

Gymnázium, Praha 6, Arabská 14

Obor programování



MATURITNÍ PRÁCE

Filip Hruška

Webová aplikace

Duben 2024

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská<sup>14</sup> oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne \_\_\_\_\_

podpis \_\_\_\_\_

# Obsah

1. Úvod.....	2
1.1. Zadání.....	2
2. Použité technologie.....	3
2.1. Backend.....	4
2.2. Struktura aplikace.....	5
3. Graf funkce.....	5
3.1. Matematická funkce.....	5
4. Design.....	6
5. Autentifikace.....	6
6. Problematika.....	7
6.1. Canvas.....	7
6.2. Logický graf.....	7
6.3. Řešení problematiky.....	7
7. Logika hry.....	8
8. Závěr.....	8
8.1. Prostor pro zlepšení.....	8
9. Zdroje.....	9
10. Seznam obrázků.....	9

# 1. Úvod

Toto téma jsem si vybral, jelikož za dobu studia na Arabské jsem měl možnost poznat různorodé grafy, jejich průběh, zvláštnosti, postup při analýze předpisu funkce a mnohé další. Tyto znalosti mě přiměly k myšlence, jak by asi mohly grafy funkcí vypadat v programu, jak je implementovat, zobrazit a nakonec využít k něčemu zajímavému.

Moje první myšlenka byla udělat grafickou kalkulačku, jako je například velmi známý desmos, ale tuto myšlenku jsem velice rychle zavrhnul, jelikož mi její interpretace nepřišla zajímavá.

Nakonec jsem přišel s hrou, kterou jsem nikde před tím neviděl. Hra s grafy, kterou jsem vytvořil se skládá z jednoho grafu a mnoho obdélníků po stranách s překážkami uprostřed. Cílem každého z hráčů je za pomoci předpisu matematické funkce vepsané do textového pole trefit zeleně vyznačenou plochu na druhé straně. Popis, jak hra probíhá a její následnou implementaci, najdete v podkapitolách níže.

## 1.1. Zadání

Jedná se o stránku s přihlášením a vlastním profilem, na které je možné hrát s ostatními hráči zápas, který spočívá v trefování protivníka.

Hra má vzhled jednoduchého plátna, ve kterém se náhodně objevuje cílová zóna a úkolem je trefit danou zónu grafem, který po zadání funkce začne na místě hráče a pokračuje ve směru protivníka. Na hrací plochu se vygenerují překážky, aby hra nebyla příliš jednoduchá.

## 2. Použité technologie

Pro vývoj aplikace jsem zvolil vue.js na frontendu a FastApi rozšířené o FastApi websockets a databázi sqlite na backendu. Pro spojení frontendu a backendu za pomoci síťové komunikace jsem zvolil technologii axios. Za pomoci JSON web tokenů jsem vytvořil uživatelskou autentifikaci. Jako vývojové prostředí jsem zvolil visual studio code od microsoftu.

Začneme frontendem. Jak jsem již zmínil, využil jsem vue.js, což je progresivní javascriptový framework. Důvod výběru je jednoduchý. Jde o framework, se kterým jsem se již seznámil v předchozí týmové ročníkové práci, takže jsem nebyl na tomto poli nováčkem. Dalším důvodem, proč jsem si vybral právě vue, je rozmanitost možností a vychytávek, které vue nabízí. Dále má vue velmi podrobnou a přehlednou dokumentaci, se kterou se velmi jednoduše pracuje. Nejde jen o dokumentaci, ale i o rozmanitost využívání vývojáři, kteří jednoduše konzultují své problémy na různých fórech a stránkách, a to vždy usnadňuje práci.

