



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SADA WEBOVÝCH NÁSTROJŮ PRO GEOLOKALIZAČNÍ SYSTÉM

A SET OF WEB TOOLS FOR GEO-LOCATION SYSTEM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PETR BEČKA

VEDOUCÍ PRÁCE

SUPERVISOR

Doc. Ing. MARTIN ČADÍK, Ph.D.

BRNO 2016

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2015/2016

Zadání bakalářské práce

Řešitel: **Bečka Petr**

Obor: Informační technologie

Téma: **Sada webových nástrojů pro geolokalizační systém**
A Set of Web Tools for Geo-Location System

Kategorie: Web

Pokyny:

1. Proveďte rešerši existujících řešení pro vizualizaci zorného pole a pozice kamery ve webovém prostředí. Seznamte se s technologiemi, které budete potřebovat pro vývoj - např. OpenStreetMap API, Google Maps API.
2. Navrhněte a implementujte nástroje pro vizualizaci výsledků geolokalizačních systémů.
3. Navrhněte a implementujte webové rozhraní existujícího systému pro odhad pózy kamery z obrazu.
4. Navrhněte a implementujte nástroj pro manuální registraci obrazu do digitálního modelu terénu.
5. S implementovanými nástroji experimentujte a diskutujte možnosti budoucího vývoje.
6. Dosažené výsledky prezentujte formou videa, plakátu, článku, apod.

Literatura:

- <http://cadik.posvete.cz/>

Pro udělení zápočtu za první semestr je požadováno:

1. Body 1-4 zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Čadík Martin, doc. Ing., Ph.D., UPGM FIT VUT**

Datum zadání: 1. listopadu 2015

Datum odevzdání: 18. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
602 00 Brno, Bězetichova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Hlavním cílem bakalářské práce je vytvoření webové sady nástrojů, pomocí kterých lze zobrazovat výsledky a využívat funkčnosti geolokalizačního systému LOCATE. Výsledné informace lze zobrazit v podobě webové mapy a digitálního modelu terénu. Pro interakci s uživatelem je připraveno rozhraní pro nahrávání fotografií. Dále byl vyvinut způsob, pomocí kterého lze upravit vzhled a funkcionalitu zobrazovacích prostředků.

Abstract

The main goal of the thesis is to develop a set of web tools to visualize results and exploit functionality of geo-location system LOCATE. Obtained informations could be shown in web map and digital terrain model. For interaction with user is prepared interface to upload photos. Furthermore was developed a method, witch allows to customize appearance and functionality of visualization means.

Klíčová slova

web, webová aplikace, LOCATE, projekce, panorama, Leaflet, OpenStreetMap, vizualizace, digitální model terénu, mapa

Keywords

web, web application, LOCATE, projection, panorama, Leaflet, OpenStreetMap, visualization, digital terrain model, map

Citace

BEČKA, Petr. *Sada webových nástrojů pro geolokalizační systém*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Čadík Martin.

Sada webových nástrojů pro geolokalizační systém

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana docenta Martina Čadíka. Dále prohlašuji, že jsem uvedl všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Petr Bečka
16. května 2016

Poděkování

Velice bych zde chtěl poděkovat panu docentu Martinu Čadíkovi za poskytnutý čas, odborné vedení a cenné rady při řešení bakalářské práce.

© Petr Bečka, 2016.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Teoretický základ	4
2.1	System LOCATE	4
2.2	Teorie map	4
2.2.1	Geoid, referenční elipsoid a souřadnicové systémy	5
2.3	Online webové mapy	5
2.3.1	Google Maps	6
2.3.2	OpenStreetMap	7
2.3.3	Mapy.cz	7
2.4	Zorné pole	9
2.5	Panorama	9
2.5.1	Planární (rovinná) projekce	10
2.5.2	Sférická (kulovitá) projekce	10
2.5.3	Cylindrická (válcová) projekce	10
3	Použité technologie	12
3.1	HTML5	12
3.2	CSS	13
3.3	JavaScript	13
3.3.1	jQuery	14
3.3.2	AJAX	15
3.3.3	Leaflet	15
3.4	WebGL	15
3.5	PHP	16
3.6	MySQL	16
4	Návrh a implementace	18
4.1	Klientská část	19
4.1.1	Nahrávání fotografií	19
4.1.2	Webová mapa	19
4.1.3	Digitální model terénu	22
4.1.4	Nastavení prostředí	23
4.2	Serverová část	24
4.2.1	Testovací sada	25
4.2.2	Uložení fotografií	25
4.2.3	Získávání dat	26
4.2.4	Uložení dat	28

4.2.5	Zorný úhel	28
4.2.6	Odhad nadmořské výšky	29
5	Testování aplikace	30
5.1	Webová mapa	30
5.2	Digitální model terénu	30
5.3	Nahrávání fotografií	31
6	Ukázkové aplikace	32
6.1	Webová mapa s nahráváním fotografií	32
6.2	Digitální model terénu	33
6.3	Webová mapa a digitální model terénu	33
6.4	Webová mapa a digitální model terénu s nahráváním fotografií	33
7	Závěr	35
	Literatura	36
	Přílohy	38
	Seznam příloh	39
A	Obsah DVD	40

Kapitola 1

Úvod

V dnešní době si téměř každý pořizuje fotografie, ve kterých uchovává své vzpomínky z výletů a zážitků. Je to zcela nejběžnější způsob jak oživit své vzpomínky a podělit se o své životní příběhy s ostatními. Mnohokrát ale není ani možné dozvědět se o pořízené fotografii více informací. Zjistit nadmořskou výšku, zeměpisné souřadnice a nebo okolí fotografovaného místa zabere nemálo času stráveného vyhledáváním na internetu či v atlasech. Právě pro tyto případy je vyvíjen geolokalizační systém LOCATE¹, který dokáže odhadnout na základě fotografie zeměpisné souřadnice místa, kde byla pořízena, poskytnout informaci o odhadované nadmořské výšce, vyrenderovat panorama fotografovaného místa a v neposlední řadě i odhadnout velikost zorného úhlu fotoaparátu, kterým byla fotografie pořízena.

Ve své práci se budu dále věnovat vývojem webových nástrojů, které umožní uživatelům využívat funkčnosti takového systému. Pomocí těchto webových nástrojů bude moci každý, kdo má přístup k internetu, systém používat. Nástroje jsou vyvíjeny jako na sobě nezávislé části. Každý nástroj lze tedy v případě potřeby použít i samostatně, aniž bychom využili funkčnosti ostatních nástrojů.

V kapitole 2 se věnuji systému LOCATE, od kterého se práce odvíjí, existujícím webovým mapám a teoretickým pojmům, které je nezbytné znát pro pochopení práce. Kapitola 3 představuje použité technologie, jejich výhody, vývoj a funkčnost, která bude v práci použita. Kapitola 4 slouží k popisu návrhu a také implementace jednotlivých nástrojů. Je zde popsána interaktivita s uživatelem, výměna dat se serverem a prezentace výsledků systému. V kapitole 5 se zabývám testováním aplikace. Testování proběhlo nejprve na uživatelích a poté na různých platformách a různých prohlížečích. Poslední kapitola, kapitola 6 obsahuje závěr práce a zhodnocení výsledků.

¹Více informací o systému LOCATE: <http://cadik.posvete.cz/locate/>.

Kapitola 2

Teoretický základ

V této kapitole představím a vysvětlím pojmy, které jsou důležité znát pro pochopení procesu vytváření bakalářské práce. Nejprve si stručně vysvětlíme, co představuje vyvíjený systém LOCATE, který je pomocí webových nástrojů, jenž jsou smyslem této práce, vizualizován. Poté vysvětlím jaké se používají souřadné systémy a referenční elipsoidy ve webových mapách.

2.1 Systém LOCATE

Systém LOCATE je projekt, který se zabývá oblastí výzkumu výpočetní fotografie. Pomocí specifických výpočtů lze zmírnit nebo odstranit omezená data vznikající při pořizování fotografie. Například pomocí technologie HDR lze zaznamenat rozložení jasu zaznamenané fotografie a později digitálně upravit rozložení jasu.

Projekt LOCATE zpracovává a lokalizuje, fotografie na základě obrazové informace v nich obsažené. Pomocí digitálního 3D modelu DEM¹ a metod výpočetní fotografie lze odvodit pozici místa a orientaci fotoaparátu. Dále lze odvodit informace, přesněji řečeno objekty, které jsou na fotografii zaznamenány a jejich vzdálenost od objektivu. Dalším z cílů projektu LOCATE je možnost kompenzace ztráty kontrastu, změna osvětlení nebo přidání umělých stínů[4].

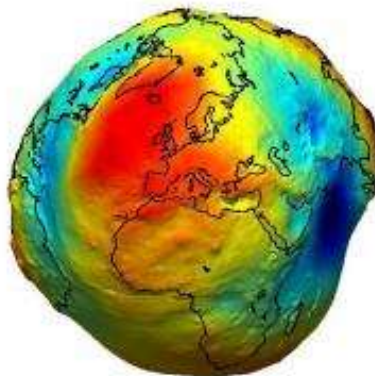
2.2 Teorie map

Základním problémem určování souřadnic je, že mapu, stejně tak i webovou mapu, nelze zobrazit jako geoid (viz obrázek 2.1). Mapy bývají typicky dvourozměrné a nastává nutnost aplikace třetího rozměru z důvodu zachování přesnosti. Zemské těleso, pro tyto účely, bývá nahrazeno referenční plochou (jelikož je na rozdíl od geoidu matematicky definovatelná), která se promítne na zobrazovací plochu, mapu. Referenční plochou se nejčastěji myslí elipsoid, který může být definován různými rozměry, nejčastěji používanými elipsoidy jsou Besselův, Krasovského a WGS84. Nakonec se provádí kartografické zobrazení na válec nebo rovinu při zkreslení úhlů, ploch nebo délek.

¹Více informací o modelu DEM: <http://www.satimagingcorp.com/services/dem/>.

2.2.1 Geoid, referenční elipsoid a souřadnicové systémy

Geoid můžeme popsat, stejně jako Zemi, jako nepravidelné, složité a členité těleso[5]. Jedná se o fyzikální model Země. V geofyzice se Geoid označuje jako ekvipotencionální plocha, která se co nejvíce přimyká klidové hladině moří a oceánů[21].



Obrázek 2.1: Ukázka tvaru geoidu[5].

Zeměpisné souřadnice, jiným názvem také koordináty, umožňují určení konkrétního bodu nebo polohy na povrchu Země. Poloha je určena pomocí dvou úhlů, zeměpisnou šířkou a zeměpisnou délkou. Je také velice důležité znát, podle kterého elipsoidu se tyto hodnoty určují, na jiných elipsoidech dostaneme různé hodnoty pro tentýž bod.

Kartografické souřadnice jsou velice podobné zeměpisným s rozdílem, že počátek systému se nenachází ve středu země nýbrž v bodě, kde zobrazovací plocha nejlépe přimyká zobrazovanému území.

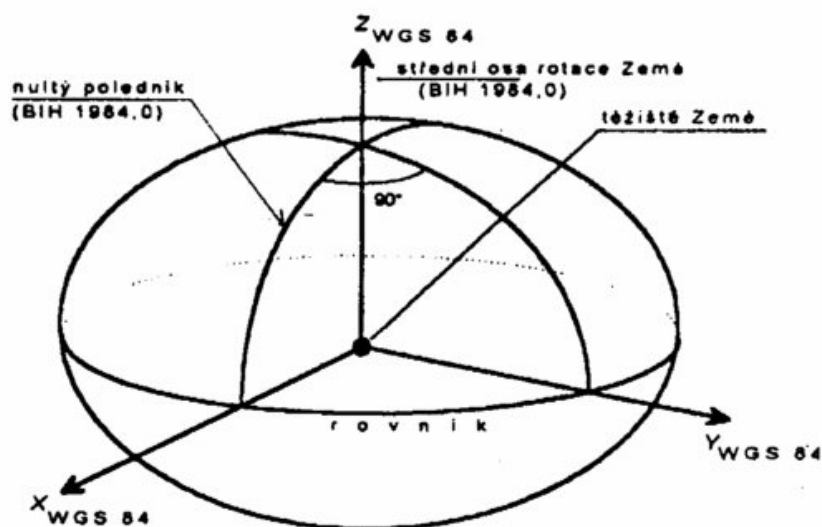
Z důvodu členitosti a nepravidelnosti zemského povrchu bylo nezbytné vytvoření referenčního elipsoidu, který je přibližně podobný tvaru Země. Nyní se můžeme setkat s různými typy elipsoidů, přičemž je obtížné určit, který je nejpřesnější.

WGS84 je označení systému zeměpisných souřadnic, který byl schválen organizací NATO a se kterými se pracuje ve většině webových map. Používají je také GPS navigace[6]. Pracuje za pomoci využití elipsoidu stejného názvu, WGS84 (viz obrázek 2.2), který je v dnešní době používán jako standard.

S42 byl systémem především pro tvorbu vojenských map. Je zde použit Krasovského elipsoid. Souřadnice na mapě mohou být změřeny pravítkem. Systém je v současné době využíván pro tvorbu turistických map.

2.3 Online webové mapy

Pro uspokojení potřeby práce s webovou mapou, jako vyhledávání míst a zobrazování dat, je zde mnoho hotových řešení mezi kterými je možné se rozhodnout. Celosvětově nejznámějšími službami jsou Google Maps a OpenStreetMap, v České republice jsou navíc hodně používány Mapy.cz společnosti Seznam.cz.



