

3D Portfolio Gallery

DIPLOMARBEIT

verfasst im Rahmen der

Reife- und Diplomprüfung

an der

Höheren Abteilung für IT - Medientechnik

Eingereicht von:

Ema Halilovic
Lorenz Litzlbauer
Fabian Maar

Betreuer:

Patricia Engleitner
Natascha Rammelmüller

Leonding, April 2022

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

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

Die vorliegende Diplomarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

Leonding, April 2022

S. Schwammal & S. Schwammal

Abstract

3D Portfolio Gallery is a web application that was developed by Ema Halilovic, Lorenz Litzlbauer and Fabian Maar as part of their diploma thesis. 3D Portfolio Gallery wants to enable designers to share their design portfolio with the world in an innovative way. The designers can use the 3D Portfolio Gallery to create a 3D portfolio from their own media (film, photo or 3D data), which interested parties can view on the website with various types of interaction in 3D space. For this purpose, Angular is used in the frontend for the SPA (single-page application) logic, ThreeJs for the 3D display and technologies such as Quarkus, Maven, JPA, Panach and Hibernate ORM are used for the server. To ensure user security, we use JWT (JsonWebToken)



Zusammenfassung

3D Portfolio Gallery ist eine Webapplikation, die von Ema Halilovic, Lorenz Litzlbauer und Fabian Maar im Rahmen der Diplomarbeit entwickelt wurde. 3D Portfolio Gallery will Designer*innen ermöglichen, ihr Design-Portfolio auf eine innovative Art und Weise mit der Welt zu teilen. Die Designer*innen können mithilfe von 3D Portfolio Gallery aus eigenen Medien (Film, Foto oder 3D - Daten) ein 3D Portfolios erstellen, welche sich Interessierte mit verschiedenen Interaktionsarten im 3D Raum auf der Webseite anschauen können. Dafür werden im Frontend Angular für die SPA (Single-Page-Applikation) - Logik benutzt, für die 3D-Darstellung ThreeJs und für den Server werden Technologien wie Quarkus, Maven, JPA, Panach und Hibernate ORM verwendet. Um die Sicherheit der User zu gewährleisten, verwenden wir JWT (JsonWebToken)



Inhaltsverzeichnis

1	Einleitung	1
2	Glossar	2
3	Umfeldanalyse	3
4	Aufgabendifferenzierung	4
4.1	Lorenz	4
4.2	Fabian	4
4.3	Ema	4
5	Technologien	5
5.1	Angular	5
5.2	Webstorm [M]	6
5.3	ThreeJs [L]	7
5.4	Cinema 4D [M]	10
5.5	Figman [L]	10
6	Umsetzung der Interaktions-Seite [M]	12
6.1	Border-Collision	12
7	Umsetzung	14
7.1	Plannung	15
7.2	Design	17
8	Zusammenfassung	22
	Literaturverzeichnis	VI
	Abbildungsverzeichnis	VII
	Tabellenverzeichnis	VIII

Quellcodeverzeichnis

IX

Anhang

X

1 Einleitung

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

2 Glossar

Framework [M]

Ein Framework soll dem*der Programmierer*in helfen neue Applikationen zu schaffen und ihm*ihr ein Grundgerüst liefern auf dem er*sie seine*ihre Software aufbaut.

Open-Source [L]

Open Source bei einer Applikation bedeutet, dass der Code dieser öffentlich zugänglich ist. Der Open Source Status sagt aber nichts über die Lizenzierung der Applikation aus.

Framework[M]

Single-Page-Webanwendung: Eine Single-Page-Webanwendung interagiert mit dem*der User*in, wodurch sich einzelne Komponenten der Website dynamisch verändern.

Design-Pattern [L]

Im Programmieren gibt es Probleme, die immer wieder auftreten. Diese Probleme wurden schon einmal sehr effizient gelöst und es besteht kein Nutzen sie immer wieder zu lösen. Design Patterns sind Strategien, wie man diese Probleme lösen kann.

Framework[M]

Event-Listener //TODO

Framework[M]

Third-Party [M] //TODO

3 Umfeldanalyse

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula. Citing [1] properly.

Was ist eine GUID? Eine GUID kollidiert nicht gerne.

Kabellose Technologien sind in abgelegenen Gebieten wichtig [2].

4 Aufgabendifferenzierung

4.1 Lorenz

4.2 Fabian

4.3 Ema

5 Technologien

5.1 Angular

5.1.1 Allgemeines [M]

Angular ist ein Framework für Webapplikation, das auf der Programmiersprache Typescript basiert. Es ist eines der renommiertesten Frameworks zur Front-End-Entwicklung und wird als Open-Source-Software zur Verfügung gestellt. Besonders bietet sich Angular für Single-Page-Webanwendung an, da es ein komponentenbasiertes Framework ist. Das bedeutet, der Code ist wiederverwendbar und verkapselt. Komplexe Logiken werden auf ihre Grundelemente reduziert und beeinflussen sich nicht gegenseitig. [3]

5.1.2 Dependency [L]

Um eine nachvollziehbare Auswahl zu treffen, muss verstanden werden, wie Angular funktioniert. In den folgenden Unterabschnitten wird sich mit den Technologien auf die Angular aufgebaut ist, auseinandergesetzt.

