



Tournamently

Die Webpage, welche im Rahmen dieses Projekts erstellt wurde, befasst sich mit den Vorrundenspielen bei Sportveranstaltungen. Hierbei werden die Teams in 4er-Gruppen eingeteilt, welche innerhalb der Gruppe gegeneinander spielen. Nach dem Spiel werden die Punkte verteilt, auf der Website Tournamently funktioniert dies mit 3 Punkte für den Sieg, 1 Punkt für Unentschieden und 0 Punkte fürs Verlieren. Bei Punktegleichstand entscheiden die Anzahl Tore die eingetragen wurden. Das stärksten Team jeder Gruppe zieht in die K.O. Phase.

Die Website ist so aufgebaut, dass sie über eine Startseite und vier Unterseiten verfügt. Diese sind Registration, Gruppenübersicht, Spielplan und Beendet.

Auf der Seite Registration kann ein Team hinzugefügt werden, diese Teams werden anschliessend in der Gruppenübersicht angezeigt. Für die Gruppen wird ein Spielplan erstellt, welcher auf der Seite Spielplan ersichtlich ist. Auf dem Spielplan werden nach der Austragung des Spiels das Endresultat eingetragen und bestätigt. Das Projektteam hat hier ebenfalls einen Bearbeitungsbutton eingefügt, damit falsche Erfassungen keine problematischen Konsequenzen davon tragen. Schliesslich werden die Punkte korrekt an die Teams verteilt, diese Übersicht findet auf der Seite Gruppenübersicht statt. Zum Schluss der Vorrunde werden die Gruppensieger, durch Bestätigung des Turnierendes, auf der Seite Beendet dargestellt. Nach der Betrachtung der Schlussresultate und des Siegers, wird die Möglichkeit gegeben, die gesamte Datenbank zu löschen und neu zu beginnen.

Das vorliegende Dokument bildet die technische Dokumentation und erklärt den Code, damit externe Programmierer diesen ebenfalls nachvollziehen können.



Grundgerüst

Bootstrap

Ein Bootstrap-Template mit dem Namen ***index.html*** bildet das Grundgerüst für die Webseite, bei welchem das CSS eingebunden wurde. Auf der Startseite wird der Nutzer willkommen geheissen und erhält die Möglichkeit über die Felder **Registration**, **Gruppenübersicht oder Spielplan** zu navigieren.

Auf den restlichen Seiten wurde auf eine einfache Navigation am oberen Rand geachtet. Zudem wurden alle Seiten entsprechend der Startseite aufgesetzt, damit die verschiedenen Programmier-Teams mit dem gleichen Grundgerüst starten konnten.

Design

Das Design wurde anhand des Hintergrundbildes gestaltet. Die Primärfarbe ist goldgelb, Sekundärfarben sind weiss und der Text ist stets in anthrazit und schwarz gehalten. Es wurde versucht ein möglichst übersichtliches und modernes Design zu kreieren. Dies nicht zuletzt, damit sich Teams leicht durch die Seiten navigieren und alle Aktivitäten in Zusammenhang mit den Gruppenspielen tätigen können.

Name

Zunächst wurden mehrere Namen in der Gruppe zusammengetragen. Eine Abstimmung im Plenum lieferte anschliessend **“Tournamently”** als Name für die Gruppenspiele.

Logo

Das Logo wurde selbst durch Grafikprogramme erstellt und an die Primärfarbe goldgelb angepasst. Das Logo wurde anschliessend auf den Server geladen und in die Webseite eingebunden.



Seite Registration

Die Seite Registrierung wird mit einem leeren Input-Feld dargestellt, welcher als Eingabefeld für die Teamnamen dient. Die Teams können sich auf dieser Seite registrieren. Die eingetragenen Teamnamen erhalten beim abspeichern in der Datenbank eine Id. Dies ist wichtig, damit zu einem späteren Zeitpunkt gespielte Resultate mit der jeweiligen Id abgespeichert werden können. Unterhalb des Input-Feldes werden bereits eingetragene Teams abgebildet.

