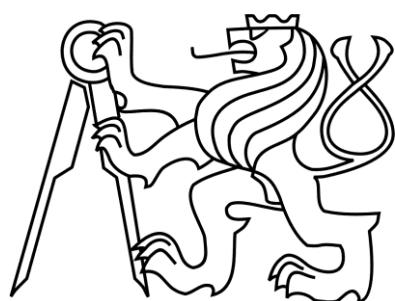
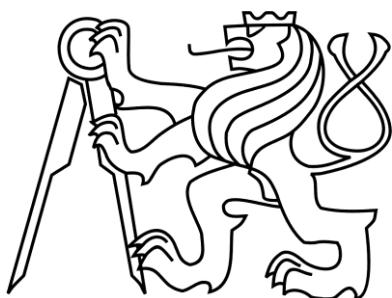


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STAVEBNÍ



BAKALÁŘSKÁ PRÁCE

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STAVEBNÍ
STUDIJNÍ PROGRAM GEODÉZIE A KARTOGRAFIE
OBOR GEOINFORMATIKA



BAKALÁŘSKÁ PRÁCE

Využití Google Maps API pro mapovou aplikaci sportovních aktivit v Orlických horách

Vedoucí práce: Doc. Ing. Jiří Cajthaml, Ph.D.
Katedra geomatiky



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

studijní program: Geodézie a kartografie

studijní obor: Geoinformatika

akademický rok: 2013/2014

Jméno a příjmení studenta: Jan Klíma

Zadávající katedra: K155 - Katedra geomatiky

Vedoucí bakalářské práce: Doc. Ing. Jiří Cajthaml, Ph.D.

Název bakalářské práce: Využití Google Maps API pro mapovou aplikaci sportovních aktivit v Orlických horách

Název bakalářské práce v anglickém jazyce: Usage of Google Maps API for web map application of sport activities in Orlické hory

Rámcový obsah bakalářské práce: Hlavním tématem této bakalářské práce je tvorba webové mapové aplikace s využitím Google Maps API a Google Earth API, které umožňují překrývat vlastní data nad mapovými podklady Googlu. Cílem je vytvoření interaktivní mapové aplikace, která by sloužila pro sportovní aktivity v Orlických horách.

Datum zadání bakalářské práce: 17.2.2014 Termín odevzdání: **16.5.2014**
(vyplňte poslední den výuky příslušného semestru)

Pokud student neodevzdal bakalářskou práci v určeném termínu, tuto skutečnost předem písemně zdůvodnil a omluva byla děkanem uznána, stanoví děkan studentovi náhradní termín odevzdání bakalářské práce. Pokud se však student rádně neomluvil nebo omluva nebyla děkanem uznána, může si student zapsat bakalářskou práci podruhé. Studentovi, který při opakováném zápisu bakalářskou práci neodevzdal v určeném termínu a tuto skutečnost rádně neomluvil nebo omluva nebyla děkanem uznána, se ukončuje studium podle § 56 zákona o VŠ č. 111/1998. (SZŘ ČVUT čl. 21, odst. 4)

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

.....
vedoucí bakalářské práce

.....
vedoucí katedry

Zadání bakalářské práce převzal dne: 17.2.2014

.....
student

Formulář nutno vyhotovit ve 3 výtiscích – 1x katedra, 1x student, 1x studijní odd. (zašle katedra)

Nejpozději do konce 2. týdne výuky v semestru odešle katedra 1 kopii zadání BP na studijní oddělení a provede zápis údajů týkajících se BP do databáze KOS.

Anotace

Tato bakalářská práce se zabývá návrhem a realizací webové mapové aplikace, která by v budoucnu mohla sloužit pro plánování sportovních a dalších volnočasových aktivit v Orlických horách. Úvodní část je zaměřena na stručné seznámení se společností Google a jejími mapovými službami Google Maps a Google Earth. V druhé části se předložená práce věnuje podrobnějšímu popisu rozhraní Google Maps JavaScript API v3, pomocí kterého byl vytvořen základ aplikace. Následující část zahrnuje praktické příklady využití rozhraní Maps API a tvorbu grafické podoby aplikace s použitím programovacích jazyků HTML, CSS a JavaScript. Součástí práce je i kapitola popisující možnosti aplikace z pohledu běžného uživatele.

