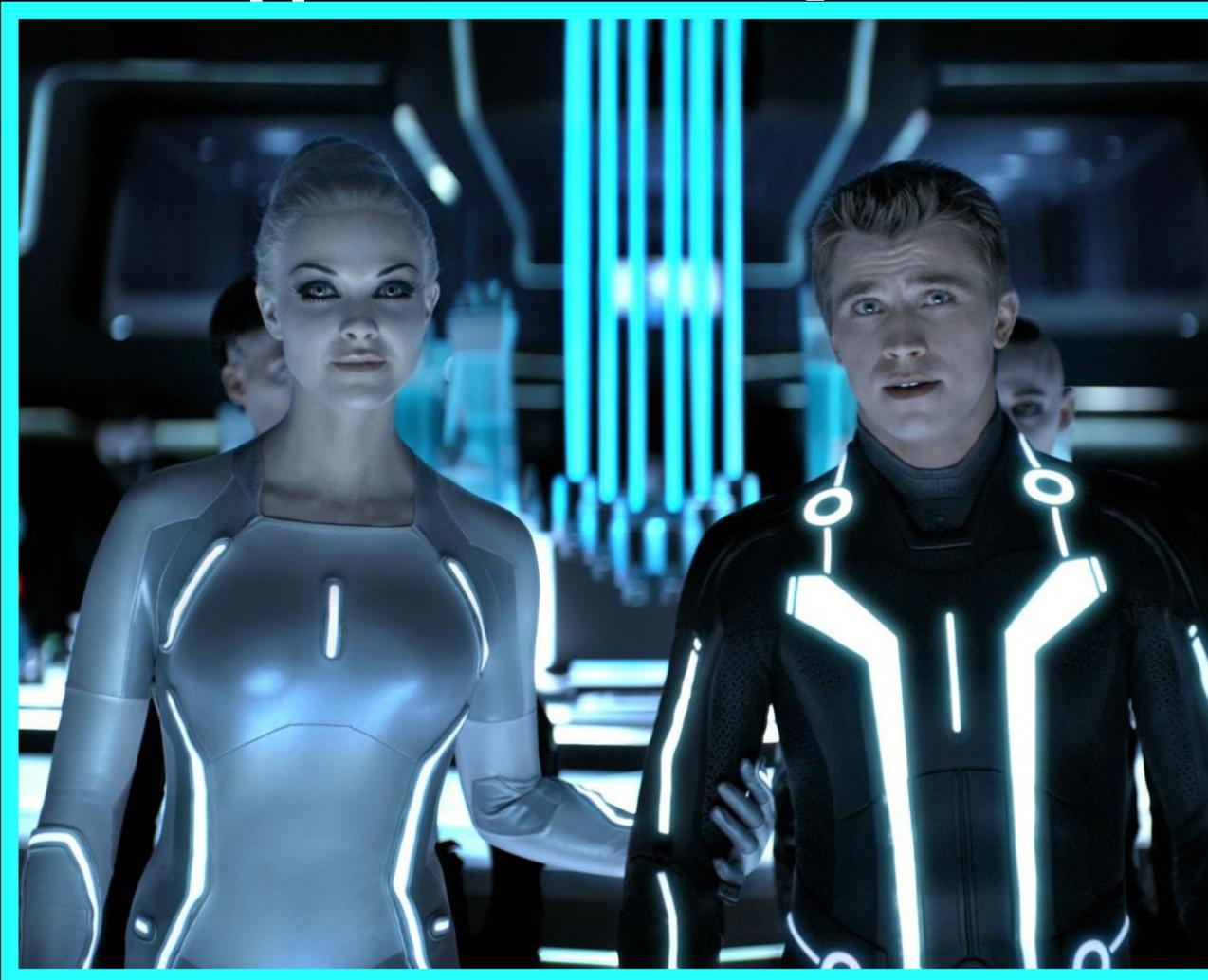


**Короче говоря**

**МЫ СДЕЛАЛИ ТРОН**

**Трон — многопользовательская аркадная игра, основанная на одноименном фильме 1982 года.**



# Новое видение





# CodeLoft

®

