

WELTPORTALE: INTERAKTIVE 3D WEB-ANWENDUNG ZUR INFORMIERUNG DEN REISENDEN ÜBER DIE NATURKATASTROPHEN AUF DER ERDE

Serhiy Bolkun & Hamed Sadegh

Technical Report

Master Angewandte Informationstechnologien
Studienrichtung Computergrafik und Medientechnologien
WS-2021/22

ZUSAMMENFASSUNG

Im Projekt Weltportale werden die modernsten Technologien, wie Angular und Firebase, die von Google entwickelt sind, aus dem Bereich Webentwicklung, angewendet. Entstanden ist das Projekt in der Veranstaltung “3D-Programmierung” unter Vorführung von Marco Block-Berlitz im Wintersemester 2021/22 an der HTW Dresden. Das Thema basiert sich auf Klimawandel und muss möglichst interaktive, innovative und intuitive Ideen verfolgen. Beim Aufruf der Anwendung wird ein kleines Universum simuliert, in dem man sofort die kritischen Orte, als Rot markierte Punkte auf dem 3D Modell von Erde abgebildet sind. So bekommt man schnell ein Überblick über aktuelle Geschehen, als auch beim Anklicken auf diese Orte die ausführlichen Informationen. Wobei man kann auch schnell und bequem die Infos per Soziale Netzwerke teilen, als auch Kommentare hinterlassen. Bei Letzteren muss man ein Account anlegen oder mit Google authentifizieren lassen. Als auch die Aktualisierung von Daten passiert in Echtzeit. Wobei die Daten per REST API-Schnittstelle geholt werden. Für statistische, als auch analytische Zwecke steht ein Filter zur Verfügung, der nach Land, Datum (von bis heutzutage) und Katastrophentyp sortieren kann. Das Projekt ist aber eingeschränkt auf Desktop Bildschirme, als auch Google Chrome Browser. Diese Einschränkungen kann man aber in nächsten Versionen reduzieren und das Projekt weiterentwickeln.

1. MOTIVATION UND EINFÜHRUNG

Die Inspiration für den Projekt würde aus der oft besuchten Webseite www.awwwards.com [1] geholt. Dort versuchen die Webentwickler ihre besten und kreativsten Lösungen in Projekten darzustellen. Man sieht dort die Trends, als auch die Highlights von Profis. So wollten wir auch, als Team von zwei Personen unsere Fähigkeiten unter Beweis stellen.

Die Kenntnisse beim Webentwicklung haben wir schon seit mehreren Jahren gesammelt, jedoch nicht in Kombination mit 3D. Das hatten wir, als nächste Stufe für unsere persönliche Entwicklung in diesen Bereich gesehen. Da wir wollen uns Richtung “Virtual Reality” als nächstes bewegen, da momentan es ist ein Trend, die durch Covid-19 entstanden hat und entwickelt sich rasant schnell.

2. TECHNISCHER AUFBAU

Die Werkzeuge sind so ausgewählt damit wir nicht von Null anfangen sollen und unseren Entwicklungsprozess ganz bequem gestalten können. So haben wir der modernsten ausgestatteten Technik benutzt, als auch Software mit konfigurierten Plugins, um die Entwicklung zu beschleunigen.

Für Projektmanagement System haben wir uns für Jira in der Verbindung mit GitHub entschieden. Visual Studio Code haben wir als Editor benutzt mit vorinstallierte Node.js. Der anderen Software ist Git Bash und Google Chrome Browser.

Als Fundament für unserer Projekt haben wir Angular 13.0.1, in der Verbindung mit NoSQL Datenbank Firebase 9.4.1 benutzt. Google Firebase

bietet uns das Hosting an, wo wir unser fertiges Projekt auch gehostet haben [14].

Die Programmiersprachen die in Projekt angewendet sind JS (vor allem 3D Bibliothek Three.js), Typescript, HTML, CSS, als auch GLSL für Shader Programmierung und Firebase CRUD Abrufe. Die Daten über Katastrophen haben wir uns per REST API-Schnittstelle aus öffentlich zugänglicher Ressource von OCHA geholt. Es bietet uns kostenlos max. 1000 Abrufe pro Tag mit max. 1000 geholten Datensetzen, was für den Entwicklung komplett ausreichend ist.

2.1 Projektausarbeitung

Für unsere Ideen haben wir Papier Prototype benutzt. Und alle ‘Muss’ und ‘Nice-to-have’ Kriterien dort visualisiert. Danach haben wir die Aufgabenbereiche zwischen mir und Hamed in 3 Gruppen verteilt: Management, Backend und Frontend. Basierend auf der Zeitplanung und Arbeitsablauf (2-wöchigen Sprints) durch agile Softwareentwicklung haben wir mit dem Aufbau des Projekts angefangen.

