Seminararbeit
CSS Animationen und Transformationen

Raetz, Fabian
Praktische Informatik
Fachhochschule Dortmund
Matrikelnummer: 7073500
fabian.raetz@gmail.com

Betreuerin: Prof. Dr. Sachweh
Abgabe der Arbeit: 23.06.2013

Inhalt

Einleitung

Traditionell wurden Webseiten immer mit einer festen Breite (z.B. 960px) angelegt. Dieses Vorgehen hat viele Vorteile und vereinfacht die Entwicklung ungemein, da sich die Entwickler auf eine fest definierte Größe konzentrieren können. Seit Apple's iPhone auf dem Markt ist, steigt die mobile Nutzung von Webseiten jedes Jahr. Zu den Nutzern mit kleinen Bildschirmen, der Mobilfunkgeräte, kommen auf der anderen Seite immer mehr Nutzer mit Bildschirmen die eine 16:9 Auflösung besitzen.

In diesem neuen Markt, mit unterschiedlichsten Auflösungen, in der die Differenz zwischen kleinster und größter Auflösung immer größer wird, ist ein Seitenlayout mit fester Breite nicht mehr praktikabel. [Frai12, S.7ff]

Für die Optimierung von Webseiten für verschiedene Auflösungen steht eine Vielzahl technischer Möglichkeiten zur Verfügung. Diese Optimierungen werden unter dem Begriff "**responsive web design**" zusammengefasst. [Frai12, S.10f]

Diese Arbeit untersucht die Verwendung von CSS Animationen und Transformationen im Kontext von responsive web design.

Im ersten Teil der Arbeit wird dargelegt, für welche Anwendungszenarien die Verwendung von Animationen einen Mehrwert bietet. Teil zwei evaluiert verschiedene Technologien, mit dessen Hilfe Animation und Transformationen im Web umgesetzt werden können. In Teil drei werden die CSS Module CSS-Transforms, CSS-Transitions und CSS-Animations evaluiert, mit denen komplexe Animationen und Transformationen möglich werden. Zum Schluss wird die Verwendung von CSS Animationen und Transformationen anhand eines kleinen Fallbeispiels verdeutlicht.

Anwendungszenarien

Es gibt verschiedene Anwendungszenarien in denen Animationen einen Mehrwert bieten können. Einige davon sind z.B.:

Aufmerksamkeit erzielen

Mit Animationen und bewegten Inhalten erhält man die Aufmerksamkeit des Betrachters. Dieser Effekt wird z.B. von Werbebannern genutzt und soll zum Anklicken der Werbung bewegen. Für die Vorstellung neuer Produkte oder Dienstleistungen wird derselbe Effekt genutzt.

Veranschaulichung

Komplexe Abläufe und Prozesse lassen sich mit Animationen besser veranschaulichen um so die Verständlichkeit zu erhöhen.

Techniken zur Umsetzung

Mit der Zeit haben sich verschiedenste Techniken etabliert, mit denen Animationen und Transformationen im Web realisiert werden können. Jede dieser Techniken hat dabei seine Stärken und Schwächen. Zu den bekanntesten Techniken gehören:

- Adobe Flash
- JavaScript und
- Cascading Style Sheets 3 (CSS3). [Stor12, S.1]

Es gibt weitere Techniken um Animation und Transformation im Web zu realisieren. Diese werden in dieser Arbeit aber nur der Vollständigkeit halber aufgeführt und werden nicht näher betrachtet. Zu den Techniken gehören:

- Microsofts Silverlight bzw. Moonlight. [MacD12, S.xxxff]
- JavaFX. [Dea11, S.xiii]
- Scaleable Vector Graphics (SVG) welche sich in Webseiten einbetten lassen. [SVGIntro, 1.1]

Adobe Flash

Die älteste Technik, die sich etabliert hat, basiert auf dem proprietären Browser Plugin Adobe Flash welches im Jahr 1996 in der ersten Version erschienen ist. Bis heute hat Adobe Flash eine beachtliche Verbreitung erlangt. [HistoryOfFlash, S.4] Dank der Entwicklungsumgebung lassen sich sehr einfach Animationen erstellen. Zudem bietet Flash mit ActionScript eine Programmiersprache die Schleifen, Variablen, Funktion und mehr unterstützt und sich dadurch flexibler einsetzen lässt als Cascading Style Sheets. [Stor12, S.7]

Flash bringt aber auch eine Reihe von Nachteilen mit. Da Flash ein Browserplugin (Flash Player) voraussetzt, muss dieses Plug-In von jedem Nutzer installiert und ständig aktualisiert werden. Es tauchen immer wieder Sicherheitsprobleme im Zusammenhang mit dem Flash Player auf. [FlashPlayerSec] Flashinhalte innerhalb einer Webseite sind nicht für Suchmaschinen und Bildschirmlesegeräte zugänglich. Des Weiteren steht der Flash Player nicht auf allen Plattformen zur Verfügung. Dies betrifft vor allem mobile Endgeräte wie z.B. IOS, Android und Windows Mobile.

[Stor12, S.7]. Damit ist der Einsatz von Adobe Flash für eine responsive Webseite, die auf mobilen Endgeräten funktionieren soll, nicht möglich.

