

Winterthur Yard: Abschlussbericht Projekt

Maja Fritschi, Raphael Spörri, Florian Bosshard

Contents

1	Applikation	2
2	Projektteam	2
3	Funktionen	2
3.1	Login	2
3.2	Kartenanzeige	2
3.3	Catch Me	3
4	Aufbau der Applikation	4
4.1	JavaScript	4
4.2	Kommunikation zum Server	5
4.3	Datenbank	5
5	Erweiterungen	5
6	Fazit	6
6.1	JQuery Mobile	6
6.2	LeafletJS	6
6.3	Slim	6
6.4	RevealJS	6

1 Applikation

Die aktuelle Applikation kann unter <http://yard.prusik.ch> angeschaut werden.

Die Präsentation für den WBE-Unterricht befindet sich unter <http://yard.prusik.ch/presentation/>

Der Code befindet sich in einem Repository auf Github: <https://github.com/florianbosshard/Yard>

2 Projektteam

Das Projekt wurde von folgenden Personen entwickelt:

- Maja Fritschi (fritsmaj)
- Raphael Spörri (sporr0)
- Florian Bosshard (bosshflo)

3 Funktionen

3.1 Login

Der Spieler kann sich auf der Startseite mit einem Usernamen einloggen. Geprüft wird nur, ob er etwas eingibt. Kein Username ist nicht erlaubt. Ausserdem kann man auch nicht auf eine Subseite zugreifen, wenn man sich nicht eingeloggt hat.

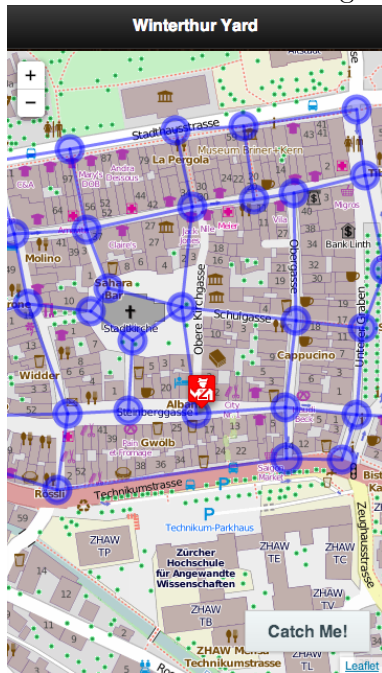


3.2 Kartenanzeige

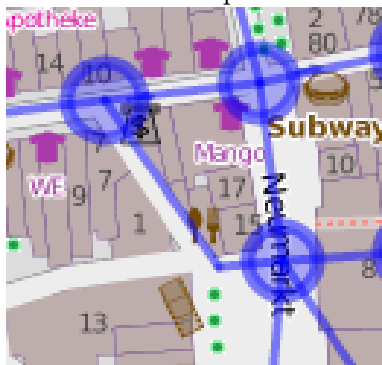
Hat der Spieler sich erfolgreich eingeloggt, sieht er einen Kartenausschnitt der Winterthurer Altstadt. Seine eigene Position ist mit einem roten Icon markiert.

Auf der Karte liegt ein Graph mit verschiedenen Knoten. MisterX bewegt sich jede Minute einen Schritt von einem Knoten zum nächsten.

Die Karte wird live bezogen von Open Street Map¹. Für die Darstellung wurde das Framework LeafletJs² eingesetzt.



Da die Strassen nicht gerade sind, haben wir in der Datenbank die Möglichkeit einer Strecke Zwischenpunkte hinzuzufügen.



3.3 Catch Me

Befindet sich der Spieler auf einem der eingezeichneten Koordinaten-Punkte, kann er den Button "Catch Me" drücken.

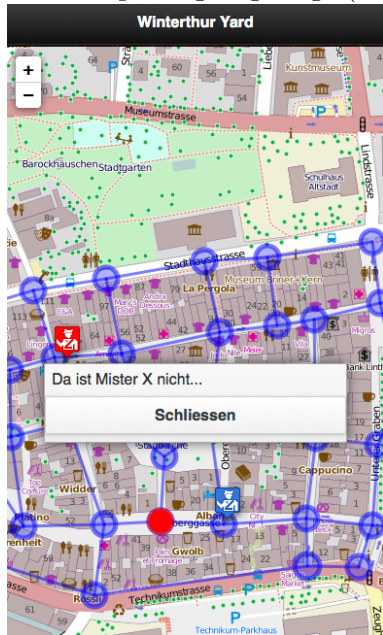
Solange sich der Spieler nicht auf einem gültigen Spielfeld (innerhalb eines Kreises auf der Karte) befindet, erhält er eine entsprechende Meldung.

Ist der Spieler auf einem gültigen Spielfeld, wird ihm die Position von MisterX vor seinem

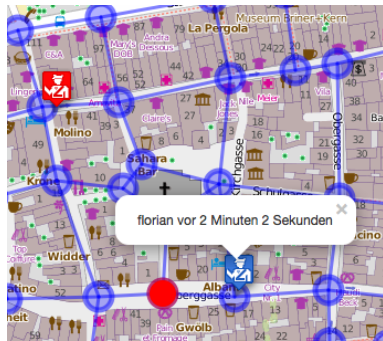
¹<http://www.openstreetmap.org>

²<http://leafletjs.com>

letzten Spielzug angezeigt (roter Kreis).



Zudem sieht man mit blauen Symbolen, wo sich andere Spieler aufhalten.



Drückt man zweimal auf “Catch Me“ auf dem selben Feld, wird man darauf hingewiesen, dass dies nicht erlaubt ist.

