**Rock-Paper-Scissors-Game**

**Web-App**

## Einführung

Ich habe im Rahmen vom Modul 150 ein Project meiner Wahl gestalten können.   
Ich habe eine PWA entwickelt, mit Hilfe welcher man das klassische «Schere, Stein, Papier»-Spiel gegen einen **BOT** spielen kann.

## Technologie-Entscheid

Da ich auf der Arbeit mehrheitlich mit C# arbeite, wollte ich eine Abwechslung zum Arbeits-alltag und habe mich somit für eine Web-App entschieden.  
Will ich persönlich ein Mensch bin, dem Glücksspiele grossen Eifer bereiten, habe ich mich für ein Spiel, welches International bekannt ist entschieden: Schere, Stein – Papier. Dank seiner Bekanntheit kann ich meinen User-Traffic möglichst hochhalten. Dies könnte mir in Zukunft eine Monetarisierung bringen.   
Anfangs wollte ich meine App auf der Bootstrap-Basis gestalten, doch mein Mitschüler «David Gataric»-welcher lange Erfahrung mit Web-Apps hat, hat mir geraten, das Projekt mit Angular umzusetzen.

Aus dem Grund, dass die Handys heutzutage immer beliebter und zugänglicher werden, habe ich mich entschieden, auf diese Zielgruppe zu konzentrieren. Deswegen werde ich mein Programm mit der PWA-technologie umsetzen.   
PWA (Progressiv Web App) eignet sich bestens, um eine Web-Applikation für Smartphone sowie für Desktop zu entwickeln. Unter anderem unterstützt PWA offline-Funktionalitäten.   
  
Ich habe ich mich dazu entschieden, mein Produkt vorwiegend für Handy-Benutzer zu gestalten, aus diesem Grund verfolge ich den Ansatz «**Mobile-First».**  
Dies aus dem Grund, dass es um einiges Schwieriger ist, die Information auf einem grösseren Bildschirm darzustellen, als sie übersichtlich auf einer kleineren Fläche darzustellen. (Als Referenz verwende ich das Iphone X - Modell)

## Management-Tools

Für mein Project habe ich eine verschiedene Management-Tools verwendet. Diese sowie auch meine Entwicklungs-umgebung, beschreibe ich detaillierter in diesem Kapitel.

## GitHub

Den Source-Code pflege ich mit Hilfe von Git-hub. Git-hub ist eine Entwicklungs-Code Management Tool.

## Trello

Im Git hub Reporsitory, verlinke ich ebenfalls auf mein **ProductBacklog** imTrello verlinke. Trello verwende ich aus dem Grund, dass mir dieses Tool in einem der Überbetrieblichen-Kurse vorgestellt worden ist. Da dieses Tool gratis zur Verfügung steht und ich bereits Erfahrungen damit habe, habe ich mich für dieses entschieden.

### Entwicklungs-umgebung

Als Entwicklungs-umgebung, habe ich mich für Visualstudio-Code entschieden, da dies die geeignetste Umgebung fürs Arbeiten an Web-Projekten ist. Mir ist die Umgebung bekannt und ich habe bereits mehrere Projekte damit realisieren können. Ebenfalls kann ich verschiedene Formatierungs-Extensions herunterladen und es enthält eine Benutzeroberfläche für das Verwalten von Git-Befehlen.

## Vorgehen

Ich habe damit angefangen, dass Default-projekt der Angular CLI anzupassen. Ich habe alles Überflüssige (was für mein Projekt nicht notwendig ist) gelöscht. Ich habe bloss den Header, die Buttons und den footer sowie jeweils die styles übriggelassen, aber für mein GUI angepasst. Anschliessend habe ich meine Bilder, welche ich im Internet ausgesucht habe, als Hintergrund der jeweiligen Buttons gesetzt.

### 4.1 Material Design

Ich habe als nächstes das Material Design für meine Applikation konfiguriert. Ich habe es per Terminal im Projekt mit folgendem command hinzugefügt:  
ng add @angular/material

### 4.2 Projektmanagement

Bei der Projektmanagement-Methode habe ich mich für Scrum entschieden.   
Scrum eignet sich vor allem für Projekte, welche unendlich-langen Entwicklungsprozess haben.  
Da mein Produkt später auf einem Server gehostet wird und ich in der Zukunft vorhabe, das Spiel zu erweitern (mehr Elemente nebst Schere, Stein und Papier), eignet sich diese Methode bestens.  
Ich habe dabei mich selbst als Product Owner, Scrum Master und Entwickler.   
Da wir Wöchentlich unseren Fortschritt der Klasse präsentieren müssen, habe ich die Sprint-Dauer auf «Wöchentlich» gesetzt. Das bedeutet das ich wöchentlich einen Stand haben muss, wo ein Teilstück der Arbeit funktionell vollständig abgeschlossen ist.

