

GYMNÁZIUM JANA KEPLERA

COUNTER STRIKE STATISTIKY

Maturitní práce z předmětu Informatika

Vedoucí práce: Tomáš Obdržálek

Prohlašuji, že jsem jediným autorem této maturitní práce a všechny citace, použité 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 Jana Keplera, Praha 6, Parlérova 2 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 20. března 2016

.....

Úvod

Zadání

Mým cílem, zadáním, bylo vytvořit webovou aplikaci, která by zpracovávala a zobrazovala statistiky ze hry Counter Strike: Global Offensive. Samotná stránka by měla být minimalistická a responsivní. K získání dat, normálně ve hře nedostupných, by mělo být použito oficiální Steam API.

Counter Strike: Global Offensive

Counter Strike: Global Offensive (dále CSGO) je First Person Shooter hra, v češtině prostě střílečka. Ve standardní hře jdou proti sobě dva týmy, teroristů a policistů, které se snaží porazit tým druhý, buď pomocí eliminace nepřátelského týmu, nebo odpálení bomby. Hráči mohou využít velké množství zbraní a hrají na několika různých mapách. Counter Strike má dlouhou historii „profesionální scény“, tzn. ve hře se soutěží na profesionální úrovni a v současnosti je zatím na dosavadním vrcholu popularity.

Uživatelská část

Offline instalace

Kvůli problémovosti sehnání hostingu, který podporuje node.js a zároveň je zdarma, je možnost využít lokální spuštění webové aplikace. Prvním krokem je nainstalovat Node.js s frameworkem Express a modulem request¹. Po instalaci Node.js, a následovném spuštění command řádku Node.js, je potřeba se dostat, pomocí commandu `cd`, do složky se souborem `server.js`, který je ve složce s celou aplikací (`.../csgostatistiky/`). Dalším krokem je instalace modulů `request` a `express`. To provedeme commandem `npm install request express`. Po úspěšné instalaci stačí již spustit lokální server commandem `node server.js`.

¹ <https://nodejs.org/en/download/>

Použití

Na webovou stránku je možné přistoupit zadáním adresy `http://localhost:4000/` do adresního řádku webového prohlížeče.

Na index stránce je nyní nutné, aby uživatel mohl přistoupit ke svým datům, vložit buď SteamID, nebo odkaz na Steam Profil (pokud ho nemá, je možné použít můj 76561198035581354 nebo `http://steamcommunity.com/profiles/76561198101455802/`). Po úspěšném vložení a potvrzení, stránka uživatele přesměruje na stránku s daty. V případě neúspěšného vložení, je samozřejmě uživatel informován, že se nejedná o správně zadané SteamID či odkaz na Steam profil.

Samotná stránka, je z důvodu „různonárodnosti“ komunity Counter Striku v angličtině. Proto bude popsáno, co uživatel vidí. Data jsou rozdělena do několika bloků, podle nejrůznějších kritérií ze hry.

Na obrázku² vidí uživatel jako nejzákladnější informace o svém profilu.

V druhém bloku jsou k vidění nejdůležitější obecné informace. Celková přesnost střelby hráče, celkový počet zabití atd.

Třetí, interaktivní, blok obsahuje obdobné hodnoty, jako v případě obecných, nicméně se jedná o konkretizované hodnoty pro určité zbraně. Uživatel tak může porovnat, jak dobře míří s jakými zbraněmi apod.

Čtvrtá část se věnuje mapám. Stejně jako zbraně, jsou mapy srovnány od hráčovy „nejlepší mapy“ po nejhorší. Zajímavé je, že Valve z neznámého důvodu, nezpřístupnilo data ke dvěma mapám, ve kterých se soutěží, zatímco k zbylým 6ti kompetitivním mapám data jsou.

Předposlední blok je tvořen statistikami z posledního zápasu - stav zápasu, výhra, či prohra, nejvíce hraná zbraň.

² Viz obrázek 1 v přílohách

Na konci, spíš jen tak pro zábavu, najde uživatel mix nejrůznějších statistik.

Programátorská část

Jako hlavní jazyky jsem zvolil klasické webové trio HTML, CSS a JavaScript.

Node.js

Pro serverovou část projektu jsem také zvolil framework Node.js. Důvody jsou dva. Zaprvé, neuměl jsem ani s PHP, ani s Node.js, ale tím, že je Node.js v podstatě javascriptové prostředí pro server a v javascriptu se relativně orientuji, byla to jasná volba. Zadruhé Node.js je novější, svižnější a obecně “nahajpovanější” než PHP. Při práci s Node.js jsem se nevyhnul npm, což je jakýsi soubor modulů. Konkrétně jsem pracoval s moduly express a request.

jQuery

Ani já jsem se nevyhnul dnes již všudypřítomnému jQuery. jQuery je Javascriptová knihovna sloužící k propojení HTML a Javascriptu. Na své stránce jsem jQuery využil jak pro pár animací, tak pro vyplnění stránky daty.