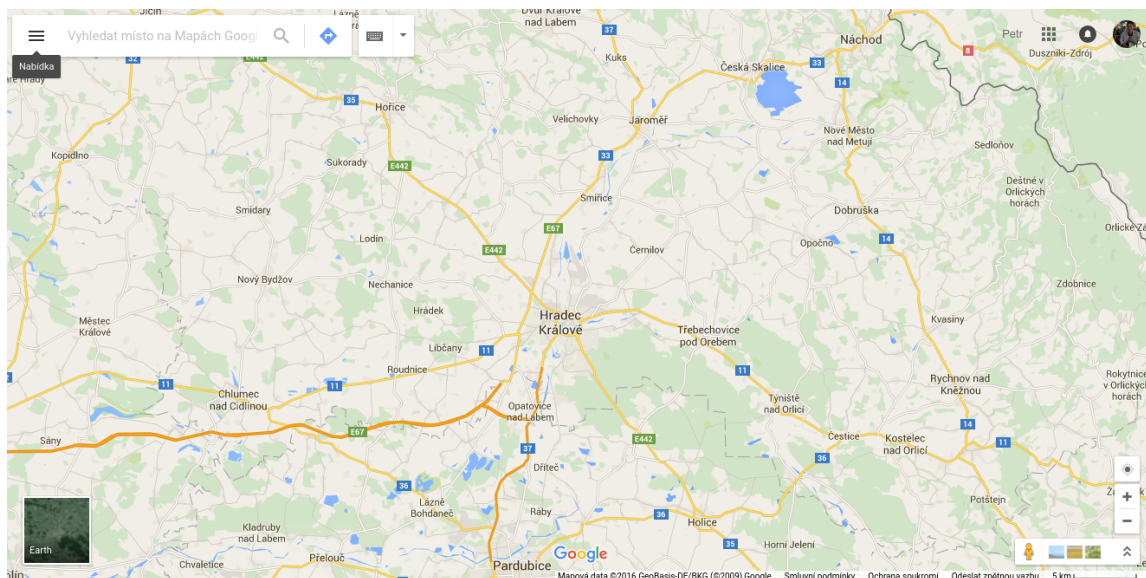
Obrázek 2.2: Schéma geocentrického souřadného systému WGS84[6].

2.3.1 Google Maps

Technologie Google Maps byla spuštěna zdarma pro nekomerční použití v únoru roku 2005[22]. Za pomoci této technologie si lze prohlížet ulice měst, plánovat trasy a měřit vzdálenosti mezi zvolenými body na mapě. V dnešní době mapy poměrně detailně pokrývají téměř celý svět. Nejvíce detailně jsou zde pochopitelně zmapovány Spojené státy americké a západní Evropa. Pokud jde o zbytek světa, uživatelé mohou požádat o úpravu místa. Na tvorbě map se aktivně mohou podílet i uživatelé nahráváním fotografií z vybraných míst. V roce 2008 byla zavedena do map čeština a to nejen v podobě české lokalizace. Služba navíc umožňuje vyhledávání a přidávání nových firem pomocí Adresáře firem, vyhledávání tras za pomoci českých výrazů a stažení mobilní aplikace v českém jazyce[10].

Základní vrstva služby Google Maps je vektorová, jedná se o tzv. satelitní zobrazení. Některá města mají navíc přidanou možnost prohlížení v náklonu 45° a nebo ve formě Street View. Street View je možno využít ve větších městech, ve kterých se uživatel může z pohledu první osoby „procházet“ ulicemi měst. Snímky pro tuto službu jsou pořízeny pomocí specifických automobilů, které při průjezdu zaznamenávají okolí. Další možností je zobrazení terénní mapy, která je tvořena pomocí vrstevnic, naopak zde zcela chybí turistická mapa. Možnost zobrazení veřejné dopravy nebo silničního provozu je dostupná pomocí doplňkových vrstev. Podklady jsou v některých případech, narozdíl od lokálních mapových služeb, méně přesné a to z důvodu, že společnost Google poskytuje mapy globálně a špatně se zde definují specifikace pro jednotlivé země[9].

Dále byla vydána API dovolující použití map v nekomerčních projektech třetích stran. Je založena na jazyku JavaScript pro pohodlnou práci s DOM strukturou a jednotlivými elementy jazyk HTML. API uživatelům poskytuje několik základních vrstev (silniční, satelitní a hybridní), dále umožňuje na základě GPS souřadnic do mapy vkládat body, pohyblivé útvary, geometrické tvary a nebo vyskakovací okna.



Obrázek 2.3: Ukázka webové mapy Google Maps.

2.3.2 OpenStreetMap

„OpenStreetMap tvoří komunita uživatelů, kteří přidávají a udržují data o silnicích, cestách, kavárnách, železničních stanicích a mnohém dalším po celém světě.“ [15] Tento projekt vytvořil v roce 2004 Steve Coast z Velké Británie. Jedná se o neziskovou organizaci, založenou na koncepci otevřeného softwaru, umožňující tvorbu geografických dat a jejich vizualizaci v podobě topologických map. Inspirací se stal projekt Wikipedie. Editace map probíhá za účasti dobrovolníků, kteří využívají převážně záznamů z GPS. Změny se ve veřejné mapě projevují ihned, technologie zachovává také historii provedených změn na mapě.

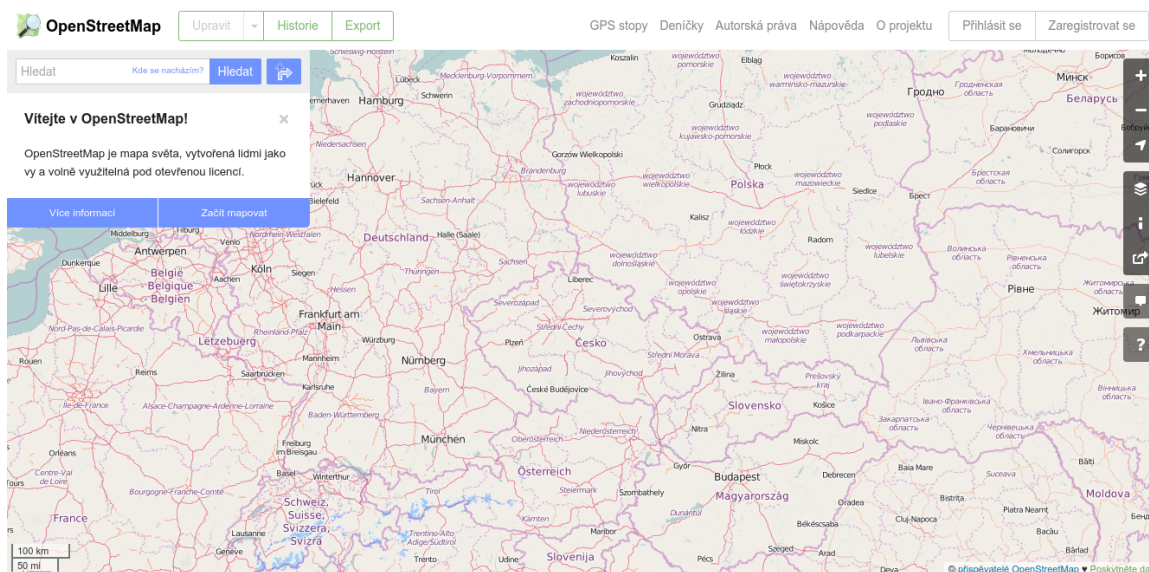
Projekt se řídí koncepcí otevřených dat, což znamená, že při uvedení autorství organizace OpenStreetMap a jeho přispěvatelů lze využívat veškerá dostupná data, která technologie nabízí. Tento koncept je reakcí na to, že naprostá většina webových map je pro použití technicky a právně omezena. Technologie tedy uživateli umožňuje svobodně a volně nakládat s geografickými daty, začleňovat je do projektů a využívat nejaktuálnější data.

Snahou je pokrytí kompletně celého světa, nicméně v současné době je nejdetaileji zpracována západní Evropa a Spojené státy americké. Oblast České republiky je zde zpracována poměrně kvalitně. Základních poskytovaných vrstev není mnoho a chybí zde například vrstva letecká a satelitní. Mezi dostupnými vrstvami jsou Základní, Cyklomapa, Dopravní a nebo Humanitární. Systém je kompletně tvořen vektorovými prvky a ukládání je realizováno ve formátu XML.

Pomocí dostupné API můžeme vkládat mapu na stránku a zobrazovat do ní fotografie, body, trasy či pohyblivé objekty. Pomocí rozhraní, napsaného v jazyce JavaScript, je možné vytvářet interaktivní mapy, kdy si uživatel sám zvýrazňuje obsah a přidává prvky do mapy.

2.3.3 Mapy.cz

Společnost Seznam.cz, specializující se především na český trh, poskytuje uživatelům aplikaci Mapy.cz. Tato aplikace detailně pokrývá Českou republiku, ale nechybí zde ani pokrytí



Obrázek 2.4: Ukázka webové mapy OpenStreetMap

zbytku evropského kontinentu. Mapy jsou vhodné na plánování tras a cest, v rámci České republiky jsou zde navíc detailní vrstvy vytvořené pro turistiku a sport. Vzhledem k ne zcela důkladně zmapovanému území mimo Českou republiku zde chybí vrstvy pro turistiku pro zbytek Evropy.

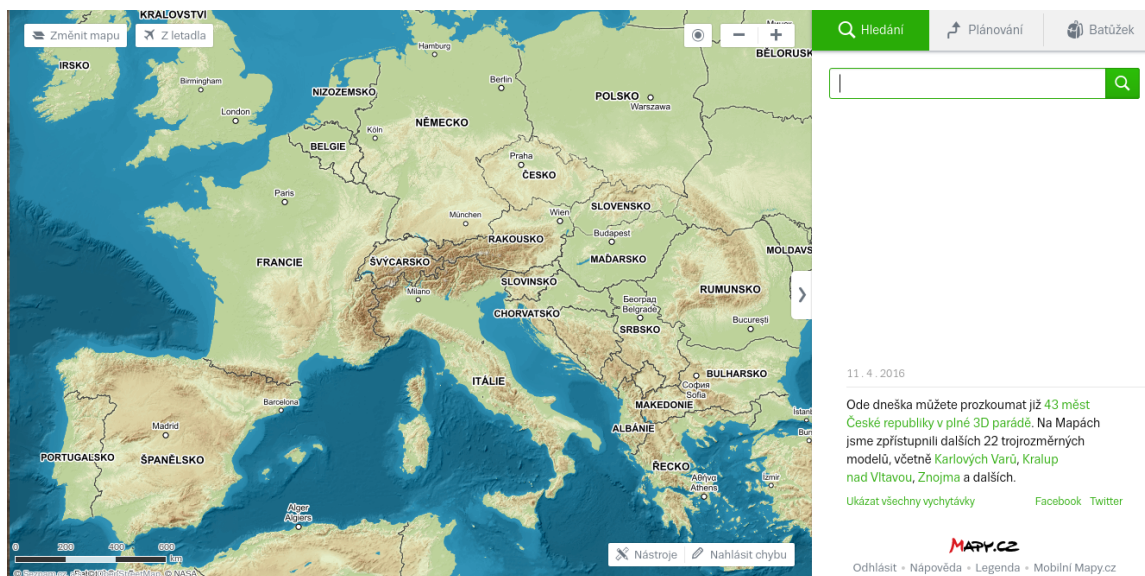
Společnost nabízí prohlížení mapy hned v několika základních vrstvách. Nejběžněji používanými jsou Turistická, Dopravní, Zeměpisná a Letecká. Letecké snímky pro území České republiky jsou přitom dostupné v mnohem lepším rozlišení než pro okolní zbytek Evropy. Dále má uživatel možnost zvolit si způsob prohlížení, kromě základního pohledu technologie nabízí navíc letecký a panoramatický pohled, navíc zde přibyla novinka v podobě 3D pohledu.

Výhoda aplikace Mapy.cz spočívá ve velmi detailním zpracování území České republiky. Při prohlížení mapy aplikace sama navrhuje zajímavá místa pro navštívení a přikládá např. otevírací doby míst, ceníky prohlídek a kontaktní informace. Detailně jsou zmapována města i obce na území celé republiky, kdy jsou na mapě vyznačeny názvy ulic, čísla domů a významná místa pro sportovní a kulturní vyžití, restaurace, nemocnice a v neposlední řadě i informační centra. Ve zbytku Evropy je takto detailně zpracováno několik desítek významnějších měst.

Jedna z nejvýznamnějších aktualizací aplikace byla vydána v roce 2015, kdy je možné si do chytrého telefonu stáhnout mapy všech evropských států a mít je offline v telefonu. Uživatelé tedy mohou využívat službu i bez přístupu k internetu. Aktualizace nabízí i Turistickou mapu a to včetně cyklostezek i pro zahraniční státy. Zavedena byla i nová služba Batůžek, která dovoluje synchronizaci naplánovaných tras mezi počítačem a mobilní aplikací. Mobilní aplikace navíc automaticky lokalizuje jazyk systému a zobrazuje se ve zvoleném jazyce[11].

Uživatelé mohou na své stránky umístit webovou mapu za pomoci Mapy API, která používá framework JAK, který umožňuje mimo jiné objektově orientovaný přístup. API zpřístupňuje veškerou funkčnost, na kterou jsou uživatelé navyknutí, jako letecké snímky a turistické mapy. K dispozici jsou i SMART vrstvy, které intuitivně přepínají na zobrazení

leteckých snímků při větším přiblížení.



