Diplomarbeit

Höhere Technische Bundeslehranstalt Leonding
Abteilung für Informatik

Bad-Designer

Eingereicht von: **Philipp Auinger, 5BHIF**
Nenad Tripic, 5BHIF

Datum: **3. April 2020**

Betreuer: **Prof. Dipl-Ing. Michael Bucek**

Projektpartner: **Lang+Lang GmbH**

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorgelegte Diplomarbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Gedanken, die aus fremden Quellen direkt oder indirekt übernommen wurden, sind als solche gekennzeichnet.

Die Arbeit wurde bisher in gleicher oder ähnlicher Weise keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Leonding, am 3. April 2020

Philipp Auinger, Nenad Tripic

Declaration of Academic Honesty

Hereby, I declare that I have composed the presented paper independently on my own and without any other resources than the ones indicated. All thoughts taken directly or indirectly from external sources are properly denoted as such.

This paper has neither been previously submitted to another authority nor has it been published yet.

Leonding, April 3, 2020

Philipp Auinger, Nenad Tripic

Zusammenfassung

- *Aufgabenstellung*
  Um den Kunden der Firma Lang+Lang GmbH die Visualisierung eines für sie designten Bades zu erleichtern, ist es notwendig, ihnen eine möglichst genaue und detaillierte Darstellung anzubieten. Dies bezieht sich sowohl auf das Design, die Komponenten als auch für die Bauanleitung. Da nicht nur die Hotelketten das Modell zu Gesicht bekommen, sondern auch Monteure, die sich daran beim Zusammenbauen orientieren können. Dadurch das der Fortschritt und das fertige Bad aus verschiedenen Perspektiven angesehen werden kann, wird die Vision für alle klarer und einfacher. Da die ganze Applikation webbasiert ist, ist sie Plattform unabhängig und speicherbar für spätere Zwecke. Bisher wurde dies immer mit Präsentationsvideos umgesetzt, die jedoch sehr unflexibel sind und zeitintensiv waren. Die Probleme waren dabei, dass das Bad immer nur aus einer Perspektive zu sehen war, das Modell nicht interaktiv war und Kundenwünsche nicht sofort umgesetzt werden konnten. Der Umgang mit dem Tool wird möglichst intuitiv erfolgen und besondere Ressourcenanforderungen sind nicht präsent.

- *Umsetzung*
  Die Web-Anwendung basiert auf den gängigen Technologien HTML, JavaScript, Three.js und WEB.GL. Dies ermöglicht es über einen beliebigen Browser und beliebiges Betriebssystem darauf zu zugreifen. Die einzige Anforderung ist eine Internetverbindung. Die Wahl für die oben genannten Technologien ist darauf zurückzuführen, dass alle robust, zukunftssicher, gut dokumentiert und weitverbreitet sind. Damit eignen sie sich perfekt für die Applikation und sind ein wichtiger Beitrag dafür das die Anwendung wartbar ist und bleibt.

- *Ergebnisse*
  Die Software wurde nach Abschluss der Arbeit an das Unternehmen Lang+Lang GmbH übergeben. Demonstrationen und Tutorials an die zukünftigen Anwender wurden durch das Entwicklerteam durchgeführt. Unter folgender Adresse `http://vm85.htl-leonding.ac.at/` kann die Diplomarbeit begutachtet und verwendet werden.

Abbildung 1: Bad Designer; Frontalansicht

Abbildung 2: Bad Designer; Schrägansicht

Abbildung 3: Bad Designer; Seitenansicht

Abstract

- *Definition of the project*
  In order to present the customers of Lang+Lang GmbH a visualization of a custom-made modular bath it is necessary to have a precise and detailed illustration. This is especially important for the custom design, the components and for the construction manual. As the model will be showed to the assembler as well it is important that it is possible to build it with the manual the application provides. The progress and the complete bath can be viewed from multiple angles with the intention of making the concept clearer and simpler for all. As it is a web application it can be run on every operating system and the demonstration is saveable for later recall. Up to now this was done with presentation videos, but they were too inflexible and time-consuming. Moreover, the bath was only visible from one angle, the model was not interactive and customer requirements could not be implemented immediately. The tool is as intuitive as possible and does not require special resources.

- *Implementation*
  The web-app is based on the commonly used technologies HTML, JavaScript, Three.js and WebGL. This allows it to be run on every operating system and any modern browser. It only requires a internet connection. The mentioned technologies were uses because they are all robust, future-proof, well documented and extendable. Thanks to these attributes they are perfect fit for the application and make it easier to service.

- *Results*
  The software was delivered after the thesis to the company Lang+Lang GmbH. Demonstrations and tutorials were provided to the future users by the developing team. The work is publicly accessible on this website `http://vm85.htl-leonding.ac.at/`.

