionalen Darstellung in X3D.

3.1 Gelände

Die Erstellung eines Geländes stellt in allen Programmen stets eine Herausforderung dar. Meist zählen präzise Höhendaten, statt Benutzerfreundlichkeit. Ein Beispiel sind Höhenlinien. Diese werden meistens aus Vektorprogrammen in 3D-Programme importiert und erst dort auf die entsprechende Höhe gezogen. Darüber wird dann die Fläche gelegt. Genutzt wird diese Methode vor allem im vermessungstechnischen Bereich wegen der hohen Genauigkeit. Allerdings ist das für Laien ohne viel Übung sehr schwierig zu handhaben, da Höhenlinien für ein beliebiges Gelände nur schwer abzuschätzen sind und nur im seltensten Falle sind die Daten für das eigene Gelände vorhanden.

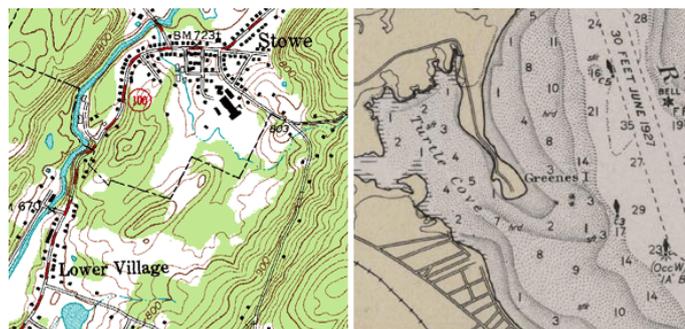


Abbildung 6: Beispielkarten mit Höhenlinien (l) und Höhenpunkten (r)

Eine andere Variante ist das Setzen einzelner Höhenpunkte. Dabei werden einzelne Punkte im Gelände gesetzt, deren Höhe bekannt ist, und dar-

auf wiederum die Oberfläche gelegt. Einige der zur Zeit erhältlichen Gartendesignerprogramme benutzen diese Methode. Aber selbst mit mehreren Versuchen ist es sehr schwer das gewünschte Gelände auch nur grob darzustellen, da man für ein realisiertes Terrain schon sehr viele Punkte benötigt, deren Höhen interpoliert werden.

Für die Diplomarbeit wurde die Erstellung mit Hilfe einer Heightmap gewählt. Diese Methode wird beispielsweise in Terragen²³ oder Bryce²⁴ verwendet, um Landschaften zu generieren. Die Heightmap besteht aus einem Grauwertbild, wobei Schwarz einer Höhe von 0 Metern und Weiß der maximalen Höhe entspricht. Alle anderen Werte werden dazwischen interpoliert. Zur Zeit werden Höhenwerte zwischen 0 und 5 Metern benutzt. Allerdings sind 5 Meter sehr viel und so ein Höhenunterschied wird meist nicht gebraucht. Eine andere Möglichkeit wäre, es dem Benutzer zu überlassen, welchen maximalen Höhenunterschied er nutzen möchte.

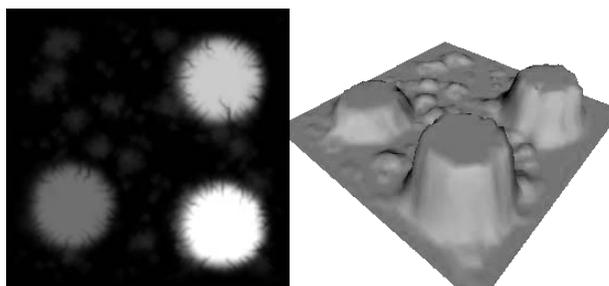


Abbildung 7: Beispiel einer Heightmap (links) und deren dreidimensionale Ansicht (rechts) aus dem Programm Bryce. [4]

Durch das Umrechnen der Gelände Größen kann bei der X3D-Vorschau eine Überlastung des Rechners verhindert werden und das Gelände wird im Flash-Plugin geeignet skaliert. Bei einem großen Gelände ist ein Patch bis zu 60 cm x 60 cm groß, während ein kleines 5x5 Meter Gelände Patches der Größe 3,5 x 3,5 cm hätte und so viel mehr Details zulässt. Bei einem rechteckigen Gelände sind dementsprechend auch die Patches rechteckig. Die x- und die z-Größe werden unabhängig voneinander berechnet.

3.1.1 Webseite

Um ein Gelände zu erstellen, stehen dem Benutzer drei verschiedene Funktionen zur Verfügung:

²³<http://www.planetside.co.uk/terrigen/>

²⁴<http://www.daz3d.com/program/bryce/>

1. Neues Gelände erstellen, wobei man die Größe des Geländes angeben kann (später sollte auch die Höhe angegeben werden können)
2. Einlesen eines X3D-Projekts. Man gibt eine X3D-Datei an, die mit Arborata erstellt wurde, die Daten werden automatisch eingelesen und können bearbeitet werden
3. Vorbereitetes Standardgelände einlesen (Das sind vordefinierte Gelände, wenn man kein eigenes erstellen möchte)



Abbildung 8: Die Übersicht zur Geländeerstellung. Man hat die Optionen: neues Gelände erstellen, Arborata Projekt laden, fertiges Gelände laden.

3.1.2 Flash

Die Höhenformation des Geländes kann in der Weboberfläche mit Hilfe der erstellten Flash-Zeichenapplikation "gemalt" werden. Die Länge und Breite des Terrains kann der Benutzer über die Weboberfläche festlegen. Um zu verhindern, dass das Gelände zu klein für ein vernünftiges Arbeiten und nicht zu groß für die Flashanzeige ist, wird eine interne Auflösung berechnet. Diese wird wie folgend berechnet:

$$\text{round}(\text{Größe in Pixel} = \log(\text{Größe in Meter}) * 200)$$

Diese Auflösungswerte werden für die Länge und Breite separat berechnet, das heißt ein Gelände der Größe 10 x 15 Meter wird in Flash mit 200 x 235 Pixeln angezeigt.

Größe einer Geländeseite in Meter	Größe einer Geländeseite in Pixel
5	140
10	200
15	235
50	340
100	400
300	495

Tabelle 1: Beispielwerte für die Auflösung in Flash

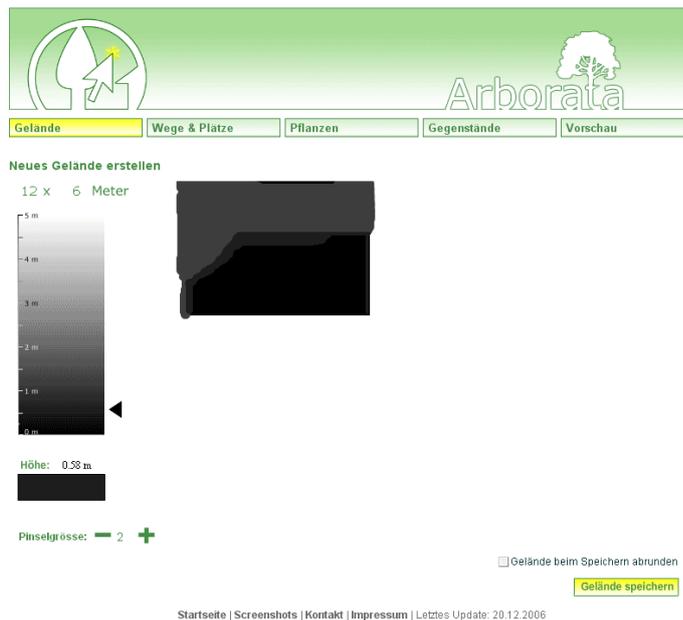


Abbildung 9: Zeichnen der Höhenwerte in der Flashapplikation gelaende.swf

Als die Benutzertests abgeschlossen waren, stellte sich heraus, dass das separate Berechnen der Auflösung zweier Geländeseiten ungünstig gewählt ist, da Verzerrungen entstehen. Wegflächen, die in der Flashansicht rund gemalt wurden, haben in der X3D-Ansicht eine ovale Form (gut zu sehen in Abbildung 10). Wenn Arborata weiterentwickelt wird, sollte die Berechnung der Auflösung auf Basis von nur einer Seite erfolgen.

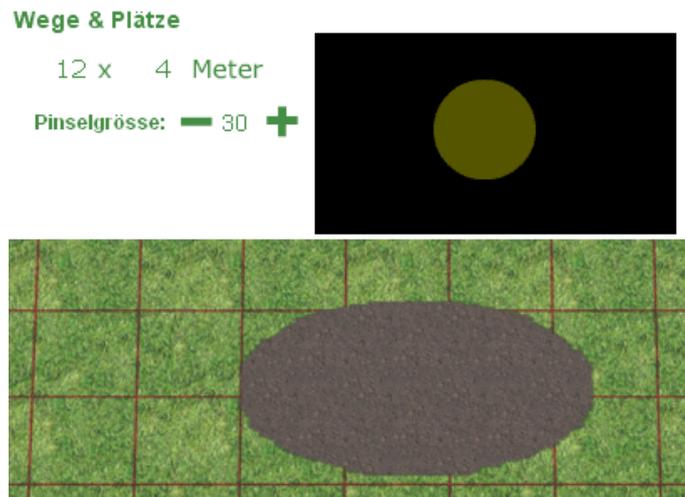


Abbildung 10: Verzerrung der Wegform bei einem Gelände der Größe 12x4 Meter. Das rote Raster markiert die Quadratmeter.

