

Web UI Animation

Group 5

Rok Kogovšek, Alexei Kruglov, Fernando Pulido Ruiz, and Helmut Zöhrer

706.041 Information Architecture and Web Usability WS 2016
Graz University of Technology
A-8010 Graz, Austria

04 Dec 2016

Abstract

TO DO

Writing a survey can be a traumatic endeavour. It might be a student's first foray into academic research. There are often obstacles and false dawns along the way. This survey paper takes a fresh look at the process and addresses new ways of accomplishing this daunting goal.

The abstract should concisely describe what the survey is about. State the areas which are covered and also those which are not covered. Market your survey to your readership. Also, make sure you mention all relevant keywords in the abstract, since many readers read *only* the abstract and many search engines index *only* the title and the abstract.

This survey explores the issues concerning the writing of an academic survey paper and presents numerous novel insights. Special attention is paid to the use of clear and simple English for an international audience, and advice is given as to the use of technical aids to production.

© Copyright 2016 by the author(s), except as otherwise noted.

This work is placed under a Creative Commons Attribution 4.0 International (CC BY 4.0) licence. It uses the LaTeX template from "Writing a Survey Paper" by Keith Andrews, used under CC BY 4.0 / Desaturated from original

Contents

| | |
|--|------------|
| Contents | i |
| List of Figures | iii |
| List of Tables | v |
| List of Listings | vii |
| 1 Introduction | 1 |
| 1.1 Not a Series of Summaries | 1 |
| 1.2 Read All the Papers and Research Some More | 1 |
| 1.3 Dividing up the Field | 1 |
| 2 Animation | 3 |
| 2.1 sections by F.P.R. | 3 |
| 3 Cascading Style Sheets (CSS) | 5 |
| 3.1 Do Everything You Can With CSS | 5 |
| 3.2 CSS Animation Declaration | 6 |
| 3.2.1 Animation Property and Keframe Rule | 6 |
| 3.2.2 Transition Property | 6 |
| 3.2.3 Selector Pattern | 6 |
| 3.3 CSS Examples | 6 |
| 3.3.1 Navigation Animation | 6 |
| 3.3.2 Loading Animation | 6 |
| 3.4 Scalable Vector Graphics (SVG) | 7 |
| 4 JavaScript (JS) | 9 |
| 4.1 When to Use JS instead of CSS | 9 |
| 4.2 Examples of Useful Animations with JS | 9 |
| 5 Concluding Remarks | 11 |
| Bibliography | 13 |

List of Figures

| | | |
|-----|--------------------------------|---|
| 2.1 | Storyboard Sketching | 4 |
|-----|--------------------------------|---|

List of Tables

List of Listings

| | | |
|-----|-----------------------------|---|
| 4.1 | Some Code Snippet | 9 |
|-----|-----------------------------|---|

Chapter 1

Introduction

[TODO: An academic survey paper presents a survey or overview of the state of the art in a particular field. Every chapter and every section should have some introductory text at the beginning, like this text. Never jump straight in to the first section or subsection without one or more paragraphs of introductory text.]

1.1 Not a Series of Summaries

A survey is *not* simply a series of summaries of papers. If I have given you say 8 papers to start you off, what you should *not* do is: divide up the papers (read two each) and produce a series of 8 unconnected paper summaries.

1.2 Read All the Papers and Research Some More

Each of you should read *all* the papers and resources: both those I gave you and those you found yourselves. Make sure you search for more papers and resources yourselves. Not just a Google search. Search the ACM [ACM-DL] and IEEE [IEEE-DL] digital libraries, citeseer [CiteSeer], and mendeley [Mendeley]. You may want to use mendeley to collect your resources or maybe maintain a .bib file within an SVN repository.

Include a list of *all* the relevant papers and resources you have found and mark those you have chosen to focus on. Make sure *all* the papers and resources you found or were given appear in the bibliography.

1.3 Dividing up the Field

The hardest part of any survey is dividing up the field. Look for common concepts and threads in the papers and resources. Do they report similar or dissimilar results? Does one paper or resource support or contradict another?

Once you have all read all the papers: you need to construct a small hierarchy (taxonomy) to classify the concepts appearing in the papers and resources. Structure your survey into chapters and sections based on your taxonomy.

Chapter 2

Animation

"Animation is defined as changing some property over time. On the other hand, motion is the act of moving or the process of being moved. . . . To put it more simply, all motion is animation, but not all animation is motion."[Head, 2016]

[TODO:

2.1 sections by F.P.R.