Grundgerüst HTML

Für das Grundgerüst wurde die Datei registrieren.html erstellt. Es wurde zuerst ein Eingabebereich eingerichtet und ein Bestätigungsbutton eingefügt. Die Zeilen 39 bis 47 bestehen aus einem div-Element der Klasse "container". Im "container" befindet sich die Überschrift "Neues Team registrieren". Darunter befindet sich ein div-Element mit der id "groupnames". Die Datei ist mit den JavaScript-Dateien "js/registerteam.js" und "js/wer_ist_dabei.js" verknüpft.

Registerteam.js

Zeile 2-3 holt die bereits bestehenden Teamnamen mittels Ajax aus der Datenbank und zeigt sie dem User an. Der Code in Zeile 9 bis 15 berechnet wie viele Teams in einer Gruppe vorhanden sind und prüft ob bereits vier Teams in einer Gruppe vorhanden sind, was mit Hilfe einer While Schleife gemacht wird. Sind noch keine vier Teams in einer Gruppe, wird die Zahl in team_id gespeichert. Damit neue Teams in die Datenbank eingetragen werden können, wurde Zeile 21 bis 31 erstellt. In Zeile 21 wird die Funktion *teamssubmit(team_id)* definiert. In Zeile 25 wird durch eine IF Funktion die Bedingung festgelegt, dass die Registrierung erst erfolgen kann, wenn ein Zeichen im Eingabefeld steht.

Zeile 29-31 leert bei erfolgreicher Eingabe das Eingabefeld, somit ist das Eingabefeld wieder leer und bereit für eine neue Eingabe. Zeile 34 sorgt mittels Ajax Abfragung für eine Aktualisierung der bestehenden Team Eingaben. Wenn das klappt, dann wird durch die



Funktion von Zeile 6-16 eine *group_id* erstellt und berechnet in welcher Gruppe das Team eingetragen werden soll.

Die Zeile 43-44 erstellt ein Pop-Up mit der Mitteilung das die Eingabe erfolgreich war.

Die Zeile 48-50 meldet dem User, falls es Probleme mit Ajax gab, dass keine Verbindung zum Server hergestellt werden konnte.

wer_ist_dabei.js

Um die Daten *teamnames* aus der Datenbank zu ziehen, wurde die Funktion *receivedteamnames* erstellt (Zeile 6). Die Daten werden mittels `JSON.parse()` in der Variable *data_arr* als JavaScript Objekt gespeichert (Zeile 8). In einem nächsten Schritt wurde innerhalb einer neuen Variable *content* ein leerer String erstellt (Zeile 10). Falls noch kein Team in *data_arr* eingetragen ist, wird der User informiert (Zeile 18). Falls bereits Teams registriert sind (Zeile 12), wird mittels einer Schleife jedes Feld der *data_arr* durchlaufen und in die variable *content* eingefügt (Zeile 13-14). In einem nächsten Schritt wird die ID *groupnames* im HTML-Code anvisiert und die Variable *content* darin eingefügt (Zeile 22). Wenn keine Verbindung zum Server hergestellt werden kann, wird der User darüber mittels einer Alert-Funktion informiert (Zeile 26- 27).



Seite Gruppenübersicht

Die Seite Gruppenübersicht zeigt die Gruppen- und Teamaufstellung. Innerhalb der Gruppen befinden sich maximal vier Teams, die gegeneinander spielen. Die Endergebnisse werden jeweils unter jedem Team angezeigt.

Grundgerüst HTML

Als erstes wurde die Datei `gruppen.html` erstellt. In einem ersten Schritt wurde das Grundgerüst für diese Seite gecodet.

Die Zeile 39 bis 46 besteht aus einem `div`-Element der Klasse `“container”`. Im `“container”` befindet sich die Überschrift `“Gruppenübersicht”` und darunter ein `div`-Element mit der `id` `“gruppen_content”`. Die Datei ist mit der JavaScript-Datei `“js/gruppen_data.js”` verknüpft.

