TDE
DIGITAL CREATIVES IN SPORTS

Front-end Framework Comparison

Improving the Grand Prix experience for F1 viewers at home

S8 Graduation FHICT

4 Sept 2023 - 16 Jan 2024

By Jordi Franssen

6 November 2023

www.tde.nl

Introduction

The tool will be built for a web environment. Of course a front-end framework will be used to build the tool. As there are multiple options, a decision has to be made on what front-end framework to use.

Svelte

TDE uses Svelte as their go-to front-end framework. As I built my portfolio website in Svelte, I'm already familiar with it. So, if possible, it would be ideal to use Svelte as well for my tool. However, there are multiple frameworks for Svelte development. Just like there's Next for React and Nuxt for Vue, Svelte has SvelteKit that allows for more efficient development than when native Svelte is used.

React-three-fiber

In the ThreeJS course I followed at the beginning of the project, I found out that React has a special framework for integration with ThreeJS called React-three-fiber. Now I found out that Svelte also has a similar framework called Threlte. This framework is based on Svelte and ThreeJS and allows for a much faster development experience than with native ThreeJS.

I have tried to apply Threlte in the YukaJS POC I made. However, I found out that it's really difficult to combine Threlte with YukaJS. That's why it's required to investigate if it's possible to use Threlte or to use SvelteKit after all.

SvelteKit or Threlte

YukaJS

As explained above, Threlte cannot be combined with YukaJS. Therefore, SvelteKit has to be used if YukaJS will be used in the project. It's also possible to use Threlte for the ThreeJS parts of the tool that don't use YukaJS. However, Threlte is build on Svelte and not SvelteKit. That means that I would lose the benefits of SvelteKit.

Development approach

I am realizing that the tool that I will be building is going to be quite complex. My priority is to have a functional product as soon as possible. Therefore, I plan to start with a really basic version of the tool. The simulation part will just be a simple version of a racetrack, with dots representing cars. If there's any time left, it's possible to further develop this into a more advanced version with actual race cars driving on the track.

Considering that the datasource used updates about every 230 milliseconds, it might even not be necessary to animate the movement of the cars. So, for the first version, Yuka will definitely not be used.

Expert Interview

Before I decide to start building in Threlte, I first want to discuss my approach with John, who is an expert on Svelte development. I decided to do an open interview with John because I think this is the easiest way to get an answer on the question whether to use SvelteKit or Threlte as I don't expect to easily find an answer on the internet because my case is quite specific. Therefore, I explained what I needed to know and we had a general discussion on what's the best approach for my project.

Interview results

John pointed out that it's very unlikely that YukaJS will be used in the tool, so Threlte can be used for development. John also explained that using SvelteKit isn't an option, as RN365 is built on Svelte and things like routing are customly built that would make it incompatible with the standard routing provided by Sveltekit.

We also discovered that Threlte doesn't have to be a separate framework at all. Threlte can also be installed as a separate module in Svelte. This allows me to build the tool in native Svelte after all. Therefore, I could use the TDE webpack boilerplate that TDE uses to build applications.

However, there's a more interesting option, which is to make a branch of RN365 and integrate the tool in the website from the beginning. This allows me to actually build my tool as it would be used in the website. It also saves time because I don't have to rewrite the entire styling of RN365. This also comes with a challenge. In order to build my tool in the RN365 website, I need to fully understand the structure and code of RN365, but this can be discussed at a later stage.

Conclusion

Based on the fact that YukaJS will not be used, that Threlte can be used as a separate framework on top of native Svelte and that SvelteKit is incompatible with RN365, it makes sense to build the tool directly into the project of RN365, which is build on native Svelte.

Summary

The tool will be built in a web environment and therefore, a front-end framework will be implemented. I conducted research on which front-end framework fits best for my project. TDE uses Svelte as their standard front-end framework, but in the ThreeJS course, I learned that React has a special front-end framework for ThreeJS called React-Three-Fiber. Later I found that Threlte is a similar framework that's built upon SvelteKit. However, using Threlte makes it almost impossible to implement YukaJS, but it's almost certain that YukaJS won't be used anyway.

I conducted an open expert interview to discuss my findings on which front-end framework to use. Expert John was able to tell that Threlte can also be used as a module within Svelte and not as a standalone framework based on SvelteKit. John also opted to implement the tool in the RN365 repository right away and not to make a separate MVP. This also makes it impossible to use SvelteKit as RN365 uses custom routing that's incompatible with the pre-baked routing built into SvelteKit. Therefore, we came to the conclusion to build the tool directly into the RN365 repository, and therefore use Svelte with Threlte as a module.

Learning Outcome Clarification

- Learning Outcome 1: Professional Duties
- Learning Outcome 2: Situation-Orientation
- Learning Outcome 4: Investigative Problem Solving
- Learning Outcome 6: Targeted Interaction

This deliverable is a professional duty on a bachelor level in the activities of Analysis and Advise as I analyzed what is the best front-end framework to use for the tool and advised what front-end framework to use and how the tool will be built. Therefore, Learning Outcome 1: Professional Duties applies.

By deciding to build my tool directly into the RN365 repository and therefore using Svelte as front-end framework, I adapt to the processes and way of working of the company as TDE uses Svelte as their standard front-end framework. This deliverable is also relevant and valuable for one or more persons and creates value as I now know how to build the tool in the development phase. Therefore, Learning Outcome 2: Situation-Orientation applies.

I identified a problem, which is that I didn't know what front-end framework to use for the development of my tool and I found an effective approach and arrived at appropriate solutions by using a variety of research strategies, methods and activities as I conducted an expert interview to come to a conclusion on which front-end framework to use. Therefore, Learning Outcome 4: Investigative Problem Solving applies.

I collaborated appropriately with expert John, a partner in my project to achieve the desired impact. Therefore, Learning Outcome 6: Targeted Interaction applies.