

Projektarbeit - Webprogrammierung

KO-Runde

Dozent: Herr Ulrich Hauser-Ehninger

Klasse: DBM17VZ

Abgabe: 16. Juni 2019

Autoren:

Becker Reto, Bolli Colin, Flach Maximilian, Jurkovic Patrik, Linvers
Sascha, Marquart Christoph, Marxer Jan, Nikolic Katarina, Raschle
Florian, Rukunakumar Arthith, Wullschleger Dominique

Inhaltsverzeichnis

Einleitung	2
Projekt	2
Aufteilung	2
Eingabemaske	3
Initialdarstellung	4
Init	4
Tabelle Aufbau	4
KO-Logik	5
Final Page	6

Einleitung

Die Grundidee der Erstellung dieser Projektarbeit basiert auf den Anforderungen des Moduls „Webprogrammierung“ der HTW Chur. In diesem Kapitel wird das Projekt, die Aufteilung, sowie ein Überblick über das Gesamtkonzept beschrieben.

Projekt

Das Ziel dieser Projektarbeit bestand darin, eine Lösung bzw. eine Web-Page zur verwaltung von Endrundenspielen zu entwickeln. Dabei sollen Teilnehmer, wir definierten diese als “Teams”, hinzugefügt, geändert und wieder gelöscht werden können.

Der Prozess soll dabei wie folgt durchgeführt werden:

1. Anlegung der Teams
2. Automatisches erstellen von Paarungen bzw. Spielen
3. Eingabe der Ergebnisse
4. Automatisierte eliminierung der Verlierer
5. Nächste Runde mit neuen Paarungen der vorherigen Sieger
6. Kürung des Siegers

Die Schritte 2. bis 5. werden solange wiederholt, bis nur noch ein Team verbleibt, mit welchem der Schritt 6. erfolgt.

Aufteilung

Das Team teilte sich zu Beginn des Projektes in kleiner Untergruppen auf, welche sich den verschiedenen Aspekten der Arbeit widmeten. Die Daten und verschiedenen Variablen wurden von den Untergruppen mit statisch Werten gefüllt. Dadurch konnte effizienter an der Lösung weiterentwickelt werden, ohne auf die anderen Untergruppen zu fest angewiesen zu sein.

Die Aufteilung wurde wie folgt definiert:

- | | |
|---------------------------------|---|
| 1. Eingabemaske: | Arthith Rukunakumar, Sascha Linvers, Patrik Jurkovic |
| 2. Initialdarstellung: | Reto Becker, Christoph Marquart, Colin Bolli |
| 3. Funktion Halbierung Tabelle: | Florian Raschle, Dominique Wullschleger, Katarina Nikolic |
| 4. KO-Logik und Final Page: | Jan Marxer, Linus Niederhauser, Maximilian Flach |

Eingabemaske

In der Index.html Datei wird dem Nutzer ein Eingabefenster, welches die Eingabe von Teams erlaubt, dargestellt. Dabei wird am oberen Bildschirmrand eine Navigationsbar, welche mit Hilfe von Bootstrap umgesetzt wurde, angezeigt (Code Zeile 13 – 26). Durch das Menü kann der Nutzer zwischen den Punkten «Home, Ergebnisse Eintragen und Gewinner» wechseln.

Durch die Eingabe von Teamnamen im Eingabefeld und der anschliessenden Bestätigung mit dem «Add Team» Knopf, welcher die ID «team» hat und auf der Zeile 32 dargestellt wird, kann der Nutzer Teams hinzufügen. Bei der Betätigung des «Add Team» Knopfes wird mit Hilfe von JQuery in der submit.js Datei ein Befehl ausgeführt, welcher den eingegebenen Teamnamen in der teamnames.php Datei speichert. Bei der erfolgreichen Speicherung vom Team wird das Eingabefeld geleert (Zeile 13-25). Dabei wird dem eingegebenen Team eine ID verteilt, welche den nächst höheren freien Wert aufweist (Zeile 6-11).

Ist jedoch bei der Eingabe ein Fehler aufgetreten, erscheint auf dem Bildschirm des Users der Text «Houston, We have a problem!» (Zeile 22-29).