JavaScript

JavaScript ist eine Scriptsprache mit Variablen, Schleifen, Funktionen und mehr, weshalb JavaScript flexibler als Cascading Style Sheets ist. JavaScript wird von allen modernen Browsern unterstützt. Des Weiteren gibt es eine Vielzahl von Frameworks[1] die bei der Erstellung von Animationen unterstützen. [Stor12, S.7] Traditionell werden Animation mittels JavaScript über Timer realisiert, die in regelmäßigen Intervallen Eigenschaften eines oder mehrerer Html-Element(e) ändern bis die Animation ihren Endzustand erreicht hat. Bei diesem Vorgehen hat der Browser kaum Gelegenheit zu optimieren und Berechnungen von der GPU vornehmen zu lassen. Je nachdem welche Art von Timer verwendet wird, laufen diese noch weiter, selbst wenn das entsprechende Browser-Tab nicht sichtbar ist. Dies führt zu einem erhöhten Energieverbrauch, was bei mobilen Endgeräten problematisch ist. [JSAnimPerf] Ein weiterer Nachteil ist, dass entsprechende Frameworks mehr Datenvolumen verursachen und so den initialen Seitenaufbau verlangsamen können.

Cascading Style Sheets

Mithilfe von Cascading Style Sheets (CSS) kann das Aussehen einer Webseite definiert werden. Mit CSS werden Eigenschaften wie z.B. Schriftart, Schriftgröße und Farbe an z.B. Html Dokumente gebunden. Die Trennung von Aussehen und Inhalt erleichtert die Erstellung von Webseiten. Zudem wird die Wartbarkeit erhöht. [CSS21Spec, Abstract]

CSS3 Animation und Transformation

Das World Wide Web Konsortium (W3C) hat sich entschieden, den Nachfolger von CSS 2.1 nicht in einer einzelnen Spezifikation zu entwickeln. Stattdessen werden verschiedene CSS Module definiert, die jeweils einen kleinen Teilaspekt von CSS3 spezifizieren. [CWS+12, S.2]

Um Animationen und Transformationen mithilfe von CSS umzusetzen benötigt man die folgenden CSS Module:

- CSS Transforms (http://www.w3.org/TR/css3-transforms/)
- CSS Transitions (http://www.w3.org/TR/css3-transitions/)

[1] z.B. jQuery, jquery-animate-enhanced, Processing.js und scripty2, YUI Transition

- CSS Animations (http://www.w3.org/TR/css3-animations/)

Alle drei CSS Module liegen als Working Draft vor (Stand 12.06.2012). Die drei oben genannten CSS Module liegen also noch nicht in ihrer finalen Form vor und es können noch Änderungen erfolgen, bis Sie als Recommendation vorliegen [WorkingDraft].

Bis die oben genannten CSS Module als Recommendation vorliegen, werden den einzelnen CSS Eigenschaften Hersteller Präfixe vorangestellt [VendorPrefix].

Um CSS Animationen und Transformationen schon heute Browserübergreifend zu nutzen müssen alle Eigenschaften zusätzlich mit Hersteller Präfixen angegeben werden. Um dies zu veranschaulichen dient das folgende Beispiel:

```
-moz-transform: translateX(200px);
-ms-transform: translateX(200px);
-o-transform: translateX(200px);
-webkit-transform: translateX(200px);
transform: translateX(200px);
```

In allen nachfolgenden Beispielen werden zur besseren Übersichtlichkeit und aus Platzgründen die Eigenschaften mit den Hersteller Präfixen weggelassen.

Transformation

Mithilfe von CSS Transformationen können (Html-) Elemente in einem zwei und drei dimensionalen Koordinatensystem verschoben, gedreht, skaliert und geschert werden. [CSSTransforms, Introduction] Über die CSS-Eigenschaft *transform* werden die gewünschte Transformation(en) angegeben.

Im Folgenden werden die Transformationen *translate* und *rotate* in einem 2d Koordinatensystem näher erläutert. Des Weiteren wird gezeigt wie der Ursprung von Transformationen verschoben werden kann. Für einen Überblick über weitere 2d und 3d Transformationen wird an dieser Stelle auf die Spezifikation des w3c verwiesen. http://www.w3.org/TR/css3-transforms/

translate

Um (Html-) Elemente z.B. um 30px nach unten und 30px nach rechts zu verschieben (Abb.13), kann folgende transform Eigenschaft für das Element deklariert werden: *transform: translate(30px, 30px);*

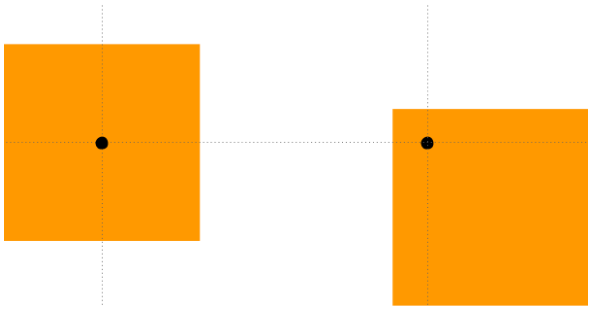

Abbildung 1: translate

Um die Verschiebung nur in eine Richtung vorzunehmen, kann alternativ auch transform: translateX(30px) oder transform: translateY(30px) verwendet werden. Alle Verschiebungen werden relativ zu ihrem Ursprung durchgeführt.

rotate

Mithilfe von rotate lassen sich (Html-) Elemente beliebig um den eigenen Ursprung drehen. Um ein (Html-) Element wie in Abb. 2 um 45 Grad zu drehen kann folgenden transform Funktion zugewiesen werden: *transform: rotate(45deg);*

