

Obsah

1	Zadání.....	3
2	Úvod.....	4
3	Zpracování.....	5
3.1	Krabka vs. Svět.....	5
3.2	Tvorba stránky.....	7
3.2.1	Domovská stránka.....	7
3.2.2	Videa / Hry.....	7
3.2.3	Nástěnka.....	8
3.3	Videa.....	9
3.4	Vytváření kreslené grafiky.....	10
4	Použité nástroje.....	11
4.1	VS Codium.....	11
4.2	NPM.....	12
4.2.1	Ukázky příkazů.....	12
4.3	Vue.....	13
4.4	Godot.....	14
4.5	Krita.....	15
4.6	Inkscape.....	16
4.7	LMMS.....	16
4.8	Firefox.....	17
4.9	Olive.....	18
4.10	Webhosting.....	19
5	Používání stránky.....	20
5.1	Domovská stránka.....	20
5.2	Videa / Přehrávač videí.....	21
5.3	Hry / Přehrávač her.....	21
5.4	Nástěnka.....	22
6	Závěr.....	23
	Seznam použité literatury.....	24
	Seznam obrázků.....	25
	Seznam termínů.....	26

1 Zadání

Zadání, které jsem podal na první konzultaci, bylo velmi obsáhlé. Stálo na něm, že vytvořím webovou stránku, kompletně doplněnou o nějaký obsah, který zahrnuje:

- **Domovskou stránku**

Obsahuje základní informace a odkazy na jednotlivé stránky

- **Videa**

Ručně natočená a sestříhaná videa související s Geocachingem

- **Hry**

Jednoduché hry s Geocaching tematikou

- **Články**

Obsahy videí v textové podobě s obrázky a odkazy

- **Nástěnku**

Načítá a přehledně zobrazuje soubory .gpx. Součástí je i mapa.

Veškeré zmíněné body byly zpracovány, až na články, které byly nadbytečné, protože veškerý jejich obsah byl již zmíněný ve videích.

Dále jsem v zadání zmínil všechny nástroje a software, se kterým budu při tvoření pracovat. Zmíněné nástroje, jsem všechny využil a v dalších kapitolách je o nich zmíněno, jak mi pomohly a jak se na práci hodily.

2 Úvod

Cílem mé práce je představit celosvětovou hru Geocaching. Toho je docíleno webovou stránkou, na které se nachází ručně vytvořená videa. Pro zjednodušení hledání je na stránce nástěnka, která umožňuje načítat a spravovat kešky. Pro trochu zábavy je na stránce i ručně dělaná hra s Geocaching tematikou.

Hotová stránka je dostupná na:

<https://gamingas.cz/geocaching>

3 Zpracování

Vytvořit webovou stránku a zaplnit ji nějakým obsahem vůbec nebylo jednoduché. Celá byla vytvořena od základů, bez pomoci žádných redakčních systémů, např. Wordpress. Věděl jsem, že mě čeká velká spousta práce, a proto jsem začal pracovat hned v září. Díky využití verzovacího systému Git si můžu zobrazit postupně zobrazit veškeré věci, které jsem od září dělal.

6. září byla do mého repozitáře se stránkou vložena první změna. Ten den jsem pracoval na osnově a přemýšlel, co všechno na stránce bude. Skoro všechno, co jsem si zadal v osnově, jsem dodržel, ale postupem času, jsem vyškrtl části, které byly nesmyslné.

8. září jsem založil stránku a přidal ji do repozitáře. Stránku jsem vytvářel pomocí frameworků a speciálních jazyků. Vše bude dopodrobna popsáno později, jen jsem chtěl vysvětlit, že založení stránky v tomto případě neznamena vytvoření HTML souborů, ale využití nástrojů v příkazovém řádku, které vygenerují všechny konfigurační soubory a prázdnou stránku. Nic víc než vytvoření stránky jsem ten den neudělal.

V repozitáři bylo několik měsíců prázdko a nebyly přidány skoro žádné změny. Mohlo by se zdát, že jsem se na všechno vykašlal a nic jsem nedělal. To by ale nemohlo být dál od pravdy. Součástí repozitáře je jen stránka. V těchto měsících jsem pracoval na hře. Mým cílem bylo vytvořit několik malých a jednoduchých her. To se samozřejmě nepovedlo. Na hrách už jsem pracoval a vím jak jednoduché je přidávat nové věci a jak se vývoj může natáhnout. Povedlo se mi dodělat jen jednu hru, Krabka vs. Svět.

3.1 Krabka vs. Svět

Měl jsem v plánu vytvořit tři hry. Toto byla první hra, která mě napadla. Krabka vs. Svět je hra na motivy her Plants vs. Zombies a Cookie Clicker. Jsou to dvě starší, ale velice známé hry. Nápad na hru jsem měl hned, ale předtím než jsem na ní mohl začít pracovat, jsem si musel všechno pořádně naplánovat. Jelikož už mám zkušenost s vytvářením her, vím, že je moc jednoduché začít s něčím novým a pracovat na tom měsíce. Pak ale nastane zádrhel, hra není zábavná! Jak to jen vyřešit? Šlo by přidat více mechanik!

Tímto způsobem se do her přidávají zbytečné mechaniky, které je nakonec nedělají zábavnější, ale naopak složitější a hůř hratelné. Kouzlo dobré hry je v tom, aby byla jednoduchá na pochopení, ale aby její mechaniky byly natolik obsáhlé, že by trvalo dlouhé hodiny, než by je šlo ovládat perfektně. A k tomu všemu, je potřeba mít dobrý plán.

Můj plán takový, že udělám hru, ve které na kešku budou útočit nepřátelé. Cílem je kešku ubránit. Bránit jde více způsoby, což posiluje znovu hratelnost hry. Do hracího pole jdou pokládat zábrany. Je to hlavní způsob obrany kešky. Dále se jde rozhodovat, zda bude hráč nakupovat další zábrany, nebo se naopak rozhodne vylepšovat ty, co už má. Každá zábrana



Obrázek 1: Hraní hry

nebo vylepšení ale stojí body. Body se získávají klikáním na logbook na pravé straně obrazovky. Existuje ale ještě třetí kategorie vylepšení, klikače. Klikače na logbook klikají samy. Pokud si hráč nakoupí velkou spoustu klikačů, hra se hraje skoro sama, což dodává to krásný pocit postupu.

Další mechanikou ve hře je opravování. Kdyby mohl hráč celou vlnu klikat na logbook, získával by až moc bodů, proto jsem ho chtěl nějakým způsobem omezit. Vtom mě napadlo opravování. Spočívá to v tom, že každá zábrana má nějakou výdrž. Po určitém čase se porouchá a například přestane střílet. Když je porouchaná, musí se opravit. O to se stará minihra, při které se musí zašroubovat všechny uvolněné šroubky. Když se při vlně rozbije zábrana, hra začne být opravdu akční. Je zapotřebí ji rychle alespoň trochu opravit, aby mohla znovu střílet, a mezitím je i zapotřebí zajistit, aby se nepřátelé nedostali ke kešce. Opravovat deset rozbitých zábran ale opravdu není zábava a až ke konci mě napadlo přidat vylepšení šroubováku, díky kterému je opravování mnohonásobně rychlejší.

Ve hře je šest druhů nepřátel:



Každý z nich má svou speciální schopnost. Někdo by se mohl podívat nad jejich účesy a vzhledy, ale vše bylo takto uděláno úmyslně, aby je bylo jednoduché v poli rozpoznat. Dále jedno ze základních pravidel navrhování postav které znám je, že by vždycky měla jít rozpoznat silueta postavy. Toto vše dle mého názoru postavy splňují.

To je z mechanik vše. Hra mi při hraní přišla zábavná, akční a přes všech 40 vln se mi povedlo dostat za 55 minut.

