

# **Informatik im Auto**

Koray Adigüzel, Raphael Hohl, Grégory Witmer  
Studiengang „Multimedia Produktion«, Major Web Application  
Dozent: Martin Vollenweider  
HTW Chur, Juli 2015

## Auftrag

Den Auftrag für dieses Projekt erhielten wir von Herrn Fredy Schwyter von der «Schweizer Informatik Gesellschaft» [sic!]. Als Präsident der neu gegründeten Fachgruppe «Multimedia» bat er unseren Major, in multimedialen Projekten den Einfluss und die Gegenwart von Informatik im Alltag aufzuzeigen.

## Idee

Aufgrund eines unserer Mitglieder entschieden wir uns bald für das Thema «Informatik im Auto». Dies auch, weil heutige Autos immer stärker von teils beeindruckenden Computern unterstützt werden und sich somit für die Lösung des Auftrags anbieten.

### 1. Schritt

Wir überlegten uns, welches Medium für die Übermittlung der Botschaft am geeignetsten wäre. Aufgrund der Tatsache, dass wir im Major Media Application das Thema *Videosteuerungen* behandelt hatten, sowie dem Aspekt des Storytellings, kamen wir zum Schluss, das Medium „Film“ würde sich hierfür am besten eignen. Unser Ziel war es, Wissen in interaktiver Form in Verbindung mit filmischem Erzählen, d.h. einer Story, zu transportieren.

### 2. Schritt

Im nächsten Schritt sammelten wir mithilfe eines Brainstorming zunächst alle möglichen Systeme, wobei wir über Kontakte eingangs erwähnten Mitglieds von Mercedes-Benz Schweiz für einen Tag einen Wagen zur Verfügung gestellt bekamen.

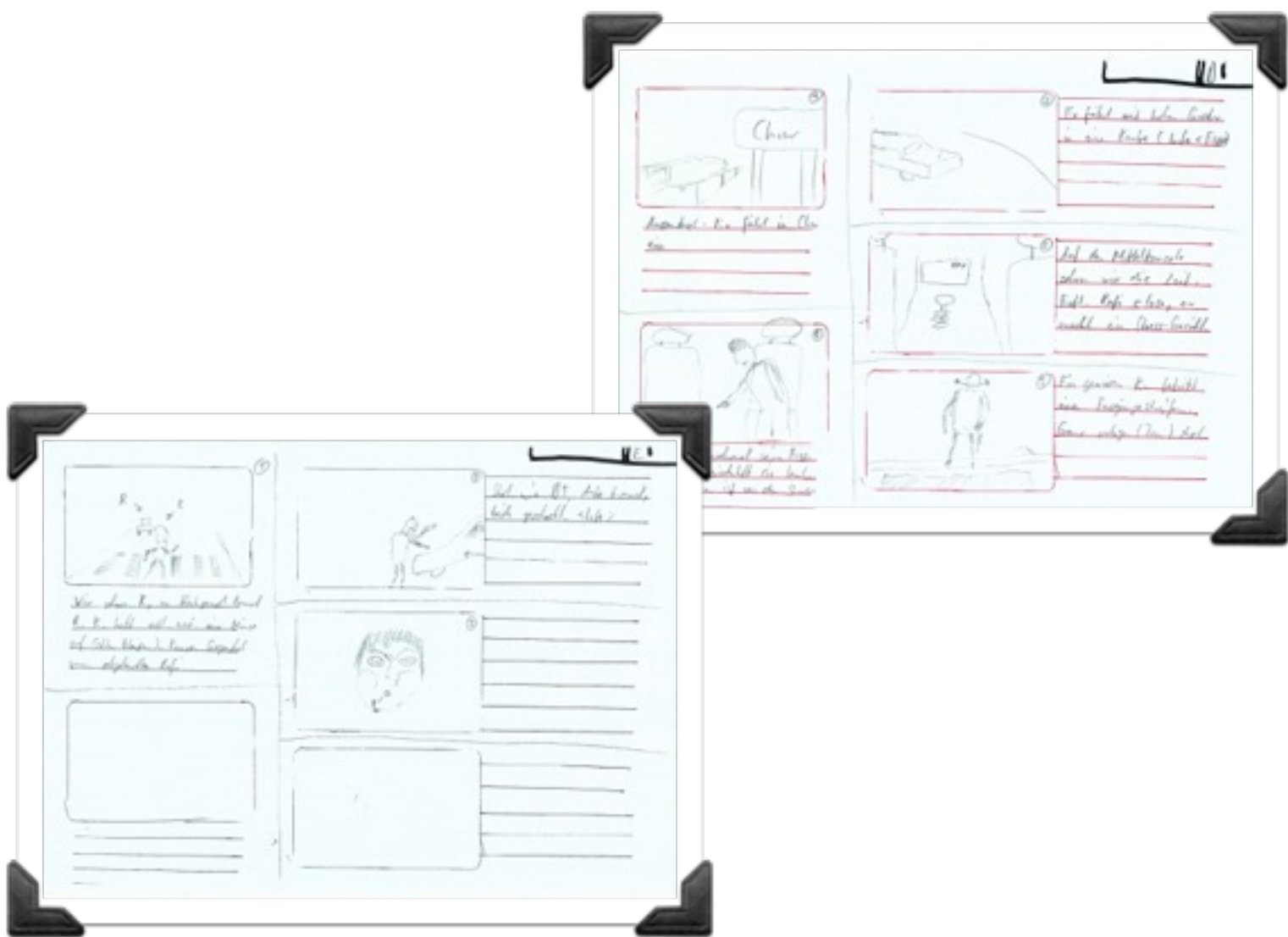
Beim Brainstorming konnten wir uns deshalb, nebst dem Fachwissen unseres Mitstreiters, auf dessen Wissen bezüglich der vorhandenen Technik verlassen. Von den gesammelten Ideen verwarfen wir einige wieder, da sie sich für eine Illustration diesen Umfangs nicht eigneten (zum Beispiel «ESP»).