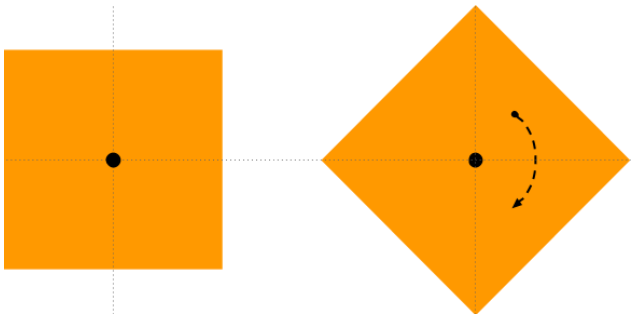

Abbildung 2: rotate

transform-origin

Will man den Ursprung von Transformationen verschieben, verwendet man die CSS Eigenschaft *transform-origin*. Standardmäßig befindet sich der Ursprung immer in der Mitte des (Html-) Elementes (*transform-origin: 50% 50%;*). Um den Ursprung in die Ecke oben links des Elements zu verschieben muss folgende CSS Eigenschaft verwendet werden: *transform-origin: 0 0;* Abb. 3 zeigt einen verschobenen Ursprung, in Verbindung mit einer Rotation um 45 Grad.

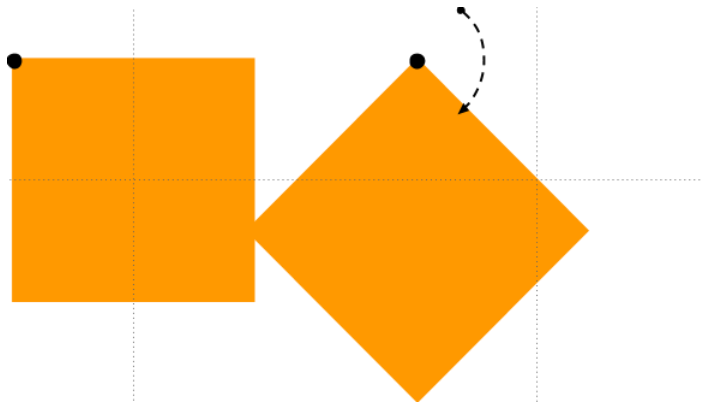

Abbildung 3: transformßorigin / rotate

Transition

Mithilfe von CSS Transitionen können Änderungen von CSS Eigenschaften über eine definierte Zeitspanne von Zustand A nach Zustand B erfolgen. [CSSTransitions, Abstract] Hierzu wird über die CSS Eigenschaft *transition-property* die zu verwendende CSS Eigenschaft angegeben. *transition-property* kann eine oder mehrere CSS Eigenschaften übergeben werden. Um alle Eigenschaften mit einer Transition zu verwenden kann die Konstante all verwendet werden. Die Zeitspanne in der die Transition erfolgen soll wird mit der CSS Eigenschaft *transition-duration* festgelegt. Um wie in Abb. 4 eine Rotation über drei Sekunden ablaufen zu lassen, kann man folgendes CSS verwenden:

transform: rotate(45deg);

transition-property: all;

transition-duration: 3s;

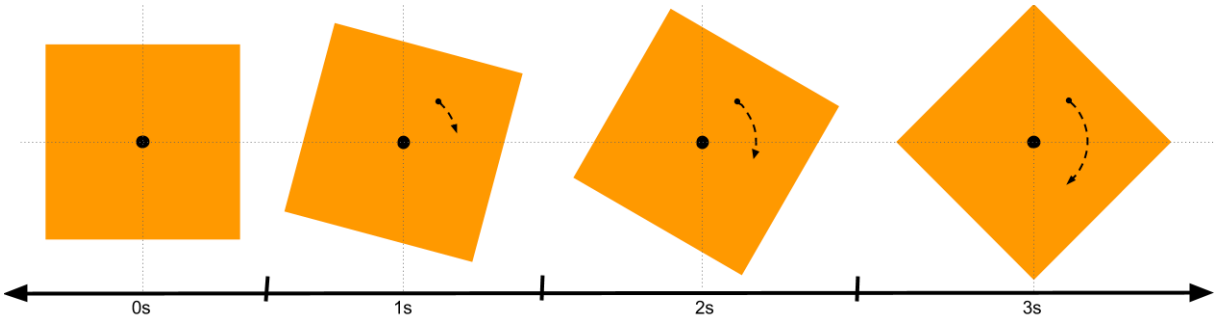

Abbildung 4: transition

Standardmäßig verwenden die Transitionen ein lineares Geschwindigkeitsprofil. Über die CSS Eigenschaft *transition-timing-function* können andere Geschwindigkeitsprofile gewählt werden. [CSSTransitions, 2.3]

Animation

Wie CSS Transitionen können mithilfe von CSS Animationen ebenfalls Änderungen von CSS Eigenschaften über eine definierte Zeitspanne umgesetzt werden. Der Unterschied ist jedoch das CSS

Transitionen implizit ausgelöst werden, wenn sich die definierten CSS Eigenschaften ändern. Währenddessen werden CSS Animationen explizit ausgelöst, indem einem (Html-) Element eine Animation zugewiesen wird. Zudem können viele Aspekte einer CSS Animation kontrolliert werden, die mit CSS Transitionen nicht kontrolliert werden können. Es kann explizit angegeben werden ob die Animation läuft oder pausiert, wie viele Iterationen die Animation laufen soll und ob die Animation verzögert startet. [CSSAnimations, Introduction]

Um ein Element von links nach rechts hin und her zu animieren kann folgende CSS Animation definiert werden:

```
#elem {
  animation-name: rotateToRectMove;
  animation-duration: 5s;
  animation-iteration-count: infinite;
  animation-direction: alternate;
  animation-timing-function: ease-in-out;
}

@keyframes rectMove {
  20% { transform: translateY(100px); }
  80% { transform: translateX(700px) translateY(100px); }
  100% { transform: translateX(700px); }
}
```

Mithilfe der *@keyframes* Regel lassen sich Keyframes definieren die im Anschluss in CSS Animationen verwendet werden können. Die rectMove *keyframe*-Definition besagt, dass bei 20% der Animation, sich das Element um 100px nach unten bewegt hat. Bei 80% befindet sich das Element immer noch 100px unter der Ausgangssituation ist aber um 700px nach rechts gewandert. Sobald die Animation beendet ist, befindet sich das Element 700px rechts von der Ausgangsposition.