Die bereits hinzugefügten Teams werden im Browser unter dem Text «Here are your Teams:» aufgelistet. Die Auflistung des Teams geschieht über den Tag in Kombination mit Elementen aus dem Bootstrap um die Darstellung zu verbessern. Dies geschieht über die ID «assignedgroups». Mit Hilfe der show.js Datei, welche die in der teamnames.php gespeicherten Teams durch die Funktion «receivedteamnames» (Zeile 10-26) abrufen.

Die Funktion weist eine if Schleife auf, welche die Meldung «No teams found. Just add some!» ausgibt, wenn keine Teams in der php Datei gespeichert sind.

Durch den «Delete all» Button mit der ID «delete» kann der Nutzer die gespeicherten Teams löschen. Dies geschieht über die «delete.js» Datei. Dabei werden die eingetragenen Teams im PHP gelöscht.

Durch den Code in Zeile 41, welcher mit der ID «nextpage» versehen ist, gelangt der Nutzer auf die nachfolgende Seite «initialdarstellung.html». Ab Zeile 44 werden verschiedene Bibliotheken wie Popper (Zeile 47) und Bootstrap (Zeile 49) eingebunden. Des Weiteren wird in Zeile 45 JQuery eingebunden, welches dafür sorgt das AJAX einwandfrei funktioniert. Zu guter Letzt wird in den Zeilen 51-54 eine andere Version von JQuery eingebunden, welche dafür sorgt, dass die Navigationsbar auch auf mobilen Endgeräten funktioniert.

Die Eingabeseite wurde mit wenigen Elementen erstellt um Verwirrungen für den Endbenutzer zu vermeiden. Die Seite wurde auf die wesentlichen Funktionen eingeschränkt und sorgt für eine selbsterklärende Bedienung.

Initialdarstellung

Init

Die init-Funktion wird im onload Event aufgerufen, das heisst, wenn die Seite im Browser aufgebaut wird (initialdarstellung.js, Zeile 55-68). Darin werden die Teams vom Server über ein ajax-Aufruf abgefragt, damit sie später für den Aufbau der Tabelle verwendet werden können.

Falls die Daten über den ajax-Aufruf nicht abgefragt werden können, wird mit Hilfe der Funktion «failedteamnames» ein Alert «Fail» ausgegeben (initialdarstellung.js, Zeile 124-125).

Zusätzlich wird überprüft, ob nur noch zwei Teams vorhanden sind, um auf die final_page weiter zu leiten (initialdarstellung.js, Zeilen 64-68).

Tabelle Aufbau

Die Funktion «recievedteamnames» wird aufgerufen, wenn der ajax-Aufruf in der init Funktion erfolgreich war und somit Daten von Teams erhalten wurden. In dieser Funktion werden die JSON-Daten in Javascript Objekte konvertiert und für jedes vorhandene Team die Funktion addTeamName aufgerufen (initialdarstellung.js, Zeilen 70-73).

Die Funktion «addTeamName» generiert anhand der Daten eines Teams die Tabelle in HTML. Je nachdem an welcher Position das Team in den Daten ist, werden gerade Teams

in der linken Spalte und ungerade in der rechten Spalte der Tabelle eingefügt. Zusätzlich zum jeweiligen Teamnamen, wird auch ein Inputfeld mit Typ «number» erstellt, wo die erzielten Punkte eingetragen werden können (initialdarstellung.js, Zeilen 74-122).

Der Inhalt dieses Inputfelds wird später dazu verwendet, um das Gewinner- bzw. das Verlierer-Team zu ermitteln.

KO-Logik

Das Prinzip der KO-Logik ist ziemlich simpel. Es treten zwei Parteien, in unserem Fall Teams, in einem Direktduell gegen einander an und nur der hervorgegangene Sieger kommt in die nächste Runde. Da es auch möglich sein sollte mit einer ungeraden Zahl von Teams zu starten wurden Wildcards mit eingeführt.

Bei einem Start mit einer ungeraden Anzahl von Teams werden die Paarungen der Begegnungen gleich hand gehabt wie bei einer geraden Anzahl von Teams. Jedoch bekommt ein Team in der ersten Runde keinen Gegner zugewiesen und ist somit automatisch für die nächste Runde qualifiziert.

