OULUN AMMATTIKORKEAKOULU

Bruno Posa

WEB ANIMATION WITH SVG

Featuring the mascot of Attractive Experiences Oy

WEB ANIMATION WITH SVG

Featuring the mascot of Attractive Experiences Oy

Bruno Posa
Bachelor’s thesis
Spring 2021
Business Information Technology
Oulu University of Applied Science

ABSTRACT

Oulu University of Applied Sciences
Business Information Technology

Author: Bruno Posa
Title of the thesis: WEB ANIMATION WITH SVG Featuring the mascot of Attractive Experiences Oy
Thesis examiner: Minna Kamula
Term and year of thesis completion: Spring 2021
Pages: 44 + 1 appendix

The cooperation between Attractive Experiences Oy and the author happened with the goal of producing working quality SVG animations of the company's mascot, Poe. The main basis for research has been topical literature, with an ending survey to validate author's applied learnings. The final animations achieved most of the outlined goals, but animating Poe requires additional efforts to achieve desired believability. The author commenced the process by learning about psychological principles and internet technologies, and later produced several animations through iterative processes with feedback. Web animation tools are present for people of all skills to join creating better internet experiences. Other companies with a mascot or a need for animation can use this report and familiarize with making animations for professional use.

SVG, web, animations, mascot, design, UX

CONTENTS

1 INTRODUCTION

Why are so many businesses investing in animations for their websites? How can simple shapes champion a brand? What does coding delightful experiences on the web look like? The author is discovering answers to these questions while animating the mascot of Attractive Experiences Oy. Attractive is a company offering actionable insights on the user experience of websites. Their mascot is most peculiar; according to Attractive, nobody knows what it is made of or how it moves, and people even say it appears from voids. What seems to be known, however, is that it is from another world, being infinitely intelligent, and its mission is to help humanity reach a new level of existence. (Attractive 2021, Cited 24.2.2021.) Day-to-day, it works as a User Experience expert and goes by the name of Poe. For our feeble human minds to comprehend Poe's existence, he takes shape of a bunch of slow-moving, colourful hexagons.

The journey of this report culminates with Attractive having Poe interact with its website visitors. The goal is conveying intended visual cues by placing Poe in contexts where he adds value. The author describes his findings about the process of animating, touching on topics of human traits and web technologies while designing and implementing scalable vector graphics (SVG) files for use by Attractive. The basis for this work is previous research and experience of industry experts, website analytics, as well as qualitative and quantitative feedback from users.

The author's interest in the topic stems from curiosity and potential for professional advancement. As a field between art and web development, SVG animations are an increasingly important part of modern websites as they bring value to businesses by increasing intuitiveness and improving user experience and interaction. Globally, the animation, visual effects, and game industry market was worth 261 billion US dollars in 2020 and has had a steady growth of 2-3% year-over-year (Research and markets 2021, Cited 24.2.2021). Even in the year of the pandemic, the IT market seemed stable and is forecasted to grow (Ballhaus, et al. 2020, Cited 24.2.2021).

The author is an amateur designer who previously made tentative forays into the world of animations. His previous work is tangential to the work at hand, consisting of bitmap motion graphics with offline non-interactive displays. User experience experts and experienced technologists from Attractive offer some guidance to the author in his process, as this report is enabled by a partnership with the company. The author's main contact from Attractive is their CTO, Jason Brower.

2 BASICS OF ANIMATING

When watching animated movies, audiences anthropomorphize characters that are behaving as persons, which means the animators did a good job. Cartoon animals and objects on the screen talk English, act out, and show emotions believably. Despite the animations being just projections of things on a screen, the audience is unobstructed in its ability to understand and connect with the story being told. But how can good animating be achieved with an abstract character such as Poe?

In trying to bring a 2D concoction of hexagons to life, it is useful to understand what behaviour is and what does it mean for humans to anthropomorphize something. According to merriam-webster.com, to behave is “to manage the actions of (oneself) in a particular way”, while anthropomorphism is “an interpretation of what is not human or personal in terms of human or personal characteristics” (Merriam-Webster 2021, Cited 31.1.2021). Another word for anthropomorphism is personification, and it is something humans start doing in their early age. Personification, can be of varying scope. A simple example is when one refers to the delta of a river as the ‘mouth of the river’, or when referring to time passage as ‘time marching on’. A slightly more comprehensive personification is ‘The sun racing to the finish line’. (Dodson 2008, 31.) Besides in language, anthropomorphizing easily occurs in human vision. We can project cognitive and emotional capability to even the simplest of two-dimensional geometrical shapes animated with movement. In a study by Fritz Heider and Marianne Simmel, participants watched a 2D animation of circles and triangles, after which they described the shapes as experiencing feelings and occupying roles such as ‘boyfriend’ and ‘girlfriend’. (Heider and Simmel 1944, 243–259.) ‘Giving minds’ to objects is a characteristic of common human behaviour, as it is done quickly and intuitively (Scholl and Tremoullet 2000, 299, 300). Orators from the beginning of time must have flourished in part due to us easily perceiving inanimate things as animate.

Modern stories in the form of films are a great source of inspiration for Poe’s behaviours. Characters such as the Magic Carpet from Disney’s Aladdin (figure 1) prove that even objects that are typically seen as boringly inanimate, can be invigorated believably. For animating such an object, there is no exact rules, but rather, a set of diverse practices that have been established by trial and error.

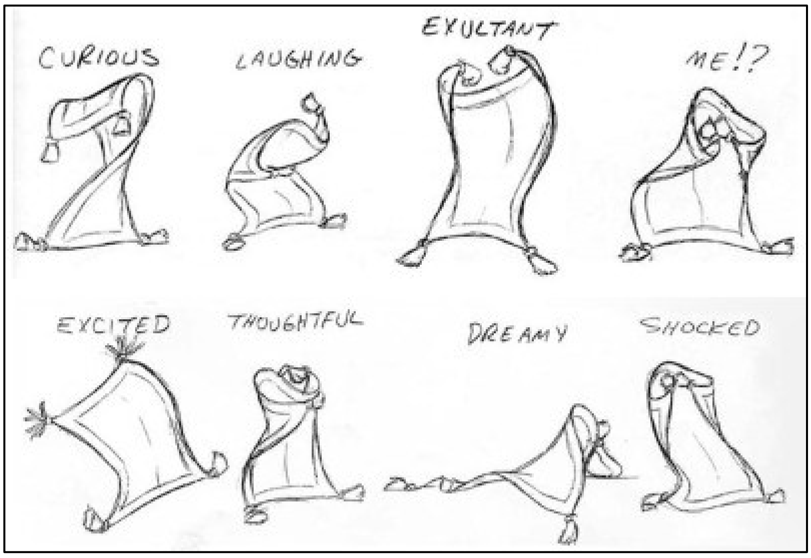


FIGURE 1: Animation of Magic Carpet from Disney's Aladdin (Culhane 1992).

According to the most prominent set of practices, 'The Principles of Animation' by Frank Johnston and Ollie Thomas from Disney Animation Studios, there are 12 principles to follow for best animation results:

1. Squash and Stretch – by expanding and compressing the body of the character during movements, a sense of inertia and weight is achieved.
2. Anticipation – small actions that take place right before the main action, such as lifting a finger before pressing a button; this achieves a sense that something is about to happen.
3. Staging – everything can be set up to illustrate a point with more clarity. When a character is happy, his movements are jolty and body open, and the opposite when he is sad, with even the background appearing darker.
4. Straight ahead action and pose to pose – two proven industry ways of animating – drawing a character continuously from beginning to end of a scene - or by first drawing a few key poses the character takes across the scene, with space in between filled afterwards.
5. Follow through and overlapping action – when a character is in motion and suddenly stops, it implies inertia if his hair and clothes continue their movements for a moment further. Realism is also approached by having different parts of the character's body move at different speeds, as one's hands may swing faster compared to its body swaying in a walk.
6. Slow in and slow out – character's attitude can be conveyed better if he spends more time in and near each of his key poses and moves more speedily in between them.

7. Arcs – animations are more realistic if paths of movements are curved, instead of rigid and straight.
8. Secondary Action – one's attitude is fortified with a small action near the main one, like wiping a tear after assuming a sad facial expression.
9. Timing – a spectrum of meanings can be achieved by changing only the amount of time a character's motion takes to complete. The identical motion of swaying one's head left-to-right can convey a punch by something if the motion is rapid; the character could be signalling someone to come towards him if it is slower, and it can appear as him just stretching when slowed even further.
10. Exaggeration – the staging is sometimes purposefully made unbelievable, so that the idea behind is clearer. The character's shock is made clearer if his jaw literally falls to the floor.
11. Solid Drawing – it is useful and efficient to have great drawing skills to effectively illustrate weight, balance, depth, and other 3D qualities in 2D animations.
12. Appeal – audiences connect better with an animation if they are drawn to it, rather than repulsed by it. Interesting and simple illustrations with a pleasing design captivate for longer. (Johnston and Thomas 1981, 47-70.)

These principles can improve how Poe conveys intended messages in the situations he is placed in. Certain ones, such as 'squashing and stretching' may not be needed, as Poe should convey that he exists only in the digital universe of ones and zeros, in which mechanical properties such as inertia have no reach. The peculiarities of human perception and principles of animation generally seem to lend themselves well to breathing life into a group of hexagons. Additionally, it is potentially good for animating Poe that hexagons are considered interesting in popular culture for many reasons, most of which are the shape's inherent natural peculiarities (CGP Grey 2020, Cited 31.1.2021).

3 BASICS OF WEB ANIMATION WITH SVG

Scalable Vector Graphics (SVG) is a markup language developed since 1999 by the World Wide Web Consortium (W3C). SVG was designed to interplay with CSS, JavaScript, HTML, and SMIL. It is based on Extensible Markup Language (XML) and can be created and edited both textually and graphically. SVG contain information for browsers to calculate mathematically what to render on a screen, instead of having specific information for each pixel that will be rendered, like JPEG and other bitmap image files. They are, therefore, cleanly scalable without loss of quality, so they do not pixelate when zoomed in or blown up to a large size. (MDN Web Docs 2021, Cited 22.2.2021.) They can compress easily to less than 20 percent of original size (W3C 2011, Cited 18.3.2021), and are easy to crop promptly, making them great for displaying as sprites (Drasner 2017, 23,24). SVG are ideal for simple graphics, such as logos, text, or icons, as the markup can be enormous in very complex scenes (Grigorik 2018, Cited 17.3.2021). The most interesting characteristic of SVG is that the format is designed to be easily animated and interacted with.

3.1 Context

In the public eye, SVG was initially just one of the many standards by W3C. Around 2011, however, smartphones were becoming popular, so scalability and file sizes became important in websites that served growing numbers of visitors from mobile phones, who had expensive and slow data plans. Search engines started ranking mobile-friendly and faster-loading pages higher in search results. Additionally, with an ever-growing number of websites online, companies wished to have their websites stand out somehow. All this was favourable for SVG, as they can be animated, scale easily, respond to hovering, scrolling, clicking - all while having smaller file sizes. (Miklos 2020, Cited 22.2.2021.) SVG format was not successful overnight, but over the span of the last decade it overtook Graphics Interchange Format (GIF) in popularity. Now, it is one of the top growing standards on the internet. (W3Tech 2021, Cited 23.2.2021.) SVG have a Document Object Model (DOM) and many commonalities with HTML, which allows its content to be manipulated easily. SVG are mostly made in vector graphics editors such as Adobe Illustrator, Corel, Sketch, Inkscape, and many others, as hand coding them to describe shapes is not as intuitive. Animating the created files can then be done in two ways: **image sprites** and **transformations**.

A **sprite** is a way of representing multiple images as a single image, or animation (Coyier 2009, Cited 17.3.2021). They are popular in the web, gaming, and animation industries, and can be made in many ways, for example by using Adobe Animate, or Blender. Sprites can be made with other image formats, besides SVG. In the context of SVG sprites in web development, CSS or JavaScript are used to display only a certain part of a larger SVG at a time. If designed to switch rapidly between different parts, animations can be made, such as the 29-frame one in figure 2.

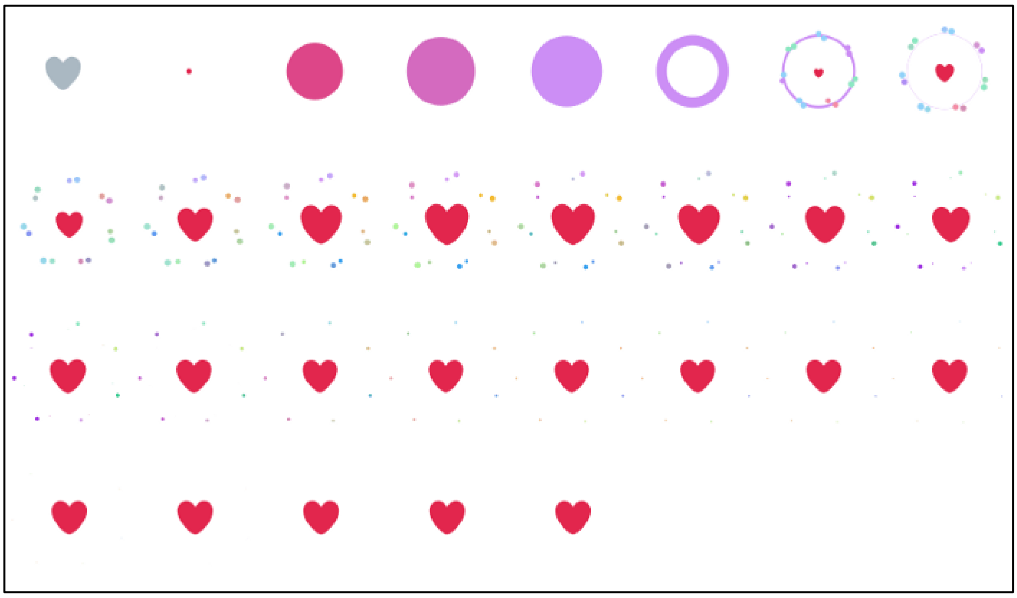


FIGURE 2: Image behind a sprite animation of Twitter's like button.