Obrázek 2.5: Ukázka webové mapy Mapy.cz

2.4 Zorné pole

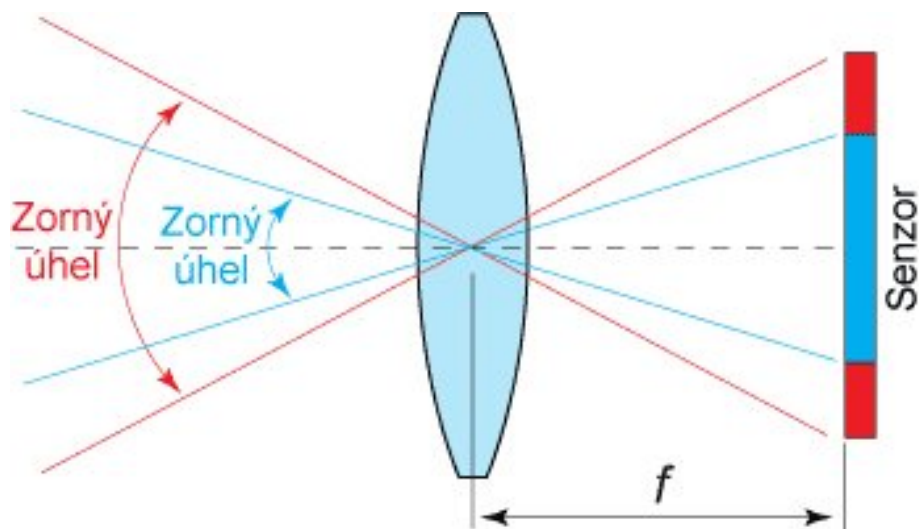
Zorné pole je název pro část prostoru, který jsme schopni zaznamenat a ze kterého přicházejí do objektivu fotoaparátu světelné paprsky. Definiuje tedy velikost scény vykreslené na zaznamenané fotografii. Lze ho také vyjádřit číselně, takové vyjádření se nazývá zorný úhel. Ten je ovlivněn ohniskem objektivu a velikostí senzoru fotoaparátu. Existují tři způsoby jak měřit zorný úhel. První možností je vertikální měření pomocí kratší strany senzoru, druhou možností je horizontální měření přes delší stranu senzoru a poslední možností je diagonální měření, které se provádí přes úhlopříčku senzoru. Zorný úhel (FOV) se počítá pomocí následujícího vztahu:

$$FOV = 2 * \text{atan}(0.5 * (SEN/f)) * (180/PI) \quad (2.1)$$

Zkratka FOV představuje počítané zorné pole, f reprezentuje ohniskovou vzdálenost a úhlopříčka senzoru je vyjádřena pomocí proměnné SEN . Největší zorný úhel lze zaznamenat pomocí širokoúhlých objektivů známých jako rybí oko, které dokáží zachytit prostor velikosti až 180 stupňů.

2.5 Panorama

Pod pojmem panorama si představíme celkový pohled na krajinu, okolí nebo jakýkoliv celek. Většinou se používá u fotografií nebo videozáznamů. Typ panoramatické fotografie se začal objevovat v polovině 19. století a jejím hlavním specifickým znakem je velký úhel záběru, větší, než jsou fotoaparáty schopny zaznamenat[13]. Pro tvorbu panoramat je často používána technika slepování snímků, kdy jsou fotografie ořezány a v grafickém editoru se z nich poskládá



Obrázek 2.6: Zorný úhel závislý na velikosti senzoru, kdy Senzor představuje proměnnou SEN zorný úhel proměnnou FOV .

výsledné panorama. Panoramatické fotografie se často využívají v běžně používaných aplikacích jako například Google Street View. Při vytváření panorama je nutné zvolit jakým způsobem budou vstupní snímky transformovány. Transformaci ovlivňuje způsob projekce (mapování) – planární, cylindrické a nebo sférické.

2.5.1 Planární (rovinná) projekce

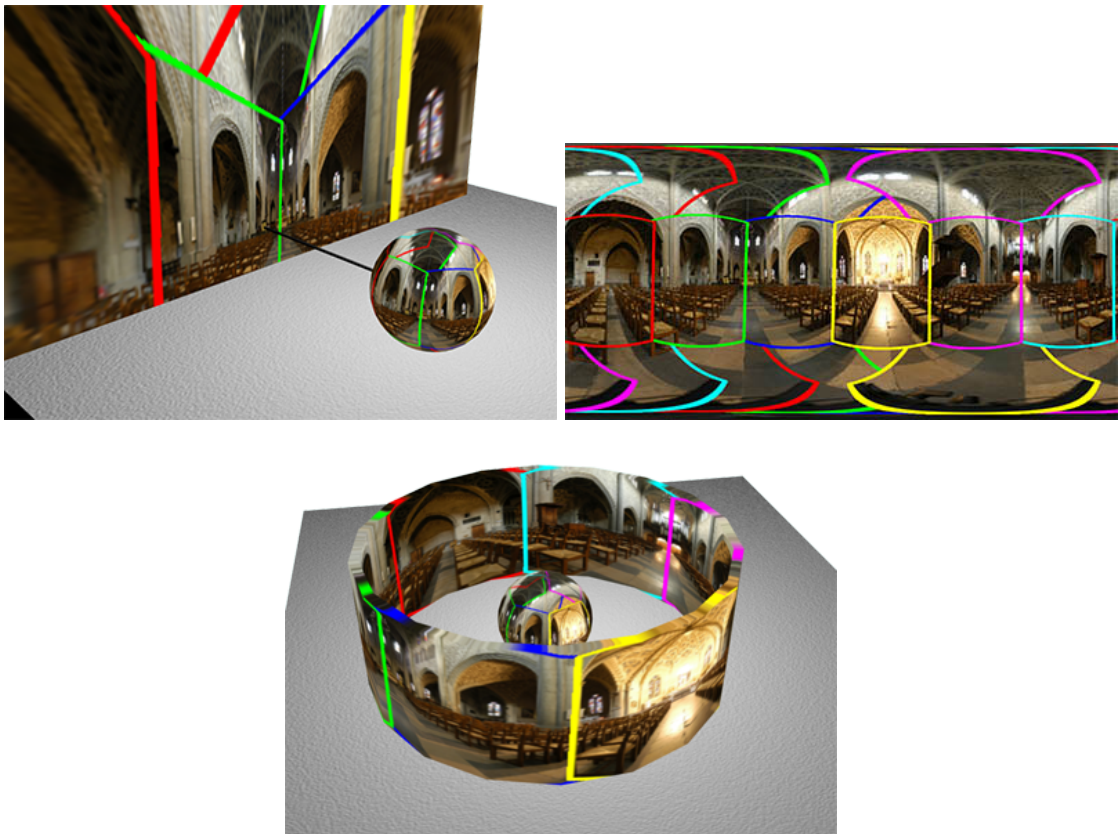
Planární projekce je velmi jednoduchá pro skládání panoramat. Jak plyne z názvu, výsledné panorama je vyobrazeno do rovinné plochy, neberou se zde v úvahu žádné perspektivní korekce[12]. Do snímku lze namapovat pouze pixely natočené k projekční rovině. Poté je zde skutečnost, že pixely, které se nachází na okraji projekční roviny, budou velmi roztažené. V praxi se nejvíce používá pro zaznamenání menších celků, které nelze vtěsnat na jednu fotografii. Není nutné pokrýt celých 360° .

2.5.2 Sférická (kulovitá) projekce

Kulová panoramatická projekce nám ze všech možných panoramatických zobrazení poskytuje nejpreciznější podání reality. Nazývá se také jako zobrazení $360^\circ \times 360^\circ$. Navíc je výhodou, že není potřeba provádět výpočty pro posun jednotlivých pixelů. Souřadnice vstupní fotografie je nutné při vytváření projekce převést do sférických. Při zobrazování sférické projekce lze horizontálně zobrazit $120^\circ - 360^\circ$, vertikálně lze zobrazit až 180° [1].

2.5.3 Cylindrická (válcová) projekce

Válcové panorama umožňuje namapování panoramatického snímku, povrchu koule, na povrch válce. Zde dochází k roztažení pixelů v oblasti středu spodního a horního okraje projekce[12]. Při tomto zobrazení jsme schopni získat vodorovně rozsah 360° a tak dochází k rovnosti pouze vertikálních linií. Horizontální zobrazení je $120^\circ - 360^\circ$, vertikální pak pouze až válcová projekce je tedy méně přesná než kulová.



Obrázek 2.7: Nahoře vlevo: ukázka planární projekce, nahoře vpravo: ukázka sférické projekce, dole: ukázka cylindrické projekce[12].

Kapitola 3

Použité technologie

V této kapitole se čtenář seznámí s technologiemi, které byly nezbytné pro vytvoření této bakalářské práce. Bude zde představeno proč jsou zmíněné jazyky, technologie nebo knihovny používány, jejich dostupné verze a porovnání s možnými alternativami. Znalost následujících technologií je klíčová pro pochopení procesu vytváření bakalářské práce.

3.1 HTML5

HyperText Markup Language, nebo-li zkráceně HTML, je nejpoužívanější značkovací jazyk, který slouží k tvorbě webových stránek, lze se snadno naučit a je podporován téměř všemi zařízeními. Umožňuje v systému World Wide Web publikování dokumentů, které jsou dále propojeny hypertextovými odkazy. Jazyk HTML je vhodný pro definování významu obsahu stránek. Vývoj jazyka byl a stále je ovlivňován webovými prohlížeči, konkrétně směrem a způsobem jejich vývoje.

Jazyk jsem do práce zařadil, jelikož se jedná o nejběžnější a nepoužívanější způsob tvorby dokumentů na internetu. Nespornou výhodou je, že kterýkoliv element je charakterizován množinou značek. Značky mohou mít vlastní identifikátory nebo je možné zařadit je do některé z tříd. Elementy uvozené značkami lze poté pomocí tříd DOM hierarchie uchopit, odkázat se na ně, a lze jim přiřadit styl nebo animaci. Jazyk HTML tedy umožňuje rozšiřovat webové stránky o další funkcionalitu.

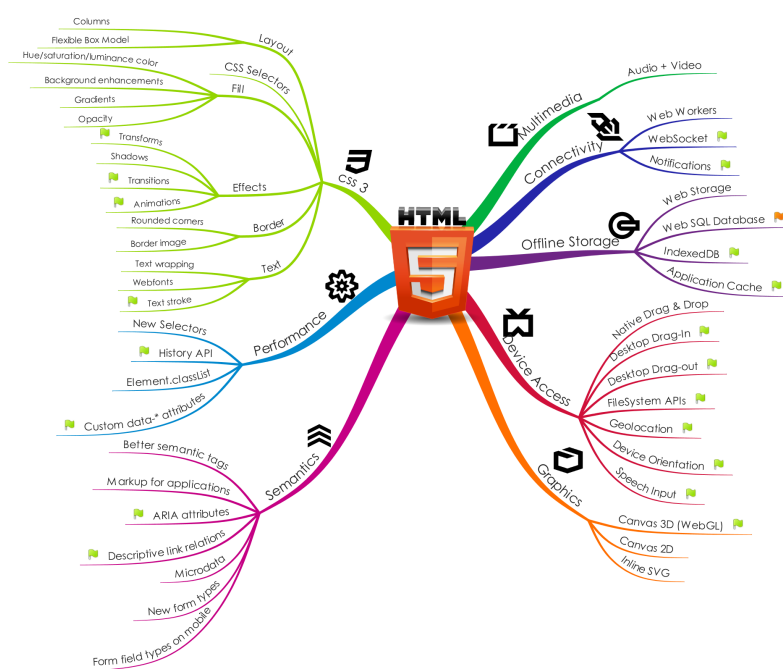
Nejnovější vydanou verzí je verze s označením 5.0, která byla vydána po dlouhých patnácti letech skupinou WHATWG¹ a konsorciem W3SC². Tato verze je významná tím, že už nepatří do rodiny SGML jazyků a opravuje chyby, nedostatky a nepoužívané prvky předešlých verzí. Vydání kromě jiného přineslo nový parsovací algoritmus, který je schopen generovat konzistentní DOM strom i z hůře označovaného dokumentu. Další výhodou je, že dokumenty spolu mohou komunikovat pomocí výměny zpráv[7]. V neposlední řadě se zde nachází nové moderní technologie, v práci jsou využity především tyto:

- **canvas** – nachází se zde podpora vkládání obrázků, další z možností je kreslení do prvku na straně klienta pomocí skriptovacích jazyků, element je využit pro práci s technologiemi pro vykreslení grafiky jako je WebGL (viz sekce 3.4),
- **formulář** – možnost automaticky dokončovat formuláře nebo vypnout validaci vstupních prvků,

¹Skupina WHATWG: <https://whatwg.org/>.

²Konsorcium W3SC: <https://www.w3.org/>.

- objekt `sessionStorage` – je využit pro uchování dat pro jedno sezení, data jsou smazána při zavření záložky prohlížeče



Obrázek 3.1: Funkcionalita jazyka HTML5[8].

3.2 CSS

V originálním názvu Cascading style Sheets je jazyk, který nám definuje styl, způsob zobrazení jednotlivých prvků definovaných jazykem HTML. Využívá se např. pro definici barvy, textu a nebo zarovnání jednotlivých prvků. Umožňuje oddělení struktury a stylu webové stránky. Jedná se o přirozené rozšíření jazyka HTML. Navíc je webový prohlížeč schopen uložit si stylový soubor do paměti typu cache a dochází tak k rychlejšímu načítání obsahu. Pomáhá také při vývoji zvláště oddělit vzhled stránky od jejího obsahu, to znamená, že jeden příložený soubor jazyka CSS může ovlivnit celý vzhled výsledné stránky. Při použití kaskádových stylů lze vrstvit jednotlivé definice, jak už ale plyne z názvu, platí pouze ta poslední. Stylový soubor navíc lze velice lehce a elegantně zařadit do HTML pomocí odkazu na soubor, ve kterém je stylový kód zapsán[7].

