

Neue Animations- und Interaktionsmöglichkeiten im Web

Gegenüberstellung von Flash mit HTML5, CSS3 und JavaScript

Vera Geier

WDD 310

24. Jänner 2012

9.480 Wörter

SAE Institute Wien

Inhaltsverzeichnis

1	Einleitung.....	5
1.1	Zielsetzung	5
1.2	Aufbau	6
2	Begriffserklärungen.....	7
2.1	HTML/HTML5	7
2.2	JavaScript.....	7
2.3	CSS/CSS3.....	8
2.4	Adobe Flash	8
3	HTML5.....	9
3.1	Video	9
3.2	Programme für HTML5-Animationen	11
3.2.1	Authoring Tool	11
3.2.2	Flash-to-HTML5 Converter	13
3.3	Vor- und Nachteile gegenüber Flash.....	14
4	JavaScript.....	16
4.1	AJAX und Frameworks	16
4.2	Canvas und SVG	17
4.3	Vor- und Nachteile gegenüber Flash.....	20
5	CSS3.....	22
5.1	Herstellerspezifische Präfixe.....	22
5.2	Animationseigenschaften	22
5.2.1	Transformationen.....	23
5.2.2	Animationsverhalten	23
5.3	Vor- und Nachteile gegenüber Adobe Flash	26
6	Browserkompatibilität.....	27
6.1	HTML5.....	27
6.2	CSS3	28
6.3	Mobile Endgeräte.....	29
6.4	Browserstatistik.....	29
7	Adobe Flash oder HTML5, CSS3 und JavaScript	31
7.1	Flash.....	31
7.2	HTML, CSS und JavaScript	33
8	Schluss.....	35
9	Literaturverzeichnis.....	37

Abbildungsverzeichnis

Abbildung 1: Unterstützung der Browser für Videoformate	9
Abbildung 2: Benutzeroberfläche "Hype"	11
Abbildung 3: Benutzeroberfläche "Flash CS5"	12
Abbildung 4: Präfixe und ihre Browsergültigkeit	22
Abbildung 5: Browserstatistik laut browser-statistik.de.....	30

Listingverzeichnis

Listing 3-1: Videoeinbindung	10
Listing 4-1: Canvas - Kreiszeichnung mit Pfad	18
Listing 4-2: Canvas-Animation	19
Listing 4-3: SVG	19
Listing 5-1: Transform	23
Listing 5-2: Transition.....	24
Listing 5-3: CSS3 Keyframe-Animation	25
Listing 5-4: Animation-Kurzschreibweise.....	25

1 Einleitung

Die Zeiten wo eine Technologie über Jahre hinweg Bestand hatte, gehören nun der Vergangenheit an. Der Webauftritt muss dadurch ständig angepasst werden, um nicht altmodisch und anachronistisch zu wirken. Wo Anfangs das World Wide Web nur von technisch faszinierten Nutzern zum Einsatz gebracht wurde, dient es heute der Gesamtheit als Informationsquelle, Eigendarstellung und Kommunikationsaustausch mit anderen Nutzern. Die Plattformen Facebook, Twitter und Co sind bedeutende soziale Netzwerke für unsere Gesellschaft geworden und die Benutzerinteraktionen somit modifiziert und simplifiziert. Damit ist für Anwender die Möglichkeit gegeben, ohne Ablenkung, schnell und direkt an ihr Ziel zu gelangen.

Auch Geräte die einen Internetzugang aufweisen, verändern sich merklich. Die Beliebtheit, von Smartphones und Tablets, erreicht bereits jetzt den Höhepunkt bei Menschen jeder Altersgruppe. Jedoch weisen diese mobilen Gerätschaften andere Funktionalitäten, als gewöhnliche Computer auf.

Sowie sich der Trend des Webs ändert, ändern sich auch die Funktionalitäten der Software und Websprachen. Die bestehenden Sprachen verbessern sich und bieten permanent neue Möglichkeiten für die Gestaltung des Layouts an, wodurch erlernte Fähigkeiten verbessert und erweitert werden müssen. Es werden auch Programme angeboten, welche das Erstellen und Verwalten von Websites erleichtern und zeiteinsparend arbeiten. Des Weiteren spielt der heutzutage, wesentliche Kostenfaktor hier eine große Rolle. Mit HTML5, CSS3 und JavaScript werden somit Optionen bereitgestellt, die bisher nur mit Adobe Flash realisierbar waren.

Doch die Effizienz der neuen Technologien hat nicht nur positive Aspekte, da jede Sprache andere Funktionalitäten ausweist. Aufgrund dieser Tatsachen, wird in dieser Arbeit dargelegt, inwieweit sich Adobe Flash mit den Sprachen HTML5, CSS3 und JavaScript unterscheidet. Es wird darüber hinaus erörtert welche Lösungen in den verschiedenen Bereichen eingesetzt oder eher verzichtet werden sollen.

1.1 Zielsetzung

Ziel dieser Facharbeit ist es, die neuen Technologien HTML5, CSS3 und auch JavaScript über ihre Fortschritte bezüglich der Animationen und Interaktionen darzulegen. Die Möglichkeiten Bewegung im Web zu gestalten, haben sich radikal vereinfacht und werden immer weiter entwickelt. Deren Vorgänge und Techniken sollen in dieser Arbeit beschrieben werden. Durch die schnelle Entwicklung und der einfachen Bedienung ist es oft nicht mehr

nötig die Komplexität von Adobe Flash und ActionScript zu erlernen und auszuführen. Wo bei vielen Websites bisher nur die Möglichkeit bestand Flash einzusetzen, können jetzt die Plattformen, durch den Einsatz der neuen Technologien und mit Hilfe von Plug-Ins, fast gänzlich darauf verzichten. Doch wo sind eigentlich die Grenzen, in denen man Adobe Flash nicht mehr oder weiterhin benötigen wird?

Ebenso wird ein umfangreicher Teil dieser Abhandlung, die Gegenüberstellung von HTML5, CSS3 und JavaScript mit Adobe Flash sein. Es wird geschildert, wie die verschiedenen Animationstechniken am effizientesten genutzt werden können. Die Kompatibilitäten und Möglichkeiten unterscheiden sich merklich, sodass es in jeder Sprache seine Vor- und Nachteile gibt. So entwickelt sich der Trend in moderne, mobile Endgeräte und plattformübergreifende Onlinepräsenzen. Die vielen verschiedenen Betriebssysteme machen es oft nicht einfacher, dass jede Sprache mit ihnen einwandfrei kompatibel ist. Auch mehrere, unähnliche Internetbrowser haben unterschiedliche Eigenschaften und sind nicht immer auf demselben Niveau. So stellt sich nun die zweite Frage, ob es nun klüger ist weiterhin mit Flash zu arbeiten, oder auf HTML5, CSS3 und JavaScript umzusteigen.

1.2 Aufbau

Im Folgenden werden die wichtigsten Begriffe kurz erklärt, die das Verständnis der Inhalte klarstellen. Des Weiteren werden neue HTML-Tags und Anwendungen für Animationen und Filme von der Technologie HTML5 vorgestellt. Im vierten Kapitel werden CSS3-Eigenschaften, die für Interaktionen und Bewegung im Web von Bedeutung sind, dargelegt. Außerdem werden hier auch noch Funktionen beschrieben, die als Ausgangspunkt für die weitergehenden Animationen relevant sind. Im nachfolgenden Teil werden Methoden von JavaScript erklärt, welcher aber auch Teile von HTML5 beinhaltet, da diese Sprache Funktionen der anderen enthält. Anschließend kommt ein allgemeiner Teil der Browserkompatibilität, da dies ein wesentlicher Faktor für den nachfolgenden Abschnitt, welcher die Vor- und Nachteile betrifft. In diesem Bereich wird einer Flash sein und der zweite HTML5, CSS3 und JavaScript. HTML5 und CSS3 werden darüber hinaus mit Codebeispielen erklärt. Im Kapitel JavaScript werden nur die neuesten Technologien mit solchen Beispielen angeführt, da Animationen dieser Sprache schon etwas länger existieren. Flash wird deshalb nur im Kapitel „Adobe Flash oder HTML5, CSS3 oder JavaScript“ über seine positiven und negativen Aspekte dargelegt, weil auch dies schon eine ältere Technologie ist. Um aber eine möglichst gute Gegenüberstellung zu finden, wird dieser Teil der Arbeit reichhaltig wiedergegeben.

2 Begriffserklärungen

In diesem Kapitel werden die wichtigsten, grundlegenden Begriffe kurz erläutert, die für den gesamten Aufbau der Abhandlung relevant sind. Dem Verständnis, der erörterten Technologien, wird somit eine Vorstellung gegeben, wie die Funktionalitäten in weiterer Folge arbeiten.

2.1 HTML/HTML5

HTML (Hypertext Markup Language) ist die Sprache, die die Struktur von Webseiten beschreibt. Es werden hier Überschriften, Texte, Listen, Links, Tabellen, etc. ausgegeben und dann mit CSS formatiert.¹ Die bisher gängigen Ausführungen HTML 4.01, XHTML 1.0 und XHTML 1.1 sollen nun von der neuen Version HTML5 verdrängt werden. HTML5 bringt sehr viele Neuerungen mit sich, wie zum Beispiel neue Tag, für oft angewandte Bereiche wie Header, Footer, und Section. Mit dem Canvas-Tag kann eine leere Zeichenfläche auf die Internetseite eingebettet werden (*Kiupel, 2011*). Diese neuen Tags sollen helfen, das Dokument vernünftiger auszuzeichnen. Der Browser kann eine Verbindung zwischen dem Element und dem Inhalt der Website erkennen und fördert somit die Suchmaschinen und Screenreader das Wesentliche der Website zu finden (*Braun, 2011, S. 8*). Es gibt noch sehr viel mehr Aktuelles, auf das aber nicht weiter eingegangen werden soll.

2.2 JavaScript

JavaScript ist eine clientseitige Programmiersprache, die zum Ausbau von HTML entworfen wurde. JavaScript ist eine eher simple Sprache, die direkt in das HTML-Dokument eingebettet wird. Für diese Sprache gibt es zahllose, nützliche Erweiterungen, die es dem Webdesigner oft einfacher machen, schnell und zielorientiert zu arbeiten.² Zum Beispiel sind sogenannte JavaScript-Frameworks, wie das jQuery, eine große Hilfe für Designer. Diese Frameworks bestehen aus Ajax-Funktionen und können mit diversen Plug-Ins ausgebaut werden. Durch Ajax (Asynchrones JavaScript and XML) werden immer nur jene Teile neu geladen, die gerade Anwendung finden. Dies ist möglich, da die Daten asynchron zwischen dem Browser und Server übermittelt werden. (*Waschier, 2011*)

¹ <http://www.w3.org/standards/webdesign/htmlcss> (aufgerufen am, 18.01.2012)

² <http://www.grammiweb.de/informativ/grundlagen/wasistjava.shtml> (aufgerufen am, 18.01.2012)

2.3 CSS/CSS3

CSS (Cascading Stylesheet) ist eine Stilsprache die am häufigsten mit HTML- und XML-Dokumenten verwendet wird.³ Sie dient zum Auszeichnen strukturierter Dokumente, sodass die Inhalte des HTML-Dokumentes und das des CSS getrennt voneinander bearbeitet werden können (*Kramer, 2011*). Bei CSS Level 1 und 2 sind Formatierungen ausführbar, wie zum Beispiel das Setzen von Farben und Schriften, und das Erstellen des Layouts. Die Elemente konnten jedoch nur statisch angeordnet werden. Das hat sich bei dem, noch in Entwicklung befindenden CSS3, deutlich geändert. Mit der dritten Version ist es nun möglich Animationen, Farbverläufe, Schatten, runde Ecken, etc. darzustellen. Dadurch wird die Umsetzung von Layouts und Aktionen, an denen früher lange daran gearbeitet werden musste, um einiges einfacher. (*Kramer, 2011*)

2.4 Adobe Flash

Flash ist eine Entwicklungsumgebung in der Spiele, Animationen, Präsentationen, Banner, etc., also multimediale Inhalte, produziert werden.⁴ Die aktuelle Version dieses Programmes ist „Adobe Flash CS5.5“. Um diese Flash-Filme ansehen zu können, wird ein Flash-Player benötigt, jener als Plug-In kostenlos erwerblich ist und immer wieder aktualisiert werden kann (*Weschkalnies, 2010, S. 29*). Die Filme können im Internet angesehen werden, wenn sie auf einem Webserver gespeichert und in ein Dokument wie zum Beispiel HTML oder JavaScript eingebunden werden. In Flash können Bilder, Texte, Audio und Videos vereint werden. Es wird jedoch hauptsächlich für Vektorgrafiken und Animationen verwendet, da dies einem geringeren Speicher bedarf (*Weschkalnies, 2010, S. 31f*).

