



F1 22

Formula 1  
LICENSED  
PRODUCT

# *Aplikacja analityczna wspomagająca rozwój wirtualnego kierowcy F1*

Autor: Rafał Zakrzewski

Promotor: mgr inż. Paweł Kamiński

# Cel pracy:

Celem pracy jest omówienie implementacji przeglądarkowej aplikacji internetowej, która poprzez analizę zgromadzonych danych telemetrycznych wspomoże rozwój wirtualnego kierowcy gry komputerowej „F1 22”.

Rozwiązanie ma oferować łatwy i czytelny dostęp do najbardziej istotnych informacji z sesji kierowcy. Do takich informacji należy: wyświetlanie w czasie rzeczywistym ewentualnych usterek bolidu, aktualnego zużycia i temperatury opon, temperatury hamulców, poziom zużycia systemu ERS.

Zgromadzone dane telemetryczne będą umożliwiały na przeglądanie historii okrążeń i porównywania ich szczegółowych danych, jednocześnie informując użytkownika, gdzie utracił, a gdzie zyskał przewagę czasową oraz co było prawdopodobną przyczyną w przypadku straty.

Użytkownik w aplikacji ma zapewnioną możliwość przeglądania przeróżnych wykresów, które mogą reprezentować na przykład: ile procent gazu/hamulca aplikował, którego biegu używał, bądź z jaką prędkością się poruszał na przestrzeni czasowej wybranego okrążenia sesji kierowcy.



# Zdefiniowany problem badawczy:

**Grono odbiorców tematu Formuły 1 jest ogromne, ponadto spora część tych odbiorców to młodzi mężczyźni, którzy lubią również rywalizować na torze, aczkolwiek w grach komputerowych, ponieważ nie każdy ma wystarczające zasoby finansowe, bądź jest na tyle zdolny, by móc udać się na zawody wyścigowe na prawdziwym torze.**

**Pomimo tego, że jest to aplikacja związana z grą komputerową, chciałbym zaznaczyć, że istnieje ogromna liczba wydarzeń e-sportowych z grą F1, w których najlepsi kierowcy walczą o ogromne stawki pieniężne oraz każdy prawdziwy zespół Formuły 1 zatrudnia po 2 graczy, aby reprezentowali oni zespół podczas oficjalnych, światowych zawodów e-sportowych organizowanych przez FIA organizację, która jest właścicielem Formuły 1.**



Logo organizacji FIA



e-sportowy Mistrz świata  
Jarno Opmeer

**Ważne jest, aby aplikacja dostarczała istotne dane w sposób łatwy do odczytu, projektując przyjemną dla oka oraz intuicyjną szatę graficzną interfejsu. Kluczowe jest, by aplikacja niezależnie od tego czy korzysta z niej początkujący gracz czysto rozrywkowo, czy aktualny e-sportowy mistrz świata Jarno Opmeer żyjący głównie z profesjonalnego grania w serię gier F1, zapewniała radość i satysfakcję z użytkowania oraz umożliwiała na szeroki wachlarz rozwinięcia umiejętności wirtualnego kierowcy.**

**F1™ 22**

# Przegląd wiedzy:

Dane telemetryczne wysyłane są przez grę komputerową „F1 22” protokołem UDP w postaci buforu o enkodowaniu „Little Endian”. Wysyłane są one w postaci kilku pakietów, każdy pakiet należy do innej kategorii i reprezentuje inny typ danych telemetrycznych. Każdy typ pakietu posiada swoją stałą długość bufora, która jest unikalna dzięki czemu przy wykorzystaniu dokumentacji telemetrii gry, można w łatwy sposób odróżnić typ odbieranej treści.

## Motion Packet

The motion packet gives physics data for all the cars being driven. There is additional data for the car being driven with the goal of being able to drive a motion platform setup.

N.B. For the normalised vectors below, to convert to float values divide by 32767.0f – 16-bit signed values are used to pack the data and on the assumption that direction values are always between -1.0f and 1.0f.

Frequency: Rate as specified in menus

Size: 1464 bytes

Version: 1

```
server.on('message', (msg, info) => {
  switch(msg.byteLength) {
    case 1464:
      //console.log("Motion");
      break;
```

The following data types are used in the structures:

Type	Description
uint8	Unsigned 8-bit integer
int8	Signed 8-bit integer
uint16	Unsigned 16-bit integer
int16	Signed 16-bit integer
uint32	Unsigned 32-bit integer
float	Floating point (32-bit)
uint64	Unsigned 64-bit integer

## Packet IDs

The packets IDs are as follows:

Packet Name	Value	Description
Motion	0	Contains all motion data for player's car – only sent while player is in control
Session	1	Data about the session – track, time left
Lap Data	2	Data about all the lap times of cars in the session
Event	3	Various notable events that happen during a session
Participants	4	List of participants in the session, mostly relevant for multiplayer
Car Setups	5	Packet detailing car setups for cars in the race
Car Telemetry	6	Telemetry data for all cars
Car Status	7	Status data for all cars
Final Classification	8	Final classification confirmation at the end of a race
Lobby Info	9	Information about players in a multiplayer lobby
Car Damage	10	Damage status for all cars
Session History	11	Lap and tyre data for session