Klíčová slova

Google Maps API, Google Earth API, webová mapová aplikace, interaktivní mapa, mashup

Annotation

This bachelor thesis describes the draft and implementation of web mapping application, which could be used in the future for planning of sports and other leisure activities in the Eagle Mountains. The first part is focused on the brief introduction of Google and its mapping services as Google Maps and Google Earth. In the second part, this thesis deals with a more detailed description of the Google Maps JavaScript API v3, by which the base of application was created. The following section includes practical examples of the used Maps API and the creation of graphic design applications, by using programming languages HTML, CSS and JavaScript. The paper also includes a chapter describing the application possibilities from the perspective of an average user.

Key words

Google Maps API, Google Earth API, web mapping application, interactive map, mashup

Prohlášení

Prohlašuji, že jsem předloženou bakalářskou práci na téma „Využití Google Maps pro mapovou aplikaci sportovních aktivit v Orlických horách“ vypracoval samostatně pod vedením Doc. Ing. Jiřího Cajthamla, Ph.D. Použitou literaturu a podkladové materiály uvádím v seznamu zdrojů na konci tohoto dokumentu.

V Praze dne

.....

(podpis autora)

Poděkování

Na tomto místě bych chtěl poděkovat vedoucímu bakalářské práce Doc. Ing. Jiřímu Cajthamlovi, Ph.D. za odborné vedení práce a za podporu a trpělivost při jejím vytváření. Děkuji také Ing. Janu Langrovi ze společnosti T-Mapy spol. s.r.o. a vydavatelství SHO-Cart za poskytnutá data. Rád bych poděkoval také své rodině, všem blízkým a přátelům, kteří mě při vytváření této práce podpořili, a bez jejichž pomoci by nebylo možné práci dokončit.

Obsah

Úvod	1
1 Google	3
1.1 Google Maps a Google Earth	5
1.1.1 Společná historie	7
1.1.2 Zdroje dat	8
1.1.3 Kartografické zobrazení	10
1.1.4 Souřadnicový systém	11
2 Využití Google Maps API v3	12
2.1 Úvod do JavaScriptu	13
2.2 Vložení mapy na stránku	14
2.2.1 Rozbor kódu	16
2.2.2 Adresářová struktura	18
2.2.3 Mobilní zařízení	20
2.3 Nastavení uživatelského rozhraní	20
2.3.1 Pozice objektu na mapě	23
2.3.2 Metody třídy Map	24
2.4 Vlastní styl mapy	26
2.5 Přidání překryvné vrstvy	28
2.5.1 Vrstva KML a GeoRSS	28
2.5.2 Dynamické tabulky	28
2.5.3 Další vrstvy	29
3 Realizace aplikace	31
3.1 Specifikace a historie HTML	31
3.2 HTML5 a Geolocation API	32
3.2.1 Zjištění polohy	33
3.2.2 Aplikace lokalizace	34
3.2.3 Použití s Google Maps API	35
3.3 Vzhled stránky	37
3.4 JavaScript a knihovna jQuery	38
3.4.1 Použití knihovny jQuery	38
3.4.2 Kalendar plugin	39
3.5 Příprava a import turistických tras	40
3.5.1 Úprava dat	40

3.5.2	Publikační služby ArcGIS Serveru a Google Maps API	41
3.6	Virtuální prohlídka a Google Earth API	42
3.6.1	Vytvoření okna s prohlídkou	43
3.6.2	Zobrazení prohlídek v mapě	44
3.7	Testování aplikace	46
4	Aplikace z pohledu běžného uživatele	47
4.1	Seznámení se s uživatelským rozhraním	47
4.2	Sledování polohy	48
4.3	Vyhledávací pole	50
4.4	Přepínání vrstev	50
4.4.1	Virtuální prohlídky	52
4.5	Kalendář akcí	53
Závěr		54
Seznam použité literatury		55
A Elektronické přílohy		58