Verknüpfung mit `“js/gruppen_data.js”`

Der Code in Zeile 1 bis 3 greift auf die Datenbank `“games.php”` (wurde von Herr Hauser erstellt) zu und verknüpft sie mittels Ajax mit der Datei `js/gruppen_data.js`. Der Code in Zeile 7 bis 15 legt fest, dass die erhaltenen Teamnamen in `gruppen_id` und anschliessend in `data_arr` gespeichert werden. Duplikate werden in der Zeile 18 gelöscht. Der Code auf Zeile 21 bis 42 kontrolliert, ob die Gruppen-Arrays leer oder nicht leer sind. Sind sie nicht leer, werden die Teams in Gruppen aufgeteilt. Alle Teams beginnen mit der Punktzahl 0. Wenn die `gruppen_id` mit der `gruppen_id` in `data_arr` übereinstimmt, dann erhalten sie die Punkte vom eigenen Team. In den Zeilen 44 bis 51 werden die Teams absteigend nach Punktzahl sortiert. Die Zeilen 54 bis 59 regeln das Beenden eines Spiels und die Ernennung eines Gruppensiegers. Falls ein Gruppen-Array leer ist, besteht die Möglichkeit, neue Teams zu registrieren (Verlinkung zu `registrieren.html`) (Zeile 61-64). Zeile 66 sorgt dafür, dass alle Daten auf dem aktuellen Stand sind und diese der Datei `gruppen_content` weitergegeben werden. Wenn ein neues Team registriert werden will, dann wird der User zur Datei



registrieren.html weitergeleitet (Zeile 70-72). In der Zeile 75 werden alle Punktestände gespeichert, damit diese in weiteren Funktionen eingesetzt werden können. Die Funktion *who_win* (Zeile 77-124) regelt, dass die Daten von der Datenbank *games.php* geholt und in *data_arr* gespeichert werden. Für jedes Team wird zunächst die *team_id* in 'Punktestand' gespeichert. Als Basis hat jedes Team 0 Punkte. Die erhaltenen Punkte werden daraufhin addiert. Zeile 94 regelt in Form einer Schleife, dass das Gewinnerteam drei Punkte erhält (sofern ein Gewinnerteam vorhanden ist) und das Verliererteam 0 Punkte. Bei Gleichstand erhält jedes Team einen Punkt (Zeile 111-121). Der Code auf Zeile 122 löst einen Ajax Call zu *teamingroups* aus. Wenn die Datenbank *teamingroups.php* erreichbar ist, dann ruft er die Funktion *receivedteamnames* (Zeile 7-67) auf. Falls keine Verbindung zur Datenbank besteht, wird der Alert "keine Verbindung zum Server" ausgegeben (Zeile 132-134). Falls das Gruppenspiel beendet werden will, wird auf die Datei *beendet.html* verwiesen, wo die Sieger der Gruppen bekannt gemacht werden. Zeile 136 legt fest, dass die Funktionen init ausgeführt wird, sobald Window vollständig geladen ist.



Seite Spielplan

Die Seite Spielplan zeigt an, welche Teams gegeneinander spielen. Jedes Team kann seine Ergebnisse des Spiels eintragen. Nachdem ein Spiel beendet ist, werden die eingegebenen Ergebnisse bewertet und die Punktzahl erscheint auf der Seite Gruppenübersicht.

Grundgerüst HTML

Für das Grundgerüst wurde die Datei Spielplan.html erstellt, wo zu Beginn die Tabelle erstellt wurde. Die Tabelle wurde anfänglich statisch erstellt und diente als Platzhalter für den jetzigen JS-Code.

Die Zeile 39 bis 47 besteht aus einem div-Element der Klasse "container". Im "container" befindet sich die Überschrift "Spielplan" und darunter ein div-Element mit der id "spielplan". Die Datei ist mit der JavaScript-Datei "js/spielplan_data.js" verknüpft.

spielplan_data.js

Mittels der Funktion *init* wird auf *games.php* in der Datenbank zugegriffen und die eingetragenen Punkte herausgezogen. (Zeile 1-3) In einem weiteren Schritt wurde die Variable *punkte_arr* erstellt (Zeile 7). Um die Daten der Datei *games.php* aus der Datenbank zu ziehen, wurde die Funktion *receivedpunkte* erstellt (Zeile 9). Unter dem Parameter *tore* wird der JSON-formatierte Text, der eingelesen wurde, referenziert. (Zeile 10) Mittels *JSON.parse()* wurden die Daten in ein JavaScript-Objekt umgewandelt und in die erstellte Variable *punkte_arr* gespeichert (Zeile 11). Gleichzeitig werden aus der Datei *teamingroups.php* die Daten zu *teamnamen* und *gruppen_id* gezogen. Ist dieser Abruf erfolgreich, wird die Funktion *receivedteamnames* (Zeile 16-25) gestartet. Andernfalls wird der User mittels einer Alert-Funktion informiert, dass keine Verbindung zum Server hergestellt werden konnte (Zeile 13). Unter dem Parameter *data* wird der JSON-formatierte Text, der eingelesen wurde, referenziert. (Zeile 16) In einem nächsten Schritt wurde innerhalb einer neuen Variable *content* ein leerer String erstellt (Zeile 18). Mittels *JSON* wurden die Daten in ein JavaScript-Objekt umgewandelt und in die erstellte Variable *data_arr* gespeichert (Zeile 20). Die neu erstellte Variable *gruppen* enthält ein leeres Array (Zeile 22). Mittels einer Schleife wird durch die Variable *data_arr* iteriert und die *gruppen_id* mittels der *push*-Methode am Ende des Arrays *gruppen* hinzugefügt. (Zeile 24)



