

# Research on CSS3 Features and Animation Performance

Ke Wang, 605285 Bingjing Xu, 605311

**Abstract**—CSS3 is the latest evolution of CSS. It brings a set of new features into Cascading Style Sheet language.[1] This article is aimed to study some of the new features in CSS3, which includes: Web Responsive Design and 3D transforms and transition. Sass is another alternative language with simpler syntax compared with CSS, but offer same effects as CSS3 does. Also, two performance tests will be conducted between CSS3 animation and JavaScript animation in order to evaluate their performance regarding to animation effect and speed. From the result, it is found that CSS3 performs better than native JS as the aspect of small and light animation, while some advanced JavaScript libraries such as GSAP and Velocity.js outperform CSS3 in the animation speed test we conduct.

## I. INTRODUCTION

CSS3 is the latest revolution of CSS. It brings a lot of long-awaited novelties, like animations and grid-layout to website design.[1] In the current state, most of websites implement the technology of CSS3 and use it to control the layout and style of web pages. For this study, a small web application is developed in order to provide an environment for testing CSS3 and the theme is set to be a small and light note-taking application, which might help students take notes and reminders in a single page. The consideration for the theme selection is that the note-taking web application is somehow simple and concise, which allows the researchers to focus on the study of CSS3. Also, since there could be a number of 'note papers' on the same page, it can be a test field to implement different CSS3 features especially animations into those note papers. This will help to revolutionize the design of notes-taking applications from 2D to 3D.

The rest of essay will be structured as follows; Firstly, the related concepts and works will be introduced in the next section, then it is followed by the method how CSS3 will be tested and researched in such a context. Also, the result of the research is given in the fourth section and the conclusion and more discussion can be found in the last part of the essay.

## II. RELATED WORKS

CSS, stands for Cascading Style Sheet, is a language used for describing the presentation of a document written in a markup language.[2] It is widely implemented in most of websites. And CSS3, as the 3rd version of CSS language, provides a set of brand new features with web developers. This makes the style of website with more probabilities. Sass, on the other hand, is a similar script language which can be interpreted into CSS. It extends CSS by providing

several mechanisms available in more traditional programming languages, but it is not available to CSS3 itself.[3] Therefore, Sass is more readable and easy-to-use in some degree.

## III. METHODOLOGY

In order to deeply understand the topic: CSS3, it is planned to study it from various aspects. Web Responsive Design is a hot topic currently. And CSS3 leads the trend and provides the very convenient solution to developers. Sass redefines the syntax about Cascading Style Sheets language, which offers a more flexible and clear framework for the website style. 3D transforms and transition is one of the new features in CSS3, this allows developers to add various three dimensional animation effect in the web pages. Since CSS3 has the ability to present 3D effects and animation, it is now accessible to compare the animation effects between from native Javascript language and some other JS libraries. However, it is necessary to get a basic understanding on the research environment before formal research.

### A. The research environment: Smart-Notes Application

Smart-Notes is a small and light-weight note-taking application which is built on the web platform. The functions are fairly simple: click '+' button to add notes and a piece of 'note card' will show up in order, and some special effect notes will also appear on the web with the operation of click, including: rotation, transition and the combined one.

Also, there is a list of buttons in the bottom of the web page, which control the function of theme color changing. The color of the buttons stands for the theme color which is going to be changed on the web. The performance test also owns a button on the web, linking to its on test page.

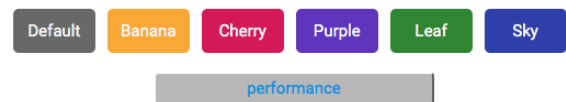


Fig. 1. color button and performance button.

The web application was built by pure HTML5, the combination of CSS3 and Sass and pure JavaScript.

### B. CSS3: Responsive Web Design

One of the important features on notes-taking application is that it should be very convenient to use in any smart device,

no matter in personal computers, mobile phones or tablets. And this is for the purpose of adjusting to various studying situations.