Seznam obrázků

1.1	Služby poskytované Googlem	4
1.2	Desktopová aplikace Google Earth	6
1.3	Online aplikace Google maps	6
1.4	Velitelství NATO v Nizozemí	9
1.5	Aktualizace snímků ze dne 21.1.2014	9
1.6	Mercatorovo zobrazení	10
1.7	Souřadnice mapových dlaždic na úrovni 2	11
2.1	Praha - zoom level 0, 10 a 17	18
2.2	Příklad adresářové struktury	18
2.3	Základní mapa	20
2.4	Horizontal bar a Dropdown menu	22
2.5	ZoomControlStyle Large a Small	23
2.6	Styled Map Wizard - vlastní styl mapy	27
3.1	Povolení sledování polohy v prohlížeči Google Chrome	33
3.2	Schéma Geolocation API	35
3.3	Vizualizace polohy a její přesnosti	37
3.4	Transformace souřadnic - ArcGIS Project Tool	41
3.5	Posun u souběžných tras	42
3.6	Nástroj vývojáře Google Chrome	46
4.1	Hlavní okno aplikace	48
4.2	Sledování polohy	49
4.3	3D model Země	49
4.4	Vyhledávání míst a adres	50
4.5	Ukázka turistických a cyklistických tras	51
4.6	Lyžařské areály, běžkařské tratě, počasí a dopravní info nad zimní mapou	51
4.7	Google StreetView	51
4.8	Výběr virtuální prohlídky	52
4.9	Virtuální prohlídka	52
4.10	Kalendář akcí	53

Seznam tabulek

2.1	Rozšiřující API	12
2.2	Limity pro využití Google Maps JavaScript API	14
2.3	Nastavení ovládacích prvků	22
2.4	Metody pro změnu nastavení	25
2.5	Události	25
2.6	FeaturesTypes	26
2.7	ElementTypes	26

ÚVOD

Poslední desetiletí přineslo do světa internetu celou řadu novinek a nových technologických řešení, které značnou měrou ovlivnily další trend tohoto moderního odvětví. Dramatický vývoj zaznamenaly a stále zaznamenávají především webové aplikace a služby, které jsou schopny plně nahradit běžné programy instalované na lokálních počítačích. Na poli webových služeb působí mnoho společností, ale pravděpodobněji tou nejznámější je americká společnost Google. Tato společnost se proslavila internetovým vyhledavačem a později vlastní webovou mapovou aplikací Google Maps. Tyto internetové mapy si během krátké doby získaly oblibu mezi běžnými uživateli, ale také vývojáři, kteří pomocí veřejného rozhraní mohou vytvářet rozmanité mapové aplikace.

Mapovou aplikaci jsem se rozhodl vytvořit jednak proto, že mě samotné vytváření internetových map vždy zajímalo a také proto, že jsem se chtěl odlišit od bakalářských prací, které danou problematiku řeší spíše teoreticky.

Hlavním cílem bakalářské práce je seznámit se s rozhraním Google Maps JavaScript API a s jeho pomocí vytvořit webovou mapovou aplikaci, která by v budoucnu mohla sloužit pro plánování sportovních a dalších volnočasových aktivit v Orlických horách. Součástí aplikace by měla být síť značených pěších, cyklistických a běžkařských tras, ale také nejvýznamnější lyžařské areály v této lokalitě. Od podobných projektů by se tato aplikace měla odlišit vysokou mírou interaktivnosti a integrovanými virtuálními prohlídkami tras ve 3D.

První kapitola se bude věnovat krátkému seznámení se společností Google, její historii a také přehlednému souhrnu nejznámějších služeb. Podrobněji se zaměřím na mapové služby Google Maps a Google Earth. Hlavní důraz budu klást na původ podkladových map, použité kartografické zobrazení a souřadnicové systémy.

Ve druhé kapitole se pokusím objasnit relativně nový pojem API a jeho spojení s Google Maps. Dále popíši alespoň základní principy programovacího jazyka JavaScript, který knihovna Google Maps API využívá. Na praktickém příkladu předvedu jednoduché vložení mapy na stránku spolu s podrobným rozbořem zdrojového kódu. Pokusím se také vytvořit vlastní styl mapy.