3.2 Tvorba stránky

Po vytvoření hratelného prototypu hry, jsem se vrátil zpět k webové stránce. Začal jsem výběrem nějakého barevného schématu pro stránku. Logo Geocachingu je zelené, proto mi už od začátku bylo jasné, že stránka bude zelená. Mám rád tmavé stránky proto jsem jako hlavní barvu a barvu pozadí vybral tyto hezké zelené (● #36D267, ● #05130E).

3.2.1 Domovská stránka

Od počátku jsem věděl, že domovská stránka bude krátká prezentace o základech hry s odkazy na videa. Začal jsem tím, že jsem přidal text. Popisek, který je na hotové stránce vyjádřen jednou větičkou, byl zprvu celý odstavec. Toto ale nevypadalo dobře na telefonech a obecně tam bylo až moc textu.

Dále jsem se vrhl na pozadí. Je ve formě vektoru a pokrývá celou stránku. Potom jsem začal dělat druhou stránku prezentace a samozřejmě vytvářet další pozadí. V tu chvíli jsem se ale setkal s problémem. Rolování stránky bylo kvůli pozadím velice pomalé. Dumal jsem a nakonec se povedlo rolování vyřešit a teď je plynulé i na nejpomalejších počítačích.

Tyto dvě stránky byly dlouhou dobu jediné co na domovské stránce bylo, ale ke konci jsem ještě přidal dvě další s odkazy a fotogalerií. Na galerii se mi moc líbí efekt, kdy je pozadí stejné jako zobrazený obrázek, ale černobílé. Rám s obrázkem je ale osvětlen původní barvou z obrázku. Toto lze hůř vidět na tmavých fotkách, ale na světlých vypadá efekt skvěle.

3.2.2 Video / Hry

Vytvoření vzhledu stránky s videi nebyl žádný problém. Nejtěžší zde ale bylo malovat náhledy pro videa. Je jich celkem osm a každý náhled trvalo namalovat několik hodin. Když klikneme na náhled videa, přesune nás na přehrávač. Ten bylo úplně nejtěžší vytvořit.

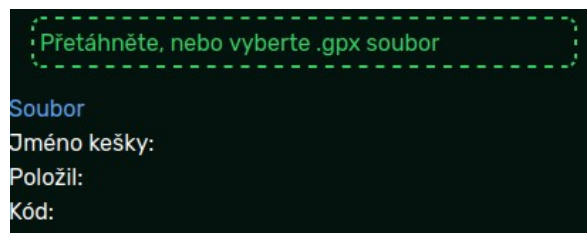
Mohl jsem na to jít jednoduchou cestou a ponechat základní HTML5 přehrávač, ale já si to udělal těžší tím, že jsem chtěl vlastní vzhled ovládání videa. Tímto jsem musel přehrávač dělat od začátku ručně. Tady bylo nejtěžší, aby se přepínání do celé obrazovky chovalo správně. Nemohl jsem použít zabudovanou funkcionalitu v prohlížeči, protože přepsala mé ovládání videa. Přidat sekce s popiskem, poznámkami a zdroji, bylo velice jednoduché.

Přehrávač her je skoro identický k přehrávači videí a nebyl problém tyto stránky vytvořit.

3.2.3 Nástěnka

Od začátku jsem věděl, že nástěnka bude načítat .gpx soubory. Tyto soubory sice jde číst v poznámkovém bloku, ale kvůli nepřehlednosti je to velice obtížné.

Díky tomu, že jsou .gpx soubory zapsané ve formátu XML, pro mě bylo velice jednoduché najít knihovnu, která mi se čtením souborů usnadní práci. Zobrazování obsahu jednoho souboru bylo jednoduché a toto (viz Obr. 2) jsem ukázal na první konzultaci.



Obrázek 2: První verze nástěnky

Zobrazit jednu jednu kešku je ale k ničemu. Proto první na čem jsem začal pracovat dál, byly dva sloupce, ve kterých se bude moct zobrazovat stovky kešek. Jelikož už jsem dokázal přečíst soubor, vytvořit tuto funkcionalitu nebylo vůbec těžké. Nejtěžší zde bylo asi přetahování souborů do prohlížeče. Do sloupců s keškami lze bez problému přetáhnout mnoho souborů naráz a hned se zobrazí. V tu chvíli jsem měl hotovou i lištu, která umožňuje přidávat keše manuálně z dialogu pro výběr souborů.

Import a export kešek bylo další na co jsem se vrhl. Export nebyl složitý. Jednoduše jsem vzal všechno uložené v prohlížeči, dal to do souboru

```
for (var x = 0; x < givenKesky.length; x++) {  
  let [stringCurrent, stringGiven]:string[][] = [], []  
  stringCurrent[x] = (currentKesky?.[x] ?? []).map(y => JSON.stringify(y))  
  stringGiven[x] = givenKesky?.[x].map(y => JSON.stringify(y))  
  
  stringKesky.push(Array.from(new Set((stringCurrent?.[x] ?? []).concat(stringGiven[x]))))  
}  
  
for (let i = 0; i < stringKesky.length; i++)  
  celkemKesky[i] = stringKesky[i].map(x => JSON.parse(x))
```

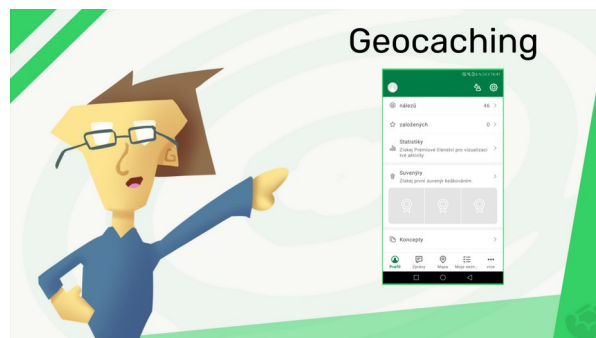
Obrázek 3: Kód pro slučování kešek

a stáhl ho. Import byl ale mnohem náročnější, a to kvůli jednomu tlačítku, ke slučování kešek. Dlouho mi trvalo, než jsem napsal kód ke slučování (viz Obr. 3), ale nakonec jsem vymyslel celkem efektivní řešení. Hlavní bylo, aby zůstaly všechny staré kešky a nebyly ve sloupcích žádné duplikáty z nových kešek. V tu chvíli jsem měl hotovou i nápovědu, nastavení nástěnky a přejmenovávání sloupců.

Nejtěžším na celé nástěnce byla mapa. Pracoval jsem zde s knihovnou OpenLayers, se kterou se ne moc příjemně pracovalo. Občas se mapa zobrazovala dvakrát, nebo se zaměřila doprostřed oceánu. Toto všechno se ale povedlo opravit. Na mapě se zobrazí i aktuální poloha a všechny body trasy, které lze i přidávat. Poslední co jsem přidal byla živá poloha, která aktualizuje vzdálenost od kešky a směr kompasu. Kvůli limitacím prohlížeče tohle ale není zcela přesné a vždy na tohle bude lepší použít nějakou aplikaci.

3.3 Video

Po tom, co byla stránka a hra hotová jsem se vrhl na videa. Už od začátku jsem věděl, že tyto videa nebudou natočená kamerou, protože nemám skoro nic, co by dokázalo udělat sledovatelný záznam. Videa jsou tedy animovaná s ručně malovanými obrázky a postavami. Nejspíše to není nic nového, ale v dnešní terminologii se toto nazývá pngtubering¹.



Obrázek 4: Snímek z videa

Pro každé video jsem prvně napsal scénář. Nejen, že je to v dnešní době moderní, ale i jsem to tak od začátku zamýšlel, dělat krátká videa, která se hned dostanou k tématu, proto nebyly scénáře moc dlouhé.