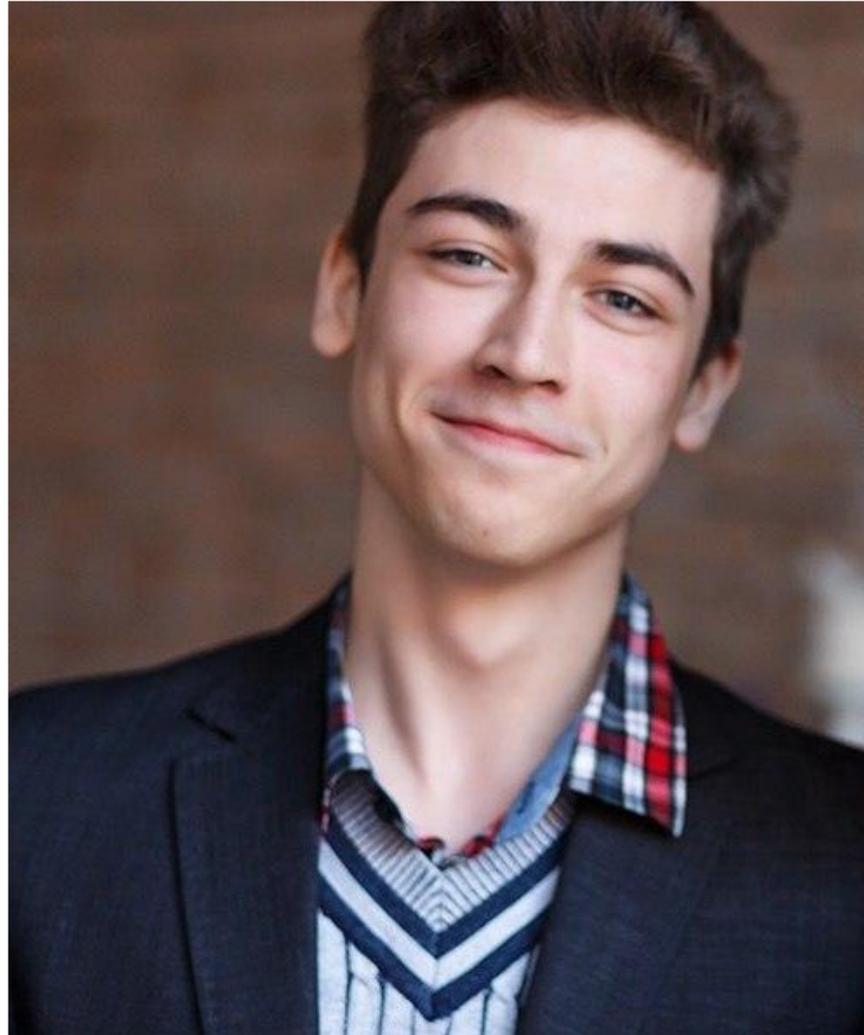
# Игорь Дыров frontend



Саркисян Артур frontend



# Анохин Данил backend

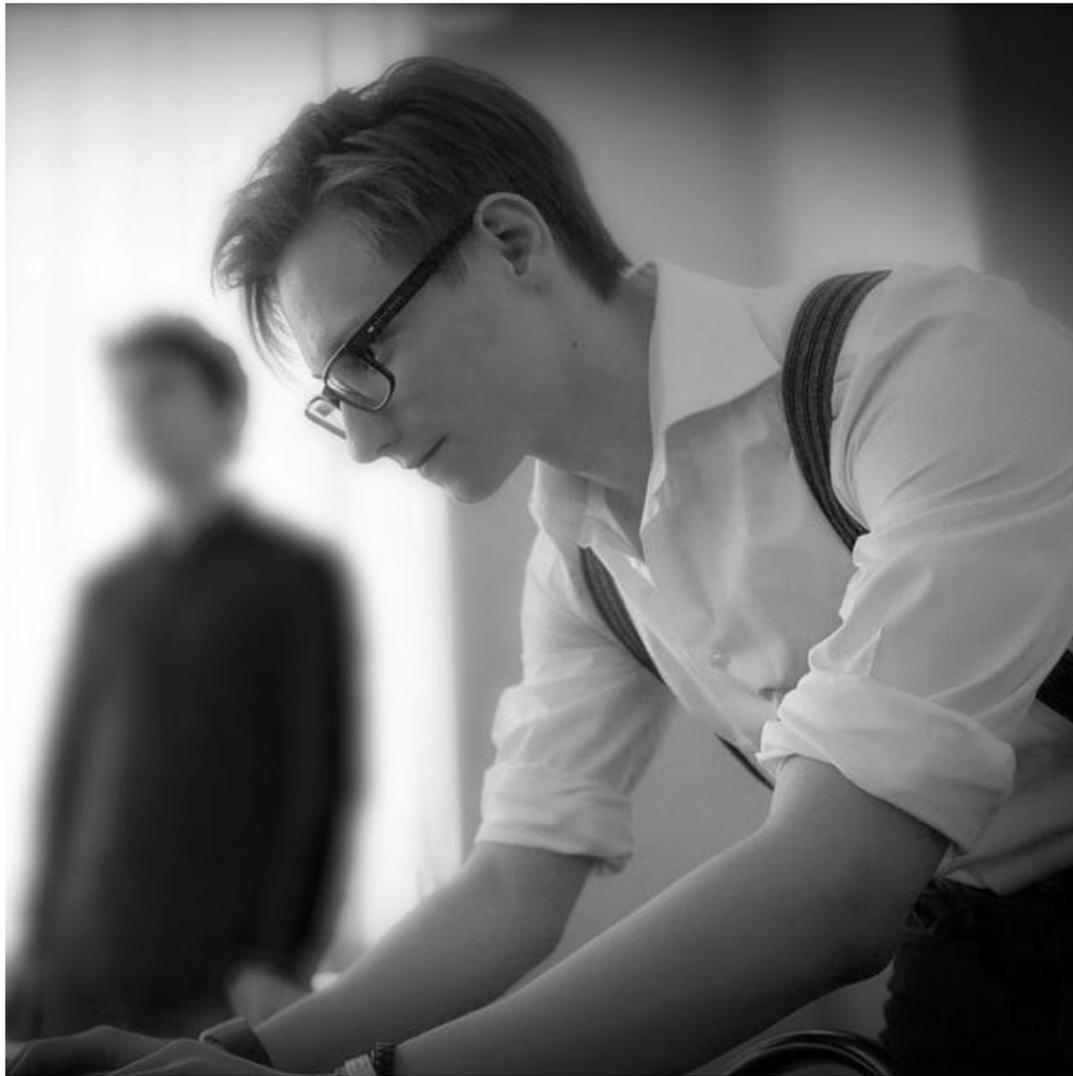


# Рязанов Максим backend



# Герескоков Владислав

## МЕНТОР





# Когда-то было и так...

Sign In

Sign Up

High Score

Profile

About

# Развитие

Tron 2D

[Sign In](#) [Sign Up](#)