³ <http://www.css.nu/was-ist-css.html> (aufgerufen am, 18.01.2012)

⁴ Christiane Siedler, Lektorat Galileo Design aus dem Buch: Adobe Flash CS5, das umfassende Handbuch

3 HTML5

Obwohl HTML5 noch ziemlich neu ist, entwickelt es sich mit rasanter Geschwindigkeit in Richtung Interaktivität, 3D und Multimedia (*Kaltschmidt, 2011, S. 29*). Was früher noch als Markup Language galt, ist heute eher eine Aggregation verschiedener Webprozeduren. Deshalb beinhaltet HTML5 auch verschiedene JavaScript-Schnittstellen und zählt CSS3 zu seinem Bereich. Natürlich ist HTML5 trotzdem noch eine Sprache um Dokumente zu gliedern, nur ist es um einiges vielfältiger geworden (*Braun, 2011, S. 7*). In diesem Kapitel wird geschildert, welche Möglichkeiten es bereits mit HTML5 gibt und inwieweit die neue Technologie an Flash grenzt.

3.1 Video

Der Weg damit Videos in Browser eingebunden und abgespielt werden können, läuft nicht ganz unkompliziert und einwandfrei ab. Mit dem `<video>`-Tag ist es HTML5 zwar gelungen dies zu simplifizieren und zu verbessern, doch sind hier verschiedene Videocodecs im Umlauf, dessen Standard noch nicht deutlich ist (*Zota, 2011, S. 13*). Zurzeit sind drei verschiedene Formate im Umlauf, welche hier erwähnenswert sind. Zum einen gibt es das Format Ogg/Theora, wovor allerdings aufgrund der U-Boot-Patente abgeraten wird. Des Weiteren hat Google, durch das Open Web Media Project (WebM), den VP-8-Codec entwickelt, jener als Open Source bereitsteht. VP-8 hat eine große Ähnlichkeit mit dem dritten Format H.264 /MPEG 4 AVC (*Weschkalnies, 2010, S. 36*). Um die Videodateien beim Öffnen der Internetseite selbstständig abspielen zu lassen, müssten die Filme in allen drei Codecs eingebunden werden (*Zota, 2011, S. 13*), da nicht jeder Browser dasselbe Format unterstützt. In der nachfolgenden Tabelle wird aufgeführt, welcher Browser nun welchen Video-Codec lesen kann:⁵

Format	IE	Firefox	Opera	Chrome	Safari
Ogg (Theora)	-	3.5+	10.5+	5.0+	-
MPEG 4 (H.264)	9.0+	-	-	5.0+	3.0+
WebM (VP-8)	-	4.0+	10.6+	6.0+	-

Abbildung 1: Unterstützung der Browser für Videoformate

Außerdem ist dies notwendig, da es etwas unübersichtlich ist, Videos mit Flash auf mobilen Endgeräten abzuspielen, da jedes Gerät über einen anderen Funktionsumfang verfügt. Der

⁵ http://www.w3schools.com/html5/html5_video.asp (aufgerufen am, 18.01.2012)

Spitzenreiter auf Smartphones ist vorübergehend H.264 (Zota, 2011, S. 13). Der VP-8-Codec ist bisher nur in Opera Mobile⁶ und dem Androidbrowser 2.3.3+⁷ integriert. Die vorinstallierten Browser des iPhones (iOS) lehnen Flash gänzlich ab, sowie die älteren Browser von Android, weswegen die Einbindung per <video>-Tag eine relevante Bedeutung hat. Bei den neueren Androidgeräten ist Flash zwar möglich, aber nur mit eingeschränkter Leistung. Zum Beispiel benötigen etliche Spiele in Flash längere Ladezeiten, wofür es aber Applikationen als Alternative gibt, in denen der Bildaufbau deutlich zügiger vorangeht. Auch bei dem Windows Phone 7 und Blackberry OS ist nach wie vor kein Flash-Plug-In verfügbar. Obwohl HTML5 schon von fast allen modernen Smartphones unterstützt wird, gibt es auch bei dieser Technologie Komplikationen. Viele Geräte sind noch nicht so fortgeschritten um die erforderliche Leistung zu erzielen, die nötig ist, um HTML5-Animationen und -Interaktionen einwandfrei darzustellen. Es hängt nicht immer von der Browserkompatibilität ab, sondern auch von der Auflösung des Displays oder ob genügend RAM vorhanden ist. Die Hardware muss also leistungsfähig genug sein, außerdem sollten die Bedienelemente und Grafikausgabe für HTML5 ausgerichtet sein.⁸

Videos in Websites einzubinden ist derzeit noch etwas mühselig und umstritten. Um den Film überall richtig darzustellen, sollten alle drei Formate herangezogen und einzeln eingebettet werden. Um den Usern alternative Codecs anbieten zu können, wurde zusätzlich zu <video>, das <source>-Element entwickelt. Dieser Tag ermöglicht es, alle drei Formate einzubinden. Erkennt ein Browser einen Codec nicht, lässt er diesen Typ einfach aus und bearbeitet den nächsten. Außerdem kann auch gegenwärtig noch nicht komplett auf Flash vergessen werden. Bei nicht aktuellen Versionen von Browsern, funktioniert dieser Vorgang am unproblematischsten, wenn zusätzlich SWF-Dateien mit dem <object>-Tag eingebettet werden. Ältere Browser umgehen die ihnen fremden Elemente und springen direkt auf das ihnen Bekannte, das sie dann anzeigen können. (Zota, 2011, S. 13f). Das unten stehende Beispiel zeigt, wie am besten alle Formate eingebunden werden. Wird nur HTML5 verwendet, so fällt der Teil ab <object> weg:

```
<video width="320" height="240" controls="controls">
  <source src="film.mp4" type="video/mp4" />
  <source src="film.ogv" type="video/ogg" />
  <source src="film.webm" type="video/webm" />
  <object width="320" height="240" type="application/x-shockwave-flash" data="player.swf">
    <param name="movie" value="player.swf"/>
    <param name="flashvars" value="autostart=trues&controlbar=overs&file=film.mp4"/>
    
  </object>
</video>
```

Listing 3-1: Videoeinbindung

⁶ <http://dev.opera.com/articles/view/opera-supports-webm-video/> (aufgerufen am, 18.01.2012)

⁷ <http://developer.android.com/guide/appendix/media-formats.html> (aufgerufen am, 18.01.2012)

⁸ <http://www.teltarif.de/html5-flash-browser-vergleich/news/42443.html> (aufgerufen am, 18.01.2012)

Im nächsten Abschnitt wird inszeniert, welche Möglichkeiten es gibt, um Flash in HTML5 umzuwandeln oder Animationen direkt mit Anwendungen zu verwirklichen, diese HTML5-, CSS3- und JavaScript-Code ausgeben.

3.2 Programme für HTML5-Animationen

Nachdem bei HTML5 trotzdem noch Code geschrieben werden muss, haben einige Firmen Applikationen entwickelt, jene sehr einfach und ähnlich wie Flash zu bedienen sind und automatisch in den richtigen HTML5-, CSS3- und JavaScript-Code umgewandelt werden. Diese Anwendungen werden Authoring-Tool genannt. Außerdem werden mittlerweile sogenannte „Flash-to-HTML5 Converter“ angeboten, welche es ermöglichen FLA-Dateien in den dafür notwendigen HTML5-, CSS3- und JavaScript-Code zu konvertieren.

3.2.1 Authoring Tool

Die neue Applikation „Hype“ von Tumult, ist ein für Mac OS X generiertes HTML5-Authoring-Tool das im Gegensatz zu dem kostenintensiven Programm Adobe Flash sehr preiswert ist. Die Handhabung vom Hype ist simple, sodass sich visuelle Effekte und Animationen ohne fortgeschrittene Programmierkenntnisse bilden lassen.⁹ Wo bisher der Source Code für HTML5-Animationen manuell per Hand geschrieben werden musste, gibt es nun die Möglichkeit, die Komplexität der Programmiersprachen zu umgehen. Der Aufbau von Hype ist Flash sehr ähnlich, so dass auch hier mit Szenen und Zeitleisten gearbeitet wird, Keyframes angelegt und die Kondition der Button abgestimmt werden können. Die nachfolgende Abbildung zeigt die Ähnlichkeit von Hype und Flash CS5:

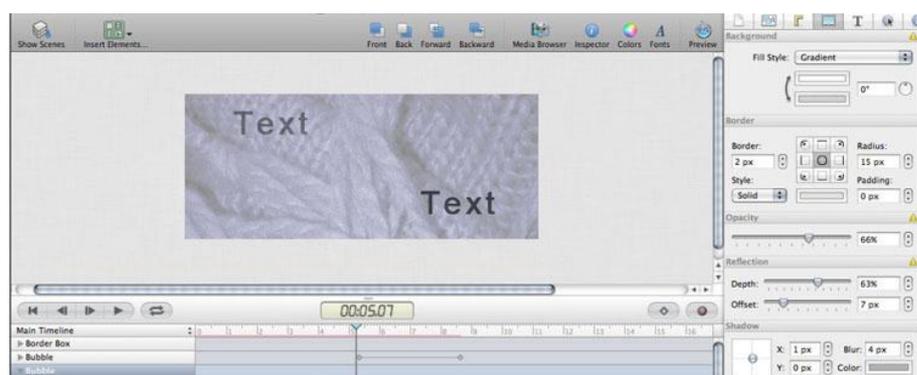


Abbildung 2: Benutzeroberfläche "Hype"

⁹ <http://tumultco.com/hype/documentation/overview/#hypeforhtml5> (aufgerufen am, 18.01.2012)

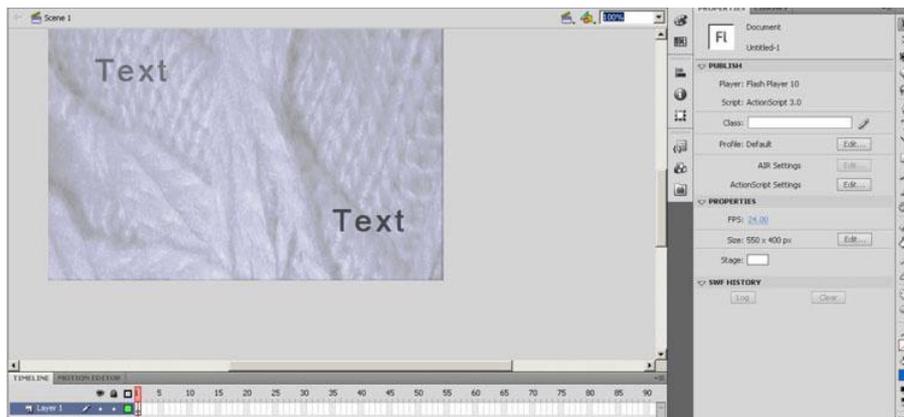


Abbildung 3: Benutzeroberfläche "Flash CS5"