Poté bylo potřeba nahrát hlas. K tomu jsem využil program Audacity, který není ani v sekci s použitými nástroji zmíněný, ale je to skvělý program na nahrávání a střih zvuku. Po nahrání jsem odstranil všechny přechyby a dlouhé pauzy. Také jsem odstranil nepříjemné zvuky mlaskání a nadechování při mluvení.

Ted' bylo načase dát všechno dohromady. To se odehrávalo v editoru Olive. Každé video obsahuje znělku a stejné pozadí, ty byly taky vytvořeny v Olive. Zde ani nevím, co bych zmínil za obtíže. Veškerý proces vytváření videí proběhl zcela bez problémů.

Videa jsou také doplněna o různou grafiku. Může to být buďto snímek obrazovky, záznam obrazovky nebo jiná ručně kreslená grafika. Je běžné nakupovat balíčky s grafikou, ale nic takového nebylo ve videech využito.

Videi provází stejná postava, která je vidět ve hře Krabka vs. Svět. Vybral jsem tuto, protože se mi její design líbil asi nejvíc a šla jednoduše namalovat.

Veškerá videa doprovází hudba od skladatele Kevina MacLeoda, který už velkou spoustu let komponuje hudbu pro videa zadarmo a ve své sbírce má přes 2000 skladeb. Použil jsem jeho hudbu, protože má opravdu skladbu pro každou situaci a jeho webová stránka disponuje skvělými filtry pro tempo, použité nástroje a emoce, které má skladba vyvolávat.

¹ Název vychází z formátu pro obrázky .png a youtubering, tedy vytváření videí na platformu YouTube.

3.4 Vytváření kreslené grafiky

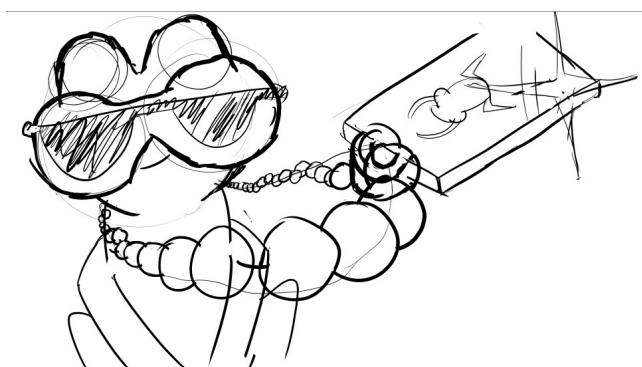
Stránka, hra a videa obsahují obrovský počet mnou ručně kreslené grafiky. Mám od nich všechny projektové soubory, a proto mohu ukázat celý postup tvorby.

Vše začíná náčrtem. Snažím se rychle na plátno namalovat přesně co mám na mysli. Vůbec nemusí vypadat hezky, hlavní je, aby na něm bylo to hlavní. U složitějších kreseb dělám ještě druhý náčrtek, který je čistší a jde na něm vše lépe vidět.

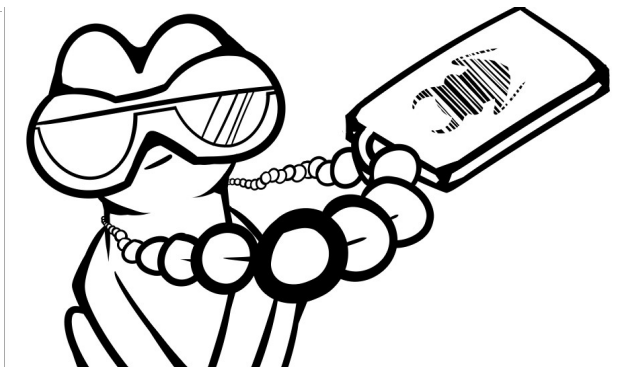
Potom na jiné vrstvě nakreslím lineart. Tady často využívám stabilizaci pera v Kritě. Zde si dávám pozor na tloušťky čar. Tloušťky čar dokážou vyjádřit vzdálenost, emoce, osvětlenost atp.

Další vrstvou je barva. Občas využiji plechovku, občas dělám výplň ručně, záleží na tom, jak se mi zrovna chce. Po vybarvení začnu dělat stíny a světla. Na stíny vždy používám studenější ztmavenou barvu a na světlé části teplejší a zesvětlenou barvu. Dále taky vezmu štětec airbrush a na hodně světlá místa jím přidám ještě další záři. Pomocí airbrushu také obarvím i celý lineart (viz Obr. 8).

Nakonec vždy namaluji pozadí. Zde bych chtěl zmínit, jak nádherně tento náhled využívá komplementární barvy (zelenou a fialovou). Toto je určitě můj nejoblíbenější náhled videa.



Obrázek 6: Náčrtek



Obrázek 5: Lineart



Obrázek 7: Vybarvení, stíny, světla a záře



Obrázek 8: Hotový náhled s pozadím

4 Použité nástroje

4.1 VS Codium

Vývojové prostředí VS Codium je odnoží nejspíše nejpoužívanějšího editoru kódu VS Code, od Microsoftu. Výhodou této odnože je zcela vypnutá telemetrie, tedy odesílání dat o používání.

Rozhodl jsem se tento editor používat díky jeho bezkonkurenční podpoře modulů a různých pluginů, které neuvěřitelně zjednodušují a urychlují psaní kódu.

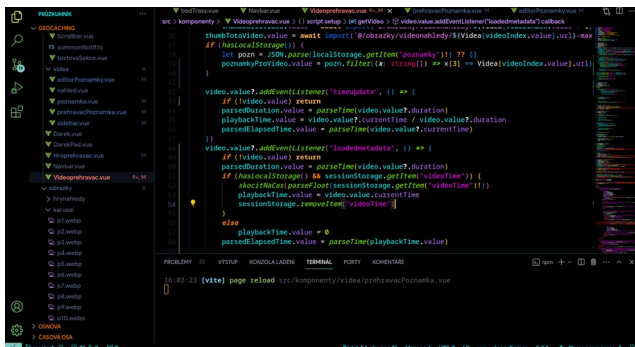
Mezi takové pluginy patří podpora pro níže zmíněné nástroje.

- **Typescript**
 - Editor mě upozorní na neshodující se datové typy.
 - Editor mě upozorní, zda mám v objektu všechny potřebné vlastnosti.
- **Tailwind** – Editor doplní jakoukoliv třídu a po uložení souboru třídy logicky přeskládá.
- **Vue** – Plugin zvýrazňuje syntaxi pro Vue.

Mezi další funkce patří

- Zabudovaný terminál a prohlížeč souborů
- Úprava více řádků najednou
- Zvýrazňování syntaxe
- Rozdělování okna
- LSP a IntelliSense pro doplňování kódu
- Emmet pro psaní kódu zkratkami
- Automatické odsazování
- Podpora pro Git

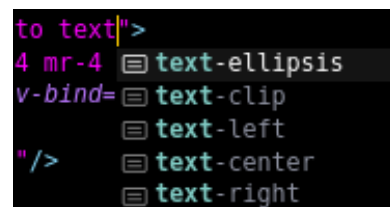
Jednou z jeho nevýhod je, že je postaven na webových technologiích (Electron). Electron aplikace jsou vlastně webové stránky s jádrem Chromu, bez ovládání a s větším přístupem k funkcím systému. Tímto vším je VS Codium pomalejší a využívá spoustu paměti RAM.