## GUI

Beim Erstellen des GUIs habe ich mich an einen Vorsatz gehalten: Weniger ist mehr.

Ich wollte, dass die Bedienung dem Nutzer auch möglichst intuitiv fällt. Aus diesem Grund habe ich mich für drei grosse Buttons auf dem GUI entschieden.   
Das Scoreboard ist oben-mittig an der Website verankert.   
Die Resultate-Dialoge erscheinen mit der jeweiligen Meldung (gewonnen, verloren oder unentschieden). Dazu gibt es jeweils ein passendes GIF.

Die Dialoge habe ich in einem Array erfasst und gebe jeweils zufällig einen der Texte aus.  
Der Dialog verschwindet jeweils nach zwei Sekunden.

## Kurzfassung des PA-Berichtes

Ich habe im Rahmen vom Modul 150 ein Projekt meiner Wahl entwickeln dürfen. Da der Technologie-entscheid frei zu wählen war und ich eine Web-App (vorwiegend für Mobile-Nutzer) gestallten wollte, habe ich mich für Angular entschieden. Da ich meine App für Handy-Nutzer optimieren will, (da der meiste Internetverkehr über Smartphones verläuft) liegt der Entscheid für PWA nah.  
Nachdem ich mein Projekt geplant hatte, habe ich mit dem Initialisieren des Projekts angefangen. Da Angular fakultativ eine Standart-Website erstellt, habe ich dies getan. Danach habe ich das Projekt von all den Elementen, welche für mein Projekt nicht notwendig sind – befreit.  
Als nächstes habe ich mir Bilder aus dem Internet gesucht, welche aussagekräftig und lustig sind. Ich habe diese Bilder jeweils als Hintergrundbild für die jeweiligen Buttons gesetzt.   
Ich habe danach versucht mittels Typescript, die Logik für das Scoreboard zu implementieren. Nachdem langen refactoring und Zeit Investition, habe ich es fehlerlos commiten können.   
Des Weiteren habe ich das Resultat-Dialog implementiert. Ich wusste anfangs nicht, wie ich die Ausgabe gestallten sollte. Ich wollte etwas «alert ()»-mässiges implementieren, doch das standart-alert ist einfach hässlich und nicht mehr zeitgemäss. So machte ich mich auf die Suche nach Ideen im Internet, zum Darstellen von Dialogen. So bin ich auf «Sweetalert» gestossen. Ich habe mir danach die Dokumentation vom Sweetalert durchgelesen und versucht in mein Projekt miteinzubeziehen. Passend zum Dialog erscheint ebenfalls ein (lustiges) GIF. Da ich noch neu in Angular bin, ist es mir Anfangs schwergefallen, den Sweetalert sowie all die Komponente richtig mit einzubeziehen. Doch als ich eine Komponente hinzugefügt hatte, kannte ich den Ablauf und konnte somit das Prinzip verstehen und nachvollziehen.

Auf ein Zeil musste ich leider verzichten. Ich habe aus Zeitlichen Gründen nicht mehr geschafft, eine Session für Online-Spiele zu erstellen. Man kann somit (stand 7.01.2020) nur gegen den BOT nicht aber gegen online-Spieler spielen.

Ebenfalls habe ich eine Benutzer-Statistik eingebaut. Der Benutzer sieht unterhalb vom Scoreboard, seine jeweilige gewinn-Chance sowie auch seine Lieblingswahl der jeweiligen Session.

## Quellenverzeichnis

Sweetalert: <https://sweetalert2.github.io/>

Trello: trello.com

Github: github.com  
PWA: https://de.wikipedia.org/wiki/Progressive\_Web\_App

Copyright:

