TDE
DIGITAL CREATIVES IN SPORTS

Advice for Further Development

Improving the Grand Prix experience for F1 viewers at home

S8 Graduation FHICT

4 Sept 2023 - 16 Jan 2024

By Jordi Franssen

10 January 2024

www.tde.nl

Introduction

This project aimed to improve the experience for Formula 1 viewers at home. This is a broad objective and at the start of the project, it wasn't clear what kind of digital solution would solve this problem statement. In the course of the project this objective shifted towards the goal of improving the current liveblog for RacingNews365. Now at the end of the project, an MVP (Minimum Viable Product) has been delivered that overhauls the liveblog experience for both RacingNews365 and the liveblog user itself.

An MVP isn't a final product however. In this document I'll explain how this MVP would run in real life and include my advice on how to elaborate on this MVP to make it a final product that's ready for production. In addition, I'll include possible additional features that would elevate the success of this project.

Project Overview

This project overview summarizes the project and the activities that have been carried out.

Project Assignment

As explained in the introduction. The initial assignment for this project was to improve the Formula 1 experience for viewers at home. It's a project by TDE with RacingNews365 as stakeholder and the problem that was discovered was that the new RacingNews365 premium subscription numbers were stalling significantly. Therefore, the plan was to enrich the premium plan with new features that utilize Formula 1 data to gain more subscribers.

Concepting

Because the objective of this assignment was so broad, a thorough orientation and research phase was required to define a concept that would fit the problem statement. It was discovered that the target audience for RacingNews365 is ready for a better overview of a race. Therefore, the idea was to make a timeline with events that are automatically generated based on live racing data so the user could scroll through the events to find out what happened during the course of the race. Because the user could also use more information and context, a 3D visualization of the race would be added so the user could see where events were happening on the track.

Later, the idea of an automated timeline was combined with the current liveblog of RacingNews365. However, RacingNews365 wants control over the liveblog. Therefore, the automated timeline is added to the CMS of the liveblog so the editor can use it as input for the existing liveblog. Some features were added to the liveblog to provide a better overview, like the track status indicator for example.

Design

The design phase was relatively short in this project. It was decided that the project would be directly implemented in the repository of RacingNews365. This meant that existing components like liveblog items could simply be reused with some small modifications.

To test the user interaction of the concept, I designed some wireframes for the new liveblog. Because there's no official brand guide available for RacingNews365, I did a visual audit on the current website and made a temporary brand guide to apply the corporate branding of RacingNews365 to the wireframes I made. This way I could usertest a functional prototype.

Development

The development phase required some research as well. For one, the software and server architecture needed to be discussed. For now, a development API is built to simulate the data stream that comes from the Formula 1 API.

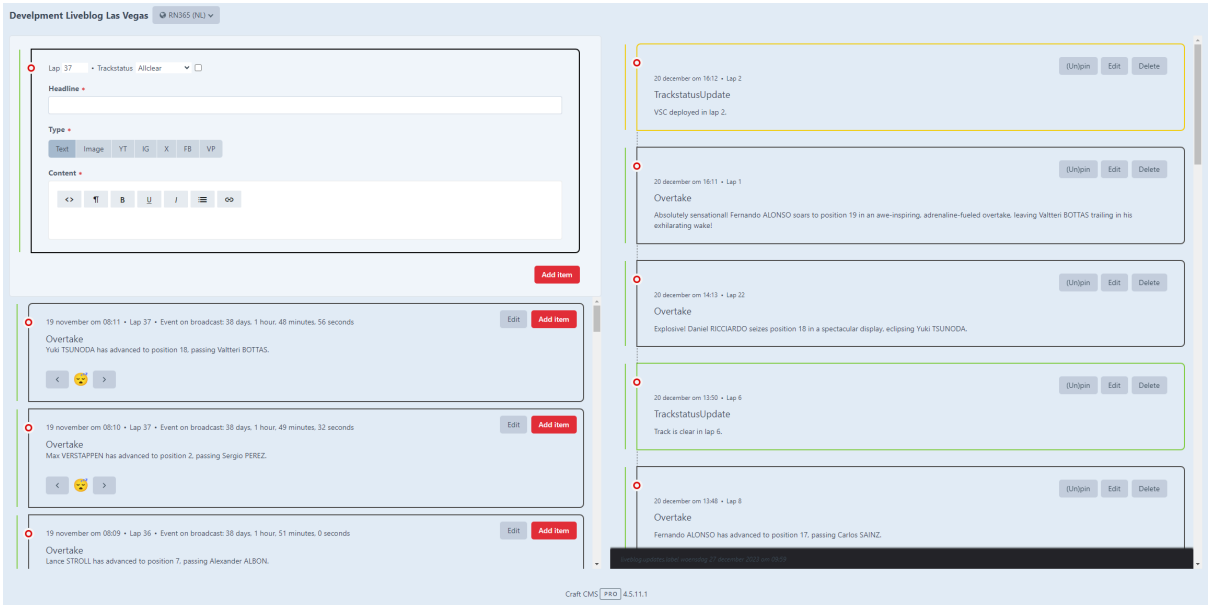
A separate script that runs on the server receives the data from Formula 1 and processes it to detect the events happening during a race. Data analysis was conducted to find out how to detect certain events. Overtakes, pitlane enters and exits including the tyre compound, track status, session status and potential crashes are now detected. Research was conducted to confirm that SSE (Server Sent Events) is the best communication method to send these events to the clients. However, if this script would run on a AWS server, SSE would have to be replaced with Websockets.