Steam API

Součástí mé práce byla práce se Steam API. Steam API zprostředkovává nejrůznější data Steam účtů. Steam je herní platforma, na které má hráč svůj účet se hrami, které přes něj může kupovat a pak k nim přistupovat. Zároveň poskytuje jakousi „herní sociální síť“. Steam API umožňuje přístup k vybraným datům o konkrétních účtech jako jsou zakoupené hry, odemknuté achievementy či právě statistiky o jednotlivých hráčích. Poslední jmenovanou možnost využívám ve své práci nejčastěji. Výsledná data vyhazuje Steam API v jsonu. Aby mohl programátor přistoupit k datům, potřebuje unikátní klíč přidělený pro konkrétní steam účet, doménu, ze které bude přistupovat a ID účtu, jehož data chceme získat.

Kód

Jediným úkolem uživatele je zadat správné Steam ID či odkaz na účet na Steamcommunity.

Po zadání SteamID do textového pole formuláře, javascript rozpozná, resp. upraví adresu Steam profilu, SteamID, pomocí kterého je možné komunikovat se Steam API. Pomocí metody `get`, je toto SteamID appendnuto do url.

V `datascript.js`, který běží na stránce, na kterou jsme přesměrováni, se do proměné uloží hodnota parametru Steam ID, který se nachází v url. Konkrétně jsou využity regulární výrazy. Zde je nutné poznamenat, že přesné znění výrazu pochází ze externího zdroje³.

```
$.urlParam = function(name){
    var results = new RegExp('[\?&' + name + '=(^&#)*').exec(window.location.href);
    if (results==null){
        return null;
    }
    else{
        return results[1] || 0;
    }
}
```

Po předání SteamID z `index.html` do nové stránky, `data.html`, je pomocí `.getJSON` s použitím SteamID přistoupeno na adresu na serveru, kde se nachází JSON s daty pro konkrétní účet. V momentě, kdy je na tuto adresu přistoupeno, serverový script vezme SteamID z adresy, pomocí `.req.params.steamID`, na kterou bylo přistoupeno (`steam/stats/steamID`) a pošle na ni data pro konkrétní Steam profil, SteamID.

```
app.get('/steam/stats/:steamId', function(req, res){
    var steamId = req.params.steamId;
    var url = api + '?appid=' + appid + '&key=' + key + '&steamid=' + steamId;
    request.get(url, function(error, response, body) {
        res.setHeader('Content-Type', 'application/json');
        res.send(body);
    });
});
```

Toto je provedeno dvakrát, jednou pro data ze hry Counter Strike a jednou pro data Steamového účtu.

³ <http://www.sitepoint.com/url-parameters-jquery/>

Dále, pro vyplnění dat do arrayů (zbraně, mapy) či rovnou do jquery, které appenduje do html, bylo nutné vytvořit funkci, která bude prohledávat json s daty účtu. Důvod je jednoduchý. Výsledný json nemá jednotné pořadí, protože, když například hráč nehrál na nějaké konkrétní mapě, nebyl například počet výher na dané mapě rovný. V jsonu neexistoval vůbec žádný záznam o dané mapě, čímž se mění pořadí jednotlivých objektů v závislosti na účtu.

Ke srovnání dat zbraní a map, které mám uložené ve “vícerozměrném poli”, jsem využil metody sort s funkcí sortLists napsanou tak, aby se podle relevantního sloupce dat srovnalo celé pole. Jednoduše, aby se mi zbraně srovnaly podle počtu zabití.

```
function sortLists(a, b) {  
  if (a[1] === b[1]) {  
    return 0;  
  }  
  else {  
    return (a[1] < b[1]) ? -1 : 1;  
  }  
}
```

Dále už se, po kontrole, zda-li byl json vůbec nalezen, tzn. bylo zadáno správné SteamID, vyplní stránka pomocí appendování do “statické” kostry, již zapsané v html. Zde jsou použity různé metody, podle charakteru statistik.

Design

Mým cílem bylo vytvořit pohlednou, minimalistickou a responsivní stránku. Dle mého názoru, samozřejmě subjektivního, stránka splnila tyto cíle. Pokud možno co nejefektivněji zobrazuje přehledně veškerá data. Například místo obrázků zbraní byl použit font, což činí načítání stránky mnohem jednodušší⁴ stejně jako následnou úpravu pomocí CSS.

⁴ <http://www.unknowncheats.me/forum/counterstrike-global-offensive/169881-csgo-weapon-icons-ttf-font.html>

Vzhled je vytvořen mnou a oproti původnímu návrhu jsem, právě kvůli nálezu fontu pro zbraně, opustil od carouselů pro mapy a zbraně.⁵ Jinak je na stránce využito několik jednoduchých jQuery animací, pro efektnější zobrazení dat.