### 3. Schritt

Nach der Auswahl der Technik, entwarfen wir eine Story, in welcher Möglichst viele der zuvor eruierten Technologien sinnvoll untergebracht werden konnten. Hierfür notierten wir stichwortartig unsere Ideen und formten daraus eine Storyline. In einem weiteren Schritt musste die Story zu Papier gebracht bzw. skizzenhaft visualisiert werden. Folglich kreierten wir ein Storyboard, welches die wichtigsten Key Visuals und eine kurze Beschreibung enthält.





#### 4. Schritt

Nachdem wir den Wagen bei Mercedes-Benz Schweiz in Zürich abgeholt hatten, filmten wir in einer (zuvor durchgeplanten) Fahrt nach Chur alle Szenen. Dabei war jeweils eine Person für das Fahren und Schauspielern, eine fürs Filmen sowie eine für die Sicherheit zuständig.

#### 5. Schritt

Die fünfte Etappe des Projekts bestand darin, die Aufnahmen auszuwerten, auszusortieren und zu einem fertigen Produkt zu vereinen. Zeitgleich zur Postproduktion entwickelten wir die Videosteuerung, die dem Film Interaktivität hinzugefügt. Des Weiteren verfassten wir Titel, Text und trafen eine Auswahl von Bildern zu den einzelnen technischen Erklärungen, die je nach Zeitpunkt als Dialog-Boxen über dem Video erscheinen.

# Technische Dokumentation

## «Informatik im Auto»

### Inhalt des Projektordners

- **\_notes** (Ordner): Von Dreamweaver erstellte Dateien. Eine \_notes-Datei befindet sich auch in jedem Unterordner.
- **Bilder** (Ordner): Enthält sämtliche im Projekt verwendete Bilder, das heisst die Bilder der Dialogboxen sowie das <video>-Poster.
- **images** (Ordner): Enthält zwei pngs, die von jQueryUI benutzt werden.
- **index.html** (HTML-Dokument): Die (einzige) Seite des Dokuments.
- **jquery-ui-1.11.4.custom** (Ordner): Enthält verschiedene CSS- und Javascriptdateien von jQuery, jQueryUI und dem custom theme für jQueryUI.
- **MyScripts** (Ordner): Enthält die Dateien «[Script-Informatik\\_Im\\_Auto.js](#)» (Hauptscript) sowie «[boxentexte.js](#)», in welcher Titel, Texte und Bildquellen der Dialogboxen in Arrays gespeichert sind.
- **MyStyles** (Ordner): Enthält die CSS-Datei des Projekts.
- **Videos** (Ordner): Enthält das Video des Projekts, «InformatikImAuto.mp4».