RxJS

ReactiveX ist eine Library für das Erstellen von asynchronen und Event-basierenden Programmen, dafür benutzt es *observable sequences*. Die Library erweitert so, dass *Observer-Design-Pattern* mit verschiedenen neuen Operatoren und händelt dabei "Lowlevel"-Funktionen wie Threading, Synchronisation, Thread-Sicherheit und das nicht-blockieren von I/O vergleiche [4]

RxJS ist eine Implementierung von ReactiveX für die Programmiersprache Javascript. Angular verwendet RxJS für reaktive Programmierart.

Webpack

Die Hauptfunktion von Webpack ist es, viele verschiedenen Daten zu einem Paket für eine JavaScript Applikation zusammenzufassen, dabei optimiert es die Dateigröße,

indem es mithilfe von einem selbst generierten *dependency-graph* die Abhängigkeiten der Applikation überprüft und nur notwendige Teile für die JavaScript-Applikation der Daten nimmt. Bei der Zusammenfassung werden Sprachen, die der Webbrowser nicht Interpretieren kann wie Typescript, Sass uvm. in die für den Webbrowser verständlichen Gegenstücke gewandelt. Webpack hat aber auch viele andere Features. Vergleiche [5] Angular benutzt Webpack um TypeScript in JavaScript und Sass bzw. Scss in Scss zu wandeln, um beim Bauen des Projektes alle Module in ein einziges zusammenzufassen und um die Applikation bei der Entwicklung zu *Live-Reloading* zu unterstützen. Dabei wird bei einer Änderung im Code die gesamte Applikation geupdatet und neu gestartet.

5.1.3 Vorteile/Auswahlkriterien [M]

Dieses komponentenbasierte Programmieren war eines der vielen Gründe warum Angular eine geeignete Option ist. Im Abschnitt 5 //TODO wird nochmals deutlich, wie dieser Aspekt genutzt wird. Ein weiterer Grund, ist die einfache Einbindung von Libraries und deren Übersichtlichkeit, die durch ngModules ermöglicht werden. Die ständige Erweiterung und die Unterstützung von Third-Party-Libraries, die unter anderem für die 3D-Darstellung verwendet werden, wird ebenfalls von diesen ngModulen ermöglicht. Beispiele wie wir dieses ngModule aufgesetzt haben befinden sich im Abschnitt Landing-Page aufsetzen //TODO VERLINKEN. Auch aufgrund von persönlichen Erfahrungen im Unterricht und in anderen Projekten war Angular eine passende Auswahlmöglichkeit. [6]

5.2 Webstorm [M]

Webstorm ist eine Entwicklungsumgebung vom Unternehmen JetBrains, die sich auf die Programmiersprache JavaScript spezialisiert hat. Ebenfalls unterstützt es besonders Frameworks wie Angular. Das Webstorm besonders auf Angular optimiert wurde, zeigt sich durch viele Features, die das Entwickeln von Angular-Projekten erleichtern. So macht es Webstorm möglich, mit nur wenigen Mausklicks eine neue Angular-Dependency oder Komponente zu erstellen. Auch wird die Entwicklungszeit durch intelligente Code-Vervollständigung, Code-Formatierung, einfache Navigation und viele weitere hilfreiche Features deutlich verkürzt. [7]

5.3 ThreeJs [L]

ThreeJs ist eine JavaScript Library für die Darstellung von 3D-Grafiken im Web. Für die 3D-Darstellung nutzt ThreeJs meistens WebGL (mehr dazu im nächsten Abschnitt ThreeJs Dependency), ein low-level Framework. WebGL hat eine hohe Komplexität. ThreeJs bietet eine Abstraktion, bei der es viele Sachen wie die 3D-Szene, Lichter, Schatten, Materialien, Texturen und 3D-Mathe händelt, um die 3D-Darstellungen im Web einfacher zu gestalten. In ThreeJs werden Geometrie, Objekte und Materialien verbunden, um ein 3D-Objekt zu erstellen. Dabei kann die Struktur einer Szene der Abbildung 2 ähneln. Vergleiche [8, ThreeJs fundamentals]

