Maciej Klos

✉ maciekklospl@gmail.com 📞 +48 695 151 380 📍 Warsaw ⌾ jotpeEe

My software development path started at university, using mostly Java, and some C++. Projects were focused on functionality or design patterns. In my spare time, I wanted to try something more modern, so I used JavaScript. I realized that doing something visual gave me the drive to pursue the front-end/web development further. While working on my first project I have read about TypeScript and decided that I am going to use it on the second one. Now I am working on my website using both TypeScript and React. I would love to continue this path with some commercial projects and enjoy the creative environment that would come with working as a team.

Skills

JavaScript | TypeScript | Java | HTML | CSS | Webpack | Git | Linux | LaTeX

Education

Polish-Japanese Academy of Information Technology

Computer Science

📅 2017 – 2020 📍 Warsaw

Projects

Phaser-farm ⧉ ⌾ (in progress)

Parcel | Webpack | Phaser 3.0 | TypeScript | HTML | CSS | Git

📅 February – May 2021

Arcade game, which simulates farm. My main goal with this project was to implement an object-oriented approach, exploring TypeScript, use the external framework as a game engine, create a development environment with Parcel instead of Webpack.

- Set up the development environment with both Parcel and Webpack.
- Working with Phaser documentation.
- Maintained version control (git) throughout the project, focused on commit messages best practice.
- Data stored across scenes using a game registry system.

Snake ⧉ ⌾

Webpack | JavaScript | HTML | CSS | Git

📅 January – February 2021

Canvas game using Vanilla JavaScript:

- Created map generator which renders levels from given strings.
- Implemented basic game state management.
- Set up the development environment using Webpack documentation.
- Maintained version control (git) throughout the project.

Character audit scripts ⌾

JavaScript

📅 2017

Google spreadsheet automated audit of every player details in the World of Warcraft guild:

- Made HTTPS request for every player, received a response as a JSON file with character details.
- Cached received information for 10 minutes, to avoid too many requests.
- Implemented automated checks based on optimal values taken from Blizzard API.
- Returned information was automatically shown in the Details section in the guild spreadsheet.