SVG sprites are used for displaying an icon from a large icon set, instead of icon fonts (Nikhil 2019, Cited 22.4.2021). They can be used for 'microinteraction' animations on social media, as in figure 2. Complex, or long sprite animations can produce prohibitively large files when designed for smooth, 30 or 60 frames per second (fps) playback. SVG sprites can be prepared with any vector graphical editor, and they can be animated with CSS or JavaScript (Drasner 2017, 39-42).

Transformations, as the other main way of animating SVG, represent manipulations of various properties of the SVG. They can be made in two ways: by **coding**, or by using **graphical tools**.

3.2 Approaches to SVG transformations

On top of the work browsers do to initially display an SVG, the file itself or the website's scripts may contain instructions for rotation, scaling, translation, or skewing properties of the file's elements to change over time. In that case the browser recalculates the values for those properties and initiates a process to implement the new values into the appearance of the page (Irish and Lewis 2013, Cited 18.3.2021). This may happen once before each frame, often sixty or more times per second.

SVG elements were originally envisioned to be manipulated with **Synchronized Multimedia Integration Language (SMIL).** SMIL is an easy way of achieving well-performing animations of DOM elements inside SVG. Chrome developers suspended its support several years ago in favour of CSS and WAAPI, but they have re-established it afterwards (MDN 2021, Cited 18.4.2021). Since then, it has been popularized among experts as not the most secure way of animating. (Drasner 2017, 21-37).

Due to having a DOM, SVG elements can be animated easily with **CSS**, **JavaScript**, and numerous **JavaScript libraries** (Drasner 2016, Cited 10.2.2021). CSS is a great native method for simple animations like toggling element states, whereas JavaScript is slightly more complex for developers to write but offers much more power over the elements.

Web Animations Application Programming Interface (WAAPI) is a native specification of W3C and a working recommendation based on JavaScript, meant to bridge the gap between the performance of CSS and flexibility of JavaScript. It allows access to the browser's animation engine, ensuring the best performance of any technology, without the need for hacks (Schot 2021, Cited 18.4.2021). Upon the release of Safari's version 13.1, WAAPI became supported natively in all "evergreen" browsers (Schiemann 2020, Cited 19.4.2021). However, it requires a polyfill to be viewed in older browsers, which is detrimental to performance (MDN 2021, Cited 18.4.2021).

Based on various articles written by web development experts, **Green Sock Animation Platform 3 (GSAP3)** is one of the favourite ways of animating user interfaces on the web. GSAP3 is the second fastest-growing JavaScript library on the internet (W3Tech 2021, Cited 23.2.2021). It offers a massive set of functionalities for free, but it also has special functions as plugins that require payment. Other popular libraries include: **Three.js, Mo.js, Snap.svg, SVG.js, jQuery, and React.js.**

There are different ways of animating SVG for the web without the need for coding. One of the most prominent ways, which designers are often familiar with, is to use **Adobe After Effects** to animate vector shape layers, and a plugin called Bodymovin to export the animation into a JSON file. Then, the JSON file can be included as a link on the website, along with a small library called **Lottie.js**, which was made by Hernan Torrisi but popularized by Airbnb. The implementation takes the shape of a simple call of a function with several attributes (Airbnb, Cited 23.2.2021). Lottie.js serves as a player that renders JSON files from After Effects natively for the web, iOS, Android, React Native, and Windows. It exists since 2017, accumulating a decent number of supported After Effects features for exporting.

SVGator is a powerful and easy method of animating SVG, also created in 2017, but without the need for any additional solutions (Friedman 2018, Cited 22.4.2021). It is a web-based graphical animation platform, simple to learn, and popular (Reed 2020, Cited 22.4.2021). It can produce complex SVG animations that have embedded interactive functionality and are minimal in size. There are also many other graphical SVG animators, such as **Haiku Animator, SpiritApp, Keyshape, SVGcircus**, **Snap.svg Animator**, and more.

4 PRE-ANIMATION STAGE

Upon the author becoming involved with the project of Animating Poe, there was already a concept of how Poe looks like, seen in figure 3, and what his background story is. Poe is Attactive's virtual user experience expert who appears from voids and interacts with websites. It is believed Poe comes from a planet of deep serenity and he is here to help humanity reach a new plane of existence. (Attractive 2021, Cited 24.2.2021.) He is made of hexagons arranged in an abstract shape and coloured in three of the brand colours: red, yellow, and orange. When internet users are viewing a website, the concept of behaviour on the website comes down to motion, with a bit of attitude, culture, personality, and context instilled (Saffer 2010, 44). The only motion of Poe in the beginning was hexagons slowly moving from one side of him to the other, through the middle, resembling the motion associated with breathing. He gives the general appearance of calmness and composure, which are traits one might associate with expertise. The high number of hexagons in combination with their overlapping effects can be associated with complexity, which is perceived in popular culture as a trait of intelligent systems. The high transparency might remind users that Poe is not entirely a part of the same solid reality we inhabit and has connection to the digital and perhaps other worlds. As the currently singular animation of Poe is not supposed to entirely resemble his final look, there is opportunity for the author to use creativity in freshening his 'breathing' animation and animating him for new scenarios.

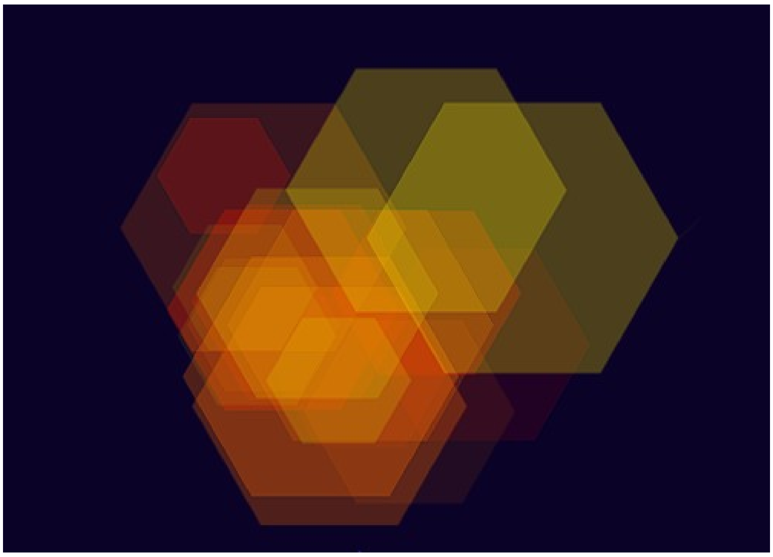


FIGURE 3: Screenshot of Poe's breathing prior to author's involvement

4.1 Design constraints

Design constraints are imposed or physical limitations of a design or the development process (Leffingwell & Widrig 2003, 298, 299). Websites impose constraints on user actions in the form of physical limits or psychological techniques (Redaelli 2013, Cited 16.4.2021). These constraints are like “guide rails”, ensuring the user navigates efficiently and avoids errors, by having certain elements of the website emphasized, scaled down, disabled, or entirely hidden (Philips 2017, Cited 16.4.2021). Making a successful website seems to require good understanding of users’ needs and business goals, so unnecessary clutter can be eliminated, and only essential features kept at every step of the user’s journey. Other design constraints affect the technologists and designers, and may come from requirements, regulations, deadlines, budget, manpower, skill levels, tools, style guidelines, platforms, et cetera (Bowers 2020, Cited 16.4.2021).

Time and skill - The author’s thesis submission deadline is 23rd of April 2021. Likewise, Attractive’s ambition for updating their website is by the end of April 2021 (Brower discussion, 18.4.2021). This leaves roughly eleven weeks for the author to board the project, learn to animate, and produce animations for online use. Attractive’s web technologists and experts can help with areas that the author has little or no experience with, like coding or implementation of logic for interaction, etc. However, due to the author being the sole person tasked with animating Poe, preference is given to independent working, which creates other constraints.

Animation - A perhaps obvious constraint is that Poe is better off convincing users he is a UX expert if he moves in time as opposed to being a static image. Humans have been interested in making static pictures move since over 35,000 years ago, as images that walk, talk, or think are objects of fascination for us (Williams 2001, 11-18).

Placement - Another somewhat obvious constraint is that Poe’s animations are best placed where they add value. Poe is the face of Attractive’s brand and can be introduced early in the user’s journey through the website, right on the homepage. In the concept document for Attractive’s website, Poe is depicted as “breathing” in the background of the headline. He is covered slightly by the headline and the text box which is the main interaction point to which users are guided to click on and type their web address into. (Kallio, 2021). When the user clicks on the text box, Poe should come from behind the box to its side and appear bigger and closer, as in figure 4. Later, when the user typed his website address and pressed enter, Poe can appear busily working to give the user something to look at while waiting for the analysis. Assuming Poe is well-animated and not obstructing the experience with latency or choppiness, he can add value in these places by increasing engagement of first-time visitors, emphasizing the personality of the website and style of the brand, and by being aesthetically pleasing (Singh 2020, Cited 17.4.2021). His movements could encapsulate his otherworldliness, and he can appear true to his character: calm, friendly, insightful, and unconventional.

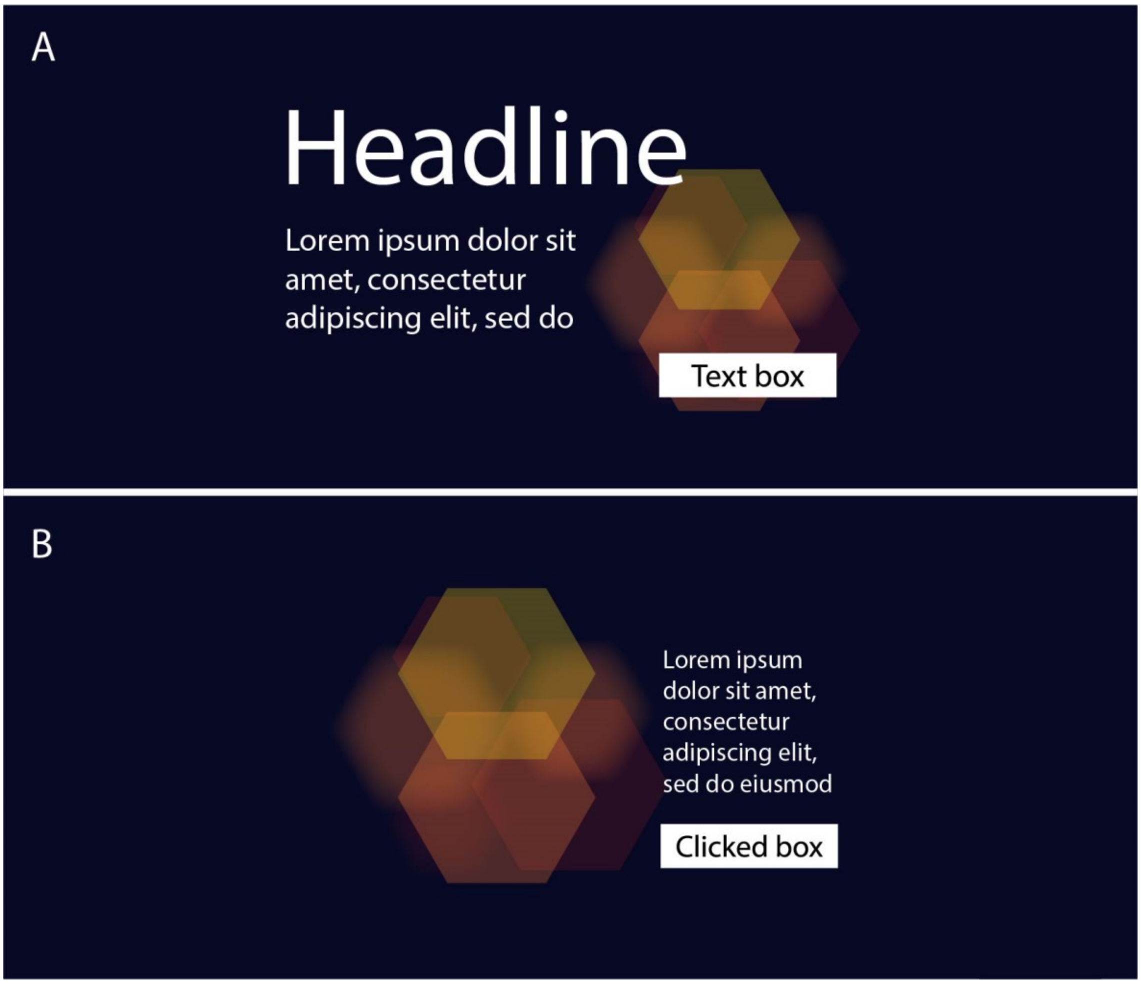

A
Headline
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
Text box
B
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
Clicked box

FIGURE 4: Author’s rough mock-up of the vision for conceived states Poe takes on the homepage.