Final Deliverable

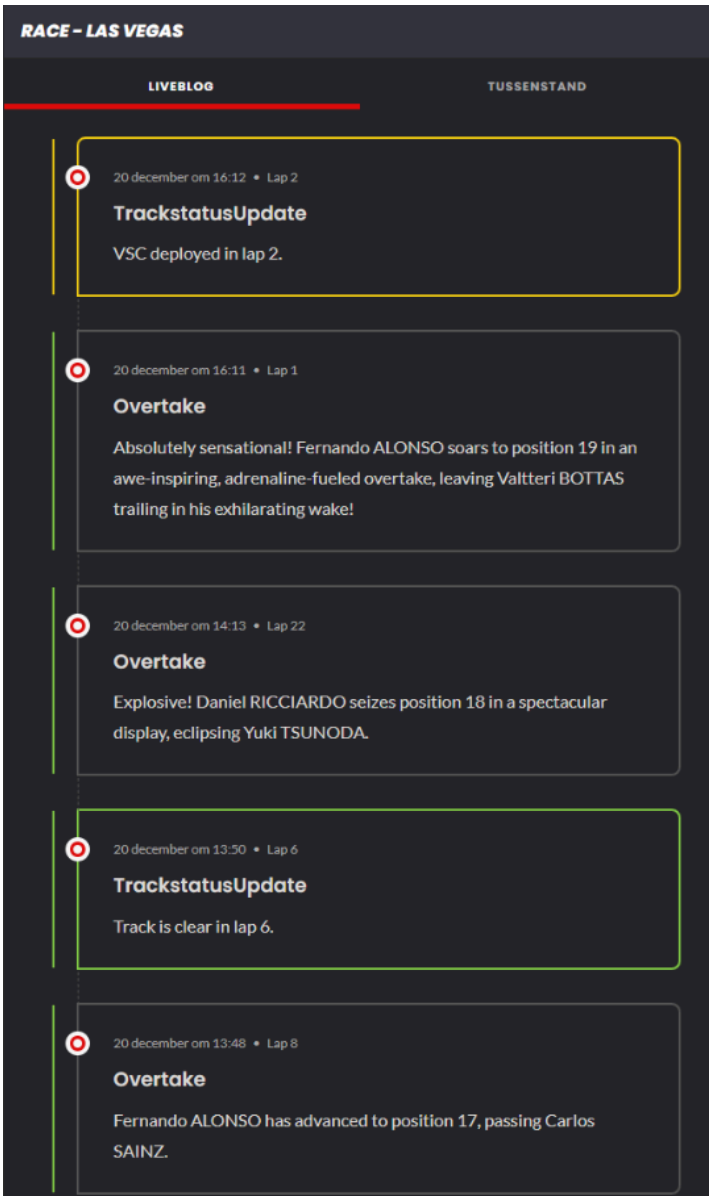
The final deliverable is an MVP. This means that functionality has priority over quality. This applies to the code of the MVP. John reviewed the changes in the commits in Gitlab. Surprisingly, John didn't find any Major issues with the code, but don't expect the code to be worthy for an end product.