Nejnovější verze, CSS3, byla do jisté míry reakcí na nejnovější verzi jazyka HTML, konkrétně na verzi 5. Přináší podporu pro animace, přetékání prvků, zkracování obsahu s možností expanze, definování zaoblených rohů, vlastnosti pro drag'n'drop nebo 2D a 3D animace.

3.3 JavaScript

JavaScript je skriptovací objektově orientovaný jazyk, za jehož vznikem stojí Brendan Eich. Používá se pro tvorbu dynamických WWW stránek, navíc je možné vytvořený script vložit

i přímo do kódu jazyka HTML. Jedná se o interpretovaný programovací jazyk a můžeme díky němu vkládat interaktivní prvky do webové stránky a tvořit moderní webové aplikace. V současné době je podporován a zabudován v naprosté většině webových prohlížečů. Jako alternativu jazyka JavaScript lze uvést jazyky JSX, který je k dispozici pod licencí MIT, PythonJS a CoffeeScript. Nicméně pro tyto jazyky není tak rozšířená podpora ze strany prohlížečů, nestojí za nimi tak rozsáhlá komunita a nelze k nim vyhledat velké množství knih, učebních materiálů a tutoriálů.

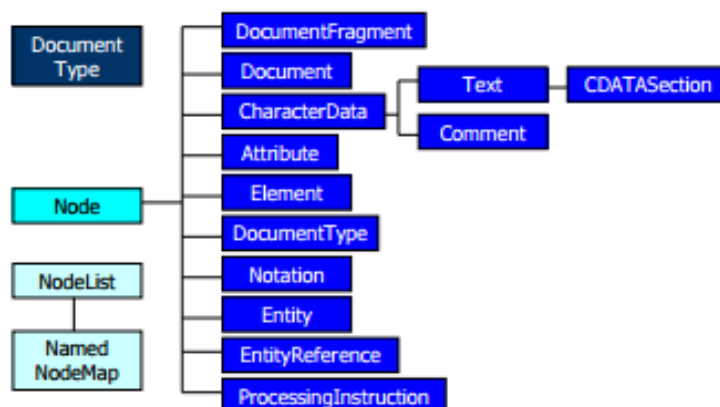
Jazyk se syntaxí podobá jazyku C a ačkoliv je slovo Java obsaženo v názvu, s jazykem Java nemá krom lehké se podobající syntaxe, některých převzatých jmen a jmenných konvencí, nic společného a nachází se tu jen z komerčních důvodů[2].

Klasické interpretované jazyky, jako PHP nebo ASP, pracují na straně serveru, což znamená, že pracují ještě předtím, než je z internetu stažena stránka. Naopak pracuje program, který je napsaný v JavaScriptu. Takový program se spouští na straně klienta, což znamená, že webová stránka je nejprve kompletně stažena z internetu a až poté je spuštěn skript.

3.3.1 jQuery

„jQuery je rychlá a lakonická knihovna, která ulehčuje zpracování událostí, animací a interakci s Ajax pro rychlejší vývoj webu. Jazyk jQuery byl vyvinut proto, aby změnil metody psaní JavaScript kódu.“[16] Jedná se o knihovnu, která byla vydána na akci s názvem BarCamp ve městě New York a jejím autorem se stal John Resig. Software, současně pod licencí MIT, je integrován a nebo dodáván na nejrůznějších platformách společností jako jsou např. Nokia nebo Microsoft. Jedná se o svobodný a otevřený projekt.

Pomocí jQuery knihovny lze psát kratší a čitelnější kód v porovnání se samotným JavaScriptem. Velice jednoduché je zde zacházení s knihovnou AJAX, kdy je často potřeba jen málo řádků kódu pro napsání požadavku. Technologie jQuery odděluje a specifikuje chování od struktury HTML. Dále je zde dále silná interakce s DOM elementy, umožněná selektorovým enginem Sizzle. Pomocí již zmíněné interakce k HTML prvkům můžeme jednoduše přistupovat, modifikovat je, a nebo můžeme vytvářet prvky zcela nové.



Obrázek 3.2: Ukázka tříd DOM struktury[17].

3.3.2 AJAX

V originálním znění Asynchronous JavaScript and XML umožňuje asynchronní zpracování webových stránek a interaktivních webových aplikací. Jedná se o knihovnu napsanou v jazyce JavaScript, díky které může stránka změnit svůj obsah a zároveň nemusí být celá znovu načtena, což je hlavním důvodem použití technologie v této baklářské práci.

Pro asynchronní přenos dat mezi klientskou částí a webovým serverem se používají zejména dva formáty. Prvním z nich je XMLHttpRequest, jedná se o rozhraní umožňující komunikaci pomocí protokolu HTTP. Připravovaná specifikace organizace W3C si slibuje sjednocení chování napříč webovými prohlížeči.

Druhým využívaným formátem je JSON, definuje způsob zápisu dat, který není závislý na konkrétní platformě. Pracuje na principu převedení libovolné datové struktury do textového řetězce, který je výstupem. Složitost vstupní datové struktury teoreticky není omezena. JSON nebere v úvahu kódování textu, jako výchozí kódování je použito UTF-8.

3.3.3 Leaflet

Leaflet je velice často používanou open source JavaScript knihovnou, která uživatelům umožňuje pohodlně vytvářet aplikace typu webových map. Je podporována na většině mobilních i desktopových platformách[20]. Jedná se o přímo srovnatelnou knihovnu s knihovnou OpenLayers, nicméně Leaflet je modulárně tvořena knihovna obsahující mnohem méně řádků velmi čitelného kódu a je postavena na nejnovějších funkcích jazyka JavaScript. Technologie má výborně zpracovanou dokumentaci a návody k použití narozdíl od Mapi API společnosti Seznam.cz. Jak už bylo zmíněno, knihovna Leaflet má veřejně přístupné zdrojové kódy, což byl jeden z faktorů, proč jsem dal knihovně přednost před Google Maps API.

Leaflet knihovna pracuje se souřadnicemi typu WGS-84 (viz obrázek 2.2). Vyskytuje se zde podpora nejnovějších technologií jako HTML5 a CSS3. Leaflet knihovna je využívána na velkých webových sítích jako např. Flickr, Pinterest, OpenStreetMaps a FourSquare. Dalšími významnými knihovnami pro tvorbu webových map jsou OpenLayers a Google Maps API.

Služba je hostována na veřejném serveru. Poskytuje uživatelům možnost vytvářet interaktivní vrstvy jako jsou markery, vyskakovací okna či mnohoúhelníky. Typicky se mapuje Leaflet map element do HTML div elementu.

Leaflet také podporuje několik druhů vrstev – vektorové, WMS (Web Map Service), a nebo GeoJSON. Další je možné dodefinovat formou pluginů. Základním zobrazeným modelem je jedna mapa, ke které je možné přidat žádnou nebo více vektorových vrstev. Dále je podporováno několik základních typů objektů:

- rasterové (raster types): `ImageOverlay`, `TitleLayer`,
- vektorové (vector types): `Path`, `Polygon`, `Circle`,
- skupinové (grouped types): `GeoJSON`, `LayerGroup`,
- kontrolní (controls types): `Zoom`, `Layers`.

3.4 WebGL

WebGL je JavaScriptové API, používané pro vykreslování grafiky za pomoci elementu `canvas` jazyka HTML. Architektura je identická s technologií OpenGL ES 2.0. V současné

době je to jediná možnost práce s grafikou ve webových aplikacích. Nejčastěji se používá k tvorbě her. Jedná se o velmi flexibilní a výkonný nástroj, jehož syntaxe je poněkud zdlouhavá. Složitější je zde ladění, kdy je lepší postupovat vždy po menších částech[14].

Hlavní součástí technologie jsou shadery. Ve WebGL jsou dostupné dva, fragment shader a vertex shader. Vertex shader je volán pro každý vrchol objektu, naopak fragment shader je vykonáván pro každý vykreslovaný pixel. Pokud předáváme proměnnou z vertex shaderu do fragment shaderu, dochází k interpolaci hodnoty, výsledek interpolace tedy určuje, jak daleko se nacházíme od vykreslovaného vrcholu. Shadery jsou programy, které jsou napsané v jazyce GLSL, jenž má syntaxi velice podobnou jazyku C. Programy jazyka GLSL jsou ovladačem grafické karty kompilovány do kódu, který se poté vykoná na samotné grafické kartě. Tento způsob vykonávání programu zajišťuje extrémní rychlost a výkon. Pokud chceme předat shaderu nějakou informaci, je zapotřebí použít buffer. Pomocí něho se alokuje paměť, ke které má grafická karta rychlý přístup.

3.5 PHP

Začátky vývoje jazyka PHP sahají do roku 1994, v té době se Rasmus Lerdorf snažil vytvořit počítadlo přístupů k jeho webové stránce. Za použití jazyka C vznikly skripty, které poté dostali název PHP. V současné době se jedná o jeden z nejrozšířenějších skriptovacích jazyků, podporujících tvorbu dynamických webových stránek. Navíc se jedná o naprosto nejpoužívanější modul webového serveru Apache. PHP lze využít i pro psaní desktopových nebo konzolových aplikací. Používá se na straně serveru, skripty jsou provedeny ještě než je stránka přenesena uživateli. Syntaxe jazyka je odvozena a inspirována jazyky C a Perl. Interpret skriptu je možné zavolat třemi způsoby, prvním je přes webovou službu, druhou možností je příkazový řádek a poslední způsob představuje dotaz přes hypertextový protokol HTTP[3].

Hlavními důvody použití jazyka bylo, že se jedná o multiplatformní jazyk, který lze provozovat s naprostou většinou webových serverů, snadno komunikuje s databázemi, především tedy s MySQL (viz sekce 3.6), podporuje mnoho souvisejících technologií, formátů a standardů, dále se jedná o otevřený projekt, který je podporován širokou komunitou a v neposlední řadě je dnes podporován naprostou většinou poskytovatelů webhostingu.

Dalšími možnostmi, mezi kterými se lze rozhodnout při psaní serverové části jsou jazyky Python a Rubi. Obě varianty jsou velice zajímavé, jelikož za nimi stojí velké projekty jako GitHub a Google Sketchup na straně Rubi a Yahoo Map nebo Shopzilla na straně Pythonu. Nicméně Python ne zrovna dobře zvládá práci s více vlákny, jsou zde limity pro přístup k databázovým vrstvám a je považován za pomalejší jazyk než je PHP. Jazyk Rubi zase postrádá informační zdroje, což způsobuje pomalejší progres při učení se jazyka, navíc vývoj a aktualizace jsou oproti konkurenci pomalejší. Za oběma jazyky stojí menší komunita vývojářů než za PHP.

Verze PHP s pořadovým číslem 5, vydaná v roce 2004 nabízí vylepšenou podporu objektově orientovaného programování, pro databáze bylo zavedeno konzistentní rozhraní pro napojení a výrazné zlepšení výkonu a rychlosti.

3.6 MySQL

MySQL je multiplatformní databázový systém (systém řízení báze dat), který je v současné době vlastněn společností Sun Microsystems, jejíž dceřinou společností je Oracle Corpo-

ration. Systém MySQL je dostupný pod bezplatnou licencí GPL, což je jeden z hlavních důvodů jeho použití. Jelikož jde o multiplatformní, snadně implementovaný a volně šiřitelný software, je v dnešní době jeden z nejpoužívanějších systémů. Velice známá a často používaná je kombinace LAMP - Linux, Apache, MySQL a PHP[3], která byla použita při vytváření bakalářského projektu.

Komunikace s databází probíhá pomocí jazyka SQL, ke kterému jsou přidána drobná rozšíření. Od počátku vývoje této technologie se kladl důraz na rychlost a to i na úkor zjednodušeného zálohování. V posledních letech docházelo k přidávání triggerů, vložených procedur a pohledů.

Základem MySQL jsou tabulky, které se skládají z položek jednoho typu. Položky jsou ukládány takzvaně na řádek, sloupce se poté vyznačují jako jejich atributy nebo-li vlastnosti. Atributy jsou typované, což znamená, že hodnoty atributu jsou omezeny na základě typu. Každý řádek tabulky by měl být specifikován unikátním identifikátorem, který umožňuje odkazovat se do jiné tabulky a zpřístupnit její položky.

Rozdílem oproti jednoduchému ukládání textových souborů nebo XML souborů je, že řeší zabezpečení a optimalizaci výkonu. Zabraňuje nežádoucím stavům, jako je nekonzistence dat, ošetřením současného přístupu více uživatelů k jedné položce, což znamená, že jsou zde zavedeny zámky, které uchovávají data konzistentní. Dalším rozšířením je kontrola stavu transakcí a zamezení částečné nebo nedokončené transakce.

Kapitola 4

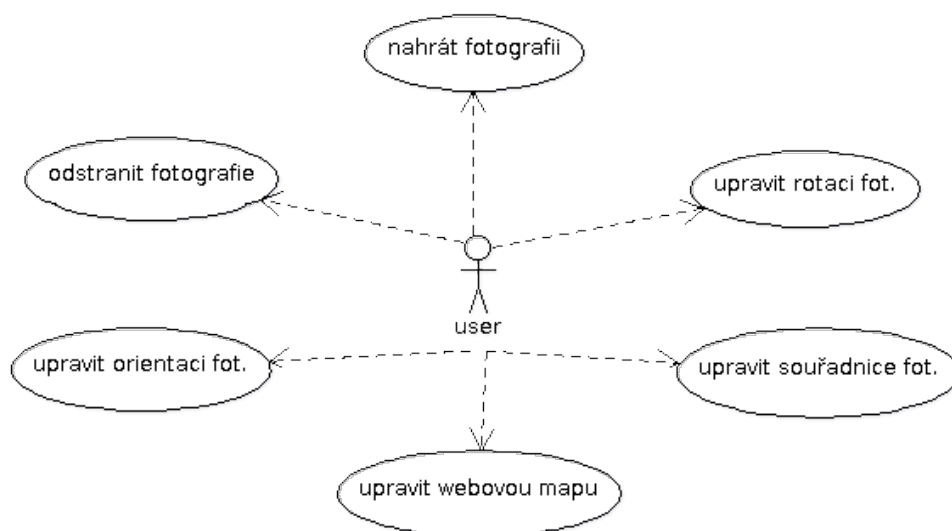
Návrh a implementace

Požadavky na funkčnost, kterou by měly webové nástroje poskytnout si lze odvodit ze zadání práce samotné. Za prvé je potřeba vytvořit nástroj pro vizualizaci výsledků, které nám poskytuje existující geolokalizační systém. Tento úkol lze vyřešit pomocí online webové mapy, kterou bude potřeba zahrnout do projektu a která elegantně vizualizuje nahrané fotografie uživatele.