Die Höhenwerte werden mit Hilfe der Grauwertauswahl auf die Fläche gemalt. Es entstehen dabei Hexadezimalwerte zwischen 00 und FF (dezimal: 0-255), die an PHP übergeben und dann in der Datenbank gespeichert werden. Würde Flash eine Kompression dieser Daten ermöglichen, würde das Senden der Daten nicht zu viel Bandbreite benötigen und auch die Größe der Daten in der Datenbank könnte so gering gehalten werden.

3.1.3 X3D

Speziell für das Speichern und Anzeigen von Geländen stellt X3D die Knoten ElevationGrid und GeoElevationGrid zur Verfügung. Das GeoElevationGrid unterstützt Gelände, welches der Erdkrümmung angepasst werden. Das wird hier jedoch nicht benötigt, so dass nur ElevationGrid benutzt wird. Unter Angabe der Länge und Breite in Metern, sowie der Abstand der einzelnen Höhenpunkte, wird ein Terrain generiert, welches die angegebenen Höhenpunkte durchläuft.

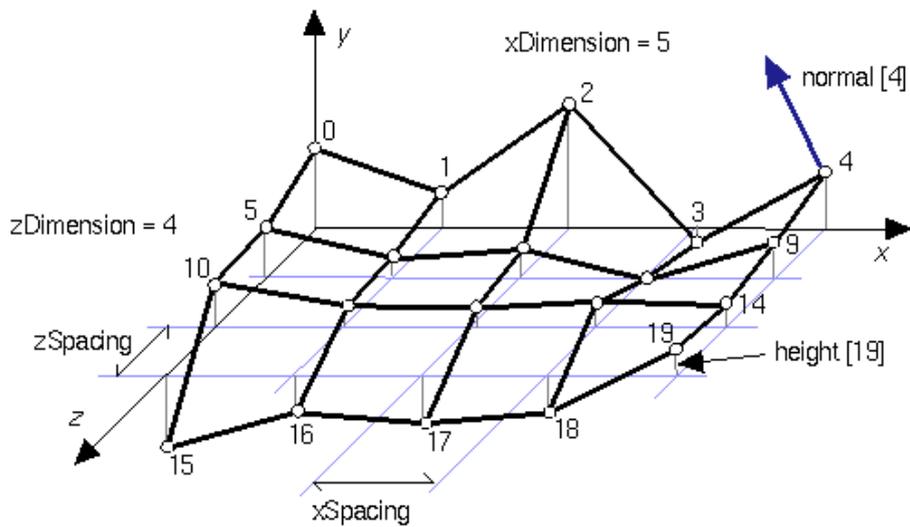


Abbildung 11: ElevationGrid: zeigt die Zusammenhänge von Spacing und Dimension mit den Höhenwerten.

Das folgende Beispiel zeigt die X3D-Notation eines Geländes der Größe 4,0 x 9,0 Meter mit insgesamt 22920 Patches und somit 23232 Höhenwerten. Mit dem Knoten Appearance wird dem Gelände eine Textur zugeordnet, die auf die Größe 2,4 x 0,8 Meter skaliert wird.

```
<Shape>
  <ElevationGrid
    DEF='ElevationGrid'
    colorPerVertex='false'
    xSpacing='0.033'
    zSpacing='0.047'
    xDimension='120'
    zDimension='191'
    height='0.00 0.00 1.02 1.02 1.04 ...' />
  <Appearance>
    <ImageTexture
      repeats='true'
      repeatT='true'
      url='surface0.png' />
    <TextureTransform
      center='0,0'
      scale='2.4 0.8' />
  </Appearance>
</Shape>
```

Größe einer Geländeseite in Meter	Anzahl der Patches pro Geländeseite	Abstand zwischen den Höhenpunkten in Metern
5	140	0,035
10	200	0,05
15	235	0,06
50	340	0,15
100	400	0,25
300	495	0,6

Tabelle 2: Beispielwerte für die Auflösung des Geländes

Durch die entstandenen Gelände-größen ergibt sich auch der Speicherbedarf für ein Gelände in der Datenbank. Die Speicherung erfolgt in Floatwerten mit zwei Nachkommastellen, jeweils getrennt durch ein Leerzeichen, wodurch pro Höhenangaben 5 Zeichen benötigt werden. Für das Speichern des Geländes muss also ein Datenbankfeld der Größe `MEDIUMBLOB` mit maximal 2^{24} Byte verwendet werden. Es wäre in Zukunft besser auch hier die Hexadezimalzahlen zu speichern, da immer Schwarz (00) bis Weiß (FF) benutzt wird und somit nur zwei Zeichen pro Höhenwert. Speichert man zusätzlich noch die maximale Höhe des Geländes, so sind die Hexadezimalwerte in der Datenbank skalierbar.

Größe beider Geländeseiten in Meter	Anzahl der Patches des Geländes	Anzahl der Dreiecke im ElevationGrid	Speicherbedarf der Höhenwerte in der Datenbank in Byte
5	19 600	39 200	96 000
10	40 000	80 000	202 005
15	55 225	110 450	278 480
50	115 600	231 200	581 405
100	160 000	320 000	804 005
300	245 025	490 050	1 230 080

Tabelle 3: Größe und Speicherbedarf einiger quadratischer Beispielgelände

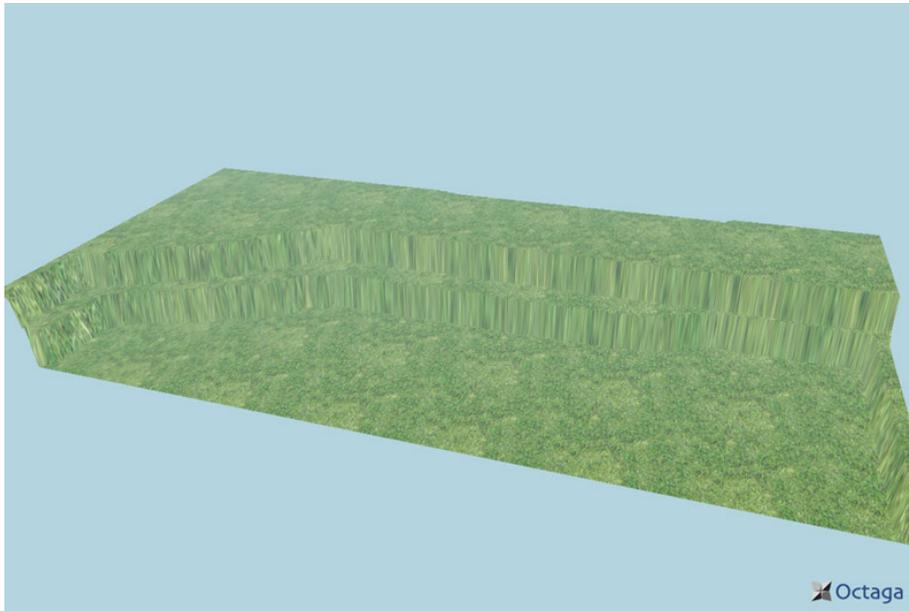


Abbildung 12: Das entstandene Gelände im X3D-Browser Octaga.

3.2 Wege/Plätze

Das Ziel ist das Erstellen möglichst realistischer Wege, aber auch Plätze. Dabei sollen Wege auch unterschiedlich breit sein und Plätze sollen auch "Löcher" enthalten können. Am besten sollen die Wege und Plätze mit der selben Methode erstellt werden.

Um diese Anforderungen zu erfüllen, wurden verschiedene Möglichkeiten betrachtet, wie man Wege/Plätze auf das Terrain legen kann:

- **Splines** mit einer definierten Breite, deren Kontrollpunkte und Grad der Kurve gespeichert werden. Mit dieser Methode kann man jedoch keine runden Plätze realisieren.
- **TrimmingCurves**: Mit Hilfe von Splines können Trimming Curves auf einer NURBS-Oberfläche definiert werden, um Löcher oder glatte Wegbiegungen zu erstellen. X3D verfügt über eine NURBS-Komponente, welche auch Trimming Curves beinhaltet, jedoch wären die NURBS sehr schwierig zu handhaben, da sie je nach Geländeform das Terrain durchdringen könnten und nur schwer so anzupassen ist, dass sie glatt auf dem Gelände liegen.
- Eine **Textur** mit Hilfe einer Bitmap patchweise auf das Gelände legen. Die erste Idee war einfach die Textur des Weges/Platzes auf das Terrain zulegen. Das würde Probleme mit kollidierenden oder ineinandergreifenden Geometrien verhindern und gleichzeitig erlaubt es höchste Flexibilität, wie zum Beispiel Löcher oder mehrere Flächen-teile. Außerdem kann auf diese Weise die Malmetapher aus der Geländeerstellung beibehalten werden.

Da die Texturmethode die meisten Möglichkeiten bietet und den Vorteil hat, dass die Malmetapher auch hier erhalten bleibt, wurde mit dieser Methode weitergearbeitet. Es soll ebenenweise gearbeitet werden, so dass man für jede Textur eine eigene Ebene auf das Gelände legt. Die Reihenfolge der Ebenen soll geändert werden können.