Mittels der CSS Eigenschaft *animation-name* lassen sich keyframes zu Elementen zuordnen. Die CSS Eigenschaft animation-duration legt die Dauer der Animation in Sekunden fest. Damit die Animation unendlich lange läuft, muss die CSS Eigenschaft *animation-iteration-count* mit *infinite* initialisiert werden.

Die Animation läuft nun unendlich lange. Am Ende der Animation springt Sie aber auf ihre initiale Position zurück und fängt von vorne an. Indem man die CSS Eigenschaft *animation-direction* mit *alternate* initialisiert, springt die Animation nicht mehr an den Anfang. Die Animation läuft nun beim zweiten Mal Rückwerts ab und animiert nun wie gewünscht hin und her.

Browser Kompatibilität

Alle modernen Browser unterstützen die drei CSS Module (CSS-Tranforms, CSS-Transitions und CSS-Animations) mit dessen Hilfe sich Animationen und Transformationen im Web umsetzen lassen. Einzig

der Internet Explorer 8/9 und Opera Mini 5-7 unterstützen keine Animationen und Transformationen. [BrowserCompat]

Viele Browser verwenden noch Hersteller-Präfixe, weshalb zum jetzigen Zeitpunkt (22.06.2013) noch alle CSS Eigenschaften mit diesen versehen werden müssen um in allen Browsern zu funktionieren.

Praktische Umsetzung

Um Animationen und Transformationen mithilfe von CSS3 zu evaluieren, wird eine Webseite entwickelt, die für verschiedene Endgeräte optimiert ist. Hierbei werden sowohl Smartphones mit kleinen, aber auch große Bildschirme mit einer 16:9 Auflösung berücksichtigt.

Ziel ist es, an möglichst vielen sinnvollen Stellen mit Animationen und Transformationen zu arbeiten.

Um mehr Funktionen innerhalb des dafür vorgesehen Zeitraums zu ermöglichen wird auf Browserkompatibilität weitestgehend verzichtet und die Webseite für den Browser Google Chrome optimiert. Generell wird sich bei der Entwicklung an bestehende Recommendations und Working Drafts des W3C gehalten, weshalb die Webseite in absehbarer Zeit in anderen Browsern fehlerfrei betrachtet werden kann.

Die Webseite ist unter folgender Adresse erreichbar:

http://mischi.github.io/seminar-responive-webdesign/

Szenario

Es wird eine neue Webseite für die Brockhaus AG Labs erstellt. Ziel der Webseite ist es, über die Vorgehensweise, Ziele und Mitarbeiter der Brockhaus AG Labs zu informieren. Zudem wird es einen Bereich geben, in dem Stellenausschreibungen für die verschiedenen Abteilungen zu finden sind.

Umsetzung

Die Webseite wird als eine einzelne Seite entworfen, bei der man durch scrollen zu den einzelnen Unterseiten kommt. Dies hat den Vorteil, dass z.B. Seitenübergänge animiert werden können. Diese Art von Webseiten wird auch unter dem Begriff “Single Page Apps” zusammengefasst.

Damit der Nutzer nicht den Fokus verliert, wird versucht so wenig Inhalt wie möglich auf einer Seite darzustellen. Müssen mehr Informationen auf einer Seite untergebracht werden, werden diese durch bestimmte Aktionen, wie z.B. ein Klick auf einen Button, oder zeitgesteuert sichtbar. Diese Aktionen werden mit passenden Animation unterlegt, sodass der Nutzer ein visuelles Feedback bekommt.

Seitennavigation

Animierter Seitenwechsel

Bei der Seite kommt eine auf Ankern basierte Navigation zum Einsatz. Hierbei wird jede Unterseite in einen *DIV-Container* geschachtelt, welcher eine eindeutige ID zugewiesen bekommt. Die *Anker-Elemente* im Menü bekommen im *href* Attribut die entsprechende ID mit angegeben. Bei einem Klick auf die *Anker-Elemente*, springt der Browser zu dem entsprechenden *DIV-Container*.

Um das "springen" des Browsers zu verhindern und stattdessen eine Animation zu verwenden, wird das *skrollr-menu* JavaScript-Framework verwendet. Mit dem skrollr-menu Framework ist es möglich die Dauer der Animation und ein Geschwindigkeitsprofil (easing function) anzugeben. Wird auf ein *Anker-Element* geklickt wird zu dem dazugehörigen *DIV-Container* hin animiert (sanft gescrollt).

Aktives Menü-Element

Bei einem Klick auf ein Menü-Element wird mithilfe von JavaScript die *CSS-Klasse "active"* auf das angeklickte Menü-Element gesetzt und bei allen anderen Menü-Elementen wird die CSS-Klasse entfernt.

Der Farbwechsel wir mithilfe einer CSS Transition untermalt, sodass die Farbe langsam von einer zur anderen wechselt.