Stránka je responsivní do 600px, dál to již není nutné, protože relativní hodnoty v CSS budou zůstatvat pořád stejné.⁶

Závěr

Původně jsem zamýšlel vytvořit něco obdobného pro hru Hearhstone. Vývojářská společnost Hearhstonu, Blizzard, však včas, ani do současnosti, nevydala api, které slíbila⁷. Proto jsem musel zvolit obdobný projekt. Problémem je, že Steam API existuje již delší dobu a podobná stránka, jako jsem tvořil již existuje⁸. Tím pádem, byť stránka nabízí velice obdobné funkce, je nějaká veliká popularita mého projektu nepravděpodobná.

Se samotným SteamAPI se pracovalo dobře. Json je vyveden přehledně, na internet existuje jak oficiální dokumentace od Valve, tak nepřeborné množství uživatelských rad a návodů. Jediná výtká je směrem k zveřejněným datům. Těch je sice hodně, ale jedná se o velice “povrchní hodnoty”, které neumožňují jakousi hlubší práci s nimi. Tím že jsou data triviální sice poslouží běžnému hráči Counter Striku, kterému jde primárně o například poměr zabití a smrtí, ovšem pro zkušeného či profesionálního hráče, který by se pomocí nich chtěl nějak výrazně zlepšovat (například, jak správně házet granáty – efektivita granátů házených na konkrétní místa apod.) neposlouží.

Celkově, i přes výše zmíněné “problémy”, hodnotím svůj projekt za úspěšný. Stránka zobrazuje potřebná, ve hře normálně nedostupná data, efektivně a přehledně, i navzdory charakteru zveřejněných dat od Valve. Při práci jsem si

⁵ Viz Obrázek 2 v příloze

⁶ Viz Obrázky 3 a 4 v příloze

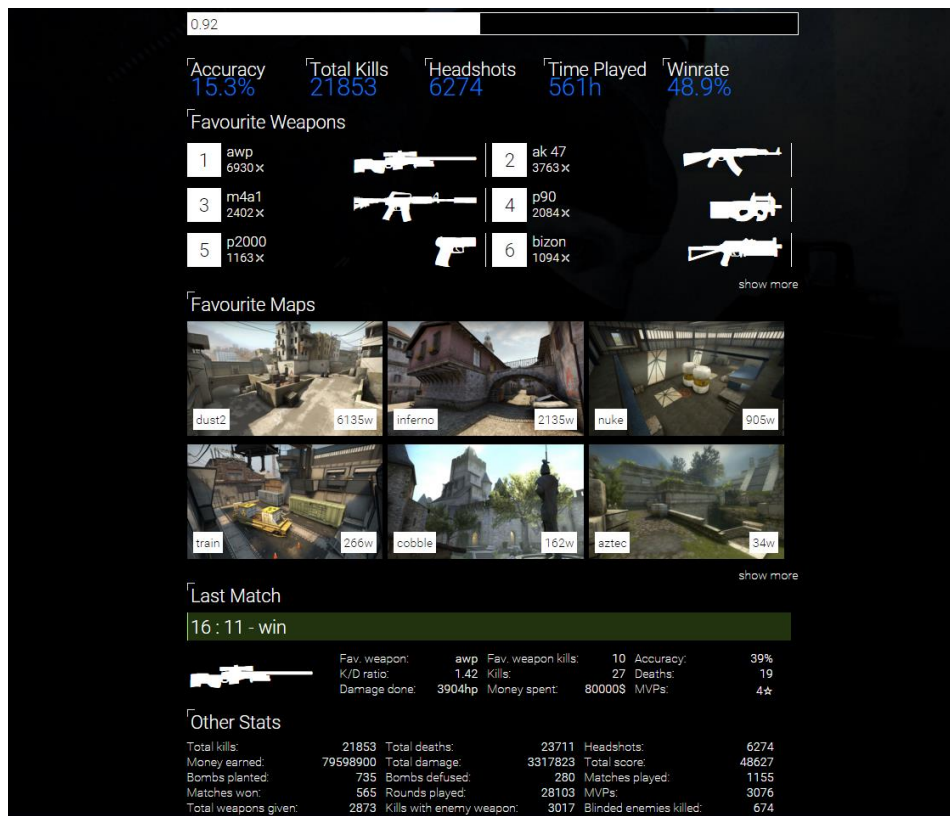
⁷ <http://us.battle.net/en/forum/topic/18719271908#2>