3.2.1 Webseite

Da das erstellte Gelände immer mit einer Grastextur belegt wird, sieht man in der Übersicht zunächst ein kleines Bild der Grastextur. Diese Grunde-bene lässt sich auch nicht löschen. Wenn ein Terrain existiert, können wei-tere Ebenen hinzugefügt werden, in dem man eine Textur auswählt und in der Flashoberfläche einzeichnet. Es können beliebig viele Ebenen er-stellt werden. Jede der so erstellten Texturebenen ist dann in der Wege-/Plätze-Übersicht zu sehen. Mit Hilfe von Pfeilen kann die Ebenenreihen-folge geändert werden, wobei die grasbedeckte Grundebene immer die un-

terste bleibt. Für eine Weiterentwicklung des Programms würde es sich anbieten auch die Textur der untersten Ebene änderbar zu machen und auch die Texturen der anderen Ebenen neu auswählen zu können, ohne den Weg neu zeichnen zu müssen.

Über die Preisangabe pro Quadratmeter kann auch eine Kostenabschätzung erzeugt werden, dabei wird berechnet, wie viel Prozent des Gesamtgeländes von dem Weg bedeckt wird.



Abbildung 13: Auflistung aller platzierten Wege mit ihrer Textur.

3.2.2 Flash

Hat man auf der Weboberfläche eine Wegtextur ausgewählt, wird die Geländeheightmap in Flash geladen und der Benutzer kann hier mit verschiedenen Pinselstärken (Tasten 1 bis 9 und oben/unten) den Weg auf das Terrain "malen". Nach der Rasterisierung der Zeichnung ist das Ergebnis der Speicherung eine Bitmap mit einer 1 für jede Wege-Vertex und einer Null für jede Nicht-Wege-Vertex. Der Speicherbedarf in Byte ist demnach genauso hoch, wie die Anzahl der Höhenwerte (siehe Tabelle 3)

3.2.3 X3D

Die Idee Texturen auf das Gelände zu legen sah vor, dass man mehrere Texturen mit Hilfe von Masken so mischen kann, dass nur eine Terraingometrie benötigt wird. Dafür wurden mehrere Tests mit dem X3D-Knoten "MultiTexture" gemacht. Laut X3D-Spezifikation kann der Multitexture-



Startseite | Screenshots | Kontakt | Impressum | Letztes Update: 20.12.2006

Abbildung 14: Die Flashapplikation flaechen.swf, wobei auf das Höhenbild des Geländes die Form des Weges gemalt wird.

Knotenbeliebig viele Texturen verarbeiten²⁵ und unterstützt unterschiedliche Blending-Modi der Texturen. Die Versuche funktionierten, solange ausser der Grundebene noch eine Maske und Textur hinzugefügt wurden, aber bei allen weiteren Maske/Textur-Paaren funktionierte die Darstellung nicht mehr richtig.

Um dennoch die Texturmethode zu benutzen, wird für jede Textur eine eigene Geländeinstanz erzeugt, die jeweils 10 cm höher liegt als die vorherige. Dadurch kann die Animation bei größeren Terrains und/oder vielen Wegebenen ruckeln.

Das folgende Beispiel zeigt die X3D-Notation eines Weges/Platzes auf dem Gelände. Es wird eine weitere Instanz des Terrains `ElevationGrid` erzeugt, aber 10 cm höher und statt des Grases wird eine Multitextur darüber gelegt, die aus der Bitmaske `Pixeltexture` und dem gewünschten Muster `ImageTexture` besteht. Das Erstellen der Bitmaske erfolgt über Angabe der Breite, Länge, Anzahl der Farbkomponenten, sowie den einzelnen Farbwerten. Die 2 Kanäle geben an, dass nur Grauwerte mit Opazität genutzt werden²⁶, demnach werden die einzelnen Farbwerte nur mit `0x0000` für Schwarz und Transparent und `0xFFFF` für Weiß und undurchsichtig angegeben.

²⁵<http://www.web3d.org/x3d/specifications/ISO-IEC-19775-X3DAbstractSpecification/Part01/components/texturing.html#MultiTexture>

²⁶Für weitere Details zu den Farbkanälen siehe <http://www.web3d.org/x3d/specifications/ISO-IEC-19775-X3DAbstractSpecification/Part01/components/texturing.html#Concepts>

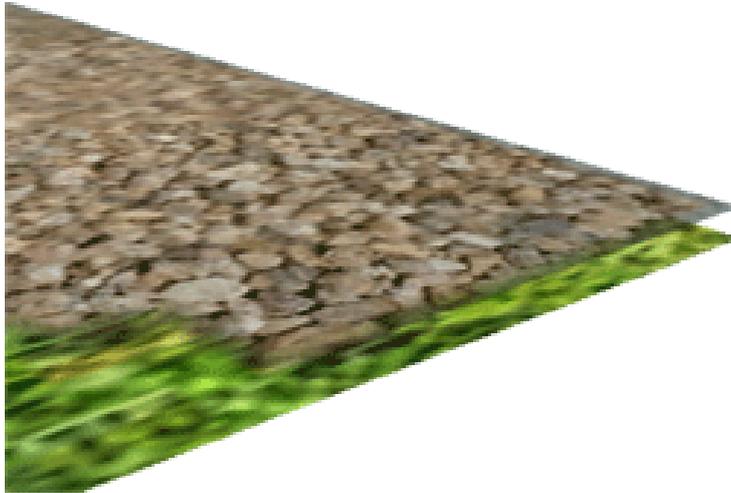


Abbildung 15: Die Wegtextur liegt 10 cm über dem eigentlichen Gelände

Mit MODULATE wird die Texturfarbe mit der aktuellen Farbe multipliziert (Arg1 x Arg2).

```

<!-- Wege / Flaechen -->
<Transform DEF='surfaces_7_1' translation='0 0.01 0'>
  <Shape>
    <ElevationGrid USE='ElevationGrid' />
    <Appearance>
      <MultiTexture mode=' "MODULATE" "MODULATE"' >
        <ImageTexture
          repeatS='true'
          repeatT='true'
          url=' "surface7.jpg" ' />
        <PixelTexture
          image='120 191 2 0x0000 0xFFFF ...' />
      </MultiTexture>
      <MultiTextureTransform>
        <TextureTransform
          center='0,0'
          scale='12 4' />
        <TextureTransform />
      </MultiTextureTransform>
    </Appearance>
  </Shape>
</Transform>

```

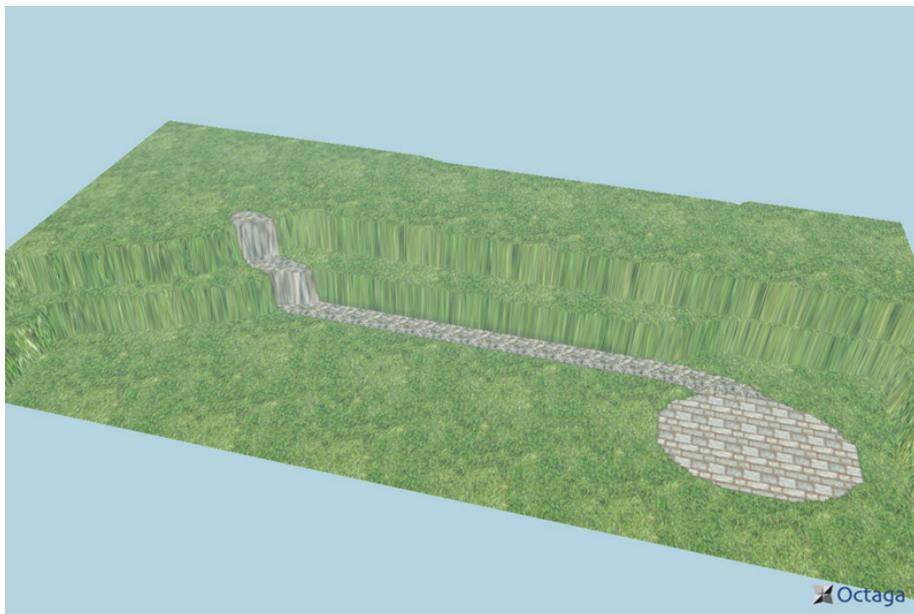


Abbildung 16: Vorschau des Gartens mit den gewählten Wegen.

3.3 Objekte: Pflanzen und Gegenstände

Unter Objekten fasse ich Pflanzen und Gegenstände zusammen, obwohl sie in verschiedenen MySQL-Tabellen gespeichert werden. Aber für beide ist der Arbeitsablauf der gleiche: Objekt auswählen und Objekt platzieren.

3.3.1 Webseite

Zu Anfang erhält man eine Übersicht über die bereits platzierten Pflanzen oder Gegenstände. Unabhängig davon, ob bereits ein Gelände erstellt wurde, kann man die Objektdatenbank durchsuchen. Eine Platzierung des ausgewählten Objekts kann dagegen erst erfolgen, wenn ein Gelände gespeichert ist.

Zum Durchsuchen der Objekte stehen verschiedene Methoden zur Verfügung. Man kann sie nach deutschen Namen (die Pflanzen auch nach lateinischen Namen) sortieren und man kann eine Volltextsuche nutzen. Bei den Pflanzen kann man sogar eine Suche über die Eigenschaften formulieren. Dabei gibt man Werte für verschiedene Pflanzeigenschaften an, z.B. (Blütenfarbe=rot oder Blütenfarbe=blau) und Wuchsform=kletternd. So werden alle Kletterpflanzen mit roten oder blauen Blüten gefunden.



The screenshot shows the 'Arborata' website interface. At the top, there is a navigation bar with tabs for 'Gelände', 'Wege & Plätze', 'Pflanzen' (which is highlighted), 'Gegenstände', and 'Vorschau'. Below the navigation bar, the page title 'Pflanzen' is displayed, followed by the text 'Derzeit haben Sie folgende Pflanzen ausgewählt:'. A table lists the selected plants with columns for 'Name', 'lat. Name', 'Preis/Stück in €', 'Anzahl', and 'Preis in €'. Each row includes a small image of the plant and a trash icon. At the bottom right of the table, the total price is shown as 'Gesamtpreis: 8021.00 €'. Below the table, there is a button labeled 'Neue Pflanze suchen'.