]

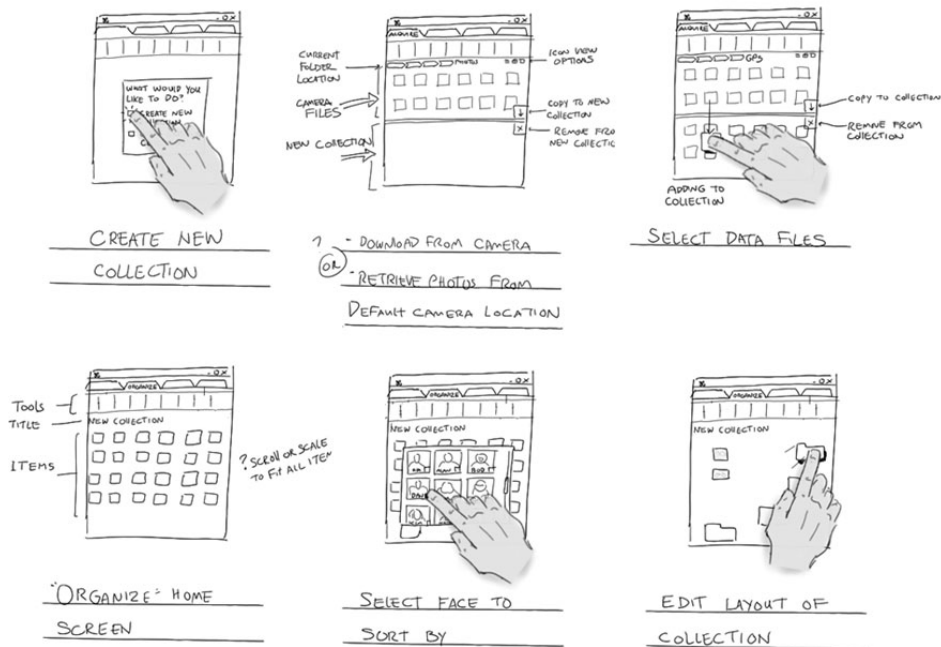


Figure 2.1: Example of storyboard sketching for drag and drop animation [Windows Dev Center, 2016].
[Used with permission from Microsoft - Microsoft Copyrighted Content Guidelines]

Chapter 3

Cascading Style Sheets (CSS)

Knowing the usefulness of animation in web UI and the correct way of animation planing are just the fundamantals for our conceptual plans. Those still need to be implemented to get the end product and here we usually hit a wall build from the various tools, that say they can all solve our problems. Even well established people in the field have stories as such to tell. Val Head actually started with animation due to an interesting Flash workshop. Flash was at that time the de facto king in its era, however as we know, that era is already dead. Nowadays we can acomplish all we could with Flash and more with just the core parts of the web, namely HTML, CSS and JS [Head, 2016].

3.1 Do Everything You Can With CSS

With Responsive web design (RWD) in our websites and animation being part of the design, see section [\[TODO: 2\]](#), it should be natural to use the guidelines of RWD also in animation planning. Andrews [2016] teaches us that one of the RWD guidelines is also Progressive enhancement, which is best described with words of the conceptual authors Champeon and Finck [2003]: *"Leave no one behind. . . . accessibility is for everyone, not just the disabled"*. With CSS nowadays being a core part of the web and at the same time being the lowest web component that enables animation with RWD guidelines¹, one should always implement with CSS and HTML alone as much of the desired animation as possible. One has only to make sure the browser support for the animated attribute.

Other supporting arguments for use of CSS as the starting point for web UI animation beside responsiveness can be summarized with the the so called "Simple CSS Truths", a list of truths by Palermo IV [2015] enhanced with the teachings of Andrews [2016]:

CSS allows for separation of concerns - With CSS the form is separated from the page's HTML structure and content. Makes it easier to read, maintain and crawl the code.

CSS has a captive audience - Support for CSS development is huge. At the same time more and more libraries, tools and frameworks focus on improving and simplifying CSS development.

CSS is fast - External CSS speeds up HTML downlaod and loading compared to HTMLs with duplicated inline styles. Compared to JavaScript it also processes transitions and animations faster.

CSS is fault-tolarent - Browser-unknown enhancements are simply ignored by the browser, while the remainder is still used and displayed.

CSS is everywhere - Modern browsers embrace CSS and feature support by each can be easily found online.