Die if-Schleife in Zeile 28 überprüft, ob der Array *gruppen* mehr als eine Gruppen-ID enthält. Falls bereits Gruppen existieren und Duplikate bestehen, werden diese aus dem Array *gruppen* gelöscht. Mit der neu erstellten Variable *resultat_existiert* wird dem Boolean-Wert *false* definiert, da davon ausgegangen wird, dass am Anfang noch keine Resultate vorhanden sind (Zeile 31). Der Variable *counter* wird der Wert 0 zugewiesen (Zeile 33). Danach wird für jede vorhandene Gruppe mittels der Funktion *parseInt* der String in einen Int umgewandelt, um beispielsweise Gruppe 0 im html als Gruppe 1 abzubilden (Zeile 35 - 44). Es wird in *data_arr* nach jedem Team gesucht, das die selbe *gruppen_id* besitzt. (Zeile 46 & 47). Nach der ersten Suche wird erneut nach diesen Teams gesucht, da es zwei Teams für ein Spiel benötigt (Zeile 49 & 50). Erneut wird die Variable *resultat_existiert* eingebunden, da zu diesem Zeitpunkt noch keine Ergebnisse zu den Spielen bekannt sind (Zeile 51). Mittels der If-Schleife in den Zeilen 53 bis 56 wird gewährleistet, dass alle Teams einer Gruppe gegeneinander spielen. Mit der For-Funktion in Zeile 59 wird die Datenbank *games.php* durchgegangen. Mittels der If-Schleife wird geprüft, ob es Übereinstimmungen der *team_id* mit den *team_id* in *games.php* gibt. Ist dies der Fall bestehen bereits Resultate zu den Spielen dieser Teams (Zeile 62-68). Innerhalb der If-Schleife besteht die Möglichkeit das existierende Resultat zu ändern (Zeile 66). Die Variable *resultat_existiert* wird auf "true" gesetzt, da nun Resultate zu den einzelnen Spielen existieren. Falls noch kein Resultat zu einem Spiel vorhanden ist, wird der Platzhalter "kein Eintrag" mittels einer If-Schleife eingefügt (Zeile 72-79). In Zeile 81 wird der Counter um 1 erhöht, um für jede Begegnung eine eindeutige Identifikation-Zahl zu erhalten. Zeile 82 schliesst den <tr> Tag für jede Begegnung. Zeile 89 bewirkt, dass jede Gruppe in einer einzelnen Tabelle dargestellt wird.

Zum Ende der Gruppenspiele wird die Frage gestellt, ob die Gruppenspiele beendet sind und der Gruppensieger benannt werden soll. Dies kann mittels den "Yes, do it!"-Button bestätigt werden (Zeile 92-96).

Mit der else if-Schleife aus Zeile 99 wird der Fall bearbeitet, wenn nur ein Team in einer Gruppe registriert ist. Ist dies der Fall erscheint die Meldung: Es müssen mindestens zwei Teams registriert sein!. Anschliessend kann mittels einem Button ein zweites Team registriert werden (Zeile 99-102).

Mit der else-Schleife aus Zeile 104 wird der Fall abgebildet, wenn gar kein Team in einer Gruppe registriert wurde. Es wird darauf hingewiesen, dass noch kein Team registriert wurde



und erneut kann mittels einem Button ein Team registriert werden (Zeile 104-107). Zeile 109 leitet die gesammelten Informationen #spielplan in spielplan.html weiter. Die Zeilen 112-115 zeigt die Funktion "register", welche den User zur HTML-Seite registrieren.html weiterleitet.