Name	lat. Name	Preis/Stück in €	Anzahl	Preis in €
 Japanischer Gold-Ahorn	Acer shirasawanum 'Aureum'	6650.00	1	6650.00
 Kirschlorbeer	Prunus laurocerasus	83.00	16	1328.00
 Schmucklilie	Agapanthus campanulatus	4.30	10	43.00

Gesamtpreis: 8021.00 €

Neue Pflanze suchen

[Startseite](#) | [Screenshots](#) | [Kontakt](#) | [Impressum](#) | Letztes Update: 20.12.2006

Abbildung 17: Auflistung aller gewählten und platzierten Pflanzen. Eine ähnliche Übersicht gibt es auch für die Gegenstände.

3.3.2 Flash

Um die Objekte auf dem Gelände zu platzieren, wird nach Auswahl der gewünschten Pflanze eine Flashoberfläche mit dem erstellten Gelände geladen. Darauf kann eine beliebige Anzahl von Objekten platziert werden. Pflanzen werden mit grünen Punkten markiert und Gegenstände mit blauen Quadraten. Beim Speichern werden die Koordinaten der Objekte in die Datenbank geschrieben.



Abbildung 18: Die Platzierung der Pflanzen werden mit einem grünen Punkt auf der Höhenkarte des Geländes markiert. Gegenstände erhalten ein blaues Quadrat.

3.3.3 X3D

Je nach Eintrag in der Datenbank kann jedes Objekt als Billboard oder als 3D-Objekt importiert werden. Gerade bei Pflanzen lohnt es sich aber ein Billboard zu verwenden, da die Anzahl der zu rendernden Polygone schon bei einer Pflanze extrem steigen kann. Bei anderen Objekten hingegen lohnt sich eine dreidimensionale Darstellung, da man schon mit geringer Polygonzahl sehr schöne Objekte erzeugen kann.

Die Objekte werden in Höhe des Geländes an den x-y-Koordinaten platziert. Da die Pflanzen später vielleicht einmal parametrisierte Größen haben sollen (je nach Alter), wird jede Pflanze zuerst mit 1 x 1 x 1 Meter er-

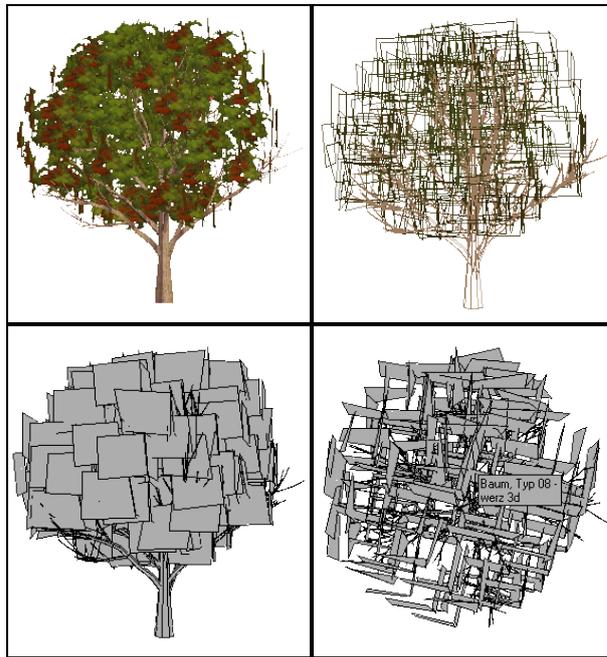


Abbildung 19: Beispiel für ein dreidimensionales Modell eines Baums: Texturierte Darstellung (links oben), Gitternetz (rechts oben), Drahtgitter mit verdeckten Kanten einmal von vorne und vorn oben (unten)

stellt. Erst beim Generieren der X3D-Datei wird für die Pflanze eine Skalierung angegeben. Bei den Gegenständen ist das nicht notwendig, da die Größe nur von den Herstellerangaben definiert wird.



Abbildung 20: Das Gelände mit Wegen und den platzierten Pflanzen.

3.4 Das Projekt

Arborata kann auf Wunsch eine X3D-Projektdatei generieren, die alle Informationen über Gelände, Wege und Objekte beinhaltet. Es werden auch Metainformationen, wie die Namen der Pflanzen gespeichert. Diese Projektdatei kann bei der nächsten Nutzung von Arborata eingelesen und normal bearbeitet werden. Obwohl alle Informationen gespeichert sind, liegen die Texturen und Objektbilder nicht in der Datei vor, so dass das Anschauen der X3D-Datei ohne Verbindung zu Arborata (Internetverbindung) nur die reinen Geometrien anzeigt.

Um eine Unabhängigkeit zu gewährleisten, könnte eine Exportfunktion dafür sorgen, dass auch alle benötigten Texturen und Bilder mit der Projektdatei heruntergeladen werden.



Abbildung 21: Eine Ansicht auf das fertige Gelände mit Wegen, Pflanzen und Gegenständen.

3.5 Was fehlt

Da die Implementierung nur zum Prüfen des Konzepts erstellt wurde, enthält sie nur die notwendigsten Funktionen. So fehlen beispielsweise noch einige Elemente, die den Arbeitsablauf beschleunigen und auch verbessern können. Die meisten dieser Funktionen müssen in der Flashoberfläche eingebaut werden:

1. Pinsel: Umschalten zwischen runden/eckigen Pinselformen und auf Airbrush-Pinsel.
2. Bei der Geländeerstellung würde sich eine Pipette anbieten, um einen verwendeten Höhenwert aufzunehmen.
3. Das Nachbearbeiten eines bereits erstellten Geländes.
4. Das Nachbearbeiten bereits erstellter Wege/Plätze.
5. Beim Platzieren der Objekte alle vorher platzierten Objekte anzeigen.
6. Beim Zeichnen der Wege alle anderen Wege im Gelände anzeigen, sowie einem Radierer zur Korrektur des Weges.
7. Bei der Platzierung von Objekten sollten vorhandene Objekte gelöscht, verschoben und rotiert werden können.

Ausserdem gibt es eine Reihe von Funktionen, die für den ein oder anderen Benutzer interessant und nützlich sind. Wenn das Programm zukünftig weiterentwickelt werden sollte, können diese Funktionen eine wichtige Rolle spielen. Die einzelnen Methoden und wie sie in das Programm integriert werden können, wird in Kapitel 5 näher erläutert.

4 Bewertung

Um den Prototyp in seiner Funktionalität zu prüfen, werden einige Benutzertests durchgeführt. Da das Programm Arborata noch nicht vollständig fertig entwickelt ist, werden die Tests nicht nach normalen Usabilitytests ablaufen, sondern sollen nur einen Eindruck vermitteln, ob die Entwicklung des Systems tatsächlich Vorteile in der Benutzerfreundlichkeit bietet, wie erwartet wurde. Der Prototyp selbst ist nicht in allen Funktionalitäten ausgereift, trotzdem kann man schon jetzt erkennen, ob die Designentscheidungen für dieses Projekt geeignet sind, oder ob einige Funktionen für den Benutzer unverständlich bleiben. Für die Benutzertest werden einige Probanden gewählt, deren Kenntnisstand mit Computern nicht ausschlaggebend ist, da die Abläufe des Programms trotzallem neu sind. Das Verständnis der Arbeitsabläufe und Erwartungskonformität des Programms sind dabei Hauptaugenmerk der Tests.

Im Folgenden wird der Ablauf der Benutzertests beschrieben und ihre Ergebnisse vorgestellt. Dazu gehören Probleme der Programmnutzung, sowie mögliche Verbesserungen.

4.1 Die Benutzertests

Zuerst wird das System im Eigenversuch getestet, da die benötigten Abläufe bekannt sind. Dabei wird versucht eine bestehende Gartenvorlage so originalgetreu wie möglich mit dem Programm nachzugestalten. Danach werden weitere Benutzer, die das Programm zum ersten Mal sehen, versuchen diese Vorlage zu reproduzieren. Die Korrektheit der Pflanzen und Objekte und ihre genaue Positionierung ist dabei nicht relevant. Es ist vielmehr Ziel, die Arbeitsweise des Programmes auf Anhieb zu verstehen und einen realistischen Eindruck von dem gestalteten Garten zu bekommen.

Ausgangspunkt der Tests stellen zwei Referenzbilder eines Gartens dar von der Webseite "Baumschule und Gartengestaltung Gartenarchitekt Gerhard Karl [5].



Abbildung 22: 1. Referenzbild [5]



Abbildung 23: 2. Referenzbild [5]

4.1.1 Eigenversuch

Mit dem Programm soll versucht werden, schrittweise den Garten aus den Abbildungen 22 und 23 nachzubauen. Dabei wird für jeden Arbeitsschritt ein Screenshot gemacht, auch um die Arbeitsweise von Arborata vorzuführen. Diese Screenshots befinden sich in voller Größe auf der beiliegenden CD, da die hohe Zahl an Bildern die Dokumentation der Ergebnisse nur erschweren würde. Gezeigt werden Ansichten des fertigen Gartens, die den Referenzbildern nachempfunden werden.