In this MVP, the CMS from RacingNews365 is modified so it receives and displays the automatically detected events. These events are presented like how they would look in the actual liveblog, only with a few buttons to edit an item or post it right away.

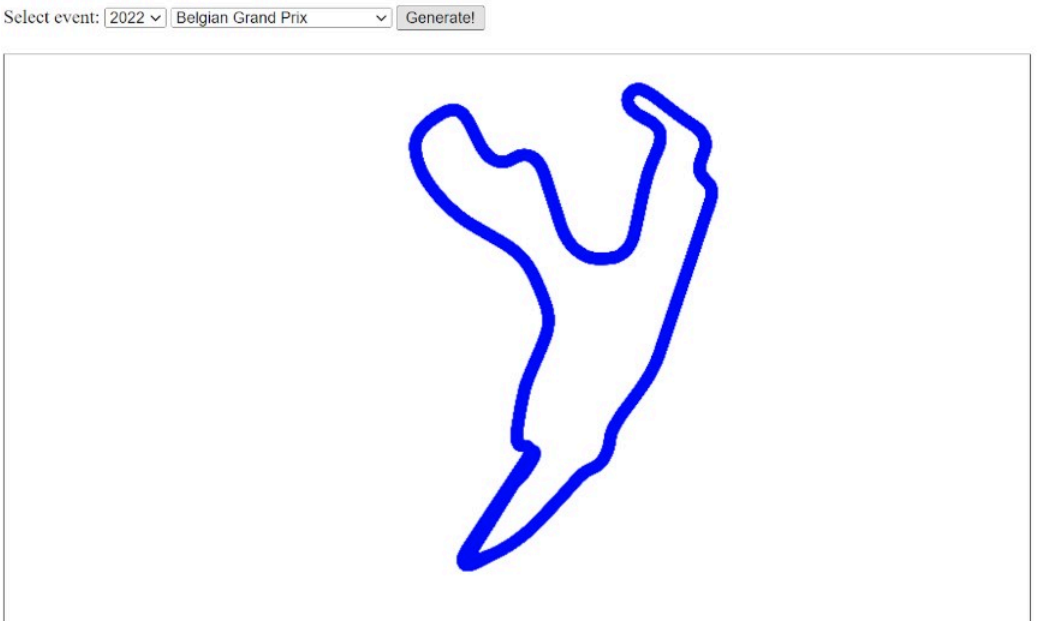
The concept also includes the position visualizations of where an event takes place. First the idea was to visualize the entire race in 3D. However, because only the position of an event is visualized, a 3 dimensional feature doesn't add any more value than just a simple image with a dot that represents an event. The challenge with this feature is the scale and size of the image used. Therefore, I built a tool that uses the position data from the cars to draw a shape that can later be used in the CMS. Only the tool has been built. The functionality itself is yet to be developed in the CMS.



Left bottom is the automated event feed



Updated liveblog with new features



Tool to generate the shape of a track

Running the MVP

The project was built in the repository from RacingNews, racingnews-v2. The changes can be found in the branch “feature/liveblog-improvements”. Run the project like any other project using Docker and DDEV.

Run the MVP with the development API

The development API was built so the project could be developed and tested at any time. For this API, the live data from the 2023 Las Vegas GP was recorded, and the development API simply “replays” this recorded data so the CMS of RacingNews365 can access it. To use the development API, run “DevelopmentAPI.js” and “BackendSSE.js” in a node server.

```
let fileId = 835; //Start race: 835
```

On line 19 in “DevelopmentAPI.js”, a variable is defined called “fileId”. This variable is set to 835 and represents the amount of seconds from the moment the recording started. The race started at 835 seconds. This variable can be modified to start the development API at a different moment in the race. Another interesting moment is the crash from Lando Norris for example. This takes place from 1150 seconds after the recording started.

If these two servers are running, the detected events can be retrieved in the CMS.

Running the MVP with live data

The MVP can also run using live data during a race. This essentially works the same, except “DevelopmentAPI.js”, run “liveAPI.js”. Instead of reading the recorded data from the Vegas GP, this API hooks onto the live data stream from Formula 1 and sends it to “BackendSSE.js” using SSE. Starting the project with Docker and DDEV works exactly the same.

From MVP to Final Product

Before the new liveblog can be used on the live website, it has to be elaborated into a final product. There are a few things that need to be changed.

What needs to be improved?