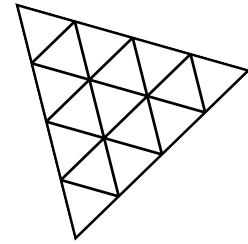


Abbildung 1: ThreeJs Logo

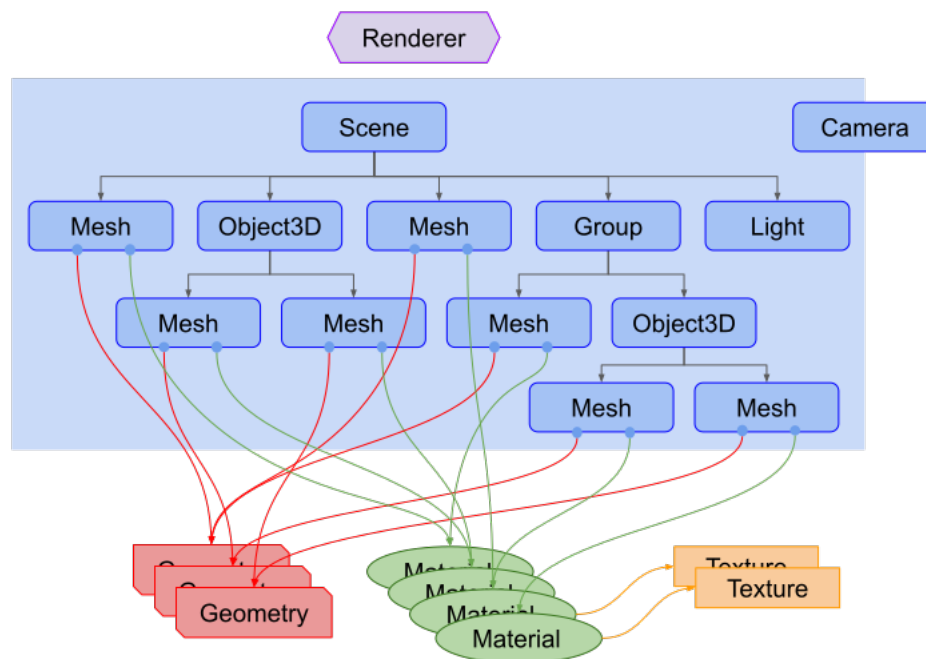


Abbildung 2: Die Struktur von ThreeJs

5.3.1 Dependency

Um einen Großteil der 3D-Darstellungen zu rendern, benutzt ThreeJs WebGL.

WebGL

WebGL ist eine lizensfreie Api, die dafür benutzt wird 3D-Grafiken im Web darzustellen. Sie basiert auf OpenGL



Abbildung 3: ThreeJs Logo

ES 2.0 und benutzt auch dieselbe shading language GLSL. WebGL ist eine Low-Level-API, das bedeutet, dass schon sehr einfache Projekte einen hohen Programmieranteil haben. Vergleiche[9, WebGL Getting Started]

5.3.2 Auswahlkriterien

Es gab verschiedene Auswahlkriterien für die 3D-Web-API.

- Effizienzen der 3D Engine (Hardware Acceleration, RAM-Auslastung)
- Benutzerfreundlichkeit (Programmerexperience)
- Zusätzliche Features
- Die Features des Projektes müssen damit umsetzbar sein
 1. Das Laden von 3D-Modellen aus 3D-Dateien und aus dem Internet
 2. Video-Texturen support

Alternativen 3D Web Apis

Es gibt viele Technologien, die 3D-Grafiken im Web ermöglichen, wie Three.js, Babylon.js, A-Frame, X3DOM und WebGL

A-Frame A-Frame wird von der Morzilla Foundation als OpenSource-Projekt entwickelt. Die 3D-Szene wird durch eine deklarative Sprache mit XML-Syntax definiert. Über die WebVR-API bietet die Libray auch die Möglichkeit, die 3D-Szenen durch eine VR-Brille zu erfahren. Bei der 3D-Darstellung setzt A-Frame auf ThreeJs. [10, A-Frame Wikipedia]

A-Frame hat ähnliche Funktionen und Leistung im Vergleich zu ThreeJs, da es ja schlussendlich darauf aufbaut. A-Frame sticht bei der VR support hervor (ThreeJs hätte auch eine Unterstützung dafür, diese ist aber schwieriger einzubinden), doch ist das kein unbedingt nötiges Feature. ThreeJs ist ja bereits eine Abstraktion von WebGL. Für das Projekt wird keine weiter Abstraktion gebraucht.

WebGL Im Vorherigen Kapitel wurde sich bereits mit WebGL befasst. 5.3.1 Um nur eine einfache 3D-Szene in WebGL darzustellen, muss sehr viel Code geschrieben werden. Denn WebGL übernimmt keine Low-Level-Funktion. Es müssen Matrixrechnungen für die Transformationen von 3D-Objekt Manuel und Vertex buffers, die die Daten der

Vertex Positionen, Normaldaten, Farben und Texturen, selbst programmiert werden und das in einer eigenen Programmiersprache. Deshalb war WebGL keine Option für das Projekt.

Angular Three

Angular Three ist ein Open Source Projekt von Matt DesLauriers. Es zielt darauf ab die Vorteile von Angular und ThreeJs zu kombinieren. Dabei verbindet es das Prinzip der Komponenten von Angular mit der 3D Darstellung von ThreeJs.

Listing 1: Angular Three - Komponentenbasiertes 3D Szenen in HTML

```

1 <ngt-canvas>
2   <ngt-ambient-light intensity="0.5"></ngt-ambient-light>
3   <ngt-spot-light [position]="10" angle="0.15" penumbra="1"></ngt-spot-light>
4   <ngt-point-light [position]="-10"></ngt-point-light>
5
6   <app-cube [position]="[1.2, 0, 0]"></app-cube>
7   <app-cube [position]="[-1.2, 0, 0]"></app-cube>
8
9   <ngt-soba-orbit-controls></ngt-soba-orbit-controls>
10 </ngt-canvas>

```

Listing 2: Angular Three - App Cube

```

1 <ngt-mesh
2   (beforeRender)="onCubeBeforeRender($event)"
3   (click)="active = !active"
4   (pointerover)="hovered = true"
5   (pointerout)="hovered = false"
6   [scale]="active ? 1.5 : 1"
7   [position]="position"
8 >
9   <ngt-box-geometry></ngt-box-geometry>
10  <ngt-mesh-standard-material [color]="hovered ? 'turquoise' :
    'tomato' "></ngt-mesh-standard-material>
11 </ngt-mesh>

```

Ein Vorteil von Angular Three ist, dass durch nur wenige Zeilen Code 1 eine 3D-Szene mit Lichtern und Orientierungsfunktionen erstellt werden kann und die Business-Logik, App-Cube 2 durch die Verwendung einer Komponente ausgelagert werden kann.