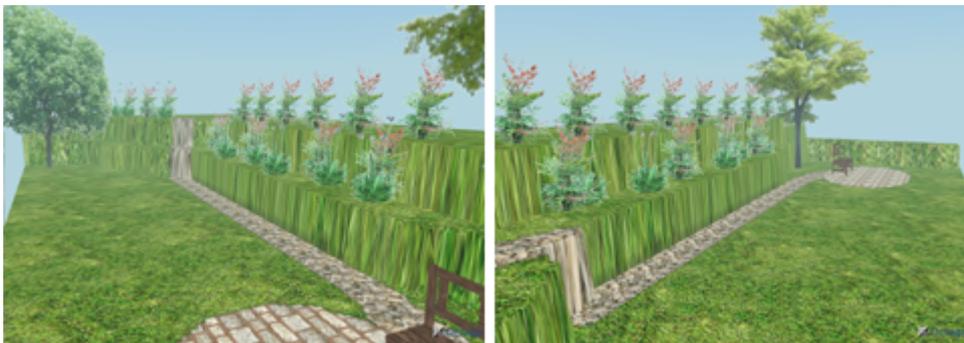


Abbildung 24: Ergebnis nach dem Erstellen des Geländes und dem Platzieren aller Objekte



Abbildung 25: Ergebnis nach dem Versuch, die Mauern nachzubilden

Das Gelände wurde auf die Größe 12x6 Meter geschätzt und auch so erstellt. Das Malen der Höhenkarte wurde bereits mehrmals probiert und war so kein großes Hindernis, auch wenn hier auffiel, dass eine Radierfunktion oder ein Rückgängigmachen sehr praktikalbel wäre. Schwieriger war das Einzeichnen der Wege, da die Grauwertunterschiede zwischen 0 Meter und 0,5 Meter so gering sind, dass es schwer ist, die genaue

Geländeformation auszumachen. Das gleiche Problem tritt auch beim Platzieren der Objekte auf, wo die einzelnen Höhenstufen manchmal schwer zu erkennen sind.

Es wurde noch versucht, das Mauerwerk nachzubilden, was die verschiedenen Höhenstufen des Gartens abgrenzt, dazu wurde unter "Wege/Plätze" eine Ebene erstellt, die genau auf den Höhengrenzen verlief. Das Ergebnis sieht nicht wirklich nach einer Mauer aus, trifft aber eher den Gesamteindruck des Gartens.

Zusammengefasst war der Gartennachbau kein echtes Problem für den Arborata-geübten Benutzer, allerdings gibt es trotzdem einige Funktionen, die die Benutzung vereinfachen würden.

4.1.2 1. Proband

Es wurde mit einem Gelände der Größe 10x20 Meter gestartet, dessen Höheneinstellung recht schnell von der Hand ging. Der erste Versuch missglückte jedoch, weil der Knopf zu Speichern der Höhenwerte übersehen wurde und stattdessen versuchte der Proband über das obere Menü direkt zu den Pflanzen zu gelangen. Gleich zu Anfang wurde die fehlende Pinselgrößenvorschau bemängelt und ein Fehler beim Erstellen des Geländes und der Wege: Verlässt man das Zeichenfeld bei gedrückter Maus und lässt dann die Maus los, so kann es passieren, dass beim erneuten Überfahren der Zeichenfläche mit der Maus unbeabsichtigt weitergezeichnet wurde. Der Fehler konnte nach den Tests behoben werden.



Abbildung 26: Ansichten des von Proband 1 erstellten Gartens

4.1.3 2. Proband

Die Erstellung des Geländes mit 5x10 Metern war recht schwierig. Da es keinen Radierer gab oder eine Funktion zum Rückgängigmachen der letzten Aktion, wurde versucht mit einem angenäherten Höhenwert die vermalteten Teile zu korrigieren. Dabei übersieht man allerdings einzelne Pixel, deren Grauwerte schwer zu unterscheiden sind. Dadurch entstanden im fertigen Gelände einzelne Spitzen.

Da durch das pixelbasierte Malen oft unregelmäßige Geländeformationen entstanden, fragte der Proband nach einer Funktion um Formen zeichnen zu können, wie unterschiedlich breite Linien oder (Bezier-) Kurven. Auch Kreise und Rechtecke könnten hier hilfreiche Werkzeuge sein.

Beim Speichern des Geländes dauert es oft einen Moment bis die Daten an den Server geschickt wurden und die nächste Seite dargestellt wurde. Das führte zu der Annahme, dass das Speichern nicht funktionierte oder man etwas falsch gemacht hätte. Erwartet wurde stattdessen eine Meldung mit der Bitte um Geduld während des Speicherns oder zumindest eine Anzeige einer kleinen Sanduhr.

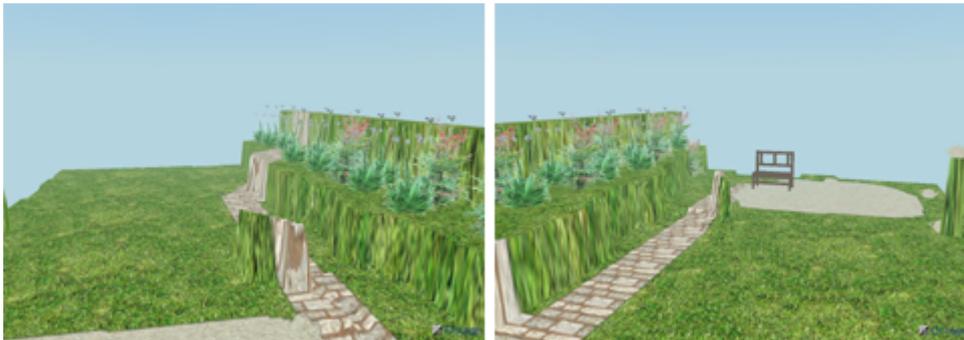


Abbildung 27: Ansichten des von Proband 2 erstellten Gartens

4.1.4 3. Proband

Obwohl alle Probanden gebeten wurden, den Bildvorlagen so nahe wie möglich zu kommen, probierte der dritte Tester alle Funktionen von Arborata aus.

Es wurde ein Gelände mit 15x11 Metern erstellt mit einem Hügel (ca. 2 Meter Höhe) in der Mitte. Das Malen der Höhen bereitete etwas Probleme. Auch hier wurde um eine Vorschau der Pinselgröße und beim Zeichnen die Anzeige der Größe der Pinselspitze gebeten. Der Proband erwartete, dass der Hügel erstellt wird, indem mit einem Pinsel ein (nicht-ausgefüllter) Kreis gezeichnet wird. Nach einer kurzen Erklärung, wie das Zeichnen

funktioniert, war das Erstellen des Geländes leicht abzuschliessen. Aber auch hier wurde das Fehlen eines Rasters und eines Lineals an den Geländekanten angemerkt.

Es wurden zwei Wege-Ebenen erstellt, davon eine Ebene mit einer animierten Wassertextur, die zum Testen der Animation implementiert wurde. In der dreidimensionalen Ansicht lag die Wasserebene über dem Weg, obwohl der Proband hier ein automatisches Erstellen von Brücken erwartet hat. Auch hier wurde auf eine fehlende Weg/Weg-, Weg/Objekt- und Objekt/Objekt-Kollisionserkennung hingewiesen.

Um einen waldähnlichen Eindruck zu erhalten, wurden 16 Bäume gepflanzt. Der X3D-Browser ruckelte bei der Navigation durch das Gelände etwas, da die Anzahl der Objekte und die animierten Texturen eine große Belastung darstellten.

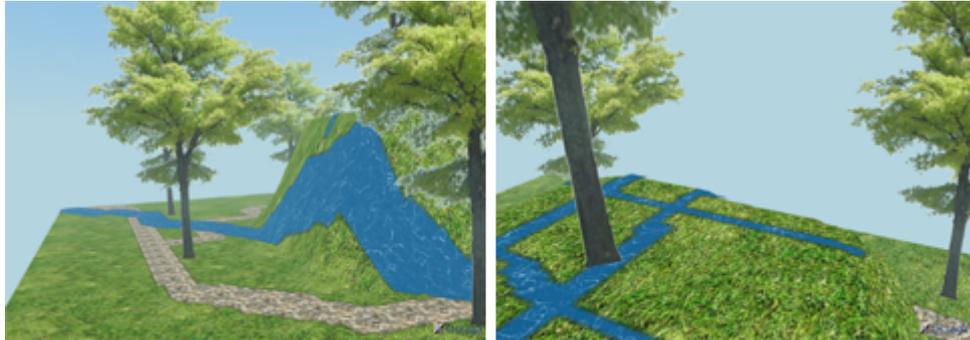


Abbildung 28: Ansichten des von Proband 3 erstellten Gartens

4.1.5 Fragen an die Probanden

Im Anschluß an den Test, wurden den Probanden noch einige ergänzende Fragen gestellt, um zu sehen, wie der Prototyp wahrgenommen wird.

- **Ist die Geländeerstellung mit einer Heightmap intuitiv?**

Alle Tester finden eine Höhenkarte generell gut und auch verständlich, jedoch sollte der Farbverlauf nicht von Schwarz nach Weiß, sondern von Grün nach Braun gehen, wie in Landkarten üblich. Ausserdem sollte man vor dem Erstellen der Karte die maximale Höhe angeben können. Beim ersten Anschauen der Karte sollte ein kurzer Hinweis erfolgen, dass es sich um eine Höhenkarte handelt und Tips wie die Ansicht zu verstehen ist. Ein Proband meinte, er hätte das schwarze Gelände zunächst für einen Fehler gehalten, wo ein Bild nicht richtig geladen wird. Eine Grün/Braune-Ansicht würde dem Abhilfe schaffen.

