TDE
DIGITAL CREATIVES IN SPORTS

Available Technologies Analysis

Improving the Grand Prix experience for F1 viewers at home

S8 Graduation FHICT

4 Sept 2023 - 16 Jan 2024

By Jordi Franssen

8 September 2023

www.tde.nl

Introduction 3
ThreeJS-Journey 4
Blender3D 5
Loading models in ThreeJS 6
React-three-fiber 8
Yuka js Library 9
Conclusion 11
Summary 12
Learning Outcome Clarification 13

Introduction

This project will involve multiple technologies that are new to me and some of them contain quite a steep learning curve. ThreeJS is probably the largest one. Fortunately, TDE has a course available that consists of about 80 hours of material. Next I'll write small summaries of technologies that I found during my research. In the end I'll conclude how these technologies could be useful for the project.

ThreeJS-Journey

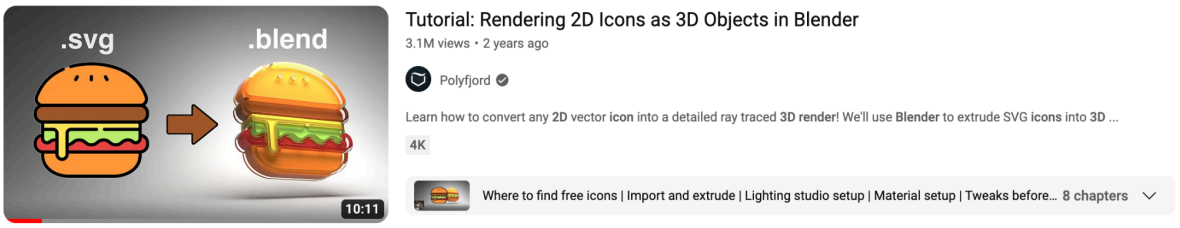
ThreeJS-Journey is an online course consisting of almost 80 hours of video material. The course is built by Bruno Simon, a creative web developer who's famous for his work in 3D web development and his awesome 3D portfolio.

Plowing through all 80 hours of material is definitely not going to happen. Therefore, I first watched the videos explaining the basics of ThreeJS. With this knowledge, I got to know the basic possibilities of ThreeJS.

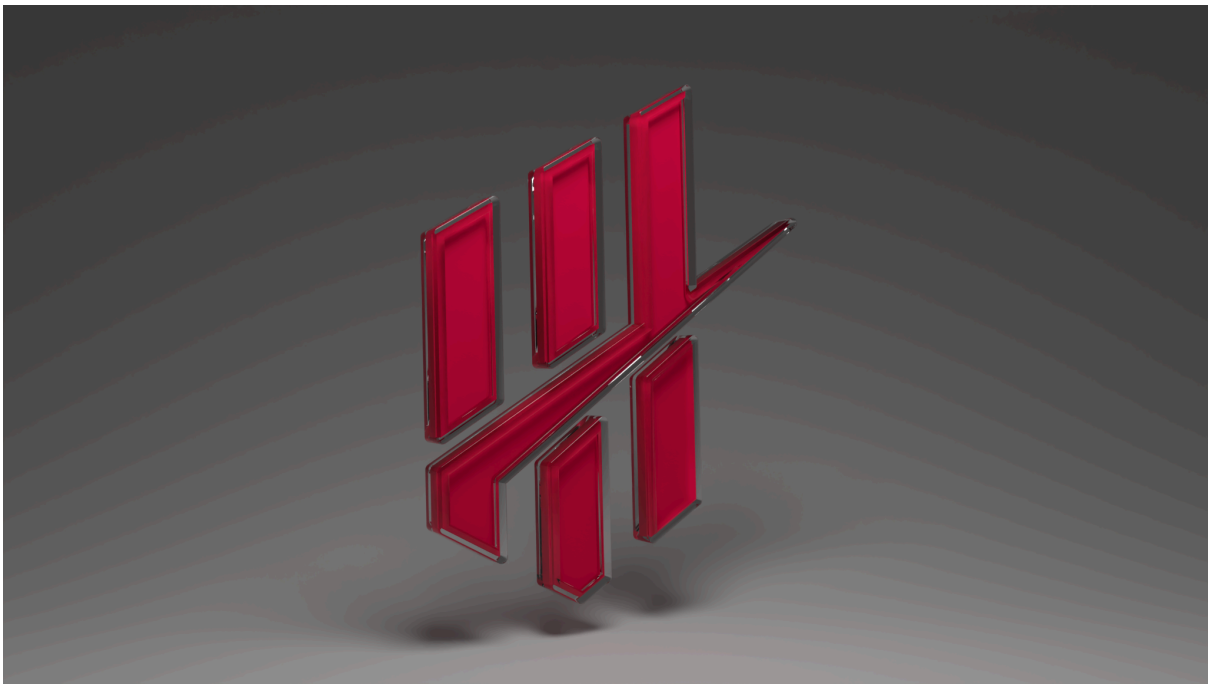
This got me the idea to 3D animate the logo of TDE in a web environment to make a first proof of concept. However, first I needed a 3D model of the logo to load in ThreeJS. I made this logo with Blender3D.