Therefore, it is very crucial to implement responsive web design. This design only relies on HTML5 and CSS3, without JavaScript or other complicated frameworks or programs. And it enables all the web pages which implement responsive web design to adjust to various sizes of browser window. Even if the user shrinks, resizes or enlarges the window, the content and layout still looks good. [5]

In this research, the notes-taking web also implements the design. The whole page is divided into 12 columns, and this is for better layout all the elements on the page. The normal width of browser window in the computer is 1280 pixels. With this background, some breakthrough points are set, and if the width of the window exceeds the breakthrough point, the layout is about to change to adjust to the new size of window. The figure below is the web in the normal size of window, and the next one is when the window is shrinking. Figure 3 shows

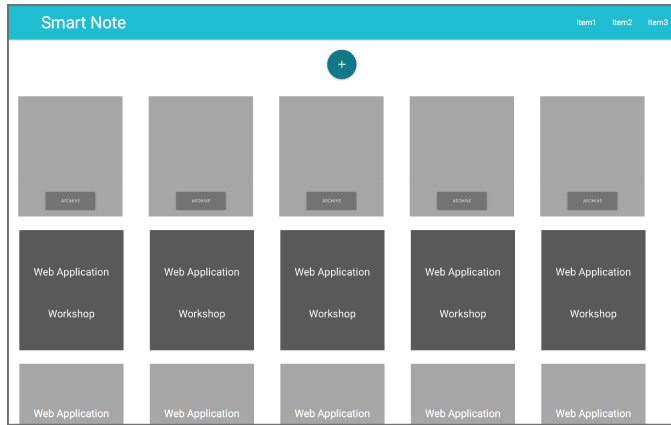


Fig. 2. The normal view of smart-notes

the effect of responsive web design. The notes which cannot be put on the same row is transited to next row compared with the above graph. Also, the navigation bar on the very top changes accordingly.

If width of the window continues reducing, the list of buttons in the bottom is also affected, the result is like figure 4.

### C. CSS3 and Sass

One of the features in this note-taking application is changing the color of note cards, and it refers to a certain percentage of CSS codes which is in charge of this feature. Since CSS3 does not allow users to define variables, it usually requires many rigid repeats on the same point in the stylesheet code. In this research, it is especially true in the function of color changing. If there is no corresponding color variables, the CSS code can be very redundant and nasty. Hence, Sass is introduced and implemented in the development of application.

Sass is completely compatible with CSS3, consequently, there is no conflict problem when Sass and CSS3 are used together