Um die Größenverhältnisse besser einschätzen zu können, wünschten sich alle Probanden eine Art Raster auf dem Gelände, welchen ein- und weggeblendet werden kann. Ausser einer Vorschau der Pinselgröße, wurde auch um die Möglichkeit gebeten, Geraden oder Kurven zeichnen zu können und Vorgänge rückgängig machen zu können.

Bei einigen Speichervorgängen war unklar, ob das Programm tatsächlich arbeitete, hier sollte ein Hinweis erfolgen, dass gerade gespeichert wird.

- **Erscheint in diesem Zusammenhang das Zeichnen der Wege und Platzieren der Objekte logisch?**

Nach dem Erstellen des Geländes war allen klar, wie das Zeichnen der Wege und das Platzieren der Objekte funktionierte, hier müssen nur noch bereits platzierte Wege/Objekte eingeblendet werden, was Fehlplatzierungen auf Wegen ausschliessen würde. Einige wünschten sich allerdings noch eine extra Meldung, wenn ein Objekt auf ein Weg gesetzt wird.

- **Glauben Sie, dass die X3D-Vorschau tatsächlich einen realistischen Eindruck von einem Garten wiedergeben kann?**

Zwei der Probanden kannten bereits andere Programme, wie Raumplaner, die nur eine 2D-Ansicht von oben darstellen. Die dreidimensionale Ansicht des X3D-Players wurde als sehr übersichtlich und - gerade die Texturen - als realistisch gesehen. Nur die Bäume wurden auch als dreidimensionales Objekt gewünscht.

Die Standard-Navigation bereitet Anfängern Probleme. Man bewegt sich mit gedrückter Maustaste etwa 1m/s (Voreinstellung), was manchmal sehr langsam wirkt. Die Geschwindigkeit ist im Nachhinein noch

korrigiert worden auf 2 m/s. Die Erwartungshaltung der Probanden bezüglich der Navigation entsprach dem eines Computerspiels, wobei mit der Tastatur vor- und rückwärts gelaufen werden kann und die Maus die Richtung angibt. Mit dem Octaga Player kann diese Einstellung gemacht werden, jedoch entspricht diese Navigation nicht dem X3D-Standard.

- **Gibt es eine Funktion, die Sie am meisten vermissen? Wenn ja, welche und warum?**

Das Testgelände beinhaltet eine Treppe, die mit dem aktuellen Prototyp nur mit großen Mühen eingebaut werden kann. Man müsste beim Erstellen des Geländes die einzelnen Höhen pro Stufe einzeichnen. Keiner der Probanden wollte das auf diese Art umsetzen und wünschte sich eine Funktion, die Treppen als platzierbare Objekte einsetzt.

Obwohl das Referenzgelände keinen Teich oder andere Gewässer hat, suchten die Probanden nach einer Möglichkeit, Wasser in das Gelände zu setzen. Testweise wurde daraufhin eine animierte Wassertextur in die Wege-/Plätze-Erstellung implementiert. Das Ergebnis sieht auf flachem Terrain sehr glaubwürdig aus, nur auf unebenem Gelände entstanden unrealistische Effekte, wo das Wasser am Berghang entlang lief, statt herunter. Für einfache Gewässer ist das eine schnelle Lösung, aber gerade für fließende Gewässer ungeeignet.

Weiterhin fiel auf, dass die Probanden die Gewässerfunktion beim Vorgang der Geländeerstellung vermuteten. Es wurde erwartet, dass ein Gewässer entsteht, wenn man das Terrain auf unter 0 Meter absenkt. Allerdings ist das trotz Erwartungshaltung nicht immer der Fall (z.B. Bergseen) und vor allem in künstlich angelegten Gärten seltener.

- **Weitere Ideen und/oder Bemerkungen?**

Für die Ansicht im X3D-Browser wurden mehr Funktionen gewünscht. Beispielsweise ein HUD (Head Up Display), welches eine Liste aller Objekte zeigt und die Kamera springt beim Anklicken auf das entsprechende Element. Das ist allerdings nur ein Beispiel, andere Ideen waren eher vage formuliert, wie z.B. ein Konstruktionsfenster. Leider konnte der Proband diese Idee nicht näher erklären.

Ebenso gewünscht war das Ein- und Ausschalten eines Rasters im Browser, um die Größenverhältnisse besser erkennen zu können. Diese Funktion ist nach den Benutzertests noch implementiert worden. Drückt man die Tasten r (für Raster) oder g (für Gitter oder Grid), wird ein rotes Gitter über die Grasebene gelegt.

Eine weitere Idee ist das Deaktivieren der Hauptmenüpunkte beim

Arbeiten auf der Flashoberfläche, da man sonst aus Versehen das Speichern seiner Einstellungen vergisst.

4.1.6 Ergebnisse der Benutzertests

Der schwierige Punkt der Geländeerstellung benötigt zwar noch sehr viele Zusatzfunktionen, um einwandfreies Arbeiten zu ermöglichen, jedoch war bei den Tests schon zu erkennen, dass Benutzer das Malen auf einer Heightmap grundsätzlich verstehen und anwenden. Natürlich gab es viele Probleme, da es sich bei Arborata um einen Prototyp handelt und die Probanden dadurch die in Kapitel 3.5 genannten Funktionen vermissten. So mussten einige Vorgänge, wie das Einzeichnen der Wege mehrmals wiederholt werden, bloß weil man sich vermalt hatte.

An den stark variierenden Geländegrößten, die für die gleichen Referenzbilder angegeben wurden (12x6, 10x20, 5x10 und 15x11 Meter), bestätigte sich der Verdacht, dass die Größenverhältnisse eines Geländes (egal ob real oder Photo) sehr schwer eingeschätzt werden können.

Die Probanden zeigten sich sehr begeistert über die dreidimensionale Darstellung und probierten gerne neue Funktionen in Arborata aus. Wenn die Benutzerfreundlichkeit durch die Implementierung der gewünschten Funktionen verbessert wurde, so sinkt die Hemmschwelle das Programm öfter zur Planung zu benutzen.

5 Ergebnisse

In Kapitel 2.1.2 sind die verschiedenen Prioritäten der Funktionen genannt. Wie dort gefordert kann mit Arborata ein **Gelände** erstellt und gelöscht werden. Das Bearbeiten des Geländes ist dagegen aus zeitlichen Gründen nicht umgesetzt worden, da es momentan nicht möglich ist, das erstellte Pixelbild wieder zu vektorisieren.

Gleiches gilt für **Wege/Plätze**, die auf verschiedenen Ebenen erstellt werden können. Da bei vielen Ebenen und/oder großem Gelände der X3D-Browser ruckelt, sollten zukünftige Weiterentwicklungen diesen Punkt noch einmal überdenken. Allerdings kann mit dieser Methode für die einzelnen Ebenen eine Kostenschätzung erfolgen.

Eine Suche über die **Objekte (Pflanzen und Gegenstände)** ist implementiert, aber von den Benutzern nicht getestet worden, da bei derzeitigem Objekumfang in der Datenbank keine Suche notwendig ist. Die Pflanzen und Gegenstände können auch auf einen Merktzettel gesetzt werden, allerdings hat das keine weiteren Auswirkungen. Der Zugriff auf den Merktzettel müsste erst noch implementiert werden. Über eine Flash-Oberfläche können die Objekte auf dem Gelände platziert werden. Die Objekte können sowohl als 2D- als auch als 3D-Modell dargestellt werden, das ist allein von dem Eintrag in der Datenbank abhängig. Die Pflanzen können auch wieder aus dem Gelände entfernt werden.

Gewässer sind nicht implementiert, jedoch sind einige Tests mit animierten Texturen gemacht worden. Die Texturen wirken in ebenen Gelände sehr realistisch, erzeugen aber an Berghängen ungewünschte unrealistische Effekte.

Die Funktion **Treppen** in Arborata zu platzieren, ist nicht umgesetzt worden. Im Bereich der Geländeerstellung könnte über Angabe der Breite und des Start- und Endpunktes der Treppe, ein platzierbares Objekt generiert werden, welches automatisch an die bestimmte Position gesetzt wird.

Mit Hilfe der Wege/Plätze-Funktion lässt sich auch die unterschiedliche **Bodenbeschaffenheit**, wie Rindenmulch, Erde, Grad, etc., simulieren. Allerdings sind Flächenbepflanzungen, die eine stärker erhabene Form aufweisen, damit nicht möglich.

Steingärten lassen sich zum Teil mit platzierbaren Objekten, z.B. ein Stein als Gegenstand, umsetzen. Umfangreichere Steingartenkonzepte, wie Mauerbepflanzungen sind dagegen nicht möglich.

5.1 Ausblick

Im Laufe der Diplomarbeit entstanden durch Gespräche mit Aussenstehenden oder durch andere Programme sehr unterschiedliche Ideen, wie Arborata erweitert werden könnte und sollte.

5.1.1 XML als Datenbank

Statt einer MySQL-Datenbank würde sich eine XML-Datenbank anbieten. Zum einen, da X3D auch schon zur XML-Familie gehört und zum anderen kann Flash nicht direkt auf MySQL oder andere Datenbanksysteme zugreifen, aber auf XML-Dateien. Es wäre nicht mehr nötig alle Informationen per URL an Flash weiterzureichen, sondern man kann von Flash aus auf die benötigten Dateien zugreifen.

