ISSN 1846-6168 (Print), ISSN 1848-5588 (Online)
https://doi.org/10.31803/tg-20230621113640

Original scientific paper
Received: 2023-06-21, Accepted: 2023-06-30

Technologies for Web Animations and their Positive and Negative Sides Regarding Web Page Metrics

Nikolina Stanić Loknar*, Tajana Koren Ivančević, Mateo Bekavac, Snježana Ivančić Valenko

Abstract: The development of web technologies has led to the availability of various multimedia content on the web. In addition to text, images, and videos, transitions, animations and interactivity are increasingly used. Transitions and animations present the content on the page in more interesting, attractive ways and can attract and retain users longer, improving the measurable parameters of page visits as well as the user experience. Animations on the web can be created in many ways. And if the animations are not executed well, they can slow down the page. The goal of this article is to compare the measurable parameters of animations created with different technologies. HTML, CSS, SVG, P5js, WebGL technologies and animations created with a 3D modeling tool were used. The article uses two different examples of animations created with different technologies and then tested on several devices.

Keywords: CSS; HTML; p5js; SVG; web animations; WebGL

1 INTRODUCTION

On today's web you can find very different web content and pages. From completely static pages to very dynamic and interactive pages. There are also pages that are fully responsive and pleasant to view and use, and those that are displayed the same on a mobile device and a desktop computer, for example. The importance of creating responsive web pages for user experience is highlighted and explained in articles [1-3]. The number of visitors, how long they stay, and whether they come back certainly depends on how the website was designed. Among other multimedia content available on the Web, there are various interactive elements, which nowadays often include transitions and animations. With them, the content stands out or is presented in a more interesting way. Animations can be triggered by themselves when the page loads, or they can be triggered by a mouse movement. They can be small and barely visible, or they can contain various graphical elements and define a large part of the content on the page. Every year there are changes and advances in the way animations are programmed and executed. From CSS animations of various HTML elements to completely new technologies that animate elements according to completely different principles. In addition to the functional animations themselves, they can also be artistic, interactive, where the visitor of the website creates a new and completely individual appearance of the website and changes the parameters of the animation. Web animations can be two-dimensional or three-dimensional. Some technologies support the third dimension of the object (HTML, CSS) and some do not (SVG). However, by combining technologies it is also possible to create a 3D SVG animation.

2 AN OVERVIEW OF THE HISTORICAL DEVELOPMENT OF WEB ANIMATION

Animation is the perception of motion and the illusion of change through sequences of images that differ only slightly from each other [4]. Today, there are several technologies for creating animations on web pages. Each of them has advantages and disadvantages and the area in which it is used [5].

In the following, we describe the technologies that were used to perform the examples for the experimental part of this work.

HTML or HyperText Markup Language is a description language used to structure the web page and the content on the web page. The HTML elements are visible on the web page. An ordinary text editor (e.g., Notepad) and knowledge of the basic HTML elements are sufficient to write HTML [6]. HTML5 emphasizes simplifying the code needed to create a page that conforms to W3C standards and links CSS, JavaScript, and graphics files [7].

CSS (Cascading Style Sheet) consists of properties that define a value and is added to HTML elements. Thus, CSS is used to style the HTML elements that are displayed on the web page. CSS animations provide a lightweight solution for creating animations without the need for additional scripts. They can be controlled via CSS properties, making them relatively easy to implement. At the same time, complex and overused CSS animations that involve multiple elements or complex movements can increase page load times and rendering costs [8].

Scalable Vector Graphics (SVG) is a popular vector graphics format that provides good support for interactivity and animation [9]. With vector graphics, images are stored using mathematical formulas based on the coordinates of points and lines in a grid. This allows for significant resizing or scaling of SVG files without loss of resolution, making it ideal for creating icons and logos for websites. Scalable vector graphics are also used on websites for illustrations, 2-D and 3-D animations, interface parts, infographics and visual data representations [10]. The challenge with SVG animations is that SVG animations and interactions may not work equally well in all browsers, platforms, or versions. They may also require large resources and affect performance as they may consume a lot of CPU or GPU power. This can cause the website to lag, freeze, or crash if there are many elements [11].