Ein weiterer Vorteil von Angular Three ist die ausführliche Dokumentation mit Codebeispielen. (link) Angular Three Dokumentation <https://angular-three.netlify.app/docs/getting-started/overview>

Wegen der vielen Vorteile hohe Benutzerfreundlichkeit, ähnliche 3D-Leistung zu ThreeJs (Angular Three basiert auf ThreeJs, welches selbst die WebGL-Renderengine benutzt) und wegen der Verbindung von Angular und ThreeJs Features bietet sich die Libray für das Projekt an.

Mithilfe eines Prototypen (mehr dazu im Abschnitt fortlaufendes Prototyping) wurde getestet, ob Angular Three alle Features, die für das Projekt benötigt werden, unterstüt-

zen kann, die für das Projekt nötig sind. Dabei wurde festgestellt, dass sich die Libray Angular für das Projekt nicht eignet. Denn beim Prototyping sind Fehler aufgetreten. Beim erneuten Laden von 3D-Objekten aus derselben Datei haben sich keine Daten mehr im Objekt befunden. ThreeJs Module funktionierten Teilweise nicht, wie die FristPersonControls.

Weil Angular Three nicht die für das Projekt aufgestellten Anforderungen erfüllt hat, wurde der Prototyp zu ThreeJs migriert, weil sich die Syntax ähnelt und dadurch nur wenig Aufwand bei der Migrierung entstand. Diese 3D-Library wurde schlussendlich auch im Projekt verwendet.

5.4 Cinema 4D [M]

Cinema 4D ist ein professionelles Programm vom Unternehmen Maxon zur 3D-Modellierung und Animation. Zum einen ist die Software leicht zu bedienen und zum anderen bietet Maxon eine Vielzahl an Tutorials und Anleitungen. Auch aufgrund von persönlichen Erfahrungen im Unterricht wurde das Programm zur Erstellung von diversen 3D-Modellen benutzt. Schon bereits in der Vergangenheit modellierte Werke konnten sich zu Nutze gemacht und teilweise auch im 3D-Raum verwendet werden. Ein weiterer wichtiger Aspekt ist, dass Cinema 4D den Export von GLTF-Files unterstützt. GLTF ist das Dateiformat, das genutzt wird, um ein 3-Modell in der Three.js Szene zu laden. Da ähnliche Programme wie zum Beispiel Blender diese Exportmöglichkeit nicht anbieten, war Cinema 4D die gewählte Option für die 3 Modellierung. [11]

5.5 Figman [L]

Figma ist ein Programm für die Erstellung und Testung von UI und UX Prototypen. Das Programm ist einsteigerfreundlich, denn es hat eine benutzerfreundliche Oberfläche und es gibt viele Tutorials auf der Webseite und von der Figma Community. Figma ist primär eine Web-Applikation, bietet aber auch eine Desktop-Version für macOS und Windows und eine mobile Version für iOS und Android an. Figma arbeitet mit mehreren Konzepten, um den Designprozess zu vereinfachen:

Kollaboration In Figma können mehrere Personen gleichzeitig auf verschiedenen Geräten designen. Es gibt verschiedene Rollen, wie Zuschauer*in, Kritiker*in oder

Mitarbeiter*in. Die Möglichkeiten können genutzt werden um einen*einer Kunden*in in den Designprozess zu involvieren und schon früh Feedback auf das Design bekommen zu können.

Plugins Figma hat eine große Community, gibt es ein Feature nicht, kann dieses von der Community im PluginStore als Plugin hinzugefügt werden. Ein Plugin ist eine Software-Erweiterung, welche die Fähigkeiten oder die Features eines Softwareprojektes erweitert.

Assets Figma arbeitet mit dem Konzept der Assets. Bereits designte Komponenten können als Assets gespeichert werden. Figma legt einen großen Wert auf die Modularität, es können Farben und Pixelgrößen als Variable gespeichert werden, wenn diese sich verändern, verändern sich gleichzeitig die darauf referenzierenden Assets.

5.5.1 Auswahl

Wegen dieser vielen Features und persönlichen Erfahrungen in privaten Projekten bietet sich Figma für dieses Projekt als UX/UI-Design- Tool an.

6 Umsetzung der Interaktions-Seite

[M]

6.1 Border-Collision

Um den Benutzern und Benutzerinnen ein möglichst realistisches Gefühl für die Ausstellung zu geben, ist es wichtig, den Ausstellungsraum auch so zu gestalten. Daher ist der Raum durch Wände abgegrenzt, wodurch es nicht mehr möglich ist, sich über den Raum hinaus zu bewegen. Um eine Kollision mit der Wand zu berechnen, gibt es durch die Three.js Bibliothek einige Möglichkeiten.