Pro registraci obrazu do digitálního modelu terénu byl použit jako základ nový prvek <canvas> technologie HTML5 (viz sekce 3.1). Nabízí nám definovat plátno, do kterého lze vkládat a překrývat přes sebe nové vrstvy. Interakce s uživatelem je zde zajištěna pomocí posuvníků a nasloucháním signálů, která je implementována knihovnou jazyka JavaScript jQuery.

Pro potřeby výměny dat mezi klientskou a serverovou částí je využita technologie AJAX, pomocí které jsou přeneseny požadované informace na stranu serveru, kde jsou pomocí SQL dotazů zpracovány.

Uživatel má pochopitelně možnost zkoumat nebo upravovat poskytnuté výsledky. U fotografiích lze měnit jejich souřadnice za pomoci interaktivní drag'n'drop online mapy, navíc je možnost měnit umístění fotografie v digitálním modelu.



Obrázek 4.1: Use case diagram uživatele aplikace.

4.1 Klientská část

V této části se budu věnovat klientské části aplikace. Tedy prvkům aplikace, které uživatel uvidí na stránce. Některé, povětšinou textové elementy, jsou navrženy pro předání získaných informací. Ostatní prvky, jako digitální model terénu nebo webová mapa, jsou navrženy pro interakci s uživatelem, který může dodefinovat nebo upravit získané výsledky.

4.1.1 Nahrávání fotografií

Tento prvek slouží pro získání fotografií od uživatele, který může nahrát jednu nebo více fotografií do systému. Pro pohodlnost uživatele a ovládání aplikace existují dva druhy výběru fotografií.

Prvním způsobem je metoda prohledávání souborové hierarchie, kdy se po stlačení tlačítka otevře systémové okno se soubory uloženými na disku uživatele. Vybrané soubory se nahrají do aplikace stiskem tlačítka „Otevřít“. Druhou možností nahrání fotografií do systému je přetažením jednoho nebo více souborů do připraveného `div` elementu. Ten pro přehlednost reaguje změnou okrajů, pokud je nad ním přejeto kurzorem s vybranými soubory.

Nahrávání souborů pomocí prohledávání souborové hierarchie je vytvořeno pomocí elementu `form`, který využívá `POST` metody pro přenesení vybraných dat do souboru s názvem `uploading_files.php` po stlačení tlačítka „Nahrát“. Důležitou roli zde hraje atribut formuláře `enctype="multipart/form-data"`, který povoluje vybrat a nahrát více než jeden soubor. Pomocí `jQuery` je na tlačítko aplikována metoda `addEventListener()` kontrolující jeho stisknutí. Poté, co je stlačeno, se vyberou z formuláře veškeré soubory a předají další funkci: `upload_files(document.getElementById('std_upload_files').files)`.

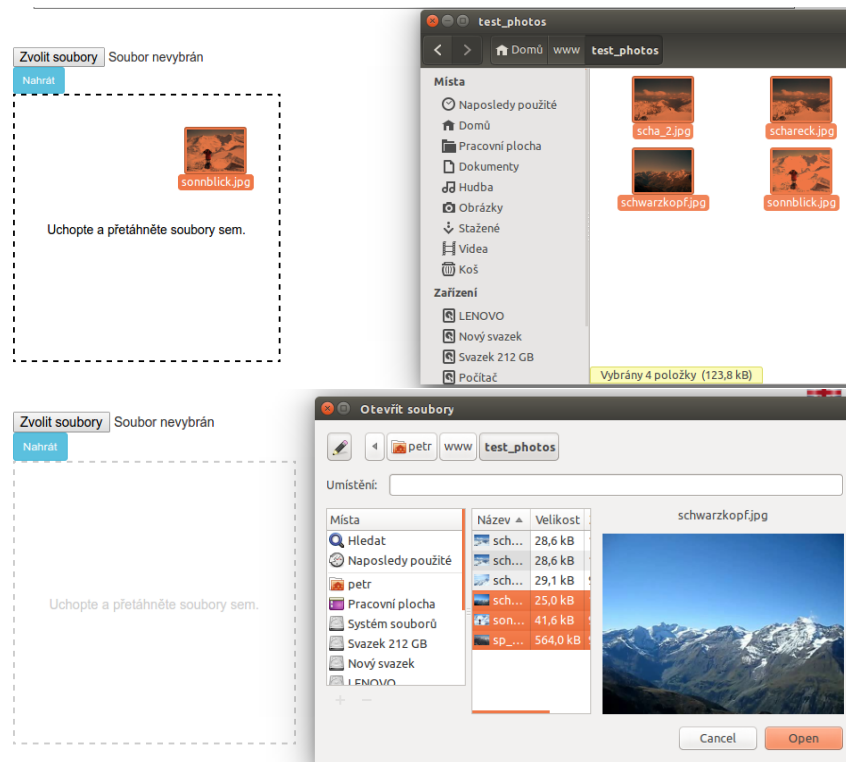
Přetahování souborů je implementováno pomocí `div` elementu, kterým byl vytvořen obdélníkový prostor pro možnost jejich upuštění. Na element jsou vytvořeny události `ondragover` pro jeho zvýraznění, je-li na něm kurzor nesoucí soubory a `ondragleave`, který vrátí původní vzhled elementu pokud ho kurzor myši opustí. Událostí `ondrop` se zavolá funkce, která z objektu události získá nahrané soubory a předá je pro další zpracování následujícím způsobem: `upload_files(e.dataTransfer.files)`.

4.1.2 Webová mapa

Webová mapa je jeden z nejdominantnějších prvků reprezentujících výsledky geolokalizačního systému. Uživatel si může na základě vlastního rozhodnutí zvolit z více druhů map (topografická, mapa ulic, oceánů atd.). Do mapy jsou, na získané souřadnice, vloženy náhledy fotografií nebo body. Další vrstvu tvoří polygon představující vypočítaný zorný úhel, který navíc vyobrazuje orientaci fotografie. Pro přehlednost je polygon vyplněn zvolenou barvou, kterou je možné změnit pomocí kontrolních tlačítek vložených do mapy, kdy každé představuje jinou barvu jeho výplně.

Mapa je tvořena objektem `L.map`, jehož konstruktoru se jako jeden z parametrů předává identifikátor `div` elementu, do kterého se objekt zobrazí. Pomocí volitelných parametrů se definuje úroveň přiblížení, souřadnice místa, které se zobrazí po načtení a v neposlední řadě ohraničení mapy.

Náhledy fotografií a body jsou reprezentovány objektem `L.marker`, který má povinný parametr určující souřadnice místa, kde se zobrazí. Jediným rozdílem při vytváření náhledu je, že je mu nastaven parametr `icon`, kterému je předán objekt `L.icon` obsahující cestu k fotografii, ze které je vytvořen náhled. Pomocí globálního parametru `drgging` lze určit,



Obrázek 4.2: Nahoře: nahrávání fotografií přetažením, dole: nahrávání fotografií pomocí formuláře.

jestli bude možné přesouvat objekty mapy. Při přetahování je pomocí metody `on('drag')` volána funkce, získávající modifikované souřadnice objektu metodou `getLatLng()`. Takto získané souřadnice slouží pro přepočítání pozice všech ostatních vrstev, které souvisí se změněným objektem.

Další vrstvou mapy je polygon reprezentující, jak už bylo zmíněno, zorný úhel objektivu (viz obrázek 2.6), kterým byla fotografie pořízena. Polygon představuje rovnoramenný trojúhelník skládající se ze tří bodů. První bod C se nachází na získaných souřadnicích, druhý bod B se získá přičtením výšky h k zeměpisné výšce a odečtením poloviny délky strany c od zeměpisné šířky. Poslední bod A je určen taktéž přičtením výšky trojúhelníku h k zeměpisné výšce, ale k zeměpisné šířce tentokrát přičítáme polovinu délky strany c . Úhel γ je shodný s velikostí zorného úhlu.

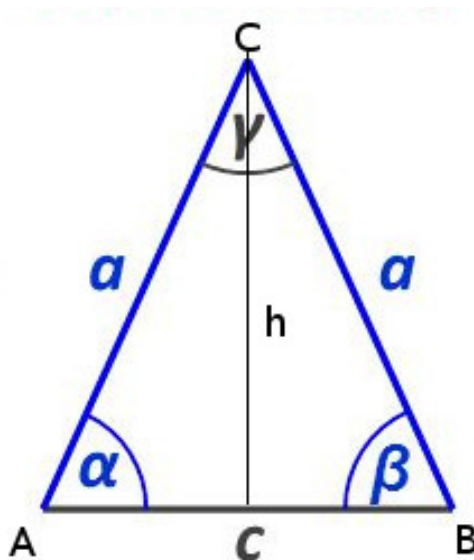
$$c = 2 * \sin\left(\frac{\gamma}{2}\right) \quad (4.1)$$

$$h = \sqrt{1 - c^2} \quad (4.2)$$

Takto získaný polygon má nicméně orientaci 0 stupňů. Pro zobrazení polygonu se správnou orientací slouží funkce `rotatePoint()`, které je jako parametr předáván přepočítávaný bod trojúhelníku, otočení ve stupních (`angle`) a počáteční bod (Cx, Cy). Funkce vrací nově vypočítané pozice zeměpisné výšky (x) a zeměpisné šířky (y) bodů.



Obrázek 4.3: Ukázka webové mapy s kontrolními tlačítky.



Obrázek 4.4: Rovnoramenný trojúhelník pro odvození výpočtu zorného úhlu.

$$x = \cos(\text{angleRad}) * (x - Cx) - \sin(\text{angleRad}) * (y - Cy) + x \quad (4.3)$$

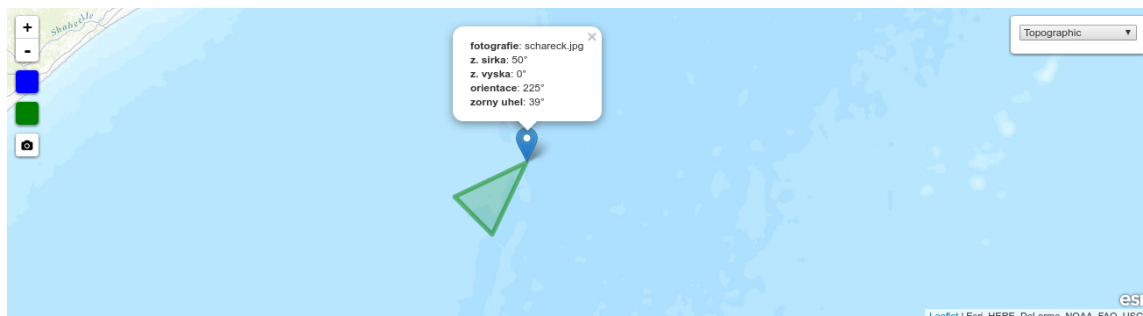
$$y = \sin(\text{angleRad}) * (x - Cx) + \cos(\text{angleRad}) * (y - Cy) + y \quad (4.4)$$

Otočený polygon se poté přidá do mapy pomocí metody objektu `L.map addTo()`.

Kontrolní tlačítka pro změnu barvy polygonu jsou tvořeny objektem `L.easyButton`. Jako parametry konstruktoru mu jsou předávány informace o umístění v mapě, identifikátor pro přístup pomocí DOM struktury, vyskakovující popis po přejetí kurzorem, ikona tlačítka a stav, který popisuje akci po jeho stisknutí. Změna barvy polygonu se provádí iterací nad globálním polem `polygonLayer`, ve kterém jsou uloženy veškeré polygony zobrazené na mapě, kdy každému polygonu je nastaven atribut `fillColor` metodou `setStyle()` na novou hodnotu barvy. Pro vytvoření tzv. `toggle` tlačítka, pomocí něhož přepínáme mezi zobrazením bodů a náhledů fotografií, je potřeba definovat dvakrát funkční atribut `state`. Všechny body se nachází v globální proměnné `markerGroup` a náhledy jsou uchovány v proměnné `photoGroup`, předáváme je metodě `removeLayer()` pokud chceme některou ze

skupin odstranit a metodě `addLayer()` pokud naopak chceme skupinu přidat na mapu.

Změna základní mapy je provedena pomocí elementu `select`, který při změně zavolá funkci `setBasemap()` s hodnotou nově zvolené mapy. Tato hodnota se použije v konstruktoru objektu `L.esri.basemapLayer` a metodou `addTo()` se nově zvolená mapa zobrazí uživateli.



Obrázek 4.5: Výsledná podoba webové mapy s přidávanými vrstvami.