¹Of course one can just use an animated image, e.q. a GIF with an image sequence, and just append it with HTML into the design. However, this image will become a static component of the design and will not follow RWD guidelines.

3.2 CSS Animation Declaration

As stated in section [TODO: 2] , animation is about changing an element's attribute(s) over time. In CSS we can redefine it as a switch between CSS styles for a HTML element that happens gradually over time. It is stated that CSS animation should be done with *Animation* property(/ies) and *Keyframe* rule(s). However, that is not the only CSS way to accomplish animation, but of course the most efficient way [W3Schools, 2016].

[TODO:

3.2.1 Animation Property and Keframe Rule

]

[TODO:

3.2.2 Transition Property

]

[TODO:

3.2.3 Selector Pattern

]

3.3 CSS Examples

3.3.1 Navigation Animation

As stated in [TODO: ref section useful] , animation can help with navigation through tha page. In the examples we show two typical cases of such usage.

[TODO:

Hamburger Icon

]

Rand [2015] [Rand, 2015]

[TODO:

Current menu position indicator

]

3.3.2 Loading Animation

Another use of animation is user feedback, see [TODO: ref section useful] .

[TODO:

Rotating Icon

]

2D and 3D

[TODO:

Horizontal movement

]

dots, progress bar, pulz (wave motion)

3.4 Scalable Vector Graphics (SVG)

[TODO: section by A.K.]

Chapter 4

JavaScript (JS)

[TODO: CHAPTER INTRO IS MISSING - check chapter 2 and 3]

4.1 When to Use JS instead of CSS

As a rule of thumb, JS should not be used if the same effect could be achieved with plain CSS. This is due to performance and resource management reasons. Just when CSS is stretched to its limits, JS should be used. There is a rough distinction whether to use CSS or JS for particular kinds of tasks:

- Use CSS animations for simple transitions, like changing the state of a UI element.
- Use JS animations to get advanced effects like bouncing, stop, pause, rewind, or slow down (there is more control over animations).
- When choosing to animate with JS, considering using the Web Animations API or a modern framework (comfortable to work with) may be desirable.
- Using both CSS and JS works also well:
 - perform animations with CSS
 - control states with JS

4.2 Examples of Useful Animations with JS

[TODO: this!] [TODO: citing with: Lewis and Thorogood [2016] or [Lewis and Thorogood, 2016]]

```
1 // create some nodes
2 var headline = document.createElement('h1');
3 var text = document.createTextNode('Dies ist eine Überschrift');
4 // "offline" node manipulation
5 headline.appendChild(text);
6 // adding node to DOM
7 document.getElementsByTagName("body")[0].appendChild(headline);
```

Listing 4.1: This is a code snippet where JS is used in a meaningful way.

Chapter 5

Concluding Remarks

Bibliography

- Andrews, Keith [2016]. *Information Architecture and Web Usability*. 2016 (cited on page 5).
- Champeon, Steven and Nick Finck [2003]. *Inclusive Web Design for the Future*. 2003. http://hesketh.com/publications/inclusive_web_design_for_the_future.html (cited on page 5).
- Head, Val [2016]. “Designing interface animation” [2016] (cited on pages 3, 5).
- Lewis, Paul and Sam Thorogood [2016]. *Web Fundamentals. CSS Versus JavaScript Animations*. English. Google Developers. 2016. <https://developers.google.com/web/fundamentals/design-and-ui/animations/css-vs-javascript> (cited on page 9).
- Palermo IV, J. Michael [2015]. *CSS in the Modern World*. English. Microsoft. 12th Nov 2015. <https://www.sitepoint.com/css-modern-world> (cited on page 5).
- Rand, Jesse [2015]. *The Genius—and Potential Dangers—of the Hamburger Icon (Flyout Menu). Meet the Hamburger Icon*. English. vital. 2015. <https://vtldesign.com/web-strategy/website-design-development/hamburger-icon-flyout-menu-website-navigation> (cited on page 6).
- W3Schools [2016]. *CSS3. CSS3 Animations*. English. W3Schools. 2016. http://www.w3schools.com/css/css3_animations.asp (cited on page 6).
- Windows Dev Center [2016]. *Hilo: Developing C++ Applications for Windows 7. Designing the Hilo User Experience*. English. Microsoft. 2016. <https://msdn.microsoft.com/en-us/library/windows/desktop/ff800706.aspx> (cited on page 4).