Obrázek 9: Vývojové prostředí VS Codium



Obrázek 10: Chyba po násobení textu s číslem



Obrázek 11: Doplnění Tailwindu

4.2 NPM

Je to správce balíčků pro Node.js. Stará se o to, abych jednoduše instalovat a aktualizovat všechny moduly a knihovny pro svou webovou stránku, či jakoukoliv jinou JavaScriptovou aplikaci.

```
1 {
2   "name": "geocaching",
3   "version": "0.0.0",
4   "private": true,
5   "scripts": {
6     "dev": "vite",
7     "build": "run-p type-check build-only",
8     "preview": "vite preview",
9     "build-only": "vite build",
10    "type-check": "vue-tsc --noEmit"
11  },
12  "dependencies": {
13    "fast-xml-parser": "^4.3.2",
14    "geo-coordinates-parser": "^1.6.3",
15    "kompas": "^0.0.1",
16    "node-geo-distance": "^1.2.0",
17    "ol": "^8.2.0",
18    "proj4": "^2.9.2",
19    "vue": "^3.2.47",
20    "vue-router": "^4.1.6"
21  },
22  "devDependencies": {
23    "@types/node": "^18.14.2",
24    "@vitejs/plugin-vue": "^4.0.0",
25    "vue-tsc": "^1.8.1"
26  }
27 }
```

Obrázek 12: Soubor package.json

Veškeré závislosti pro projekt jsou ukládány do konfiguračního souboru package.json. Je to vůbec první soubor, který je vytvořen při založení nového projektu s všemi těmito součástmi.

Dále se stará o rozjždění nadefinovaných skriptů. Tyto skripty mohou například spustit lokální server, abych mohl stránku zobrazit, nebo mohou stránku sestavit. Skripty si mohou nadefinovat i vlastní a může jich být i mnohem víc.

V souboru jsou také informace o projektu, například jméno, email, verze atd. Informace slouží k tomu, abych mohl projekt nahrát jako balíček, využitelný v projektech ostatních.

4.2.1 Ukázky příkazů

Spuštění lokálního serveru: npm run dev

Toto byl příkaz, který jsem spouštěl nejčastěji. K tomu, abych mohl v prohlížeči stránku zobrazit, musím spustit server Vite. Po spuštění serveru se v konzoli zobrazí adresa a port stránky. Dále se objevují všechny načtené soubory a chyby. Jednou z výhod serveru Vite je, že dokáže automaticky upravit obsah stránky, když upravím nějaký soubor. Nástroje jako např. nodemon toto dokážou také, ale ty obnovují stránku. Vite nemusí stránku obnovovat pro uplatnění změn.

Sestavení stránky: npm run build / build-only

Na rozdíl od používání základního HTML, musím stránky vytvořené za pomoci frameworku sestavit. Stránka je složená z Vue (.vue) souborů, TypeScript (.ts) souborů atp., a s těmi si prohlížeč poradit nedokáže. Sestavování lze provést dvěma příkazy zmíněnými nahoře. Používat by se měl spíše **build**, ale já většinu času používal **build-only**. Rozdíl mezi příkazy je takový, že build zkontroluje všechny soubory a stránka se nesestaví, pokud existují nějaké TypeScript chyby. Toto začne být docela otravné, protože je těžké, aby soubory neměly žádnou chybu.

Instalace: npm install _

Aktualizace: npm update

Mazání: npm remove _

Tyto příkazy se starají o správu balíčků.

4.3 Vue

Vue je webový framework k vytváření frontendu (vizuálu) stránky. Účelem frameworku je propojit JavaScript s HTML.

Řekněme, že mám ve skriptech nadefinovanou proměnnou. Abych ji mohl zobrazit na stránce, musím ve skriptech vybrat daný element a změnit jeho hodnotu. Bez toho by se změna proměnné neuplatnila. Při použití frameworku mohu nadefinovat reaktivní proměnné, které samy od sebe při změně hodnoty, uplatní změny v dokumentu. Nikdy se tedy nemůže stát, že by se změna nezobrazila.

Mezi další výhody patří rozdělení kódu do komponentů. Bez frameworku bych musel každý náhled (viz Obrázek), dělat ručně. Kdybych chtěl do náhledu něco přidat, musel bych ručně upravovat každý náhled. S frameworkem vznikne z každého náhledu jeden element.

Vue také zvládne dynamicky přepínat styly, třídy, zobrazování elementu, podle nějaké podmínky, čehož jsem mnohokrát využíval. Nebo lze podle objektu vytvářet více elementů naráz (viz Obr. 9).

Existují také další speciální elementy jako `<Transition>` a `<Component>`. Pomocí `transition` lze animovat zobrazování elementu. `Component` dokáže dynamicky měnit komponenty, což se neuvěřitelně hodilo při změně stylu zobrazení v Nástěnce.

```
<section class="flex flex-wrap gap-4 mr-4 mb-8 max-w-  
  <Nahled v-for="video in Video" v-bind="video" />  
</section>
```

Obrázek 13: Osm náhledů videí, jedním řádkem

Dále se vue díky modulu `vue-router` stará o routing webové stránky, tzn. přecházení mezi stránkami kliknutím na odkaz. Tento modul přidává dva elementy. `<RouterLink>` a `<RouterView>`. Element `RouterView` se používá hned v těle stránky a je v něm všechen obsah který se mění, tzn. celá stránka, až např. na navigační lištu a patičku stránky. Pokud chci přejít na jinou stránku, nemohu jednoduše použít element `<a>`. Používá se speciální element `<RouterLink>`. Ve vývojářských nástrojích v prohlížeči na první pohled vypadá stejně jako element `<a>`. Má ale speciální akci po kliknutí, a to tu, že změní zobrazovanou komponentu v `<RouterView>`.

Element `<a>` nemohu použít z toho důvodu, protože stránky vytvořené pomocí Vue používají model SPA (Single-page application). V podstatě to znamená, že po prvním navštívení se stránka celá načte a pro přechod mezi stránkami se nemusí stránka obnovovat a navigace je tak plynulejší a rychlejší. Elementem `<a>` by se musela celá stránka znova načítat, místo plynulého přechodu.

4.4 Godot

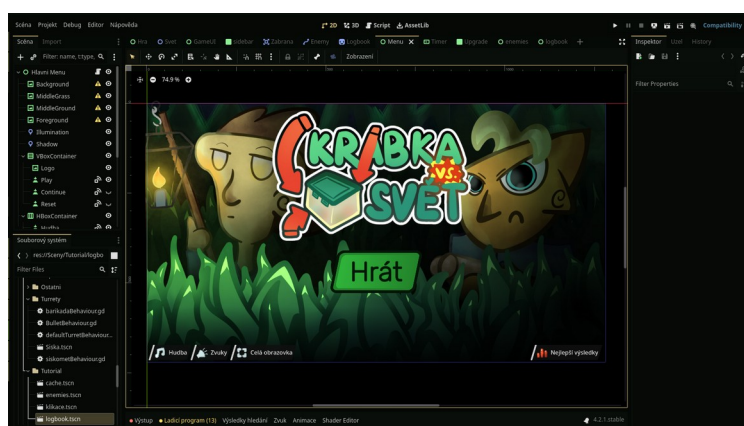
Godot je bezplatný multiplatformní herní engine s otevřeným zdrojovým kódem. Momentálně je to jeden z nejpopulárnějších 2D engineů. Využil jsem ho na výrobu hry Krabka vs. Svět.

Celý engine pracuje na bázi uzlů. Uzel může být cokoliv, např. Hráč, součást menu nebo tlačítko. Tyto uzly se pak spojí ve větší celky ve scénách. Takto vznikne přehledný strom událostí.