Da PHP jedoch keine entsprechenden XML-Funktionen zur Verfügung stellt, sondern nur einen XMLReader und XMLWriter, die kein DOM unterstützen, ist die XML-Datenbank in dieser Diplomarbeit noch nicht zum Einsatz gekommen.

5.1.2 Gewässer und Wasserläufe

Gewässer sind eine besondere Herausforderung gerade wenn es sich um fließendes Wasser handeln soll. Auch hier gibt es verschiedene Möglichkeiten:

Brunnen: Brunnen sind meist dekorative Objekte mit eingebauter Pumpe und Wasserspender. Einmal mit Wasser aufgefüllt, benötigen sie nichts weiter als einen Stromanschluss.

Brunnen sind von allen Gewässern am einfachsten im Programm zu realisieren, sie können einfach als Gegenstände im Gelände platziert werden. Dabei wäre auch eine Animation des Wassers als Textur oder sogar des Brunnens als Objekt möglich.

Teichschalen/-becken: Teichschalen/-becken bestehen aus einer fertigen Plastikform, die in das Gelände eingesetzt wird. Mit Wasser aufgefüllt und mit Pflanzen begrünt, stellt es eine einfache Lösung für Teiche dar.

Im Programm wären sie ganz ähnlich wie die Brunnen zu platzieren und können damit auch animiert werden. Allerdings muss bei Teichschalen darauf geachtet werden, dass das Gelände darunter abgesenkt werden muss.

Folienteiche: Möchte man Teiche in einer freier Form, so muss man auf Teichfolie zurückgreifen. Der gewünschte Teich wird ausgegraben und mit der wasserdichten Folie ausgelegt. Meist wird noch dekorativer Kies über die Folie gestreut, so dass der Teich natürlich wirkt.

Die Folienteiche können wie die Wege/Plätze gemalt werden und als Textur auf das Gelände gelegt werden. Dabei sollte das Gelände wie bei den Teichschalen abgesenkt werden.

Bachläufe: Mit Teichfolie oder auch Plastikmodulen kann ein Bachlauf im Gelände verlegt werden. Die Plastikmodule sind ineinandersteckbare Steinimitate, die den Verlauf des Baches definieren. Auch hier gilt, dass die Teichfolie mehr Freiheiten bietet.

Diese Funktion ist schon weit schwieriger in diesem Programm umzusetzen, denn egal wie der Benutzer dem Bachlauf legt, so muss beachtet werden, dass er bergab fließt.

Zur Berechnung der Flussrichtung des Wassers können Partikelsysteme eingesetzt werden. X3D enthält auch eine Komponente dafür, aber auf Grund der enormen Rechenlastigkeit, die damit auf den Clientrechner zu käme, ist das nicht zu empfehlen.

5.1.3 Jahreszeiten

Die Simulation von Jahreszeiten ist wichtig, um zu sehen, ob der Garten auch während der Wintermonate noch ansehnlich ist und nicht vollkommen grau wirkt. Meistens wünschen Anwender solcher Software zu sehen, welche Pflanzen zu bestimmten Monaten in Blüte sind oder Früchte tragen.

Diese Simulation erfordert allerdings eine parametrisierte Pflanzengenerierung, wobei für jede Pflanze Blüten und Früchte für die jeweiligen Monate bekannt sein müssen. Einfacher wäre es, markante Zeitpunkte im Jahr zu wählen, wie z.B. die Jahreszeitenanfänge (21.03., 21.06., 23.09., 21.12.)

Eine weitere einfache Möglichkeit sind einige wenige Wintermodelle zu erstellen für alle sommergrünen Pflanzen, während alle wintergrünen ihre originalen Modelle behalten.



Abbildung 29: von links nach rechts: Frühling, Sommer Herbst, Winter. Aus einem Eifelpanorama von Steffen Heinz

5.1.4 Tageszeiten

Zur Planung des Lichteinfalls und des daraus folgenden Schattenfalls, ist es wichtig die Schatten der Pflanzen und Gegenstände auf den Boden zu werfen. So kann beispielsweise ein großer Baum durch sein Laub den gesamten Garten verdunkeln. Um solche Überraschungen zu verhindern ist die Simulation der Tageszeiten wichtig. Es müsste dann eine Sonnenstandsrechnung erfolgen. Einige Programme bieten dies und auch eine Simulation für die Nacht an, um zu sehen, ob die Lampen den Garten genug ausleuchten, was eine Speicherung der Lampenspezifikation erfordern würde.

Der Schattenfall für den Garten kann in eingeschränkter Form genutzt werden, so dass zwar alle Objekte einen Schatten auf den Boden werfen, aber ein gegenseitiges Verdunkeln der Objekte ist nicht unbedingt notwendig.

Schatten sind weit schwieriger zu realisieren, da sie in X3D nicht mit generiert werden, müssen sie extra implementiert werden. Das wäre möglich mit Java oder ECMAScript über den Script-Node, Das Skript kann auf die einzelnen Elemente der X3D Datei zugreifen.

5.1.5 Pflanzenalter

Das Pflanzenalter spielt besonders bei dem Aufbau des Gartens eine wichtige Rolle, da man die Pflanzen nicht in jeder Größe kaufen und anpflanzen kann, so sind 5-jährige Bäume unter Umständen extrem teuer zu kaufen und je nach Größe auch sehr schwer anzupflanzen. Außerdem besteht das Risiko, dass der Baum den Umzug nicht überlebt. Also interessiert viele Gärtner nach wie viel Jahren sie mit einem kleinen Baum, das gewünschte große Ergebnis erhalten.

5.1.6 Duftkarten

Eine weitere Idee sind sogenannte Duftkarten, die auf einer Gartenübersicht die von Pflanzen ausströmenden Düfte je nach Stärke als farbigen Kreis eingezeichnet haben. Unterschiedliche Düfte, wie süß oder fruchtig, können mit unterschiedlichen Farben gekennzeichnet werden.

5.1.7 Online-Service

Arborata bietet viele Möglichkeiten für einen kommerziellen Einsatz auch nach Abschluß des Projekts. Dabei gibt es verschiedene Konzepte, so könnte es beispielsweise als Gartenplanersoftware verkauft werden. Interessanter dagegen ist die Möglichkeit eines Online-Services, der es Garten- und Landschaftsarchitekten, Baumschulen und/oder Baumärkte, usw. ermöglicht, ihre Produkte darin zu präsentieren. Es würde so als Werbepattform

dienen, die aber dem Benutzer den Vorteil einer kostenlosen Nutzung und größt möglicher Aktualität bietet. Dem Kunden kann dann auch die Möglichkeit geboten werden, das Angebot auf Dienstleister in seiner Umgebung einzuschränken.

Angemeldete Nutzer können sich über Newsletter über anstehende Arbeiten an ihren ausgewählten Pflanzen informieren lassen oder einen Gartenkalender erstellen lassen, der alle nötigen Informationen über ihre Pflanzen anbietet, wie zum Beispiel, wann Bäume zurückgeschnitten werden sollten und welche besondere Pflege sie benötigen.

Für Allergiker hingegen wäre eine Allergikerwarnung von großem Nutzen, die bei der Auswahl der Pflanze darauf hinweist, wenn die Pflanze für ihn problematisch ist und wann die Pollenflugzeit ist.

5.2 Fazit

Auf Grund der Arbeit eines Landschaftsarchitekten entstanden Ideen für eine einfache Arbeitsoberfläche, mit der ein Garten und darin enthaltene Elemente aufgebaut werden können. Daraus entstand im Rahmen dieser Arbeit der Prototyp Arborata, der bereits alle nötigen Grundfunktionen besitzt. Der Benutzer kann ein rechteckiges Gelände erstellen und die Geländeform mit Hilfe einer Höhenkarte manipulieren. Es können beliebige Wege auf das entstandene Gartengelände gelegt werden, was ebenfalls mit der Metapher umgesetzt wurde. Über einen einfachen Mausklick können auch Objekte, wie Pflanzen oder andere Gegenstände, wie Sitzbänke oder Gartenfiguren, platziert werden.

Alles, was der Benutzer in seinem Arborata-Garten erstellt, kann er mit einem X3D-Browser in einer dreidimensionalen Darstellung betrachten und über Tastatur oder Maus hindurchnavigieren.

Die Benutzertests zeigen, dass Arborata eine gute Grundlage für die komplette Entwicklung eines Gartendesignwerkzeugs bietet, da die Arbeit mit einfachen Grundfunktionen leicht verständlich ist und kein Vorwissen benötigt.

Natürlich fehlen noch viele Funktionen bis ein vollständiger Garten erstellt werden kann, aber die Grundkonzepte von Arborata, besonders die Metapher des "Geländemalens" zeigen einen Weg vor, wie ein Garten ohne große Konstruktionsvorgänge in kürzester Zeit erstellt werden kann.