2.2 Projektentwicklung

Zuerst haben wir das Projekt im Angular generiert [2] mit dem Befehl **“ng new projektname”** und entsprechend auch Komponente **“ng generate component componentname”** als auch Dienste **“ng generate service servicename”** und Pipes **“ng g p pipename”** [3] automatisch gebildet. Jeder einzelne Teil kann man in unserer GitHub Projekt [4] anschauen unter Pfad ‘src/app/components’, ‘src/app/services’ und ‘src/app/pipes’. Alle externe benötigte Module sind ebenfalls in README.md Datei unter dem Kapitel “Installation commands for development” zu finden.

Als nächstes haben wir auf unsere API-Schnittstelle [5] konzentriert und haben die Daten, als json Datei in Array gepackt. Der Service haben in Datei **“src/app/services/api-reliefweb.service.ts”** ausgelagert, so kann man schnell von jeder Komponente zugreifen, falls man braucht.

In dritten Schritt haben wir mit unserer Hauptkomponent **“dashboard”** beschäftigt. Hier haben wir unsere 3D Modell von Erde gebildet. Am Anfang haben wir ein Canvas für gesamte Bildschirm Fläche angelegt. Mit Hilfe von

Dokumentation [6] als auch schöne Beispiele aus YT [7, 8], haben wir eine Geometrie ‘Sphere’ mit drauf gelegter Textur erzeugt. Danach mussten wir die ‘lan’, ‘lot’ aus dem API geholten Daten in die Kugelkoordinaten umwandeln durch die Formeln [9].

$$x = r * \sin(\text{Theta}) * \cos(\text{Phi})$$

$$y = r * \sin(\text{Theta}) * \sin(\text{Phi})$$

$$z = r * \cos(\text{Theta})$$

Diese Formeln müssen wir jedoch umschreiben, um im Browser korrekte Darstellung zu erzeugen. Die Punkte z und y sind getauscht, als auch Phi und Theta. Und x bekommt einer Minus Zeichen wegen Browser Koordinatensystem Regeln.

$$\text{Theta} = (\text{lon} + 180) * (\text{Pi} / 180)$$

$$\text{Phi} = (90 - \text{lat}) * (\text{Pi} / 180)$$

$$x = - (r * \sin(\text{Phi}) * \cos(\text{Theta}))$$

$$y = r * \cos(\text{Phi})$$

$$z = r * \sin(\text{Phi}) * \sin(\text{Theta})$$

So haben wir uns zwei Punkte ausgesucht und ein Normalisierte Kurve von Punkt A auf Punkt B gezeichnet. (siehe Abb. 1)

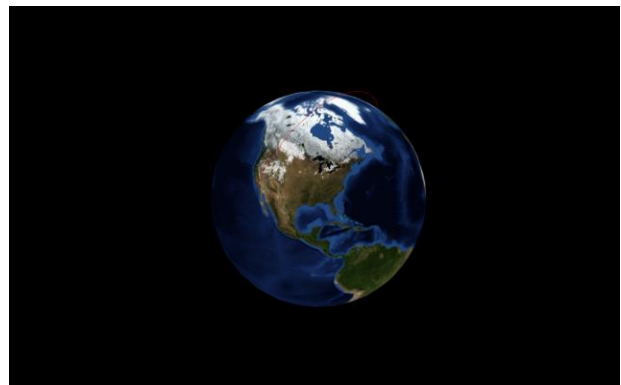


Abbildung 1

Als das geschafft wurde, haben wir schon der GUI, ‘Event Listeners’ für den Erde Rotation und Skalierung implementiert. Da unsere Erde viel zu dunkel erscheint haben wir eine Atmosphäre mit Fragment und Vertex Shader gestaltet. Als auch letztendlich alle aktuellen Orte mit Katastrophen auf der Kugel gezeichnet die aus API-Quelle vorkommen. (siehe Abb. 2)



Abbildung 2

Da die Kurven keine richtige Beziehung zueinander haben, sind Sie ausgetauscht mit Normierten Vektoren. So haben wir durch 'Raycaster' [10] ein Klick Ereignis erzeugt, die den Benutzer ermöglicht selbst Orte auszuwählen, die ihm interessieren. So taucht auch der Modal Fenster mit ausführlichen Informationen. (siehe Abb. 3)

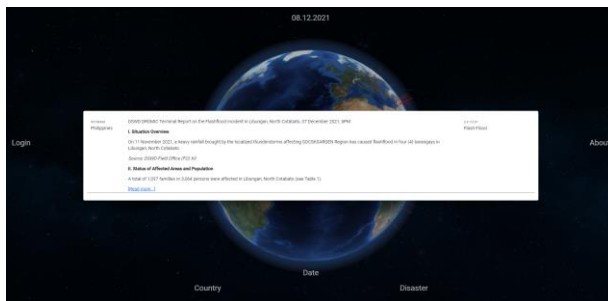


Abbildung 3

Danach haben wir den GUI angepasst, Musik [11] eingebaut, Sterne animiert mit Zufallszahlengenerator, 'Skybox' [12] eingebaut. Datenbank Verbindung [13] hergestellt für den Registrierung, um Kommentare hinterlassen zu können, als auch die animierten Übergänge. (siehe Abb. 4)