WebGL or Web Graphics Library is a JavaScript API for displaying interactive 2D and 3D graphics in any compatible

web browser without using plugins [12]. The API allows users to experience interactive content on GPU-accelerated web pages without having to download or install plugins. WebGL was originally developed by Mozilla [13]. WebGL applications reside centrally on a web server and can be updated relatively easily for all users. The platform is independent of HTML and JavaScript. Some disadvantages are that the applications are slower than compiled programs whose code is translated directly into machine language and that there is no access to the full functionality of the specific platform [14].

In 2013, Lauren McCarthy developed p5.js, a native JavaScript alternative to Processing.js that is officially supported by the Processing Foundation. p5.js is a JavaScript library for creative programming, is free and open source [15]. The advantage of the JavaScript programming language is that it is available and supported everywhere. Every web browser has a built-in JavaScript interpreter, which means that p5.js programs will (usually) work in any web browser [16].

3 ANIMATION DESCRIPTION AND EXAMPLES

3.1 Windmill

Fig. 1 shows three frames from the windmill animation. Uniform animation is achieved with a) HTML/CSS; b) SVG/CSS; c) SVG/HTML; and d) SVG technologies.

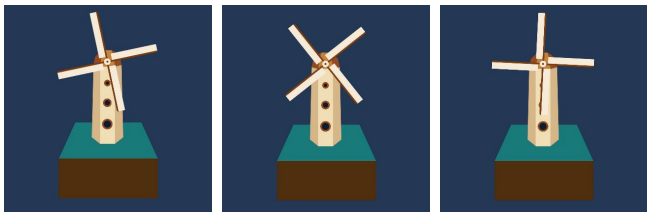

Figure 1 Three frames from 2D animations achieved by combining HTML, CSS, and SVG technologies

SVG/HTML, SVG

SVG Embed in HTML and SVG Standalone (.svg) are created with SVG animations. The command <animateTransform> was used and the transformation type is rotation. A group of 10 elements takes part in the rotation. For a group of elements that rotates, the rotation center, start angle and end angle are defined.

SVG/CSS

The SVG hierarchy is the same as in the previous example, but CSS was used for the animation. The transform-origin property defines the rotation centers for 10 animated elements. Help @keyframes are start and end angles defined at the beginning and end of the animation.

HTML/CSS

In this example, only HTML tags and CSS properties are used, and the entire model is rebuilt with div tags. As in the previous examples, only the group of 10 elements is animated, that is, the Rotor class, and the transformation parameters remain the same. An important difference in animating this example from the previous one is the need to change the origin of the animation for the sails. The transform-origin property defines a new origin so that the sails rotate together around the same point.

3.2 3D Animation Rotating Sphere HTML/CSS

The animation consists of 90 div elements placed in an inner and an outer container. The outer container contains perspective 1200px properties. The inner container contains animation with rotation along all three axes. At 0% animation, the rotation on all three axes is 0. At 20%, the sphere rotates 180 degrees along the X and Y axes and 10 degrees along the Z axis. At 50% animation, the sphere rotates 360 degrees along the X and Y axes and 45 degrees along the Z axis. At 100% animation, the sphere rotates 180 degrees along the X and Y axes and 0 degrees along the Z axis. All inner div elements (circles) are also transformed: 90 degrees along the X axis and for each subsequent circle 2 degrees more along the Y axis with a color change. The animation was created using HTML and CSS technologies only and contains 405 lines of code (internal CSS). The file weighs 9.48 KB. Fig. 2 shows 3 frames of the sphere animation.

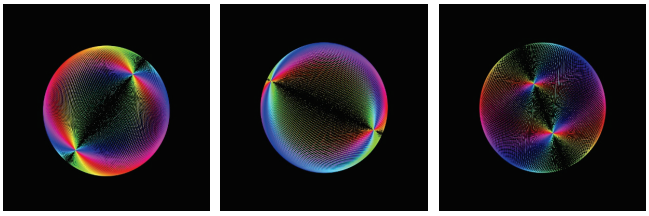

Figure 2 Three frames from a 3D sphere animation created using HTML and CSS technologies

3.3 3D Animation of the Sphere using p5.js and WebGL Technologies

Animation using p5.js and WebGL technology is obviously much simpler. It uses a for loop and draws equal circles with rotation and color change. All the HTML and p5.js code consists of 34 lines of code in total. The files weigh 799 KB. Fig. 3 shows frames of a sphere animation created with HTML, p5.js and WebGL technologies.