Location Visualization

Developing the location visualization function turned out to be more challenging than expected. The tool to generate a scaled image of the shape of the track works quite well. However the shape is mirrored and cropping the canvas to where the image is generated is quite difficult.

Cropping the image could be done by generating the image, then saving the width and height of the shape, using these values to resize the canvas and rerender the shape based on the data with the values subtracted.

Then this shape somehow needs to be included in a liveblog item. I first tried using .svg, as .svg is a text based image that can easily be passed as an input field in the form on the CMS. However, rendering an .svg gave some performance issues, so HTML canvas was used instead. This could mean that the shape needs to be exported from the canvas in an image file. Since this isn't a text based image, some backend development is required to pass the image in a liveblog item. There are some modules available that could export a canvas to .svg, https://gliffy.github.io/canvas2svg/.

It was discussed with RacingNews365 that the location information would only be relevant for overtakes and potential crashes. So the backend should include the image with the position for these events. The location data for these events is already sent as a property with these liveblog items.

RacingNews365 should also be able to manually add a position visualization. I had in mind to make some kind of popup in the form that brings up the shape of the track. Based on the x and y values of a click event on this shape, the position of this click can be calculated and passed with the liveblog item.

Detecting Events

The script for detecting events is quite messy. ChatGPT was used to help with the logic for detecting the events. However, this document isn't well structured. Ideally, every type of event would need its own function with logic to detect this event, but now some functions detect multiple events which makes the script difficult to debug and read.

The script also detects pitlane enters and exits. This is done based on the InPit variable in the data source. When the race is started, the backend detects that InPit is false for all drivers, and therefore detects a pitlane exit for each car.

For now, only overtakes, pitlane events, track status and potential crashes are detected. Things like fastest lap, or fastest speed trap could be added to generate more events.

CMS layout

The liveblog page in the CMS is quite busy with the detected events added. This layout could be optimized. The form for manually adding a liveblog item could be hidden in a popup for example. This way the automated liveblog and the actual liveblog can be easily separated to make the page more user friendly.

Custom text suggestions

Each event has 20 different suggestions for the text for this event. These suggestions range from least exuberant to most exuberant to give RacingNews365 some control over the tone of voice in their liveblog. However, these suggestions are now in English

and they're generated with ChatGPT. Ideally RacingNews should bring some pre-written texts how they like them. These can then be coded in the project.

What could be added?

Group events into separate lists

The list of detected events is now cut off from 25 items. I have never seen so many events detected at the same time that the editor won't be able to keep up. However, if more events will be detected, like fastest laptimes etc., it might be interesting to look for a way to group the events per lap, or per 10 laps for example. These separate groups can be folded down so they won't take up too much space. This way a lot more events, and maybe the entire race can be saved in the list.

What should be changed?

Server architecture

Since it's decided that the position of the racing cars isn't continuously displayed, the entire server architecture can be simplified. In a meeting with Jordi it was discussed that a new backend was necessary to continuously process the data from the API to send it to the liveblog users. However, now only the position of an event is added when a car is stationary, or an overtake takes place, the entire logic for receiving the API data and processing it into events can be incorporated into the CMS.

Project Timeline

The project could launch as soon as the 2024 season. If there's plenty of time to elaborate the MVP into a final product before the pre-season test, the project could test it's full potential in February 21-23, allowing for some final touches before the first racing weekend of 2024 which is planned for February 29 - March 2nd.

Summary

This document contains the advice for further development of the project. It can also be seen as a transfer document for the project. In this document, I explained the project overview including the project assignment, the different project phases and what has been delivered at the end. Next I explain how the MVP should be started and how it runs with the development API and with the API when using live data. At last I explained how the MVP should be developed into a final product that's ready for production. Location visualization, the CMS layout, event detection and text suggestions can be improved. Features like grouping events could be added. And the server architecture should be simplified. It would be ideal if the final product could be tested during the pre-season test, and launched before the first race of the season.

Learning Outcome Clarification

- Learning Outcome 1: Professional Duties
- Learning Outcome 3: Future-Oriented Organisation
- Learning Outcome 6: Targeted Interaction

Learning outcome 1 applies to this deliverable as I advised on the development of the project.

Learning outcome 3 applies to this deliverable as I advised on the future of this project.

Learning outcome 6 applies to this deliverable as this deliverable is written for colleagues who will continue this project. Therefore, I communicated appropriately to ensure the desired impact.