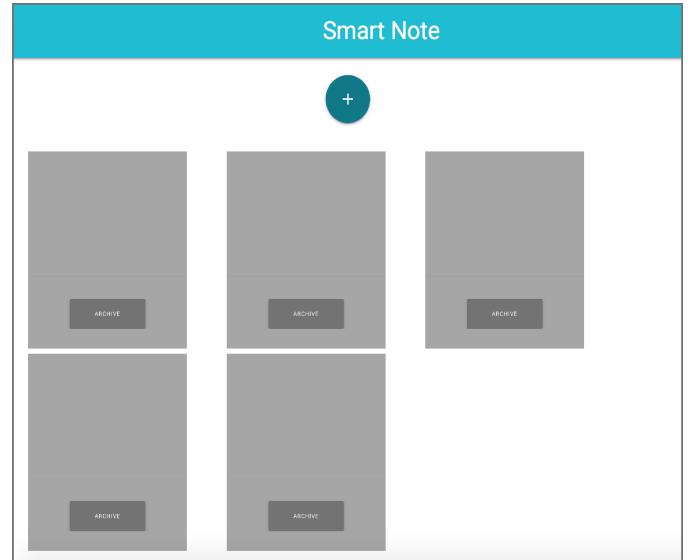


Fig. 3. The normal view of smart-notes

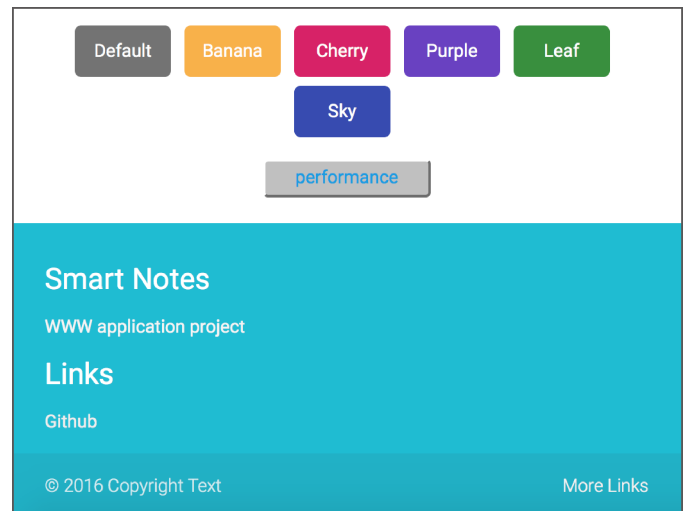


Fig. 4. The shrinking window view of smart-notes

on the same HTML file. In the notes-taking application, Sass is mainly used to define variables on colors. When users click corresponding buttons to change theme color of note cards, the variable will be transferred by click handler, which is a small snippet of JavaScript. This piece of JavaScript can change the color by transforming the new variable into target CSS snippet. Therefore, the variables in Sass helps avoid certain numbers of repeats compared with pure CSS3 case. The result of function color changing is as following picture.

Also, Sass provides a series of features which simplifies the stylesheet coding. For example, it allows nesting on CSS selectors. In this way, the cascading style sheet will have same visual hierarchy as the corresponding HTML file. Furthermore, Sass allows imports and operators, and these features all helps to the simplification of CSS code.[6]

The features illustrated below would be showed in the follow-

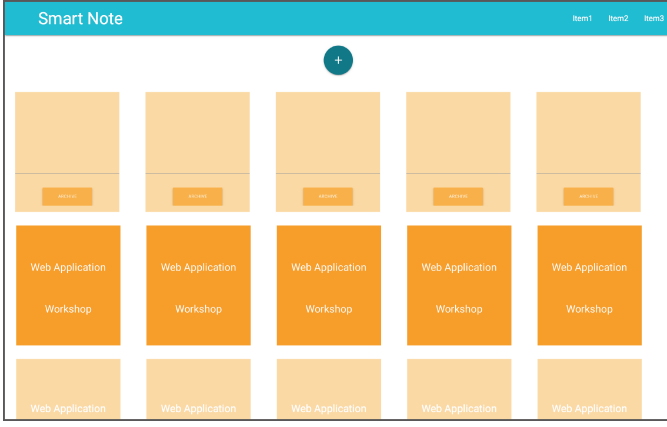


Fig. 5. color button and performance button.

ing code comparison. Both CSS3 and Sass snippet serves to the same function, however, the syntax is somehow different. Table I below presents a comparison of CSS3 and Sass syntax when defining the same page style.

TABLE I  
CSS3 VS SASS

CSS3	SASS
.box { background:grey; border:1px solid #d7d7d7; }	\$grey=grey; .box { background:\$grey; border:1px solid #d7d7d7; }
.box.button { font-weight:bole; text-align:center; padding:4px 8px; background: red; color: white }	button{ font-weight:bole; text-align:center; padding:4px 8px; background: red; color: white }

#### D. CSS3: 3D transforms and transition

This section will focus on discussion on the features of 3D transform and transitions in CSS3. Compared with 2D transform, 3D transform breaks out the usual visual effect on HTML elements, creating more 3D dimensional effect by allowing the operations on z-axis. Meanwhile, CSS transitions enables separate still 3D effects to change smoothly in given time slots. With the combination of those two features, it is easy to create some small 3D animation effects on the platform of websites.

In order to let the users of smart-notes application have better visual experience , the research fully implemented those two features into the development. Except that the first row of note cards in the page is normal cards without any animation effect, the other rows all have more or less level of animation effects. The cards in the second row implements flip effect: click some card, then the card will rotate along with z-axis with 180 degree, and some texts will appear on the back of card. And if the click is operated again, then the card rotates to its original position.

Then the third row has a totally different animation effect. Through the click on the note, a new card will smoothly move from left to right till it fully covers the old card.