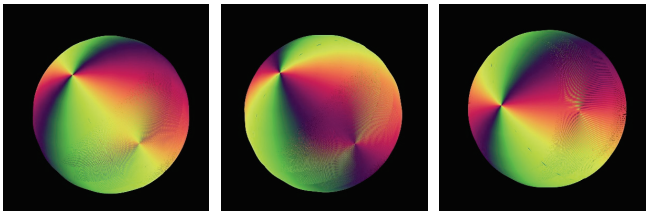

Figure 3 Three frames of a 3D sphere animation created with HTML, p5.js, and WebGL technologies

4 EXPERIMENTAL PART

In this research, Chrome DevTools were used to measure fps, CPU, GPU and animation load time until a stable maximum fps was reached. The devices used have the following specifications:

DEVICE 1: Laptop Asus
OS Name: Microsoft Windows 10 Pro
Version: 10.0.19045 BUILD 19045
System Manufacturer: Microsoft Corporation
System model: N76VZ
Processor: Intel® core (TN) i7-3630QM CPU @2.40GHz,2401 Mhz, 4 Core(s), 8 Logical Processor(s)

DEVICE 2: Laptop Asus Zenbook
OS Name: Microsoft Windows 11 Pro
Version: 10.0.22621 BUILD 22621
System Manufacturer: Microsoft Corporation
System model: Zenbook UX8402ZE_UX8402ZE
Processor: 12th Gen Intel(R) core (TM) i9-12900H, 2500 MHz, 14 Core(s), 20 Logical Processor(s)

DEVICE 3: Desktop HP
OS Name: Microsoft Windows 7 Professional
Version: 6.1.7601 Service Pack 1 Build 7601
System Manufacturer: Microsoft Corporation
System model: HP Compaq Elite 8300 CMT
Processor: Intel(R) core (TM) i5-3470 CPU, 3.2 GHz, 3201 MHz, 4 Core(s), 4 Logical Processor(s)

DEVICE 4: Laptop Lenovo
OS Name: Microsoft Windows 11 Home
Version: 10.0.22000 Build 22000
System Manufacturer: Microsoft Corporation
System model: LAPTOP-OGQAONQO
Processor: AMD Ryzen 7 5700U with Radeon Graphics, 1801 MHz, 8 Core(s), 16 Logical Processor (s)

