



## ON EFFECTIVE APPLICATIONS OF ANIMATION IN THE WEB

*Iegorova Iryna, professor, MST department, KhNure*

No web resource can deal without animation, and this is explained not only by the user's desire to work comfortably, but also by the need to visually track all actions that users can perform on this web resource. Elements of animation in interfaces take their usability to the next level, making them intuitive and accessible to the average user [1].

Web animation is a technique for creating the illusion of moving images that is used in development of various web projects. Animation in the Web can attract and hold attention longer than static images and can convey the idea or concept of a website more clearly and effectively.

Web animation should be smooth, meaningful and help the visitor in their experience with the product. UI/UX knowledge is a must for developer to help create an animation in such a way to stimulate the user perform intended actions.

Loading time is a critical indicator of effectiveness for any online resource, so perfectly chosen animation can make a significant impact on it.

At the very beginning of web design industry, developers actively used GIF and Flash animations [2]. GIFs were good for short animation loops, image sequences, and even short videos. But the GIF file format does not have variable transparency and does not support an alpha channel - so all pixels are either completely opaque or completely transparent. Flash animation creation became a simplified process that did not significantly increase the loading time of the web page. But at the same time, there was a significant drawback - a browser plug-in is required for Flash animation playback. Complexity of interaction with Flash animation, inability to work on all possible devices and lack of responsiveness to user requests led to the fact that the popularity of Flash animation fell sharply, pushing most of web developers to abandon the use of Flash.

Today's needs of web animation are different: the tools must be flexible and lightweight. The requirement to create responsive designs for different devices (desktop, tablet and mobile) needs to consider different pixel densities and screen sizes, device orientations, aspect ratios and viewing distances.

Currently, animation effects are implemented using a number of technologies, and CSS (cascading style sheets) in particular. CSS animation supports transition between different states using keyframes. CSS has indisputable advantages: a possibility to introduce changes to design of many pages at once, simple adaptive design using media queries, no external libraries required – everything is already built into the browser engine. But the difficulties of creating complex effects and mimic realistic movements limit the fields of application of CSS animation [3].

Canvas animation uses pixels in its operation. Animations created with Canvas have stable performance and impose no boundaries for creating truly complex objects and animations. Developer can stay focused when drawing even the most complex animation using Canvas. However, the disadvantage of this technology is the need



from browser to constantly redraw every element of interface and recalculate positions of every pixel.

WebGL is a web graphics library that provides a 3D graphics API without using third-party plugins [4]. WebGL is fully integrated into all modern standards, allowing hardware acceleration to process images and effects on the web application canvas. Animation for virtual reality is created with WebGL, which is then played at a speed of 60 frames per second. Many website developers use WebGL to create standing out visual effects. This web standard has a large catalog of visual effects, and 3D graphics greatly expand the possibilities to interact with it.

JavaScript (JS) animation offers more dynamism and flexibility than transitions or CSS-based animation and can be used to move DOM elements of a page [3]. JavaScript animation is widely used for bouncing, pausing, stopping, or slowing down animations. The performance of JavaScript heavily depends on the selected library.

React, Angular and Vue.js are modern and widely used JS libraries. All of them allow to develop not only animations but fully functional websites with backend interaction, user input validation, complex business logic, unique set of prebuilt as well as plethora of 3d-party components.

Motion UI is a Sass library [5] for creating own and unique CSS transitions and animations, allowing to instantly animate the application UI. Learning curve of Motion UI is very steep. With basic knowledge of CSS and JavaScript one can learn how to animate almost any element of the application empowering it with desired behavior. Motion UI combines all the advantages of CSS and JavaScript: transitions, shadows, bounces, loops – you name it.

Based on the discussed technologies, we can conclude that now there is number of technologies for creating animations, each of which has its own set of pros and cons, so a web developer has a lot to choose from.

It is advisable to take advantage of both CSS and JavaScript animations. CSS animations are used to create lightweight basic animations such as transitions, hover effects and non-interactive animations. JavaScript animations are commonly used to create dynamic elements. Using CSS for initial animations and JavaScript for user interaction or data-driven animations makes the entire website way more efficient.

Also, a correct proportion of required and heavyweight animations impact not only the loading time but the general effectiveness of a website.

#### References

1. Wojciechowski, J. (2023). The Impact of Animation in UI Design: From Delightful Microinteractions to Complex Transitions. <https://medium.com/@JakubWojciechowskiPL/the-impact-of-animation-in-ui-design-from-delightful-microinteractions-to-complex-transitions-76aac84336d6>.
2. A new age for animation. <https://www.adobe.com/products/animate.html>.
3. Ayebola, J. (2023). Web Animation Techniques – CSS vs JavaScript. <https://www.freecodecamp.org/news/web-animation-css-vs-javascript/>
4. WebGL Fundamentals. <https://webglfundamentals.org/>.
5. Motion UI library. <https://get.foundation/sites/docs/motion-ui.html>.