Třetí kapitola se bude zabývat praktickou tvorbou webové aplikace s využitím programovacích jazyků HTML, CSS a knihovny jQuery. Velká část bude také patřit seznámení s novou specifikací, která dovoluje sledovat aktuální polohu zařízení. Zaměřím se na způsoby zjištění polohy z pohledu jejich přesnosti, ale také na začlenění této služby do vlastní aplikace. Dále popíši úpravu vektorových dat, které pro účely této bakalářské práce poskytlo kartografické vydavatelství SHOCart ve spolupráci se společností T-Mapy.

Ve čtvrté kapitole se budu věnovat popisu aplikace pro běžného uživatele, kde uživateli

poskytnu návod jak aplikaci ovládat a jak využít všechny její možnosti. Kromě základních funkcí pro práci s mapou by měla aplikace obsahovat funkce vyhledávání či sledování polohy.

Google

Americká společnost Google se sídlem v Silicon Valley je předním hráčem na trhu s informačními technologiemi. Od ostatních firem v Silicon Valley se liší především rychlostí svého růstu a vysokými ambicemi zorganizovat informace z celého světa a poskytovat je svým uživatelům.

Zakladatelé společnosti Sergey Brin a Larry Page, studenti postgraduálního studia na Stanfordské Universitě, začínají v roce 1996 pracovat na společném projektu webového vyhledavače s názvem BackRub, který se až do roku 1997 používá na Stanfordských serverech. Později tento vyhledavač přejmenovávají na Google. Název představuje vysoké ambice zakladatelů - vznikl z přepisu matematického označení „googol“, který označuje nesmírně veliké číslo znázorněné jako posloupnost jedničky a sto nul.

Smělých plánů studentů si všimne spoluzakladatel společnosti Sun Andy Bechtolsheim a vypisuje šek v hodnotě 100 000 USD společnosti Google Inc., který umožní společnosti raketový start. Společnost Google Inc. se 4. září 1998 registruje ve státě Kalifornie a o pár měsíců později již odpoví 10 tisíc dotazů denně. To neujde pozornosti investorů a odborných časopisů, jako je PC Magazine, který vyhledavač zařadí mezi 100 nejlepších webů. Google tak zažívá explozivní nárůst dotazů - již v roce 1999 odpoví více jak 3 miliony dotazů denně. [26]

Díky novým technologiím, nabízeným službám a vztuřujícímu počtu uživatelů se Google dokázal prosadit i přes velikou konkurenci a mohl si začít budovat dominantní pozici nejen na Americkém trhu. V současné době je Google celosvětově nejrozšířenějším vyhledavačem s mnohojazyčným uživatelským prostředím. To také dokazují souhrnné statistiky používanosti vyhledavačů za rok 2013, kde se vyhledavač Google umístil na prvním místě s velikou převahou 90%. Druhé a třetí místo obsadili vyhledavače Bing a Yahoo!, které mají shodně 3%. Zbylá 4% procenta připadají méně rozšířeným vyhledavačům, jako je například čínský vyhledavač Baidu a další. [25]

Služby poskytované Googlem

Během svých začátků se Google musel potýkat s konkurenčními společnostmi, nabízejícími mimo internetových vyhledavačů i jiné webové služby, včetně emailových

klientů. To samozřejmě způsobilo odchod návštěvníků z webových stránek Googlu. Mezi tyto společnosti patřili i takoví softwaroví velikáni jako Yahoo! nebo Microsoft. Google se tak odkloňuje od svého původního zaměření pouze na webové vyhledávání a začíná investovat peníze získané z cílených reklam do webových služeb, označovaných jako Software as a Service.