Single Player

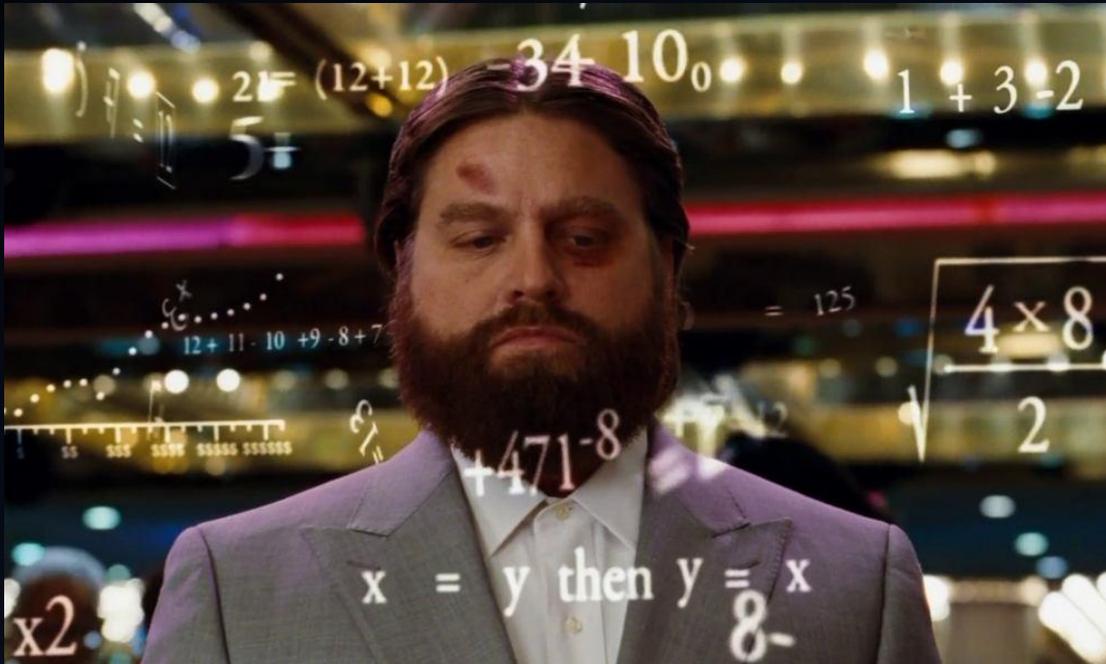
Rules

High score

# Но были и не самые удачные варианты...

In our game you will play for a motorcyclist, which leaves a bright trace. Other players or bots will also draw a line for themselves, and your task is to avoid contact with this line, regardless of whether it's yours or not. Also, in order to win, you must draw a line so that opponents can not avoid your trace. On the playing field will be spawned various bonuses that will help you win. So do not yawn!

# Матчасть



# Технические особенности



```
preRender() {
  return Transport.Get('/user').then((responseJSON) => responseJSON.json())
    .then((response) => {
      this.pageAmount = response.pagesCount;
    });
}

afterRender() {
  return new Promise((resolve) => {
    this.mainEvent = this.submit;
    this.inputs = {
      login: this.elementsArray[1],
      password: this.elementsArray[3],
    };
    this.errorLabels = {
      login: this.elementsArray[0],
      password: this.elementsArray[2],
    };
    Object.keys(this.errorLabels).forEach((field) => {
      this.errorLabels[field].hide();
    });
    this.validator = new Validator(this.inputs, this.errorLabels);
    Object.keys(this.inputs).forEach((input) => {
      this.inputs[input].render().addEventListener('blur', () => {
        this.validator.checkInput(input);
      });
    });
  });
  resolve();
});
});
```

```
mux := http.NewServeMux()

mux.HandleFunc("/", handlers.MainPage)
mux.Handle("/user", &handlers.UserHandler{db.DataBase, sessManager})
mux.Handle("/session", &handlers.SessionHandler{db.DataBase, sessManager})
mux.Handle("/user/", &handlers.UserById{db.DataBase, sessManager})

logHandler := logMiddleware(mux)
corsMW := c.Handler(logHandler)
panicMW := panicMiddleware(corsMW)
port := os.Getenv("PORT") // for heroku
if port != "" {
    zap.S().Infow("get port from env: ", port)
} else {
    port = "8080"
}

if len(os.Args) > 3 {
    port = os.Args[3]
    fmt.Println(port)
}
addr := fmt.Sprintf(":%s", port)

fmt.Println("starting server on http://127.0.0.1:8080")
http.ListenAndServe(addr, panicMW)
```

---

# Handlers

- User handler
- Session handler
- Chat handler
- Upload handler

 [chathandler.go](#)

 [mainpage.go](#)

 [sessionhandler.go](#)

 [uploadhandler.go](#)

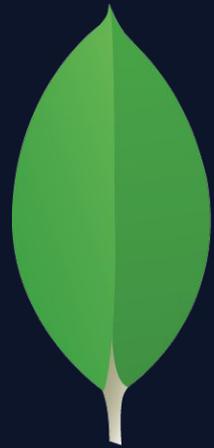
 [userhandler.go](#)