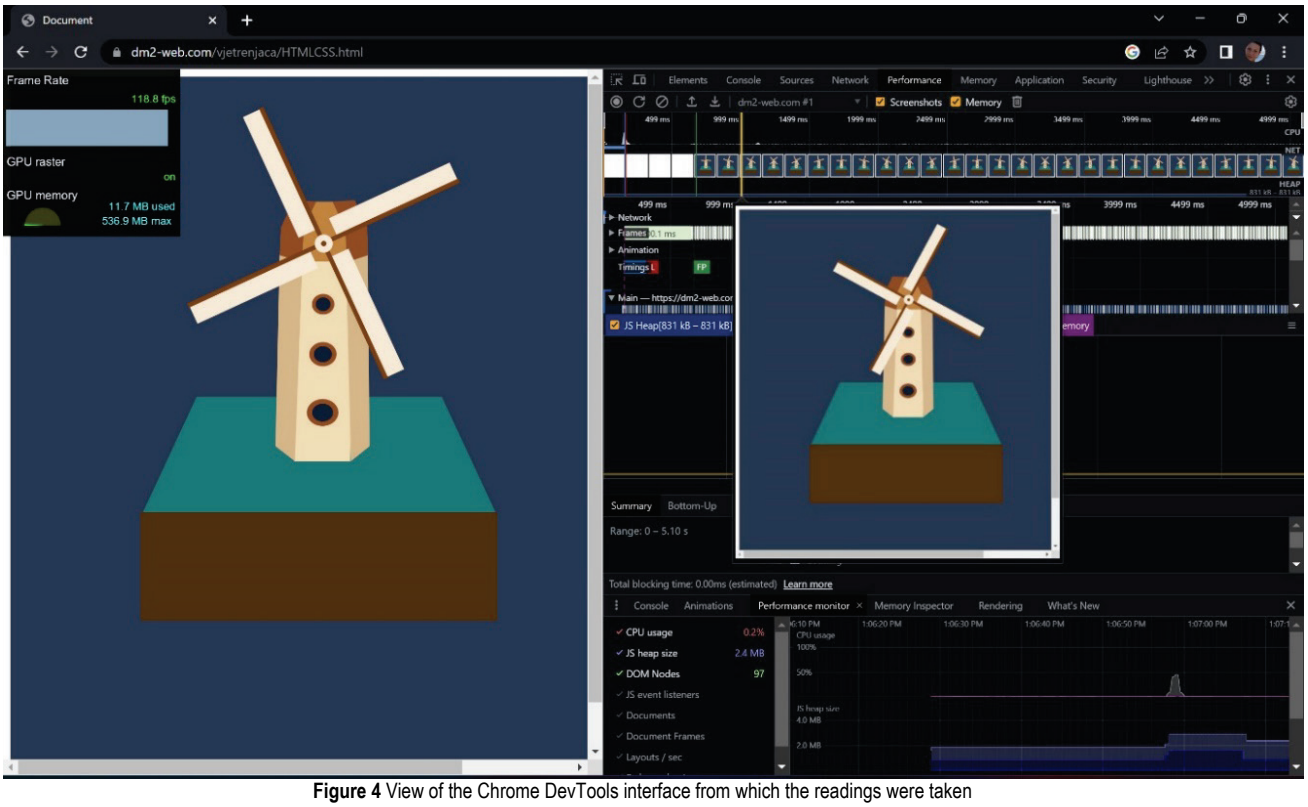

Figure 4 View of the Chrome DevTools interface from which the readings were taken

On each device, each animation was measured 10 times, and the average values were taken for the displayed results. All measurements were taken over the Eduroam network with a download speed of 155.16 Mbps and an upload speed of 154.96 Mbps. The speed was tested with the Ookla Speedtest.

4.1 Measurements 2D Animation of a Windmill

HTML/CSS, SVG/CSS, SVG/HTML, and standalone SVG animations were measured on the four computers mentioned previously. Fig. 4 shows the appearance of the Chrome DevTools interface for an example of an animation created with HTML and CSS technologies.

The measurement results are presented in graphs. Fig. 5 shows the measured fps values of the individual animations on the individual devices.

The three devices performed the test of these four animations almost equally well, with a result very close to 60 fps. On device 2, the average readings are 118.8 fps for the first three animations. Only the standalone SVG animation has a slightly lower frame rate compared to the others, an average of 102.3 fps. Device 2 is the newest and most powerful computer, using an i9 processor.

The following measurement is for GPU memory usage (MB) for the four animations listed and four different devices. There are significant differences in the results of each device in this measurement. Fig. 6 shows the result of this measurement.

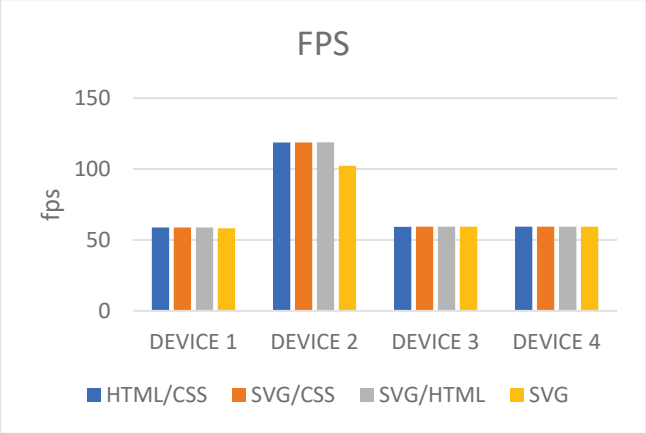

Figure 5 Measured values of frames per second of four animations on four different devices

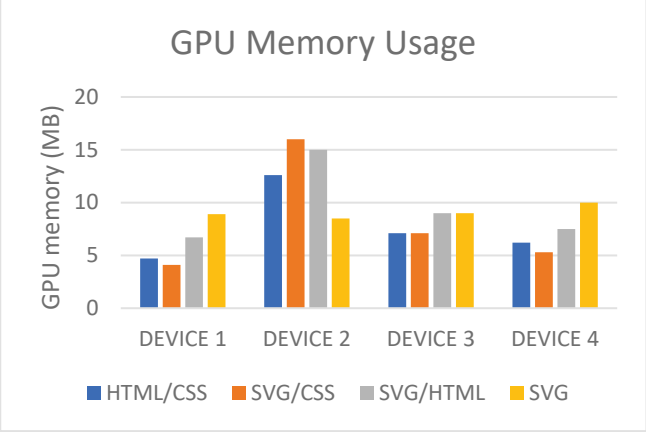

Figure 6 Shows the GPU memory usage in MB for four animations and four devices

On the three devices, GPU memory usage is lowest when performing SVG/CSS animations and highest when performing standalone SVG animations. On Device 2, the exact opposite values were measured. The highest GPU memory usage is recorded for standalone SVG animation, and the lowest for SVG/CSS animation.