[2]



Obrázky: vue.js [1]; jwt [3]; vs code [4]

## 2.1. Backend

A teď backend. Hnací jednotkou backendu je FastApi, což je moderní a rychlý framework pro budování API za využití pythonu. Důvod výběru této technologie je podobný, jako s vue. S danou technologií jsem se již setkal minulý rok. Tato technologie nabízí mnoho výhod oproti konkurenci: FastApi je rychlé, jednoduché na napsání, obsahuje mnoho vychytávek a poskytuje frontend rozhraní pro zkoušku jednotlivých endpointů. I přes to, že má FastApi velmi zvláštně nasanou dokumentaci, kde jednoduše vnucuje jednu verzi programu a o dalších možnostech se v ní nedozvíte, tak jsem neváhal ani minutu. [3]

Pro obsluhu klientů na frontendu a pro jejich komunikaci s jinými klienty jsem zvolil nadstavbu FastApi, a to jejich websockety. Jak jsem již zmínil, FastApi má velmi jednostrannou dokumentaci, takže vývoj, zejména této části, byl značně obtížný. Websockety jsem použil tak, že po obdržení všech potřebných dat se do listu uloží id hry, podle kterého spolu klienti komunikují. [1]

Na backendu se vyskytuje i databáze sqlite, kterou jsem zvolil kvůli jednoduché implementaci a jelikož není můj projekt orientován na práci s velkým obsahem dat, tak je sqlite dostačující. Za pomoci sqlite a FastApi jsem vytvořil autentifikaci pro uživatele s využitím JSON web tokenů. Jak tato komunikace funguje, se dozvíte v další části dokumentace.



Obrázky: sqlite [5]; FastApi [2]

Python, jakožto jazyk využívaný s prací s FastApi, jsem si musel vybrat, jelikož nebyla jiná možnost, avšak s touto možností jsem velice spokojen, protože jsem python již používal v předchozích projektech a práce s ním mi vyhovuje.

Axios je populární JavaScriptová knihovna určená pro provádění asynchronních HTTP požadavků v prohlížeči nebo na straně serveru pomocí Node.js. Tato knihovna umožňuje snadné vytváření, zasílání a zpracování HTTP požadavků a odpovědí. [9]



Obrázky: python [6], axios [7]

## 2.2. Struktura aplikace

Frontend je část webové aplikace, která je přístupná uživatelům a interaguje s nimi přímo v prohlížeči. Obsahuje veškerý obsah, který uživatel vidí a s jakým pracuje, jako jsou jednotlivé viditelné komponenty. Frontend se obvykle skládá z CSS a JavaScriptu pro interaktivitu a dynamické chování stránky. Jednoduše jde o viditelnou složku programu a také o rozhraní, kde se uživatel pohybuje.

Backend je část webové aplikace, která probíhá na serverové straně a provádí zpracování dat a interakci s databází. Tato část není viditelná pro uživatele a běží na serveru, který poskytuje data pro frontend. Asi nejznámější frameworky pro backend jsou django, ruby a rails.

Dohromady spolu tvoří ucelenou webovou aplikaci.

## 3. Graf funkce

Graf funkce je vizuální reprezentace matematického vztahu mezi vstupními a výstupními hodnotami dané funkce. Graf funkce se zobrazuje jako soubor bodů v kartézské soustavě souřadnic, kde vstupní hodnoty jsou na vodorovné ose a odpovídající výstupní hodnoty, které jsou závislé na vstupní hodnotě, jsou na svislé ose. Tyto body jsou pak spojeny křivkou nazývanou graf funkce. Graf funkce poskytuje přehled o chování funkce. Z grafu funkce dokážeme odvodit mnoho souvislostí vstupních a výstupních dat. [6]

Graf funkce je jednoduše tvořen body, které vyhovují předpisu dané funkce a jsou spojeny křivkou, což znamená, že pro jednu vstupní hodnotu můžeme dostat více hodnot výstupních. Na příklad kružnice, hyperbola, elipsa atd. Funkce, které mají pro jedno vstupní  $x$  jen jednu hodnotu, myšleno tak, že se nenachází 2 body nad sebou v kartézské soustavě, nazýváme matematickou funkcí. [6]