První takovou vlaštovkou bylo vytvoření emailového klienta, později pojmenovaného jako Gmail. Google přišel hned s několika novinkami. Podařilo se mu spojit algoritmy na vyhledávání cílených reklam v závislosti na textu emailové zprávy, k přehlednějšímu vyhledávání použil sílu svého vyhledavače a svým zákazníkům poskytl prostor 1 GB, s možností zakoupení větší kapacity. V současné době nabízí kapacitu 15 GB, která je však sdílena spolu s prostorem pro fotografie na sociální síti Google+ a dokumenty na Google Disk. [26]

Google Docs (Dokumenty Google) je další webová služba umožňující online/offline vytváření, editování a prohlížení dokumentů, tabulek nebo prezentací. Všechny tyto dokumenty jsou uloženy na serverech Googlu (tzv. cloud) a přístup k nim je možný z jakéhokoli zařízení, bez ohledu na operační systém. Nutné je pouze připojení k internetu a v případě editace podporovaný internetový prohlížeč. Výhodou je stálá aktuálnost softwaru, který není nutné instalovat do počítače, neboť běží na centralizovaném serveru. To dělá především vrásky na čele společnosti Microsoft a jejímu kancelářskému balíku Microsoft Office. [12]

Další rozvíjející službou jsou Google Maps (Mapy Google), poskytující mapové podklady každého koutu naší planety, a aplikace Google Earth, která nabízí souvislé zobrazení satelitních nebo leteckých snímků na virtuálním glóbu. Těmito službami se budu na následujících stránkách textu zabývat podrobněji.

Jak je vidět, Google je blízko naplnění svojí vize získat a zorganizovat všechny informace světa. Je však otázkou, jestli je to vůbec správné, aby jedna společnost operovala s takovým množstvím informací a znalostí o nás.



Obrázek 1.1: Služby poskytované Googlem

1.1 Google Maps a Google Earth

Google Maps a Google Earth jsou velmi podobné aplikace poskytující interaktivní mapové a satelitní snímky. Níže v této kapitole jsou popsány jednotlivé aplikace samostatně.

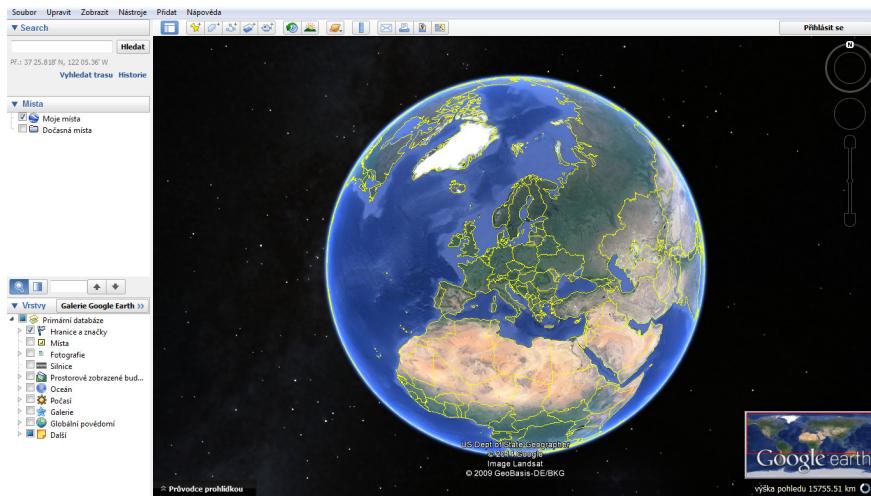
Aplikace Google Earth poskytuje prostřednictvím virtuálního glóbusu realistické zobrazení světa a možnost cestovat po celém světě, prohlížet si satelitní snímky ve vysokém rozlišení nebo se procházet městem s 3D budovami a terénem. Díky režimu Oceán v Google Earth je možné nahlédnout pod hladinu oceánů a prozkoumat reliéf mořského dna. Další režimy Obloha, Měsíc a Mars umožňují prozkoumat i místa mimo naši planetu.

Google Earth je distribuován v podobě desktopové aplikace dostupné pro nejrozšířenější operační systémy jako je Windows, Mac a Linux. Jsou poskytovány hned tři verze této aplikace. Základní verzi pro běžné uživatele je možné bezplatně stáhnout na oficiálních stránkách produktu. Pro profesionální použití je však vhodnější použít verzi Pro, která oproti bezplatné verzi nabízí speciální funkce navržené pro firemní uživatele. Třetí verzi Enterprise ocení hlavně firmy a organizace, které potřebují vizualizovat velké množství dat.