Figure 4: Bad Designer; Frontal view

Figure 5: Bad Designer; Oblique view

Figure 6: Bad Designer; Side view

Danksagungen

If you feel like saying thanks to your grandma and/or other relatives.

Auftraggeber

Lang+Lang GmbH ist ein Unternehmen mit Sitz in Leonding, das sich auf Innenarchitektur und Glasdekore spezialisiert hat, die verschiedenste Räumlichkeiten wie Bars, Hotelzimmer und etc. für den Kunden attraktiver machen. SanMod-Bäder bieten individuelle, freistehende Sanitärmodule für einen flexiblen und zeitsparenden Einbau in Gebäuden Die Anordnung der Module ist flexibel und richtet sich sowohl nach den räumlichen Vorgaben als auch nach der gewünschten Offenheit des Badezimmers zum restlichen Raum hin. Auf dieser Basiskonstruktion werden sämtliche Bestandteile eines hochwertigen Hotel-Badezimmers montiert.

LANG+LANG
werbearchitektur

Inhaltsverzeichnis

1 Einleitung **6**
1.1 Ausgangslage 6
1.2 Zielsetzung 6
1.3 Aufbau der Diplomarbeit 6
1.4 Basic Terminology 7
1.5 Related Work and Projects 7
1.6 Structure of the Thesis 7

I Verwendete Technologien **8**

2 Software **9**
2.1 HTML5 9
2.2 JavaScript 10
2.3 WebGL 11
2.4 Three.js 12
2.5 jQuery 13
2.6 NGINX 14
2.7 AutoCAD 15
2.8 Electron 17
2.9 Git 19

3 Dateien **22**
3.1 .FBX 22
3.2 .OBJ 23
3.3 .GLB 24

4 Lizenzen **25**
4.1 MIT-Lizenz 25
4.2 BSD-Lizenz 26
4.3 GNU General Public License 27

II Grundlagen und Methoden **28**

III Umsetzung **29**

5 Aufbau und Systemarchitektur **30**
5.1 Systemarchitektur – Web-Applikation 30
5.2 Systemarchitektur – Electron Anwendung 31

6 Summary **32**

A Additional Information **38**

B Individual Goals **39**

Kapitel 1

Einleitung

1.1 Ausgangslage

Lang+Lang GmbH ist ein Unternehmen mit Sitz in Leonding, das sich auf Innenarchitektur und Glasdekore spezialisiert hat, die verschiedenste Räumlichkeiten wie Bars, Hotelzimmer und etc. für den Kunden attraktiver machen. SanMod-Bäder bieten individuelle, freistehende Sanitärmodule für einen flexiblen und zeitsparenden Einbau in Gebäuden. Zurzeit werden die Bäder dem Kunden mit einem Video präsentiert. Diese Methode bringt aber Nachteile mit sich da die Bäder immer nur aus einer Perspektive gezeigt werden, die Modelle unflexibel sind, nicht interaktiv und Kundenwünsche nicht sofort umgesetzt werden können. Diese Probleme werden durch die Applikation behoben und macht es möglich die Designentwürfe für spätere Zwecke abzuspeichern.

1.2 Zielsetzung

Das Ziel ist es die Planung zu erleichtern und gegebenenfalls Änderungen sofort im Kundengespräch umzusetzen. Durch diese virtuelle Stütze können sich alle Beteiligten das fertige Bad besser vorstellen, und mögliche Missverständnisse sofort klären. Kundengespräche werden dadurch aufgewertet und effizienter.

1.3 Aufbau der Diplomarbeit

Die Diplomarbeit ist in drei Teile gegliedert. Der erste Teil besteht aus den verwendeten Technologien. Der zweite Teil geht näher auf die Grundlagen und Methoden der Arbeit ein. Der dritte und letzte Teil behandelt die Umsetzung und Realisierung des Projektes. Außerdem wird hier auf die Systemarchitektur eingegangen und die Implementierung so wie der Release der Diplomarbeit.

Abbildung 1.1: Don Knuth, the inventor of TEX

1.4 Basic Terminology

As usual the very basic terminology is briefly explained here. Most probably the explanations here only scratch a surface level. More detailed explanations of terminology goes into chapter

1.5 Related Work and Projects

Here a survey of other work in and around the area of the thesis is given. The reader shall see that the authors of the thesis know their field well and understand the developments there. Furthermore here is a good place to show what relevance the thesis in its field has.

1.6 Structure of the Thesis

Finally the reader is given a brief description what (s)he can expect in the thesis. Each chapter is introduced with a paragraph roughly describing its content.

