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 endevour. 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.

Contents

List of Figures

List of Tables

List of Listings

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 secion 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]

2.1 Not Just Motion!

In animation we change an object's attribute(s) over time to achieve an objective. As stated by Head [2016], animation is much more than adding movement to an object. Movement of an object can be described by a change of its 6 degrees of freedom, them being the coordinates in 3D space and the rotation around the 3D space axises. However an object has usually many more changeable attributes or properties. One could animate color changing and fading, invisibility through transparency, growth with scaling, focus with blurring, etc.

2.2 Why Animation in Web UI?

[TODO: Missing: easier navigation, less brain memory load, improve decision making, uder feedback, save screen space]

While writing this survey, with so much animation theory, a hilarious idea was just coming up to mind. Have you ever stopped to think about why some people just seem to be more funny than others? And of course, this happens everyday. Just think about a good joke told by someone with no real humour. It is meaningless. Now, think about the same joke being told by a profesional comedian on TV telling jokes over the weekend. Way more funny, is it not? But the point is, why and how is this be possible? It has lots to do with how they speak and how they move! As well of other minor factors. That, I am afraid, is the same as with animation.

Animation, be it in a cartoon, webpage, app or anywhere else, adds a new level of communication which can be hard to explain , but easy to feel. It creates a special connection between what the message is intended to be, and the receiver. It allows an additional cannel of invisible information to exist, and makes the viewer feel part of the process, and gets his attention grabbed in a practical way.

We always need to understand, where we are and where are we going to. Animation can helps us with this and even can improve the use of resources. Just think about when trying to navigate through a small screen. Having some animated window coming out when you put your mouse on, does not only save space (which is already a good enough reason to implement it), but makes the entire process of navigating easier and simpler. A user does not really want to think while navigating, and this can be fulfilled with animation. They can be guided, orientated or helped in going through a process just with animation, which will of course improve their user experience and make them feel more sure and comfortable about what they are doing, without thinking they are in a real puzzle and need to make big use of their brains.

Nowadays, it´s very common to access many sites through different devices. Just think about when trying to know any train timetable. You probably have the app in your Smartphone, but depending on where are you

at the moment, you might just google it up and search it through their webpage. Having animation allows sites to connect context and media. A user will just find himself guided throught the process no matter where he is or what device is he using, because of being under some potential animation process.

But, how do these animations grab people´s atention? There are over twelve basic principles, written by Disney, but two of them are like the foundation of animation, and the rest basically build up from them. Timing and spacing. Timing can be understood as the time length of an action to happen, let's say the duration of it, while spacing could be explained as the speed changes during one of these actions in an animation. Head [2016] (Chapter 2, page 18) describes their impact with: *"Timing and spacing convey the mood, emotion, and reaction of an object."*

2.3 Aim For Invisible Animation

In animation, it is very common during the designing process to end up messing things up. As a general rule, as a designer, your one good way to show love to the user is actually not making his or her life harder accomplishing the task at hand. Certainly designers do not hold such ill intentions, but as said before, it is quite easy and happens very often just to have users unnecesarily waiting for an animation to finish or having a hard time finishing a task due to animation. It could be said that the perfect animation is the one which is not noticed. That´s not 100% accurate, but a good idea in general.

*"Good interface animations need to be flexible and always feel responsive to a user's input even if the animation is currently animating."*Head [2016] (Chapter 3, page 48)

Users need feedback to feel themselves listened and understood. Animation can be understood as a bridge which takes the user to a higher quality experience. If an animation doesn't respect having the user happily informed with feedback, its experience quality will of course drop and so will the trust towards the UI. A user will never feel comfortable in a system which does not take into consideration its input.

It is also important, that an animation should never be a show-off work. It should be always remembered, that in general, the user interacting with an animation, is probably willing to do or finish something. A UI user is not just laying there waiting for an animation to happen and look at it itself and enjoy it. That could be the aim of animators years ago where they were really telling stories and people were fascinted about it. But times changed, and now the timing requirements are no longer the same as in those days. Study and carefully think about the suitable timing for each animation, because a short amount of milliseconds can be the limit between failing or succeeding. Good timing is more an art than a science [Head, 2016].

2.4 Development of animation

[TODO: Missing part, COPIED JUST THE POINTS FROM SLIDES]

Often problems in development

Usually discussed at the end stages of the project. Seen just as extra decoration. Should be part of the design! When everything is finished, it is "too late" to replan design (animation). Animation is underestimated in design.

Where to start planning?

Start with planning already in early stages. Animation can replace a fixed design element. Add only animation that helps user and does not distract or lowers the UX. Write about animation also in style guides and documentation. Understand basic principles of animation (Disney 12 or other) and how they effect the user. All important information most support accessibility(Animation is visual information).

How to plan?

In early stages storyboard sketching. During design prototyping prototype also animation (it is part of the design).