Blender3D

Using Blender, I made my first ever 3D model. I had access to the TDE digital assets. Here I found an SVG of their logo. The only thing I had to do is to load the SVG in Blender and make it 3D. Therefore, I followed this tutorial:



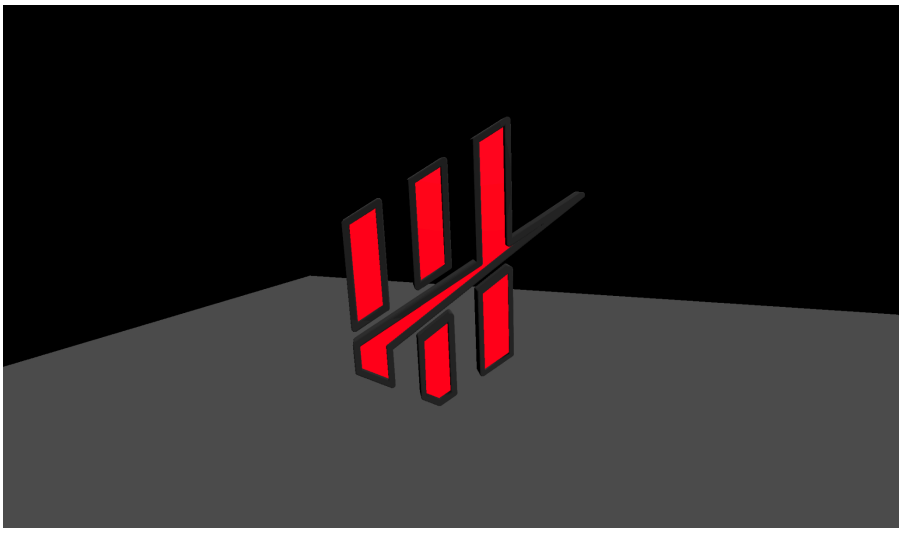
Tutorial: Rendering 2D Icons as 3D Objects in Blender



With the help of this tutorial, I was able to render this cool 3D TDE logo! After figuring out how to export this to a GLB file, I was able to load the model in ThreeJS.

Loading models in ThreeJS

The ThreeJS-Journey course has some great examples that I used to experiment with and I was able to load my own TDE 3D model in a few examples.



info@tde.nl
(+31) 040 782 00 01
www.tde.nl

S8 Graduation FHICT
s8-graduation.jordifranssen.com
Jordi Franssen

In the second example, I loaded the model in an environment to show that lighting has an effect on the model's colors. The outer lines of the model have a different color for both examples. This is because I made this line out of a glass material in Blender which is transparent. However, it doesn't look like the render I made in Blender itself. This is probably because the model is missing certain textures. Textures is a quite advanced topic in ThreeJS and I'll probably dive into this at a later stage of the project.

React-three-fiber

TDE uses Svelte and SvelteKit as their go-to front-end framework. Svelte and ThreeJS can work together quite well. However, React-three-fiber is a front-end framework that's based around React and it makes integrating 3D graphics in a front-end framework much easier.

I can always just build my internship project in Svelte because TDE says I have to. However, I think it's worth the extra effort to investigate what actually is the best front-end framework for my project and bring advice for the development phase. This will be done in a separate document during the development phase.