Bei Hype ergibt sich lediglich der Vorteil, dass gewissermaßen kein Code geschrieben werden muss, sondern in Form eines WYSIWYG-Editors, die Elemente gestaltet und animiert werden. Zum Beispiel ist dafür eigens eine Liste vorhanden, in dieser diverse CSS-Eigenschaften wie `border-radius` oder `shadow-color` verfügbar sind.¹⁰ Trotzdem verfügt Flash über viele Tools und Effekte, die in Hype nicht integriert sind, da diese Funktionen von HTML5 nicht unterstützt werden. Für das exportieren der Animation werden die Komponenten in HTML5-, CSS3- und JavaScript-Code umgewandelt und ausgegeben. Hierfür wird direkt vom Programm ein geeigneter Ordner, samt der dazugehörigen Grafiken erstellt. Es folgen drei Zeilen Code um auf das entsprechende JavaScript zu verweisen und die Animation in die Website einzubinden.¹¹

Natürlich gibt es mehrere Firmen die ein Authoring Tool entwickelt haben. Zum einem ist da der Sencha Animator, der von Windows, ab XP, Mac OS X 10.6 und auch von diversen Linux-Distributionen unterstützt wird. Der Sencha Animator ist etwas teurer als Tumults Hype, wird er jedoch mit Flash verglichen, ist das Tool recht preiswert. Die Animationen lassen sich auf allen modernen Browsern abspielen, in denen CSS3 Animationen unterstützt werden. Allerdings ist die sichere Variante jene, WebKit Browser zu verwenden. Der Aufbau des Sencha Animators ist auch hier Flash sehr ähnlich und konzentriert sich auf die Zeitleiste. Komfortabel ist bei dieser Applikation, dass sich der Quellcode direkt im Programm bearbeiten lässt. Externe Dateien lassen sich mühelos importieren und werden dann, im von Sencha Animator erstellten Projekt-Ordner, gespeichert.¹² Im Projekt liegen ausschließlich ORMMA-kompatible Dateien, damit alle großen Mobilfunknetze unterstützt werden.¹³

¹⁰ <http://html5tutorial.de/html5-entwicklungsumgebung-hype-konkurrenz-fur-flash/> (aufgerufen am, 18.01.2012)

¹¹ <http://tumultco.com/hype/documentation/overview/#hypeforhtml5> (aufgerufen am, 18.01.2012)

¹² <http://www.netzwelt.de/news/88844-sencha-animator-flash-ersetzt-html-5.html> (aufgerufen am, 18.01.2012)

¹³ <http://www.sencha.com/products/animator/> (aufgerufen am, 18.01.2012)

Zum anderen hat auch Adobe ein Authoring Tool konzipiert, das als „Edge“ bezeichnet wird. Edge ist jedoch noch im Anfangsstadium und funktioniert den vorhin erwähnten Anwendungen ziemlich ähnlich. Ein positiver Aspekt von Edge ist, dass es mit SVG-Grafiken arbeitet, und somit stets scharf abgebildet wird (Kaltschmidt, 2011, S.31).

3.2.2 Flash-to-HTML5 Converter

Adobe entwickelte eine kostenlose Adobe-Air-Anwendung, das sich „Wallaby“ nennt, um FLA-Dateien in HTML5, CSS3 und JavaScript konvertieren zu können. Der Grund weshalb Adobe die Entwicklung von HTML5 so wichtig ist, ist das auch Banner und jene Websites die überwiegend Flash beinhalten, auf iOS-Geräten dargestellt werden können. Außerdem müssen Flash-Designer mit Wallaby keine zweite Lösung finden, um die Animationen auf allen Geräten wiedergeben zu können. Dabei wird aber hauptsächlich auf Funktionen geachtet, die in WebKitbasierenden Browsern funktionieren. In anderen Browsern wird die Animation also nicht immer richtig oder überhaupt nicht dargestellt. Wallaby, ist wie HTML5 noch in der Entwicklungsphase, weshalb nicht alle Flash-Elemente in HTML5 konvertiert werden können. Zudem kann auch ActionScript nicht in JavaScript umgewandelt werden (Gilbertson, 2011). Die Handhabung dieser Anwendung ist sehr einfach gehalten. Es muss lediglich der Pfad der zu konvertierenden Datei angegeben werden.

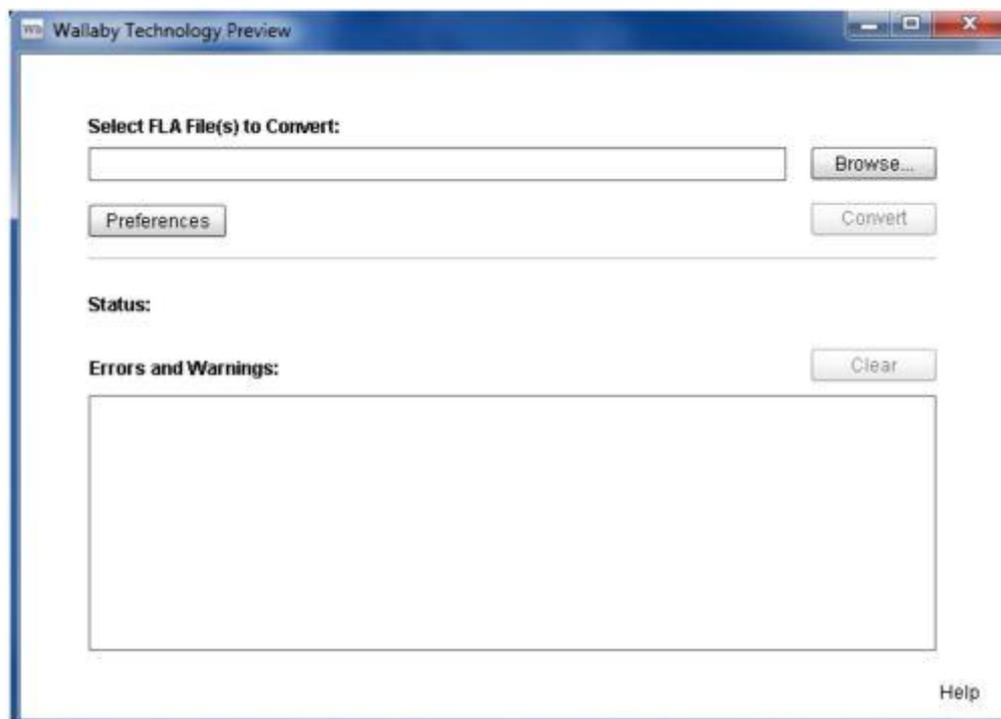


Abbildung 4: Wallaby

Die Updates von Wallaby sind jedoch knapp, denn es gibt seit März 2011 keine mehr. Da die Konkurrenz und Weiterentwicklung im Webbereich ziemlich hoch ist, hat Google bereits einen weiteren Flash-to-HTML5 Converter „Swiffy“ auf den Markt gebracht. Anders als

Wallaby wird hier nicht die FLA-Datei konvertiert, sondern es werden SWF-Formate in HTML5 umgewandelt. Auch hier arbeitet der Konverter am besten mit Safari und Google Chrome. Firefox hingegen stellt die Elemente teilweise richtig dar (Kaltschmidt, 2011). Swiffy steht ebenfalls noch in Anfangszeiten, jedoch werden Interaktionen und Animationen bereits einwandfrei angezeigt. Es wird eine kompakte JSON-Darstellung von der Animation benutzt, diese dann mit SVG, etwas HTML5 und CSS3 gerendert wird. Das JSON-Objekt kann auch ActionScript 2.0 einschließen und im Browser richtig reproduziert werden. Die Dateigröße verändert sich kaum und der Qualitätsunterschied ist gering (Gordon, 2011).

3.3 Vor- und Nachteile gegenüber Flash

Nahezu alle die mit Webdesign zu tun haben meinen, dass die größte Problematik von HTML5 jene sei, dass die Browser die Elemente unterschiedlich anzeigen. HTML5 ist noch kein fertiger Standard, sondern entwickelt sich stetig weiter, und somit können die neuesten Features nicht auf allen Browsern funktionieren, sondern nur auf den modernsten. Herbert Braun vom „Magazin c't extra Webdesign“ formuliert das so:

„Wenn eine Webtechnik auf allen Browsern funktioniert, ist sie veraltet.“

Grundsätzlich sollte es doch kein Problem darstellen, ein Update oder einen neuen Browser zu downloaden. Sehr gute Browser wie Mozilla Firefox, Google Chrome oder Opera sind kostenlos erhältlich und stets auf dem neuesten Stand. Solange die Benutzerrate des Internet Explorers bis Version 8 noch so hoch ist, stellt das durchaus ein Problem dar. Doch Designer und Entwickler müssen ihre Websites trotzdem auf den einzelnen Browsern überprüfen, bis sie einheitlich sind und auch denselben Effekt erzielen (Kaltschmidt, 2011, S. 31). Bei Flash fällt dieses Problem weg, denn die Animationen sehen, sofern ein Plug-In installiert ist, überall identisch aus. Nichtsdestotrotz muss auch hier ein Flash-Plug-In eingebaut und Updates durchgeführt werden, die aber ebenfalls kostenlos sind. Im Prinzip wird entweder ein Flash-Plug-In oder ein moderner Browser benötigt. Die Entscheidung, welche Technologie verwendet wird, liegt also nicht beim Entwickler der Website, sondern an den Usern. In diesen Fall ist es sicherlich am vernünftigsten, wenn Designer und Programmierer beide Methoden in ihren Internetauftritt einbauen.

Zudem kann das Einbinden der Animation ziemlich mühsam sein. Mit dem <video>-Tag ist es zwar einfach geworden, Videos einzubinden, nur bei den vielen verschiedenen Codecs müssen alle Formate und zusätzlich das Flash eingebettet werden, um auf allen Browsern richtig dargestellt zu werden. (Zota, 2011, S. 13)

Die Authoring-Tools scheinen dafür eine gute Lösung zu sein. Im Unterschied zu Adobe Flash sind sie um einiges kostengünstiger und lassen sich einfach bedienen. Durch

<video> ist es sehr unkompliziert die Animationen anschließend einzubinden. Programmierkenntnisse werden durch das Authoring-Tool nicht benötigt, doch die Möglichkeiten sind begrenzt. Wo Flash unzählige Effekte bereitstellt, können die Applikationen nur mit jenen dienen, die auch mit HTML5, CSS3 und JavaScript umsetzbar sind. Bei aufwendigen Animationen wird es außerdem schwierig sein, ganz ohne Programmierung zum erwünschten Ziel zu gelangen (*Kaltschmidt, 2011, S. 31*). Des Weiteren sind Animationen ohne Programmierung auch in Flash lösbar. Die Software bietet zahlreiche Tools, mit denen auch anspruchsvolle Arbeiten geleistet werden können, welche mit geringem Aufwand, einfach umzusetzen sind. Natürlich muss für komplexe Ausführungen auf ActionScript zurückgegriffen werden, jene Sprache sehr schwer zu erlernen ist.¹⁴ Diese Programmiersprache ist aber außergewöhnlich facettenreich und professionell. Es gibt darüber unzählige Dokumentationen, wo deren grafische Möglichkeiten aufgezeigt werden. Eine Einschränkung in Flash ist ohne Zweifel, dass nur Adobe das Programm weiterentwickeln und testen kann.

Die Aussage, dass Flash viel mehr Strom verbraucht als HTML5 ist sehr umstritten. Es stimmt das Flash-Animationen reichhaltig Performance benötigen, jedoch brauchen Animationen derselben Konzeption von HTML5 genauso viel. Es liegt an der Anwendung und den immer aufwendigeren Animationen, also nicht an der Technik des Programms selbst. In der Tat ist das ein Problem für Firmen, die einen so großen Gefallen daran finden, anspruchsvolle Websites mit vielen animierten Elementen online zu stellen, denn der Kunde ist heutzutage die langen Wartezeiten leid.¹⁵ Laut dem Test von Jan Ozer, hängt der Stromverbrauch von Flash davon ab, ob die Hardwarebeschleunigung eingesetzt werden kann oder nicht. Flash 10.1 lässt das Plug-In auf den Grafikprozessor zuzugreifen, die Version 10.0 jedoch nicht. Größtenteils hängt das Performanceproblem also vom Betriebssystem und den verwendeten Browser ab (*Ozer, 2011*).