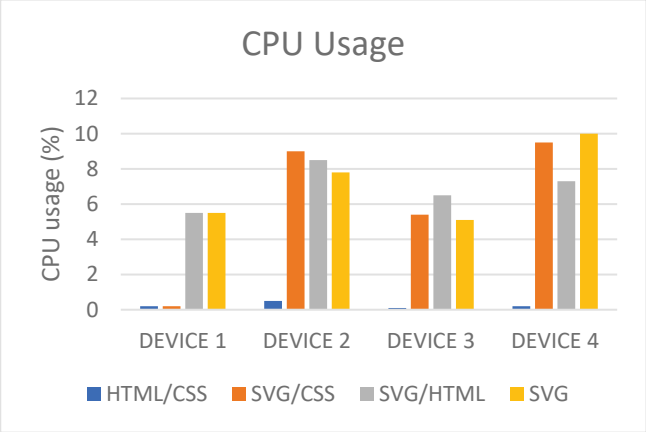

Figure 7 Measured values of CPU usage in % for four animations and four devices

The third measurement refers to the use of CPU (%). Fig. 7 shows the measured values. This graph shows that the use of CPU for HTML/CSS animations is by far the lowest on all four devices, at 0.1 and 0.5%. Other animations are measured quite colorfully on all four sites. SVG embedded in HTML and standalone SVG have the highest values.

The last measurement refers to the loading speed up to the moment when a stable maximum frame rate is reached. Fig. 8 shows the results of this measurement. The graph shows that the loading speed is highly dependent on the speed of the computer. For animations created with different technologies, different values were obtained on each computer. The HTML/CSS animation took the longest to load on two devices, the SVG/HTML animation on one, and the SVG/CSS animation on one.

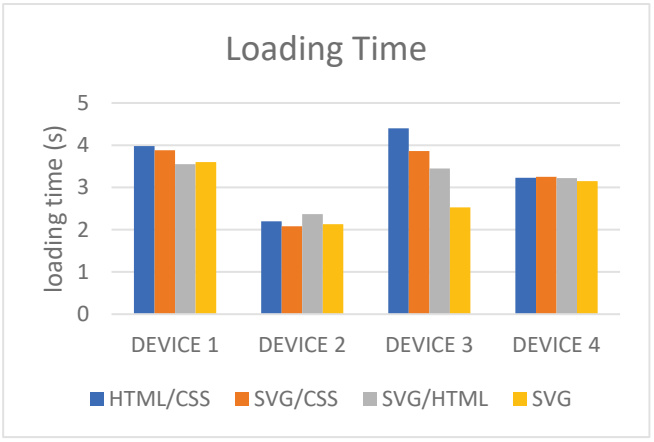

Figure 8 Animation loading speed in seconds to get a constant fps

4.2 Measurements 3D Animation of the Sphere

As with the windmill animation, the same values were measured for these examples. Due to the complexity of performing animations, the differences are larger than in the previous measurements. Fig. 9 shows the appearance of the Chrome DevTools interface when measuring an example 3D animation of a sphere.

Fig. 10 shows the fps comparison of two 3D animations. One was created with HTML and CSS technologies and the other with HTML, p5.js and WebGL.

On the first and fourth devices, the animations responded the same, which is understandable since the characteristics of these two devices are quite similar. On the 2nd device, a big difference is visible in the number of fps. The HTML and CSS animation again reached almost 120 fps, while the HTML/p5.js/WebGL animation was measured just below 80 fps. Device 3 is an old desktop computer and on it these animations reached 12 and 8 fps respectively.

In the following measurement, the GPU usage values, expressed in MB, were measured for the same examples shown in Fig. 11.

When rendering HTML/CSS animations on all four devices, GPU usage is higher than when rendering HTML/p5.js/WebGL 3D sphere animation.