Teil I

Verwendete Technologien

Kapitel 2

Software

2.1 HTML5

Erläuterung

HTML5 ist die fünfte Generation der Hypertext Markup Language. HTML wird hauptsächlich zur Darstellung von Websites im Internet genutzt. Die HTML-Dateien werden dabei von einem Browser abgerufen und dargestellt.

Funktionalität

Dadurch das HTML mit CSS und JavaScript um ein Design und Funktionen erweitert werden kann ist es sehr flexibel und vielseitig einsetzbar. Ganze Programme werden heute als WebApp programmiert und bieten denselben Funktionsumfang wie normale desktop-basierte Programme. Weil nur ein Browser benötigt wird um die Websites darzustellen sind sie außerdem plattformunabhängig.

Verwendung

HTML bildet das Fundament für den Bad-Designer, auf dem alle anderen Features aufbauen. Die Modelle der sanitären Anlagen werden hier geladen. Die restlichen Funktionen werden mit JavaScript implementiert.

2.2 JavaScript

Erläuterung

JavaScript ist eine Skriptsprache, die dazu dient, um HTML zu erweitern. Damit kann man auf Benutzerinteraktionen reagieren und auswerten, Inhalte verändern, generieren und nachladen. Somit sind die Seiten dynamisch und ergänzen HTML und CSS so dass sie interaktiv werden. Des Weiteren wird JavaScript auch für Server und Microcontroller verwendet. Eine bekannte serverseitige Plattform, nämlich Node.js, basiert auf der JavaScript-Laufzeitumgebung.
Die Syntax der 1995 erschienenen Sprache ähnelt der von C. Obwohl Java auf Gemeinsamkeiten vermuten lässt ist JavaScript deutlich anders und vieles abweichend implementiert. Durch die Variabilität ist es möglich objektorientiert, prozedural oder funktional zu programmieren.

Funktionalität

JavaScript bildet mit seinen umfangreichen Funktionen ein Grundgerüst für die meisten Dinge. Animationen, Berechnungen, Benutzer Interaktionen und Datenverarbeitung werden es dadurch möglich.

Verwendung

Das Laden der Modelle, Animationen, Benutzer Eingaben und Berechnungen wurden allesamt in JavaScript realisiert. JavaScript ist der wichtigste Bestandteil der Arbeit, die meisten Funktionen wären ohne diese Skriptsprache nicht möglich. Außerdem basiert die Bibliothek three.js auf dieser Sprache und erweitert sie. Mehr zu three.js wird auf den folgenden Seiten näher erläutert.

2.3 WebGL

Erläuterung

WebGL, kurz für Web Graphics Library, ist eine auf JavaScript basierende Programmierschnittstelle, die es ermöglicht 3D-Grafiken hardwarebeschleunigt im Webbrowser ohne zusätzliche Erweiterungen, Bibliotheken und Technologien darzustellen. Des Weiteren basiert WebGL auf den Spezifikationen von OpenGL ES, kurz für Open Graphics Library for Embedded Systems. Diese beschreibt eine plattform- und sprachenunabhängige Programmierschnittstelle für die Entwicklung von 3D-Computergrafiken.
Die 2011 erschienene Library ist lizenzfrei von der Khronos Group und Mozilla entwickelt worden. Mit der Zeit haben immer mehr Unternehmen die Entwicklung unterstützt, dazu zählen unteranderem Google Chrome, AMD, Ericsson, Nvidia und Opera.

Funktionalität

Dadurch das die Grafik-Bibliothek im Browser läuft, ist sie plattformunabhängig und hat unteranderem deshalb auch eine große Reichweite. Zusätzlich gestattet sie Grafikern mit den beliebten Softwarewerkzeugen wie Blender, Maya oder CopperCube zu arbeiten ohne sich um die Darstellung anschließend kümmern zu müssen. WebGL konfiguriert und verarbeitet die Modelle für den Browser.

Verwendung

WebGL kann für eine Vielzahl von Dingen verwendet werden, durch immer leistungsfähigere Geräte und Anwendungen wird das Anwendungsgebiet größer. Hauptsächlich wird es für 3D Modelle im Internet genutzt. Vor allem im Marketing Bereich wird die Anwendung kontinuierlich beliebter, da die Hersteller nun die Möglichkeit haben dem Kunden das Produkt auf einem zwei dimensionalen Bildschirm drei dimensional darzustellen. Dadurch steigt die Verkaufswahrscheinlichkeit wesentlich an da der Kunde eine Übersicht bekommt als hätte er sich das Produkt in echt angesehen.
Das Laden der Badezimmer-Module übernimmt WebGL und dient gleichzeitig als Basis für three.js. Die ganzen Animationen und Effekte werden von dieser Bibliothek abgewickelt.