HTML5 ist noch überaus innovativ, was aber nicht heißen soll, dass Flash besser ist. In den letzten Jahren, hat sich HTML5 so schnell weiterentwickelt, dass es durchaus vorstellbar ist schon bald, Flash gleichgestellt zu sein.

¹⁴ <http://html5tutorial.de/html5-oder-flash-was-ist-besser-was-setzt-sich-durch/>
(aufgerufen am, 18.01.2012)

¹⁵ <http://praegnanz.de/weblog/neue-wege-zur-erzeugung-von-html5-animationen>
(aufgerufen am, 18.01.2012)

4 JavaScript

Unter HTML5 ergeben sich unterschiedliche Vorstellungen, da diese Sprache nichts mit HTML zu tun hat, sondern diverse und moderne Webtechniken umfasst. Besonders JavaScript gehört hier zu den führenden Techniken, diese im gesamten Bereich Webdesign, zu der am fortschrittlichsten Sprache zählt. Zum einen zählen an dieser Stelle die JavaScript APIs wie Canvas dazu, zum anderen nach wie vor Frameworks wie jQuery (*Braun, 2011, S. 20*).

4.1 AJAX und Frameworks

Animationen basieren alle auf dem gleichen Prinzip: Es werden verschiedene Bilder in einer schnellen Abfolge so geändert, dass eine Bewegung entsteht, sowie sie schon aus Zeichentrickfilmen bekannt sind. Auch die Animationen von JavaScript beruhen auf dieser Funktionsweise. Schon zur Anfangszeit von dieser Sprache konnten mit `setTimeout()` Bewegungen programmiert werden. Die Intervalle, in denen sich die einzelnen Zustände ändern sollen, sowie die Zeit in der die Animation laufen soll, können in dieser Funktion festgelegt werden. Der Vorgang wird sofort neu geladen, bis die Animation abgeschlossen ist. Das hat nun die Möglichkeit geboten, Schriften und einfache HTML-Objekte dynamisch zu ändern. Die Größe, Farbe und Position konnte von den Elementen verändert werden, sodass zum Beispiel ein Lauftext oder ein Ladebalken codiert werden konnte (*Nicholas, 2011*). Im Jahr 1996¹⁶, kam das Image-Objekt das erste Mal, in einer Beta-Version des Netscape Navigator 3.0 vor. Es konnten nun endlich Grafiken mit JavaScript angesprochen, ausgetauscht, animiert und grafische Effekte in Bildern angebracht werden. Im Grunde genommen funktionierte die Animation, nicht anders als bei der Laufschrift, nur mit codierten Elementen (*Wenz, 2010, S. 309*).

Nachdem JavaScript eine stark clientseitige Programmiersprache ist, bietet sie eine reichhaltige Interaktivität, insbesondere durch die Einführung von asynchronen HTTP-Anfragen (AJAX), an. Es müssen dadurch Teile von Seiten nicht länger neu geladen werden, sondern Daten können anhand des XMLHttpRequest-Objektes asynchron mit dem Webserver ausgetauscht werden. Um aber nicht immer mühsam den langen JavaScript-Code schreiben zu müssen, haben Programmierer nun Bibliotheken, oder sogenannte Frameworks, entwickelt. Das sind Sammlungen von Funktionen und Anwendungsprogrammen, die das Programmieren von JavaScript und das Bearbeiten von DOM-Objekten um einiges kürzer und leichter verständlich machen (*Lennon, 2010*). Die wohl beliebtesten JavaScript-Frameworks sind jQuery, MooTools und Prototype. Die Statistik von

¹⁶ <http://sillydog.org/netscape/verinfo.php> (aufgerufen am, 18.01.2012)

W³Tech besagt, dass jQuery mit 83,3% Marktanteil, das mit Abstand am häufigsten verwendete Framework ist, weswegen in der Abhandlung nur diese Bibliothek erörtert wird.¹⁷

jQuery ist ein leicht erlernbares und verständliches Framework, dessen Syntax mit jener von CSS relativ vergleichbar ist. Es gibt dafür unzählige Plug-Ins, mit denen sehr schöne und anspruchsvolle Animationen erzeugt werden können. Aber auch ohne Erweiterungen beherrscht dieses Framework recht stilvolle Effekte. Einfache Funktionen wie `fadeIn()` oder `toggle()` können zum Beispiel bei der Navigation verwendet werden, dass die Wirkung ästhetisch steigert. Die Funktion `animate()` ist imstande, sehr niveauvolle Animationen herzustellen. Es können dabei grundlegende Eigenschaften festgelegt werden, um auch Emotionen darzustellen. Mit `duration` wird beschrieben wie lange die Bewegung andauern soll, wobei `complete` die Prozedur beendet. Die Beschleunigung wird mit der Funktion `easing` bestimmt, diese nun eine Animation realitätsgetreu aufbaut. Damit die Animation nach jeden Schritt aufrufbar ist, gibt es bei `animate()` auch noch die Eigenschaft `step`. Mit `queue` kann bestimmt werden, ob die Animation sofort beginnen, oder eine Verzögerung aufweisen soll. Für all das müssen in der Regel nur das Framework in das HTML-Dokument eingebaut und ein paar Zeilen Code geschrieben werden.¹⁸

4.2 Canvas und SVG

Im deutschen Sprachraum bedeutet Canvas „Leinwand“. Jedoch ist hier nicht die Leinwand gemeint, auf der Maler mit Ölfarben ihr Werk vollbringen, sondern ein neuer HTML5-Tag. Das `<canvas>`-Tag wird in diesem Kapitel berücksichtigt, weil es aus vielen JavaScript-Funktionen besteht. Dieses Element stellt uns eine Fläche zur Verfügung, auf der, mithilfe von JavaScript-Funktionen, Geometrische Formen und Pixelgrafiken gezeichnet und bemalt werden können. Alle aktuellen Browser können dieses neue Element darstellen und interaktive, dynamische Grafiken anzeigen (*Lau, 2011, S.39*). Eigentlich hat das Canvas-Element nur 2D-Objekte vorgesehen, dennoch wurde mit JavaScript eine 3D-Engine konzipiert, um 2D-Elemente zu dreidimensionalen Objekten transformieren (*Schulz, 2011, S. 76*). In dieser Arbeit werden diese Funktionen nicht weiter besprochen, sondern nur wie es dazu kommt Animationen und Interaktionen mit dem Canvas-Tag darstellen zu können.

Wie auch bei der realen Leinwand, wird beim `<canvas>`-Tag zuerst der Hintergrund bemalt und die vorderen Elemente erst später. Zuerst wird der 2D Kontext-Typ der Zeichenfläche mit der `getContext('2d')`- Methode spezifiziert, deren Ursprung die Koordinaten (0,0) sind und dessen x- und y-Werte sich erhöhen lassen.¹⁹ Danach könnte zum Beispiel im

¹⁷ http://w3techs.com/technologies/overview/javascript_library/all (aufgerufen am, 18.01.2012)

¹⁸ <http://api.jquery.com/animate/> (aufgerufen am, 18.01.2012)

¹⁹ <http://www.whatwg.org/specs/web-apps/current-work/multipage/the-canvas-element.html> (aufgerufen am, 18.01.2012)

Hintergrund mit `createLinearGradient()` ein Verlauf angebracht werden, wo als Parameter die Start- und Endkoordinaten übergeben werden. Mit der Funktion `addColorStop()` wird als erstes mit einer Fließkommazahl, von Null bis Eins, die Farbe an einem gewissen Punkt festgelegt. Mit dem zweiten Parameter wird die Farbe definiert und mit `fillStyle()` werden alle Objekte eingefärbt. Bei dieser Methode müssen auch die Höhe und Breite des Elementes bestimmt werden und sie dient zugleich zum verschieben und rotieren des Objektes. Ist der Hintergrund fertig erstellt, kann die nächste Schicht codiert und zum Beispiel Pfade gezeichnet werden, diese mit `beginPath()` gestartet und mit `closePath()` beendet werden. Dazwischen können mit `moveTo()` Formen und Grafiken gezeichnet werden. Diese Funktion hebt einen unsichtbaren Stift an, der an den Ausgangspunkt der nachfolgenden Aktion positioniert wird. `lineTo()` stellt vom angegebenen Ursprung eine Linie zu den übergebenen Koordinaten her. Es gibt auch Funktionen die direkt ein Rechteck (`rect`) oder einen Kreis (`arc`) herstellen. Im nachfolgenden Beispiel wird angeführt, wie ein gelber Kreis gezeichnet werden würde:

```
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript">
      function drawOnCanvas() {
        var canvas = document.getElementById("kreis");
        if (canvas.getContext){
          var kreis = canvas.getContext("2d");
          kreis.beginPath();
          kreis.fillStyle = "rgb(255 ,215 ,0)";
          kreis.arc(100, 100, 50, 0, 2 * Math.PI, true);
          kreis.fill();
        }
      }
    </script>
  </head>
  <body onLoad="drawOnCanvas()">
    <canvas id="kreis" width="200" height="200">
    </canvas>
  </body>
</html>
```

Listing 4-1: Canvas - Kreiszeichnung mit Pfad

Da die Elemente unmittelbar per Script auf der Leinwand kontrolliert werden, ist es mühelos die Objekte zu manipulieren und animieren. Canvas wurde nie für bewegte Bilder konzipiert, weshalb es auch Restriktionen gibt. So ist es nicht möglich dieselbe Form zu verändern, sondern die erstellte Form muss stets neu generiert und entfernt werden. Das bringt den Nachteil mit sich, dass umfassende Frames zeitaufwendig sind und die Leistung des Computers darunter leidet. Es gibt zwei Verfahren um mit Canvas Animationen herstellen zu können. Zum einen mit den üblichen `setTimeout()` und `setInterval()` Methoden. Diese Technik wird verwendet, wenn über einen festgelegten Zeitraum animiert werden soll

und außerdem keine Benutzerinteraktionen von Bedarf sind. `setInterval()` führt alle parametrisierenden Millisekunden die bestimmte Anweisung aus, diese mit `reDraw()` wieder gelöscht werden (*Braun, 2011, S. 46*). Ein kurzes Beispiel zeigt zur Übersicht, wie `setInterval()` funktioniert:

```
canvas = document.getElementById('canvas');
context2D = canvas.getContext('2d');
setInterval(draw, 1000 / FPS);
```

Listing 4-2: Canvas-Animation

Zum anderen gibt es die Verfahrensweise, diese für Animationen relevant sind, welche Eingaben von Anwendern erfordern. Mit dieser Methode können Spiele hergestellt werden, wo die Tastatur oder die Maus benötigt wird. Durch die Konfiguration des JavaScript-EventListeners können alle Interaktionen des Benutzers abgefangen und die Animation ausgeführt werden.²⁰

Eine wesentlich ältere, aber ähnliche Sprache wie das Canvas-Objekt, ist SVG (Scalable Vector Graphics). Dies ist eine Text-basierende Grafiksprache, die Bilder mit Vektorformen, Text und eingebettete Rastergrafiken beschreibt, wodurch die erstellten Objekte kompakt und qualitativ hochwertig sind.²¹ SVG basiert auf der Sprache XML und wird mit der Dateierdung `.svg` abgespeichert. Außerdem unterstützt diese Sprache auch Animationen, die sich mit dem Multimedia-Standard SIML (Synchronized Multimedia Integration Language) zeitlich steuern und wechselseitig austauschen lassen.

```
<rect x="100" y="100" width="100" height="100">
  <animate attributeName="x" to="1000" begin="mouseover" dur="5s"/>
</rect>
```

Listing 4-3: SVG

Im oberen Beispiel wird durch den `<rect>`-Tag ein Rechteck gestaltet, dessen Position und Größe in den Attributen festgelegt werden. Durch `animate attributeName="x"`, wird die Eigenschaft festgelegt, dass entlang der X-Achse animiert wird, sowie `to` den Bestimmungsort des Objektes beschreibt. Mit `begin` wird die Bedingung beschrieben, bei welchem Ereignis die Bewegung in Gang gesetzt werden soll. Hier können Mausereignisse eingesetzt werden, wohingegen Tastaturereignisse nicht zugelassen sind. Bei solchen Begebenheiten sollten Hyperlinks gesetzt werden. Um die Zeit festlegen zu können, in welcher Geschwindigkeit sich das Objekt bewegen soll, wird das Attribut `dur` verwendet. Mit JavaScript-Funktionen können SVG-Animationen durchaus ausgebaut werden (*Braun et. al., 2011, S.33*). Doch die Renderzeiten von SVG sind langwierig, denn alles was mit SVG erstellt wird, wird als DOM-Knoten interpretiert. Somit ist eine flüssige Animation, besonders

²⁰ https://developer.mozilla.org/en/Canvas_tutorial/Basic_animations (aufgerufen am, 18.01.2012)

²¹ <http://www.adobe.com/svg/> (aufgerufen am, 18.01.2012)

bei aufwendigen Projekten, nicht möglich. Im nachfolgenden Kapitel wird SVG nicht besprochen, da die schon etwas ältere Sprache für Bewegte Bilder im Web kein gutes Resultat erzielt.