6.1.1 Erster Ansatz

Eine davon ist, eine Bounding Box zu erstellen. Dabei kann zwischen zwei verschiedenen Arten unterschieden werden. Zum einen werden Axis Aligned Bounding Box verwendet, um eine Box rund um das 3D-Objekt zu erstellen, die sich nicht an die Rotation des Objekts anpasst. Die zweite Variante ist die Oriented Bounding Box, die im Endeffekt gleich funktioniert, sich aber darin unterscheidet, dass sie sich an die Achsen des Objekts anpasst. Da sich die Bounding Boxen jedoch über das ganze 3D-Objekt erstrecken, wird eine Kollision direkt berechnet, nachdem der*die Benutzer*in den Raum betritt. Da sich die Bounding Boxen nicht an jede einzelne Wand anpassen konnten, musste ein anderer Lösungsweg gefunden werden.

6.1.2 Zweiter Ansatz

Um eine Veränderung der Position des*der Benutzers*in festzustellen, muss die Veränderung der Kameraposition evaluiert werden. Dabei wird beim initialisieren der Kamera eine Kopie von ihr erstellt. In der Animate-Funktion wird eine Veränderung der Kamera überprüft, indem die Positionen der Kamera mit der Kopie verglichen werden. Die Kamera nimmt dabei immer eine neue Position ein, wenn sich der*die Benutzer*in im Raum bewegt, während die Kopie dabei die alte Position der Kamera einnimmt. Jedesmal wenn eine Veränderung geschieht, wird überprüft, ob die Kamera mit der

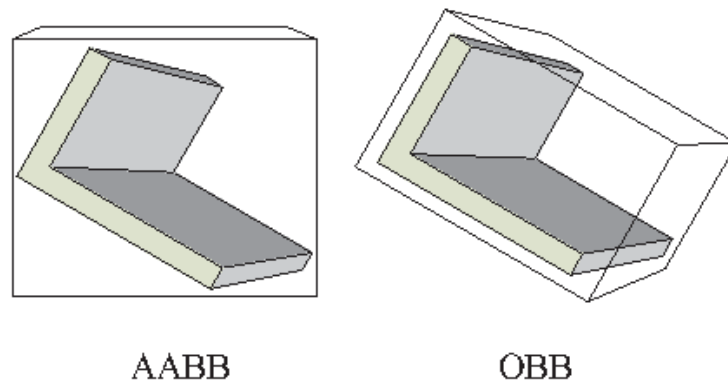


Abbildung 4: Der Unterschied zwischen AABB und OBB

Wand kollidiert. Dies geschieht, indem ein Raycast mit den Positionen der Kamera und der Kopie initialisiert wird.

6.1.3 Far und Near

Die Attribute Far und Near werden dafür verwendet, um die Objekte, die im Ray liegen, einzugrenzen. Dabei können die Werte nicht negativ sein und der Far-Wert muss größer als der Near-Wert sein. Um die Kollision erst direkt am Ursprungsort, im Falle der Kamera, des Rays zu berechnen, wird der Far-Wert auf 100 gesetzt.

Nachdem der Raycast korrekt initialisiert und angewandt wurde, muss bei einer Berührung mit der Wand nur noch richtig damit umgegangen werden. Dabei wird die Bewegung des Benutzers gestoppt. Um diese auch wieder zu starten, muss sich der*die Benutzer*in weg von der Wand begeben. Überprüft wird dies nach jeder Benutzer*inneneingabe mit einem Event-Listener. Falls nach dieser keine Berührung mehr mit einer Wand besteht, wird die Bewegung fortgesetzt und der*die Besucher*in kann sich wieder frei im Raum bewegen.

7 Umsetzung

Siehe tolle Daten in Tab. 1.

Siehe und staune in Abb. 9. Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetur at, consectetur sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

	Regular Customers	Random Customers
Age	20-40	>60
Education	university	high school

Tabelle 1: Ein paar tabellarische Daten



Abbildung 5: Don Knuth – CS Allfather

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetur. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus. Dann betrachte den Code in Listing 3.

Listing 3: Some code

```

1  # Program to find the sum of all numbers stored in a list (the not-Pythonic-way)
2
3  # List of numbers
4  numbers = [6, 5, 3, 8, 4, 2, 5, 4, 11]
5
6  # variable to store the sum
7  sum = 0
8
9  # iterate over the list
10 for val in numbers:
11     sum = sum+val
12
13 print("The sum is", sum)
```

7.1 Planung

7.1.1 Ideenfindung [M]

Die Ideenfindung ist der Start eines Projektes und bildet das Fundament, auf dem das Projekt entsteht. Die Idee für die Entstehung eines Projektes kann viele Motive haben, so ist es wichtig, als Projektteam diese zu erkennen und als Motivation zu

nehmen. Während oder nach dem Prozess der Ideenfindungs wird schon bald klar, welche Projektwürdigkeit das Projekt besitzt. Somit kann schon frühzeitig entschieden werden, ob es sinnvoll erscheint, das Projekt zu realisieren oder ob nochmals eine neue Ideenfindung von Nöten ist. Auch bei dieser Diplomarbeit wurde der Prozess der Ideenfindung mehrmals wiederholt, wodurch die Wichtigkeit und die Methodik der Findung von Ideen besonders bemerkbar wurde. Die Ideenfindung erfolgt meist mit Hilfe der Anwendung einer passenden Kreativitätstechnik.