### 3.1. Matematická funkce

Matematická funkce je předpis, který každé vstupní hodnotě (dále jen  $x$ ) přiřazuje právě jednu výstupní hodnotu (dále jen  $y$ ). Hodnotu  $x$  nazýváme nezávislou, protože jí volíme libovolně a naproti tomu nazýváme závislou proměnnou, jelikož je vždy závislá na vstupní hodnotě  $x$ . [7]

Ve své práci využívám právě matematickou funkci, jelikož jinak by vždy nemuselo být jednoznačné vyhodnocení funkce v konkrétních klíčovém bodech. Algoritmus na vykreslování křivky by nedokázal zpracovat funkci, která nemá pro jedno  $x$  právě jedno  $y$ .

## 4. Design

Design webové stránky je klíčový pro jednoduchý pohyb klienta po stránce. Design stránky musí být vždy navržen tak, aby byl přehledný, jednoznačný a příjemný, v opačném případě dokáže klienta odradit.

Pro svou aplikaci jsem zvolil horizontální menu, ve kterém jsou zahrnuty veškeré potřebné odkazy pro pohyb na stránce. Konkrétně: *Domů, O nás, Hrát, Můj účet*. Po kliknutí na odkaz v menu se klient přesměruje na jiné url.

Při vybírání vhodné barevné kombinace jsem použil webovou stránku Colors, která poskytuje možnost výběru barev, které společně ladí a vytváří příjemnou kombinaci. Na stránce jsem si nechal generovat kombinace, dokud se mi nějaká nezalíbila a tu jsem dále doladil do kombinace, se kterou jsem byl spokojen. [5]

Nakonec jsem zvolil barvy fialové až růžové, které jsem předtím nepotkal a po implementaci byly překvapivě pohodlné. V pozadí je vždy tmavší odstín a v popředí jsou barvy výraznější. U některých komponentů aplikace používám bílý okraj, který k designu pasuje.

## 5. Autentifikace

JSON Web Token (JWT) je otevřený standard (RFC 7519), který definuje kompaktní a samostatný způsob bezpečného přenosu informací mezi klientem a serverem ve formě JSON objektu. Tyto informace lze ověřit a jsou věrohodné, protože jsou digitálně podepsány. JWT mohou být podepsány použitím klíče (s algoritmem HMAC) nebo veřejného/soukromého klíčového páru pomocí RSA nebo ECDSA. [4]

Podepsané tokeny mohou ověřit integritu informací v nich obsažených, zatímco zašifrované tokeny skrývají tyto informace před ostatními stranami. Když jsou tokeny podepsány pomocí veřejného/soukromého klíčového páru, podpis také potvrzuje, že pouze strana, která vlastní soukromý klíč, je ta, která jej podepsala. [4]

Tokeny se skládají ze tří částí: hlavička, obsah, podpis. Pokud se dají tyto tři informace k sobě, vytvoří řetězec, který je složen ze zdánlivě náhodných znaků, takže z něho nejde nic jednoznačně určit. Po poslání tokenu si ho backend zpracuje a provede decode, což znamená extrahování dat z řetězce, které lze nadále použít. [4]

S touto charakteristikou se dají tokeny jednoduše využít pro identifikaci uživatelů. Uživatelé dostanou po přihlášení token, který má předem stanovenou trvanlivost, a následně ho používají při posílání dotazů jako ověření. JWT je uložen v lokálním uložišti. Takové řešení není úplně optimální, ale stále poskytuje jistou výši ochrany. [4]



## 6. Problematika

Největším problémem, který jsem musel řešit byly odlišné souřadnice v logickém grafu. Aby byla tato problematika více smysluplná, podíváme se nejdříve na canvas a logické souřadnice mojí hry.

### 6.1. Canvas

Jde o plochu, která slouží k dynamickému vykreslování bitmap, což znamená, že je canvas vlastně pole pixelů, které se dynamicky mění dle potřeb. Na plochu lze pomocí rozmanitých metod kreslit objekty a přímky.