4.3 Vor- und Nachteile gegenüber Flash

In diesem Kapitel werden die Vor- und Nachteile der vorher dargelegten JavaScript-Technologien kundgegeben. Die meisten JavaScript-Anwendungen sind kostenlos verfügbar und einfach zu bedienen. Trotzdem sind ihre Programme qualitativ hochwertig und im Source Code bequem zu bearbeiten. Dennoch sind die Funktionen, wie 3D oder das Einbetten von Schriften, begrenzt. Wenn komplexe Animationen nötig sind, kann es sein, dass diese nicht einwandfrei dargestellt werden können.

Einer der wohl positivsten Aspekte von JavaScript-Frameworks ist die Benutzerfreundlichkeit, da die Bedienung ziemlich einfach ist. Durch tausende von Tutorials, ist der Code schnell erlernbar, und unter Einsatz von reichlichen Plug-Ins ermöglichen es Frameworks, mit wenig Code ein ansehnliches Ergebnis zu erzielen. Durch den schlanken, übersichtlichen Code ist die Pflege des eigenen Scripts unkompliziert. Besonders die jQuery Bibliotheken sind sehr groß, und haben eine ausgesprochen detailreiche Dokumentation. Durch die ständigen Wartungen und Kontrollen, ist jedes Script überaus leistungsfähig, sicher und effizient.²² Sollten irgendwelche Schwachstellen entdeckt werden, müssen diese lediglich der Online-Community gemeldet werden, damit sie anschließend ausgebessert werden. Die Bibliotheken ermöglichen es mit Ajax Templates zu entwickeln, ohne die Seite neu laden zu müssen. Die meisten Frameworks werden kostenlos zum Download bereitgestellt und müssen mit nur einer Zeile Code eingebunden werden. Jedoch sind die Funktionalitäten begrenzt und in einigen Fällen muss auch rohes JavaScript gekonnt und verwendet werden. Dies ist schwieriger zu schreiben und der Code weitet sich somit aus, wodurch das Script unübersichtlicher wird. Die Dateigrößen der eingebundenen Files sind zwar nicht sonderlich beachtlich, dennoch sollte bedacht werden, dass es zu einer Belastung des Client-Computers und Webservers kommen kann. Dazu kommt noch, dass alle verschiedenen Frameworks ihre eigenen Regeln haben, diese beherrscht werden sollten. Soll nun von der einen Bibliothek zu einer anderen gewechselt werden, muss die gesamte Website umgeändert und angepasst werden. Es wird außerdem Vorsicht erfordert, da Open Source-Programme recht einfach zugänglich sind und Hacker das Script ändern können.²³

Das große Defizit bei Canvas ist, dass die erstellten Elemente im Nachhinein nicht mehr manipuliert werden können. Animationen können nur erfolgen, wenn die einzelnen Frames

²² <http://www.jscripters.com/jquery-disadvantages-and-advantages/> (aufgerufen am, 18.01.2012)

²³ <http://computersight.com/programming/php/advantages-and-disadvantages-of-framework/> (aufgerufen am, 18.01.2012)

durch die Funktion `setInterval()` immer wieder neu gezeichnet werden müssen. Außerdem bestehen die von Canvas erstellten Elemente, aus Pixelgrafiken, das die Performance jedes Bildaufbaus erheblich steigert, weshalb bei größeren Projekten Canvas wohl gemieden werden sollte. Bei modernen und leistungsfähigen Rechnern ist dennoch eine flüssige Wiedergabe realisierbar. Unter anderem kann die elektronische Leinwand mit JavaScript erweitert werden, das die Möglichkeiten der Animation erheblich verbessert. Die Browserabhängigkeit des neuen Tags ist ein weiterer Nachteil. Ein ziemlich großer Teil der Internetnutzer in Österreich verwendet ausschließlich Versionen des Internet Explorers, von diesen Canvas aber nicht unterstützt wird. Trotzdem sollte nicht vergessen werden, dass Canvas kein Plug-In benötigt und somit direkt, einfach und systemnah in den Code der Website eingebaut wird. Die konstruierten Elemente, Bilder und Videos können komfortabel verändert werden. Sehr beliebt ist das `<canvas>`-Tag bei mobilen Geräten, wo wenige davon Flash darstellen können, doch die meisten davon HTML5 und JavaScript. In einigen Bereichen ist Canvas so weit entwickelt, dass es problemlos Bereiche von Flash adoptieren kann. Das freut nicht nur Webdesigner, sondern auch Kunden, dessen wesentliche Ansprüche wenig Zeit und niedrige Kosten sind. Mit dem neuen Tool werden keine externen Programmierer benötigt, die die komplizierte Sprache beherrschen, sondern fast jeder Webdesigner beherrscht die Sprache JavaScript und kann kleine Animationen mitmachen. Canvas ist eine noch sehr frische, aber rasch fortschrittliche Technologie. Trotzdem wird es sich, zumindest in den nächsten Jahren, nicht gegen Flash durchsetzen können, da viele Effekte und ausschlaggebende Tools fehlen. Wohl eher ist Canvas eine Assistenz, um Webdesign dynamischer gestalten zu können. Die überwiegenden Einsatzgebiete werden vermutlich animierte Hintergründe und Banner sein (Michel, 2010).

5 CSS3

Auch die Funktionen der CSS3-Animationen sind immer noch in der Entwicklungsphase. Trotzdem kann schon sehr viel damit realisiert werden. In den nachfolgenden Seiten wird erklärt wie diese Animationen zustande kommen und was an ihnen beachtet werden muss.

5.1 Herstellerspezifische Präfixe

Nachdem die verschiedenen Browser ständig weiterentwickelt werden und neue Versionen erscheinen, wird es oft schwierig ältere Websites auf modernen Browsern wiederzugeben. Oft werden die früheren Eigenschaften nicht mehr unterstützt, sondern nur noch die Neuen. Da CSS3 immer noch ausgebaut wird, werden versuchsweise vor vielen neuen Eigenschaften Vorsilben, die sogenannten Präfixe, verwendet. Das Präfix wird ausschließlich vom betreffenden Browser gelesen und angewendet. In der folgenden Tabelle werden die am häufigsten verwendeten Präfixe aufgelistet.

Herstellerspezifisches Präfix	Browsergültigkeit
-moz-	Gecko-basierte Browser wie Firefox
-o-	Opera
-webkit-	Safari, Google Chrome und andere WebKit-basierte Browser
-ms-	Internet Explorer

Abbildung 5: Präfixe und ihre Browsergültigkeit

Es wird allerdings von Präfixen abgeraten, da sie nicht dauerhaft Bestand haben werden. Die alten Eigenschaften werden durch standardisierte Attribute ausgetauscht und verlieren so ihre Gültigkeit (*Maurice, 2011, S.22f*). Da aber die nachfolgend geschilderten Funktionalitäten von CSS3 relativ neuartig sind, müssen bei diesen immer die herstellerspezifischen Kennzeichnungen angegeben werden.

5.2 Animationseigenschaften

Um eine Animation in CSS3 durchführen zu können, ist es wichtig mehrere CSS3-Eigenschaften zu kennen. Mit den anschließend genannten Attributen, können einfache, sowie komplexe Bewegungen im Internet erstellt werden.

5.2.1 Transformationen

Schon mit einfachen Anweisungen kann CSS3 viel bewirken, um dynamische Websites zu gestalten. Es gibt an dieser Stelle auch Eigenschaften die keine direkte Bewegung aufzeigen, sondern lediglich ein Objekt statisch verändern. In dieser Arbeit sind diese Funktionen aber trotzdem sehr wichtig, da sie ein wesentlicher Bestandteil für die nachfolgenden Animationen sind. Mit `transform` und `perspective` können Elemente so transformiert werden, dass später nur noch der Anfang und das Ende festgelegt werden muss, um eine Bewegung des Objektes darstellen zu können (Maurice, 2011, S.186ff). Das ist möglich, da CSS ein Koordinatensystem darstellt, in dem die einzelnen Komponenten angeordnet sind. Mit den Transformations-Eigenschaften können die verschiedenen Glieder im Koordinatensystem verändert werden.²⁴

```
transform:rotate(45deg);  
transform:skew(45deg);  
transform:translate(60px, 70px);  
transform:scale(1.5);  
transform-origin:100% 100%;
```

Listing 5-1: Transform

Das obere Beispiel zeigt, dass sich ein Objekt mit den Eigenschaften `rotate`, `skew`, `scale` und `translate`, drehen, verschieben, skalieren und verzerren lässt. Mit `origin` lässt sich der Ankerpunkt festlegen, von welcher Position des Objektes die Eigenschaft angewendet werden soll (Maurice, 2011, S.186ff). Es konnten zwar schon mit CSS2 Objekte verschoben und transformiert werden, jedoch nicht in diesem Ausmaß. Außerdem konnten zu dieser Zeit noch keine Verzerrungen oder komplexe Änderungen vorgenommen werden (Braun, 2011, S.17).

Mit der CSS3-Eigenschaft `perspective` ist den Entwicklern ein gewaltiger Sprung vorwärts gelungen, da nun 3D-Transformationen auf Elemente angewendet werden können. Der Grundsatz von `perspective` ist derselbe wie bei der 2D-Transformationseigenschaft, nur dass es hier auch noch die z-Achse gibt, deren Werte verändert werden (Maurice, 2011, S.202ff).

5.2.2 Animationsverhalten

Im vorigen Absatz ist nun geschildert worden, wie Objekte entsprechend statisch verändert werden können. Um jedoch eine Bewegung in das Objekt einzubringen, müssen einige Funktionen hinzugefügt werden. Dieser Vorgang wird im nächsten Abschnitt beschrieben.

²⁴ <http://www.w3.org/TR/css3-2d-transforms/> (aufgerufen am, 18.01.2012)

5.2.2.1 Transition

Mit der Eigenschaft `transition` besteht die Möglichkeit, CSS3 Effekte zu gestalten, die bisher nur mit Flash oder JavaScript realisierbar waren. Zum Beispiel kann dieses Attribut Elemente, mit einem fließenden Übergang, ein- und ausblenden lassen. Diese Eigenschaft hat viele Funktionen, so kann mit `transition-duration` die Zeit bestimmt werden, jene die Animation beanspruchen darf. Mit `transition-delay` wird eine Verzögerung festgelegt, die Bewegung beginnt dann erst nach einer definierten Zeit, oder sofort. Beide Werte werden in Sekunden oder Millisekunden angegeben. Der Standardwert für Beschleunigungen ist `ease`, durch `ease-in` wird eine Beschleunigung am Anfang, oder `ease-out`, eine Geschwindigkeitsverringerung am Ende definiert. Zudem wird mit `linear` hingegen, wird ein regelmäßiger Verlauf bestimmt. (Maurice, 2011, S.212ff)

```
#uebergang li{
  color:blue;
  font-size:150%;
  -moz-transition: all 1s ease-in;
  -webkit-transition: all 1s ease-in;
  -o-transition: all 1s ease-in;
  -ms-transition: all 1s ease-in;
}
#uebergang li:hover{
  padding-left:10px;
  color:green;
}
```

Listing 5-2: Transition

Das obige Beispiel zeigt, wie sich ein HTML-Element bei Antreffen der Maus, in einer Sekunde um zehn Pixel nach links verschiebt. Der Effekt `ease-in` drückt aus, dass das Objekt langsam beginnt und bis zum Ende schneller wird. Mit den herstellerspezifischen Präfixen, wird die Animation in jeden Browser dargestellt.