Kreativitätstechniken sind vielseitig einzusetzen, werden aber häufig am Anfang eines Projektes angewandt. Sie sind besonders hilfreich, wenn ein rationales Problemlösen nicht zielführend erscheint. Sie helfen, Ziele, Lösungen und Risiken zu finden, evaluieren und festzulegen. Für die Diplomarbeit bot sich besonders die Kreativitätstechnik Brainstorming an, da durch sie viele verschiedene Ideen in kürzester Zeit gesammelt werden können.

Beim Brainstormen findet jede*r Teilnehmer*in zu einem bestimmten Thema so viele Ideen wie möglich. Das Brainstormen erfolgt in 2 Phasen. Zuerst werden alle Ideen niedergeschrieben und anschließend die Beste herausgefiltert. Aufgrund der Expertise der Betreuungslehrerinnen im Bereich 3D-Modellierung und aus eigenem Interesse, konnte sich relativ schnell auf das Thema *eine Software mit 3D-Integration* geeinigt werden. Schlussendlich konnte sich für eine Idee entschieden werden, nachdem das Brainstormen nach folgenden Kriterien erfolgt hat:

- Quantität vor Qualität
- Freies Assoziieren und Fantasieren sind erwünscht
- Keine Kritik, Korrektur oder Meinungsäußerung
- Möglichst viele Ideen
- Inspirieren lassen von anderen

[12]

7.1.2 fortlaufendes Prototyping

7.2 Design

7.2.1 Userexperience [L]

In keinem Softwareprojekt darf nicht der*die Benutzer*innen vergessen werden. Feature und Design müssen immer mit dem Gedanken entwickelt werden, wie hilft das dem*der Kunden*in oder der Zielgruppe weiter.

In den folgenden Kapiteln wird das Thema User Experience und wie dieses Thema im Projekt umgesetzt wurde, behandelt.

UX Design (Userexperience)

Ein gutes Userexperience-Design ist die Grundlage für eine erfolgreiche Positionierung und Kommunikation mit dem*der Benutzer*in. UX bezieht sich dabei auf die Interaktion des Benutzers mit der Umwelt, aber auch mit dem angebotenen Service oder Produkt. In diesem Kontext hat Design mehrere Bedeutungen. Erstens gibt es den Punkt der Gestaltung der Interaktionen, hierzu gehört der Begriff Interface Design (UI Design), er bedeutet visuelle Kommunikation über Zeichen und Symbole. Zweitens gibt es Design als den Designprozess. Im Designprozess muss erkannt werden, was dem*der Benutzer*in wichtig ist und dieses dem Produkt zugewiesen werden, sodass es auch der*die Benutzer*in erkennen kann [13].

Anwendungen von UX im Projekt

Das Projekt wurde schon von Anfang an mit einem Gedanken für den Benutzer gestartet. Userstories waren aus der Sicht eines Benutzers formuliert und das Projekt startete mit der Frage, welchen Nutzen kann das Projekt einem*einer Nutzer*in geben.

Im Userinterface-Design wurde Figma (siehe Kapitel Technologien Figma) benutzt. Die Oberfläche wurde so designt, damit sie leicht verständlich, nützlich und benutzerfreundlich ist. Dafür wurden viele Prototypen (siehe Abbildung 6) in Figma gestallten und durch Umfragen die besten bestimmt und dementsprechende Anpassungen am Design gemacht.



Abbildung 6: OberflächenDesign: Prototypen in Figma

7.2.2 Corporate Design [M]

Das Corporate Design unterstützt die Corporate Identity und deren Ziele. Unter der Corporate Identity versteht man das äußere Erscheinungsbild eines Unternehmens. Hierbei sollen Bestandteile eines Unternehmens nach außen hin einheitlich, unverwechselbar und positiv wirken und mit der Gesamtstrategie stimmig sein. Zum Beispiel werden Mitarbeiter, Abteilungen und Produkte sowie ihre Beziehungen zur Firma nach gewissen Normen strategisch gestaltet und geregelt. Dabei lässt sich die Corporate Identity in 3 Teilbereiche gliedern:

- Corporate Behaviour
- Corporate Communication
- Corporate Design

[14]

Bei der Diplomarbeit kam vor allem das Corporate Design zum Einsatz. Hierbei wird vor allem das äußere Erscheinungsbild des Produkts als Einheit repräsentiert. Dabei werden zum Beispiel Hausfarbe, Logo, Gestaltungsraster und weitere Design-Elemente aufeinander abgestimmt. Dies war auch fester Bestandteil der Design-Phase. [15]

//TODO Text über Bild + Bilder richtig anordnen Der erste Schritt des Designs war es, zwei unterschiedliche Konzepte der ersten Webseiten-Elemente zu erstellen. Anschließend wurden diese verglichen und sich für eines entschieden. Dieses erste Grundkonzept wurde das Grundgerüst für den späteren Verlauf der Designarbeit.