Způsob, jakým vykresluji křivky je prostý. Dostanu předpis funkce, jako vstup a pomocí cyklu procházím po malých hodnotách osu  $x$  a pro každé  $x$  vypočítám funkční hodnotu, kterou vykreslím jako pixel. Po průchodu všech hodnot na ose  $x$  se detailně vykreslí křivka. Průchod je velmi jednoduchý a funguje následovně: na začátku se posunu na funkční hodnotu v bodě  $x = 0$ , vypočítám další hodnotu, konkrétně v bodě  $x = 0.1$  a spojím tyto body křivkou, tento proces opakuji, dokud se nedostanu na konec canvasu.

Tento proces zní velmi slibně, ale problém nastává v okamžiku, kdy si uvědomíme, že canvasové souřadnice začínají v levém horním rohu, přičemž se hodnota  $x$  zvyšuje horizontálně zleva doprava a hodnota  $y$  se zvyšuje od shora dolů vertikálně. Z toho je patrné, že největší problém nastává při práci s funkčními hodnotami.

### 6.2. Logický graf

Logický graf je v tomto případě jednoduše kartézská soustava souřadnic, jak jí známe z matematiky. To znamená, že počátek  $P$  se nachází v poloviční výšce a šířce, kde se protínají osy  $x$  a  $y$  a mají zde hodnotu 0.

### 6.3. Řešení problematiky

Z výše uvedených informací ohledně canvasu a logického grafu je patrné, že nelze při vykreslování pracovat jen s jednou skupinou souřadnic, ale že se mezi danými souřadnicemi musí vytvořit metoda, která souřadnice převede.

Algoritmus na převedení funguje na principu jednoduchého posunu v případě souřadnic  $x$  a překlopení a posunu v případě souřadnic  $y$ . V praxi to vypadá následovně: metoda dostane jako parametr hodnoty  $x$  a  $y$ , hodnotu  $x$  nemusí řešit, jelikož daný počátek je právě v bodě 0 (canvasové)(v bodě -6 logické) a hodnotu  $y$  vypočítá následovně: nová hodnota  $y$  se rovná součtu mínusové hodnoty vstupního  $y$  a poloviny výšky canvasu.

Dalším problémem je fakt, že logický graf používá pixely, které jsou logické pro orientaci grafu a ne takové, které jsou fyzicky viditelné. To znamená, že souřadnice bodu  $[1,1]$  budou v programu reprezentovány za pomoci pixelů závislých na rozlišení obrazovky.

## 7. Logika hry

V dřívějších kapitolách jsem popisoval obecné informace, ale zde se podívám na celý průběh hry.

Ještě před hrou se hráč dostane na endpoint `/play`, kde jednoduše zadá své jméno, kterým se chce ve hře prezentovat. Dále ho stránka pošle na endpoint `/graf`, ve kterém se zobrazí samotný canvas s vykreslenými souřadnicemi a překážkami v podobě náhodně vygenerovaných kruhů v rozhraní  $\langle -3, 3 \rangle$  na obou osách a o náhodně zvoleném průměru  $\{1/2, 1\}$ . Hráč nehraje, dokud se nepřipojí jeho protihráč.

Hra probíhá tak, že má hráč na výběr mezi třemi počátky grafu, který následně zadá. Po zadání předpisu funkce se graf vyhodnotí a vykreslí. Vyhodnotí znamená, že se při průchodu hodnotami osy  $x$  postupně kontroluje, zda se graf nedotkl vrchní nebo spodní stěny  $(-4, 4)$  nebo překážek. Je také nutno zmínit, že hráč má za úkol trefit zelené políčko pomocí funkční hodnoty v bodě šířky canvasu. Poté je na řadě druhý hráč, kterému se pokus protihráče vykreslí.