# PostgreSQL

- [database/sql](#)
- [github.com/lib/pq](#)

```
6 create table if not exists users (  
7   id bigserial not null primary key,  
8   login citext unique ,  
9   password varchar(30),  
10  email varchar(30),  
11  score bigint default 0,  
12  lang varchar(2) default 'en',  
13  avatar varchar(30) default 'defaultimg.jpeg'  
14 );  
15  
16 create table if not exists sessions (  
17   value text unique,  
18   id int references users(id) on delete cascade  
19 );
```





mongoDB®



mLab



API

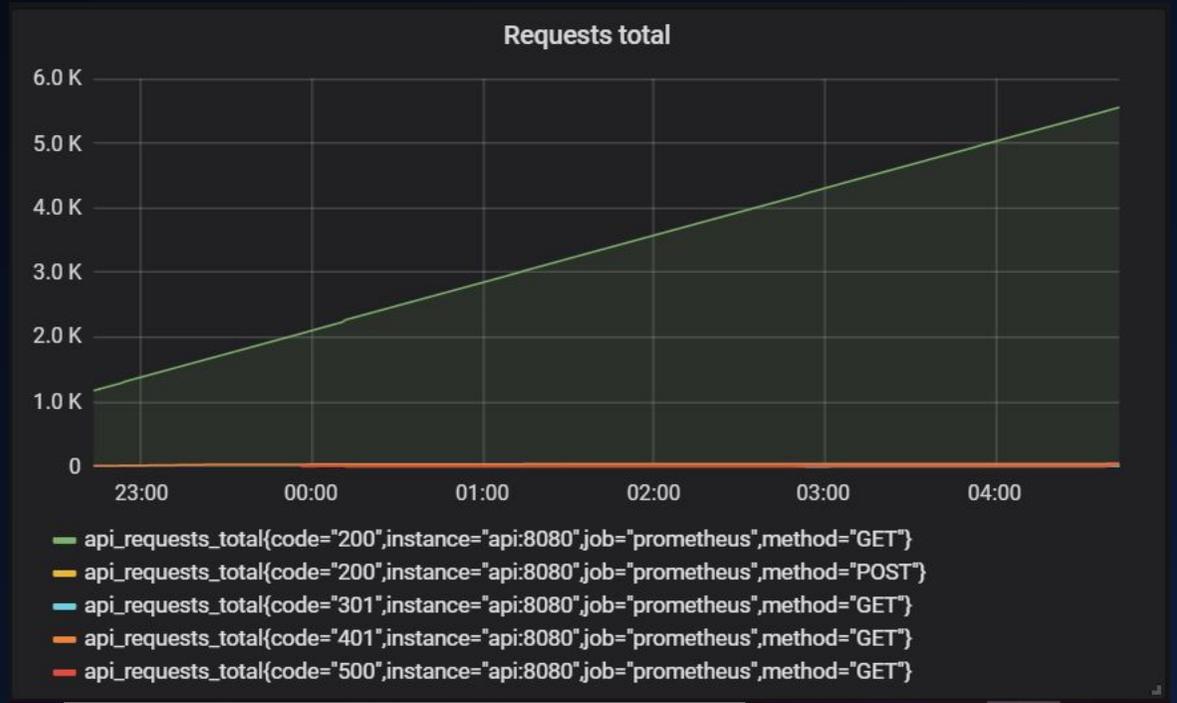


Auth

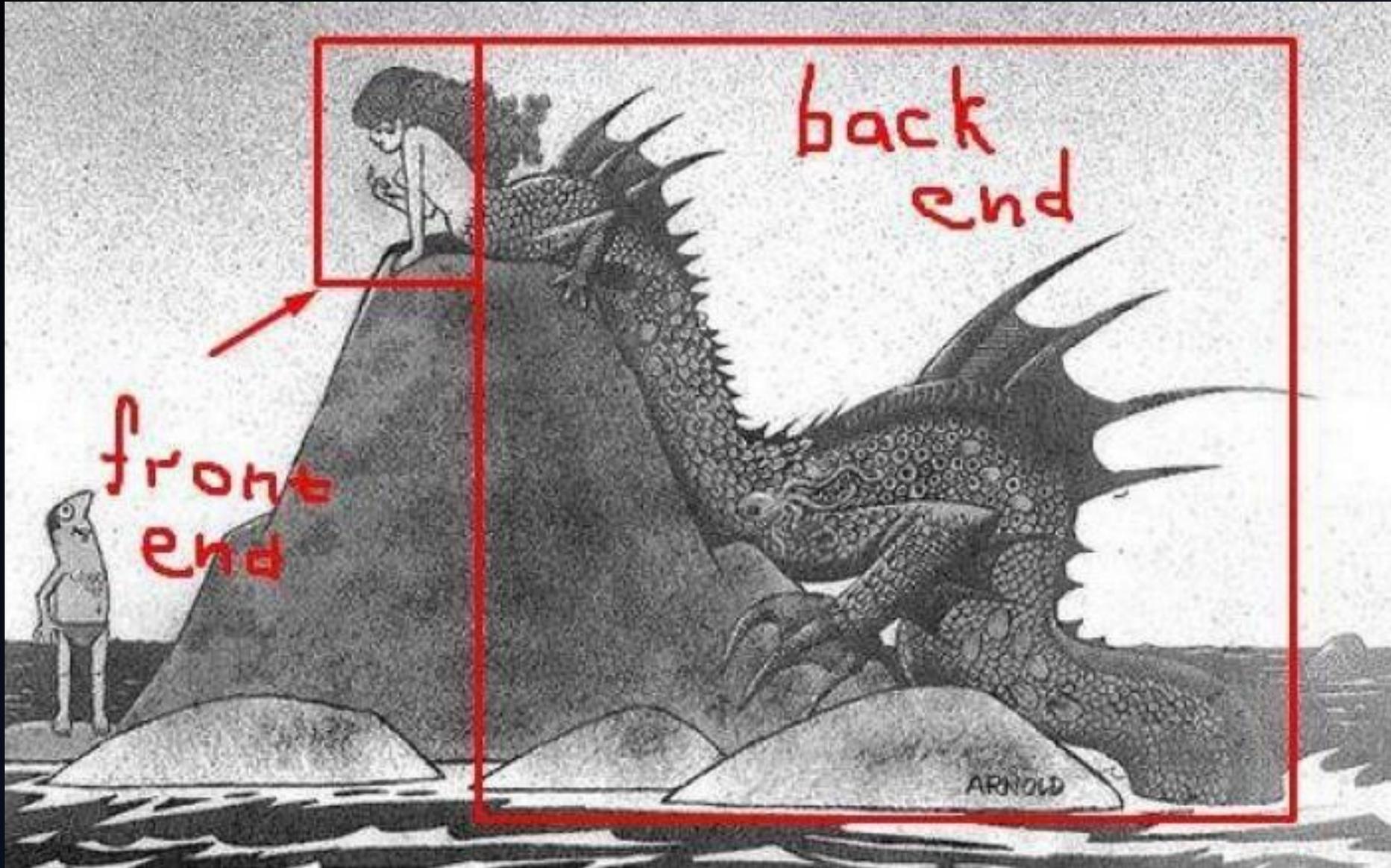


Game

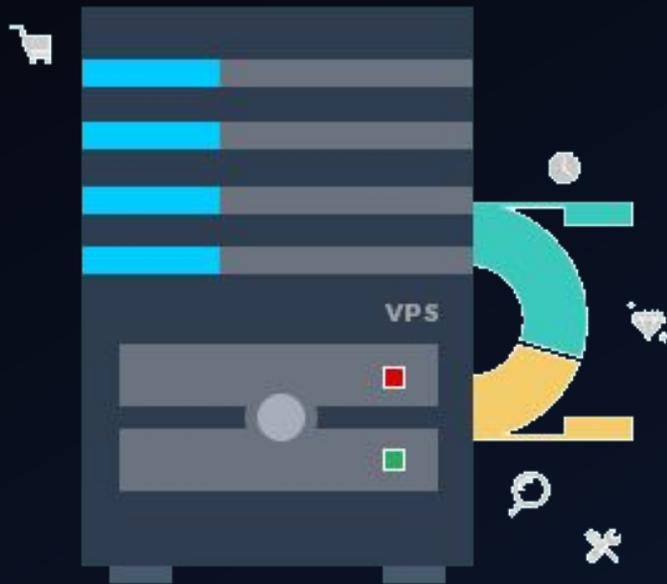




# Интеграция Frontend и Backend



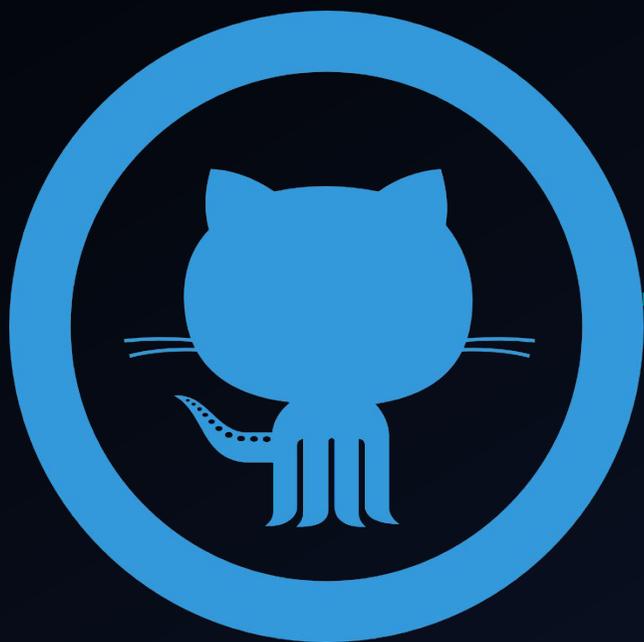
# VPS



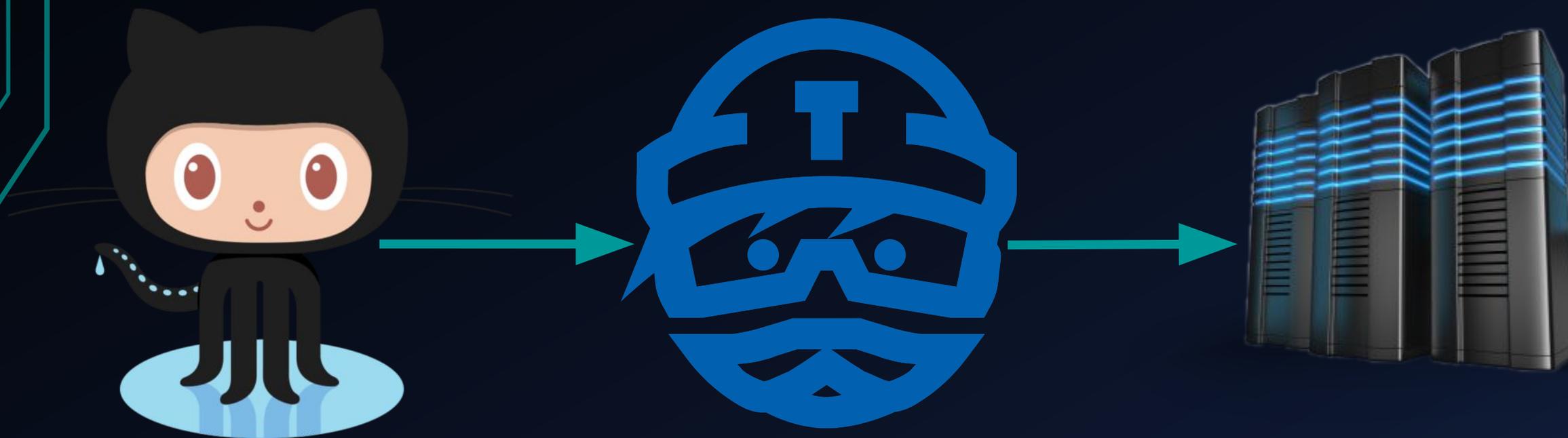
Один Nginx – хорошо, а два -  
лучше



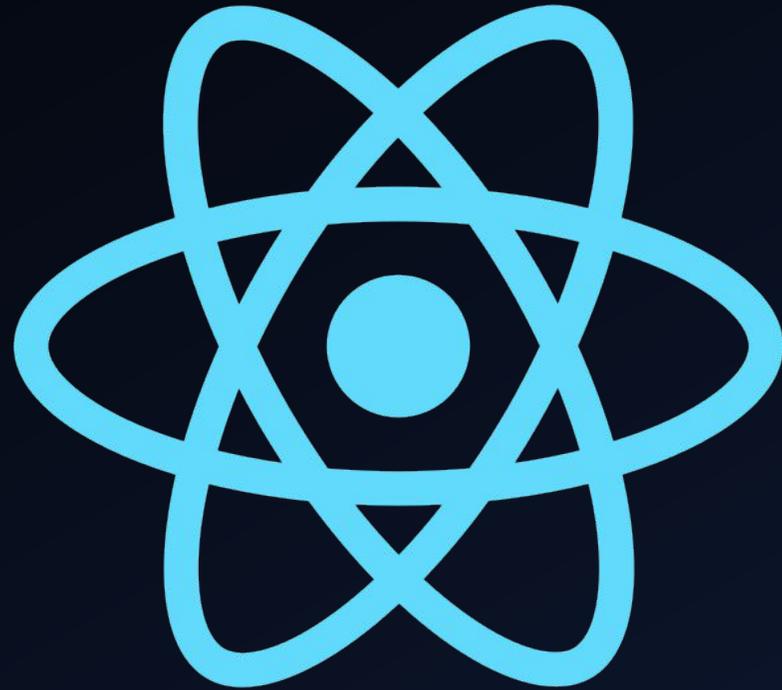