Wenn der Nutzer der Webseite nach unten oder oben scrollt, muss das aktive Menü-Element immer synchronisiert werden, je nachdem welcher DIV-Container sich gerade im Viewport des Browsers befindet. Hierzu guckt man welcher DIV-Container sich am oberen Rand des Viewports befindet, findet den entsprechenden Menü-Anker und setzt die CSS-Klasse. Anschließend wird die entsprechende CSS-Klasse von allen anderen Menü-Ankern entfernt. Diese Funktionalität wurde mithilfe des jQuery-waypoint JavaScript Framework realisiert.

Desktop Ansicht

Auf Bildschirmen modernen Desktop Rechner hat man in der Regel sehr viel Platz. Aus diesem Grund wurden alle Elemente der Webseite sehr groß gestaltet um den verfügbaren Platz zu nutzen.

Animierter Inhalt bei Seitenwechsel

Um z.B. Überschriften zu animieren sobald diese den Viewport des Browsers betreten und die Animation ihren Endzustand in der Mitte des Viewports erreichen soll, bedarf es eines JavaScript Frameworks. Ohne JavaScript ist es nicht möglich Animationen relativ zum Viewport zu definieren. Für diese Funktionalität wurde in dieser Arbeit das JavaScript Framework skrollr verwendet. "skrollr allows you to animate any CSS property of any element depending on the horizontal scrollbar position. All

you need to do is define key frames for each element at certain points in top scroll offset.” [skrollr, Abstract]

Mithilfe von *skrollr* und CSS Transformationen wurden animierte Überschriften umgesetzt, die bei Sichtbarwerden einer neuen Unterseite von rechts hineingeflogen kommen. Sobald die Überschrift in der Mitte des Viewport ist, bleibt die Überschrift an ihrer endgültigen Position. Des Weiteren sind die Inhalte ausgeblendet und werden bis zur Mitte des Viewports langsam eingeblendet.

Startseite

Auf der Startseite wird der Besucher von einer Animation empfangen. Dabei ist die Animation in zwei Teile gegliedert. Teil eins der Animation heißt den Besucher der Webseite willkommen. Der zweite Teil der Animation veranschaulicht das Vorgehen der Brockhaus AG Labs von der Idee, über das Projekt, hin zu dem Ergebnis anhand eines Scrumboards. Beide Animationen wurden mithilfe CSS Animationen umgesetzt, welche wiederum CSS Transformationen und Transitionen nutzen. Dabei wurde jede der beiden Animationen in viele kleine Animationen unterteilt, die mithilfe der CSS Eigenschaft *animation-delay* zu fest definierten Zeitpunkten abgespielt werden. So aneinander gereiht, entsteht der Eindruck einer einzelnen großen Animation.

Abbildung 5: Startseite - Animation Teil 1

Abbildung 6: Startseite - Animation Teil2

Abbildung 7: Startseite - Animation Teil 3

Über Labs / Ziele / Wir suchen

In den Bereichen Über Labs, Ziele und Wir suchen werden die Inhalte auf verschiedene Art und Weisen präsentiert und mit Animationen versehen.

Um optisch anspruchsvolle Präsentationsformen für Inhalte zu erstellen und diese mit sinnvollen Animationen zu untermalen, ist Wissen in Design und Technik erforderlich.

In dieser Arbeit wird daher auf bestehende Lösungen zurückgegriffen und optisch dem Design der Webseite angepasst. Konkret wurden folgende Lösungen verwendet:

- Circle Hover Effects with CSS Transitions [CodropsCircle]
- Fluid CSS3 Slideshow with Parallax Effect [CodropsSlideshow]
- Responsive Content Navigator with CSS3 [CodropsContentNav]