2.4 Three.js

[Cab10] [thr20]

Erläuterung

Three.js ist eine browserübergreifende JavaScript [2.2] Bibliothek und API mit der dreidimensionale im Browser Objekte kreiert und angezeigt werden können. Die Besonderheit ist dabei das Three.js auf WebGL [2.3] basiert, dadurch werden Inhalte über die Grafikkarte gerendert was wiederum zu mehr Performance und kürzere Ladezeiten führt. Die Bibliothek ist das erste Mal im April 2010 unter der MIT-Lizenz [4.1] erschienen und wird in einem Repository auf GitHub [2.9] gehostet.

Funktionalität

Dadurch dass die Library in JavaScript [2.2] geschrieben wurde und auf WebGL [2.3] aufsetzt ist die Interoperabilität zwischen den verschiedensten Browsern und Betriebssystemen gewährleistet, ohne zusätzliche proprietäre Plug-Ins zu verwenden. Außerdem unterstütz sie auch virtual reality und vieles mehr.

Verwendung

Alle Module des Badezimmers werden per Three.js geladen und animiert. Jegliche Animationen, Effekte und Texturen werden erst durch diese Technologie möglich. Die Bibliothek ist verantwortlich für die meisten Vorgänge im Bad-Designer und ist somit der wichtigste Bestandteil des Projektes.

2.5 jQuery

[jqu20]

Erläuterung

JQuery ist die meistverwendete und freie JavaScript-Bibliothek die für Manipulationen und Navigation per DOM (Document Object Model) entwickelt wurde. Die plattformunabhängige Bibliothek ist das erste Mal 2006 unter der MIT-Lizenz [4.1] erschienen. 74% der 10.000 meistbesuchten Websites [w3t20] und bereits jede zweite Seite [Gel12] nutzen jQuery. Des Weiteren werden Webframeworks und Content-Management-Systeme wie WordPress, Joomla und Drupal mit jQuery ausgeliefert, die zusätzlich zur Beliebtheit beitragen.

Funktionalität

JQuery ist eine schnelle, leichtgewichtige und funktionsreiche JavaScript-Bibliothek die Dinge wie das Event Handling, Animationen und AJAX vereinfacht. Multiple Browser unterstützen diese Bibliothek und ist somit plattformunabhängig.

Verwendung

JQuery wird primär im Bad-Designer für das Wählen von HTML-Elementen im JS-Code [2.2] angewendet. Dadurch kann man auf die Attribute und das Styling von Objekten via jQuery zugreifen und diese dynamisch manipulieren.

2.6 NGINX

Erläuterung

NGINX ist eine modular aufgebaute Webserver-Software, die von Igor Sysoev entwickelt wurde. Sie wird unter der BSD-Lizenz[4.2] entwickelt. Die Software ist das erste Mal Ende 2004 erschienen und wird bis heute in regelmäßigen Abständen aktualisiert und weiterentwickelt. Außerdem wurde er in der weitverbreiteten und robusten Programmiersprache C geschrieben.

Funktionalität

Der Webserver bietet durch den modularen Aufbau eine Vielzahl an möglichen Techniken, wie zum Beispiel Lastverteilung, Reverse proxying, SSL, Flash-Video-Streaming, WebSocket-Protokoll und vieles mehr.

Verwendung

NGINX gehört zu den marktführen und wird heutzutage bei rund 42% der 1.000 Webseiten [w3t19] mit dem höchsten Traffic verwendet. Als http-Server hat er in Österreich sogar knapp 10% Marktanteil [ref14].
Der Webserver dient beim Bad-Designer als Back-End wo die ganzen Modelle vom Bad und die Website selbst gehostet werden.

2.7 AutoCAD

Erläuterung

AutoCAD ist ein von AutoDesk entwickelter grafischer Zeichungseditor. Dieser ermöglicht es speziell technische Zeichnungen zu modellieren. Außerdem ist es ein vektororientiertes Zeichentool, das auf den Grundlagen für komplizierte 3D-Objekte basiert wie Linien, Polylinien, Kreisen, Bögen und Texten.
Durch die umfangreichen 3D-Funktionen findet das Programm primär in den Bereichen Maschinenbau, Architektur, Design, Geoinformatik häufig Verwendung. In den genannten Bereichen ist die Software konstitutiv und manche Bauwerke beziehungsweise Maschinen sind erst dadurch möglich gewesen.

Funktionalität

Die von der Firma AutoDesk entwickelten Dateiformate .dwg sowie .dxf sind mittlerweile ein Industriestandard im Austausch von CAD-Daten. Dies ermöglicht es das die Dateien auf den verschiedensten Plattformen wie Windows, Unix und MacOS mit Hilfe des Editors verwendbar sind. Desweitern kann man es auch als WebApp und Mobile App für Smartphones und Tablets verwenden.

