JavaScript und Softwaretechniken

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

# Hintergrund

# Konventionelles Script

* Dimensionen aus HTML
* Minen verstecken
* Vorgabe 0-Feld
* Sperren
* Prüfen mit Gewinnen und Verlieren
* Keine Doppelprüfung
* Spielende erkennen

# 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
* …