Interactivity - The user’s click on the text box on the main page triggers Poe’s movement. Here, a technical constraint becomes apparent - objects in the website’s DOM need to be manipulated based on certain logic. JavaScript or CSS could be used for the implementation of the movement, as the author is familiar with their basics. These tools could be used as the sole way of animating Poe, or just for cleverly weaving together different animation files which contain different stages of the movement, and which could be made with another tool.

Psychology - Poe should not take too much attention away from the explanatory headline and important guidance elements. The concept of the homepage features Poe in the background, with a few high-contrast elements above and in front of him, to which the eye of the user is drawn. Bright colours attract the user’s attention, and high-contrast text decreases the cognitive load for reading, which saves the reader time in going through it (U.S. Department of Health and Human Services 2006, 101, 105). As Poe is placed near key elements, he can comparatively have low contrast. His movement could be smooth and gradual, so it does not draw the user’s attention, as animations generally do that (Weinschenk 2011, 5). It is perhaps advantageous to make Poe blurry while in the background, so to lessen contrast even further.

File format - Pages should load quickly to lower the probabilities of a negative user experience or a bad first impression of the company or its offering (Singh 2020, Cited 17.4.2021). This is paramount for Attractive’s website, so the team agreed SVG should be the file format for delivering animations. Despite having experience only in video animations, the apparent variety of tools and methods for animating SVG made this constraint entirely reasonable for the author to work with. Poe’s appearance with his simple geometric shapes plays well into the strengths of the SVG format when compared to GIF or others, as described in chapter three.

Playback and duration – Poe should appear continuously moving, which can be achieved by starting the animation with the loading of the page and repeating it in a loop. The restarting can be designed to be seamless. The duration of the loop can be short to save space and load faster, but not too short for the motion to become predictable and boring to the user. Roughly thirty seconds is reasonable, ensuring small file size as well as enough time to exhibit certain behaviours. The behaviours can start with the loading of the page, so to engage the user right away.

File size – Smaller files take less time to get transferred across the internet, so file size is important for quicker loading times. An optimized version of a relatively simple, 10-second sprite animation created in Adobe Animate was 12 megabytes when testing, because there were 300 graphics stored in it, 30 for each second. That file would add a full second to the loading time on a 100mbps internet connection. Animating with transformations produced a file of around 200 kilobytes. Using SVG transformations is the way to smallest file size, however, that eliminates many of the available tools for SVG animation that focus on sprites. Additionally, the physical scale of the SVG file is chosen to be 1920x1080 pixels, as that is the resolution of author's working computer. SVG format allows it to be changed quite easily, according to later needs.

Performance - Internet users have come to expect fast, responsive, and smooth websites, so browsers must be able to render content at consistently high framerates to ensure good experience (Singh 2020, Cited 17.4.2021). Each transforming property of an SVG has a unique performance cost attached to it, based on how much processing is required to do it. Rotation, position, scale, and opacity are the cheapest properties, and filters such as blur are some of the most expensive (Irish and Lewis 2013, Cited 18.3.2021). Processing translates to time, as less powerful devices can take longer to process transformations. When working with complex animations, in which more than a dozen hexagons and perhaps other elements move simultaneously, only devices with processing power above a certain level will be able to provide smooth playback (Emmanuel 2018, Cited 18.3.2021). An aspirational constraint is to aim for smooth playback on the first phone with a 64-bit processor, iPhone 5s from 2013, which could be achieved by animating only cheap properties, but not all simultaneously. However, if blur is to be used, an efficient way would be to have the blurred elements stagnant for the duration of the loop.

Cost – The main cost comes from software that the author could use to produce animations efficiently. There is no explicit budget constraint, but lower cost is advantageous. Adobe After Effects, while very popular, is not designed for web animations. It requires use of Lottie library that supports only some features and exports files in JSON format (Airbnb, Cited 22.4.2021). Its price for 12 months of business use is 340 euros (Adobe, Cited 22.4.2021). That is more than double of Haiku Animator and SVGator, which are specifically designed for web animations with SVG. SVGator is 120 euros and entirely online (SVGator, Cited 22.4.2021). Haiku Animator requires installation and is slightly more expensive. JavaScript libraries provide some or all features for free, but GSAP3's licensing with a full suite of features for all Attractive's future needs would cost 82 euros (GreenSock, Cited 22.4.2021).

Tools – Popularity of SMIL has gradually been increasing since 2016 and has now amassed above 1 percent usage for some of its functions, as evident from the data of all page loads on the Chrome platform (Google, Cited 19.4.2021). However, developers of Chromium browsers have not yet published an updated reference on its future support. It is best avoided in this case, as newer technologies are present. WAAPI would be the most performant way to animate, combining the best of JavaScript and CSS, but its features are not supported in all browsers, and it is necessary to use performance-impeding polyfills in older browsers (Schot 2021, Cited 18.4.2021). Making Poe's animations solely with CSS and JavaScript, while being the next most efficient method, can become messy in case of longer and more complex animations, and would increase the probability of browser inconsistencies, requiring debugging, which would add time to the process (Irish and Lewis 2013, Cited 18.3.2021). Among the plethora of other tools for SVG animation, SVGator is chosen due to its ease of use and comprehensive features at a reasonable price. Its more complex animations can be exported with embedded JavaScript, and simpler with embedded CSS. Although the software allows for basic interactivity like starting and stopping on load, click, hover, and scroll – it does not support pausing the animation loop to reposition the elements, and smoothly resuming. Free features of GSAP3 could be used for accomplishing that. While including a library is generally not the most performance-friendly act, GSAP3's minified file is already broadly cached when using a CDN that it could fit into the performance requirements. GSAP3 also has a large community and plentiful resources for learning, and is designed to ensure consistency across browsers, minimizing time the author would invest into learning and implementing it.

Constraints and design decisions are **circular** and iterative, as current constraints guide us to make decisions which may produce new constraints. This process requires understanding of the goal and proper management, as it can lead to endless iterating, especially when new technology is introduced, making previous constraints obsolete. (Leffingwell & Widrig 2003, 268-269.)

4.2 Design sprint

Design sprint is a method of solving problems by designing, prototyping, and testing ideas with customers for maximum real-world impact and minimum investment. It is a particular way of applying design thinking and lean development philosophies and was popularized by Google. It encompasses a five-day process with the following stages:

1. Clarifying the goal and gathering information.
2. Combining old and new inspiration into drawings.
3. Making decisions and storyboarding for a prototype.
4. Building a mock-up and a working prototype.
5. Collecting insights from customers and learning.

The method is ideally suited to big problems with high-stakes situations and a pressing deadline, where multidisciplinary people from the company would be engaged, including the chief officers. (Knapp, et al. 2016, 5-16, 24, 27.) At the end of one sprint, the team can start another with new knowledge and goals, cycling through the process until a satisfying solution is created. In this thesis project, the scope of the design sprints is much smaller, and the process is used only for making the prototypes, and not the final animations.

Prior to starting the sprint process with Attractive, a period of about two weeks was allowed for the author to learn about SVG animations and explore and assess different animation tools. Two test animations are made during this period, with the aim to appear identical as the original Attractive's homepage animation. Trial versions of various software were installed, and the author's personal subscription of Adobe After Effects used for one of the test animations. The version closest to the original animation comes from SVGator, the software specifically made for SVG animations that is also easy to use and available in a free plan.

The process then starts with purchasing the more powerful professional tier of SVGator and understanding requirements. Initial goal is set to create an animation of Poe in an interactive situation and see what he is like so we can learn from that for future versions (Brower discussion, 1.2.2021). The best inspiration is Microsoft's Clippit Office assistant - being just a paper clip, yet so expressive and interesting. An idea of an interactive animation where Poe is surprised by the user's click and reacts in shock is chosen as a start, as his expressiveness can be pushed far. The process of ideating and prototyping starts on the second day, with a pen, paper cut-outs, and SVGator for prototyping. The most scattered of the imagined looks seems most appropriate. Prototyping is where the process slows down, due to author's lack of experience in implementing animations online. A prototype animation is best if implemented online, as browser performance and implementation techniques can be explored early. A server space is set up and JavaScript is used to position and display two animation files from SVGator on one blank website. After a bit of exploration and help with the site's logic, the animation works as planned, and six frames from it are shown in figure 5. While this design sprint resembles Google's outlined process, it is not facilitated nor structured, and feedback is gathered without real users. However, the finished animation is shown to the team and a circle of author's friends, generating the following feedback:

1. Brightness should not be animated due to accessibility concerns and questionable appeal.
2. The maximum distance Poe's hexagons can move away from him should be limited.
3. Poe scales poorly, so scaling can be avoided, and fading can be introduced to conceal it.
4. Appearance and disappearance can be smoother if it lasts less than 100 milliseconds.
5. Combining animations online can be smoother using JavaScript arrays, GSAP3, or by designing the seams to be unnoticeable.

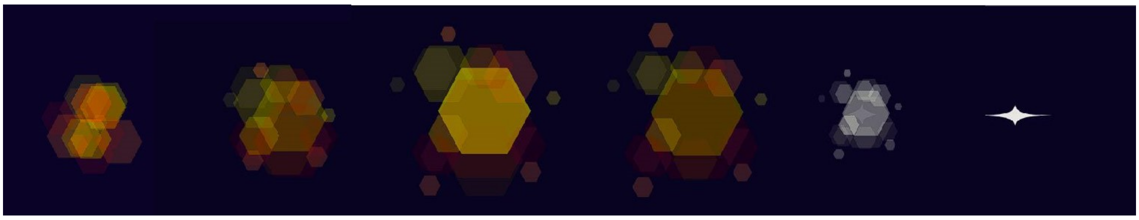


FIGURE 5: Stages of Poe after a click - jumping in shock, shaking, paling, and disappearing.

Another cycle of the sprint process has the goal of exploring Poe's appearance from nothing and assembly into the shape of a magnifier. There is accent on attempting to simulate the third dimension with blur filters. The same sprint stages are cycled through to produce an animation shown in figure 6. This time the following conclusions are reached:

1. Blur filters are a good proxy for appearance of depth.
2. Performance suffers if all elements animate blurriness concurrently.
3. A repository can be used to track different versions of animations.

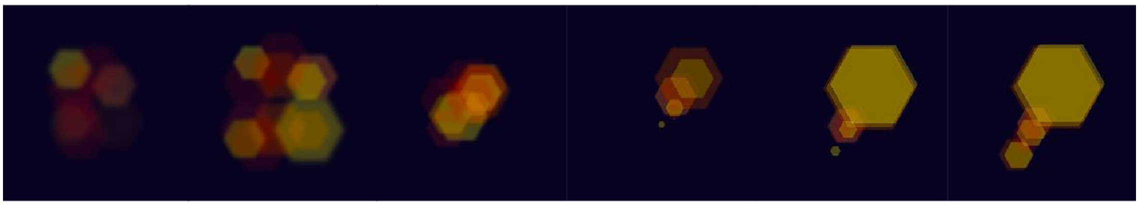


FIGURE 6: Poe in different stages of appearing from nothing and taking shape of a magnifier.

4.3 Developing the visions

The next task is to develop a more specific vision for the animations that are aimed to be used on Attractive's main website. The first animation is from the homepage, of Poe breathing, and the second is from the 'loading' page, of Poe analysing. Poe's breathing requires three separate animations of different complexities for Intern, Advisor, and Expert pricing plans. In this stage of the process most work is done independently, as restrictions in Oulu are not permitting the author to visit Attractive's premises. Regular online communication and sharing of ideas and implementations with Attractive's CTO is fruitful to the process and generates feedback.

Poe's breathing is interesting to ideate upon, as it is not specified in the concept. There are ideas to use blur, and to have the hexagons move slowly. Upon gathering inspiration, ideating, and sketching, the vision of Poe resembling qualities of a lava lamp became appealing. Lava lamps are made to be looked at, so they look very interesting. The slow and formless movements of the lamp's contents are also perceived as somewhat mysterious for most, which would be a suitable trait for Poe. To add flamboyancy to the serene motion, the author envisions each hexagon shrinking, growing, and occasionally bursting into nothingness, setting off smooth chain reactions in doing so. After the burst, some surrounding hexagons could "absorb" the released energy and grow, while a new, small hexagon can appear in the previous one's place - resembling explosions of stars in the universe. The breathing of Intern Poe could be drastically simplified, with only two hexagons orbiting each other and appearing to do it three dimensionally with clever use of scaling and blurs. The three versions of Poe's breathing are illustrated in figure 7.