5.2.2.2 Keyframes

Um eine Bewegung auf der Website darstellen zu können, müssen zumindest die Start- und Endwerte bekannt sein. Wird eine genauere Kontrolle der Zwischenwerte erwünscht, können jeweils die einzelnen Schlüsselbilder definiert werden.

Keyframes werden mit einer bestimmten CSS-Regel definiert, wobei das Schlüsselwort `@keyframes` lautet. Anschließend wird die Animation mit `animation-name` benannt und innerhalb von geschweiften Klammern die Style-Regeln festgelegt. Bei den einzelnen Schlüsselbildern müssen anhand von Keyframe-Selektoren die Start- und Endwerte bestimmt werden. Die Selektoren werden in Prozenten angegeben, wobei 0% das Startbild und 100% das Endbild enthält. Diese zwei Werte können auch mit `from` und `to` ersetzt

werden.²⁵ Mit `animation-timing-function` kann wie bei der `Transition`-Eigenschaft die Beschleunigung festgelegt werden. Auch die Werte sind dieselben, nämlich `ease`, `ease-in`, `ease-out` und `linear`. Mit `animation-duration` wird die Animationsdauer festgesetzt und `animation-iteration-count` bestimmt die Anzahl der Bewegungsdurchläufe. Wird hier der Wert `infinite` angegeben, läuft die Animation unendlich durch. Eine noch sehr den `transition` ähnliche Eigenschaft ist `animation-delay`, mit der sich eine Verzögerung initiieren lässt. Die Bewegung kann durch die Funktion `animation-direction` auch rückwärts abgespielt werden. Nachfolgend wird ein Beispiel gezeigt, indem sich ein Objekt zuerst bis zur Hälfte der Zeit um 100 Pixel nach links verschiebt. Bei 75% wird es wieder um zehn Pixel zurück, nach rechts gerückt und die restliche Zeit bis 200 Pixel nach links. Durch die Eigenschaft `infinite` wird dieser Vorgang, bis zum Beenden der Website wiederholt. Um diese Animation jedoch in jedem Browser anzeigen zu können, müssen die Keyframes mit den Präfixen einzeln geschrieben werden.

```
@-moz-keyframes bewegen{
  0% {left:0px;}
  50%{left:100px;}
  75%{left:90px;}
  100% {left:200px;}
}
@-webkit-keyframes bewegen{
  0% {left:0px;}
  50%{left:100px;}
  75%{left:90px;}
  100% {left:200px;}
}
@-ms-keyframes bewegen{
  0% {left:0px;}
  50%{left:100px;}
  75%{left:90px;}
  100% {left:200px;}
}
#objekt{
  animation:bewegen 5s infinite;
  -moz-animation:bewegen 5s infinite;
  -webkit-animation:bewegen 5s infinite;
  -ms-animation:bewegen 5s infinite;
}
```

Listing 5-3: CSS3 Keyframe-Animation

Wird nur der Parameter `animation` angegeben, kann damit die knappere Schreibweise herangezogen werden, bei der jedoch die anschließende Reihenfolge eingehalten werden muss (*Maurice, 2011, S.211ff*):

```
animation: name duration timing-function delay iteration-count direction
```

Listing 5-4: Animation-Kurzschreibweise

²⁵ <http://www.w3.org/TR/css3-animations/> (aufgerufen am, 18.01.2012)

5.3 Vor- und Nachteile gegenüber Adobe Flash

Die Vorteile liegen auf der Hand. Webdesigner benötigen keine Kenntnisse in der aufwendigen Sprache ActionScript, sondern können bereits beim Stylen der Website, Animationen gestalten. Die neuen Code-Blöcke sind nicht schwer zu erlernen und es kann mit ein paar Zeilen Code, eine Bewegung zustande gebracht werden. Es lassen sich dadurch schnell Änderungen, bei Fehler oder sonstigen Bedürfnissen, durchführen. Dies ist ein Gewinn sowohl für die Designer, als auch für den Kunden. Der Webauftritt ist schneller fertiggestellt und somit kostengünstiger. Außerdem sind alle Elemente für Suchmaschinen und Screenreader lesbar, das für die Wirtschaft eine große Bedeutung haben kann.

Die Kehrseite von CSS3-Animationen ist, dass sich die Browser sehr unterschiedlich auf diese Sprache auswirken. Nur neuere Browser sind für diese Sprache kompatibel und setzen sie einwandfrei um. Außerdem werden für viele Eigenschaften nach wie vor die herstellerspezifischen Präfixe benötigt. Durch die Vorsilben, vereinfacht sich der Source Code keineswegs, sondern wird je Attribut um vier Zeilen länger (Maurice, 2011, S.23). Ist also viel CSS3-Code vorhanden, muss immer bedacht werden, dass ältere Browser, insbesondere der Internet Explorer 8 und Versionen darunter, das Layout nicht passend darstellen werden. Oft werden die neuen Funktionen überhaupt nicht dargestellt.

„Diese lästigen Vorspanne sind auch der Grund, dass die meisten Demos nur mit Chrome, Safari und allenfalls Firefox funktionieren – für andere Browser interessieren sich manche Webdesigner offenbar nicht mehr.“ (Braun, 2011, S.17)

Des Weiteren, kann `transition` kein Hintergrundbild in CSS3 animieren. Wird in einem DIV-Container ein Bild eingebunden, kann auf dieses problemlos die Transition-Eigenschaft angewendet werden. Soll aber ein Hintergrundbild direkt in CSS ein- oder ausgefadet werden, muss dafür auf JavaScript zurückgegriffen werden.²⁶

Gegenwärtig ist es im Grunde genommen ziemlich ausgeglichen, ob Flash oder CSS3 verwendet wird. Doch so schnell wie sich die Funktionen von CSS3 und den Browsern erweitern, ist es durchaus möglich, diese Sprache in Zukunft in sehr vielen Bereichen einsetzen zu können. Welche Eigenschaften in welchem Browser zur Verfügung stehen, wird im nächsten Kapitel dokumentiert.

²⁶ <http://www.thewhykiki.de/css3-hintergrundbild-animieren-buttons-mit-hintergrundbildern/>
(aufgerufen am, 18.01.2012)

6 Browserkompatibilität

Die Sprachen CSS3 und HTML5 sind noch im Anfangsstadium, weswegen sie immer noch ausgebaut werden. Laufend werden neue Features und Funktionen zur Verfügung gestellt. Die Browserhersteller können nicht andauernd alle neuen Eigenschaften in ihre Software einbauen, sondern versuchen möglichst viele Neuerungen unterzubringen. Somit hat jeder Browser seine Eigenheiten.

In den nachfolgenden Kapiteln werden die oben genannten Tools und Funktionen protokolliert, wie sie sich auf den verschiedenen Browsern auswirken. Bei HTML5 und vor allem CSS3 gibt es einige Unterschiede. Diese Unterschiede entfallen hingegen bei dem schon etwas ausgereifteren JavaScript, da die Sprache schon in nahezu allen Browsern implementiert ist. In Folge dessen, wird sich diese Arbeit, hauptsächlich auf HTML5 und CSS3 konzentrieren. Es wird sich hierbei nur auf die gängigsten Browser, wie Mozilla Firefox²⁷, Internet Explorer²⁸, Opera²⁹ und die die webKitbasierenden Google Chrome und Safari³⁰, näher eingegangen wird.

6.1 HTML5

Hier werden die vorher genannten HTML5-Tools und Elemente geschildert, unter anderem auch das `<canvas>`-Tag. Canvas wird in den Browsern recht gut dargestellt. Im Internet Explorer ist es ab der Version 9.0 implementiert, in den anderen Browsern, wie Firefox, Chrome, Safari und Opera, ist dieses Element schon in früheren Varianten verfügbar. Dasselbe gilt für das `<video>`-Tag, jene Kompatibilitäten bereits im Kapitel 3.1 genauer beschrieben wurden. Bei dem Flash to HTML5 Converter „Wallaby“ funktionieren die Animationen nur auf WebKit-basierenden Browsern einwandfrei. Die Filme werden in HTML5 und CSS3 umgewandelt, dessen Code nicht jeder Browser gleich lesen kann. Zum Beispiel können CSS3-Transformationen nur in WebKit-basierenden Browsern dargestellt werden, worauf im nächsten Abschnitt genauer eingegangen wird. „Swiffy“ hingegen basiert auf der SVG-Technik und wird von allen aktuellen Browsern unterstützt³¹, was bei dem Internet Explorer ab Version 9.0 der Fall ist. Bei den Authoring-Tools gibt es relativ wenig Probleme mit den verschiedenen Browsern. Sie werden schon von relativ alten, sowie allen aktuellen Browsern unterstützt.

²⁷ <https://developer.mozilla.org/de/> (aufgerufen am, 18.01.2012)

²⁸ <http://msdn.microsoft.com/de-de/library/ms123401.aspx> (aufgerufen am, 18.01.2012)

²⁹ <http://dev.opera.com/web> (aufgerufen am, 18.01.2012)

³⁰ <http://developer.apple.com/library/safari/navigation/> (aufgerufen am, 18.01.2012)

³¹ <http://www.google.com/doubleclick/studio/swiffy/gettingstarted.html> (aufgerufen am, 18.01.2012)

6.2 CSS3

Die Browserkompatibilitäten von CSS3 unterscheiden sich enorm. Dauernd kommen neue Features hinzu und die Browserhersteller nehmen unterschiedliche Module in den verschiedenen Versionen ihres Programmes auf. Nachfolgend sind die animationsbasierenden CSS3-Eigenschaften, wie und ob sie mit den Browsern kompatibel sind, aufgelistet.

Die Transform-Methode wird in allen gängigen Browsern mit ihren herstellerspezifischen Präfixen unterstützt. Nur der Internet Explorer macht hier eine Ausnahme, da dieser die Eigenschaft erst ab Version 9.0 interpretiert. Doch gibt es auch für ältere Internet Explorer Techniken, um Transformationen herstellen zu können, da mit dem sogenannten Matrix-Filter dasselbe Resultat erzielt werden kann. Diese Matrix ist etwas umständlich zu erzeugen, allerdings gibt es dafür gewisse Tools um den Source Code einfacher zu erstellen. Eine zweite Alternative kann dafür JavaScript sein (*Maurice, 2011, S. 200f*). Die 3D-Transformationen sind schon weit weniger verbreitet. WebKit-basierende Browser sind den anderen Herstellern weit voraus (*Maurice, 2011, S.21*). Safari unterstützt dieses Modul mit den herstellerspezifischen Präfixen ab Version 4.0 und Chrome ab Version 12.0. Hier ist auch der Internet Explorer 10.0 an der Spitze und hat diese Funktion integriert.

Transitions sind bei allen Browsern, außer jenen vor Internet Explorer 10.0, integriert. Bei älteren Browsern, die diese Funktion nicht unterstützen, kann jedoch wie schon bei der Transform-Methode ein JavaScript eingebunden werden, damit die Übergänge zwischen den Zuständen den gewünschten Effekt erzielen.

Die Animations-Eigenschaft ist in Mozilla Firefox ab Version 5.0 verfügbar. Im Chrome ist `animation` ab 11.0 und in Safari ab 4.0 verwendbar. Der Internet Explorer besitzt dieses Modul erst ab Version 10.0. Der einzige Browser, der noch nicht über dieses Feature verfügt ist Opera.