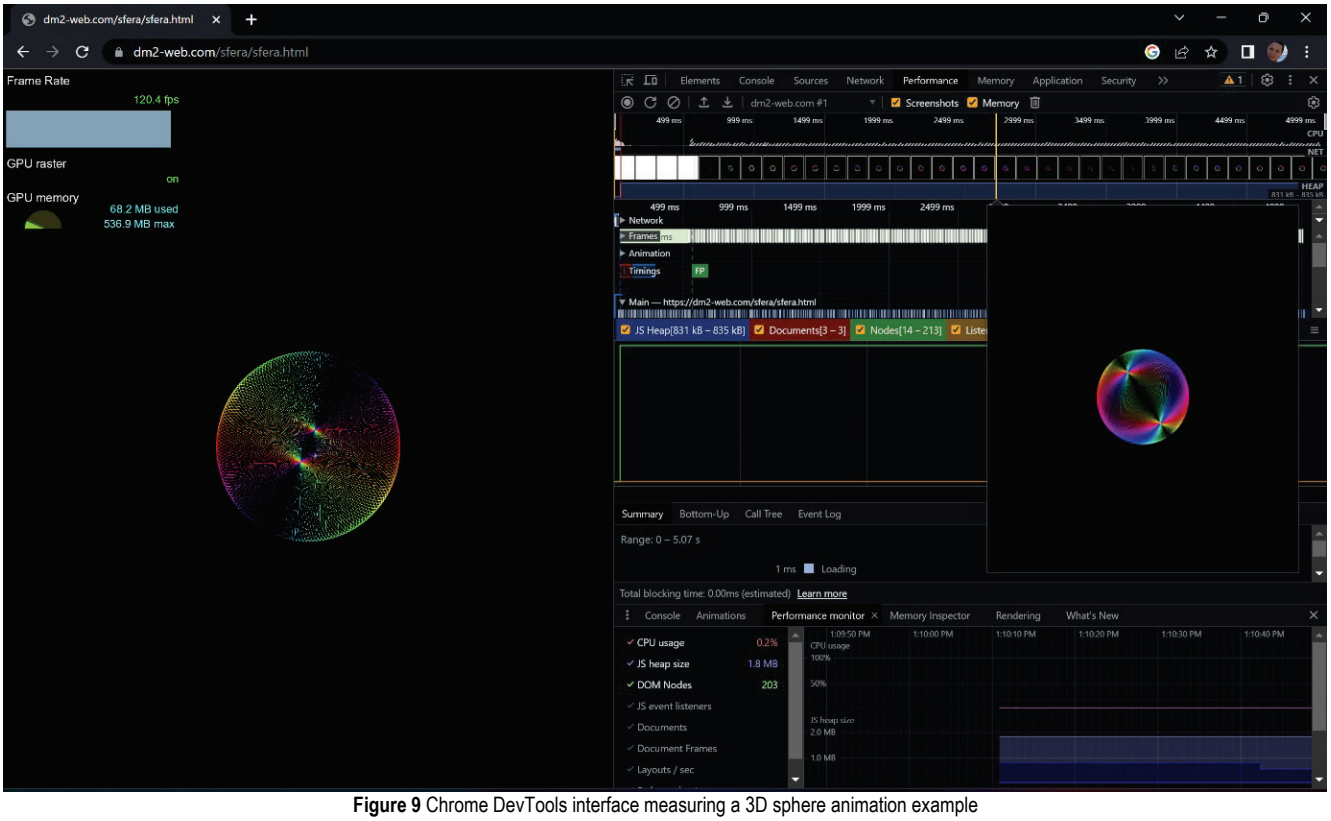

Figure 9 Chrome DevTools interface measuring a 3D sphere animation example

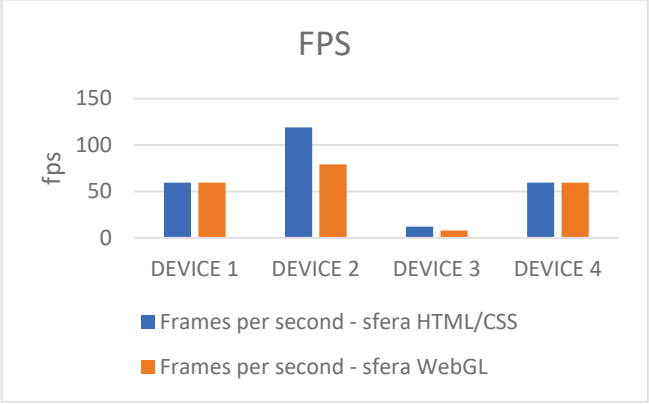

Figure 10 Comparison of measured fps of 3D animations created with HTML and CSS, and HTML, p5.js and WebGL technologies on 4 different devices