⁸ <http://csgo-stats.com/>

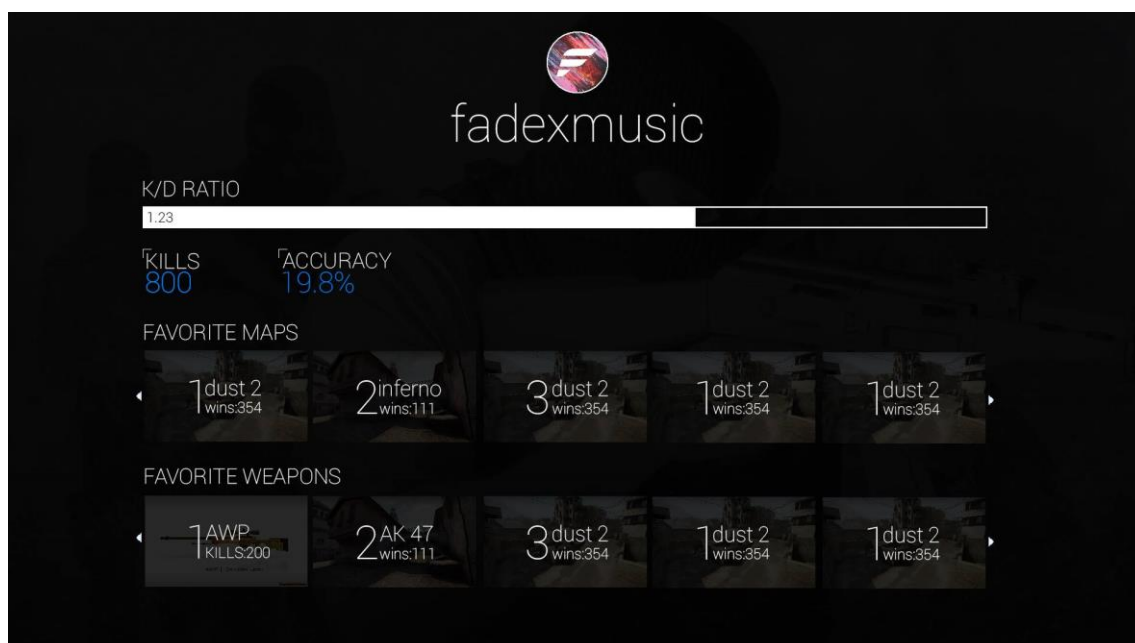
“osahal” Node.js, což kvůli jeho rostoucí popularitě považuji za pro mne přínosné, protože jsem do tohoto projektu nedělal nic, co by takto pracovalo se serverem. S celkovou funkčností a vzhledem stránky jsem nadmíru spokojený. Responsivnost funguje jak má.

Zbývá otázka zveřejnění. V této otázce ještě nejsem rozhodnutý, kvůli existence obdobné stránky ve hře Counter Strike. Nicméně jakmile bude vydáno původně plánované Hearhstone API, tak, i proto že mne práce na projektu bavila, využítuji zkušenosti z tohoto projektu a pokusím se, co nejdříve, realizovat a zveřejnit původní nápad.

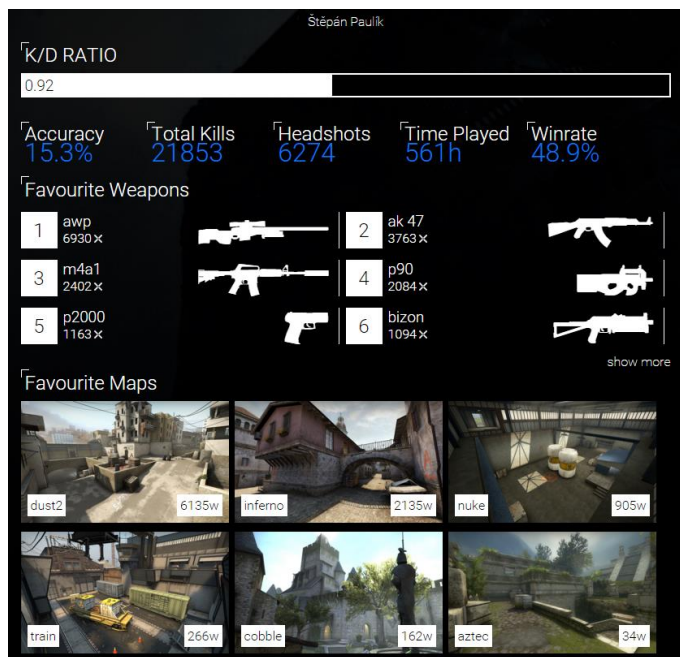
Přílohy



OBRÁZEK 1 OVERVIEW CELÉ STRÁNKY



OBRÁZEK 2 - PRVOTNÍ NÁVRH



OBRÁZEK 3 - VĚTŠÍ ŠÍŘKA



OBRÁZEK 4 - MENŠÍ ŠÍŘKA