
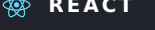


O mnie

Lubię programować komputery za pomocą  ,  . Aktualnie koncentruję się na pisaniu aplikacji webowych w szczególności interfejsów użytkownika.

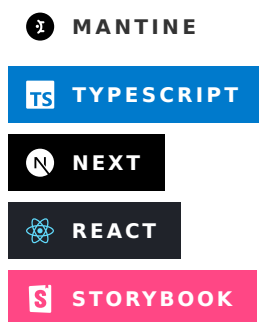
Projekty

Symulator Statków Żaglowych



Symulator/Gra 2D napisana w TypeScript. Użytkownik za pomocą myszki zmienia parametry żagli i steru aby kontrolować statek. Symulator prezentuję esencje działania żaglowca. Można ostrzyć, odpadać, robić zwrot przez rufę, sztag, płynąć z wiatrem, a pod wiatr tylko halsować. Cały statek jest zbudowany z molekuł, które są połączone za pomocą "sprężyn"(soft body dynamics).

Strona Reklamowa Siłowni



PROF

Statyczna, jednostronna aplikacja hostowana i budowana na GH (CI/CD) reklamująca siłownię. Wykorzystuje w niej hooks i effects dla zapewnienia przyjemnego doświadczenia użytkownika. Aplikacja posiada takie elementy jak: rozwijany pasek górny w zależności od przewijania strony, czy możliwość wyboru motywu kolorystycznego, a od strony kodu: izolacja danych od komponentów, podział kodu na warstwy (Stratified Design), programowanie funkcyjne (immutability of data).

Umiejętności

Języki programowania 



C#PYTHONPHP

- **TypeScript** -Aktualnie mój ulubiony język programowania. Cenię w aplikacjach pisanych w TS/JS łatwość uruchamiania na wielu urządzeniach (wystarczy przeglądarka internetowa, lub node). Unikam JS ze względu na brak typowania. Staram się pisać kod w stylu funkcyjnym(korzystać tylko ze stałych), ponieważ jest łatwiejszy w zrozumieniu(brak efektów ubocznych funkcji).
- **C/C++** - Pierwszy język, który poznałem. Lubiłem za pomocą niego programować do czasu kiedy np. zobaczyłem jak wygodnie można posługiwać się funkcjami w TS/JS (obywatele pierwszej klasy). C++ dają więcej kontroli (brak garbage collector).
- **Java** - Drugi język który poznałem, rozwinąłem za pomocą niego koncepcję programowania obiektowego.
- **C#** - Podobny do Javy.
- **Python** -Korzystałem z niego nie zbyt dużo podczas programowania Raspberry Pi. podobny do JS (brak typowania 😞, funkcje obywatelami pierwszej klasy 😊), nie przypadło mi do gustu zastąpienie nawiasów klamrowych tabulacją.
- **PHP** - nie przepadam.

Frameworki/ biblioteki 📖

REACTNEXTJESTPRISMA

- **React** - Po poznaniu React, polubiłem tworzenie interfejsów użytkownika.
- **NextJS** - Wygodny framework/narzędzie, korzystam do budowania aplikacji TS+React.
- **Jest** - Staram się pokryć jak najwięcej kodu testami.
- **Prisma** - fajny ORM z pełnym typowanie.

Narzędzia 🛠️

GITVISUAL STUDIO CODELINUX MINTDOCKER

- **Git** - Używam do kontroli wersji projektów, które piszę. Dla każdej nowej funkcjonalności tworzę nową gałąź i po zakończeniu pracy nad nią łączę ją z gałęzią główną. Staram się nazywać commit-y zgodnie z `conventional commits`.

- **Visual Studio Code** - Jest to moje ulubione IDE. Cenię jego prostotę i uniwersalność (ogromną ilość dodatków).
- **Linux** - System operacyjny o otwartym kodzie źródłowym daje znacznie większą kontrolę nad sprzętem.
- **Docker** - lekkie wirtualne środowiska.

Bazy danych



- **Supabase**- aktualnie tworzę projekt korzystając z Supabase jako backendu. S udostępnia orm, który generuje typy na podstawie utworzonej bazy.

Języki naturalne

- **Angielski** - B2.
- **Polski** - Ojczysty.

Edukacja

PANS w Nysie kierunek Informatyka (2019-2023) (Nie oddana praca dyplomowa)

-->