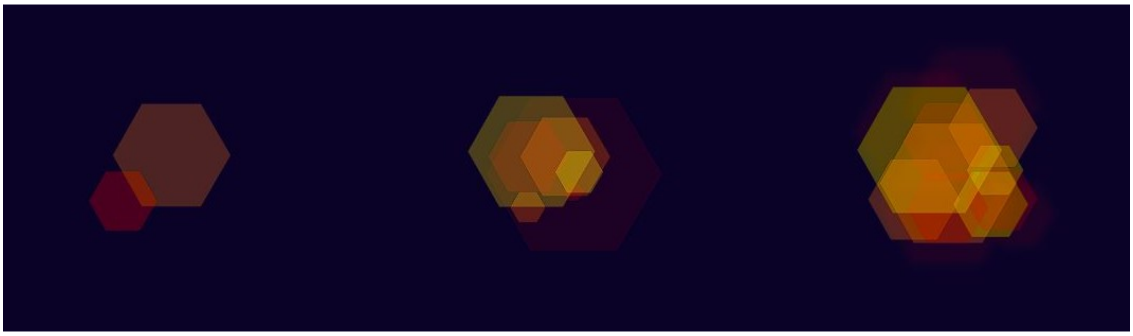
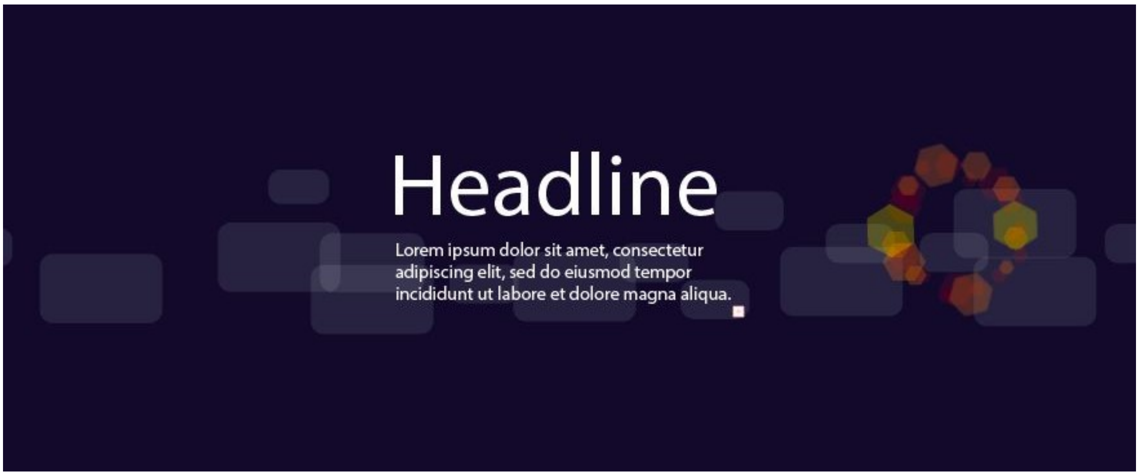


FIGURE 7: Poe's Intern, Advisor, and Expert mock-ups.

For Poe's analysing animation, the goals are to convey a message to the user that his request is being processed, and to provide him with something interesting to watch while he awaits completion (Brower discussion, 18.2.2021). The concept document describes Poe taking a circular shape and moving around rectangles of different sizes that fly from one end of the screen to the other, representing a stream of webpages, as shown in figure 8 (Kallio, 2021). Poe could stop at a certain window, analyse it, drop it back into the stream, and continue to the next window - repeating the process.



Headline

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

FIGURE 8: Author's rough mock-up of the page with Poe analysing, based on Attractive's concept.

The first inspiration for Poe's motion during analysis is a circular missile target searching system, as it seems to be geometrically simple and yet clearly convey a message of something being analysed. This is universally appealing across the team and will be a behaviour Poe does after taking a flying window. To move between windows, Poe can be true to his story and appear from voids, which can benefit performance as there would be less movement of Poe's many hexagons. Inspiration of hexagons appearing from nothing is taken from the animation film Cloudy with a Chance of Meatballs 2. To signify intelligent processing, many small white dots can flicker rapidly, resembling digital "walls" of ones and zeros which are seen in many films - of which the most prominent one is Matrix. Some inspiring concepts are shown in figure 9.

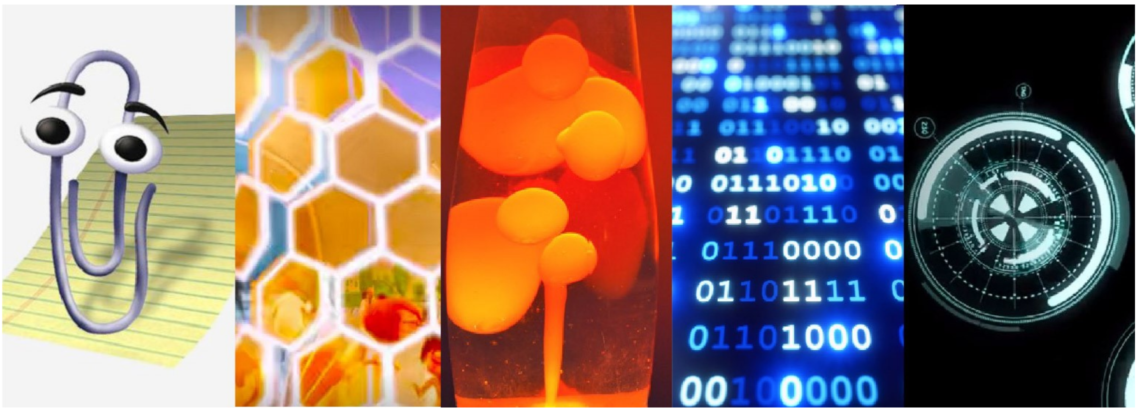


FIGURE 9: Images of inspiration for Poe's behaviours.

The vision for the animation of Poe analysing has potential to be complex and may benefit from turning it into a story. Upon ideating, discussing, and deciding on concepts, the following guidance story is produced, written for the perspective of the user:

“All the pages of your website are streaming through this digital landscape, when suddenly, from the fabric of digital reality, hexagons start appearing. They peel from the background, and start spinning. They are joined by more hexagons, and now you can be confident you are witnessing Poe’s arrival to help tell you ways to improve your website. He notices a particular page in the stream, and in one swooping motion plucks it out. He clicks on it to open it and starts processing away. Light glows from the page as he inspects and evaluates it. He then quickly closes the window and disappears into nothing, leaving behind only fading traces of a hexagon… But he appears again, peeling off from a place further away, spinning up, and grabbing a different page to analyse. You are in awe of the comprehensive analysis Poe keeps on doing, as he inspects new pages “.

The process of envisioning these animations has been a slimmer version of a design sprint with the author mostly cycling through the inspiration, decision, and mock-up stage. Some decisions created new constraints and needed to be iterated upon again. A view of three iterations of Poe’s analysis behaviour is shown in figure 10.

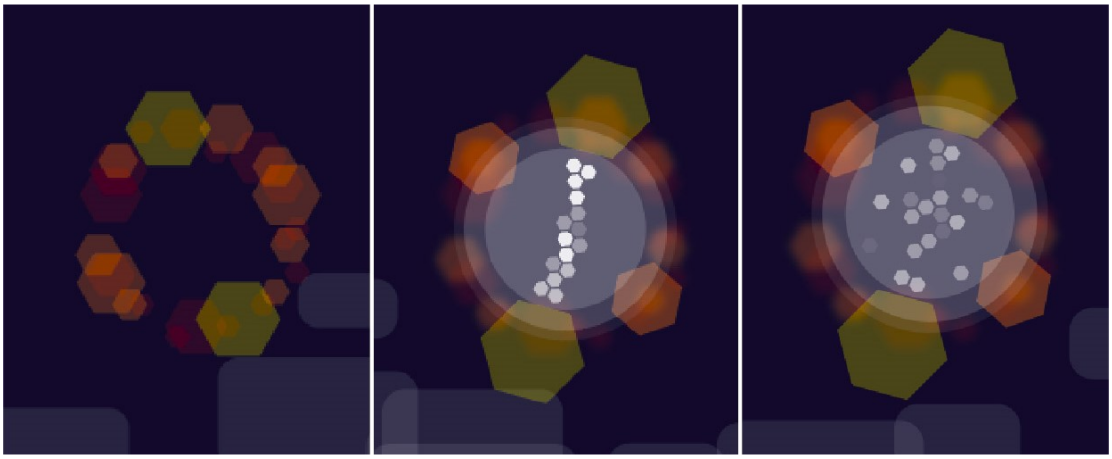


FIGURE 10: The evolution of Poe's analysing behaviour.

5 ANIMATION STAGE

This chapter is intended to describe the author's process of producing the previously outlined animation of Poe analysing (appendix 1). Poe analysing is chosen as it features a story-like scenario and is more comprehensive than Poe breathing. Due to the relative ease of use and plentiful online resources for SVGator, instructions are omitted, and focus is dedicated instead to the design stages and challenges in production. Instructions for SVGator can be found on their YouTube channel, reachable via the following link: https://www.youtube.com/c/SVGator/videos

Poe analysing animation is suitable for being split into two files: the background's rectangles can be an independent SVG animation exported with CSS, while the complex motion of Poe will require exporting with JavaScript. This approach decreases the total performance cost but requires combining Poe and his background in the implementation stage.

5.1 Part 1: Stream of windows

The concept design of the page with Poe analysing only shows there are windows in the background, but omits details such as direction, speed, relative motion, and other effects. As it is a standard in most countries to read from left to right, it was tempting to make the windows fly from left to right. However, most modern European software has a 'back' button pointing to the left, so it is best if the windows would come from the forward, or future direction (right side), and move towards the back, or past (left side). Early in the process of ideating, a realization was reached that a parallax effect is important to show between windows, as it can add a sense of depth to the scene and make the background seem more interesting and modern. The windows are split into groups of three sizes: large, medium, and small. The large ones are to appear the closest and therefore move the fastest, with medium and small ones being progressively slower, as to appear more distant. The three sets are each grouped and animated to move left at a steady speed, as shown in figure 11. The animation should loop seamlessly, so the layout of the windows in the first frame is duplicated to the last frame, upon which the animation is shortened by one frame at the end.

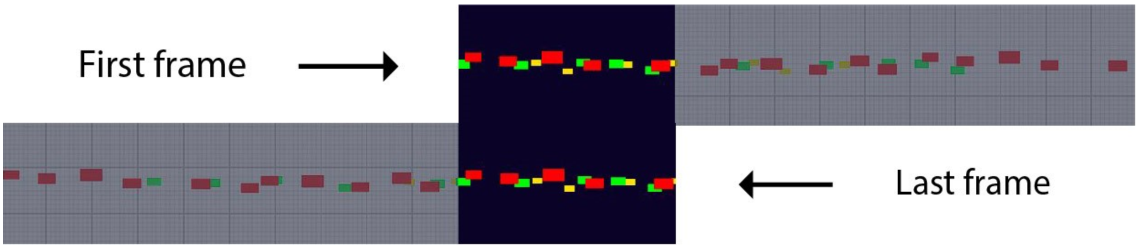


FIGURE 11: Zoomed out view of SVGator artboard, with red being the fastest, green the middle, and yellow being the slowest moving windows in the scene.

While animating Poe's movements at a later stage, it becomes clear there are too many windows to begin with, so some are deleted to avoid clutter. Additionally, to make Poe appear as grabbing some of the windows, the author needs to revisit this stage after Poe's analysis behaviour is animated. The action of grabbing a window requires precise adjustment of the rectangle's path to move in synchronization with Poe's movement during the swooping gesture, as shown in figure 12. The affected windows are made to disappear after analysis, as otherwise a smooth loop of windows would be difficult to achieve.

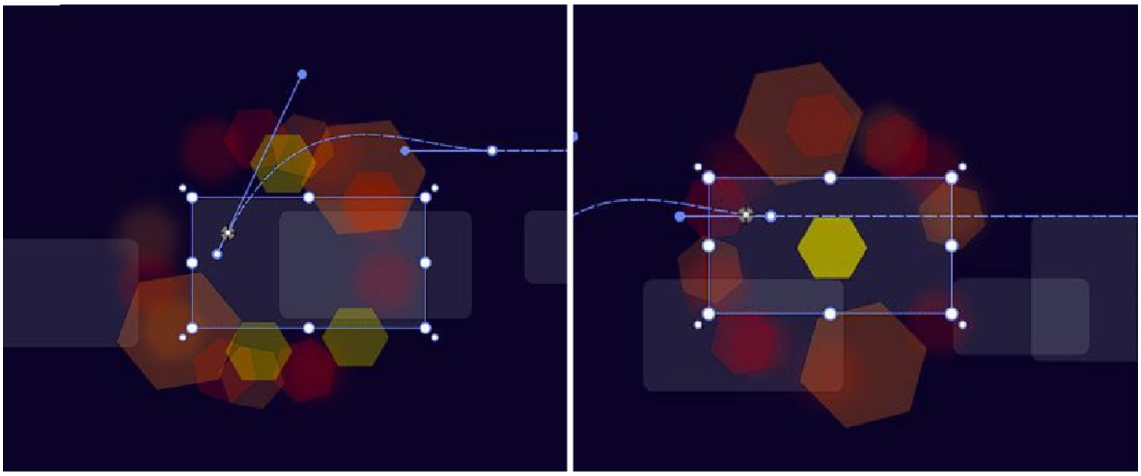
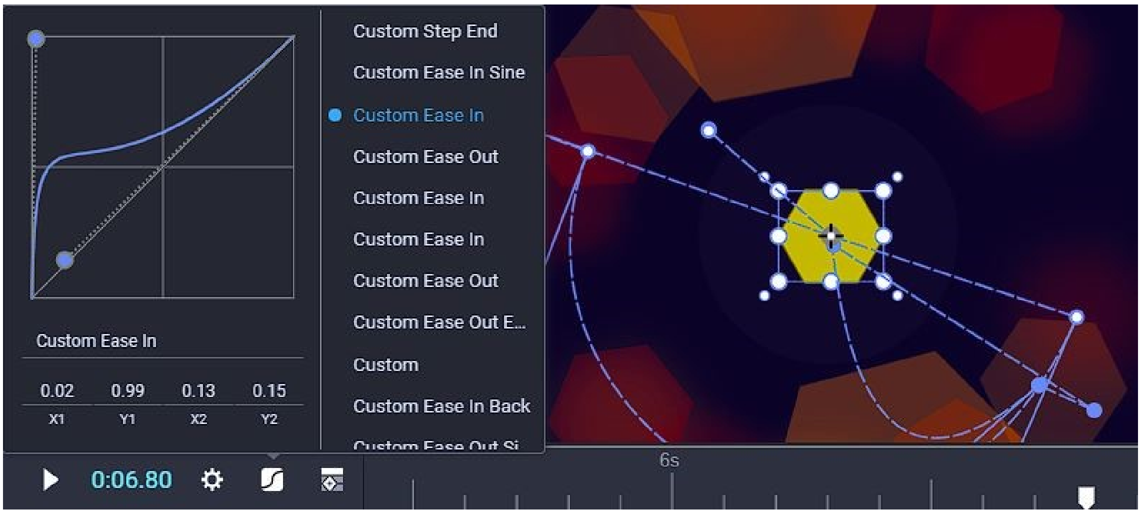


FIGURE 12: Two frames from Poe's swooping with the motion path of grabbed window in view.

