1. Dezember 2022

Robin Nater

GBS St.Gallen

Dokumentation ePortfolio

M152 - Multimedia-Inhalte in Webauftritt integrieren

Inhalt

[Entwicklung ePortfolio 2](#_Toc122641229)

[Eingesetzte spezielle Technologien 2](#_Toc122641230)

[Dynamische Logofarbe 2](#_Toc122641231)

[Abweichungen zum Konzept 3](#_Toc122641232)

[Design vom Range Input 3](#_Toc122641233)

[Farbstreifen als Hintergrund 3](#_Toc122641234)

[Quellen 4](#_Toc122641235)

[Bisherige Webseiten 4](#_Toc122641236)

[Google & ChatGPT 4](#_Toc122641237)

[Reflexion der Bildmanipulation 5](#_Toc122641238)

[Reflexion zum Video 5](#_Toc122641239)

[Reflexion Webseite 5](#_Toc122641240)

[Fazit 5](#_Toc122641241)

# Entwicklung ePortfolio

## Eingesetzte Technologien

### Dynamische Logofarbe

Bilder im SVG-Format lassen sich mit CSS verändern. Allerdings wird eine ganze Reihe von Schritten benötigt, um die pfade im SVG zu designen. Mein Ziel war es, mittels eines in der Navigation abgebildeten Sliders die Farbe vom Logo zu verändern.

Als erstes habe ich das Logo mithilfe von vectr.com erstellt. Von dort habe ich es als SVG exportiert. Ich habe herausgefunden, dass es am besten ist, wenn man möglichst keinen leeren Rand um das Logo herum hat. So muss man später beim Einbinden auf der Webseite nichts mit CSS abschneiden.

Danach habe ich das Logo mit dem HTML Object Tag eingebunden. Noch einfacher wäre es, das SVG direkt als code in die HTML-Datei hineinzukopieren. Allerdings finde ich das keine gute Idee, weil durch die normalerweise viel Code enthaltenden SVGs die HTML-Datei unübersichtlich wird.

Weil man von ausserhalb des Object Tags keine CSS-Stylings vornehmen kann (Ausser mithilfe von JavaScript), muss man den Sourcecode vom CSV anpassen. Ich habe also als erstes einen Code Formatierer benutzt, um den Code vom CSV übersichtlich darzustellen.

Um das Logo beliebig skalieren zu können, habe ich in der CSV-Datei alle vector-effect="non-scaling-stroke" entfernt. Vor dieser Änderung wurden die Linien im Logo nicht dünner, wenn ich das Logo verkleinert habe:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Dann habe ich die inline fill Styles von allen Path Tags entfernt. Anstelle dieser habe ich ein css file im CSV verlinkt, welches die Farbe der Path Tags setzt. Dazu habe ich eine CSS-Variable verwendet, weil ich deren Wert später einfach mit JavaScript bearbeiten kann.

Um mit JavaScript auf die CSS-Variable innerhalb des Object Tags zuzugreifen habe ich diese Zeile Code ausgetüftelt:



Den RGB-Wert berechne ich basierend auf einem HTML Range Input, welcher einen Wert zwischen 1 und 100 zurückgibt. Um diese Zahl zusammen mit vordefiniertem Start und Endfarben in einen RGB-Wert umzuwandeln habe ich folgende Funktionen geschrieben:

Ein Bild, das Text, Bildschirm, Screenshot, silbern enthält.

Automatisch generierte Beschreibung

### Sass

Ich bin mir aus dem Lehrbetrieb gewohnt immer mit SCSS anstelle von CSS zu verwenden. Für das ePortfolio habe ich dazu die Visual Studio Code Extension Live Sass Compiler verwendet.

### Bootstrap

Auch Bootstrap kenne ich aus einigen vorherigen Webprojekten, in welche ich involviert war. Für das ePortfolio habe ich es hauptsächlich für die dynamischen Schriftgrössen für mobile und die verbesserten Standarddesigns von HTML-Elementen benötigt. Die Bootstrap Klassen habe ich kaum eingesetzt. Darum denke ich, dass ich nächstes Mal eher einen Ansatz mit einer kleineren Library wie z.B. Reset.css wählen werde. Ausser ich weiss bereits zu Beginn, dass ich viele Bootstrap Elemente verwenden werde.