Das Positive an Browsern wie Firefox, Chrome, Safari und Opera ist, dass laufend neue Updates zu Verfügung stehen, um immer auf dem neuesten Stand zu sein. Insgemein sind die WebKit-basierenden Browser in dieser Angelegenheit, die am fortschrittlichsten (*Maurice, 2011, S. 21*). Diese sind in älteren Versionen des Internet Explorers sehr spärlich.

Die meisten Eigenschaften von CSS3 funktionieren auch in den heutzutage stark wachsenden Mobilwebbrowsern, auf diese im nächsten Abschnitt näher eingegangen wird.

6.3 Mobile Endgeräte

Im Kapitel 3.1 wurde bereits abgearbeitet wie mobile Browser auf HTML5 reagieren. Eigentlich funktioniert diese Sprache recht gut auf den modernen Geräten. Fast ähnlich ist es bei CSS3, wo die WebKit-basierenden Browser Android³² und iOS Safari³³ sogar 3D-Transformationen unterstützen. Opera ist denen, bezüglich dieser Funktion, etwas hinterher, jedoch andere Features, wie zum Beispiel den Transitions, werden unterstützt. Im Gegensatz dazu gibt es hier eine größere Problematik bei Flash, das von mehreren Smartphones und Tablets nicht angezeigt werden kann. Infolgedessen hat Adobe am 9. November 2011 beschlossen, ihre Software nicht weiter für mobile Endgeräte weiterzuentwickeln, wie Danny Winokur, Vice President & General Manager, Interactive Development von Adobe berichtet:

„Our future work with Flash on mobile devices will be focused on enabling Flash developers to package native apps with Adobe AIR for all the major app stores. [...] These changes will allow us to increase investment in HTML5 and innovate with Flash where it can have most impact for the industry, including advanced gaming and premium video. Flash Player 11 for PC browsers just introduced dozens of new features, including hardware accelerated 3D graphics for console-quality gaming and premium HD video with content protection.“

6.4 Browserstatistik

Eigentlich gäbe es in Browserangelegenheiten keine Probleme, wenn jeder Internetbenutzer die neuesten Browser mit ihren Updates installieren würde. Doch gibt es immer wieder Anwender, jene nicht wissen welche Updates oder Browser zur Verfügung stehen, oder welchen Vorteil das für sie bringen könnte. Dies sind vor allem ältere Menschen, oder jene die kein großes Interesse an Computern zeigen. Nun ist es aber trotzdem sehr schwer zu sagen, welche Zielgruppe und wie viele Nutzer an welchen Browser arbeiten. Somit sind auch die Statistiken sehr unterschiedlich. Laut browser-statistik.de³⁴ liegt der Firefox im Jänner 2012 mit 42,6% an erster Stelle und der Internet Explorer mit 28,9% an Zweiter. Hiervon verwenden aber in etwa die Hälfte der Nutzer noch den Internet Explorer 8.0, und zirka ein Fünftel Versionen darunter. Nachdem doch noch sehr viele User Browser benutzen, deren Funktionalitäten eher gering sind, ist es wichtig zu bedenken, welche Features und Tools verwendet werden, da die Gesamtheit die Websites betrachtet. In der nachfolgenden Tabelle werden alle gängigen Browser mit ihren Nutzern aufgezeigt:

³² <http://developer.android.com/reference/packages.html> (aufgerufen am, 18.01.2012)

³³ <http://developer.apple.com/library/safari/#documentation/AudioVideo/Reference/> (aufgerufen am, 18.01.2012)

³⁴ <http://www.browser-statistik.de/> (aufgerufen am, 18.01.2012)



Abbildung 6: Browserstatistik laut browser-statistik.de

7 Adobe Flash oder HTML5, CSS3 und JavaScript

In diesem Kapitel wird zuerst geschildert, welche Effizienz, aber auch welche Nachteile Flash, im Gegenzug zu den anderen Websprachen hat, wobei es natürlich auch Nachteile gibt. Im Anschluss darauf wird es nochmals eine Zusammenfassung von HTML5, CSS3 und JavaScript geben, die aufzeigen wird, welche Lösung nun die effektivste ist.

7.1 Flash

Das Gute an Flash ist, dass es seit 1996 (*Weschkalnies, 2010, S. 29*) auf dem Markt ist und ständig weiterentwickelt wird. Somit ist das Programm auch browserunabhängig und die Website wird überall korrekt dargestellt, wo der Flash-Player auf dem Rechner installiert ist und der kostenlos zum Download verfügbar ist. Es ist eine sehr gut dokumentierte und hochprofessionelle Software, die sämtliche Schnittstellen zu diversen Systemen aufweist. Das Programm ist überaus vielseitig verwendbar und unterstützt Audio, Animationen, erweiterte Videobearbeitung und Interaktivität. Diese Tools sind sinnvoll für Online-Spiele, Audio, Filme, Diagramme, Warenkörbe Feedback-Formulare, Foto-Diashows, und verschiedenste Webanwendungen, die für Dynamik und Interaktivität sorgen sollen. Solange die verwendeten Flash-Bereiche nicht übermäßig sind und vorwiegend Vektorgrafiken verwendet werden, wird auch die Dateigröße gering gehalten. Doch erlaubt das Programm auch die Einbindung von Bitmap-Dateien, wo berücksichtigt werden muss, dass die Bilder nicht im Überfluss verwendet werden, da sich dies schlecht auf die Performance auswirkt. (*AbiChedid, 2011*). Oft genügen die Schriften, die dem Browser zur Verfügung stehen nicht aus, so können mit der SIFR-Technologie (Scalable Inman Flash Replacement), im Gegensatz zu gewöhnlichen HTML-Seiten, alle beliebigen Schriftarten verwendet werden. SIFR ist eine Mischung aus Adobe Flash und JavaScript und ermöglicht, dass Textelemente auf allen HTML-Websites gleichartig ersetzt werden. Hier müssen jedoch auch Kenntnisse von JavaScript vorhanden sein, welches bei den Anwendern aktiviert sein muss.³⁵ Wie bei den Authoring-Tools können auch in Flash Animationen erstellt werden, ohne nur eine Zeile Code schreiben zu müssen. Im Allgemeinen sind die Zeitleistenanimationen nicht allzu schwer umzusetzen, sollen jedoch komplexere Anwendungen angefertigt werden, muss auf ActionScript zurückgegriffen werden. Diese Programmiersprache ist schwer zu erlernen, weswegen für diese Tätigkeiten meistens ein Techniker engagiert werden muss, das mit einer Kostensteigerung einher geht. Sollen Flash-Animationen wiederum selbst hergestellt werden, ist die hierfür erforderliche Software besonders teuer.

³⁵ <http://www.swfir.com/> (aufgerufen am, 18.01.2012)

Im Gegensatz zu den Open-Source Programmen, kann Adobe Flash nur vom eigenen Unternehmen weiterentwickelt werden, weshalb weitere Updates erneut zu bezahlen sind. Soll die gesamte Website in Flash realisiert werden, treten einige Probleme auf, weil die Websites nicht immer und überall richtig dargestellt werden können. Das liegt einerseits daran, dass Flash nicht allein vom Browser dargestellt werden kann, sondern den sogenannten Flash-Player benötigt. Ist dieser oder die richtige Version nicht installiert, muss der User das Programm downloaden und installieren. Diesen Vorgang finden viele Anwender als störend oder irritierend und beenden den Prozess.

Ein weiterer großer Nachteil ist, dass Suchmaschinen die Inhalte nicht interpretieren können. Die Flashsite ist in der Regel eine einzige Datei aus Bildern, bei der auf Inhalte nicht separat zugegriffen werden kann. Deshalb ist es wichtig, die Suchmaschinenoptimierung richtig umzusetzen. Wird Flash in HTML eingebunden und als Meta-Daten die wichtigsten Suchbegriffe angegeben, können diese von Webcrawlern interpretiert werden. Nichtsdestotrotz lassen viele Flash-Entwickler die Suchmaschinen außer Acht oder gehen schlampig damit um, was zu einem schlechten Resultat bei der Suche führt.³⁶ Einige Suchmaschinen, wie zum Beispiel Google, bieten die Option an, dass direkt neben dem Suchergebnis eine Vorschau abgebildet wird. Das ermöglicht den Besucher eine bessere Vorstellung der Website zu bekommen. Doch reine Flash-Websites können diese Vorschau oft nicht korrekt wiedergeben, sondern es wird entweder nur ein leeres Bild oder lediglich ein Bild der Startseite angezeigt. Suchmaschinen sind auch nicht in der Lage, die richtige Seite innerhalb der Datei zu finden. Dasselbe Problem liegt bei der Barrierefreiheit. Flashsites können von Screenreadern nicht gelesen werden, sondern nur den alternativen Text. Wird dieser nicht ausführlich angegeben, ist der Inhalt nur schwer für Screenreader zu lesen und wiederzugeben.

Ein zusätzliches Defizit von Flash ist, dass die Schriften nicht größer skaliert oder gar markiert werden können. Außerdem funktioniert bei reinen Flash-Websites der Vor- und Zurückbutton oft nicht und die Verlinkung ist kompliziert zu generieren. Möchten Lesezeichen gesetzt werden, kann das nur von der Startseite erfolgen. Da Flash als alleinstehendes Objekt im Browser gesehen wird, kommen interaktive Tastaturbefehle abhanden. Je mehr Flash bei einer Internetseite eingebaut wird, desto längere Ladezeiten werden für dessen Aufbau benötigt, was viele Anwender als störend empfinden und deshalb zu einer anderen Seite wechseln. Außerdem können bis auf wenige Ausnahmen, Mobile Geräte, Flash nicht darstellen. Darüber hinaus wurde Flash nicht für Touchscreen konzipiert, und kann daher Fehler auf Smartphones aufweisen. Laut Adobe, wird Flash auch nicht für diese Geräte weiterentwickelt. Da die Zahl dieser Anwender aber laufend ansteigt, muss für mobile Anwendungen JavaScript, HTML5 und CSS3 verwendet werden (*AbiChedid, 2011*). Wird

³⁶ <http://www.seoline-blog.de/seo-und-flash/> (aufgerufen am, 18.01.2012)

nicht die gesamte Website in Flash generiert, sondern nur Teile davon, funktioniert die Suchmaschinenoptimierung und auch die Performance recht gut. Hierzu einige Anmerkungen, die bei einzelnen Bereichen einer Website beachtet werden sollten:

- *Flash sollte nicht zur Seitennavigation verwendet werden.*
- *Flash Dateien sollten in HTML Seiten eingebettet werden.*
- *Aussagekräftige Seiten Titel und Beschreibungen sollten verwendet werden.*
- *Eine gesamte Seite sollte nicht in ein einzelnes Flash File gepackt werden. Besser ist es, den Inhalt auf mehrere Dateien aufzuteilen mit unterschiedlichen HTML Seiten.*
- *Flash sollte vorwiegend zur optischen Gestaltung verwendet werden und nicht um wichtige Inhalte hineinzupacken. Die zentralen Inhalte sollten in HTML auf der Seite platziert werden.*

(<http://www.seoline-blog.de/seo-und-flash/>)

Fazit ist, dass die Usability von Flash in vielen Bereichen, nicht im ausreichenden Maße vorhanden ist, sondern in Zukunft eher als Rich Media Application oder plattformübergreifende Publikationstechnologie eingesetzt werden wird.