The effect on the cards in last row is more about 2D transforms, which allows a new card to rotate centering with the left-up corner till it fully covers the fixed old one, if the user moves the mouse onto the card.

Part of the effects are showed in the following figures.

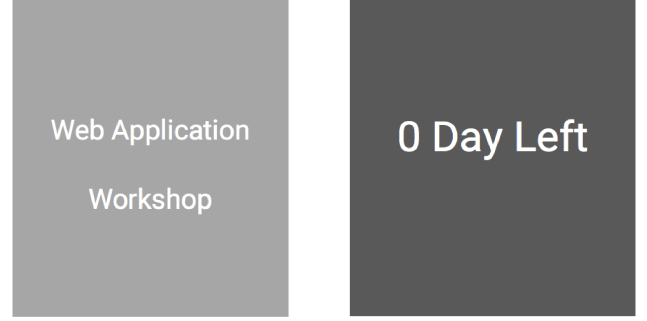


Fig. 6. The original view and changed view of one card before and after animation effect

#### E. CSS3 vs Native JS Animation

Since the new features on animation has been added into CSS3, some JS coding effects can be replaced by CSS3. This is a breakthrough in the history of CSS, because it will reduce the difficulty of programming for developers on the aspects of creating 3D effects on websites. However, it still remains a myth that how CSS3 animation performs in the browser or in the development environment.

To uncover the myth, a small performance test is conducted both in CSS3 environment and native JS environment. Some criteria are given as parameters to evaluate the result. Those criteria includes frame rates, rendering time, memory management and events in time frame. As the essay mentioned in the section of introduction, the performance test is via a button linking to another address. In this web page, a small animation effect is used as test sample: 1000 rotated yellow rectangles are set to move from left to right and from right to left repeatedly. The test implements the transition features in CCS3 and requestAnimationFrame() API in Javascript separately. The switch between these two is via a flat button:toggle. And the whole testing process is conducted in Google chrome browser and all the data comes from Google inspect tool. After the test, the result is collected and written in the table as follows:

TABLE II  
ANIMATION PERFORMANCE BETWEEN CSS AND NATIVE JS

	CSS3	Native JS
Frame rate	43 – 60 fps	30 – 60 fps
Rendering Time	0.42s	0.73s
Memory Management	8.1 MB	13.9 MB
Events intensity in every timeframe	loose	intensive

Frame rate is an important parameter which indicates how many frames is rendered in the browser. For CSS3, the frame rate range is from 43 to 60 fps and the accurate frame rate from real test is 59.1 fps; Whereas the range of FPS in native JS is from 30 to 60 fps and the accurate number is 48 fps. Then, rendering time is also compared between CSS3 and JS. To test this, 1 second is set as the test unit, and the duration of event rendering time accounting for the whole 1 second is collected. The rendering time for CSS3 is about 0.42s, however, JS takes more 0.31s for rendering. GPU Memory is also another parameter needing to be monitored. From the collected data, it is found that CSS3 animation also takes less memory than JS animation. The last evaluated field is events intensity in every timeframe and the result shows CSS3 is very loose compared with the intensive graph of JS.

Therefore, the table can conclude that regarding on the small animation effect in this case, CSS3 has a much better performance than native javascript. It has more frames per second and faster rendering speed, also takes less GPU memory when running special moving effect, and very loose events in every timeframe.

#### F. CSS3 vs GSAP and Velocity.js

JavaScript itself is obviously not a wise choice for webpage animation, this is normal since its not created for animation. CSS3 is optimal for relatively simple animations, for it forces hardware acceleration and is efficient at optimizing DOM interaction. The browser helps it to execute the animation logic. However, some tools and libraries based on JavaScript has been optimizing animations in browsers. GSAP is one of these tools that robustly animates anything JavaScript can deal with including automatic GPU-acceleration of transforms. [7]

The other library to be tested in this paper is Velocity.js. It contains all the functionality that jQuery animate provides, while it also offers transforms, loops, color animation and so on. It is a combination of jQuery and CSS transitions. [8]

Therefore, another speed test is conducted for CSS transitions, GSAP and Velocity.js. The test animates simple transitioning (scale and transform) dots in the browser, and in the test the total amount of the dots will be gradually increased and the data of fps will be conducted afterwards. The results are shown in the table below.