Mit der Funktion "eintragen_tore" (Zeilen 118-146) können die Tore der jeweiligen Begegnungen gespeichert werden. In die Variable team1 werden mit dem Befehl document.getElementsByName die Datenfelder von punkte_team1, die sowohl den Namen des Teams als auch das eingetragene Ergebnis beinhalten, gesucht. Mit einer For-Schleife wird jedes Element iteriert und eine neue Variable "eintrag" mit den Inhalten Team-Id und Ergebnis erstellt. Danach wird überprüft, ob die Variable "eintrag" den gleichen Wert wie die Variable "counter" hat. Auf diese Weise kann sichergestellt werden, dass der angeklickte Button "eintragen" zur geklickten Zeile passt. Ist dies der Fall kann die Schleife abgebrochen werden, da die geklickte Zeile gefunden wurde. Die jeweilige Team-Id und die Punkte werden in die zwei Variablen "punkte_team1" und "team1_id" gespeichert. Das gleiche Vorgehen wird für team_2 angewendet. In den Zeilen (Zeilen 147 - 149) werden die Objekte "team1_id", "team2_id", "punkte_team1", "punkte_team2" an die Datenbank (resultadd.php) gesendet, um die Anzahl an Toren der jeweiligen Begegnung zu speichern. War die Übertragung erfolgreich, wird die Funktion eintrag_erfolgreich aufgerufen, wobei die Information, ob ein Eintrag bereits existierte oder neu verfasst wurde, mitgeliefert wird.

In den Zeilen 153 - 164 wird mit der Funktion eintrag_erfolgreich im Fall, dass es sich um einen bereits bestehenden Eintrag gehalten hat, der verändert wurde, ein Alert mit der Nachricht "Eintrag neu gespeichert!" ausgegeben. Handelt es sich um einen neuen Eintrag, wird der Alert "Erfolgreich eingetragen" ausgelöst. Am Schluss wird die Funktion init aufgerufen.

Mittels der Funktion *beenden* wird auf die Datei *beendet.html* verwiesen, wo die Sieger der Gruppen bekannt gegeben werden. Hierfür wurde das *window.location*-Objekt eingesetzt, um zur angegebenen URL zu gelangen (Zeile 167-169).

Die Zeilen 172 und 173 definieren, was im Falle eines Ajax Fehler passiert. Tritt ein solcher Fehler auf wird mittels einem Alert informiert, dass keine Verbindung zum Server besteht.

Auf der Zeile 176 wurde mittels *window.onload=init* definiert, dass die Funktion init ausgeführt wird, sobald das Fenster vollständig geladen ist.



beendet.html

In den Zeilen 22 bis 40 befindet sich ein div-Element der Klasse “container”. Im “container” befindet sich die Überschrift “Die Sieger” und darunter zwei Paragraphen mit einer Beschreibung der Punktevergabe. Danach folgt ein weiteres div-Element mit der Id “sieger_content”, welches als Platzhalter für den Inhalt aus der verknüpften JavaScript-Datei “wer_hat_gewonnen.js” fungiert. Es folgt eine weitere Überschrift “Wie weiter?” und drei Buttons, welche verschiedene Optionen aufzeigen. Die jeweiligen Buttons sind mit dem Event onclick erweitert, welche wiederum die entsprechenden Funktionen aus der JavaScript-Datei “wer_hat_gewonnen.js” aufruft.