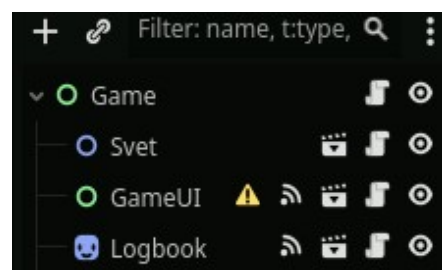
V engineu existuje spousta před chystaných uzlů pro zjednodušení pracování s fyzikou, hudbou, animacemi, UI hry a spoustou dalších vlastností. Většina uzlů sama toho sama od sebe moc nedělá, a proto je žádoucí každý uzel obohatit o skript. Skriptování probíhá v jazyce GDScript, který je podobný Pythonu. Dále lze využít C#. Předchozí verze engineu nabízely i VisualScript, kterým šlo kódovat graficky. Tento jazyk byl v nejnovějším vydání odstraněn.

```
58 func _init():
59     var file
60     # Načítání nastavení
61     if (FileAccess.file_exists("user://Nastaveni.dat")):
62         file = FileAccess.open("user://Nastaveni.dat", FileAccess.READ)
63         settings = file.get_var(true)
64         file.close()
65     else: writeSettings()
```

Obrázek 16: Ukázka jazyka GDScript



Obrázek 14: Zobrazení scény v Godotu



Obrázek 15: Uzly ve scéně

Godot umožňuje exportovat hry jako desktopové (Windows, Mac, Linux) i mobilní (Android, iOS) operační systémy. Dále lze vytvořit i webový export, který jsem právě při výrobě hry využil. Za pomoci třetích stran lze exportovat i na konzole.

S engineem se mi pracovalo dobře, ale jak se hra začala rozrůstat, vše začalo být dost nepřehledné. Kdyby se někdo jiný podíval na můj kód, vůbec by se v něm nedokázal zorientovat.

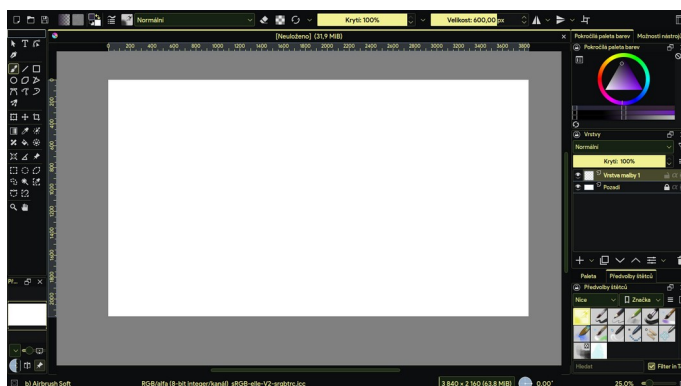
Moc se mi líbilo, že veškeré změny v editoru se hned zobrazily i v náhledu hry, což se moc hodilo i při vytváření shaderů. Dále je skvělý nespočet prvků uživatelského rozhraní, které jdou nastavit do posledního detailu, takže rozhraní vůbec nebylo těžké vytvořit.

4.5 Krita

Krita je software k vytváření rastrové grafiky. Je bezplatný a s otevřeným zdrojovým kódem a je součástí ekosystému KDE, které nabízí velkou škálu aplikací a dále i pracovní prostředí Plasma pro Linuxové distribuce a jiné Unixové systémy.

Krita má podporu pro:

- Vrstvy, masky a režimy mísení
- Animování
- Rozsáhlé nastavení štětců
- Skriptování



Obrázek 17: Nový dokument v Kritě

Krita má dále plnou podporu pro grafické tablety. Ta se pojí s rozsáhlým nastavením štětců. Lze nastavovat reakci na tlak pera, na natočení pera, na gumu pera. Tyto parametry pak mohou ovlivňovat tloušťku, průhlednost, tvar, barvu, rozptyl a neuvěřitelnou spoustu dalších vlastností.

Kritu jsem dost využíval na grafiku pro hru. Z funkcí, které jsem využil, pro mě bylo velice důležité opakující se plátno, se kterým se velice jednoduše dají dělat textury, které by se pak měly opakovat.

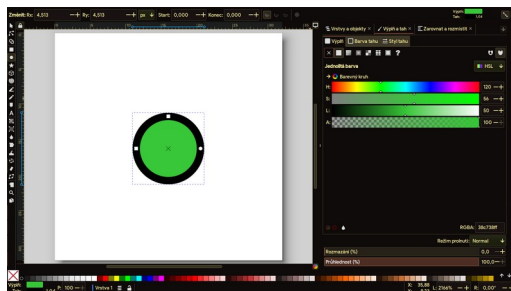
Dále jsem s ní namaloval všechny náhledy pro videa, grafiku ve videech a pár ikoněk na domovské stránce.

Z dalších úžasných funkcí bych rád zmínil masky filtrů. Masky v grafickém editoru, je vrstva, která rozšíří vybranou vrstvu další, neviditelnou a černobílou vrstvou. Touto vrstvou lze dosáhnout nedestruktivního upravování. Masky filtrů aplikuje na obrázek filtr (např. Rozmazání). Já si pak mohu vzít černý štětec a ubírat na určitých místech z filtru.

Animování jsem využíval na nepřátele ve hře. I když už je animování v Kritě delší dobu, ovládání mi stále přijde krkolomné a narazil jsem na spoustu chyb. Jinak ale obsahuje velkou spoustu funkcí, jako třeba cibulové slupky (zobrazování předchozích a následujících snímků) nebo animování masek (toto se hodí u transformačních masek, je pak jednoduché např. posouvat pozadí).

4.6 Inkscape

Inkscape je bezplatný editor vektorové grafiky. Výhodou použití vektorové grafiky, byla hlavně velikost. Všechna pozadí na domovské stránce jsou vektory, a tím jsem dosáhl toho, že obrázky mají neuvěřitelně malou velikost, pod 10kB.



Obrázek 18: Prostředí Inkscapu

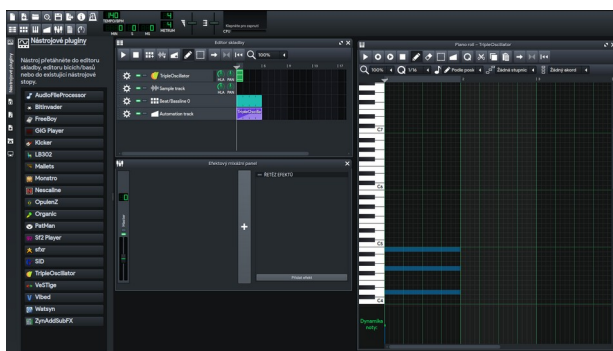
Program má podporu pro spoustu formátů, ale hlavně pracuje s formátem .svg, který je jediným vektorovým formátem podporovaným v prohlížeči.

Inkscape má nespočet nástrojů pro tvorbu křivek, čar a tvarů. Každý objekt lze transformovat; vybarvit barvou, přechodem nebo vzorkem; aplikovat velkou spoustu filtrů; využít efekty cesty, které dokážou například klovat cestu, zužovat ji, či jakkoliv jinak deformovat. Na každý objekt lze také provádět matematické operace, které objekty ořezají, spojí, rozpojí nebo zjednoduší.

S Inkscape se mi pracovalo dobře, ale nelíbí se mi nemožnost přichytávání při vytváření křivek.

4.7 LMMS

LMMS (Linux Multimedia Studio) je bezplatná digitální audio stanice pro kompozici hudby. Využil jsem program pro vytvoření hudby pro Krabka vs. Svět. Nejsem nějaký hudebník, snažil jsem se jen, aby hudba zněla jako hudba. S výsledkem jsem ale velice spokojený.



Obrázek 19: Prostředí LMMS

Pro vytváření hudby je v LMMS k dispozici spousta zabudovaných syntezátorů. Každý

instrument lze dopodrobna nastavit a přidávat nespočet různých efektů, zabudovaných do LMMS. Pomocí pluginu VeSTige, lze také využívat jakákoliv jiná VST, což je software, který generuje zvukový signál. Zvukovým signálem je myšlen např. hudební nástroj nebo efekt zvuku. [1]

Dále má LMMS podporu pro přehrávání samlů, automatizaci efektů nástrojů (např. postupné zvyšování hlasitosti nebo výšky tónu) a exportu do spousta formátů.