TABLE III  
ANIMATION SPEED TEST AMONG CSS3, GSAP AND VELOCITY.JS

Dots	CSS3	GSAP	Velocity.js
200	75 – 96 fps	98 – 100 fps	89 – 100 fps
300	55 – 67 fps	98 – 100 fps	90 – 96 fps
500	5 – 14 fps	75 – 82 fps	51 – 60 fps
1000	0.5 fps	38 – 47 fps	25 – 35 fps

Despite the small performance gap between two JavaScript libraries, they both outperform CSS transitions in this speed test. The reasons why JavaScript can reach high levels of performance are: Minimizing layout thrashing by twinning stack during animation. Caching property values across chained calls in order to minimize the occurrence of DOM querying [9] JavaScript-based animation can decide when to start hardware

acceleration and is perfect for batched animation optimization. On contrary, CSS transition forces hardware acceleration, which will become CSSs drawback when GPU is in a high-stress situations.[10]

#### IV. RESULTS AND ANALYSIS

The research is conducted in six aspects, which includes responsive web design, the syntax, new features and two animation performance tests. It generally studies nearly all aspects of CSS3 and enables the readers to gain a basic understanding about CSS3 after reading. But there are still some places which need to be improved. The performance test written in the last part of methodology part is an good example. Since the whole test is only conducted in Google chrome browser, which is not strict to meet the requirement of scientific research. A possible remedy for this is to take the same task in different browsers such as Safari, Firefox and Opera etc. This may enable the statistics to be more accurate. Also, the result of this performance test is closely related to the animation effect that is tested in this research. Because the animation sample in the test is fairly simple, the statistic might be inclined to CSS3. However, if the sample is instead of other more complex effect, results may vary widely. One additional point is that JS actually has various libraries on the market, which has improved performance and better framework, and they usually perform better in terms of animation effect. Therefore, another test was conducted to test the performance of these JS libraries. And from the result, it is clear now that JavaScript libraries like GSAP and Velocity.js can reach a better performance that CSS transitions in this speed test. The range of statistics might not be strictly accurate. When the volume of test objects reaches a significant amount, we observe a small performance drop in Velocity.js relatively compared with GSAP, while these two might have similar performance in many tests. This difference performance is caused by the system bottleneck, shifting from the limitation of the library into a lower level software limitation, such as browser engine, operating system etc..

#### V. CONCLUSIONS AND DISCUSSION

CSS3 is an excellent script language which toggles the style of web pages. It provides the very convenient and easy method to implement responsive web design, and currently this is a very necessary part of web design and development in order to meet the modern needs to view the web pages on different devices. Since CSS3 is the latest version of CSS language, it also brings a series of new features to CSS, for example, 3D transforms and transition makes animation written in CSS possible, this may stimulate more animation implementations in websites. Furthermore, CSS3 has an excellent performance on small and light animation effects, the statistics that is collected from CSS3 is better than those in pure Javascript. In the aspect of syntax, compared with Sass—the extension class of CSS language, CSS3 may have more complicated syntax and lower readability for developers, which could be one of the disadvantages on CSS3.

## REFERENCES

- [1] <https://developer.mozilla.org/en/docs/Web/CSS/CSS3>
- [2] "CSS developer guide". Mozilla Developer Network. Retrieved 2015-09-24.
- [3] H. Poor, An Introduction to Signal Detection and Estimation. New York: Springer-Verlag, 1985, ch. 4.
- [4] Kataria, Saransh. "Getting started with sass development". wisdomgeek. saranshkataria.
- [5] [http://www.w3schools.com/css/css\\_intro.asp](http://www.w3schools.com/css/css_intro.asp)
- [6] <http://sass-lang.com/guide>
- [7] <https://greensock.com/gsap>
- [8] <http://velocityjs.org/>
- [9] <https://davidwalsh.name/css-js-animation>
- [10] <http://blogs.adobe.com/webplatform/2014/03/18/css-animations-and-transitions-performance/>