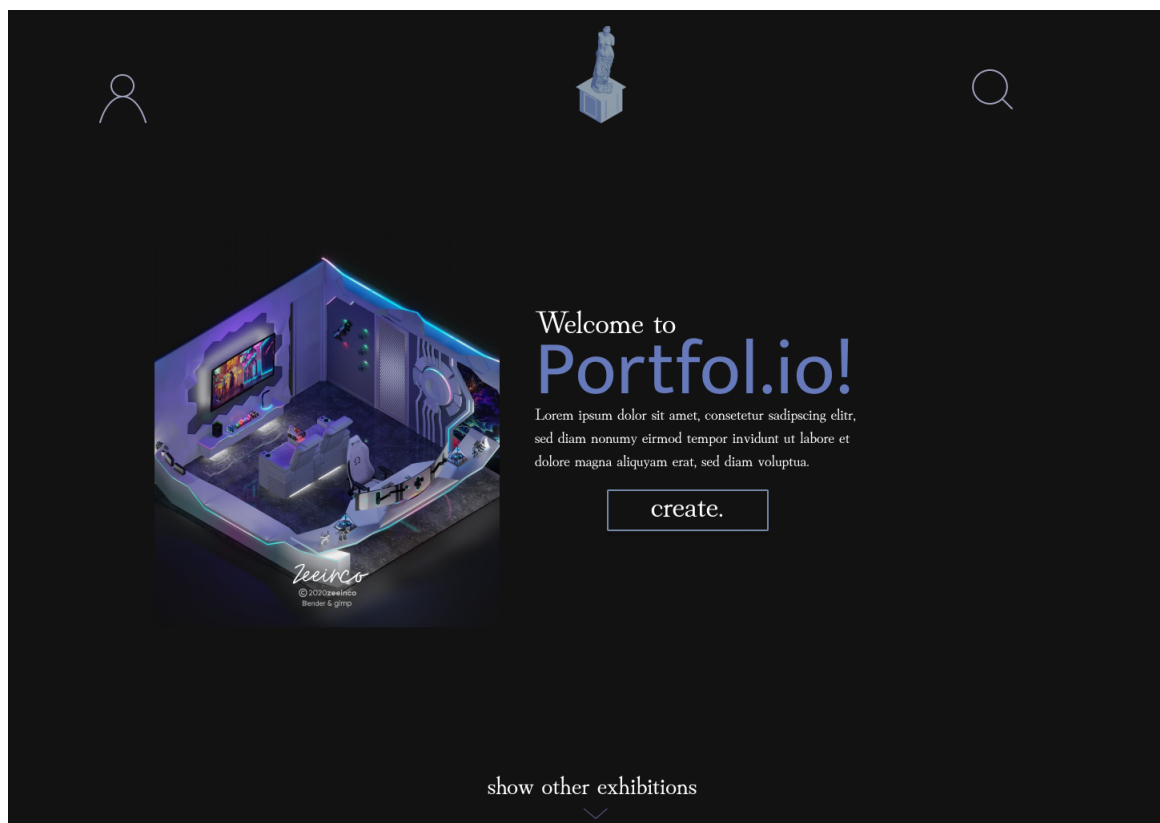


Abbildung 7: Design Konzpet 1 Page 1

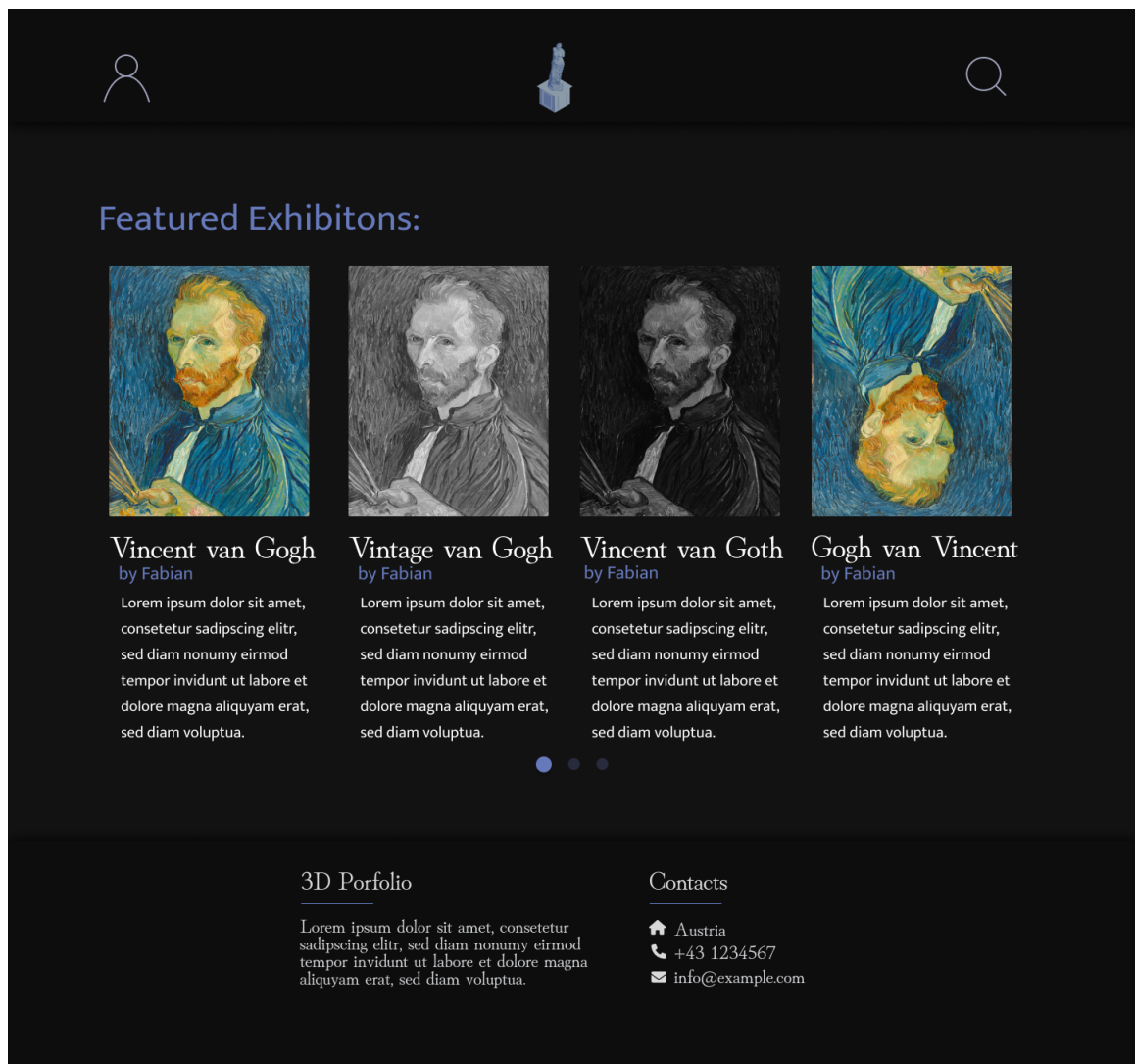


Abbildung 8: Design Konzpet 1 Page 2



Abbildung 9: Design Konzept 2

8 Zusammenfassung

Aufzählungen:

- Itemize Level 1
 - Itemize Level 2
 - Itemize Level 3 (vermeiden)
- 1. Enumerate Level 1
 - a. Enumerate Level 2
 - i. Enumerate Level 3 (vermeiden)

Desc Level 1

Desc Level 2 (vermeiden)

Desc Level 3 (vermeiden)

Literaturverzeichnis

- [1] P. Rechenberg, G. Pomberger *et al.*, *Informatik Handbuch*, 4. Aufl. München – Wien: Hanser Verlag, 2006.
- [2] Association for Progressive Communications, „Wireless technology is irreplaceable for providing access in remote and scarcely populated regions,” 2006, letzter Zugriff am 23.05.2021. Online verfügbar: <http://www.apc.org/en/news/strategic/world/wireless-technology-irreplaceable-providing-access>
- [3] Angular Team, „What is Angular,” letzter Zugriff am 20.02.2023. Online verfügbar: <https://angular.io/guide/what-is-angular>
- [4] ReactiveX Team, „Documentation Introduction for ReactiveX,” letzter Zugriff am 24.02.2023. Online verfügbar: <https://reactivex.io/intro.html>
- [5] Webpack Team, „Documentation for Webpack,” letzter Zugriff am 24.02.2023. Online verfügbar: <https://webpack.js.org/concepts/>
- [6] Angular Team, „Angular ngModules,” letzter Zugriff am 20.02.2023. Online verfügbar: <https://angular.io/api/core/NgModule>
- [7] J. Team, „Webstorm,” letzter Zugriff am 21.02.2023. Online verfügbar: <https://www.jetbrains.com/de-de/webstorm/>
- [8] ThreeJs Team, „Fundamentals of ThreeJs,” letzter Zugriff am 26.02.2023. Online verfügbar: <https://threejs.org/manual/#en/fundamentals>