Abbildung 4

Als letzter Schritt haben wir 'OnMouseOver' Ereignis eingebaut, um Titel den Benutzer anzuzeigen. Diese kurze, aber wichtige Info

ermöglicht den Benutzer zu entscheiden, ob es was für Ihn von Bedeutung ist oder nicht. Ein rot markierter Titel ist aktuell und blau bedeutet das der Artikel liegt in der Vergangenheit. Dieser Abtrennung spielt eine große Rolle beim Filtern von Daten. (siehe Abb. 5)



Abbildung 5

3. TESTS UND BEWERTUNGEN

Die Webanwendung war mehrmals bei den verschiedenen Etappen während der Entwicklung getestet. Hier hat die größte Rolle die Usability Tests gespielt. Beim fertigen Produkt waren die Aufgaben für die Testpersonen so ausgedacht, dass man im Hintergrund durch alle wichtigen Funktionalitäten der Webanwendung durchgegangen hat und ans Ziel gekommen. Die Aussagekräftige Tests müssen auch in der nächsten Version der Webanwendung implementiert werden. Da es eine lange und mühsame Routine ist, braucht man mehr Personal dafür und eine gewisse Zeit für die Entwicklung.

Die Bewertung erfolgte nach 5 Kriterien, wie Leistung, GUI, Animationen, Steuerung und Musik. Der Skala von Note lag zwischen 1 (schlecht) und 5 (sehr gut). Der Durchschnittsnote von getesteten Personen liegt bei 4,6 was ein positives Ergebnis liefert. Der Anmerkungen lag an manchen GUI Stellen, als auch Steuerung der Welt Modell. Obwohl die Hinweise für den User beschrieben sind, lag der Problematik auch auf die Konfiguration der Browser, als auch das unterschiedliche Niveau der Benutzer Erfahrungen beim Umgang mit solchen Programmen.

4. ZUSAMMENFASSUNG UND ZUKÜNFTIGEARBEITEN

Das Ziel wurde erreicht, da der primär Anwendungsgruppe eine sehr gute Bewertung abgegeben hat. Und die meisten Personas sind

zufrieden, da ihre Meinung nach der Anwendung bietet die Funktionalitäten, die Sie an erste Stelle brauchen. So haben wir auch alle “Muss” Kriterien erfüllt [14], als auch neuen Erkenntnissen für unserer Zielpersonen gewonnen. Dieser Erfahrungen nehmen wir mit für unsere nächste Projekte. Und versuchen den Fokus auf modernste Technologien als auch Trends zu behalten, um unsere Kunden der besten Produkte, die wir entwickeln können anzubieten, als auch seine Bedürfnisse am kreativsten Wege zu verwirklichen. (siehe Abb. 6)



Abbildung 6

Der Projekt Weltportale ist gut strukturiert und vor allem skalierbar und hat Potenzial für weiter Entwicklung. Man kann noch viele zusätzliche Funktionalitäten einbauen, als auch kleine 3D Miniwelten weiterentwickeln. Die zusätzlichen Tests weiterschreiben. Aber das wichtigste ist und bleibt der Zeit und Ideen, die man gemeinsam in Team auf beste Art und Weise in die Welt bringt, damit die Nutzer profitieren können.

LITERATUR

- [1] Motivation und Inspiration
<https://www.awwwards.com/sites/moment-zero>
<https://2050.earth/>
<https://www.lefigaro.fr/fig-data/covid-19>
<https://github.com/>
- [2] Getting Started
<https://angular.io/guide/setup-local>
- [3] Persönliche Webseite mit Angular Erfahrungen
<https://bolkun-2d667.web.app/article/Ia5oQGL98azWfVIZtSd4>
- [4] GitHub
<https://github.com/Bolkun/weltportale>
- [5] Dokumentation von API
<https://apidoc.rwllabs.org/parameters>
- [6] Dokumentation von Three.js Bibliothek
<https://threejs.org/docs/index.html#manual/en/introduction/Creating-a-scene>
<https://medium.com/geekculture/hello-cube-your-first-three-js-scene-in-angular-176c44b9c6c0>
- [7] Creating earth, and adding cities to a planet with Three.js
<https://www.youtube.com/watch?v=2pUzJOfeKVE>
- [8] Intermediate Three.js Tutorial - Create a Globe with Custom Shaders
<https://www.youtube.com/watch?v=vM8M4QIoVL0>
- [9] Kugelkoordinaten
<https://de.wikipedia.org/wiki/Kugelkoordinaten>
- [10] Three.js Raycasting for Mouse Picking [Checkers 3]
<https://www.youtube.com/watch?v=3SGdxk7HMGw>
- [11] Space Melody Resurrection (Remix)
<https://www.youtube.com/watch?v=myXVVrvPT1w>
- [12] Skybox Tutorial
https://www.youtube.com/watch?v=cp-H_6VODko
<https://opengameart.org/content/space-nebulas-skybox>
- [13] AngularFire Quickstart
<https://github.com/angular/angularfire/blob/master/docs/install-and-setup.md>
- [14] Fertig gehostete Projekt
<https://weltportale-7b2cd.web.app/>