### Programmablauf

- Auf der index-Seite befindet sich mittig ein in der Größe definiertes **Videoelement**. Nachdem index.html geladen wurde, werden dem Video zwei eventhandler angefügt. Der eine ruft bei jeder Veränderung der aktuellen Wiedergabezeit («timeupdate») die Funktion **check()** auf. Der andere reagiert auf Beendigung des Videos und ruft **showCredits()** auf.
- Durch Klick auf den «**Start**»-Button wird das **Video** gestartet. In der finalen Implementierung könnte dieser Vorgang jedoch auf verschiedene Arten ausgelöst werden, zum Beispiel automatisch nach dem vollständigen Laden des Videos.
- Während das Video läuft, prüft **check()** regelmässig ob einer der Einträge im Array «**stoppZeiten**» zur momentanen Zeit passt. Ist dies der Fall, und die betroffene Zeit stimmt von ihrer Position im Array mit dem Fortschritt in der Geschichte überein («**Stand**», beginnt bei Null), wird das Video pausiert, die Funktion **goModal()** aufgerufen, und «**Stand**» um 1 erhöht. Ansonsten wird hier abgebrochen, das **Video** spielt normal weiter.
- **goModal()** wählt, abhängig vom Fortschritt «**Stand**», aus den entsprechenden **Arrays** den Titel, das **Beitragsbild** und den Text, fügt diese in das noch unsichtbare (.ui-helper-hidden) **div «dialog-message»** ein, und lässt dieses von jQueryUI als «**dialog**»-Element anzeigen. Ein Klick auf den «Weiter»-button des «**dialog**» lässt diesen wieder verschwinden und startet die Wiedergabe.
- Ist das Video beendet, wird die Funktion «**showCredits**» aufgerufen, welche das bis dato unsichtbare (.ui-helper-hidden) **div «credit-message»** als «**dialog**» erscheinen lässt. Ein Klick auf dessen «**Ende**»-button lässt es wieder verschwinden und führt einen erneuten **load()** des Videos durch, sodass dieses für ein nächstes Abspielen bereit ist und das «**poster**» anzeigt. Ausserdem wird der Fortschritt «**Stand**» wieder auf null gesetzt.

# Mögliche nächste Schritte

Für das Projekt «Informatik im Auto» kamen wir nach Diskussionen auf vier Weiterentwicklungen.

## Option 1

Im Moment ist die Grösse des Videos fest auf eine Angabe in Pixel kodiert. Dies hat den Nachteil, dass sich die Seite weder verschiedenen Wiedergabegeräten noch dem für den Moment nicht bestimmten Integration anpassen kann. Mobile-friendly-ness erachten wir jedoch als wichtigen Punkt.

## Option 2

Je nach Umgebung, in die das Projekt später eingebunden wird, macht ein Button zum Videostart eventuell weniger Sinn als zum Beispiel die load-gesteuerte Wiedergabe. Diese Entscheidung ist in Rücksprache mit dem Kunden (Herrn Schwyter) zu treffen. Ebenso die Entscheidung ob das Video bei Nichtaktivität des Users von alleine weiterlaufen sollte (Timer).

## Option 3

Da das Video an gewissen Stellen auf eine Eingabe des Nutzers wartet, könnte die Geschichte mit Entscheidungen erweitert werden. Statt also nur zu entscheiden wann das Video weiterlaufen soll, hätte der User die Möglichkeit, eine Auswahl, zum Beispiel zum Handeln des Protagonisten, zu treffen.

Bei dieser Option würden die Vorgaben dieses Projekts allerdings stark erweitert, das Vorhaben drohte schnell auszuufern.

## Option 4

Zuletzt könnte die von uns entwickelte Videosteuerung als Vorlage für eine Applikation dienen, mit der ähnliche Projekte umgesetzt werden können. Ein User würde dafür ein von ihm erstelltes Video hochladen und die Werte «stoppZeiten», «Titel», «Text» sowie «Bild» in einem Formular angeben. Diese Werte werden in einer Datenbank gespeichert und, wird das Video von einem User aufgerufen, auch von dort bezogen. Neben der schnellen und simplen Erstellung ähnlicher Projekte könnten so auch bestehende Artefakte bequem im Browser angepasst werden.