4.1.3 Digitální model terénu

Digitální model terénu je druhou možností jak uživateli zprostředkovat informace získané systémem LOCATE (viz sekce 2.1). Jedná se o prvek aplikace, kdy zpracovávaná fotografie je umístěna do syntetického panorama krajiny na základě hodnot `pitch`¹, `roll`² a `yaw`³. Samozřejmě uživatel nemusí být vždy plně spokojen s výsledkem, z toho důvodu je důležité navrhnout a vytvořit způsob, pomocí kterého lze upravit umístění a rotaci fotografie v panoramatu. Jelikož je vyobrazené panorama je válcové, bude nezbytné rovinnou fotografii namapovat na válec.

Základem prvku digitálního modelu terénu je element jazyka HTML, canvas. Pomocí něho lze pracovat s grafikou na webových stránkách, která je nezbytná pro namapování obrázku na válcové panorama. Nejprve je zapotřebí pomocí elementu canvas vytvořit prostředí pro WebGL program, který se skládá z fragment shaderu a vertex shaderu. Touto `canvas.getContext("webgl")` metodou se do proměnné `gl` vytvoří prostředí pro práci s WebGL. Dále se z externích souborů `program.fs` a `program.vs` načte fragment respektive vertex shader, ty jsou dále metodou `compileShader()` sestaveny a metodou `attachShader()` přidány do WebGL programu o jehož vytvoření a přiřazení do proměnné `program` se stará metoda `createProgram()`. Obrázek panoramatu a fotografie je nutné předat do fragment shaderu jako textury, ty vytváříme metodou `createTexture()`. Následně získáme do WebGL programu uniform proměnnou z fragment shaderu definující texturu pomocí metody `getUniformLocation()` a nahraný obrázek předáváme následující metodou: `texImage2D()`.

Interakce s uživatelem je vytvořena pomocí posuvníků, kterými lze měnit hodnoty `pitch`, `roll` a `yaw`. Dalším způsobem jak upravit digitální model terénu je ruční změna pozice a rotace. Zachytáváním událostí myši lze docílit chování, kdy pokud uživatel stiskne levé tlačítko myši, může měnit pozici fotografie, pokud je stlačeno pravé tlačítko, mění se její rotace. Vizualizace změny je zajištěna funkcí `redraw()`, která se opakovaně každých 60 ms spouští

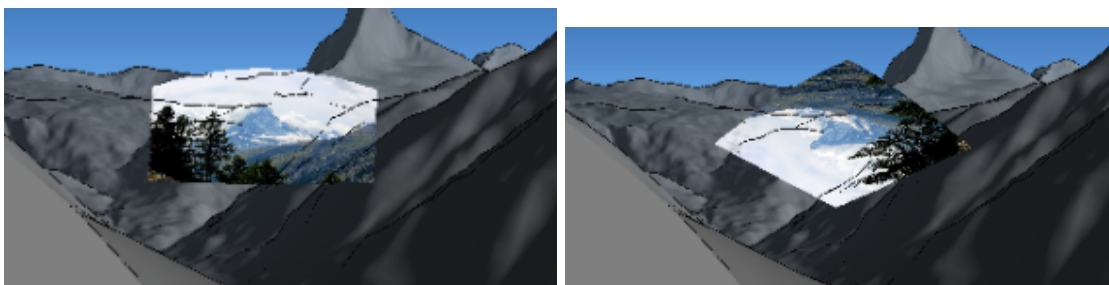
¹Pitch definuje velikost úhlu pod nebo nad horizontem.

²Roll je úhel rotace kamery kolem její optické osy.

³Yaw je úhel horizontální rotace kamery.

a volá `drawElements()`, jenž způsobí překreslení prostředí. Aby nedocházelo k neustálému překreslování, byla vytvořena proměnná `redrawStatus` nabývající `true` hodnoty jen při změně nějaké vlastnosti a pouze v tuto chvíli dojde k zavolání metody `drawElements()`.

Vykreslování celé scény je realizováno pomocí již zmíněných fragment a vertex shaderů, kdy se pro každý vykreslovaný pixel určí barva z jedné ze dvou dostupných textur, která se vykreslí. Na základě velikosti fotografie se nejprve určí souřadnice pro její texturu, poté se provede přepočítání souřadnic pro zobrazení textury na válec. Dalším krokem je aplikace transformačních matic a matice pro perspektivní projekci, násobením zmíněných matic vznikne orotovaný a posunutý pixel, jehož barva se získá z textury pomocí funkce `texture2D()`. Stejnou funkcí je pro vypočítanou pozici získávána barva z textury panoramatu, pokud se jedná o hranu černé barvy, je vykreslena tato hrana. To nám zajišťuje průhlednost fotografie na černé hrany panoramatu.



Obrázek 4.6: Vlevo: ukázka průsvitnosti fotografie na hrany panoramatu, vpravo: ukázka rotace fotografie.

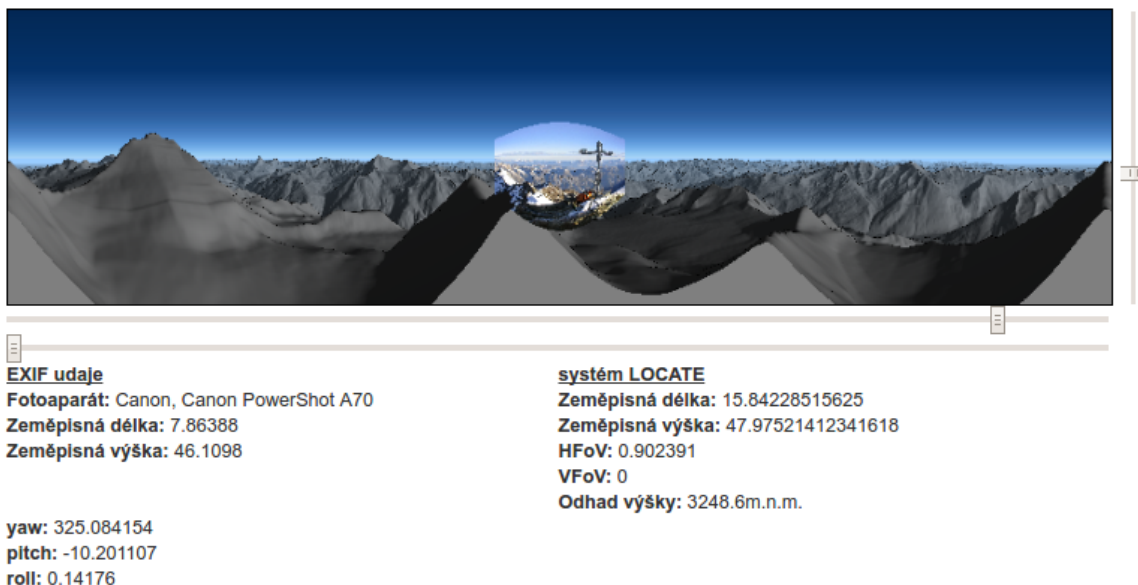
Digitální model terénu s sebou nese také doplňující informace. Jedny jsou získány z digitálních dat přibalených k fotografii (viz sekce 4.2.3), další informace jsou data získaná ze systému LOCATE. Na stránku jsou pak umístěny pomocí dynamicky generovaných `div` elementů jazyka HTML, kdy metodou `innerHTML` je předáván textový řetězec, který se na stránce vyobrazí. Vypisovány jsou pak následující informace: model fotoaparátu, zeměpisná délka a výška získaná z exif dat, zeměpisná délka a výška získaná ze systému LOCATE, horizontální a vertikální zorný úhel, odhad nadmořské výšky a hodnoty posuvníků.

4.1.4 Nastavení prostředí

Pro možnost jednoduššího nastavení prostředí a způsobu zobrazení jednotlivých elementů jsou k dispozici globální proměnné a funkce, pomocí kterých lze toto chování definovat a měnit.

Proměnné, které by se měly nastavovat při každém použití sady nástrojů jsou `modul_map` a `modul_digital_model`. Tyto proměnné definují, zda-li chceme v aplikaci zahrnout funkčnost webové mapy nebo digitálního modulu. Implicitně jsou nastavené na hodnotu `false`, která udává, že není využíván žádný modul. Nastavením některé nebo obou proměnných na hodnotu `true` definujeme, že budeme chtít daný modul používat.

Pro nastavení způsobu jakým jsou prezentované informace na webové mapě byly vytvořeny proměnné `added_type`, `polygon_color`, `polygon_size` a `dragging`. Proměnná `added_type` může nabývat hodnot `photos` nebo `markers` a definuje, jestli se na mapě budou zobrazovat fotografie nebo body. Pomocí `polygon_color` určujeme barvu, kterou se vykreslí ve webové mapě polygon reprezentující zorný úhel, `polygon_size` je koeficient



Obrázek 4.7: Digitální model terénu včetně získaných informací a hodnot jednotlivých posuvníků.

ovlivňující velikost zorného úhlu a **dragging**, která nabývá hodnot **true** nebo **false** učuje, jestli elementy v mapě půjdou přesouvat anebo ne.

Dále byly vyvinuty funkce, pomocí kterých lze docílit shování prvku pro nahrávání fotografií a webové mapy. Je nezbytné, aby HTML kód pro tyto dvě funkčnosti byl obsažen na stránce, jelikož jsou na ně vytvořeny v JavaScriptové sadě nástrojů implicitně vytvořeny vazby. K těmto účelům jsou vyvinuty funkce, které elementy shovají, jakoby nikdy nebyly ve zdrojovém kódu uvedeny. První z nich je `hide_uploading_files()`, která shová prvek pro nahrávání fotografií a druhou je `hide_web_map()`, která odstraní ze stránky webovou mapu.

Implicitní stav proměnných:

```

var modul_map = "false",
    modul_digital_model = "false",
    added_type = 'markers',
    polygon_color = 'green',
    polygon_size = 1,
    dragging = "true";
  
```

4.2 Serverová část

Tato kapitola pojednává o postupech a metodách, které jsou uživateli skryty. Nejsou vykonávány na straně prohlížeče nýbrž na straně serveru. Konkrétně budu pojednávat o způsobu ukládání fotografií, extrahování EXIF dat jenž jsou součástí digitální fotografie, výpočtu zorného úhlu pořízené fotografie a určování odhadu nadmořské výšky zaznamenaného místa.

4.2.1 Testovací sada

Pro simulaci výsledků geolokalizačního systému LOCATE (viz sekce 2.1) byla panem docentem Martinem Čadíkem poskytnuta testovací sada. Ta obsahuje fotografie hor, kde ke každé fotografii existuje také syntetické panorama krajiny a rozměry senzorů vybraných fotoaparátů. Sada navíc obsahuje o každé fotografii upřesňující informace, pomocí kterých ji lze umístit do webové mapy a digitálního modelu terénu. Pro usnadnění práce s poskytnutým materiálem byla vytvořena tabulka v databázi s názvem `photoparam`, jenž slučuje pro danou fotografii veškerá dostupná data. Nachází se zde primární klíč (`id`), zeměpisné souřadnice (`longitude`, `latitude`), hodnota horizontálního zorného úhlu (`hfov`), cesta k panoramatu a název fotografie (`filepath` resp. `name`) a údaje pro umístění fotografie do digitálního modelu terénu (`yaw`, `pitch` a `roll`).

photoparam	
* <u>id</u>	int(10)
◦pitch	double
◦roll	double
◦yaw	double
◦latitude	double
◦longitude	double
◦filepath	varchar(100)
◦name	varchar(100)
◦hfov	double

Obrázek 4.8: Tabulka databáze pro použití testovací sady.

Informace o rozměrech senzoru fotoaparátu jsou důležité pro výpočet jeho úhlopříčky. Ta je následně použita při získávání horizontálního zorného úhlu, jenž je použit při vizualizaci výsledků ve webové mapě. Pro přístup k těmto údajům byla vytvořena tabulka v databázi s názvem `model_sensor_dict`. Ta obsahuje primární klíč (`id`), název modelu fotoaparátu (`model`), velikost hrany `x` (`sensor_x`) a hrany `y` (`sensor_y`).

model sensor dict	
* <u>id</u>	int(12)
◦model	varchar(50)
◦sensor_x	float
◦sensor_y	float

Obrázek 4.9: Tabulka databáze pro získání úhlopříčky senzoru fotoaparátu.

4.2.2 Uložení fotografií

Fyzické uložení fotografie je nezbytné z důvodu jejího pozdějšího zpracování. Jelikož má uživatel možnost zobrazit stránku ve více záložkách, je nutné implementovat mechanismus,

který umožní na každé záložce zobrazit jiný obsah. Z tohoto důvodu je při každém otevření stránky generován unikátní identifikátor označující záložku webového prohlížeče. Fotografie, nahrané uživatelem, jsou předány na server, kde je do adresáře `upload/` vytvořen další adresář nesoucí název vygenerovaného identifikátoru. Do tohoto adresáře jsou již fyzicky ukládány nahrané fotografie. Server poté vrací vygenerovanou cestu. Ta je použita při vytváření náhledu do webové mapy a také při vytváření digitálního modelu terénu. Příklad zobecněné relativní cesty k fotografii:

$$\text{upload/unikatni_identifikator/nazev_fotografie} \quad (4.5)$$

Při vytváření unikátního identifikátoru je využit objekt `sessionStorage`, který je novinkou jazyka HTML5. Tento objekt ukládá data pro jednu konkrétní relaci. Data jsou následně vymazána, když uživatel zavře záložku prohlížeče⁴. Identifikátor je generován na základě objektu `Date`, ze kterého je metodou `getTime()` získána číselná kombinace reprezentující datum a aktuální systémový čas. Pomocí metody `setItem()` objektu `sessionStorage` je tato číselná kombinace nastavena do proměnné `key`.