## Abweichungen zum Konzept

### Design vom Range Input

Im Storyboard habe ich als Designelement einen grauen Strich im Slider gehabt:



Leider lässt sich so etwas mit den Standard HTML-Slidern kaum realisieren. Ich benötigte bereits viel Zeit für den Farbverlauf des Balkens, die Grösse und das Design des Kreises. Normalerweise erledigt SASS das browserspezifische Implementieren von Elementen. Aber für die Range Inputs existiert so etwas nicht. Darum musste ich selbst die CSS-Styles für mehrere Browser zusammensuchen. Am Ende habe ich es darum bei einem simpleren, ebenfalls gutaussehenden Sliders belassen:



### Farbstreifen als Hintergrund

Im Storyboard habe ich für die Desktopansicht als Hintergrund farbige Streifen hinzufügen wollen:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ich habe mit ca. 2-3h Aufwand gerechnet, weil ich für eine andere Webseite bereits etwas ähnliches mit Zahlen mit p5.js implementiert habe:

Ein Bild, das Text, Elektronik enthält.

Automatisch generierte Beschreibung

Allerdings habe ich bereits zu Beginn geplant, die Farbstreifen als letztes zu implementieren falls noch Zeit übrigbleibt. Ich habe die Zeit dann lieber in die Dokumentation gesteckt, weil die Farbstreifen sich im Vergleich weniger gelohnt hätten.

## Quellen

Um das ePortfolio zu programmieren habe ich eine vielfältige Anzahl an Referenzen benutzt.

### Bisherige Webseiten

Für die Grundidee und die Navigation habe ich mich auf Webseiten, welche ich in den letzten Jahren realisiert habe, abgestützt. Dazu gehören:

* <http://robinnater.epizy.com/skibegeisterung>
* <https://physiotherapie-wick.ch>
* <https://volleyballnacht.jublanz.ch>
* <https://fondueplausch.jublanz.ch>

Die Navigation habe ich zum Beispiel im ersten Lehrjahr nur mit CSS implementiert. Später habe ich sie für die Webseite meiner Mutter wiederverwendet und angepasst. Jetzt im vierten Lehrjahr habe ich einige Verbesserungsmöglichkeiten gesehen, welche ich dann, während dem Realisieren vom ePortfolio umgesetzt habe. Beispielsweise war der Hover Effekt nur auf dem Text anstelle vom ganzen klickbaren Bereich:



### Google & ChatGPT

Für diverse Probleme und fehlendes Wissen habe ich Google und ChatGPT benutzt. Google verwende ich schon seit ich programmiere, darum habe ich es langsam draussen, wie ich suchen muss, um die gewünschten Ergebnisse zu bekommen. ChatGPT habe ich im Rahmen dieses Projektes zum ersten Mal verwendet. Ich habe es geschafft, dass die AI mir die JavaScript Funktion, um den RGB-Wert zu berechnen zu refactoren. Hilfreich habe ich dabei vor allem das Feedback gefunden. Ich wusste beispielsweise noch nicht, dass empfohlen wird innerhalb einer Funktion falls möglich Variablen mit const anstelle von var zu definieren. Mir wäre es nicht in den Sinn gekommen nach so etwas zu Googeln.

# Bildmanipulationen

//Gewünschte Bildänderungen beschreiben  
//Zu erzielenden Verbesserungen beschreiben  
//Reflexion

//Problem 80% filter auf dunkelrotes Auto

# Video

## Dreh

Ich habe das Video hauptsächlich im Auto gedreht. Da man als Beifahrer sowieso nicht viel zu tun hat, hatte ich viele Gelegenheiten schöne Aufnahmen zu machen. Zum Teil konnte ich das Filmen mit einer Aufnahme von einem Bild kombinieren. Alle Aufnahmen habe ich nur für das ePortfolio aufgenommen.

Für die Aufnahme in der Garage hat mir mein jüngerer Bruder Benjamin geholfen. Er hatte verschiedene Ideen aus welcher Perspektive wir filmen könnten und hat mir gerne geholfen. Die Grösste Challenge war, genug aber nicht zu viel Licht zu haben. Am Ende haben wir mit eingeschalteter Handytaschenlampe aber ohne Garagenlicht gefilmt.

