



College of Engineering

CS CAPSTONE TECHNOLOGY REVIEW

JUNE 7, 2019

INTERACTIVE 2D SIMULATIONS TO SUPPORT INQUIRY-BASED LEARNING IN MECHANICAL ENGINEERING

PREPARED FOR

OREGON STATE UNIVERSITY

TOM ESTKEDT, MILO KORETSKY

PREPARED BY

GROUP 53

LEARNING SIMULATIONS

KELLI ANN ULEP

Abstract

The solution to create interactive 2D mechanical simulations involves many different components. The web application has been broken down in different components. This document compares and contrasts 3 different technologies for 3 different components used to complete the solution. Each technology has its own benefits and drawbacks, and these will be used to decide which technology to use.

CONTENTS

1	Introduction	2
2	Animation Libraries	2
2.1	Popmotion.io	2
2.2	Animate.css	2
2.3	GreenSock Animation Platform (GSAP)	2
3	Styling Libraries and Frameworks	3
3.1	Bootstrap	3
3.2	Bulma.io	3
3.3	Semantic UI	3
4	Video Players	4
4.1	HTML5 Video Player	4
4.2	YouTube	4
4.3	Vimeo	4
5	Conclusion	4
	References	5

1 INTRODUCTION

This project involves solving the problem of some universities lacking resources to visually show physical and mechanical interactions for Mechanical Engineering concepts in the classroom. To solve this problem, the project aims to build at least one 2D simulation that accurately emulates a physical or mechanical idea and allows for student interactions with the system. The simulation will run on a web browser, and students will receive links to different situations within assignments. The web application has been broken up into three components: animation libraries, styling libraries, and video players.

2 ANIMATION LIBRARIES

The simulations should use an animation library so that they can be created with greater ease. Web pages can use animation libraries with HTML, CSS, or JavaScript. Animation libraries should be simple to use, well-documented, and versatile. The library should be easy to incorporate into the code. The web page should use these libraries for creating the mechanical animations itself or other parts of the user interface - the buttons, sliders, or status updates. The options the section will cover are Popmotion.io, Animate.css, and GreenSock Animation Platform (GSAP).

2.1 Popmotion.io

Popmotion.io - specifically Popmotion Pure - is an open source, low-level animation library for any JavaScript environment. It works with most popular browsers to render any target like the DOM. Popmotion Pure seems the most appropriate of the Popmotion libraries as it offers more animation freedom. The Popmotion pure site contains a detailed "Getting Started" tutorial and simple library import instructions. The API (Application Programming Interface) is also documented well with many usage examples. It is available on npm (a node package manager) or can be simply imported or included in the code with one line. This would allow for more ease of entry. The library size is only 11.7kb which is smaller than GSAP.

2.2 Animate.css

Animate.css is "just-add-water CSS animation" library and is known for being simple to use. It has 55,181 stars on GitHub, so it is fairly popular. The library is compatible with different browsers (cross-browser), and only really involves CSS3. Like Popmotion.io, it is incorporated easily into a project by installing with npm and referencing the file. It can be combined with jQuery, another common animation library, making it a little flexible to use [1]. Video tutorials also exist online. The library is open source and licensed under the MIT license, so it would not take much trouble to use. The library does not provide much documentation except a readme file on GitHub, but at a first glance, the API seems simple enough that it does not need much explanation. Because it is such a simple animation library, even if it can be combined with jQuery, this library seems to work best only for simple web page animations like input sliders, boxes, or labels. It would not be as effective to animate the actual mechanical simulation because it is so simple, but it would be possible.

2.3 GreenSock Animation Platform (GSAP)

GSAP is an animation library based off JavaScript. GSAP is a collection of JavaScript files, compatible with many popular browsers. It claims to be "The most robust animation library on the planet" [2]. GSAP is also known for its performance

and efficiency. The site provides a speed simulation in comparison to other popular animation libraries like jQuery, and tops the list in performance. The site shows much documentation and video tutorials, so it would not take much to learn to use. A draw back is that some parts of the library is only accessible for paid users. A paid business license is necessary for a product that is sold to multiple users - which would have to be the case for this project. The simulation would be part of a site you have to subscribe to. Other libraries can do what GSAP can do without having to pay, but this well known library is recommended if robustness and speed is one of the main concerns of the web page. [2]

3 STYLING LIBRARIES AND FRAMEWORKS

The front-end of the web page should be organized that the user can easily understand the format of the web page. Styling libraries make organizing a web page easier. The library should be simple to use that it is appropriate for an educational site. Preferably, it should have responsive design functionality, but it is not a high priority for this project. Most importantly, it should be simple to use, robust, and compatible with most browsers.