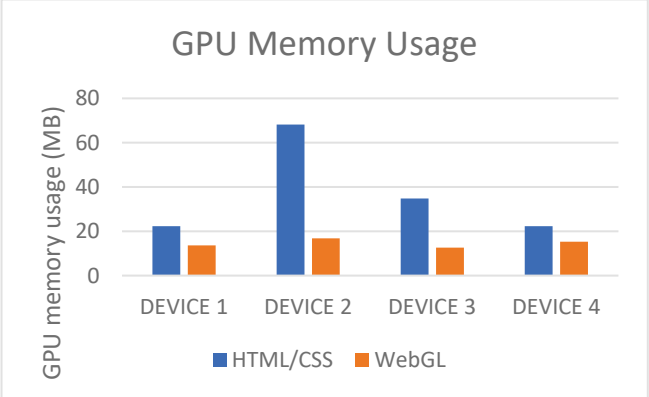

Figure 11 Comparison of measured values for GPU usage (MB) for displaying 3D animations created with HTML and CSS and HTML, p5.js and WebGL technologies on 4 different devices.

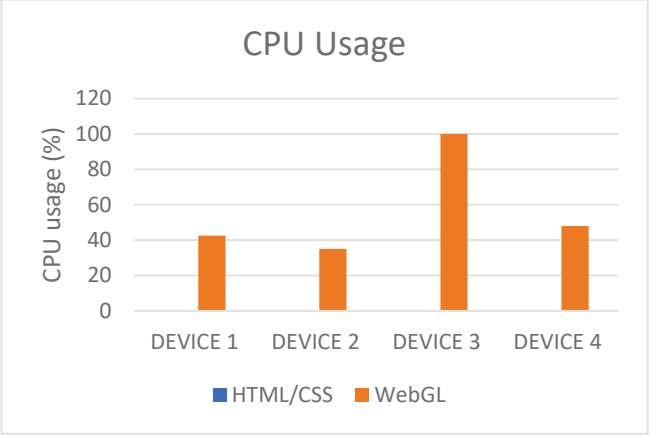

Figure 12 Comparison of measured values for CPU usage (%) for viewing 3D animations created with HTML and CSS and HTML, p5.js, and WebGL technologies on 4 different devices.

Fig. 12 shows the results of measuring the use of CPU for 3D animations. The values for the usage of HTML/CSS animations CPU range from 0 to 0.2 for all four devices. For animations created with HTML, p5.js and WebGL technologies, these values are quite high, especially for device 3 where they are 100%.

The last one, Fig. 13, shows the loading time of 3D animations until the frame rate has settled at its maximum.

Fig. 13 shows that the 3D HTML/CSS animation loaded faster on all four devices than the one created with HTML/p5.js/WebGL technologies.

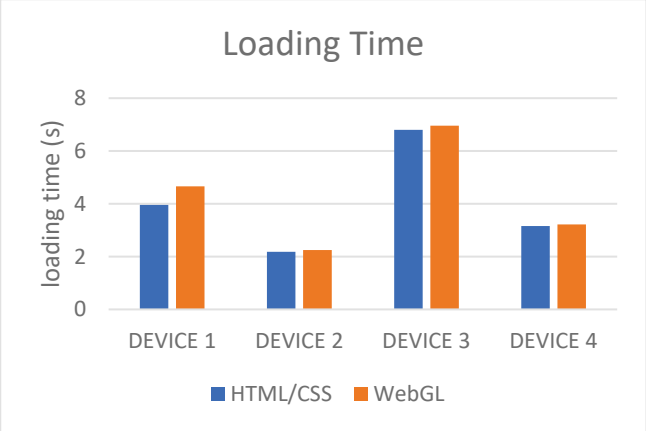

Figure 13 Comparison of measured values for the loading time of 3D animations created with HTML and CSS and HTML, p5.js and WebGL technologies on 4 different devices.