Dostupná je také verze aplikace pro mobilní zařízení používající platformu Android a iOS nebo rozšiřující plugin pro webové prohlížeče, díky kterému je možné zobrazit geografická data přímo na webových stránkách. Při přechodu na libovolnou stránku obsahující rozhraní Google Earth vám webový prohlížeč automaticky nabídne instalaci, případně aktualizaci tohoto pluginu. Plugin Google Earth je v současné době podporován operačními systémy a prohlížeči uvedenými níže. Data jsou beze změny převzata z oficiálních stránek podpory tohoto pluginu. [13]

1. Microsoft Windows (Windows XP, Windows Vista a Windows 7)
 - (a) Google Chrome 5.0 nebo novější
 - (b) Internet Explorer 7 nebo novější (32 bitů)
 - (c) Firefox 2.0 nebo novější
 - (d) Flock 1.0 nebo novější
2. Apple Mac OS X 10.5.0 a vyšší (pouze Intel)
 - (a) Google Chrome 5.0 nebo novější
 - (b) Safari 3.1 nebo novější
 - (c) Firefox 3.0 nebo novější

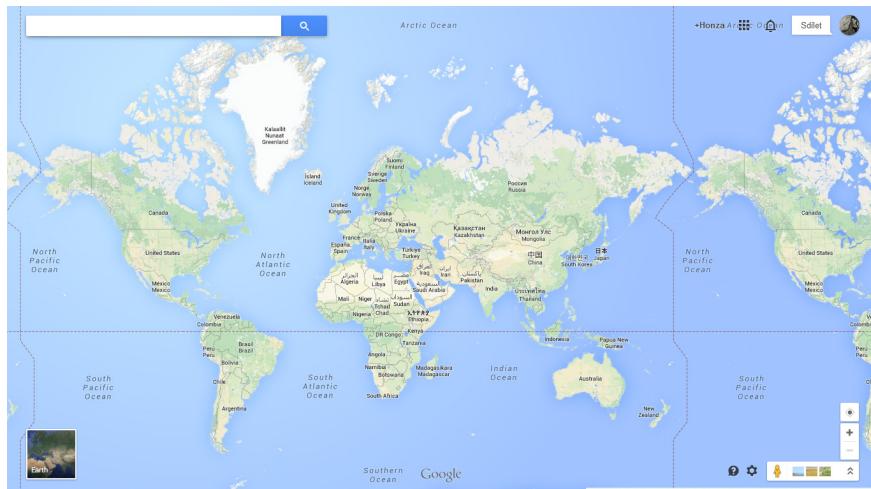
V mapách Google je možné vyhledávat, procházet se ulicemi a plánovat trasy po celém světě. Pro vyhledávání slouží vyhledávací pole, které umožní zobrazit na mapě adresy, obchody, restaurace a mnoho dalšího včetně podrobných informací. Pro některá



Obrázek 1.2: Desktopová aplikace Google Earth

větší města a oblasti světa je k dispozici mapová vrstva s informací o aktuální dopravní situaci, vrstva veřejné dopravy nebo vrstva s cyklistickými trasami. V České Republice jsou informace o dopravě dostupné pro některé hlavní silniční tahu a dálnice, zatímco informace a vyhledávání spojů ve veřejné dopravě jsou prozatím dostupné pouze v hlavním městě.

Pro použití Google Maps je nutné mít nainstalovaný pouze podporovaný internetový prohlížeč a dostatečně rychlé internetové připojení pro stahování mapových podkladů. Ty jsou pro úsporu velikosti datového přenosu uloženy v podobě mapových dlaždic pro každou úroveň přiblžení. Není tedy načítána celá mapa, ale pouze dlaždice, které jsou nutné pro zobrazení požadovaného území v závislosti na přiblžení. [14]



Obrázek 1.3: Online aplikace Google maps

1.1.1 Společná historie

Google musel reagovat především na technologický vývoj konkurenční společnosti Yahoo!, která svým uživatelům nabídla online mapové služby MapQuest. Proto Google vytvořil malou skupinu svých inženýrů, kteří měli vyvinout vlastní službu pro online poskytování map a vyhledávání cest, později známou jako Google Maps.