Wenn der Spieler sich auf dem selben Feld befindet wie MisterX erhält er eine Information, dass er MisterX gefangen hat. Das Spiel ist damit beendet.

4 Aufbau der Applikation

4.1 JavaScript

Das JavaScript wurde alles unter yard gekapselt. Im Nachhinein gesehen, wäre es sinnvoll gewesen mehrere Objekte (z.B. Benutzer, Karte) zu erstellen.

```

var yard = {
  map : null,
  ownPositionMarker : null,
  misterXCircle: null,
  userPosition: null,
  otherPlayers: new Array(),
  initMap : function() {
    // ...
  }
}

$(document).ready(function() {

  yard.initMap();
  // ...
});

```

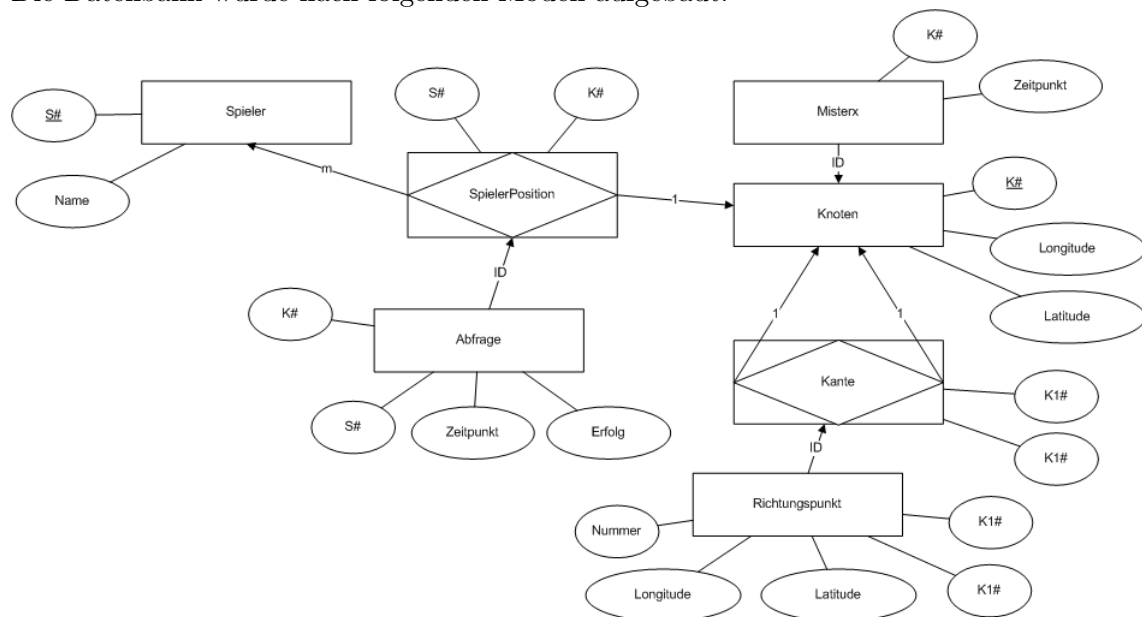
4.2 Kommunikation zum Server

Die Kommunikation zum Server funktionierte über eine REST-Schnittstelle mit JSON. So konnten sehr einfach Daten übertragen werden.

Die ganze Logik, die bestimmt, ob eine Person an einem Knoten ist, geschieht auf dem Server. Bei einem Catch Me schickt JavaScript lediglich die aktuelle Position an den Server. Dieser gibt danach eine entsprechende Antwort.

4.3 Datenbank

Die Datenbank wurde nach folgenden Modell aufgebaut:



5 Erweiterungen

Die Projektziele konnten erfüllt werden - das Spiel ist lauffähig.

Offene Punkte und Erweiterungen

- Schlussendlich soll MisterX die Koordinaten eines Geocaches kennen. Zuerst muss aber ein geeignetes Versteck für den Cache in Altstadtnähe gefunden werden.
- Über einen Button soll die Karte um den eigenen Standort zentriert werden. Im Moment ist das Zentrum immer bei der Stadtkirche Winterthur
- Über einen Button soll die Kartenanzeige zum roten Punkt (Position von MisterX vor dem letzten Zug) springen. Dies nur, wenn dies überhaupt angezeigt wird. Ansonsten ist der Button disabled.

6 Fazit

Es hat sehr viel Spass gemacht die App zu erstellen. Wir konnten diverse für uns neue Tools ausprobieren:

6.1 JQuery Mobile

JQuery Mobile hat uns nicht vollständig überzeugt. Wir hatten grosse Schwierigkeiten mit dem Styling und mit der Positionierung der Karte. Deshalb haben wir das Layout, dass wir am Anfang bestimmt haben auch nicht vollständig umgesetzt. Schön ist, dass man Controls out of the box einsetzen konnte - so zum Beispiel die Popup-Mitteilungen.

6.2 LeafletJS

LeafletJS hat uns sehr überzeugt dank einer einfachen und gut dokumentierten API. So hatten wir keinerlei Probleme die Karte mobile-tauglich darzustellen.

6.3 Slim

Slim hat uns sehr unterstützt sauberer Webservices anzubieten. Bei unserem Billig-Hoster haben wir etwas mit mod rewrite gekämpft, dass bei Slim standardmässig verwendet wird.

6.4 RevealJS

RevealJS hat uns sehr gefallen. Es ist mit sehr einfachen Mitteln möglich eine ansprechende Präsentation zu machen. Da diese auf HTML basiert, können alle Vorteile der Sourcekontrolle auch dort verwendet werden (History, Kollaboratives Arbeiten). Aus dem selben Grund haben wir unsere Dokumentation auch mit $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ erstellt.