LMMS se mi používalo moc příjemně, ale kdybych toho o hudbě věděl víc, určitě bych mohl lépe využít všechny jeho funkce.

4.8 Firefox

Za zmínku určitě stojí i webový prohlížeč Firefox a to nejen kvůli tomu, abych vůbec byl schopný stránku zobrazit, ale i za jeho skvělé vývojářské nástroje, bez kterých bych se neobešel. Oproti Chromu mi přijdou mnohem přehlednější.

Z mých nejpoužívanějších karet bych zmínil Průzkumník. Ten umožňuje procházet dokumentem a měnit atributy a styly každého

elementu. Dál je vysoce užitečná konzole. Do konzole se запиše každá chyba ve skriptech a já ji potom mohu jednoduše opravit. Dále do konzole vložit jakoukoliv zprávu ze skriptů, což se častokrát hodí pro debugování. Rád také v konzoli zobrazuji HTML elementy. V konzoli si můžu rozbalit každou vlastnost elementu a zobrazit si jeho hodnotu. To dokonce ani v Chromu není možné. Poslední na co rád konzoli využívám, je testování různých výrazů. Na spodku konzole se hned ukáže jeho výsledek a já se tak dozvím, zda jsem ho napsal správně.

Mou další dosti používanou kartou je Úložiště. Zde vidím veškeré cookies a data v prohlížeči. Stránka nevyužívá žádná cookies, ale používá místní úložiště v prohlížeči. Oboje funkce se využívají pro ukládání dat mezi restarty prohlížeče, ale mají zásadní rozdíly.

Cookies

- Lze uložit velmi malý objem dat
- Přenáší se v hlavičce požadavku
- Lze nastavit datum vypršení
- Lze vytvářet i na serveru

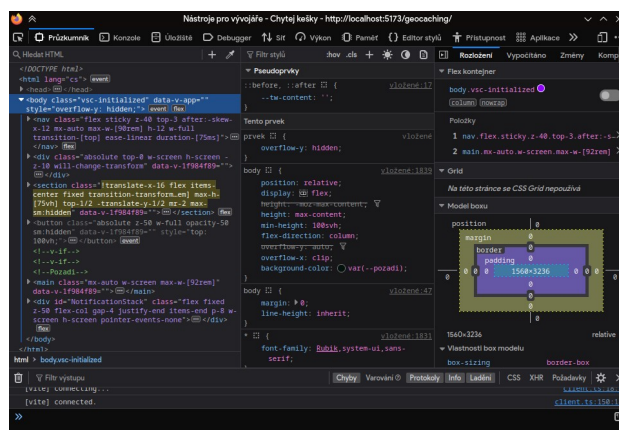
VS.

Místní úložiště

- Lze uložit až 1GB
- Zůstávají v prohlížeči
- Mohou se smazat jen manuálně
- Lze vytvářet jen u klienta

Dříve jsem používal i Debugger, který umožňuje zastavovat a sledovat proměnné skriptů, ale kvůli frameworku je mnohem méně užitečný.

Při vývoji jsem narazil i na chybu v prohlížeči. Přichytávání na domovské stránce původně mělo být uděláno pomocí kaskádových stylů, za pomoci vlastnosti **scroll-snap**. Tato vlastnost je už dlouhou dobu podporována ve všech prohlížečích. Chyba ale nastala v tom, pokud uživatel rychle roloval kolečkem. Místo přichycení se stránka zastavila někde uprostřed snímků. Proto jen proto, abych mohl podporovat Firefox, jsem celé rolování předělal pomocí skriptů. Ted' přichytávání funguje ve všech prohlížečích.



Obrázek 20: Vývojářské nástroje Firefoxu

4.9 Olive

Olive je bezplatný editor videa s otevřeným zdrojovým kódem. Vznikl na počátku roku 2019. V dnešní době je tento projekt už bohužel mrtvý.

Tento editor jsem se rozhodl využívat díky tomu, že je neuvěřitelně intuitivní na používání. Rád bych ho

porovnal s Kdenlive, dalším editorem, který už delší dobu používám. První co bych zmínil, je vytváření stop. V Kdenlive musím zdlouhavě vytvářet každou novou stopu, mezitím co v Olive se stopy vytvářejí dynamicky. Stačí přetáhnout klip do volného prostoru na časové ose, a stopa se vytvoří sama.

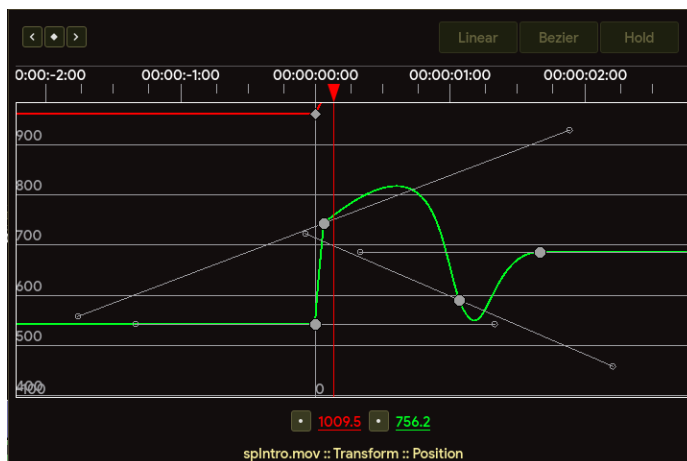
Animování mnohem jednodušší. Je velice podobné editoru Adobe Premiere. Na rozdíl od Kdenlive, jsou všechny vlastnosti oddělené, každý klíč lze interpolovat bezierovými křivkami, které lze na rozdíl od Kdenlive upravovat.

Sada efektů sice užší, ale efekty jsou mnohem robustnější a všechny lze animovat.

Editor má ale i nějaké nevýhody. Je nedodělený v alfa fázi. To znamená, že má ještě spoustu chyb. Častokrát se může stát že editor spadne. Jelikož mám zkušenost se spoustou editorů videa, vím že padají často snad všechny. Co ale nedokážu odpustit, je zasekaný náhled videa a rozbitá podpora pro proxy klipy. Editor je oproti Kdenlive také náročný na paměť RAM.



Obrázek 21: Prostředí editoru Olive

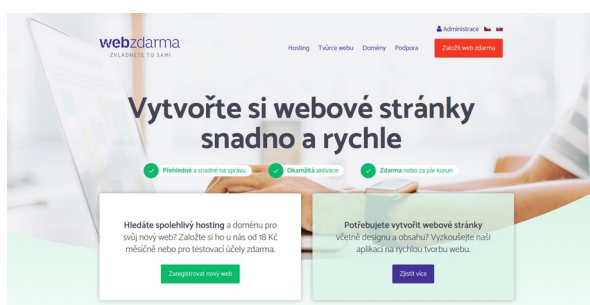


Obrázek 22: Editor bezierových křivek

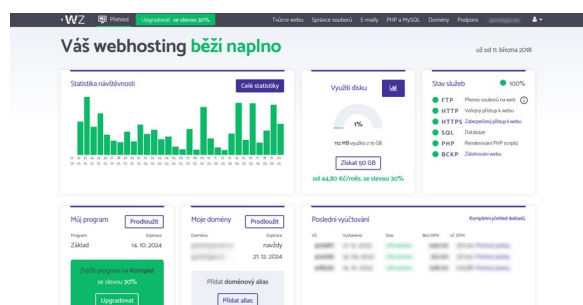
4.10 Webhosting

Na to, aby stránka byla veřejně přístupná, musí být na nějakém serveru. Existuje velká spousta společností, které poskytují hosting serverů. Já se rozhodl pro Webzdarma. Tato společnost nabízí hosting webových stránek již od roku 2001[2]. Nabízí veškeré nástroje, které jsou ke správě webu zapotřebí.