5.2 Part 2: Poe's analysis behaviour

Poe's analysing part begins with him capturing a window and clicking on it. To capture and click, a set of three yellow hexagons is used in synchronisation, converging to the centre to signify the "hold" position. Animating Poe's capturing gesture increased the prominence of the suddenness of its start, so a sharp increase in opacity of two large orange hexagons is made under a second prior to initiating the capturing motion. Upon capturing the window, Poe appears to click on it by having his central yellow hexagon grow in scale and quickly shrink with an easing as shown in figure 13. The growth is used to build anticipation, like raising a finger above a button raises the anticipation of pressing it. The clicking action is two-fold, with the large orange hexagons squeezed toward each other with same easing applied, just for a moment before the yellow hexagon's press action starts. This achieves a more appealing look overall, as if Poe's clicking starts from his extremities.



Custom Ease In

X1	Y1	X2	Y2
0.02	0.99	0.13	0.15

FIGURE 13: Easing settings for Poe's yellow hexagons during the pressing action.

An early inspiration for the analysis behaviour was an intelligent missile targeting system, which is achieved with Poe by moving two sets of hexagons slower than the rest, in changing and opposite directions from each other, as illustrated in figure 14. The hexagons look like they are in sync because they are animated on the group level, for the duration of the analysis. Instead of positions of each element, the only animated property is the rotation of the group.

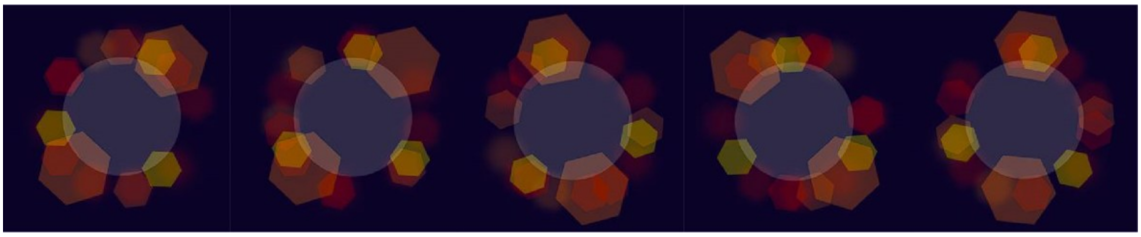


FIGURE 14: Poe's synced groups of hexagons rotating in alternate directions.

As Poe’s click on the window comes to an end, together with the yellow hexagons, a white semi-transparent circle is revealed, adding to the effect of the click, and continuing to pulse as a light that would be coming from each action Poe makes. A pulsating light effect is achieved by having two circles, one of which is fixed in scale for the duration of the pulsing, and the other increasing in sync with a fading opacity. As soon as the opacity reaches zero and the scale its maximum, the outer circle is reverted to original scale and immediately starts with normal opacity to scale up again, as illustrated in figure 15.

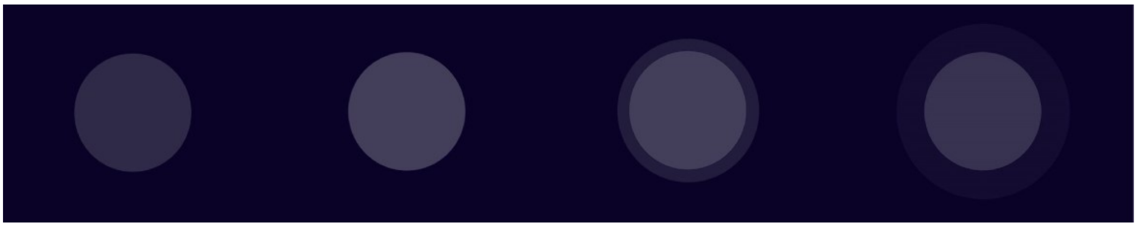


FIGURE 15: Screenshots of pulsing effect with two circles, isolated from other elements.

The effect that signifies intelligent analysis most convincingly is achieved with over 70 miniscule white hexagons grouped into six groups, which are animated at the group level to change in opacity. Each of the hexagons in each of the groups also individually varies in its own fixed opacity. The resulting effect is shown in two frames of Poe in figure 16. The document structure in figure 16 shows all elements of Poe that are used during the analysing part. The total number of elements just in this structure is around one hundred.

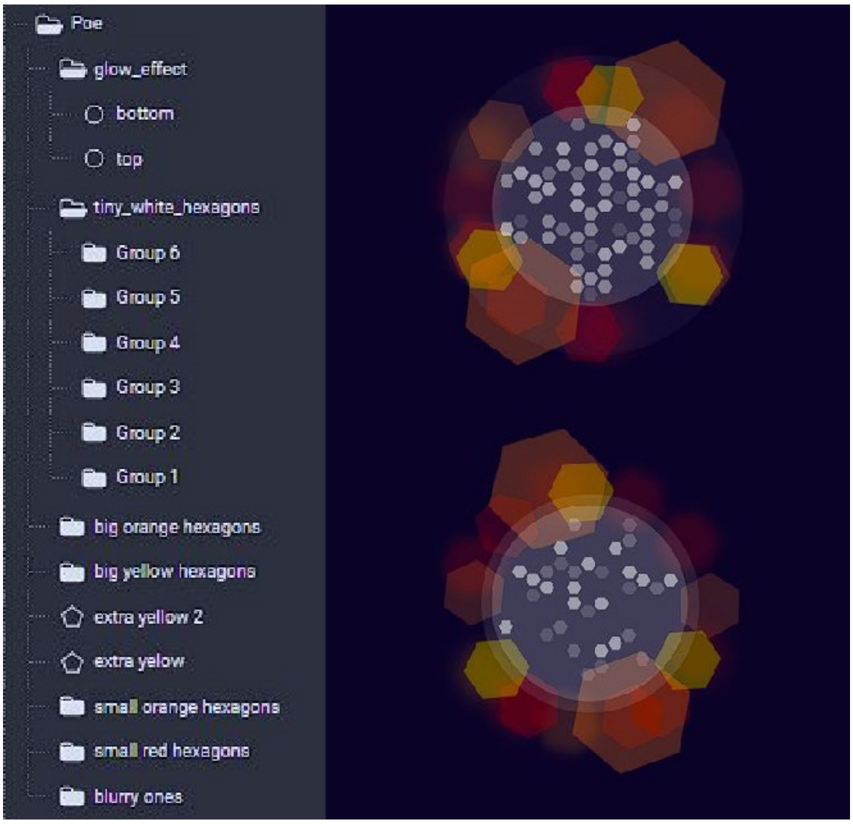


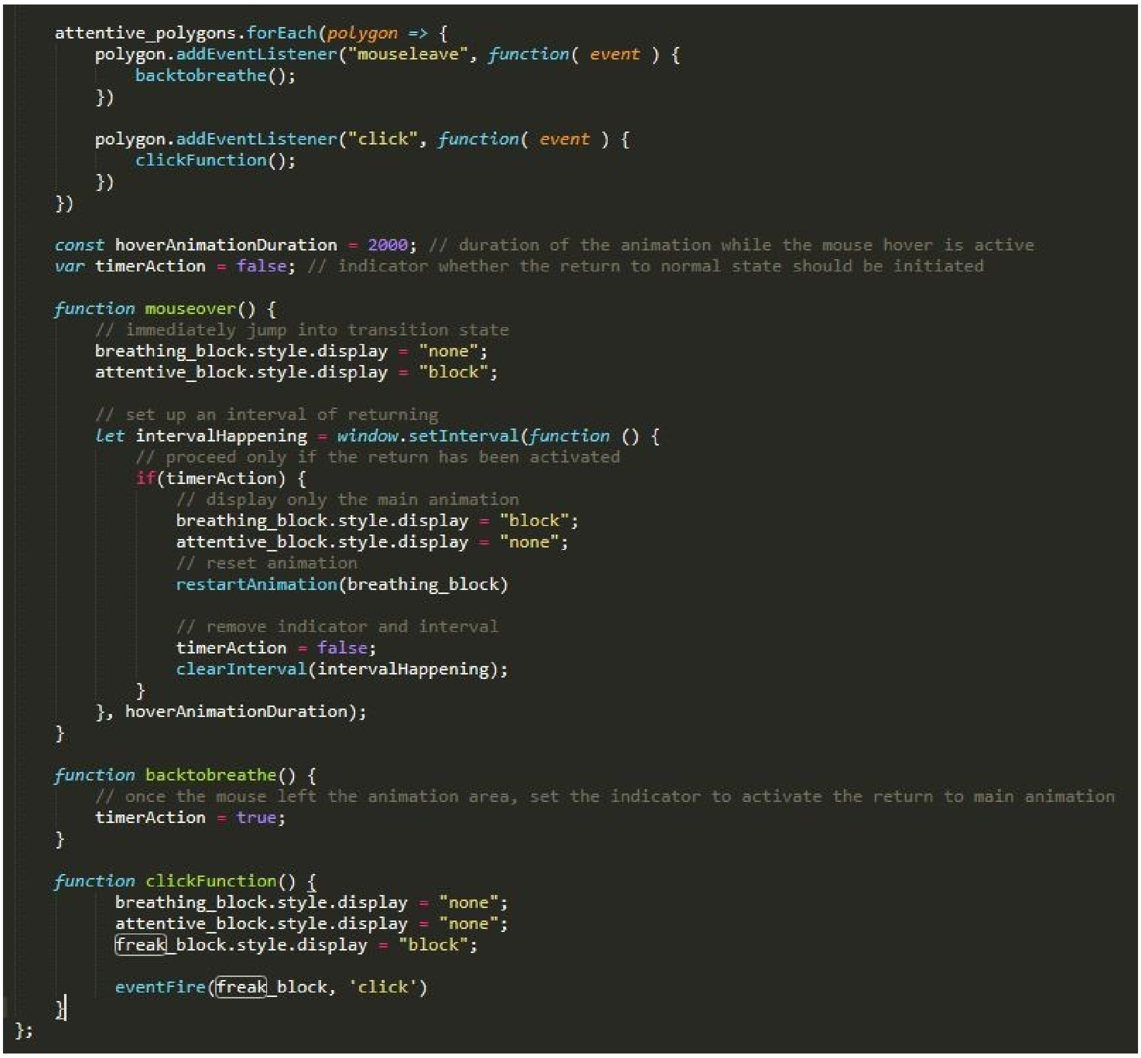
FIGURE 16: Poe’s document structure and two frames from the analysing behaviour.

The animations from the two main steps can be integrated seamlessly by animating the entirety of Poe. Adding position and opacity of the grouped Poe into his timeline is enough to get him moving across the space and disappearing and reappearing as needed. For best effects, motions can happen in accented arcs, and with prominent easings such as in figure 13 from before. SVGator allows different elements to be hidden at exporting, so separate files could be made from a single document, one containing only the animated background and being CSS-based, and the other only Poe and being based on JavaScript.

6 IMPLEMENTATION

Upon completion of Poe's prototypes in the exploration stage of design sprints, the animations are tied together with JavaScript to enable basic and limited interaction. The files are then uploaded on a hosted web server for sharing and testing. They contain minimal styles and scripts and are not production ready. Implementation is an important stage of the design process as weaving together of different animation files made in SVGator can be tested. Other useful things can be done as well, like overlaying two animation files to make an appearance of one integrated animation. That can be done simply by adding "*position:absolute*" in the style of the SVG.