Rock: <https://www.flickr.com/photos/32279598@N02/33507875820>

Paper: <https://pxhere.com/en/photo/869916>

Scissors : <https://www.wannapik.com/vectors/3081>

**Arbeits-Journal**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Datum | Was habe ich gemacht? | Was habe ich gelernt | Wo hatte ich Schwierigkeiten? | Was ist pendent? |
| 12.11.2020 | Ich habe das PB erstellt | Ich muss das PB viel einheitlicher Definieren (ganze Funktionen als PBI, nicht nur Teile davon) | Ich habe das PB zu wenig genau definiert.  Die Funktionalität vom Produkt ist noch zu mager | Ich muss die PBI anpassen |
| 12.11.2020 | Ich habe einen Technologie-Entscheid getroffen (Bootstrap) | Ich hatte durch fehlendes Wissen, nicht von Anfang an die richtige Technologie gewählt. | Ich habe mich nach langem Überlegen doch gegen Bootstrap entschieden, da ich die App Mobile-User freundlich gestalten wollte. Nach einer Diskussion mit Arbeits-Kollegen, habe ich mich für eine Angular-pwa entschieden. | Ich muss die neue Technologie anlegen, da Bootstrap für mich nicht mehr aktuell ist. |
| 12.11.2020 | Ich habe ein leeres Git Repo erstellt | Ich habe dies schon mehrere Male gemacht | - | Ich muss das Repo stehts mit Commits pflegen |
| 19.11.2020 | Ich habe das Angular Start-Projekt heruntergeladen und pwa extension hinzugefügt | - | - |  |
| 19.11.2020 | Ich habe das .yml File erstellet, welches für mich automatisch ein build anstosst | Wie man eine Workflow Datei im Github hinzufügt | Die Syntax war für mich bis anhin unbekannt und ich musste mich erst mit dieser auseinandersetzen. | - |
| 19.11.2020 | Ich habe einen init Commit ins Repo gepusht | - | - | - |
| 26.11.2020 | Ich habe passende Bilder für meine Buttons gesucht | - | - | Ich muss diese Bilder als Hintergrund für die Buttons verwenden |
| 26.11.2020 | Ich habe die Dokumentation weiter gepflegt | Ich sollte die Dokumentation immer wieder anpassen. So habe ich weniger Pendenzen. |  | Immer weiter pflegen. |
| 03.12.2020 | Ich habe das default-Projekt von Angular soweit angepasst, dass ich nur das nötigste für mich übrig gelassen habe |  | Ich habe anfangs zu viel gelöscht, was die ganze Formatierung beeinflusste | - |
| 03.12.2020 | Ich habe die passenden Bilder für das GUI herausgesucht und als Hintergrund bei den Buttons gesetzt. |  | Ich habe es nicht geschafft, die grösser der Bilder passend darzustellen. | Ich werde bei Gelegenheit mich mit der Platzierung der Bilder befassen. |
| 10.12.2020 | Ich habe die Syles der Site angepasst (Schrift, Titel, Schriftgrösse, Schriftfarbe etc.) angepasst |  | Ich musste erst wieder die HTML Syntax nachschauen, da ich diese wegen mangelhaften nutzen vergessen habe. |  |
| 10.12.2020 | Ich habe das mat-toggle-slide Modul hinzugefügt. | Wie man ein Modul ins Angular integriert | Ich konnte die Labels beim «toggle» nur jeweils auf eine Seite platzieren (davor oder danach) |  |
| 17.12.2020 | Ich habe das Score-board für mein Spiel designed | Wie man Daten aus dem Backend im Frontend darstelle. | Mit dem CSS (design): Das darstellen vom Scoreboard, wie ich es mir vorgestellt habe, erforderte viel CSS. | Ich muss die Logik hinter dem Scoreboard implementieren. |
| 17.12.2020 | Ich habe Spiel-dialoge sowie Resultat-Dialoge mit Sweetalert erstellt. | Wie ich schöne Dialoge im Web darstellen kann (nicht mit «alert ()» Befehl) | Ich habe lange gebraucht, um zu verstehen, wie ich das Modul richtig nutzen kann | - |
| 17.12.2020 | Ich habe die Logik für das Scoreboard und somit für die Logik im Spiel implementiert. | Ich kann eine einfache Logik im typescript umsetzen. | Das Scoreboard hat bei mir anfangs nicht aktualisiert. Es ignorierte teilweise eine Vielzahl an Runden. | - |
| 20.12.2020 | Ich habe weiter an der Dokumentation gearbeitet. |  |  |  |
| 06.01.2020 | Ich habe die Statistik (Gewinnchance und Lieblingswahl) implementiert. | Ich kann die percentage-pipe vom Angular anwenden. Ich kann eine Mathematische Formel erstellen. | Anfangs hatte ich als Gewinnchance oft das Ergebniss: «infinity». Nach langem Debugen, habe ich festgestellt, dass wenn der Benutzer die erste Runde verliert, es nach meiner Formel nicht aufgeht, da ich sonst durch 0-teile. | Ich muss die Statistik für die Lieblingswahl öfters testen. Es ist noch nicht ausreichend getestet worden. |
|  |  |  |  |  |