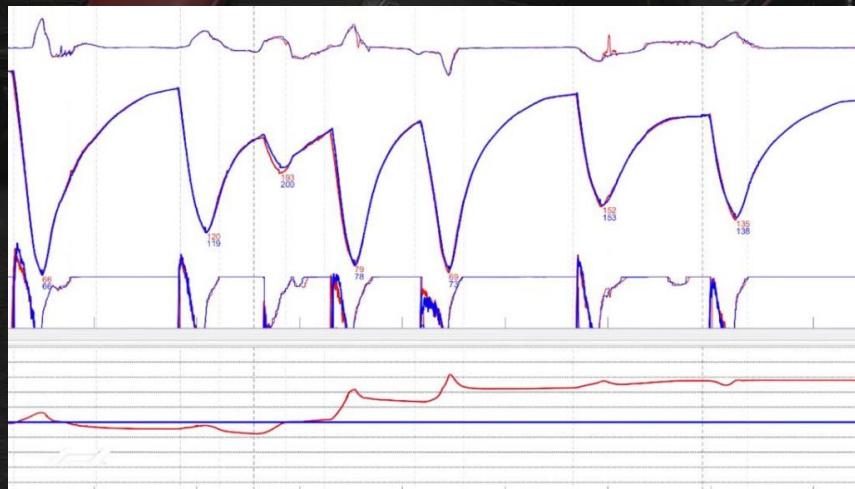
# Przegląd wiedzy:

Wykonanie aplikacji internetowej, która działa głównie po stronie użytkownika na wszystkich współczesnych przeglądarkach tworzona jest przy użyciu bardzo popularnej technologii - frameworku ReactJS oraz dodatkowych bibliotek takich jak react-icons oraz Axios. Framework wykorzystany jest do tworzenia interfejsu, wyglądu i funkcjonalności analitycznych zgromadzonych danych. Oferuje on szeroki zakres możliwości manipulowania strukturą DOM przeglądarki. Stworzony jest przez firmę Meta Platforms. ReactJS jest wykorzystywany między innymi w Facebooku, Instagramie, Netflixie oraz innych bardzo popularnych serwisach.



# Przegląd wiedzy:

Warto by aplikacja oferowała interfejs i dane, które odzwierciedlają interfejs systemów analitycznych prawdziwych zespołów Formuły 1. W rzeczywistym świecie kierowcy F1 posiadają swoich inżynierów wyścigowych i strategów, którzy przeglądają dane bolidu i środowiska, aby niwelować ewentualne błędy i sugerować propozycje dzięki którym kierowca może zyskać czas, pozycję lub punkty. Jeśli chodzi o e-sport i grę F1 22, z danych telemetrycznych może korzystać nie tylko kierowca, który na ogół jest również sam w sobie inżynierem i strategiem. Właśnie między innymi mistrz świata Jarno Opmeer posiada swoich własnych strategów, którzy informują go na bieżąco o sytuacji przeciwników podczas wyścigu. Wystarczy, aby kierowca przekierował z poziomu gry swoją telemetrie UDP na port i adres IP do którego dostęp ma druga osoba. Aby to umożliwić aplikacja analityczna nie może być tylko lokalna, powinna oferować i być dostępna publicznie.



# Planowana technologia aplikacji:

Projekt wykonany przy użyciu współczesnych technologii takich jak framework ReactJS przeznaczony do stworzenia aplikacji klienckiej, interfejsu, wyglądu i funkcjonalności analitycznych zgromadzonych danych. Jest to ostatnia część projektu do wykonania, lecz najbardziej złożona ze względu na to, że to w niej będzie się odbywała analiza danych przy wykorzystaniu zasobów obliczeniowych urządzenia użytkownika.

Dane odbierane po stronie klienta wysyłane są z aplikacji serwerowej.

Aplikacja serwerowa oparta jest na środowisku Node.js z wykorzystaniem kilku kluczowych bibliotek takich jak Express.js, socket.io, czy binary-parser. Głównym zadaniem aplikacji serwerowej jest odbierać dane telemetryczne wysyłane przez grę „F1 22” protokołem UDP w postaci binarnej z enkodowaniem „Little Endian” i przechowywać je w zmodyfikowanej strukturze ze względu na to, że dane telemetryczne gry „F1 22” wysyłane są pojedynczo, z podziałem na kategorie danych, z częstotliwością 20Hz lub 40Hz w zależności od wyboru użytkownika.

# Bibliografia:

1. EA Codemasters Team, *Specification for the UDP telemetry output system for F1 22*,  
<https://answers.ea.com/t5/General-Discussion/F1-22-UDP-Specification/tdp/11551274>, stan z dnia 24.01.2023
2. Meta Platforms. *React – A JavaScript library for building user interfaces*, <https://reactjs.org>, stan z dnia 09.01.2023
3. P. Kamiński, *React. Wprowadzenie do programowania*, Wydawnictwo Helion, 2021
4. react-icons, *SVG react icons of popular icon packs*, <https://github.com/reacticons/react-icons>, stan z dnia 24.01.2023
5. axios, *Promise based HTTP client for the browser and node.js* , <https://axioshttp.com/docs/intro> , stan z dnia 24.01.2023
6. OpenJS Foundation, *Node.js v17.9.1 documentation*, <https://nodejs.org/docs/latestv17.x/api/synopsis.html> , stan z dnia 24.01.2023
7. OpenJS Foundation, *UDP/datagram sockets*, <https://nodejs.org/docs/latestv17.x/api/dgram.html> , stan z dnia 24.01.2023
8. Socket.IO, *Bidirectional and low-latency communication for every platform*, <https://socket.io/docs/v4/> , stan z dnia 24.01.2023
9. keichi, *Parser builder for JavaScript*, <https://github.com/keichi/binary-parser> , stan z dnia 24.01.2023
10. ExpressJS, *Fast, unopinionated, minimalist web framework for Node.js*, <https://expressjs.com/en/4x/api.html> , stan z dnia 24.01.2023