Yuka js Library

During my research, I stumbled upon YukaJS. Yuka is a framework that can be used in combination with ThreeJS to implement steering of an object in a 3D environment. This enables the simulation of cars for example. Yuka includes a series of steering behaviors that can be used in different scenarios. Below you'll find a list of available steering behaviors:

- Arrive steering behavior, decelerates a vehicle onto a target position
- Flee steering behavior, makes a vehicle flee from a target position
- Flock steering behavior, group steering behavior using multiple vehicles
- Follow path behavior, makes a vehicle follow a series of waypoints
- interpose steering behavior, makes a vehicle move to the midpoint of two positions
- Obstacle avoidance behavior, makes a vehicle avoid an obstacle in its path
- Offset pursuit behavior, makes a vehicle follow another vehicle at a specific distance
- Pursuit behavior, makes a vehicle trying to intercept another vehicle
- Wander behavior, makes a vehicle steer randomly in an environment

Create A Self-Driving Car Using Three.js & Yuka - 3D/AI Tutorial
7.5K views • 1 year ago
Wael Yasmina
This is like an exercise to practice what we've learned in the previous **tutorial**. --- Article Soon --- Path Following Steering Behavior ...

To get a good understanding of this technology, I made a small POC that I might add to my portfolio. I started with the code from a tutorial on YouTube and changed it until I got a car driving around a webpage. The car drives fully autonomously by trying to intercept a virtual position on the screen based on the vertical scrollbar.

To make the POC a bit more dynamic, I added two blocks which the car has to avoid. The result is a car driving around the screen, chasing your scroll position.

Conclusion

During my own TheeJS journey, I've heard multiple times that the possibilities are limitless. This of course doesn't mean much, but I found that TheeJS is quite capable. It's possible to visualize as much detail as you want, as long as the hardware is powerful enough. ThreeJS feels a bit like Unity3D but scaled down to fit in a web package. Therefore, it's certainly possible to build entire games with it, or make highly realistic digital twins of real-world items or environments. ThreeJS also supports physics, which allows for running simulations and more.

Next, there's React-three-fiber which is a front-end framework built to make integrating ThreeJS in React or Next much easier.

At last, Yuka is a library that might be really interesting for the general goal of this project; to visualize Formula 1 data in 3D. Yuka enables you to simulate vehicle behavior in a 3D web environment. This means it's also possible to simulate entire F1 races in 3D, based on timing data.

Summary

This project will involve multiple technologies that are new to me and some of them contain quite a steep learning curve. ThreeJS is probably the largest one. Fortunately, TDE has a course available that consists of about 80 hours of material. Next I'll write small summaries of technologies that I found during my research. In the end I'll conclude how these technologies could be useful for the project.

During my own TheeJS journey, I've heard multiple times that the possibilities are limitless. This of course doesn't mean much, but I found that TheeJS is quite capable. It's possible to visualize as much detail as you want, as long as the hardware is powerful enough. ThreeJS feels a bit like Unity3D but scaled down to fit in a web package. Therefore, it's certainly possible to build entire games with it, or make highly realistic digital twins of real-world items or environments. ThreeJS also supports physics, which allows for running simulations and more.

Next, there's React-three-fiber which is a front-end framework built to make integrating ThreeJS in React or Next much easier.

At last, Yuka is a library that might be really interesting for the general goal of this project; to visualize Formula 1 data in 3D. Yuka enables you to simulate vehicle behavior in a 3D web environment. This means it's also possible to simulate entire F1 races in 3D, based on timing data.

Learning Outcome Clarification

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

This deliverable is a professional product on a bachelor level in the activities of Analysis and Advise as I analyzed technologies useful for my project and advised on the use of the found technologies. This is in line with IT-area User Interaction. Therefore, Learning Outcome 1: Professional Duties applies.

This deliverable is relevant and valuable as it plays a role in the orientation phase of the project. I also worked in a methodological and structured way. Therefore, Learning Outcome 2: Situation-Orientation applies.

This deliverable is a research method on ictresearchmethods.nl. and it is an effective approach to find valuable insights for the orientation phase. I evaluated my findings by presenting them during the focusgroup discussion and asked for feedback on my research. Therefore, Learning Outcome 4: Investigative Problem Solving applies.