Pro předávání informací serverové části slouží funkce `upload_files()`, která nejprve předá pomocí technologie AJAX do souboru `uploading_files.php` unikátní identifikátor. Poté jsou do objektu `FormData` jeho metodou `append()` nahrány fotografie uživatele. Po jeho naplnění se metodou `open()` objektu `XMLHttpRequest` otevře spojení typu `POST` mezi serverem a klientskou částí. Metoda `send()` zajistí odeslání fotografií serveru.

Na straně serveru je pomocí globální proměnné `$_POST` získána hodnota unikátního identifikátoru a funkcí `mkdir()`, pokud tak již nebylo učiněno, se vytvoří adresář pro uložení fotografií. Ty jsou získávány pomocí globální proměnné `$_FILES`. Uložení probíhá pomocí iterace nad takto získanými fotografiemi a metodou `move_uploaded_file()` jsou uloženy do vytvořeného adresáře. Server následně vrací pole s názvem a cestou ke každé takto uložené fotografii, výsledné pole se uloží jako globální proměnná `data_photo` ve skriptu `tools.js`.

4.2.3 Získávání dat

Pro potřebu pozdějšího zpracování fotografií je nutné nejprve získat a zpracovat všechna dostupná data. Ta jsou následně předávána jednotlivým funkcím, kde se data vyhodnotí a použijí při vytváření webové mapy nebo digitálního modelu terénu. Vytvořená funkce `get_data()` ve svém těle navazuje pomocí technologie AJAX (viz sekce 3.3.2) spojení se serverem, kdy do souboru `get_data.php` předává název a cestu ke zpracovávané fotografii a hodnotu `true` nebo `false`, která identifikuje, zda se používá modul digitálního modelu terénu a vrátí data z testovací sady a dostupná Exif data.

Exchangeable image file format (zkráceně Exif) je formát metadat, tedy konkrétně specifikace pro tento formát. Metadata jsou vkládána do souborů pomocí digitálních fotoaparátů a mohou sloužit pro analýzu snímků, získání příčiny jejich špatné kvality, a nebo určení jiných specifických parametrů. Formátů podporujících vkládání metadat bohužel není mnoho, za zmínku stojí především formáty typu JPEG, TIFF a RIFF WAVE. Bohužel stále zcela chybí podpora formátu PNG. Datová struktura je odvozena od souborového formátu TIFF (viz obrázek 4.10)[18].

⁴Více informací o objektu `sessionStorage`: http://www.w3schools.com/html/html5_webstorage.asp.

V souboru `get_data.php` nejprve dojde k připojení k databázi, následně se z tabulky `photoparam` dotazem `"SELECT * FROM 'photoparam' WHERE 'name' = '$name'"`, kde proměnná `$name` je název fotografie, získají informace dostupné z testovací sady. Následně se voláním funkce `exif_read_data($path, 0, true)`; získají dostupná EXIF data. Proměnná `$path` označuje cestu, kde je fyzicky fotografie uložena, hodnota 0 specifikuje, že nás budou zajímat výsledky ze všech sekcí a hodnotou `true` říkáme, že budeme prohledávat i sekci s názvem `thumbnail`. Výsledkem jsou informace o rozměrech fotografie, ohnisková vzdálenost čočky a GPS souřadnice místa, kde byl snímek pořízen. Posledním získávaným údajem je odhad nadmořské výšky (viz sekce 4.2.6). Ten je ovšem získáván když je hodnota globální poroměnné `modul_digital_model` rovna `true`.

Pomocí získaných metadat jsme schopni uživateli zprostředkovat informace o značce a modelu fotoaparátu, GPS souřadnice místa, kde byla fotografie pořízena, typ senzoru, nebo ohniskovou vzdálenost čočky fotoaparátu. Informace o GPS souřadnicích, pokud jsou dostupné, slouží především pro porovnání výsledků geolokizačního systému. Naopak ohnisková vzdálenost slouží pro vypočítání hodnoty zorného úhlu. Další důležitou informací je získání rozměrů nahrané fotografie, pomocí nichž je později proveden výpočet velikosti náhledu umístěného do digitálního modelu terénu.

Tag No.	Tag Name	Format	CompoNo	Desc.
0x010c	ImageDescription	ascii string		Describes image
0x010f	Make	ascii string		Shows manufacturer of digicam
0x0110	Model	ascii string		Shows model number of digicam
0x0112	Orientation	unsigned short	1	The orientation of the camera relative to the scene, when the image was captured. The start point of stored data is, '1' means upper left, '3' lower right, '6' upper right, '8' lower left, '9' undefined.
0x011a	XResolution	unsigned rational	1	Display/Print resolution of image. Large number of digicam uses 1/72inch, but it has no mean because personal computer doesn't use this value to display/print out.
0x011b	YResolution	unsigned rational	1	
0x0128	ResolutionUnit	unsigned short	1	Unit of XResolution(0x011a)/YResolution(0x011b). '1' means no-unit, '2' means inch, '3' means centimeter.
0x0131	Software	ascii string		Shows firmware(internal software of digicam) version number.
0x0132	DateTime	ascii string	20	Date/Time of image was last modified. Data format is "YYYY:MM:DD HH:MM:SS"+0x00, total 20bytes. In usual, it has the same value of DateTimeOriginal(0x9003)
0x013c	WhitePoint	unsigned rational	2	Defines chromaticity of white point of the image. If the image uses CIE Standard Illumination D65(known as international standard of 'daylight'), the values are '3127/10000,3290/10000'.
0x013f	PrimaryChromaticities	unsigned rational	6	Defines chromaticity of the primaries of the image. If the image uses CCIR Recommendation 709 primaries, values are '640/1000,330/1000,300/1000,600/1000,150/1000,0/1000'.
0x0211	YCbCrCoefficients	unsigned rational	3	When image format is YCbCr, this value shows a constant to translate it to RGB format. In usual, values are '0.299/0.587/0.114'.
0x0213	YCbCrPositioning	unsigned short	1	When image format is YCbCr and uses 'Subsampling'(cropping of chroma data, all the digicam do that), defines the chroma sample point of subsampling pixel array. '1' means the center of pixel array, '2' means the datum point.
0x0214	ReferenceBlackWhite	unsigned rational	6	Shows reference value of black point/white point. In case of YCbCr format, first 2 show black/white of Y, next 2 are Cb, last 2 are Cr. In case of RGB format, first 2 show black/white of R, next 2 are G, last 2 are B.
0x8298	Copyright	ascii string		Shows copyright information
0x8769	ExifOffset	unsigned long	1	Offset to Exif Sub IFD

Obrázek 4.10: Datová struktura formátu TIFF. První sloupec označuje číslo tagu, druhý jméno tagu, třetí formát uložení, v dalším se nachází počet komponent a poslední sloupec obsahuje popis tagu[18].

- GPS souřadnice – `$exif['GPS']['GPSLongitude'], $exif['GPS']['GPSLatitude'],`
- ohnisková vzdálenost – `$exif['EXIF']['FocalLength'],`
- model fotoaparátu – `$exif['IFD0']['Model'],`
- rozměry fotografie – `$exif['COMPUTED']['Width'], $exif['COMPUTED']['Height'].`

Výsledné informace jsou zasílány jako pole zpět klientské části ke zpracování ve formátu JSON metodou `json_encode()`.

4.2.4 Uložení dat

Jelikož získávání dat může být především z důvodu odhadu nadmořské výšky náročná operace, je nežádoucí, aby uživatel při aktualizaci stránky čekal na výsledky delší dobu než je nutné. Z tohoto důvodu je navrženo ukládání získaných dat do předpřipravené `photo` tabulky v databázi. Tento způsob je velice výhodný v tom, že když dojde k opětovnému načtení stránky, nemusíme vypočítávat žádná data, stačí se pouze zeptat databáze na informace, které jsou v ní obsaženy.

Ovšem v tomto případě je nezbytné rozlišovat, jestli se používá pouze modul webové mapy nebo modul digitálního modelu terénu a nebo oba moduly zároveň. Oba moduly totiž nepoužívají stejná data, například pokud používáme pouze vizualizaci pomocí webové mapy, jistě nevyužijeme informaci o rozměrech fotografie a zbytečná bude i informace o odhadované nadmořské výšce. Dále je potřeba rozlišit, z jaké karty prohlížeče jsou data ukládána, k tomu slouží automaticky generovaný unikátní identifikátor (viz sekce 4.2.2), který ji reprezentuje a který se do tabulky ukládá s každou nahranou fotografií.

K ukládání dat dochází za pomoci technologie AJAX, kterou jsou na stranu serveru, do souboru `server.php`, přenesena data k uložení. Ta se pomocí dotazu `INSERT` jazyka SQL uloží do tabulky `photos`. Po aktualizaci stránky funkce `upload_files_refresh()` zkontroluje podle unikátního identifikátoru, jestli byly už nějaké fotografie uloženy. Pokud ano, získáváme informace o fotografiích a datech s nimi souvisejícími.

4.2.5 Zorný úhel

Zorný úhel je jednou ze základních informací, která je uživateli poskytována. Je to důležitá informace při vytváření webové mapy, kde na základě vypočítaného zorného úhlu vytváříme polygon, který právě tento zorný úhel reprezentuje. Druhé využití zorného úhlu se používá pro určení velikosti fotografie umístěné v digitálním modelu terénu.

Získávání velikosti horizontálního zorného pole (zorný úhel) při umístění fotografie do digitálního modelu terénu probíhá za pomoci existujícího geolokalizačního systému. Systém zasílá získané informace ve formátu JSON. Jednou z informací je také položka s názvem `hfov`, která nese datový typ `double` ("`hfov`": `<double>`). Ze strany serveru je informace zaslána pomocí metody `json_encode()` a na straně klienta se naopak informace získá voláním metody `parse()` objektu JSON.

Při vytváření webové mapy se pracuje s hodnotou vertikálního zorného pole, které je nezbytné vypočítat na straně serveru. Pro výpočet této hodnoty je nezbytné získat potřebné informace, kterými jsou ohnisková vzdálenost a úhlopříčka senzoru fotoaparátu. Ohnisková vzdálenost se získá pomocí extrakce metadat formátu Exif, která jsou přibalena k digitální fotografii. Pro výpočet úhlopříčky senzoru fotoaparátu je využita databáze fotoaparátu. V té jsou uloženy rozměry vybraných fotoaparátů. Nejprve je tedy, opět z přibalených metadat, důležité zjistit model fotoaparátu, jímž byla fotografie pořízena, poté se dotazem `SELECT` jazyka MySQL zeptáme, zda-li se v databázi hledaný fotoaparát nachází. Při nalezení fotoaparátu aplikujeme na získané strany (a , b) Pythagorovu větu, díky které získáme výslednou úhlopříčku senzoru (c) fotoaparátu[19].

$$c^2 = a^2 + b^2 \tag{4.6}$$

Po aplikaci potřebných informací získáváme velikost zorného pole, zorný úhel. Na stranu klienta je hodnota předávána metodou `json_encode()`, kde je zpracována k použití za pomoci metody `parse()` objektu JSON.

photos	
* <u>id</u>	int(50)
◦session	varchar(200)
◦path	varchar(200)
◦file	varchar(200)
◦name	varchar(100)
◦latitude	float
◦longitude	float
◦e_latitude	float
◦e_longitude	float
◦camera	varchar(100)
◦altitude	float
◦pitch	float
◦roll	float
◦yaw	float
◦hfov	float
◦fov	varchar(200)
◦height	float
◦width	float

Obrázek 4.11: Tabulka photo sloužící pro uložení nahraných fotografií a získaných dat. `id` představuje primární klíč, `session` identifikátor záložky prohlížeče, `path` cestu k syntetickému panoramatu, `file` cestu k fotografii, `name` je název fotografie, `latitude` a `longitude` souřadnice z testovací sady, `e_latitude` a `e_longitude` souřadnice získané pomocí nástroje exif, `camera` udává název a model fotoaparátu, `altitude` odhad nadmořské výšky, `pitch`, `roll` a `yaw` upřesňují pozici fotografie v digitálním modelu terénu, `hfov` a `fov` je horizontální resp. vertikální zorný úhel a `height` a `width` rozměry fotografie.

4.2.6 Odhad nadmořské výšky

Odhad nadmořské výšky je vytvořen za pomoci zakomponování bakalářské práce Ing. Jana Vašíčka⁵. Zdrojové kódy projektu jsou napsány v jazyce Python s využitím frameworku Caffee. Autor vytvořil skript, pomocí kterého lze vyhodnocovat odhad nadmořské výšky. Ve webové aplikaci, na straně serveru, se za využití metody `chdir()` přesuneme do adresáře se zdrojovými kódy této aplikace a metoda `shell_exec()` spustí připravený skript, který dostává na vstup tři parametry. První je značen `-i` a identifikuje cestu k fotografii, nad kterou se bude odhad provádět, druhý parametr `-p` s hodnotou `basic` definuje způsob predikce, a poslední parametr `-o` s hodnotou `0` určuje, že chceme získat pouze odhad nadmořské výšky bez jiných dodatečných informací. Metoda `json_encode()` odesílá vypočtenou hodnotu nadmořské výšky na stranu klienta[19].

⁵Odhad nadmořské výšky Ing. Jana Vašíčka:
https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=115036.

Kapitola 5

Testování aplikace

Testování aplikace bylo uskutečněno ve dvou fázích. Nejprve proběhl test na uživatelích, kdy pěti vybraným osobám byly poskytnuty jednotlivé webové nástroje. Ke každému testu byly uživateli přiřazeny tři úkoly, které musel splnit bez předchozí znalosti nástroje. Hlavním cílem bylo zjistit do jaké míry je ovládání nástroje intuitivní a také pozorovat dobu, která je potřebná k orientaci v prostředí.