wer_hat_gewonnen.js

In dieser Datei werden bei Erfolg die jeweiligen Punkte mit *who_win* berechnet. Die Funktion *receivedteamnames* in der Zeile 7 (wird bei Erfolg in Zeile 177 aufgerufen) des Codes sorgt bei aufruf dafür, dass die Teamnamen erhalten werden. In den Zeilen 9 bis 11 wird die *gruppen_id* in *data_arr* gespeichert und in Zeile 12 des Codes werden alle Teams in deren entsprechende *gruppen_id* gespeichert. Duplikate werden in Zeile 18 gelöscht. Sofern der Gruppen Array nicht leer ist, wird in den Zeilen 23 bis 25 dafür gesorgt, dass die Teams in deren jeweilige Gruppe eingeteilt werden. Zunächst werden in Zeile 26 bis 31 die Platzierungen sowie die Punkte jedes Teams resettet. Falls die *gruppen_id* des teams in *data_arr* übereinstimmt, dann werden die Punkte vom aktuellen Team durch die Zeile 33 bis 41 übernommen. In Zeile 43 wird festgelegt, dass die Punkte sowie der Teamname im Array Platzierung hinterlegt wird. Zunächst wird der Array Platzierung absteigend nach Punkte sortiert (Zeile 47). Der Gewinner hat mindestens einen Punkt (Zeile 48-55). Wenn alle Teams die gleichen Punkte haben, dann werden die Teams nach Toren sortiert (Zeile 56-57). Wenn die ersten drei Teams Gleichstand haben (bei total 4 Teams), dann wird durch die Funktion *.pop* das letzte Element des Arrays entfernt (Zeile 59-62). Bei drei gleichen Ergebnissen bei total drei Teams, werden wiederum die Tore sortiert (Zeile 65-67) und wenn die ersten zwei Teams Gleichstand haben und total drei Teams vorhanden sind, dann wird wie zuvor durch *.pop* das letzte Element des Arrays entfernt (Zeile 70-73). Die Zeilen 81 bis 86 kommen zum Einsatz, wenn ein eindeutiger Gewinner besteht: *ein_gewinner = true*.



Wenn keine Resultate eingetragen wurden, dann hat der Gewinner 0 Punkte und der Text “Es wurde nicht gespielt!” wird angezeigt (Zeile 91-93). Wenn ein Gewinner *true* ist, dann wird durch den Code in den Zeilen 95 bis 99 festgelegt, dass der Text “Sieger”, die Platzierung sowie die Punkte ausgegeben werden. In den Zeilen 101 bis 106 wird die Anzahl Tore neu sortiert und der Sieger ausgegeben. Am Ende wird durch den Code in Zeile 114 alles in *content* an *#sieger_content* in der HTML-Datei *beendet.html* weitergegeben.

Funktion *who_win*: In der Zeile 118 wird der Punktestand Array für den globalen Gebrauch angelegt. Zunächst werden die Daten von der Datei *games.php* geholt (Ajax-Call Zeile 1-4) und in *data_arr* gespeichert (Zeile 120-122). Für alle Teams wird die Team id in Punktestand gespeichert. Und in Zeile 129 wird der Array mit der einzigartigen *team_id* geholt. Mit dem Code *for(var i in data_arr)* in Zeile 136 werden 0 Punkte und 0 Tore zu jedem Team hinzugefügt. Die gewonnenen Punkte jedes Teams werden dann zu den 0 Punkten addiert (Zeile 135). Wenn ein Team gewonnen hat, dann erhält es 3 Punkte (Zeile 137-142). Es wurden Team1 sowie Team2 festgelegt, wobei beide Teams jeweils 3 Punkte erhalten, wenn sie gewonnen haben (zusätzliche Zeilen 146-150). Wenn beide Teams den gleichen Punktestand und somit Gleichstand haben, dann erhalten sie je einen Punkt (Zeile 154-162). Die Anzahl Tore, die bei einem Punktegleichstand für die Sieger-Ernennung benötigt werden, werden in den Zeilen 166 bis 174 gespeichert. Sind die Punkte sowie Anzahl Tore für jedes Team gespeichert, dann wird ein Ajax-Call zu *teamingroups.php* gemacht (Siehe Funktion *receivedteamnames*).

Die Funktion *reset()* ruft *dbwipe.php* oder wenn Preset gewünscht ist zusätzlich *dbpreset.php* auf. Dadurch wird die Datenbank des Gruppenspiels zurückgesetzt und alle Einträge gelöscht. Durch den Preset werden vordefinierte Daten eingesetzt und das Gruppenspiel kann mit bereits eingetragenen Teams neu gestartet werden. Durch *window.location* in Zeile 188 des Codes wird zur HTML-Seite *index.html* verwiesen. Die Funktion *go_home* in Zeile 195 verweist ebenfalls zur HTML-Seite *index.html*. Zeile 199 bis 200 legt fest, dass der Alert “keine Verbindung zum Server” ausgegeben wird, wenn ein Ajax Fehler besteht. Die letzte Zeile 203 definiert, dass die Funktion *init* ausgeführt wird, sobald das Fenster vollständig geladen ist.