An example of implementation of an interactive animation with load, click, and hover events is to have Poe start breathing when the site loads, change to a different animation of him jumping slightly when a mouse hovers over him, and then if a mouse clicks, change to a third animation where he freaks out. This interactive animation is recorded and can be seen here: https://youtu.be/uzEEAMT-kPU. The three SVG files are embedded into the HTML, and are referred to as variables *breathing_block, attentive_block*, and *freak_block*, in the script declaration. Page loading takes place with executing a function in which initial states of all the blocks are set. All the elements of each block, which are individual SVG polygons, are grouped using a custom DOM selector. Further actions are performed on these groups, instead of individual polygons, ensuring better accuracy in selection and positioning. Changing the states of animations is done by altering "*style.display"* property between "*display*" and "*none*" values. A timer is used to ensure that the animation initiated by hovering completely plays out before resetting, in case the mouse exits the group before the animation would normally finish. With some additional functions shown in figure 17, it is possible to successfully test interactivity of different simple animations designed with SVGator.



```
    attentive_polygons.forEach(polygon => {
        polygon.addEventListener("mouseleave", function( event ) {
            backtobreathe();
        })

        polygon.addEventListener("click", function( event ) {
            clickFunction();
        })
    })

    const hoverAnimationDuration = 2000; // duration of the animation while the mouse hover is active
    var timerAction = false; // indicator whether the return to normal state should be initiated

    function mouseover() {
        // immediately jump into transition state
        breathing_block.style.display = "none";
        attentive_block.style.display = "block";

        // set up an interval of returning
        let intervalHappening = window.setInterval(function () {
            // proceed only if the return has been activated
            if(timerAction) {
                // display only the main animation
                breathing_block.style.display = "block";
                attentive_block.style.display = "none";
                // reset animation
                restartAnimation(breathing_block)

                // remove indicator and interval
                timerAction = false;
                clearInterval(intervalHappening);
            }
        }, hoverAnimationDuration);
    }

    function backtobreathe() {
        // once the mouse left the animation area, set the indicator to activate the return to main animation
        timerAction = true;
    }

    function clickFunction() {
          breathing_block.style.display = "none";
          attentive_block.style.display = "none";
          freak_block.style.display = "block";

          eventFire(freak_block, 'click')
    }
};
```

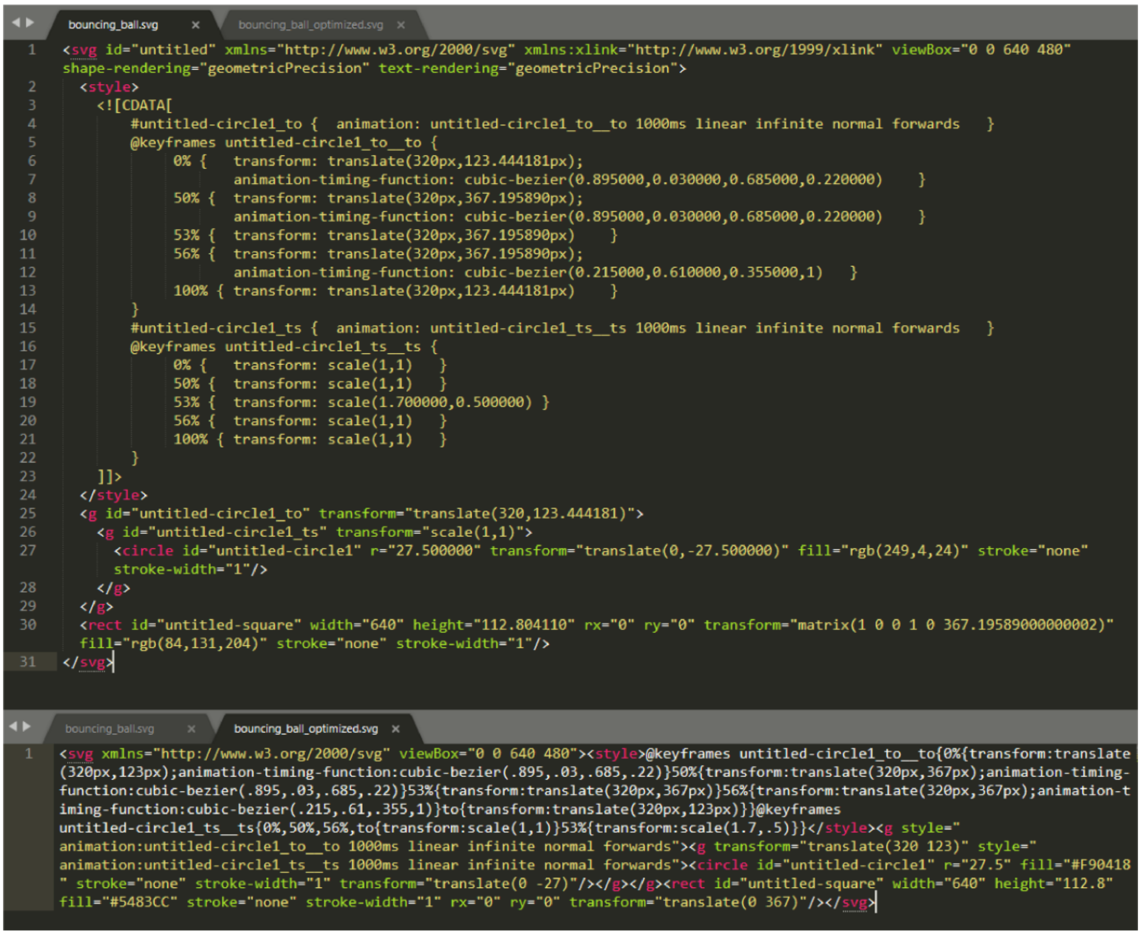
FIGURE 17: A part of the script from author's HTML file for testing.

7 OPTIMIZATION

According to Attractive, trillions of dollars in potential earnings are lost due to poor user experience on websites. On the BBC website, for every additional second a page took to load, 10 percent of users left (Clark 2018, Cited 17.3.2021). If animations are lagging while playing, the image of the brand can suffer. With more complexity and features being included in modern websites, the need for fast loading is greater than ever. Optimizing SVG animations goes a long way to offering smoother website experiences, as frame rates can increase while file sizes drop significantly, all with very little effort. Optimization is especially important when working with more complex, longer animations, such as “Poe analysing”, of which a 10-second part is shown in appendix 1.

SVG animations, even if generally smaller than video animations, can still contain unnecessary information depending how small their original file has been. It is good practice before animating to delete unnecessary shapes and simplify paths in Illustrator or Inkscape. This practice significantly affects size, especially when animating complex scenes. SVG format’s version 1.1 is best choice when exporting the document, as it has most browser support. (Drasner 2017, 30.)

SVG animations can be optimized with minifying tools such as the SVG-optimizer by Peter Collingridge, or SVGOMG by Jake Archibald. The process may require tweaking of the settings depending on properties being animated and needed precision, so animations can be seen breaking along the way. The original SVG file can be put through SVGOMG before animating. It is best to experiment with different combinations of the file’s actual dimension and the SVGOMG’s setting for decimal precision, to find the smallest file size (Schwartz 2015, Cited 18.3.2021). Generally, files put through these programs decrease in size by significant amounts. (Souedian 2015, Cited 22.2.2021.) Figure 18 shows a comparison of the markup of the “Bouncing ball” animation (figure 19) before and after being minified. The file size before minification process was 1762 bytes, and only 1039 bytes after.



```
<svg id="untitled" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" viewBox="0 0 640 480"
shape-rendering="geometricPrecision" text-rendering="geometricPrecision">
  <style>
    <![CDATA[
        #untitled-circle1_to {  animation: untitled-circle1_to__to 1000ms linear infinite normal forwards   }
        @keyframes untitled-circle1_to__to {
             0% {   transform: translate(320px,123.444181px);
                    animation-timing-function: cubic-bezier(0.895000,0.030000,0.685000,0.220000)    }
             50% {  transform: translate(320px,367.195890px);
                    animation-timing-function: cubic-bezier(0.895000,0.030000,0.685000,0.220000)    }
             53% {  transform: translate(320px,367.195890px)    }
             56% {  transform: translate(320px,367.195890px);
                    animation-timing-function: cubic-bezier(0.215000,0.610000,0.355000,1)   }
             100% { transform: translate(320px,123.444181px)    }
        }
        #untitled-circle1_ts {  animation: untitled-circle1_ts__ts 1000ms linear infinite normal forwards   }
        @keyframes untitled-circle1_ts__ts {
             0% {   transform: scale(1,1)   }
             50% {  transform: scale(1,1)   }
             53% {  transform: scale(1.700000,0.500000) }
             56% {  transform: scale(1,1)   }
             100% { transform: scale(1,1)   }
        }
    ]]>
  </style>
  <g id="untitled-circle1_to" transform="translate(320,123.444181)">
    <g id="untitled-circle1_ts" transform="scale(1,1)">
      <circle id="untitled-circle1" r="27.500000" transform="translate(0,-27.500000)" fill="rgb(249,4,24)" stroke="none"
      stroke-width="1"/>
    </g>
  </g>
  <rect id="untitled-square" width="640" height="112.804110" rx="0" ry="0" transform="matrix(1 0 0 1 0 367.19589000000002)"
  fill="rgb(84,131,204)" stroke="none" stroke-width="1"/>
</svg>
```

```
<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 640 480"><style>@keyframes untitled-circle1_to__to{0%{transform:translate
(320px,123px);animation-timing-function:cubic-bezier(.895,.03,.685,.22)}50%{transform:translate(320px,367px);animation-timing-
function:cubic-bezier(.895,.03,.685,.22)}53%{transform:translate(320px,367px)}56%{transform:translate(320px,367px);animation-t
iming-function:cubic-bezier(.215,.61,.355,1)}to{transform:translate(320px,123px)}}@keyframes
untitled-circle1_ts__ts{0%,50%,56%,to{transform:scale(1,1)}53%{transform:scale(1.7,.5)}}</style><g style="
animation:untitled-circle1_to__to 1000ms linear infinite normal forwards"><g transform="translate(320 123)" style="
animation:untitled-circle1_ts__ts 1000ms linear infinite normal forwards"><circle id="untitled-circle1" r="27.5" fill="#F90418
" stroke="none" stroke-width="1" transform="translate(0 -27)"/></g></g><rect id="untitled-square" width="640" height="112.8"
fill="#5483CC" stroke="none" stroke-width="1" rx="0" ry="0" transform="translate(0 367)"/></svg>
```

FIGURE 18: Comparison of "Bouncing ball" animation's markup before minification (upper part) and after (lower part).

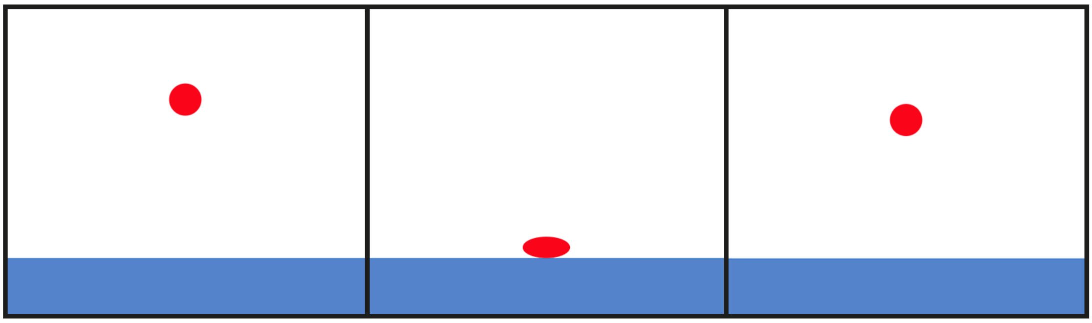
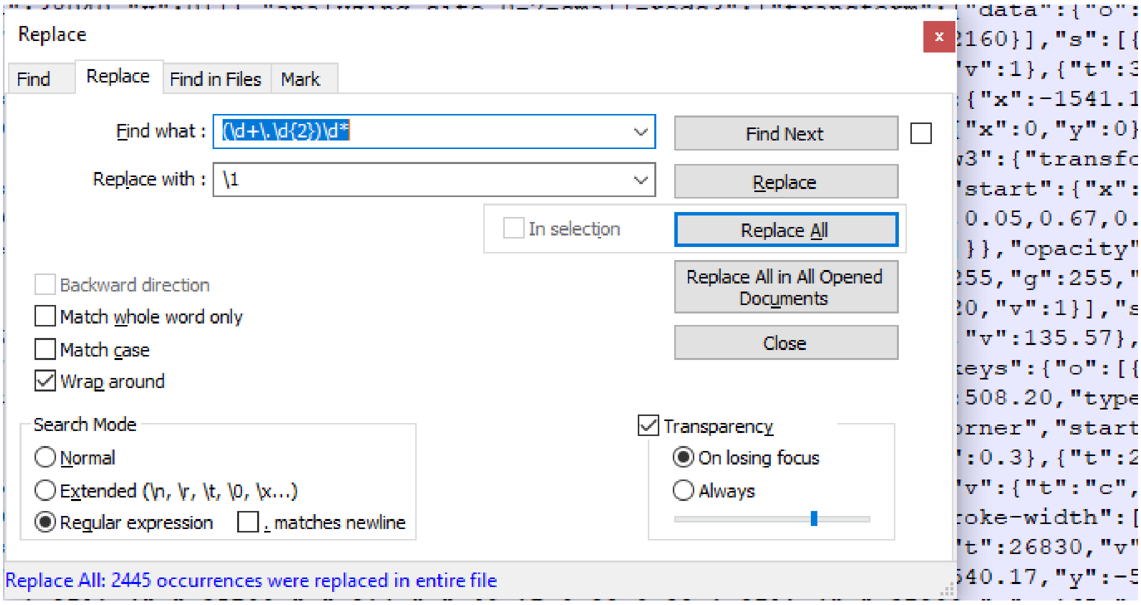


FIGURE 19: "Bouncing ball" animation shown in three frames.

It is noteworthy that decimal rounding of SVGOMG only affects numbers in the SVG syntax, and not ones found in the inline scripts that the SVG may contain when animated. As SVGator exports

CSS or JavaScript animation code inline, within the SVG, there can be thousands of instances with too many decimal places left after optimization. For rounding those numbers and improving performance as well as minimizing size, a regular expression can be used with “Find and replace” in Notepad++, as seen in figure 20.