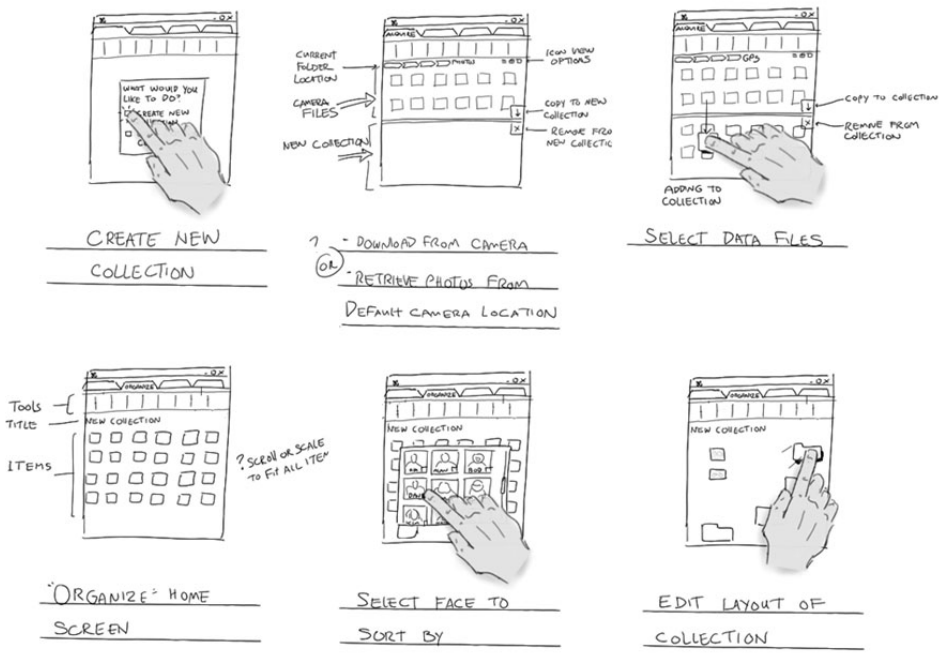

Figure 2.1: Example of storyboard sketching for drag and drop animation [Windows Dev Center, 2016].
[Used with permission from Microsoft - Microsoft Copyrighted Content Guidelines]

Chapter 3

Cascading Style Sheets (CSS)

Knowing the usefulness of animation in web UI and the correct way of animation planing are just the fundemantals 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. Nowdays 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 2.4, 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 nowdays being a core part of the web and at the same time being the lowest web component that enables animation with RWD guidelines[1], 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 simplifing 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.

[1] Of 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 2.1, 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). This may be the most efficient CSS way to accomplish animation, but CSS animation can also be achieved with *Transition* property(ies) and *selector* pattern(s)[W3Schools, 2016a; W3Schools, 2016b].

[TODO:

3.2.1 Animation Property and Keframe Rule

]

animation-name: desiredName

animation-duration: 5s

animation-timing-function: linear | ease | ease-in | ease-out | ease-in-out | cubic-bezier(x1,y1,x2,y1) | steps(stepSize, start, end)

animation-delay: 2s

animation-iteration-count: number | infinite

animation-direction: normal | reverse | alternate| alternate-reverse

animation-fill-mode: none | forwards | backwards | both

animation-play-state: paused | running

HEH

animation: name duration timing-function delay iteration-count direction fill-mode play-state;

Fine-tune the actual animation through keyframes @keyframes rule

```
1  div{
2    animation: desiredName 4s linear 0s infinite alternate;
3  }
4
5  @keyframes desiredName {
6    0%   {background-color:red;opacity: 1;transform: scale(1);}
7    25%  {background-color:yellow;transform: scale(0.8);}
8    50%  {background-color:blue;transform: scale(1);}
9    75%  {background-color:green;transform: scale(1.5);}
10   100% {background-color:red;opacity: 0.2;transform: scale(2);}
11 }
```

Listing 3.1: Simple example of non-motion animation with animation property and keyframes rule. A working example can be found in code/nonmotionAnimationCSS.html.

[TODO:

3.2.2 Transition Property and Selector Pattern

]

transition-property: propertyName | all

transition-duration: 5s

transition-timing-function: linear | ease | ease-in | ease-out | ease-in-out | cubic-bezier(x1,y1,x2,y1)

transition-delay: 2s

```
1  div{
2    background-color: red;
3    opacity: 1;
4    transition: all 4s;
5  }
6  div:hover{
7    background-color:green;
8    opacity: 0.2;
9    transform: scale(2);
10 }
```

Listing 3.2: Simple example of non-motion animation with transition property and hover selector. A working example can be found in code/nonmotionTransitionCSS.html.

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:

Figure 3.1: One the left we have a screenshot of code example hamburgerVariationCSS.html, which shows some of the different ways how hamburger icons are animated. On the right side we have hamburgerMenuCSS.html, that demonstrates the showing and hiding of the menu by hovering over the hamburger icon. [Screenshot taken by the authors of this survey. The code behind the pages is by using the Boyce, 2014; Couch, 2014 online snipets as a base.]

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 7).

Boyce, Robert [2014]. *Paper Nav*. English. CodePen. 2014. https://codepen.io/rboyce/pen/wlijs (cited on page 10).

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 7).

Couch, Jesse [2014]. *Menu "Hamburger" Icon Animations*. English. CodePen. 2014. https://codepen.io/designcouch/pen/Atyop (cited on page 10).

Head, Val [2016]. "Designing interface animation" [2016] (cited on pages 3–4, 7).

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 13).

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 7).

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 9).

W3Schools [2016a]. *CSS3. CSS3 Animations*. English. W3Schools. 2016. http://www.w3schools.com/css/css3_animations.asp (cited on page 8).

W3Schools [2016b]. *CSS3. CSS3 Transitions*. English. W3Schools. 2016. http://www.w3schools.com/css/css3_transitions.asp (cited on page 8).

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 5).