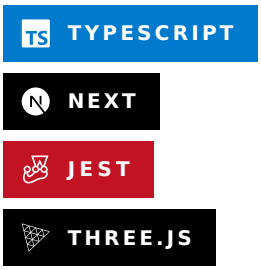
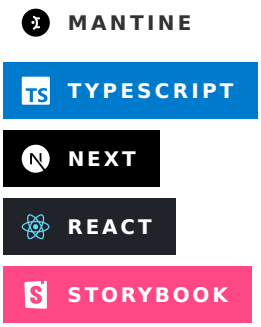
# Projekty 📁

## 🔗 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




Statyczna, jednostronna aplikacja hostowana i budowana na GH (CI/CD) reklamująca siłownie. Wykorzystuje w niej hooki i effecty 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).

## 🔗 Ploter typu SCARA


Projekt ze studiów w ramach kursu 'Systemy Wbudowane'. Jest to ploter zbudowan z ramienia o dwóch przegubach obrotowych z równoległymi osiami obrotu.

Działanie w skrócie:

- Z dowolnego pliku graficznego za pomocą programu Incsape generowany jest `G-code`
- `G-code` jest przesyłany do programu działającego na pc, który przesyła kolejne współrzędne do arduino przez interfejs usb,
- arduino odbiera współrzędne, z równań ruchu wyznacza kąty i za pomocą PWM przesyła je do serwomechanizmów
- serwomechanizmy poruszają ramieniem i powstaje rysunek

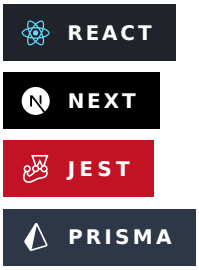
Projekt przede wszystkim dotyczył programowania, więc część mechaniczna jest skromna.
Straciłem kod źródłowy więc link odnosi się tylko do pliku wideo prezentującego działanie plotera.

## Umiejętności 💪

### Języki programowania 💻


- **TypeScript** -Aktualnie mój ulubiony język programowania. Cenię w aplikacjach pisanych w TS/JS łatwość uruchomiania na wielu urządzeniach (wystarczy przeglądarka internetowa, lub node). Unikam JS ze wzgledu 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. Ceniłem go do czasu aż odkryłem jak wygodnie można posługiwać się funkcjami w TS/JS (obywatele pierwszej klasy). Z drugiej strony C++ daję 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 podczas programowania Raspberry Pi i projektów wykorzystujących przetwarzanie obrazu. podobny do JS (brak typowania😕, funkcje obywatelami pierwszej klasy😊), nie przypadło mi do gustu zastąpienie nawiasów klamrowych tabulacją.
- **PHP** - nie przepadam.

### Frameworki/ biblioteki 📚




- **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 🛠️


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

### Języki naturalne 🌐

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

## Doświadczenie

## Edukacja 🎓

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