Find what : (\d+\.\d{2})\d*

Replace with : \1

☐ In selection

☐ Backward direction
☐ Match whole word only
☐ Match case
☑ Wrap around

Search Mode
○ Normal
○ Extended (\n, \r, \t, \0, \x...)
● Regular expression ☐ . matches newline

☑ Transparency
● On losing focus
○ Always

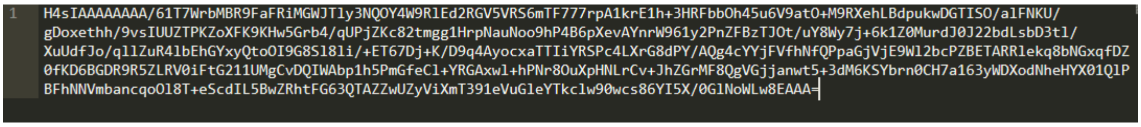
Replace All: 2445 occurrences were replaced in entire file

FIGURE 20 - Screenshot of Notepad++ Find and Replace window, with regular expressions used on “Poe analysing” markup to round 2445 numbers with more than two decimal places.

Animations can be made to take maximum advantage of browser and device functionalities, therefore requiring less time to display, which makes complex animations appear smoother. Most modern browsers render websites at 60 fps, leaving 16 milliseconds between frames for any changes to be made. The processing steps most browsers can take between frames are: recalculating styles of changed elements, laying out the restyled elements, repainting them onto the screen, and, if the painting was done in layers, compositing those layers into a single view (Irish and Lewis 2013, Cited 18.3.2021). Every animated property of an SVG has a ‘performance cost’ attached to it, based on the amount of browser processing and number of steps needed to display it. Due to the way browsers work, and the use of hardware acceleration on devices to aid in displaying on screens, the cheapest properties to animate are rotation, position, and scale; with opacity being on that list only when elements affected by it are in the same layer (Irish and Lewis 2013, Cited 18.3.2021). Most browsers are made to manipulate these properties well, and many devices can accelerate their displaying. It is good practice to use these properties for most of the work, and to choreograph timings of transformations so everything does not start or run at the same time (Sharma 2016, Cited 17.3.2021). It is also good practice to abstain from animating expensive properties such as filters.

SVG has rendering properties, such as color-, text-, shape-, and image-rendering. Each of these can be set to “optimizeSpeed”, which may improve performance, but also lead to inferior quality of rendered shapes, and therefore should be used with testing (W3C 2011, Cited 18.3.2021).

When SVG are animated with inline CSS within the SVG, the file should be included as a CSS background-image property or an HTML <img> tag. This method of inclusion offers best performance and the benefit of being cached by browsers. In this case the biggest downside is the lack of interactivity, so usage is limited to just decorative, non-interactive animations. SMIL could be used instead of CSS in this case, to preserve interactivity, but that brings its own issues described in chapter 4. An alternative that does not offer browser caching but unlocks full potential of SVG is embedding the SVG markup inline, in the HTML. This may take lots of space in the DOM and the site’s code may appear messy. Other ways of inclusion, namely <object>, <embed>, and <iframe>, are of suboptimal performance. (Hunt 2020, Cited 17.3.2021.) If not embedded in the site, it is good if animation files are compressed with algorithms like Brotli or Gzip, because browsers decompress files rapidly enough that the faster network transfer due to smaller file size usually saves loading time. Compression of the “bouncing ball” animation resulted in a file of 434 bytes (figure 21), compared to the uncompressed, optimized version that was 1039 bytes.



```
H4sIAAAAAAAA/61T7WrbMBR9FaFRiMGWJTly3NQOY4W9RlEd2RGV5VRS6mTF777rpA1krE1h+3HRFbbOh45u6V9atO+M9RXehLBdpukwDGTISO/alFNKU/
gDoxethh/9vsIUUZTPKZoXFK9KHw5Grb4/qUPjZKc82tmgg1HrpNauNoo9hP4B6pXevAYnrW961y2PnZFBzTJOt/uY8Wy7j+6k1Z0MurdJ0J22bdLsbD3tl/
XuUdfJo/qllZuR4lbEhGYxyQtoOI9G8Sl8li/+ET67Dj+K/D9q4AyocxaTTIiYRSPc4LXrG8dPY/AQg4cYYjFVfhNfQPpaGjVjE9Wl2bcPZBETARRlekq8bNGxqfDZ
0fKD6BGDR9R5ZLRV0iFtG211UMgCvDQIWAbp1h5PmGfeCl+YRGAxwl+hPNr8OuXpHNLrCv+JhZGrMF8QgVGjjanwt5+3dM6KSYbrn0CH7a163yWDXodNheHYX01Q1P
BFhNNVmbancqoOl8T+eScdIL5BwZRhtFG63QTAZZwUZyViXmT391eVuGleYTkclw90wcs86YI5X/0GlNoWLw8EAAA=
```

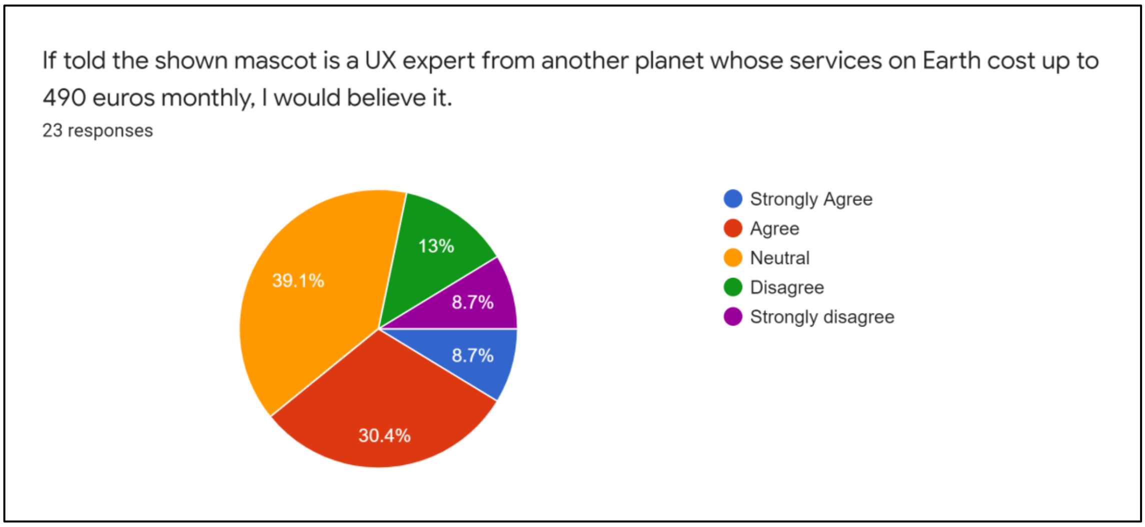
FIGURE 21: markup of the compressed "Bouncing ball" animation file.

If SVG are animated with JavaScript libraries, the libraries are best included at the end of the HTML <body> tag, and never inside the <head> tag, as any JavaScript in a website is dealt with by the browser only after HTML and CSS (Emmanuel 2018, Cited 18.3.2021). This way, the core parts of the site can load quicker, without interruptions of loading and executing scripts.

8 CONCLUSIONS

Author shows interest in many areas, so the process of writing this report could last for much longer if there are no deadlines. Most of the bases for this report come from previous research, due to the implementation of Attractive's website updates being published only after the process of writing has been completed (https://attractive.ai/en). However, many good answers are discovered through previous literature, answering sufficiently most questions posed in the beginning of the process.

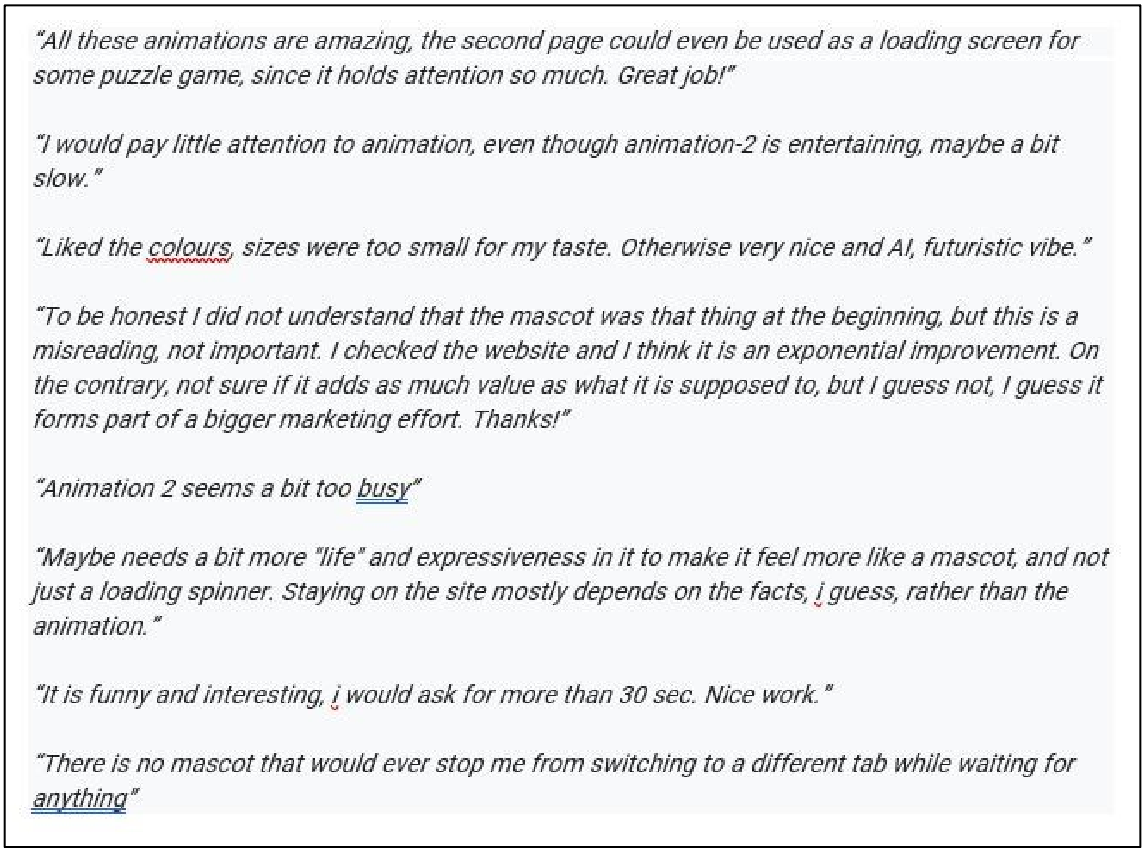
A survey is conducted of workers in the web design and development field to replace analytical metrics that were to be collected directly from users of Attractive's website. Of twenty-three respondents, eighteen may fall into Attractive's target audience. All twenty-three reported smooth playing of the animations on their device, meaning the measures taken to ensure good performance are sufficient. Eighty percent would describe Poe as 'intelligent', more than a third as 'calm', and nineteen percent as 'alien'. Achieving these associations is a great result as they correspond to what Poe's character is aimed to appear like. After seeing the animations, most (16 participants) would not believe Poe is a UX expert from another planet whose services are valued at up to 490 euros monthly, as seen in figure 22. The believability of Poe's competency is lower than that of his personality, requiring further work. Ratings of Poe's quality and behaviour average 3.65 on a scale from 1 (poor) to 5 (delightful). While waiting on a website, more than half of the respondents would rather stay on the site looking at Poe analysing animation than switching to another tab in the browser. The visual and engaging quality of Poe's animations can be improved, as evident also in responses from the participants, shown in figure 23. By combining both the quantitative and qualitative results, Poe is considered to affect the overall experience positively.



If told the shown mascot is a UX expert from another planet whose services on Earth cost up to 490 euros monthly, I would believe it.

23 responses

FIGURE 22: Screenshot of author’s Google Form results from the believability question.



“All these animations are amazing, the second page could even be used as a loading screen for some puzzle game, since it holds attention so much. Great job!”

“I would pay little attention to animation, even though animation-2 is entertaining, maybe a bit slow.”

“Liked the colours, sizes were too small for my taste. Otherwise very nice and AI, futuristic vibe.”

“To be honest I did not understand that the mascot was that thing at the beginning, but this is a misreading, not important. I checked the website and I think it is an exponential improvement. On the contrary, not sure if it adds as much value as what it is supposed to, but I guess not, I guess it forms part of a bigger marketing effort. Thanks!”

“Animation 2 seems a bit too busy”

“Maybe needs a bit more "life" and expressiveness in it to make it feel more like a mascot, and not just a loading spinner. Staying on the site mostly depends on the facts, i guess, rather than the animation.”

“It is funny and interesting, i would ask for more than 30 sec. Nice work.”

“There is no mascot that would ever stop me from switching to a different tab while waiting for anything”

FIGURE 23: Textual responses in the survey, written by eight of the respondents.

Attractive accepted author’s animations and the team gained much insight from them for future updates. The time allowed for production of the animation was sufficient to deliver four animations of sufficiently high fidelity and complexity for use in storytelling. The main two animations are seen on the following YouTube links: Poe analysing – https://youtu.be/aon1lBUHxXY, and Poe breathing – https://youtu.be/NDbpi6aOHBI.

The process of bringing a character to life is not strictly defined nor necessarily the same for each character. Inexperienced designers could cycle through stages indefinitely. Deadlines and other constraints help push the products out the door and into the world, where, again, they may be returned to and iterated upon at a later stage. Expertise and experience are valuable, but with the internet landscapes changing constantly, they become smaller factors in successful company stories. Interdisciplinary teams seem to have become the standard for reaching good results in the internet industry, and a generalist mindset seems to be good for individuals who are not certain in which section they want to work in.