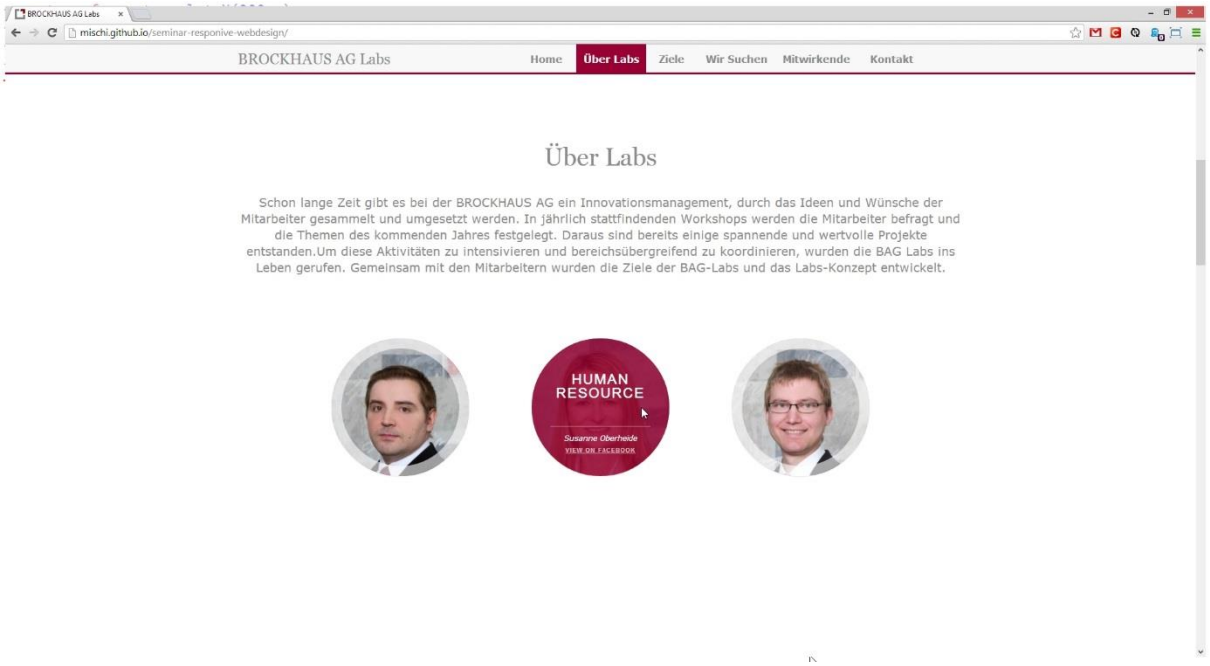

Abbildung 8: Über Labs

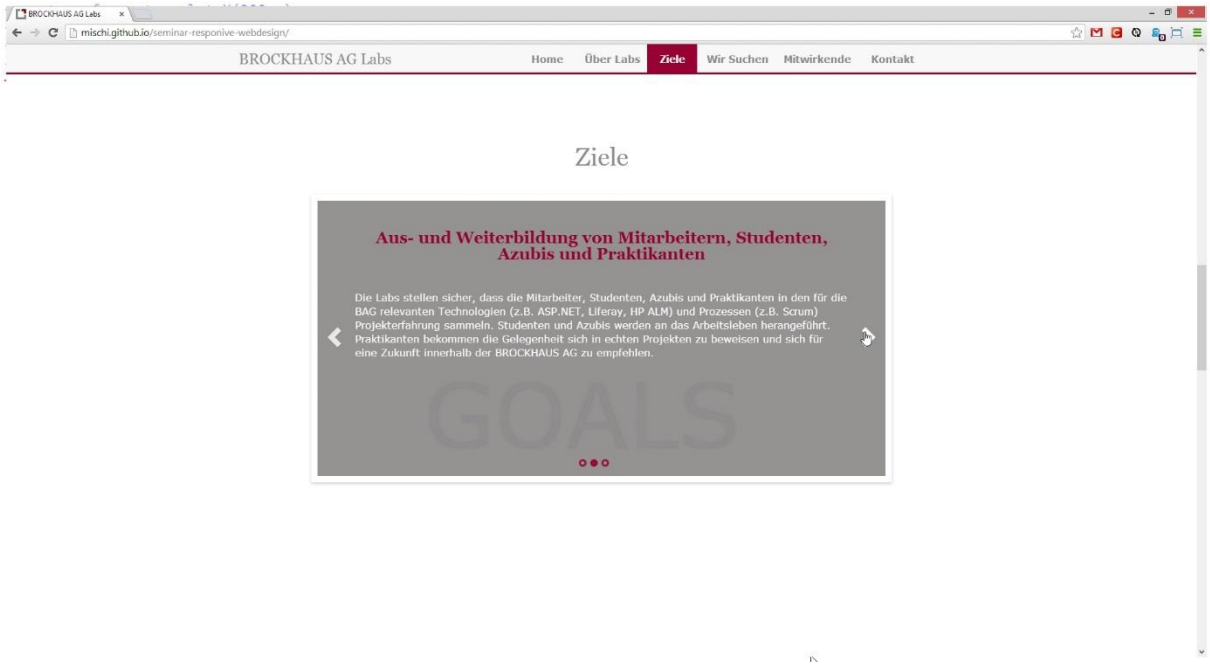

Abbildung 9: Ziele

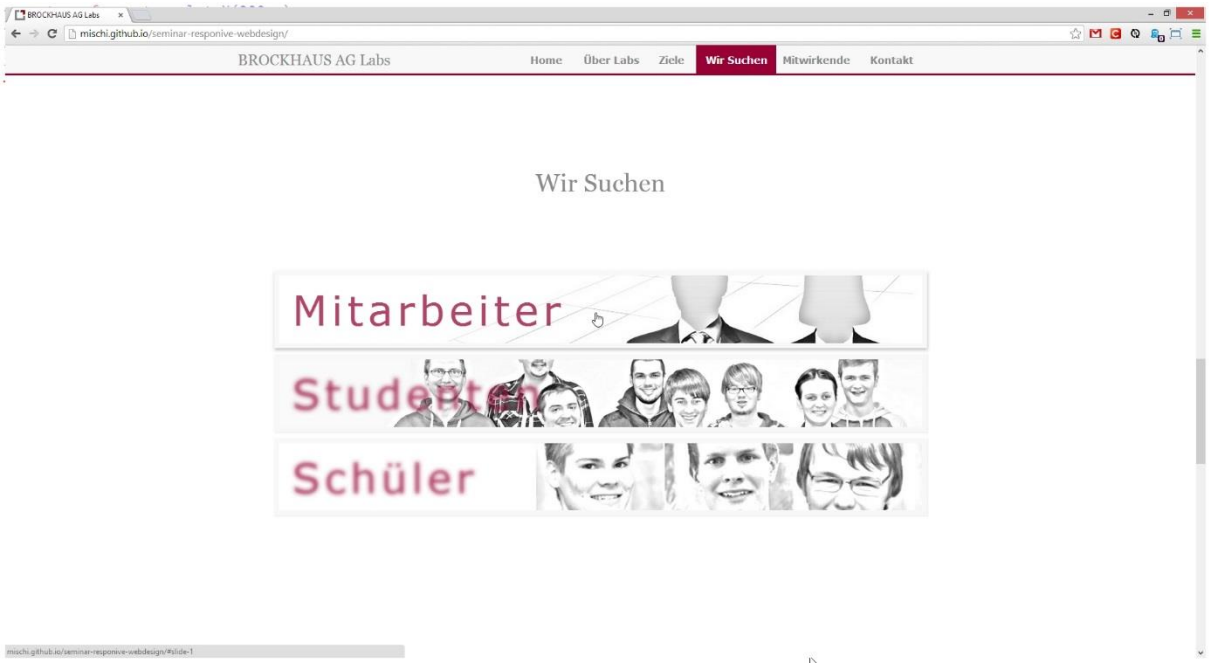

Abbildung 10: Wir suchen - Ebene 1

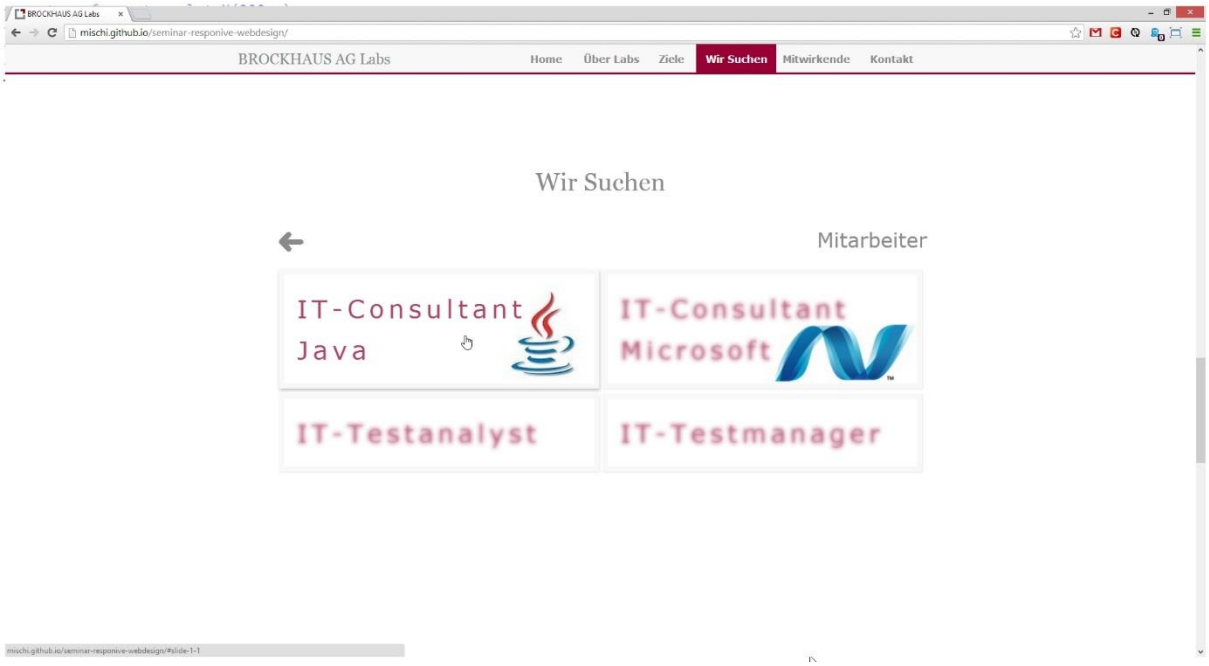

Abbildung 11: Wir suchen - Ebene 2

Smartphone Ansicht

Auf Smartphones ist der Platz begrenzt. Deshalb wird versucht, soviel Platz wie möglich, für den eigentlichen Inhalt zu verwenden. Zudem wird darauf geachtet, dass alle Inhalte gut lesbar sind und eine entsprechende Größe haben.

Kopfzeile

Um Platz zu sparen wird in der Smartphone Ansicht standardmäßig kein Menü angezeigt. Stattdessen befindet sich oben links in der Kopfzeile ein Button mit dessen Hilfe das Menü ein- und ausgeblendet werden kann.

Um noch mehr Platz für Inhalte zu schaffen, befindet sich der Titel des aktuellen Bereiches ebenfalls in der Kopfzeile. Scrollt der Benutzer zu einem anderen Bereich der Seite oder navigiert über das Menü dorthin, wird der aktuelle Titel mithilfe einer CSS Animation aktualisiert.

Browser Fullscreen Modus

In den aktuellen Versionen von Safari und Google Chrome wird die Browserleiste ausgeblendet, sobald der Benutzer auf der Seite nach unten scrollt. Scrollt der Benutzer wieder nach oben erscheint auch wieder die Browserleiste. Durch diesen Fullscreen Modus der Browser wird mehr Platz für Inhalte verfügbar.

Startseite

In der Smartphone Ansicht wird das Scrumboard komplett weggelassen, da es keinen Mehrwert an Informationen bietet. Stattdessen wird der ganze Bereich nur für die drei Punkte zum Vorgehen der Brockhaus AG Labs verwendet. Die drei Punkte zum Vorgehen der Brockhaus AG Labs werden wie in der Desktop Ansicht animiert und kommen nacheinander, von der rechten Seite, hineingeflogen.

Abbildung 12: Startseite

Menü

Durch einen Klick auf den Button in der Kopfzeile der Seite lässt sich das Menü ein- und ausblenden. Das Menü kommt ebenfalls von der rechten Seite hineingeflogen. Klickt man einen Menüpunkt an,

wird wie in der Desktop Variante, zu dem entsprechenden DIV-Container gescrollt. Währenddessen wird das Menü wieder ausgeblendet. Dies geschieht wiederum mit einer Animation.

