## pmh

## Hilfekarte 1

## Funktion on Challenge

pmhs Nürtingen Kanalstraße 29 72622 Nürtingen bar@pmhs.de

Die Funktion "onChallenge" wird im folgenden Flussdiagramm dargestellt. Daneben findet sich ihr Code. Zunächst implementiert sie nur den Aufbau einer Herausforderung zwischen zwei Spielern für das Stein-Schere-Papier. Es sind aber schon Grundlagen für die Erweiterung mit

anderen Spielen geschaffen. Start "onChallenge" mit Parameter "data Parsen der "data" vor JSON zu Object und speichern in "turn" Deklaration p1 und p2 mit Initialisierung auf "null" "reply" als Objekt mit Attributen "from" und 'text" deklarieren und initialisieren Rückgabe von "getChallenge(player1 player2, game)" in Variable "chal" speichern. ür alle Elemente der "lobby turn.player1 = player.name des lobb p1 = lobby-Element Flements? turn.player2 = player.name des lobb p2 = lobby-Element neues RPSDuel Objekt mit p1 und p2 erzeugen und in "rps" chal = nul p1 nicht "nul speichern und 2 nicht "nul Zug mit makeTurn(player1 über "chal" im "game turn)" über "rps 'makeTurn" aufrufen Objekt Nein und Zug und Spielername übergeben Objekt mit "player1" Meldung "reply" an p1 player2" und "rps" zi wenn p1 nicht "null" challenges'

```
function onChallenge(data) {
 let turn = JSON.parse(data);
 let p1, p2 = null;
 let reply = { from: 'Server', text: "" }
 let chal = getChallenge(turn.player1, turn.player2, turn.game)
 for (let i = 0; i < lobby.length; i++) {</pre>
    if (turn.player1 === lobby[i].player.name) {
     p1 = lobby[i];
    } else if (turn.player2 === lobby[i].player.name) {
     p2 = lobby[i]:
  }
 if (p1 != null && p2 != null) {
    if (!chal) {
     let rps = new RPSDuel(p1, p2);
     rps.makeTurn(turn.player1, turn.turn);
     challenges.push({ player1: turn.player1,
       player2: turn.player2, game: rps });
     return 0;
    } else {
     chal.game.makeTurn(turn.player1, turn.turn);
     challenges.splice(challenges.indexOf(chal), 1);
     return 0;
  } else {
    reply.text = "Spieler nicht vorhanden. Herausforderung ("
    + turn.game + ":"
    + turn.player1 + " vs. "
    + turn.player2 + ") gescheitert.";
    challenges.splice(challenges.indexOf(chal), 1);
  if (p1 != null) {
    p1.socket.emit('msg', JSON.stringify(reply));
  if (p2 != null) {
    p2.socket.emit('msg', JSON.stringify(reply));
```

Versuche den Code so gut es geht zu verstehen und zu kommentieren.

chal" mit "splice" aus

challenges entfernen

Mit return 0 beenden

Meldung "reply" an p2

wenn p2 nicht "null"

hinzufügen

Mit return 0 beenden

getChallenge()

bar - 17.02.2020 Seite 1 von 2

Spielernamen der beiden Spieler und die Art des Spiels werden als String übergeben

## Durchsuchen aller bestehenden challenges

```
function getChallenge(p1, p2, game) {
  for (let i = 0; i < challenges.length; i++) {
    if (challenges[i].game.gametype === game) {
      if (challenges[i].player1 === p1 ||
            challenges[i].player2 === p2 ||
            challenges[i].player2 === p2 ||
            challenges[i].player1 === p2) {
            return challenges[i];
      }
      }
      Prüfungen ob p1 bereits in einer Challenge gegen p2
      dieses Spiel spielt
            Oder (||)
      p2 gegen p1
      return null;
}</pre>
```

Gefundene Challenge mit diesem Spiel zwischen p1 und p2 zurück geben

Wenn nichts gefunden wird, dann wird "null" zurück gegeben

bar - 17.02.2020 Seite 2 von 2