The complexity of the World Wide Web is high, and many areas and questions lie unexplored at the end of the author’s journey. Browser workings, specifications, testing processes, and other relevant topics have only been touched upon, while version control and Dev Tools, lie completely unexplored. However, with WAAPI gaining full support and availability of SVGator-like tools for designers, time is right for newcomers of all backgrounds to explore the web animations field.

This cooperation with Attractive has been fruitful in targeted outcomes, learnings, and relations. By working with experts in the field of user experience, author’s exposure to the mindset of placing the customer in the centre translated into personal growth with lessons beyond just animations, web technologies, and the mascot himself. Having mentors and colleagues at Attractive, who shared their tools, methods, and attitudes, has been a great experience to undergo. Deep appreciation is expressed on part of the author. Despite imposed restrictions and delays in the development, the main goal of bringing a concoction of hexagons called Poe to life has been achieved. Now, the author asks of himself and readers, what new challenges are next, and where are the next lessons to be learned?

REFERENCES

Adobe. Creative Cloud Plans for Teams. Cited 22.4.2021, https://www.adobe.com/creativecloud/business-plans.html

Airbnb. Lottie. Getting started. Cited 23.2.2021, https://airbnb.io/lottie/#/web

Attractive Experiences Oy. 2021. The Attractive Team. Cited 24.2.2021, https://attractive.ai/en/team

Ballhaus, W., Bangah, C., Basu, R., Bunyan, D., Chow, W., DePonte, J., Gross, D., Kalra, R., Kramer, R., Munusamy, R., Papps, J., Sarkis, K., Stuart, C., Subianto, S., van Eeden, E. & Yau, C. 2020. Perspectives from the Global Entertainment & Media Outlook 2020–2024. Pulling the future forward: The entertainment and media industry reconfigures amid recovery. Cited 24.2.2021, https://www.pwc.com/gx/en/entertainment-media/outlook-2020/perspectives.pdf

Bowers, M. 2020. Design Constraints Are Not Restraints – They Stoke Creativity. Cited 16.4.2021, https://www.toptal.com/designers/ui/design-constraints

Brower, J. 2021. Chief Technology Officer. Attractive Experiences Oy. Discussion. 1.2.2021.

Brower, J. 2021. Chief Technology Officer. Attractive Experiences Oy. Discussion. 18.2.2021.

CGP Grey. 2020. Hexagons are the Bestagons. Video. Cited 31.1.2021, https://www.youtube.com/watch?v=thOifuHs6eY

Clark, M. 2018. How the BBC builds websites that scale. Cited 17.3.2021 https://www.creativebloq.com/features/how-the-bbc-builds-websites-that-scale

Coyier, C. 2009. CSS Sprites: What they are, why they're cool, and how to use them. Cited 17.3.2021 https://css-tricks.com/css-sprites/

Culhane, J. 1992. Disney's Aladdin. The making of an animated film. United States, Disney Editions

Dodson, J. 2008. The 'Powers' of Personification: Rhetorical Purpose in the 'Book of Wisdom' and the Letter to the Romans. Germany, De Gruyter.

Drasner, S. 2016. A comparison of animation technologies. CSS-Tricks.com. Cited 10.2.2021, https://css-tricks.com/comparison-animation-technologies/

Drasner, S. 2017. SVG Animations, from common UX Implementations to complex responsive animation. United States: O'Reilly Media.

Emmanuel, O. 2018. How browser rendering works – behind the scenes. Cited 18.3.2021 https://blog.logrocket.com/how-browser-rendering-works-behind-the-scenes-6782b0e8fb10/

Friedman, V. 2018. Animating SVG Files with SVGator. Cited 22.4.2021, https://www.smashingmagazine.com/2018/07/animating-svg-files-svgator/

Google. Chrome Platform Status. HTML & JavaScript usage > all features > timeline. Cited 19.4.2021, https://www.chromestatus.com/metrics/feature/timeline/popularity/501

GreenSock. Club. Cited 22.4.2021, https://greensock.com/club/

Grigorik, I. 2018. Choose the right image format. Cited 17.3.2021 https://web.dev/choose-the-right-image-format/

Heider, F. & Simmel, M. 1944 An Experimental Study of Apparent Behavior. The American Journal of Psychology, vol. 57, no. 2. Cited 7.2.2021, www.jstor.org/stable/1416950

Hunt, T. 2020. Improving SVG runtime performance. Cited 18.3.2021 https://codepen.io/tigt/post/improving-svg-rendering-performance

Irish, P. & Lewis, P. 2013. High performance Animations. Cited 18.3.2021 https://www.html5rocks.com/en/tutorials/speed/high-performance-animations/

Johnston, O. & Thomas, F. 1981. Disney Animation: the illusion of Life. United States, Walt Disney Productions.

Kallio, J. 2021. Attractive Experiences Oy. Attractive_2021_vision03.pdf. Internal source.

Knapp, J., Zeratsky, J. & Kowitz, B. 2016. Sprint. How to solve big problems and test new ideas in just five days. United States, Simon & Schuster.

Leffingwell, D. & Widrig, D. 2003. Managing Software Requirements: A Use Case Approach, Second Edition. United States, Pearson Education.

MDN 2021. SVG animation with SMIL. Cited 19.4.2021, https://developer.mozilla.org/en-US/docs/Web/SVG/SVG_animation_with_SMIL

MDN 2021. Using the Web Animations API. Cited 18.4.2021, https://developer.mozilla.org/en-US/docs/Web/API/Web_Animations_API/Using_the_Web_Animations_API

MDN Web Docs 2021. SVG: Scalable Vector Graphics. Cited 22.2.2021, https://developer.mozilla.org/en-US/docs/Web/SVG

Merriam-Webster 2021. "Behave", "Anthropomorphism". Cited 31.1.2021, https://www.merriam-webster.com/dictionary/behave , https://www.merriam-webster.com/dictionary/anthropomorphism

Miklos, C. 2020. Why it's time to focus on SVG animations – The story of SVGator. Cited 22.2.2021, https://uxplanet.org/why-its-time-to-focus-on-svg-animations-the-story-of-svgator-ac87ff2912e2

Nikhil, 2019. Icon Fonts vs SVG – Clash of the Icons. Cited 22.4.2021, https://www.lambdatest.com/blog/its-2019-lets-end-the-debate-on-icon-fonts-vs-svg-icons/

Philips, M. 2017. Boost Your UX with These Successful Interaction Design Principles. Cited 16.4.2021, https://www.toptal.com/designers/interactive/interaction-design-principles

Redaelli, M. 2013. Design fundamentals: Constraints. Cited 16.4.2021, https://www.webdesignerdepot.com/2013/02/design-fundamentals-constraints/

Reed, A. 2020. Animated SVG Tools to Save Your Time and Efforts. Cited 22.4.2021, https://designwebkit.com/web-development/svg-animation-tool-collection/

Research and markets. 2021. Global Animation, VFX & Video Games: Strategies, Trends & Opportunities (2021-25). Cited 24.2.2021, https://www.researchandmarkets.com/reports/5236328/global-animation-vfx-and-video-games?utm_source=dynamic&utm_medium=CI&utm_code=6k3gn8&utm_campaign=1342565+-+Global+Animation%2c+VFX+%26+Video+Games+Industry+Forecast%2c+2020-2025+-+The+Market+was+Valued+at+%24264+Billion+in+2019&utm_exec=joca220cid&utm_source=dynamic&utm_medium=CI&utm_code=6k3gn8

Saffer, D. 2010. Designing for Interaction, Second Edition: Creating Innovative Applications and Devices. United States, New Riders.

Schiemann, D. 2020. Web Animations API Now Supported in All Evergreen Browsers. Cited 19.4.2021, https://www.infoq.com/news/2020/06/web-animations-evergreen-browser/

Scholl, B. & Tremoullet, P. 2000. Perceptual causality and animacy. Trends in Cognitive Sciences – Vol. 4, No. 8, Elsevier. Cited 7.2.2021, http://perception.yale.edu/papers/00-Scholl-Tremoulet-TICS.pdf

Schot, N. 2021. An intro to animating with the Web Animations API. Cited 18.4.2021, https://simplabs.com/blog/2021/01/29/web-animations-intro/

Schwartz, R. 2015. Understanding and manually improving SVG optimization. Cited 18.3.2021 https://css-tricks.com/understanding-and-manually-improving-svg-optimization/

Sharma, A. 2016. The complete guide to getting 60fps animations with CSS. Cited 17.3.2021 https://blog.gyrosco.pe/smooth-css-animations-7d8ffc2c1d29

Singh, P. 2020. Ground Rules for Web Animations. Cited 17.4.2021, https://css-tricks.com/ground-rules-for-web-animations/

Soueidan, S. 2015. Tips for creating and exporting better SVGs for the web. Cited 22.2.2021, https://www.sarasoueidan.com/blog/svg-tips-for-designers/

SVGator. Start creating SVG animations for free - upgrade later based on your needs. Cited 22.4.2021, https://www.svgator.com/pricing

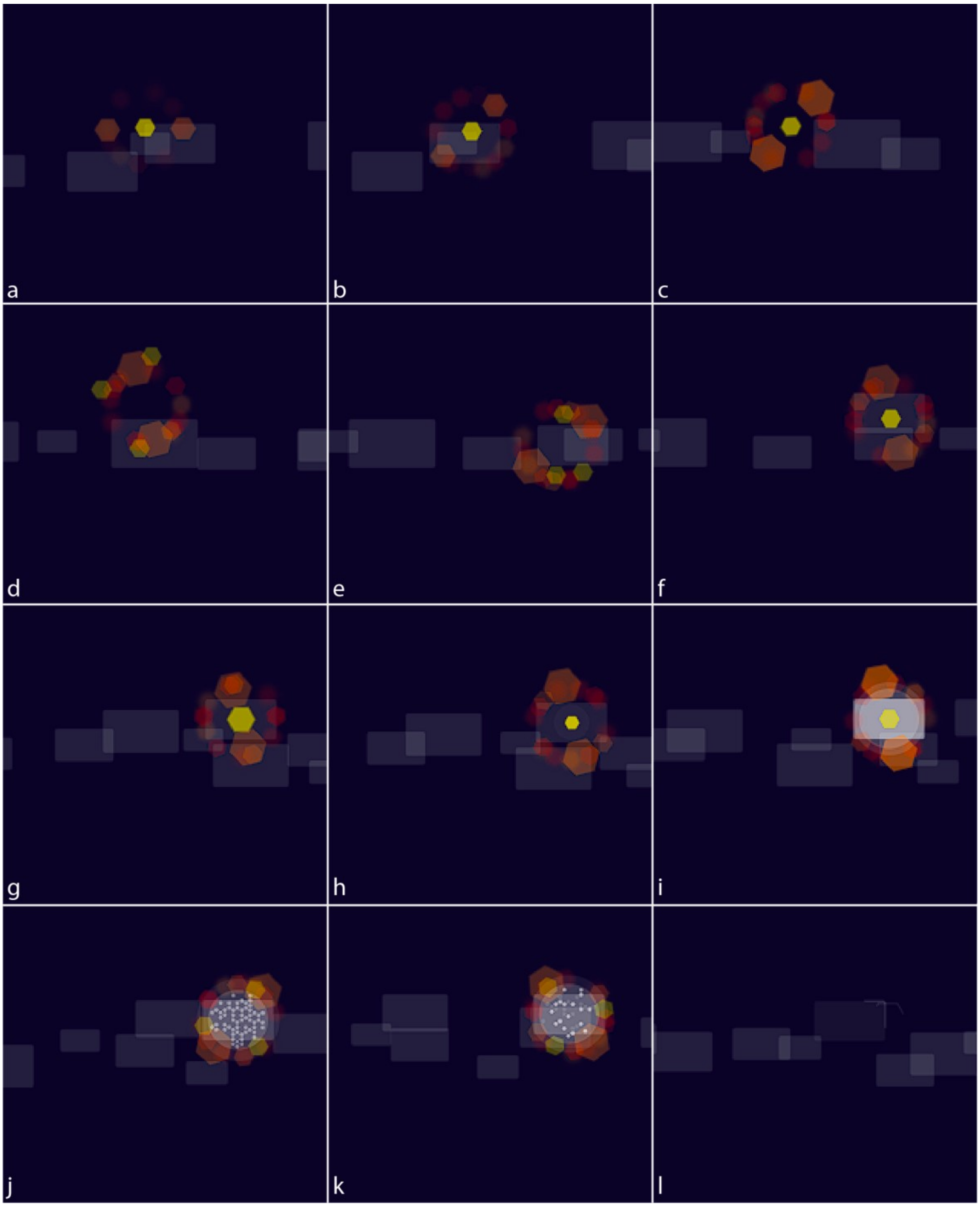
United States Department of Health and Human Services. 2006. Reseach-Based Web Design & Usability Guidelines. United States, Government Printing Office.

W3C, 2011. Scalable Vector Graphics (SVG) 1.1 (Second Edition) Cited 18.3.2021 https://www.w3.org/TR/SVG11/Overview.html

W3Tech, 2021. Historical yearly trends in the usage statistics of image file formats for websites. Cited 23.2.2021, https://w3techs.com/technologies/history_overview/image_format/all/y

Weinschenk, S. 2011. 100 Things Every Designer Needs to Know About People. United States, New Riders.

Williams, R. 2001. The Animator's Survival Kit. A manual of methods, principles, and formulas for classical, computer, game, stop motion, and internet animators. United Kingdom, Faber and Faber.


a
b
c
d
e
f
g
h
i
j
k
l