## Schnitt

Zum Schneiden vom Video habe ich DaVinci Resolve verwendet. Ich habe dieses Schnittprogramm bereits mehrmals für andere Videos verwendet. Leider kam mein PC mit dem in 4K aufgenommenen Material an seine Grenzen. Das machte sich durch Lags beim Abspielen vom Video im Schnittprogramm bemerkbar.

Als Hintergrundmusik wollte ich erst Sun von Yu-Utsu verwenden, weil mir dieser Song als Hintergrundmusik in einem YouTube Video positiv aufgefallen ist. Leider habe ich nicht herausgefunden über welches Urheberrechts- / Lizenzmodell der Song verwendet werden kann. Möglicherweise müsste man den Künstler persönlich anfragen. Darum habe ich auf Julias Empfehlung hin einen Track aus der YouTube Library verwendet.

Als erstes habe ich alle Medien importiert:  
Ein Bild, das Text, Screenshot, Monitor, Bildschirm enthält.

Automatisch generierte Beschreibung

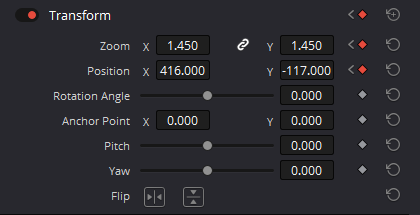
Dann habe ich die Clips in die richtige Reihenfolge gebracht. Ich habe dabei versucht, einen Tag darzustellen, an welchem jemand am Morgen aus der Garage fährt und erst am Abend wieder nach Hause kommt.

Um das Hinaus und Hineinfahren aus der Garage etwas spannender zu gestalten, habe ich die beiden Clips beschleunigt:

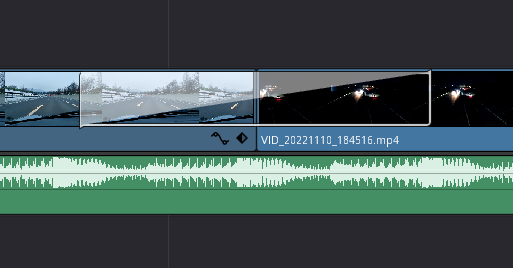
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Als nächstes habe ich die Clips und die Musik zusammengeschnitten und zusammengesetzt. Als Letztes habe ich ein paar Effekte angewendet, damit das Video mehr aus einem Guss wirkt:

* Zoom (Unter die Brücke, aus der Garage heraus und um die Motorhaube auszublenden):  
  
* Fade in und fade out, weil ich nicht den ganzen Track verwende:  
  Ein Bild, das Text enthält.

  Automatisch generierte Beschreibung
* Übergänge zwischen den verschiedenen Clips:



## Reflexion

Beim Drehen habe ich gemerkt, dass es schwierig ist die Kamera schön ruhig zu halten. Professionelle Aufnahmen aus einem Auto heraus sind aufgrund der Erschütterungen fast nur mit einem Gimbal möglich. Für die Aufnahme in der Garage hat mein Bruder eine Tonne verwendet, um das Handy ruhiger zu halten. Weil ich bereits einmal ein Video geschnitten habe, habe ich nicht erneut den Fehler gemacht im Hochformat zu filmen.

Das Schneiden ist erstaunlich schnell gegangen. Die Übergänge habe ich so noch nie verwendet, hat aber nach dem Anschauen von einem kurzen Tutorial auf Anhieb funktioniert. Für ein nächstes Mal nehme ich mir vor vor und nach der eigentlichen Aufnahme einige Sekunden länger zu filmen, so wären noch mehr ineinanderfliessende Übergänge möglich.

Dass mein Handy 4k unterstütz ist zwar eine gute Sache. Da es aber während dem Schneiden zu sehr geruckelt hat werde ich nächstes mal wieder auf eine geringere Auflösung wechseln.

# Reflexion Webseite

//Techniken  
//Erfahrungen

# Fazit

//Gelerntes, Erfahrungen etc.