Hráči nemají stejné podmínky hry v podobě překážek, jelikož po dlouhém bádání jsem došel k závěru, že mé vysněné řešení není logicky možné. Z toho tedy vyplývá, že jeden hráč má překážky zobrazené v osově souměrnosti s osou  $y$ .

Vyhrává hráč, který vícekrát trefí zelené políčko, které je po zásahu jakýmkoliv hráčem přemalováno na jiné místo podle posloupnosti náhodně vygenerované na backendu.

## 8. Závěr

S maturitní prací jsem relativně spokojen. Za dobu jejího vytváření jsem se naučil mnoha novým věcem v oboru programování, a hlavně mi tento projekt ukázal, co všechno jsem schopen vytvořit během jednoho roku. A to je za mě to nejdůležitější. Nejde jen o projekt, ale i o zkušenosti, které jsem získal během psaní tohoto kódu, jež mi budou oporou nejen v další cestě programátora, ale i v životě.

Myslím si, že zadání jsem splnil a mohu být se svým projektem spokojený. Chtěl bych poděkovat všem, kteří mě podporovali a věřili mi; konkrétně své rodině a panu profesorovi Lánovi.

### 8.1. Prostor pro zlepšení

Po dopsání mé ročníkové práce mohu říct, že prostor pro zlepšení se najde vždy. V mém případě lze uvažovat o vylepšení logiky hry a jednotlivých statistik hráčů. Prostor pro zlepšení vidím i v css, kde jsem se spokojil se slabší verzí, která není zcela optimální. Samozřejmě zde hrál svou roli i čas, který jsem značně podcenil.

## 9. Zdroje

- [1] FastApi WebSockets <https://fastapi.tiangolo.com/advanced/websockets/>
- [2] Vue <https://vuejs.org/>
- [3] FastApi <https://fastapi.tiangolo.com/>
- [4] JSON Web Tokens <https://jwt.io/introduction>
- [5] Paleta barev <https://colors.co/191308-322a26-454b66-677db7-9ca3db>
- [6] Funkce <https://www.edufix.cz/clanky/stredni-skola/matematika/funkce>
- [7] Graf-funkce <https://www.matweb.cz/graf-funkce/>
- [8] Autentifikace uživatelů <https://www.youtube.com/watch?v=5GxQ1rLTwaU&t=2392s>
- [9] Axios <https://axios-http.com/docs/intro>

## 10. Seznam obrázků

- [1] Vue.js <https://upload.wikimedia.org/wikipedia/commons/f/f1/Vue.png>
- [2] FastApi: [www.softformance.com/wp-content/uploads/2022/07/1.3-FastAPI-Logo.jpg](http://www.softformance.com/wp-content/uploads/2022/07/1.3-FastAPI-Logo.jpg)"
- [3] jwt: [cdn.worldvectorlogo.com/logos/jwt-3.svg](http://cdn.worldvectorlogo.com/logos/jwt-3.svg)
- [4] Visual studio code:  
[www.freecodecamp.org/news/content/images/size/w2000/2021/08/vscode.png](http://www.freecodecamp.org/news/content/images/size/w2000/2021/08/vscode.png)
- [5] Sqlite: [dwglogo.com/wp-content/uploads/2018/03/SQLite\\_Vector\\_logo.png](http://dwglogo.com/wp-content/uploads/2018/03/SQLite_Vector_logo.png)
- [6] Python: [/upload.wikimedia.org/wikipedia/commons/thumb/c/c3/Python-logo-notext.svg/115px-Python-logo-notext.svg.png](http://upload.wikimedia.org/wikipedia/commons/thumb/c/c3/Python-logo-notext.svg/115px-Python-logo-notext.svg.png)
- [7] Axios: [https://www.futuromediagroup.org/wp-content/uploads/2022/11/FM\\_News\\_500-2.png](https://www.futuromediagroup.org/wp-content/uploads/2022/11/FM_News_500-2.png)