Verwendung

Alle Module des Badezimmers sind in AutoCAD modelliert worden. Von den Registern bis hin zu den Modulen des Bades. Die sanitären Anlagen wurden von der Firma Lang+Lang GmbH zur Verfügung gestellt.

Abbildung 2.1: AutoCAD UI [Tho14]

2.8 Electron

Erläuterung

Electron ist ein auf C++ und JavaScript basierendes quelloffenes Framework entwickelt von GitHub. Das 2013 erstmals erschienene Framework wird unter der MIT-Lizenz [4.1] vertrieben. Die auf Chromium und Node.js basierende Architektur ermöglicht es eine Cross-Platform-Desktop-Anwendung zu realisieren.

Funktionalität

Webseiten die in HTML, CSS und JavaScript geschrieben wurden, können mit Electron in Desktop Anwendung umgewandelt werden ohne erheblichen Aufwand. Des Weiteren ist es möglich mit anderen Frameworks wie Vue.js und Angular zu arbeiten und anschließend zu einer Desktop App konvertieren. Das besondere daran ist, dass alle Electron-Executable auf den Betriebssystemen Windows, MacOS und Linux einwandfrei ohne Zutun ausführbar sind.

Verwendung

Viele weitverbreitete und bekannte Programme wie die beliebten Editoren Atom und VS Code als auch die Messenger Discord und Skype wurden in Electron erstellt.
Das Framework erlaubt es den Bad-Designer ohne eine Internetverbindung und lokal auf dem Rechner laufen zu lassen. Dafür muss nur das Executable heruntergeladen werden und schon ist es einsatzbereit.

Abbildung 2.2: Bad-Designer als Electron Anwendung

2.9 Git

Historie

Git ist eine kostenlose Software, die für die verteilte Versionsverteilung von Dateien entwickelt wurde. Git wurde vom Linux-Kernel-Entwickler Linus Torvalds initiiert und entwickelt. Torvalds kam auf diese Idee da er zuvor BitKeeper nutzte, diese aber durch Lizenzänderungen kostenpflichtig wurde. 2005 begann er mit der Entwicklung des beliebten Versionsverwaltungsprogrammes. Dabei hatte er drei Anforderungen. Er wollte Unterstützung verteilter Arbeitsläufe, hohe Sicherheit gegen Verfälschung und hohe Effizienz. Mittlerweile ist der Maintainer Junio Hamano. 2018 wurde GitHub für 7,5 Milliarden US-Dollar von Microsoft übernommen.
Git läuft auf den Betriebssystemen Linux, MacOS und Windows. Das Programm selbst wurde in den Sprachen C, Perl, Tcl, Python und C++ geschrieben und unter der GNU-Lizenz [4.3] veröffentlicht.
Zu den Mitbewerbern von Git gehören GitLab und BitBucket. Laut der Plattform Open Hub, eine Website zur Katalogisierung von open-source-software, verwenden dort 71% aller registrierten Projekte Git. [Hub20]
Der Name stammt aus der britischen Umgangssprache der so viel wie Blödmann bedeutet. Torvalds wählte diesen Namen, weil er in der Softwarewelt unbenutzt war, kurz ist und er mit dem Namen einen Witz über sich selbst machte.

> "I'm an egotistical bastard, and I name all my projects after myself. First 'Linux', now 'Git'." – **Linus Torvalds** [Git19]

Eigenschaften und Besonderheiten

Branching

Ein fester Bestandteil und Besonderheit an Git ist das Branching. Ein Branch ist dabei ein neuer Entwicklungszweig, dies ermöglicht es unabhängig vom Hauptzweig zu entwickeln und anschließend beide Zweige oder mehrere zu vereinen, dass sogenannte merging. Diese Verzweigungen sind in Git besonders effektiv implementiert, da sie lediglich eine Referenz auf einen bestimmten Commit sind.

Abbildung 2.3: Visualisierung von Zweigen in Git [Gro15]

Dezentralisierung

Dadurch das jeder Benutzer sich eine lokale Kopie des Repositories samt Versionsgeschichte herunterladen kann, können die meisten Operationen ohne Internet durchgeführt werden.

Sicherer Datentransfer

Git bietet eine Vielzahl an unterschiedlichen Netzwerkprotokollen, um Daten zwischen Repositories zu übertragen. Das gängigste ist aber das sichere SSH Protokoll für Schreiboperationen und ein eigens entwickeltes Protokoll für das Fetchen und Clonen genutzt wird.