5 CONCLUSION

From the measurements and results, it can be concluded that the 'cleanest' animation is achieved with HTML/CSS technologies. The frame rate is very high, depending on the device. Also, there is a very big difference in the data obtained from different devices. Older devices will have a hard time playing the demanding animations and displaying them well. The animations (scenes) explored in this work were quite complicated. The 2D animation of the windmill is a scene built to simulate a 3D scene. The 3D animations of the sphere are real 3D animations. So the devices used in this work were really challenged to perform well. All the animations performed very well. The technologies used for their production really have their positive and negative sides. It is important to keep in mind that the animations should be moderate so that they do not affect the loading time of the landing page. Besides, not all users have new and very fast devices, and the network speed is also not equally fast and stable everywhere and for everyone. The performance of 3D spheres created with HTML/p5.js/WebGL technologies is somewhat disappointing. This is the area that should be explored further, as well as animation behaviour on mobile devices. The use of one technology should not exclude the other but can complement each other to achieve better results and new possibilities.

6 REFERENCES

[1] Horbinski, T. & Cybulski, P. (2019). Functionality similarities of global web mapping services in the context of responsive map design. *Abstr. Int. Cartogr. Assoc.*, 1, 117. https://doi.org/10.5194/ica-abs-1-117-2019

[2] Walsh, T. A., Kapfhammer, G. M., & McMinn, P. (2020). Automatically identifying potential regressions in the layout of responsive web pages. *Journal of Software: Testing, Verification and Reliability, 30*(6), e1748. https://doi.org/10.1002/stvr.1748

[3] Li, W., Zhou, Y., Luo, S., & Dong, Y. (2022). Design factors to improve the consistency and sustainable user experience of responsive interface design. *Sustainability, 14*(15), 9131. https://doi.org/10.3390/su14159131

[4] Ferreira, M. (2017). The history of web animation. Medium. Available at: <https://medium.com/@milberferreira/the-history-of-web-animation-63b106c97fdf> (Accessed May 2023).

[5] Crews T. B. May K. Skintik C. & South-Western Cengage Learning. (2017). Digital media: concepts and applications (Fourth). Cengage Learning. ISBN: 978-1-305-66172-1 https://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=2638755. (Accessed June 2023)

[6] W3Schools, 2021. HTML attribute reference. Available at: https://www.w3schools.com/tags/ref_attributes.asp (Accessed May 2023).

[7] Frain, B. (2012). Responsive Web Design with HTML5 and CSS3, Packt Publishing; Illustrated edition.

[8] Weyl, E. (2016). Transitions and animations in CSS: adding motion with CSS. Sebastopol, CA: O'Reilly Media, 6-55.

[9] Wu, R., Su, W., Ma, K., & Liao, J. (2023). IconShop: Text-Based Vector Icon Synthesis with Autoregressive Transformers. arXiv preprint arXiv:2304.14400.

[10] Drasner, S. (2017). SVG animations: from common UX implementations to complex responsive animation. Sebastopol, CA: O'Reilly Media, 17-19, 73-75.

[11] Nolan D. A. & Lang D. T. (2014). Xml and web technologies for data sciences with r. Springer. https://doi.org/10.1007/978-1-4614-7900-0

[12] Ivanušec, S. (2021) Ostvarivanje 3D grafike u Internet preglednicima pomoću biblioteke Three.js, University of Pula, *Master's thesis*, 2-7. (in Croatian)

[13] WebGL, Available at: https://www.techtarget.com/whatis/definition/WebGL (Accessed May 2023).

[14] https://viscircle.de/webgl-basics-evalution-and-examples/?lang=en (Accessed June 2023)

[15] 5P*js. https://p5js.org/ (Accessed May 2023)

[16] https://blog.logrocket.com/creating-animations-p5-js/ (Accessed June 2023)

Authors' contacts:

Nikolina Stanić Loknar, PhD, Assist. Prof.
(Corresponding author)
University of Zagreb, Faculty of Graphic Arts
Getaldićeva 2, 10 000 Zagreb, Croatia
nikolina.stanic.loknar@grf.unizg.hr

Tajana Koren Ivančević, PhD, Assist. Prof.
University of Zagreb, Faculty of Graphic Arts
Getaldićeva 2, 10 000 Zagreb, Croatia
tajana.koren.ivancevic@grf.unizg.hr

Mateo Bekavac, univ. bacc. ing. techn. graph.
Otona Ivekovića 2a, 21000 Split, Croatia
mateobekav@gmail.com

Snježana Ivančić Valenko, PhD
University North,
Jurja Križanića 31b, 42000 Varaždin, Croatia
snjezana.ivancic@unin.hr