Druhou fází testování byla kompatibilita nástrojů v různých prohlížečích spuštěných na různých platformách. Pro toto testování byla využita služba CrossBrowserTesting¹, která umožní poskytnutou webovou stránku, přes její URL, spustit v různých operačních systémech, ve kterých si lze navíc zvolit z několika webových prohlížečů. Testování proběhlo na operačních systémech Windows 8, Windows 10, Ubuntu 14.04 LTS a Mac OSX 10.8. Na každém operačním systému byly testovány webové prohlížeče Google Chrome verze 49, Opera verze 36, Safari 5.1, Firefox 45 a Internet Explorer 10.

Dále byly jednotlivé nástroje testovány na správnost HTML5 a CSS syntaxe pomocí validátoru², který vydává mezinárodní konsorcium W3C.

5.1 Webová mapa

Položené otázky uživatelům, kteří se podíleli na testování webové mapy:

- Přepněte zobrazení míst z bodů na náhledy fotografií.
- Změňte barvu zorného úhlu naznačeného polygonem.
- Změňte polohu některého objektu vyobrazeného na mapě.

5.2 Digitální model terénu

Položené otázky uživatelům, kteří se podíleli na testování digitálního modelu terénu:

- Rotujte fotografii vyobrazenou v panoramatu.
- Přiblížte fotografii vyobrazenou v panoramatu.
- Změňte polohu fotografie vyobrazené v panoramatu.

¹Služba CrossBrowserTesting: <https://app.crossbrowsertesting.com/test-center>.

²Validátor konsorcia W3C: <https://validator.w3.org/>.

5.3 Nahrávání fotografií

Položené otázky uživatelům, kteří se podíleli na testování elementu, sloužícího pro získání fotografií od uživatele:

- Nahrajte fotografii do aplikace jejím přetažením.
- Nahrajte více fotografií do aplikace jejich přetažením.
- Nahrajte fotografii pomocí tlačítka pro nahrání fotografie.

Kapitola 6

Ukázkové aplikace

Pro ukázkou možnosti použití webových nástrojů byly vyvinuty čtyři demonstrační aplikace, kdy každá reprezentuje použití některého z modulů (webová mapa, digitální model terénu) nebo obou modulů zároveň. Dále je zde možnost vidět, jak moduly využít společně s uploaderem pro získávání fotografií od uživatele, a nebo bez uploaderu, kdy do každého modulu lze přidat staticky fotografie, které budou zracovány. Demonstrační aplikace také slouží pro ukázkou možnosti nastavní jednotlivých modulů a přiblížení práce s nimi.

6.1 Webová mapa s nahráváním fotografií

První ukázkou je aplikace, která pomocí prvku pro nahrávání fotografií získává vstup od uživatele, kterým jsou jeho fotografie. Fotografie jsou poté uloženy a zpracovány. Celý výsledek je následně reprezentován na webové mapě, kde jsou uživateli zobrazovány informace o jeho nahraných fotografiích.

Při počátečním nastavení je důležité aktivovat modul webové mapy, vytvořit webovou mapu a nastavit jí hranice, která definuje velikost základní vrstvy. Dále jsou zde přidány tlačítka pro změnu barvy polygonu reprezentujícího zorný úhel na modrou nebo zelenou a změny náhledu fotografie. Pomocí globálních proměnných nastavíme, že budeme do mapy přidávat body, polygon se bude po nahrání fotografie zobrazovat modrou barvou, dále umožníme elementy mapy přemísťovat a koeficient na hodnotu 3, pomocí kterého definujeme velikost zobrazeného polygonu.

```
// Nastaveni prostredi
modul_map = "true";
added_type = 'markers';
polygon_color = 'blue';
polygon_size = 3;
dragging = "true";
var max_val_bounds = L.latLngBounds( L.latLng(-90, 180),
L.latLng(90, -180));
create_map( [0, 0], 18, 2, max_val_bounds, 'Topographic' );
button_to_map( 'id-1', 'topleft', 'modra', 'NO ICON', 'blue' );
button_to_map( 'id-2', 'topleft', 'zelená', 'NO ICON', 'green' );
var icon = ['fa-map-marker', 'fa-camera'];
var title = ['zobrazit markers', 'zobrazit fotografie'];
toggle_button_to_map( icon, title );
```

6.2 Digitální model terénu

Druhou připravenou ukázkou je aplikace, která zobrazí pouze modul digitálního modelu terénu staticky přidaných fotografií. Není tedy možnost, aby uživatel přidával nové fotografie, jelikož se na stránce nevyskytuje prvek pro nahrání fotografií.

Nejprve bylo zapotřebí aktivovat modul digitálního modelu, poté voláním funkce `schovat` webovou mapu a prvek pro nahrávání fotografií, následně už jsen stačilo zavolat funkci, které je předána cesta k fotografii a dojde k jejímu zpracování.

```
// Nastaveni prostredi
modul_digital_model = "true ";
hide_web_map ();
hide_uploading_files ();
add_items_to_digital_model ("sp_dom1.jpg ");
add_items_to_digital_model ("scha_2.jpg ");
```

6.3 Webová mapa a digitální model terénu

Třetí ukázka představuje opět práci se staticky přidávanými fotografiemi, v tomto případě se ale budou získané informace prezentovat jak ve webové mapě tak i pomocí digitálního modelu terénu.

Nejprve musíme nastavit oba dostupné moduly jako používané, nezbytné je také schovat prvek pro nahrávání fotografií. Pomocí globálních proměnných nastavíme, že na mapě se budou zobrazovat náhledy fotografií, barva polygonu bude zelená a elementy na mapě nebude možno přesouvat. Koeficient velikosti polygonu bude ponechán na implicitní hodnotě. Funkce `add_items_to_digital_model()` potom zajistí jak vytvoření digitálního modelu tak i přidání prvků do webové mapy.

```
// Nastaveni prostredi
modul_map = "true ";
modul_digital_model = "true ";
hide_uploading_files ();
added_type = 'photos ';
polygon_color = 'green ';
dragging = "false ";
var max_val_bounds = L.latLngBounds( L.latLng(-90, 180),
L.latLng(90, -180));
create_map( [0, 0], 18, 2, max_val_bounds, 'Topographic' );
add_items_to_digital_model ("sp_dom1.jpg ");
add_items_to_digital_model ("scha_2.jpg ");
```

6.4 Webová mapa a digitální model terénu s nahráváním fotografií

Poslední, čtvrtou, ukázkou je aplikace využívající modulu webové mapy a digitálního panorama, která zpracovává fotografie nahrané uživatelem.

Opět je potřeba nastavit modul webové mapy a digitálního panorama jako aktivní, globální proměnné pro nastavení prostředí webové mapy budou ponechány jako implicitní. Zbývá tedy akorát vytvořit webovou mapu.

```
// Nastaveni prostredi
modul_map = "true";
modul_digital_model = "true";
var max_val_bounds = L.latLngBounds( L.latLng(-90, 180),
L.latLng(90, -180));
create_map( [0, 0], 18, 2, max_val_bounds, 'Topographic' );
```

Kapitola 7

Závěr

Cílem této bakalářské práce bylo se seznámit s dostupnými technologiemi pro tvorbu webových aplikací, návrh a implementace sady nástrojů reprezentujících výsledky geolokalizačního systému. Vizualizační nástroj byl vyvinut za použití webové mapy, která velice efektivně sdružuje a prezentuje získané informace. Nástroje pro manuální registraci obrazu do digitálního modelu terénu byly implementovány za pomoci technologie WebGL, která zajišťuje interakci s uživateli a poskytuje možnost manuálního upřesnění reprezentovaných výsledků. Pro demonstraci využití nástrojů byly implementovány ukázkové webové aplikace pracující nad testovací sadou fotografií poskytnutou panem docentem Martinem Čadíkem.

Výsledná sada nástrojů byla testována na uživateli různých věkových kategorií, kteří hodnotili intuitivnost a způsob práce s nástroji. Dále byl proveden test chování ve vybrané škále nejpoužívanějších webových prohlížečů a operačních systémů. Pro prezentaci způsobu použití implementovaných nástrojů byly vytvořeny testovací aplikace, které zahrnují veškerou funkčnost a znázorňují možnosti nastavení prostředí.

Pro budoucí vývoj bych navrhol zaměřit se především na optimalizaci ovládání a přesnost výsledků digitálního modelu terénu, kdy velikost nahrané fotografie ne vždy odpovídá chtěnému výsledku. Dále bych zvážil způsob získávání odhadu nadmořské výšky, která do jisté míry zpomaluje proces zpracovávání fotografií na straně serveru. Do budoucna by bylo velice zajímavé zahrnout souběžně psanou bakalářskou práci autora Jakuba Krbce, který se zabývá vylepšením obrazu fotografie za pomoci hloubkových map.

Literatura

- [1] Cacek, B. P.: Tvorba panoramatických fotografií. 2015.
- [2] Cesar Otero, Rob Larsen: *Professional jQuery*. John Wiley & Sons, Inc., 2012, ISBN 978-1-118-02668-7.
- [3] David Procházka: *PHP 6 začínáme programovat*. Grada Publishing, a.s., 2012, ISBN 978-80-247-3899-4.
- [4] Doc. Ing., Martin Čadík, Ph.D.: *LOCATE – vizuální lokalizace v přírodě*. online, [Online; navštíveno 3.4.2016].
URL http://cadik.posvete.cz/locate/locate_cz.pdf
- [5] Doc. Ing. Václav Čada, C.: *Geoid*. Západočeská univerzita, Fakulta aplikovaných věd, Katedra matematiky, [Online; navštíveno 4.4.2016].
URL <http://gis.zcu.cz/studium/gen1/html/ch02.html#id299102>
- [6] Doc. Ing. Václav Čada, C.: *Souřadnicový systém WGS 84*. Západočeská univerzita, Fakulta aplikovaných věd, Katedra matematiky, [Online; navštíveno 4.4.2016].
URL <http://gis.zcu.cz/studium/gen1/html/ch02s03.html#id301837>
- [7] Elizabeth Castro, Bruce Hyslop: *HTML5 a CSS3*. Albatros Media, a.s., 2012, ISBN 978-80-251-3733-8.
- [8] Eyal Vardi: *HTML5*. online, [Online; navštíveno 18.4.2016].
URL <https://eyalvardi.wordpress.com/resources/html-5-course-resources/>
- [9] Hořák, R.: Webový informační systém turistického průvodce. 2010.
- [10] Hráček, F.: *Mapy Google v češtině*. online, [Online; navštíveno 15.4.2016].
URL <http://google-cz.blogspot.cz/2008/05/mapy-google-v-etin.html>
- [11] Javůrek, M.: *Mapy.cz pro Android nově nabízejí offline mapy celé Evropy*. online, [Online; navštíveno 27.4.2016].
URL <http://smartmania.cz/mapy-cz-offline-evropa-android-seznam-12316/>
- [12] Kolor: *Understanding Projecting Modes*. online, [Online; navštíveno 8.4.2016].
URL http://www.kolor.com/wiki-en/action/view/Understanding_Projecting_Modes
- [13] Krpelík, V.: *Krajinářská fotografie - Panorama I.část*. online, [Online; navštíveno 24.4.2016].
URL <http://www.fotoradce.cz/krajinarska-fotografie-panorama-i-cast>

- [14] Ondřej Žára: *Vytváříme Hello World pro WebGL*. online, [Online; navštíveno 20.4.2016].
URL <https://www.zdrojak.cz/clanky/vytvarime-hello-world-pro-webgl/>
- [15] Openstreetmap: *About*. online, [Online; navštíveno 29.4.2016].
URL <http://www.openstreetmap.org/about>
- [16] The jQuery Foundation: *What is jQuery?* online, [Online; navštíveno 20.4.2016].
URL <https://jquery.com/>
- [17] Tomáš Hruška, R. B.: *Internetové aplikace (WAP) II*. 2012: s. 181–182.
- [18] TsuruZoh Tachibanaya: *Description of Exif file format*. online, [Online; navštíveno 22.4.2016].
URL <http://www.media.mit.edu/pia/Research/deepview/exif.html>
- [19] Vašíček, B. J.: *Automatický odhad nadmořské výšky z obrazu*. 2015.
- [20] Vladimir Agafonkin: *Overview*. online, [Online; navštíveno 16.3.2016].
URL <http://leafletjs.com/>
- [21] Wikipedia: *Geoid*. online, [Online; navštíveno 27.4.2016].
URL <https://cs.wikipedia.org/wiki/Geoid>
- [22] Wikipedia: *Google Maps*. online, [Online; navštíveno 24.4.2016].
URL https://en.wikipedia.org/wiki/Google_Maps

Přílohy

Seznam příloh

A Obsah DVD

40

Příloha A

Obsah DVD

- adresář server/ - zdrojové kódy serverové části aplikace,
- adresář database/ - syntetická panoramata k fotografiím testovací sady,
- adresář test_photos/ - použitelné fotografie testovací sady,
- adresář upload/ - adresář pro uložení nahraných fotografií,
- adresář mysql/ - zdrojové kódy tabulek databáze připravené k importu,
- adresář elevation/ - odhad nadmořské výšky (diplomová práce ing. Jana Vašíčka),
- soubor tools.js - zdrojový kód sady webových nástrojů,
- soubor program.fs - fragment shader,
- soubor program.vs - vertex shader,
- soubor example1.html - zdrojový kód první ukázkové aplikace,
- soubor example2.html - zdrojový kód druhé ukázkové aplikace,
- soubor example3.html - zdrojový kód třetí ukázkové aplikace,
- soubor example4.html - zdrojový kód čtvrté ukázkové aplikace,
- soubor example_style.css - základní stylový předpis,
- soubor README - zprovoznění práce,
- soubor xbecka02.pdf - text práce ve formátu pdf,
- soubor plakat.pdf - plakát ve formátu pdf.