Sicherung der Projektgeschichte

In Git ist es nicht möglich die Versionsgeschichte zu manipulieren. Das ist dem Hash-Wert geschuldet der bei jeder Revision (Commit) gespeichert wird. Dieser Wert basiert immer auf der vollständigen Geschichte, die bis zu jener Revision vorangegangen ist.

Kapitel 3

Dateien

3.1 .FBX

[fil20a] [Hel16]

Erläuterung

3D-Objekte, 2d-Objekte mit Objekthöhe, Lichtquellen, Kameras und Materialien werden in dem vom AutoDesk entwickelten Dateiformat gespeichert. Das Universalformat, das bedeutet, dass es von diversen Programmen geöffnet werden kann, enthält unteranderem auch Übergänge, Video- und Audiodateien.

Funktionalität

Dadurch das FBX-Dateien die volle Funktionalität und Wiedergabetreue der Originaldatei beibehalten, ist es zum Beispiel möglich mit Cinema 4D, 3D Studio Max und PowerAnimator den Projektentwurf zu bearbeiten obwohl er davor in AutoCAD erstellt wurde.

Verwendung

3.2 .OBJ

[fil20b] [fil20c]

Erläuterung

Das offene Dateiformat .obj dient zum Speichern von dreidimensionalen geometrischen Formen. Das von Wavefront Technologies erstmals 1989 veröffentlichte Format gilt als universelles Dateiformat für Dreidimensionale-Grafikbearbeitungsprogramme. Daher eignet es sich für sowohl plattform- als auch programmübergreifende Weitergabe von Modellen.

Funktionalität

Das OBJ-Format speichert lediglich die geometrischen Eigenschaften wie die Koordinaten der Ecken, Texturen, Normalen, Flächen und Glättungen. Die Oberflächenschattierungen können durch Referenzen auf .mtl-Dateien implementiert werden.

Verwendung

3.3 .GLB

[Iqb19]

Erläuterung

In GLB-Dateien werden Repräsentationen von dreidimensionalen Modellen im GL Transmission Format, binär abgespeichert. Das GL Transmission Format speichert dreidimensionale Objekte im JSON-Format ab. Des Weiteren werden Kameras, Materialien, Animationen und Meshes binär in der GLB-Datei abgespeichert.

Funktionalität

Durch die kompakten Dateigrößen und das JSON-Format können so komplette dreidimensionale Szenen Repräsentationen schnell und effizient geöffnet werden. Dies ist wichtig für die zügige Darstellung der Badezimmer Module im Bad-Designer.

Verwendung

Kapitel 4

Lizenzen

4.1 MIT-Lizenz

Die freizügige Open-Source-Lizenz der amerikanischen Universität Massachusetts Institute of Technology erlaubt die Wiederverwendung von Software dessen Code sowohl frei und nicht frei einsehbar ist. Frameworks wie jQuery [jQu18], Node.js [Nod19] und .NET Core [Net17] werden unter der 1988 veröffentlichten Lizenz zur Verfügung gestellt.

Inhalt der Lizenz [ope11b]

Bei der MIT-Lizenz handelt es sich um eine der duldsamsten Open-Source-Lizenzen, weshalb sie kaum Begrenzungen oder Pflichten für Nutzer enthält. Hiermit wird gebührenfrei die Erlaubnis erteilt, ohne Beschränkung mit der Software zu handeln, einschließlich der Rechte zur Benutzung, zum Vervielfältigen, Umgestalten, Zusammenführen, Publizieren, Verteilen, Unterlizenzieren und/oder Verkaufen von Kopien der Software, und Personen, denen die Software zur Verfügung gestellt wird, dies unter den untenstehenden Bedingungen zu gestatten.

Copyleft

Die MIT-Lizenz enthält keine Copyleft-Klausel. Das bedeutet, der Benützer kann für die von ihm weiterentwickelten Softwareteile eine Lizenz seiner Wahl verwenden. Dabei hat er die Wahl, ob er seine Weiterentwicklung als proprietäre Software oder als Open Source Software lizenziert. Der unveränderte Teil der Software verbleibt dabei aber weiterhin unter der ursprünglichen MIT-Lizenz.

4.2 BSD-Lizenz

Die von der amerikanischen Universität University of California, Berkeley stammende Lizenz umfasst eine Gruppe von Open-Source-Lizenzen. Ähnlich zur MIT-Lizenz ist sie freizügig, das Akronym BSD steht für Berkeley Software Distribution.