- Přístup k souborům přes FTP
- MySQL databáze
- Správa domén
- Podpora pro Wordpress
- Emailové schránky
- Podpora PHP
- Automatické zálohování
- Vlastní grafický tvůrce webů



Obrázek 23: Domovská stránka Webzdarma



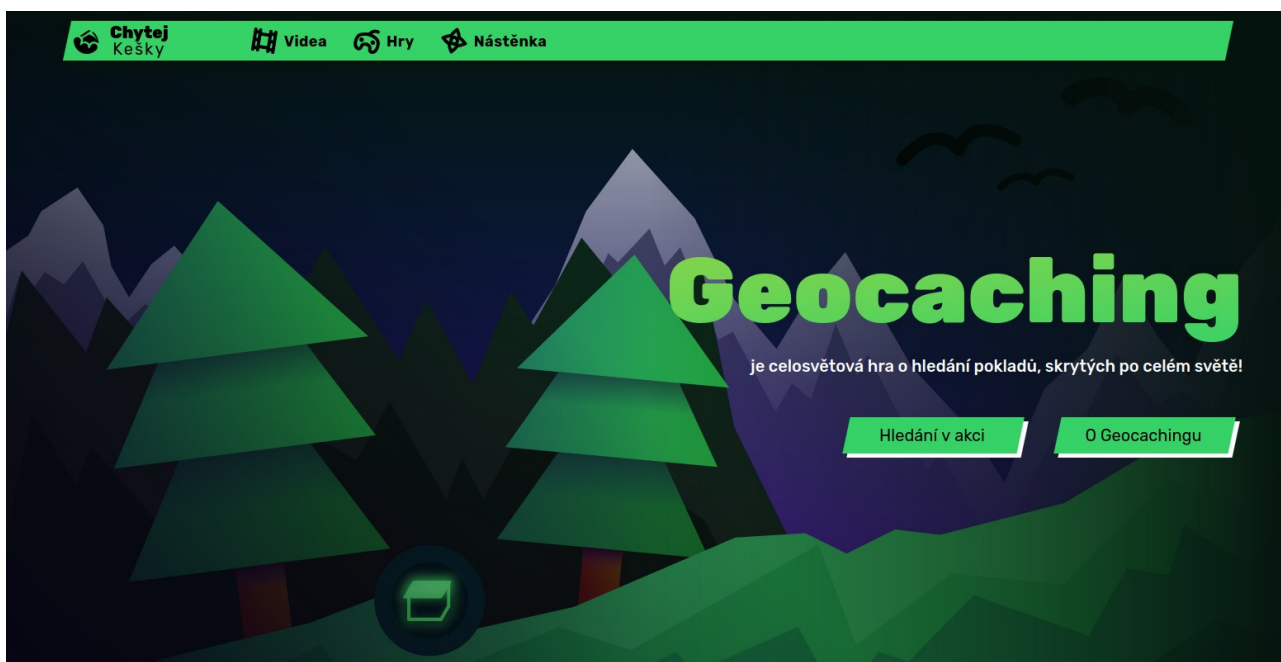
Obrázek 24: Administrace webu

Pro tohoto poskytovatele jsem se rozhodl, protože už jsem u něj spokojeně šest let. Skoro nikdy jsem se u něj nesetkal s výpadky služeb, problémy s placením či čímkoliv jiným.

Dle názvu by se dalo usoudit, že Webzdarma poskytuje jen weby zdarma. Není tomu tak! Poskytují weby zdarma, ale i dva placené programy. Já jsem plátcem základního placeného programu, který ročně vyjde na přibližně 200 Kč. Už i nejzákladnější placený program má velké výhody oproti programu Zadarmo. Významnou výhodou je podpora zabezpečeného spojení HTTPS, což je dneska nezbytná funkce. Nejen že se nezabezpečené weby zobrazí mnohem hlouběji ve výsledcích hledání, ale prohlížeč Chrome dokonce nedávno začal blokovat nezabezpečené weby. Před vstupem na takovou stránku, ukáže Chrome velkou varovnou obrazovku. Z dalších výhod je odstranění reklam, mnohem více prostoru na serveru (15GB), více emailových schránek, více místa v databázi, pokročilé statistiky, zálohování, přednostní podpora atp. Vyplatí se tedy rozhodně používat placený program.

5 Používání stránky

5.1 Domovská stránka



Domovská stránka obsahuje jednoduchou prezentaci o tom, co je Geocaching. Na každý snímek se stránka po rolování kolečkem, nebo přetáhnutím na displeji automaticky přichytí. Kvůli tomu, že webová stránka je celkem tmavá, rozhodl jsem se vytvořit i vlastní rolování na stránce, a to nejen proto, že vypadá lépe, ale i kvůli tomu, že v Chromu je rolovací lišta bílá, což vůbec nezapadá s designem stránky. Tato vylepšená rolovací lišta se nezobrazí na malých obrazovkách. Zde je nahrazena šipkou na spodku stránky, značící, že je možné rolovat. Dále obsahuje dva odkazy na jiné části stránky.

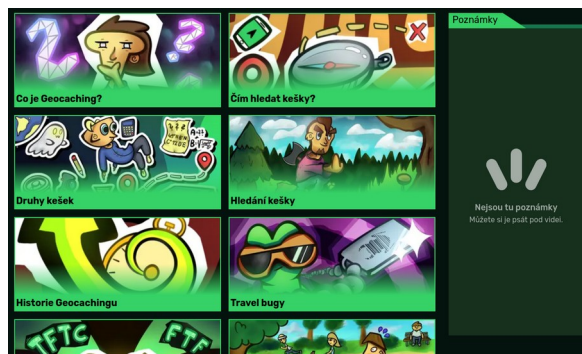
Snažil jsem se udělat domovskou stránku velice minimalistickou. Tímto šlo stránku bez větších problémů udělat responzivní. Tím, že je domovská stránka tak jednoduchá, se jí jednoduše naviguje. Ze zkušenosti vím, že čím více má stránka tlačítek a možností, tím více to běžného uživatele zmate. To, co mně se občas zdá naprosto jednoznačné, může lidem způsobovat problémy. Všechny pokročilejší funkce je lepší dát do nějaké podstránky nebo do nastavení.

Na stránce jsou dále odkazy na další zajímavé stránky související s Geocachingem.

Poslední část domovské stránky zabírá fotogalerie. Zde jsou mé fotky, které jsem vyfotil při hledání kešek.

5.2 Video / Přehrávač videí

Stránka s videi zobrazuje ručně malované náhledy pro osm videí, které se na stránce nacházejí. Vedle náhledů videí se zobrazuje i panel s poznámkami. Pro rychlejší načítání, jsou náhledy ukládány dvakrát. Jednou v nižší a jednou ve vyšší kvalitě. Pod každým videem je možné si v jakémkoliv čase vytvořit poznámku, v případě, že uživatele něco zaujalo a chce se vrátit.



Obrázek 25: Stránka s videi

Stránka s přehrávačem videí obsahuje ručně dělaný přehrávač. Video nejsou načítána z žádných externích zdrojů, jsou uložena přímo na mém serveru. Každé video je doplněné o popis a zdroj. Vedle jména videa, je tlačítko na zapsání poznámky. Toto uživateli umožňuje si na jakémkoliv čase videa uložit bod, ke kterému by se chtěli vrátit. Poznámky se zobrazí hned pod popisem, na rozcestníku s videi, ale dokonce i v samotném přehrávači jako fialový bod.

5.3 Hry / Přehrávač her

Na stránce s hrami je jedna hra doplněná o popis a náhled. Kliknutím na náhled se přesuneme do přehrávače a začne se hra stahovat. Hra je uložena na mém serveru a má 36MB, což na první pohled zní šíleně, ale díky kompresi při přenášení, se přenesou pouze okolo neuvěřitelných **8MB**.