3.1 Bootstrap

The framework is considered the best responsive front-end framework and one of the most popular open source projects. With its grid systems and different buttons, it is one of the most commonly known responsive design user interface framework. It is one of the most popular libraries, so many documentation and tutorials exist, making it easier to learn if one has no previous experience. Also it is compatible with most browsers. Other frameworks may have the same functionality as Bootstrap, but because it is such a popular library, much more resources are available like extra add-ons, plugins, or provided themes. For the educational web page this project aims to build, Bootstrap's main advantages are that its well-documented, and the simulation can easily be made for tablets or phones [3]. Responsive design is not one of the high priorities, and it can be one of the stretch goals. With Bootstrap, however, it could make it easier to change if the project ends up going in that direction. The code is released under the MIT License and released under Creative Commons and can be easily incorporated into the project.

3.2 Bulma.io

Bulma is another open source CSS framework that is used by over 100,000 developers. It is based on CSS's Flexbox, having one of its main benefits responsive design. The framework is designed so the site works on mobile platforms first. The grid system that Bulma uses is simple since the web page columns or sections resize themselves. The library is modular such that it consists of 39 .sass files that can be imported individually. Its modular nature makes this library more lightweight than Bootstrap, as developers can import only what they need. Bulma is structured as a collection of CSS classes, so that the HTML code has no impact on the page styling. The site shows well presented documentation on all the classes, and the classes are readable. It is not as popular as Bootstrap, so it might not have as many tutorials and online resources. The code, however, is still well documented and is simply stated. Only one file is required to use Bulma, so it can be easily included in the code base [4].

3.3 Semantic UI

Semantic is another open source framework helping to build responsive layouts. The library's main benefit is to make the code human-readable since it regards words and classes as "exchangeable concepts." Library syntax is readable and is

intuitive, so the code can be understandable. This is important for the project since the code must be understandable by all current developers and future developers. Then developers can easily create and debug code. Though not as popular as Bootstrap, the code is still well documented and there are tutorials. There are also many built in UI components that the others may not offer, but there are no extra plugins as it is so simple. This library is still in development in integrating with other common frameworks like Angular or React [5]. It may become issue if the project uses those potential frameworks or would eventually switch to them. However, it is compatible with most major web browsers. The library can be used simply, but there is not so much room for flexibility. The set-up is not as straightforward as Bulma or Bootstrap.

4 VIDEO PLAYERS

The website can have an option of displaying a video of the real system that the animation is based off. The way the video players are embedded into the website should be robust and simple, and the video must work.

4.1 HTML5 Video Player

This option allows more compatibility if the users come from different devices. All modern browsers support this option - so videos can also be streamed on mobile devices. This a simple option in comparison to other video services. This option also requires the video be prepared in three different formats, which can be troublesome. Also the videos must be hosted on the server which is a potential drawback. This is a possible option, as we do not have to worry about compatibility or ads, but also it is more complicated to add to the webpage.

4.2 YouTube

The option is the most common way to embed a video. It is the simplest to implement as all it requires is to click on the *Embed* button and copy the code into the web page. YouTube is also easily accessible on its own. This option is highly compatible with many browsers. There is also no need to host videos on the server, but then the website would have to depend on YouTube to host which is a potential drawback. YouTube videos may also have ads. However, this is the simplest and most accessible solution for this project.

4.3 Vimeo

This site offers free video hosting. The player is customizable unlike the YouTube player (if you pay for Premium), and it also just as simple to upload and embed. Paying for premium for a customizable player is not necessary as the video is not much of a priority for the simulation web page. It also does not show in HD unless paying for premium which is also a drawback. Vimeo also offers compatibility for other browsers and free video hosting like the previous two options[6].

5 CONCLUSION

There are many libraries and methods to use when developing a web page. Each one has its own benefits that pertain to each requirement priority of creating a 2D mechanical simulation. Popmotion.io is a suitable animation library since its robust, well documented, and versatile. Bulma.io and Bootstrap are both well documented and versatile styling libraries, but Bulma.io is simpler and modular, and does not take much to incorporate into the library. For video players, YouTube is the most common and easiest solution to embed videos.

REFERENCES

- [1] daneden. (2018) animate.css. [Online]. Available: <https://github.com/daneden/animate.css>
- [2] G. Sock. (2018) Gsap, the standard for javascript html5 animation. [Online]. Available: <https://greensock.com/gsap>
- [3] (2018) Introduction - bootstrap. [Online]. Available: <http://getbootstrap.com/docs/4.1/getting-started/introduction/>
- [4] (2018) Overview — bulma. [Online]. Available: <https://bulma.io/documentation/overview/>
- [5] (2018) Semantic ui. [Online]. Available: <https://semantic-ui.com/>
- [6] (2018) Embed video on website: 5 easy ways. [Online]. Available: <https://www.freemake.com/blog/embed-video/>