JavaScript und Softwaretechniken

Model-View-Controller (MVC) in einer HTML / CSS / JavaScript Anwendung

# Hintergrund

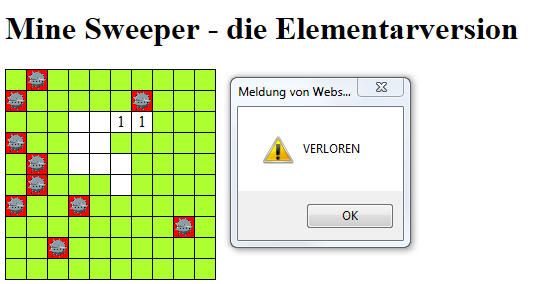
# Konventionelles Script

Im ersten Schritt werden wir mit wenig HTML und CSS sowie elementarem JavaScript Programmcode eine einfache, aber vollfunktionsfähige Version des Spiels *MineSweeper* erstellen. Im JavaScript verwenden wir dabei lediglich Variablen, Funktionen, Schleifen, einfache Objekte, Felder und Zugriffe auf das HTML DOM.

Die erste Version des Spiels wird bereits folgende Funktionen bieten:

* Die Parameter des Spielfelds wie die Größe und die Anzahl der versteckten Minen wird über HTML (data-) Attribute festgelegt und nicht im Programmcode fixiert.
* Beim Aufbau des Spielfelds werden die Minen zufällig versteckt. Als Hilfe für den Spieler wird automatisch ein Spielfeld aufgedeckt, unter dem sich keine Mine versteckt und in dessen unmittelbarer Nachbarschaft sich auch keine Felder mit versteckten Minen befinden.
* Beim Aufdecken eines Feldes wird die Anzahl der versteckten Minen in der unmittelbaren (8 Felder) Umgebung angezeigt.
* Wird ein Feld mit Mine aufgedeckt, ist das Spiel verloren. Wird das letzte Feld ohne versteckte Mine aufgedeckt, ist das Spiel gewonnen.
* Als Hilfe vor versehentlichem Aufdecken kann der Spieler mit der rechten Maustaste ein Feld sperren, wenn er den Verdacht hat, dass sich darunter eine Mine befindet. Diese Sperre kann jederzeit wieder aufgehoben werden.
* Die Anzeige unterscheidet nicht aufgedeckte von aufgedeckten Feldern. In aufgedeckten Feldern steht immer die Anzahl der Felder mit versteckten Minen in der unmittelbaren Umgebung.
* Ist das Spiel zu Ende, so werden auf jeden Fall die versteckten Minen angezeigt.

Zur Umsetzung werden hier die Techniken und Kenntnisse verwendet, die wir auch im Projekt *Interaktive Web Seiten mit JavaScript* im Rahmen der Projekttage 2015 eingesetzt haben.



# Klassen, Objekte, Eigenschaften, Methoden

* Funktionen als Klassen
* Eigenschaften per Objektinstanz
* Instanz-Funktionen, prototype und this
* Klassen-Funktionen und -Eigenschaften
* Individuelle Eigenschaften und Funktionen
* Funktionen und Funktionsvariablen

# Model-View-Controller als Idee

* Modell (Model) ganz ohne Zugriff auf die Anzeige
* Anzeige (View) ohne Referenz (DOM Erweiterungen) auf das Modell
* Idee der Verbinder (Controller), hier als statische Klassen (viele andere Ansätze)
* Benachrichtigungen auch aus dem DOM heraus
* Erst einmal immer ein Tripel (Modell, Anzeige, Verbinder)

# Model-View-Controller als Basis

* Weitere Funktionalitäten
  + Größe des Spielfelds frei wählbar
  + Neustart
  + Automatisches Auffüllen von 0-Feldern
  + Spielstandanzeige
  + Bestenliste via localStorage
* Weiterhin strenge Trennung von Anzeige und Modell
  + Mehrere (Anzeige, Verbinder) Paare für ein Modell

# User Experience

* Bestenliste optisch aufbereitet
* …