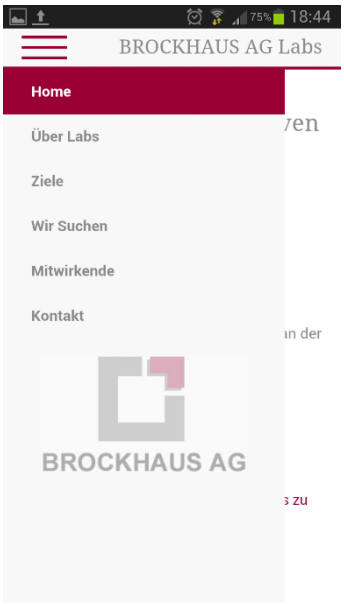

Abbildung 13: Menü

Probleme

Generell traten bei der Entwicklung drei Probleme auf.

Die Animation auf der Startseite besteht aus vielen kleinen Animationen. Es kann vorkommen, dass die Animationen die die User Stories von der „IN WORK“ Spalte zur „DONE“ Spalte verschiebt nicht im richtigen Moment abgespielt werden. Mit ein paar Sekunden Verzögerung werden dann die noch ausstehenden Animationen (gleichzeitig) abgespielt. Dieses Problem konnte nicht konstant reproduziert werden um es näher zu analysieren.

Teilweise wirken die Animationen nicht flüssig. Während der Entwicklung wurde nicht konsequent mit Transformationen gearbeitet, sondern es wurde viel über top/bottom/left/right positioniert. Wenn die Positionierung stattdessen konsequent mit translate/translateX/translateY umgesetzt worden wäre, hätte mehr von der GPU optimiert werden können [TranslateBetter].

Während der Entwicklung wurde mit Compass[2] gearbeitet um z.B. CSS Eigenschaften mit Hersteller Präfixe generieren zu lassen. Es wurde allerdings versäumt ein zusätzliches Modul für CSS Animationen zu verwenden. Dies hätte besonders bei der Animation auf der Startseite zu einer großen Einsparung von Quelltext führen können.

[2] http://compass-style.org/

Fazit und Ausblick

Für eine Webseite die sowohl auf mobilen Endgeräten als auch auf dem Desktop Animationen verwenden will, sollten diese mit Cascading Style Sheets umgesetzt werden. Sollte CSS alleine nicht ausreichen, um die gewünschten Animationen umzusetzen, können die Stärken von CSS und JavaScript kombiniert werden.

Weitere Arbeiten könnten sich mit der Optimierung von CSS Animationen in Hinblick auf Performance auseinandersetzen.

Literaturverzeichnis

[Stor12] Pro CSS3 Animation

[MacD12] Pro Silverlight 5 in C# Fouth Edition

[CWS+12] CSS3 Solutions - Essential Techniques for CSS3 Developers

[Dea11] JavaFX 2.0: Introduction by Example

[SVGIntro] SVG 1.1, http://www.w3.org/TR/SVG/intro.html, Abruf am 28.05.2013

[HistoryOfFlash] The History Of Flash, http://www.adobe.com/macromedia/events/john_gay/index.html, Abruf am 27.05.2013

[FlashPlayerSec] Security bulletins and advisories, http://www.adobe.com/support/security/#flashplayer, Abruf am 23.06.2013

[JSAnimPerf] Improving the Performance of your HTML5 App, http://www.html5rocks.com/en/tutorials/speed/html5/#toc-request-ani-frame, Abruf am 29.05.2013

[CSS21Spec] CSS 2.1 Specification, http://www.w3.org/TR/2002/WD-CSS21-20020802/, Abruf am 19.06.2013

[CSSTransforms] CSS Transforms, http://www.w3.org/TR/css3-transforms/, Abruf am 19.06.2013

[CSSTransitions] CSS-Transitions, http://www.w3.org/TR/css3-transitions/, Abruf am 22.06.2013

[CSSAnimations] CSS-Animations, http://www.w3.org/TR/css3-animations/, Abruf am 22.06.2013

[WorkingDraft] W3C Process Document, http://www.w3.org/2005/10/Process-20051014/tr#first-wd, Abruf am 12.06.2013

[BrowserCompat] Can I use..., http://caniuse.com/#cats=CSS, Abruf am 22.06.2013

[VendorPrefix] Experimental Implementations, http://www.w3.org/TR/css-2010/#experimental, Abruf am 12.06.2013

[skrollr] skrollr, https://github.com/Prinzhorn/skrollr, Abruf am 13.06.2013

[CodropsCircle] CIRCLE HOVER EFFECTS WITH CSS TRANSITIONS http://tympanus.net/codrops/2012/08/08/circle-hover-effects-with-css-transitions/, Abruf am 18.06.2013

[CodropsContentNav] RESPONSIVE CONTENT NAVIGATOR WITH CSS3 http://tympanus.net/codrops/2012/03/23/responsive-content-navigator-with-css3/, Abruf am 18.06.2013

[CodropsSlideshow] FLUID CSS3 SLIDESHOW WITH PARALLAX EFFECT http://tympanus.net/codrops/2012/04/30/fluid-css3-slideshow-with-parallax-effect/, Abruf am 18.06.2013

[TranslateBetter] Why Moving Elements With Translate() Is Better Than Pos:abs Top/left, http://www.paulirish.com/2012/why-moving-elements-with-translate-is-better-than-posabs-topleft/, Abruf am 22.06.2013