Inhalt der Lizenz [ope11a]

Software die unter der BSD-Lizenz veröffentlicht wurde darf frei verwendet werden. Konkret bedeutet das, dass es erlaubt ist die Software zu kopieren, ändern und zu verbreiten aber der Copyright Vermerk nicht entfernt werden darf. Damit wird gewährleistet das der Entwickler vom ursprünglichen Programm gewürdigt wird. Durch diese Bedingungen können Softwareprodukte auch kommerziell gehandelt werden, wenn sie auf Technologien aufbauen, die unter der BSD-Lizenz zu freien Verfügung gestellt wurden.

Copyleft

Das besondere an dieser Lizenz ist, dass es unter gewissen Umständen kein Copyleft enthält. Beispielweise muss also ein Programmierer, der eine Software die unter der BSD-Lizenz steht, verändert und anschließend binär verbreitet nicht den Quellcode mitveröffentlichen. Jedoch muss er das Programm in nichtkompilierter oder kompilierter Form weiterhin unter der BSD-Lizenz veröffentlichen samt Lizenztext.

4.3 GNU General Public License

Die General Public License, aus dem englischen übersetzt etwa *allgemeine Veröffentlichungserlaubnis* ,ist die meistbenutzte Softwarelizenz, die es gewährt eine Software auszuführen, manipulieren und zu verbreiten. Programme, die unter dieser veröffentlicht werden als freie Software bezeichnet. Das bedeutet das die Freiheit der Nutzer im Zentrum steht und ihnen gleichzeitig die Nutzungsrechte nicht eingeschränkt werden. Die Erstfassung stammt von Richard Stallman von der Free Software Foundation, die über das Copyright des Lizenztextes verfügen. Die Lizenz wird in unregelmäßigen Abständen aktualisiert.

Inhalt der Lizenz [gnu07]

Die tolerante Lizenz lässt zu das unter ihr die Software sowohl kommerziell als auch kostenlos angeboten werden darf. Jedoch muss dabei wegen des Copylefts die Änderungen und der Quellcode dem Endnutzer offen gelegt werden.

Copyleft

Änderung und Abwandlungen von GPL lizensierten Arbeiten dürfen nur unter derselben Lizenz vertrieben werden.

Teil II

Grundlagen und Methoden

Teil III

Umsetzung

Kapitel 5

Aufbau und Systemarchitektur

5.1 Systemarchitektur – Web-Applikation

Erklärung

Die Systemarchitektur der Webanwendung besteht grundsätzlich aus zwei Teilen. Dem auf NGINX [2.6] basierenden Server und dem Client der per Browser die Seite aufruft. Auf dem Server sind alle benötigten Dateien und die Logik. In den Dateien, die in JavaScript geschrieben wurden, befinden sich die Modelle, die Animationen und die Geschäftslogik. Diese werden dann an den Browser übermittelt und gerendert.

Deployment

Um die Aufwendung auf einem Server zu deployen und um sie anschließend zu verwenden, müssen einige Kriterien erfüllt werden. Es muss ein NGINX-Server [2.6] vorliegen auf dem die index.html und die JavaScript Dateien hochgeladen wurden. Erst diese ermöglichen den Start der Anwendung, zusätzliche Schritte sind nicht von Nöten.

Abbildung 5.1: Systemarchitektur der Web-Applikation

5.2 Systemarchitektur – Electron Anwendung

Erklärung

Das Electron Executable muss lediglich vom Server heruntergeladen werden und installiert werden. Danach ist die Anwendung sofort einsetzbar. Dadurch das die Applikation mit Electron 2.8 gemacht wurde lässt sie sich wie eine normale Desktopanwendung starten ohne Browser.

Deployment

Um die Diplomarbeit als Electron Anwendung zu deployen und benützen muss man lediglich die bestehende Web-App in eine Electron App konvertieren und auf dem Server zum Herunterladen stellen.

Kapitel 6

Summary

Here you give a summary of your results and experiences. You can add also some design alternatives you considered, but kicked out later. Furthermore you might have some ideas how to drive the work you accomplished in further directions.

Literatur

[Cab10] Ricardo Cabello. *JavaScript 3D library*. 2010. URL: https://github.com/mrdoob/three.js/tree/master (besucht am 09. 02. 2020).

[fil20a] file-extensions.org. *Dateiendung FBX*. 2020. URL: https://www.file-extension.org/de/extensions/fbx (besucht am 09. 02. 2020).

[fil20b] file-extensions.org. *Dateiendung OBJ*. 2020. URL: https://www.file-extension.org/de/extensions/obj (besucht am 09. 02. 2020).

[fil20c] fileformat.info. *Wavefront OBJ File Format Summary*. 2020. URL: http://www.fileformat.info/format/wavefrontobj/egff.htm (besucht am 09. 02. 2020).

[Gel12] Matthias Gelbmann. *jQuery now runs on every second website*. 2012. URL: https://w3techs.com/blog/entry/jquery_now_runs_on_every_second_website (besucht am 13. 08. 2012).