Das Weiterkommen der Gewinner wurde im ersten Ansatz versucht über eine zwischen Liste zu realisieren. Dabei wurden alle Teams, welche als Sieger geflaggt wurden, in eine separate Liste gespeichert, anschliessend wurde die Liste, welche alle Teams beinhaltete, geleert. Die Liste mit allen Teams, welche nun leer ist, wurde mit dem Inhalt der zwischen Liste gefüllt und die zwischen Liste wurde darauf ebenfalls wieder geleert. Dadurch sind nur noch die Teams welche siegreich waren vorhanden und die nächste Runde kann durchgeführt werden. Durch dieses Verfahren, ist es im nachhinein nicht mehr nachvollziehbar, welches Team gegen wen spielte und wie das Ergebnis lautete, da die Teams beim ausscheiden gelöscht werden. Dies war allerdings nicht verlangt und konnte unberücksichtigt bleiben. Es stellte sich aber heraus, das diese Variante ziemlich umständlich ist. Zudem wurde, auch dank der Hilfe von Herr Hauser, eine effizientere Methode gefunden.

So wird nun direkt bei den Paarungen geschaut, welches Team verloren hat. Dafür wurde eine globale Variable erstellt mit dem Namen "removereq" (initialdarstellung.js, Zeile 3). Mit

einer If-Bedingung (initialdarstellung.js, Zeile 86-104) wird geschaut, ob das linke Team mehr Punkte erzielt hat, als das Rechte. Wenn dies zutrifft, wird das rechte Team direkt in dieser If-Bedingung via Ajax entfernt (initialdarstellung.js, Zeile 91). Falls das linke Team weniger Punkte erzielt hat als das rechte, passiert das selbe einfach umgekehrt in der else if Bedingung. So kann auch gewährleistet werden, dass ein Team, welches keinen Gegner hat (Wildcard), automatisch weiterkommt. Die Variable "removereq" wird um eins erhöht, wenn eine Bedingung zutrifft. Wenn die Ajax Anfrage erfolgreich stattgefunden hat, wird die Variable wieder um 1 verringert (initialdarstellung.js, Zeile 112). Dies wird bewerkstelligt, in dem nach Abschluss der Ajax Anfrage die Funktion "removereturned" ausgeführt wird (initialdarstellung.js, Zeile 92 & 100).

Final Page

Das Team hat sich zum Ziel gesetzt, eine separate Seite zu laden, sobald der definitive Gewinner evaluiert ist. Auch dafür wurde eine globale Variable erstellt. Diese trägt den Namen "teamsremaining". Sie hat am Anfang den Wert 0. In der selben If-Bedingung, in der auch die Gewinner der Paarungen ermittelt werden, wird diese Variable in jedem Fall um eins erhöht. Dies, da pro Paarung immer ein Team weiter kommt. Wenn diese Variable am Ende des Durchlaufs einen Wert hat, der grösser als 2 ist, wird die Variable wieder auf Null gesetzt. Denn wenn die Variable einen grösseren Wert als 2 hat, bedeutet dies, dass noch mehr als 2 Teams im Turnier sind und folglich noch mehr als eine Partie ansteht. Wenn die Variable den Wert 2 hat, wird am Ende dieses Durchlaufs die "Final Page" aufgerufen. Denn wenn die Variable zu Beginn des Durchlaufs den Wert 2 hat, bedeutet dies, dass die letzte Partie des Turniers ansteht und der Gewinner dieser Partie automatisch auch das Turnier gewonnen hat.

In der Final Page selber werden zwei Java Script Dateien verwendet. Eine ist für die Darstellung der Konfetti. Dieser Code wurde nicht selber geschrieben, sondern von <https://codepen.io/anon/pen/JMOQzE> bezogen. Mittels der zweiten Datei (winner.js) wird das letzte verbliebene Team via Ajax (Zeile 5) abgerufen und in der Seite dargestellt (Zeile 14). Sobald die Final Page dargestellt wird, wurde im Turnier ein Gewinner ermittelt und das ganze kann wieder von vorne beginnen.