Anhang A - Datenbank

Auflistung aller Tabellen mit ihren Feldern und was sie beschreiben. Die Felder für `plants`, die die Eigenschaften von Pflanzen beschreiben wurden aus dem BRUNS Pflanzen Katalog übernommen.

```
plants(  
  id: integer Eindeutiger Schlüssel  
  latin_name: varchar Lateinische Name der Pflanze  
  name: varchar Deutscher Name der Pflanzen  
  synonym_1: varchar Deutsches Synonym der Pflanze  
  synonym_2: varchar Deutsches Synonym der Pflanze  
  synonym_3: varchar Deutsches Synonym der Pflanze  
  description: blob Beschreibung der Pflanze  
  stueck_pro_m2: integer Anzahl der Pflanzen pro qm2  
  herkunft_gebiet: varchar Ursprüngliche Herkunft  
  herkunft_hoehe: varchar Höhe der Herkunftsregion  
  herkunft_klima: varchar Klimazone des Herkunftsgebiets  
  blatt_austriebzeit: varchar Zeit des Austriebs  
  blatt_farbe: integer Foreign Key zu color.id  
  blatt_form: varchar Beschreibung der Blattform  
  blatt_groesse: integer Blattgröße in Zentimeter  
  blatt_oberflaeche: varchar Beschreibung der Oberfläche  
  blatt_rand: varchar Beschreibung des Blattrandes  
  blatt_spitze: varchar Beschreibung der Blattspitze  
  blatt_herbstfaerbung: integer Foreign Key zu color.id  
  bluete_bestaeubung: varchar Bestäubungsart  
  bluete_groesse: integer Größe der Blüte in Zentimeter  
  bluete_duft: varchar Beschreibung des Blütenduftes  
  bluete_farbe: integer Foreign Key zu color.id  
  bluete_fuellung: varchar Füllung der Blüte  
  bluete_zeit_von: integer Monatsnummer des Blühbeginns  
  bluete_zeit_bis: integer Monatsnummer des Blühendes  
  frucht_art: varchar Art der Frucht  
  frucht_farbe: integer Foreign Key zu color.id  
  frucht_form: varchar Form der Frucht  
  frucht_geschmack: varchar Geschmacksbeschreibung  
  frucht_oberflaeche: varchar Oberfläche der Frucht  
  frucht_reife_von: integer Monatsnummer des  
  frucht_reife_bis: integer Monatsnummer  
  frucht_geniessbarkeit: varchar Geniessbarkeit der Frucht  
  rinde_art: varchar Art der Frucht  
  rinde_farbe: integer Foreign Key zu color.id  
  wuchs_breite: integer Maximale Breite der Pflanze
```

```

wuchs_form: varchar Wuchsform der Pflanze
wuchs_hoehe: integer Maximale Höhe der Pflanze
wuchs_staerke: varchar Wuchsgeschwindigkeit
wuchs_wurzel: varchar Wurzeltypus
pflege_1: blob Pflegehinweise für Januar
pflege_2: blob Pflegehinweise für Februar
pflege_3: blob Pflegehinweise für März
pflege_4: blob Pflegehinweise für April
pflege_5: blob Pflegehinweise für Mai
pflege_6: blob Pflegehinweise für Juni
pflege_7: blob Pflegehinweise für Juli
pflege_8: blob Pflegehinweise für August
pflege_9: blob Pflegehinweise für September
pflege_10: blob Pflegehinweise für Oktober
pflege_11: blob Pflegehinweise für November
pflege_12: blob Pflegehinweise für Dezember
x3d_content: blob X3D-Notation für die Pflanze
x3d_width: float Skalierungsfaktor für Breite
x3d_height: float Skalierungsfaktor Höhe
x3d_length: float Skalierungsfaktor Länge
preis_pro_stueck:float Preis pro Stück
)

items(
  id: integer Eindeutiger Schlüssel
  name: varchar Name des Gegenstandes
  description: varchar Beschreibung des Gegenstandes
  item_type: integer Foreign Key zu item_type.id
  width: float Breite des Gegenstandes
  height: float Höhe des Gegenstandes
  length: float Länge des Gegenstandes
  x3d_content: blob X3D-Notation für den Gegenstand
  x3d_width: float Skalierungsfaktor für Breite
  x3d_height: float Skalierungsfaktor Höhe
  x3d_length: float Skalierungsfaktor Länge
  preis_pro_stueck:float Preis pro Stück
)

surfaces(
  id: integer Eindeutiger Schlüssel
  name: varchar Name des Oberfläche
  description: varchar Beschreibung der Oberfläche

```

```

        width: float Skalierungsfaktor der Kachelbreite
        length: float Skalierungsfaktor der Kachellänge
        pattern_url: varchar Dateiname der Textur
        animiert: char Boolean, ob die Textur animiert ist
        preis_pro_qm: float Preis pro Quadratmeter
    )

themes(
    id: integer Eindeutiger Schlüssel
    parent_id: integer Schlüssel zu themes.id
    name: varchar Name des Themas
    description: varchar Beschreibung des Themas
)

themes_objects(
    theme_id: integer Foreign Key zu themes.id
    object_table: varchar Name der Tabelle mit dem Objekt
    object_id: integer Foreign Key zu [object_table].id
)

colors(
    id: integer Eindeutiger Schlüssel
    parent_id: integer Schlüssel zu colors.id
    name: varchar Farbbezeichnung
)

item_type(
    id: integer Eindeutiger Schlüssel
    bezeichnung: varchar Bezeichnung der Gegenstandsgruppe
)

default_terrains(
    id: integer Eindeutiger Schlüssel
    name: varchar Bezeichnung für das Gelände
    description: varchar Beschreibung des Geländes
    terrain_size_x: integer Geländebreite in Meter
    terrain_size_y: integer Geländelänge in Meter
    terrain_data: mediumblob Höhenwerte des Geländes
)

```

```

session(
    id: varchar SESSIONID als eindeutiger Schlüssel
    name: varchar Bezeichnung für das Gelände
    description: varchar Beschreibung des Geländes
    terrain_size_x: integer Geländebreite in Meter
    terrain_size_y: integer Geländelänge in Meter
    terrain_data: mediumblob Höhenwerte des Geländes
)

session_objects(
    id: varchar SESSIONID als Foreign Key zu session.id
    object_table: varchar Name der Tabelle mit dem Objekt
    object_id: integer Foreign Key zu [object_table].id
    object_counter: integer Anzahl pro Pflanze
    x_position: integer x-Koordinate auf dem Gelände
    y_position: integer y-Koordinate auf dem Gelände
)

session_surfaces(
    id: varchar SESSIONID als Foreign Key zu session.id
    surface_id: integer Foreign Key zu surfaces.id
    surface_counter: integer Anzahl pro Textur
    surface_order: integer Sortierung der Texturebenen
    surface_data: mediumblob Bitmaske
)

```

Anhang B - Auf der CD

Die beiliegende CD enthält verschiedene Dateien, die Laufe der Arbeit entstanden oder gesammelt wurden. Hier folgt nun eine Auflistung der Dateien/Ordner und was sie beinhalten:

arborata enthält alle für den Webserver benötigten Dateien, um das Arborata-Projekt zu starten. Es enthält PHP-Skripte, Bilder, Flash-Applikationen und Datenbankaufbau.

arborata.projekte stellt die Testprojekte aus Kapitel 4.1 zur Verfügung. Diese Projekte können nur unter Arborata geladen werden, da sonst die Texturpfade nicht richtig aufgelöst werden.

flash.applikationen, die für Arborata erstellt wurden. Dazu gehören die FLA-Dateien aus Flash 8 und die daraus entstandenen SWF-Dateien.

programme, die von Arborata benötigt werden: XAMPP 1.5.5 für Windows und für Linux, Flash Player und Octaga X3D Player.

screenshots beinhaltet mehrere Ordner mit den Screenshots aus den Benutzertests. Für jeden Probanden ist ein Ordner angelegt mit den Ergebnisscreenshots in voller Größe (1280x906).

x3d.beispiele aus der Ausarbeitung: HelloWorld in VRML-Syntax und XML-Syntax. Der Octaga Player kann beide Dateien darstellen.

Literatur

- [1] J. Windt, "Garten 100 qm - Ausführungs- und Pflanzplanung." http://www.frischerwindt.de/Projekte/kleine_Gaerten/Garten100qm/Garten100qm_Ausfuehrungsplanung/garten100qmausfuehrungsplanung.html.
- [2] Google Inc., "Google SketchUp," Dezember 2006. <http://sketchup.google.com/>.
- [3] W. Consortium, "What is X3D?," Dezember 2006. <http://www.web3d.org/about/overview/>.
- [4] "Bryce into Mojo," Dezember 2006. www.mojoworld.org/tutorials/BryceIntoMojo/.
- [5] G. A. Karl, "Baumschule und Gartengestaltung Gartenarchitekt Gerhard Karl," Dezember 2006. <http://www.gartenarchitekt.at>.
- [6] Act-3D, "Quest3D Power Edition," Dezember 2006. <http://www.quest3d.com/>.
- [7] "Wikipedia: Adobe Flash," Dezember 2006. http://de.wikipedia.org/wiki/Adobe_Flash.
- [8] M. Gradias, Flash 5. Düsseldorf: DATA BECKER, 2nd ed., 2001.
- [9] "Wikipedia: Windows Presentation Foundation," Dezember 2006. http://de.wikipedia.org/wiki/Windows_Presentation_Foundation.
- [10] "Oekaki Shi-Painter," Dezember 2006. <http://shipainter.hokuten.net/>.
- [11] "Wikipedia: Java 3D," Dezember 2006. http://de.wikipedia.org/wiki/Java_3D.
- [12] M. Pesce, VRML Cyberspace-Welten erkunden und erschaffen. Berlin: Karl Hanser Verlag München Wien, 1st ed., 1997.
- [13] C. C. Vladimir Geroimenko, Visualizing Information Using SVG and X3D. London: Springer-Verlag, 1st ed., 2005.
- [14] "Extensible 3D: XML Meets VRML," Dezember 2006. <http://www.xml.com/pub/a/2003/08/06/x3d.html>.