Obrázek 26: Stránka s hrou

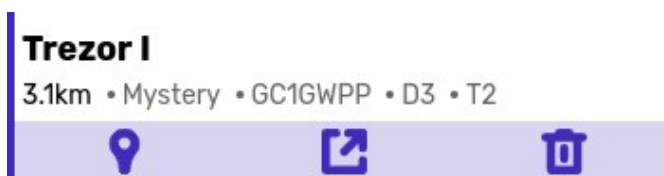
Hra je doplněná o popis ovládání a cíle. Na rozdíl od videa, neexistuje žádný HTML element, který by mi hru přehrál, proto je na stránku vložena pomocí elementu `<iframe>`, který sice omezeně, ale umožňuje na stránku vkládat jinou stránku. Omezeně znamená, že pokud pomocí hlaviček dokumentu nenastavím jinak, tak hra jde pomocí `<iframe>` vložit jen na stránku na stejné doméně.

5.4 Nástěnka

Nástěnka je nejpokročilejší a nejužitečnější součástí stránky. Každá keška se dá stáhnout ve formátu .gpx. Tyto soubory jsou ale celkem těžce čitelné a pro jejich zpracování jsou potřeba speciální aplikace. Nástěnka uživateli umožňuje přehledně zobrazit a uspořádat neomezený počet těchto .gpx souborů. Vše se děje online a bez přihlašování. Není tedy nutné stahovat žádné aplikace nebo programy.

Hlavní součástí nástěnky jsou sekce. Nad nimi je lišta s nastavením. Mám-li nějaký soubor s keškou, mohu ho na stránku přidat tlačítkem „Přidat kešky“ nebo intuitivněji, přetáhnutím souborů do prohlížeče do jedné ze sekcí. Souborů lze nahrávat neomezeně. Mohu klidně nahrát přes 50 keší najednou! Soubory se ihned načtou a zobrazí se v kartách, viz obr. 16.

Sekce jsou ve výchozím stavu pojmenovány „Nenalezené“ a „Nalezené“. Účelem karet je, aby uživatel šel ven, kešku našel, a po nalezení ji přetáhl do jiné sekce, tedy z nenalezených do nalezených.



Obrázek 27: Karta s keškou

Pokud ale uživateli nevyhovuje toto pojmenování, lze sekce bez problému přejmenovat, nebo dokonce přebarvit jejich hlavičky.

Další z funkcí je import a export. Veškeré kešky si může uživatel z nástěnky stáhnout a uložit, toto je dosti žádoucí, protože pokud se uživatel rozhodne smazat cookies z prohlížeče, smazaly by se i kešky. Dále lze perfektně tuto funkcionalitu využít na přenášení kešek mezi počítačem a mobilem. Na mobilech je přidávání kešek složitější, protože je nelze jednoduše přetáhnout. Proto to lze udělat na počítači, exportovat a importovat na mobilu. Při importování si uživatel může vybrat z dvou možností, zda chce kešky sloučit nebo přepsat. Sloučení kešek ponechá všechny kešky, které jsou nahrané. Přepsání všechny kešky smaže a ponechá jen ty ze zálohy.

Hned vedle tlačítka na přidávání kešek, je tlačítko pro výběr. Toto přepne lištu s nastavením do výběrového módu a přidá do všech karet odškrtnací tlačítko. Tato možnost umožňuje uživateli přesouvat nebo mazat více keší najednou. V režimu výběru se klikne na kešku, tím se vybere, a potom v liště je mohu přesouvat či mazat. Pokud nejsme v režimu výběru, lze se do něj také dostat tak, že držím CTRL a kliknu na kešku. Funguje to stejně jako např. v průzkumníku souborů ve Windows.

6 Závěr

Webová stránka je veřejně dostupná na internetu, je na ní vše, co jsem chtěl a jsem se svou prací spokojen. Abych to shrnul, nejtěžší na celé stránce bylo asi velké množství grafiky, které jsem musel malovat pro hru a videa. I když toho bylo dost, bavilo mě na stránce pracovat.

Dle mého názoru má přínos pro kohokoliv, kdo se něco chce naučit o Geocachingu. Na domovské stránce se dozví základy a začne ho hra zajímat víc, proto se podívá na videa. Ty obsahují velkou spoustu informací a naučí vše potřebné. Po videích možná někoho Geocaching nezaujme a zakončí brouzdaní po stránce jen vyzkoušením hry. Pokud se ale díky stránce rozhodne jít do terénu, může využít nástěnku, která ho ke kešce zavede. Dalo by se říct, že každý si na stránce něco najde, proto bych využití práce viděl pozitivně.



Obrázek 28: Logo stránky

Když se poohlédnu zpět na zadání, myslím, že už jsem od začátku věděl co chci, a přesně to jsem udělal. Samozřejmě že není všechno dokonalé, má to nějaké mouchy, ale i tak jsem udělal kopu práce a s výsledkem jsem moc spokojený.

Na závěr chci poděkovat panu Ing. Pavlu Z. za vedení této maturitní práce.

Seznam použité literatury

- [1] „Virtual Studio Technology - Wikipedia", cs.wikipedia.org. Viděno: 21. únor 2024. [Online].
Dostupné z: https://cs.wikipedia.org/wiki/Virtual_Studio_Technology
- [2] „Hosting zdarma a levné domény pro vaše stránky | Webzdarma.cz", webzdarma.cz. Viděno:
20. únor 2024. [Online]. Dostupné z: <https://www.webzdarma.cz/>

Seznam obrázků

Obrázek 1: Hraní hry.....	6
Obrázek 2: První verze nástěnky.....	8
Obrázek 3: Kód pro slučování kešek.....	8
Obrázek 4: Snímek z videa.....	9
Obrázek 5: Lineart.....	10
Obrázek 6: Náčrtek.....	10
Obrázek 7: Vybarvení, stíny, světla a záře.....	10
Obrázek 8: Hotový náhled s pozadím.....	10
Obrázek 9: Vývojové prostředí VS Codium.....	11
Obrázek 10: Chyba po násobení textu s číslem.....	11
Obrázek 11: Doplnění Tailwindu.....	11
Obrázek 12: Soubor package.json.....	12
Obrázek 13: Osm náhledů videí, jedním řádkem.....	13
Obrázek 14: Zobrazení scény v Godotu.....	14
Obrázek 15: Uzly ve scéně.....	14
Obrázek 16: Ukázka jazyka GDScript.....	14
Obrázek 17: Nový dokument v Kritě.....	15
Obrázek 18: Prostředí Inkscapu.....	16
Obrázek 19: Prostředí LMMS.....	16
Obrázek 20: Vývojářské nástroje Firefoxu.....	17
Obrázek 21: Prostředí editoru Olive.....	18
Obrázek 22: Editor bezierových křivek.....	18
Obrázek 23: Domovská stránka Webzdarma.....	19
Obrázek 24: Administrace webu.....	19
Obrázek 25: Stránka s videi.....	21
Obrázek 26: Stránka s hrou.....	21
Obrázek 27: Karta s keškou.....	22
Obrázek 28: Logo stránky.....	23

Seznam termínů

Cache/Geocache/Keška – Součást hry Geocaching. Je to krabička schovaná v terénu.

Logbook – Knížečka v kešce, do které se nálezce podepíše

Verzovací systém – Přehledně zpracovává všechny kód a změny v něm (Git)

Framework – Nástroj pro vývoj webových stránek, který propojuje dokument se skripty (VueJS)

Komponent (Vue) – Skupina HTML elementů, navenek působí jako jeden prvek

JavaScript – Skriptovací jazyk, díky kterému funguje celá stránka

TypeScript – Nadstavba pro JavaScript, která umožňuje používat datové typy

Kaskádové styly – Mění vzhled webové stránky