[Git19] Git. *Git FAQ*. 2019. URL: http://web.archive.org/web/20190806144839/https://git.wiki.kernel.org/index.php/GitFaq#Why_the_.27Git.27_name.3F (besucht am 06. 08. 2019).

[gnu07] gnu.org. *GNU GENERAL PUBLIC LICENSE*. 2007. URL: https://www.gnu.org/licenses/gpl-3.0.txt (besucht am 02. 02. 2020).

[Gro15] Torsten Groß. *Git: Mit Branches arbeiten (git branch)*. 2015. URL: https://blog.seibert-media.net/blog/2015/07/31/git-mit-branches-arbeiten-git-branch/ (besucht am 31. 07. 2015).

[Hel16] AutoDesk Help. *Importieren und Exportieren von FBX-Dateien*. 2016. URL: https://knowledge.autodesk.com/de/support/autocad/learn-explore/caas/CloudHelp/cloudhelp/2016/DEU/AutoCAD-Core/files/GUID-AE2B5A3E-2068-4706-AE83-926C26C6EE9C-htm.html (besucht am 13. 01. 2016).

[Hub20] Open Hub. *Compare Repositories*. 2020. URL: https://www.openhub.net/repositories/compare (besucht am 02. 02. 2020).

[Iqb19] Kashif Iqbal. *GLB*. 2019. URL: https://wiki.fileformat.com/3d/glb/ (besucht am 03. 12. 2019).

[jQu18] jQuery. *jQuery MIT-License*. 2018. URL: https://github.com/jquery/jquery/blob/master/LICENSE.txt (besucht am 14. 05. 2018).

[jqu20] jquery. *jQuery API*. 2020. URL: https://api.jquery.com/ (besucht am 10. 02. 2020).

[Net17] .NetCore. *.NET Core License*. 2017. URL: https://github.com/dotnet/core/blob/master/LICENSE.TXT (besucht am 25. 05. 2017).

[Nod19] Node.js. *Node.js License*. 2019. URL: https://github.com/nodejs/node/blob/master/LICENSE (besucht am 30. 11. 2019).

[ope11a] opensource.org. *The 3-Clause BSD License*. 2011. URL: http://web.archive.org/web/20110529122012/https://opensource.org/licenses/BSD-3-Clause (besucht am 29. 05. 2011).

[ope11b] opensource.org. *The MIT License (MIT)*. 2011. URL: http://web.archive.org/web/20110527102833/http://www.opensource.org/licenses/MIT (besucht am 27. 05. 2011).

[ref14] reflecte.at. *österreichisch Internet Statistiken*. 2014. URL: https://web.archive.org/web/20170214004415/https://www.reflecte.at/osterreichisch-internet-statistiken (besucht am 14. 02. 2017).

[Tho14] Michael Thomas. *AutoCAD 2015 Announced – Whats New?* 2014. URL: http://web.archive.org/web/20160625045609/https://designandmotion.net/autodesk/autocad/why-autocad-2015-is-the-best-release-ever/ (besucht am 12. 11. 2017).

[thr20] threejs.org. *JavaScript 3D library*. 2020. URL: https://threejs.org/ (besucht am 09. 02. 2020).

[w3t19] w3techs.com. *Usage of web servers broken down by ranking*. 2019. URL: http://web.archive.org/web/20191212200647/https://w3techs.com/technologies/cross/web_server/ranking (besucht am 12. 12. 2019).

[w3t20] w3techs. *Usage statistics of JavaScript libraries for websites*. 2020. URL: https://w3techs.com/technologies/overview/javascript_library (besucht am 10. 02. 2020).

Abbildungsverzeichnis

1 Bad Designer; Frontalansicht 2
2 Bad Designer; Schrägansicht 2
3 Bad Designer; Seitenansicht 1
4 Bad Designer; Frontal view 2
5 Bad Designer; Oblique view 2
6 Bad Designer; Side view 1

1.1 Don Knuth, the inventor of TEX 7

2.1 AutoCAD UI [Tho14] 16
2.2 Bad-Designer als Electron Anwendung 18
2.3 Visualisierung von Zweigen in Git [Gro15] 20

5.1 Systemarchitektur der Web-Applikation 30

Tabellenverzeichnis

Project Log Book

Date	Participants	Todos	Due

Anhang A

Additional Information

If needed the appendix is the place where additional information concerning your thesis goes. Examples could be:

- Source Code
- Test Protocols
- Project Proposal
- Project Plan
- Individual Goals
- ...

Again this has to be aligned with the supervisor.

Anhang B

Individual Goals

This is just another example to show what content could go into the appendix.