- [9] Khronos Foundation, „Webgl - Getting Started - Wiki,” letzter Zugriff am 26.02.2023. Online verfügbar: https://www.khronos.org/webgl/wiki/Getting_Started
- [10] „A-Frame Wikipedia,” letzter Zugriff am 1.03.2023. Online verfügbar: [https://de.wikipedia.org/wiki/A-Frame_\(Framework\)](https://de.wikipedia.org/wiki/A-Frame_(Framework))
- [11] M. Team, „Cinema4D,” letzter Zugriff am 22.02.2023. Online verfügbar: <https://www.maxon.net/de/cinema-4d>
- [12] I. S.-M. F. Schwab, „Ideenfindung,” in *Systemplanung und Projektentwicklung*, Wien, 2013, ch. 2, letzter Zugriff am 1.03.2023.
- [13] P. D. K. Kuenen, „User Experience Design,” letzter Zugriff am 3.03.2023. Online verfügbar: <https://wirtschaftslexikon.gabler.de/definition/user-experience-design-100263/version-368987>
- [14] P. D. F.-R. Esch, „Corporate Identity,” letzter Zugriff am 3.03.2023. Online verfügbar: <https://wirtschaftslexikon.gabler.de/definition/corporate-identity-31786>
- [15] —, „Corporate Design,” letzter Zugriff am 3.03.2023. Online verfügbar: <https://wirtschaftslexikon.gabler.de/definition/corporate-design-30453>

Abbildungsverzeichnis

1	ThreeJs Logo	7
2	Die Struktur von ThreeJs	7
3	ThreeJs Logo	7
4	Der Unterschied zwischen AABB und OBB	13
5	Don Knuth – CS Allfather	15
6	OberflächenDesign: Prototypen in Figma	18
7	Design Konzpet 1 Page 1	19
8	Design Konzpet 1 Page 2	20
9	Design Konzept 2	21

Tabellenverzeichnis

1	Ein paar tabellarische Daten	14
---	--	----

Quellcodeverzeichnis

1	Angular Three - Komponentenbasiertes 3D Szenen in HTML	9
2	Angular Three - App Cube	9
3	Some code	15

Anhang