# Деплой Frontend



# Деплой Backend



# React JS



# Redux



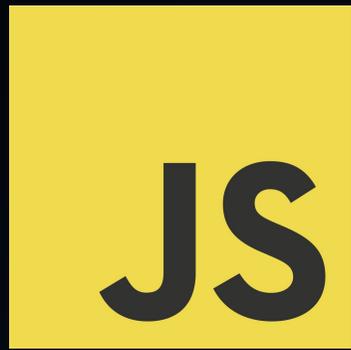
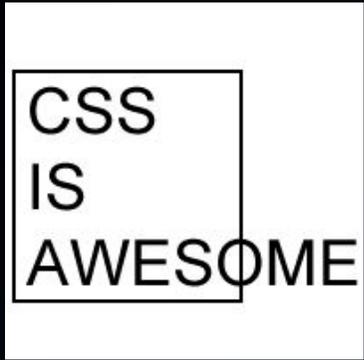
TypeScript

The logo consists of the letters 'T' and 'S' in a bold, sans-serif font. The 'T' is white with a blue outline, and the 'S' is also white with a blue outline. The two letters are connected at their base, with the 'S' overlapping the right side of the 'T'.

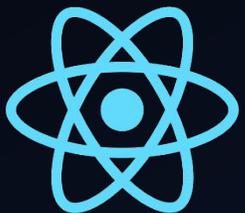
**TS**

Давайте поиграем!

<https://code loft.ru>



handlebars



Let's Encrypt



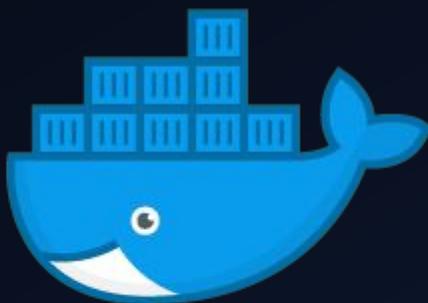
GRPC



mongoDB®



protobuf  
Protocol Buffers



docker

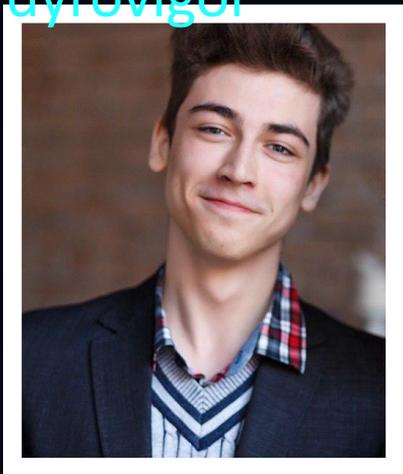




Дыров Игорь  
frontend  
@dyrovigor



Саркисян Артур  
frontend  
@arthursa



Анохин Данил  
backend  
@malefaro

# Code1 oft®



Рязанов Максим  
backend  
@RyazMax