7.2 HTML, CSS und JavaScript

HTML5, CSS3 und JavaScript sind offene Webstandards, die wesentliche Vorteile umfassen. Die Inhalte von HTML basieren auf Text und sind im Unterschied zu Flash nicht in SWF-Dateien versteckt. Durch die sinnvolle Struktur können alle Tags, mit ihren Attributen, von Suchmaschinen und Screenreadern richtig interpretiert werden, das sich für die Wirtschaftlichkeit auszeichnet. Die Websites müssen lediglich von den Entwicklern richtig ausgezeichnet werden, indem vernünftige Seitentitel, Metadaten und Alt-Attribute angegeben werden. Die Texte von HTML-Seiten können außerdem viel unproblematischer als in Flash weiterverarbeitet und geändert werden. Sie können dazu verwendet werden, bestimmte Teile der Seite einfach zu verlinken, da jede Seite eine eigene URL besitzt. Der dargestellte Inhalt der Website kann unter anderem auch kopiert, gespeichert und weiterverwendet werden. Die Fähigkeit der Browser, den Quelltext darzustellen, bietet den Entwickler-Communities die Möglichkeit, sich besser zu qualifizieren. Die Simplizität der Zugänglichkeit um den Code zu optimieren und pflegen ist bei den Sprachen HTML, CSS und JavaScript gegeben. Die Software, um mit diesen Sprachen eine Website herzustellen, ist oft kostenlos zum Download verfügbar. Es müssen also keine teuren Programme gekauft oder Plug-Ins installiert werden, die von nur einer Firma abhängig sind. Dies ist auch der Grund, weshalb offene Webstandards im Gegensatz zu Flash auf allen Geräten funktionieren (*Whitehouse,*

2009). Außerdem müssen keine eigenen Flash-Techniker engagiert werden, sondern Webdesigner erledigen diese Arbeit. Schon beim stylen können Objekte animiert werden, was sehr einfach zu praktizieren ist. Sowie ActionScript hat auch jQuery eine sehr gute Dokumentation, welches Tool leicht verständlich ist. Das spart für den Designer Zeit und Geld für den Kunden.

Trotzdem sollte nicht außer Acht gelassen werden, dass bei Standards, dessen Code frei zu Verfügung steht, Hacker unproblematisch Zugang finden. Zudem sind die Funktionen von HTML, CSS und JavaScript auch eingeschränkt. Flash ist in der Lage, alle möglichen Effekte zu gestalten. So ist zum Beispiel das Erstellen einer 3D-Animation erheblich aufwendiger mit HTML, CSS und JavaScript umzusetzen. Wo bei Flash ein Plug-In nötig ist, ist es bei HTML5 und CSS3 nötig, einen neuen Browser installiert zu haben. Bei CSS3 werden hierbei noch die umständlichen Präfixe benötigt, dessen Code sich um einiges verlängert. JavaScript ist zwar in nahezu jeden Browser integriert, kann jedoch von jedem Nutzer deaktiviert werden. Für aufwendige Bewegungsabläufe, sind sehr gute Kenntnisse der neuen Technologien erforderlich. Meistens ist für solche Animationen sehr viel Code notwendig, was sich auf die Ladezeiten negativ auswirkt und wodurch die Animationen nicht immer flüssig abgespielt werden. Auch das Canvas-Element benötigt lange Ladezeiten, da es auf Pixelgrafiken basiert. Besonders bei schwächeren Computern ist das Problem häufig zu sehen, weswegen diese Lösungen nur für kleinere Projekte verwendet werden sollte. Je nach verwendeten Browser, können Animationen und Interaktionen nicht einheitlich dargestellt werden. Dazu müssen Videos in drei verschiedenen Formaten eingebunden werden, damit diese auf jeden Browser angezeigt werden. Zwar sind Flash-to-HTML5 Converter und Authoring-Tools eine sehr gute Lösung, um Animationen, sowohl auf jeden Browser, als auch auf allen mobilen Geräten anzeigen zu können, doch um komplexe und aufwendige Projekte zu verwirklichen, müssen auch bei diesen Anwendungen Programmierkenntnisse vorhanden sein.

Seit HTML5, CSS3 und JavaScript gibt es kaum noch Beschränkungen bei Animationen und Interaktionen. Es können bereits Spiele und 3D-Transformationen damit umgesetzt werden, wobei es aber ganz deutlich auf der Hand liegt, dass das nicht immer die effizienteste Lösung ist. Nachfolgend wird wiedergegeben, welche Art nun den größten Nutzen in den verschiedenen Bereichen des Webdesigns bringt.

8 Schluss

Noch vor ein paar Jahren hätte sich keiner gedacht, dass es derart einfach ist, mit Websprachen wie HTML und CSS so viel Dynamik und Interaktion in eine Website einzubauen. JavaScript hat mit ihrem jQuery-Tool schon länger für Ansehen gesorgt. Doch dass es möglich ist, in Zusammenarbeit mit HTML5 und CSS3 solche Animationen anbieten, dass damit sogar einfache Spiele umgesetzt werden können, ist neu. Wo Slideshows früher ausschließlich mit Flash realisiert wurden, werden sie heute nur noch mit JavaScript umgesetzt. Trotzdem sollen Websites der Gesamtheit bereitgestellt werden, doch noch nicht jeder hat einen modernen Browser installiert der die neuen Features anzeigen kann. Auch bei Flash wird ein installierter Player benötigt, dieser jedoch weitestgehend auf jeden Computer zu finden ist. Nachdem Flash ein bedeutend ausgereifteres Programm ist, hat es in diesem Sinne einige Vorzüge, diese im Kapitel 7.1 ausführlich erläutert werden. Allerdings gibt es nach wie vor Websites, die in Gänze aus Flash bestehen. Heutzutage sollte das auf jeden Fall vermieden werden, da es gegenwärtig sehr aktuell ist, die Webpräsenz auf diversen Sozialen Netzwerken zu verlinken, oder mit mobilen Endgeräten im World Wide Web zu stöbern. Sehr viele dieser Geräte können Flash nur fehlerhaft oder überhaupt nicht anzeigen. Laut Adobe wird Flash auch nicht für Smartphones und Tablets weiterentwickelt, womit es für Inhaber einer Flash-Site nur empfehlenswert ist, ihren Internetauftritt zu erneuern. Auch wichtige Teile wie Navigation, Links oder Text sollten sich nicht in einer SWF-Datei befinden. Bereiche wie Werbeflächen und Banner hingegen stellen kein Problem dar, wenn es sich lohnt die Ressourcen aufzubringen. Doch können diese simplen Elemente genauso gut in HTML5, CSS3 und JavaScript erzeugt werden. Der Code ist schnell geschrieben und auch für den Kunden ist die Website mit ein bisschen Sachkenntnis einfach zu pflegen.

Nachdem HTML5, CSS3 und JavaScript sich mit rasanter Geschwindigkeit entwickeln, wird oft debattiert ob Flash womöglich nicht mehr lange Bestand hat. Bei komplexeren Projekten jedoch, wie zum Beispiel 3D-Umgebungen oder Spielen wird es sicherlich noch eine Weile erhalten bleiben. Solch umfangreiche und schwierige Animationen mit HTML5, CSS3 und JavaScript zu bauen, würde eine stark erhöhte Entwicklungszeit und somit auch höhere Kosten zur Folge haben.

Da aber die Sprachen HTML5, CSS3 und JavaScript sehr gut miteinander arbeiten und damit viele Möglichkeiten bereitstellen, sind diese sicher eine gute Lösung für kleinere Animationen, sowie Banner oder Hintergründe. Vor allem wenn die Website suchmaschinenfreundlich sein soll, sollte auf jeden Fall auf die offenen Webstandards zurückgegriffen und auf Flash verzichtet werden.

Nachdem aber auch das Canvas-Element in der Lage ist, Bereiche von ActionScript zu übernehmen und CSS3 und JavaScript als Erweiterung dienen können, sollte daran gedacht werden alle Sprachen miteinander zu verbinden. Flash hat sicher noch länger Bestand, aber vielleicht werden es HTML5, CSS3 und JavaScript doch noch schaffen es zu verdrängen.

9 Literaturverzeichnis

1. Weschkalnies, Nick. (2010). „*Adobe Flash CS5 das umfassende Handbuch*“. 1. Aufl. Bonn: Galileo Press.
2. Kiupel, Inken. (2011). „Was ist HTML5?“. <http://community.oreilly.de/blog/2011/01/28/was-ist-html5/>. (aufgerufen am 18.01.2012)
3. Kramer, Viviane. (2011) „Was ist CSS3?“. <http://community.oreilly.de/blog/2011/07/08/was-ist-css3/>. (aufgerufen am 18.01.2012)
4. Waschier, Gerhard. (2011). „Was ist jQuery? - medani webdesign wien“. <http://www.medani.at/allgemein/was-ist-jquery/>. (aufgerufen am 18.01.2012)
5. Maurice, Florence. (2011). „*CSS3 : Leitfaden für Webdesigner*“. München: Addison-Wesley).
6. Braun, Herbert. (2011). „Animationen mit CSS3“. In: C't extra Webdesign. 2/2011. S.17-19.
7. Braun, Herbert. (2011). „Warum HTML5 ein großer Wurf ist“. In: C't extra Webdesign. 2/2011. S. 6-11.
8. Dr. Zota, Volker. (2011). „Video (fast) ohne Flash“. In: C't extra Webdesign. 2/2011. S. 12-16.
9. Gilbertson, Scott. (2011). „Adobe Unveils 'Wallaby' Flash-to-HTML5 Converter“. <http://www.webmonkey.com/2011/03/adobe-unveils-wallaby-flash-to-html5-converter/>. (aufgerufen am 18.01.2012)
10. Kaltschmidt, Thomas. (2011). „Work around Flash“. In: Page. 9/2011. S. 29-31.
11. Gordon, Marcel. (2011). „convert SWF-Files to HTML5“. <http://googlecode.blogspot.com/2011/06/swiffy-convert-swf-files-to-html5.html>. (aufgerufen am 18.01.2012)
12. Ozer, Jan. (2010). „Flash Player: CPU Hog or Hot Tamale? It Depends“. <http://www.streaminglearningcenter.com/articles/flash-player-cpu-hog-or-hot-tamale-it-depends-.html>. (aufgerufen am 18.01.2012)
13. Braun, Herbert. (2011). „Neuerungen in ECMAScript 5 und JavaScript“. C't extra Webdesign. 2/2011. S. 20-25.
14. Wenz, Christian. (2010) „JavaScript – Das umfassende Handbuch“, 10. Aufl. Bonn: Galileo Press, 2010)
15. Lennon, Joe. (2010) „Compare JavaScript Frameworks - An overview of the frameworks that greatly enhance JavaScript development“. <http://www.ibm.com/developerworks/web/library/wa-jsframeworks/>. (aufgerufen am 18.01.2012)

16. Zakas, Nicholas C. (2011). „Better JavaScript animations with requestAnimationFrame“. <http://www.nczonline.net/blog/2011/05/03/better-javascript-animations-with-requestanimationframe/>. (aufgerufen am 18.01.2012)
17. Lau, Oliver. (2011). „Zeichnen auf dem Canvas-Element“. In: C't extra Webdesign. 2/2011. S. 38-43.
18. Braun, Herbert et. al. (2011). „Vektorgrafiken im Web“. In: C't extra Webdesign. 2/2011. S. 28-37.
19. Schulze, René. (2011). „Mit Canvas in die 3. Dimension“. In: Weave. 4/2011. S. 76-81.
20. Michel, Axel. (2010). „Canvas kanns was?“. <http://www.am530.de/2010/06/canvas-kann-was/>. (aufgerufen am 18.01.2012)
21. AbiChedid, Boutros. (2011). „Flash Website: Advantages and Disadvantages“. <http://bacsoftwareconsulting.com/blog/index.php/web-development/flash-website-advantages-and-disadvantages/>. (aufgerufen am 18.01.2012)
22. Whitehouse, Phil. (2009). „Flash vs. JavaScript“. <http://philwhitehouse.blogspot.com/2009/07/flash-vs-javascript.html>. (aufgerufen am 18.01.2012)
23. Winokur, Danny. (2011). „Flash to Focus on PC Browsers and Mobile Apps; Adobe to More Addressively Contribute to HTML5“. <http://blogs.adobe.com/conversations/2011/11/flash-focus.html>. (aufgerufen am 18.01.2012)