Nedaleko sídla společnosti v Silicon Valley vznikla v roce 2000 bez povšimnutí malá firma Keyhole, která měla do budoucnosti velmi smělé plány. Společnost Keyhole plánovala široké veřejnosti zpřístupnit online službu obsahující satelitní snímky získané od soukromých společností, které měly povolení vlády USA k vysílání družic, nesoucí zařízení pro pořizování fotografií Země ve vysokém rozlišení. Soubor těchto snímků pokrývající celou pevninu, která má rozlohu 149 000 000 km², zabral miliony gigabajtů. Služba měla být založena na předplatném a svým uživatelům měla nabídnout trojrozměrnou nepřerušovanou mozaiku satelitních snímků. To však bylo velmi náročné jak na výpočetní výkon osobních počítačů, tak na rychlosť internetového připojení, které v té době bylo příliš pomalé pro přenos takového objemu dat. Rozmach internetu a nová technologie společnosti Keyhole, která zobrazovala jeden snímek, ale zároveň načítala snímky sousedící, umožňovaly plynulý přechod mezi snímkami. První lidé, mimo zaměstnance Pentagonu a společnosti Keyhole, si tak mohli užít „přelet“ nad satelitními snímkami. Roční předplatné 20 000 Kč a investice do výkonných osobních počítačů však většinu uživatelů odradilo. Díky stále se zlepšujícímu výkonu osobních počítačů a reklamě televizní stanice CNN, která využívala software společnosti Keyhole Earth Viewer k přiblížení se k místu, kde se odehrávaly zprávy, služba přitáhla v roce 2003 zájem potencionálních zákazníků. Těmi však byly převážně realitní makléři, vládní agentury a inženýrské firmy. To však bylo stále daleko od myšlenky prodávat online službu veřejným zákazníkům.

Sergey Brin, spoluzakladatel společnosti Google, v roce 2004 narazil na malou společnost nabízející poutavý zázitek v podobě přeletu Země, která byla tvořena satelitními snímkami. Ihned se podělil s výkonným týmem společnosti a schůzí přerušil ukázkou, která všechny nadchla - na svém laptopu „sletěl“ z výšky až k domům každého z nich. Když téhož roku Google pozval zástupce společnosti Keyhole na návštěvu a prezentaci jejich služby, nechápali, co může mít internetový gigant společného s jejich službou. V Googlu však měli převratnou a vizionářskou myšlenku. Využít mapy a satelitní snímky jako nástroj pro zobrazení a uspořádání všech informací. Když o pár let později John Hanke, ředitel společnosti Keyhole, vzpomínal na ambiciózní plán Googlu, neubránil se komentáře: „*To nebyl jen závan čerstvého vzduchu, to byla přímo bomba! Mysleli jsme, že bud' blouzní, nebo jsou skuteční vizionáři.*“ K dokončení koupě společnosti Keyhole došlo 27. října 2004.

V únoru 2005 vydal Google prohlášení o spuštění online mapové aplikace Google Maps. První verze obsahovala pouze mapy ulic a možnost plánování cest, ale ještě nebyla integrována možnost zobrazit satelitní snímky společnosti Keyhole jako spodní vrstvu. Přinesla také novinku v podobě možnosti posouvat mapu kterýmkoliv směrem a zobrazit přilehlající oblasti bez nutnosti překreslení celé stránky. Technologii převzala z online

služby společnosti Keyhole - zároveň se stahovala i data nutná pro rozšíření právě zobrazené mapy.

Tehdejší internetové prohlížeče nebyly schopny zpracovávat obrázky s velmi vysokým rozlišením, které nabízely satelitní snímky. Proto si uživatelé museli nainstalovat speciální software Google Earth, který vznikl pouze přejmenováním aplikace EarthViewer a doplněním satelitních snímků pokryvající celý svět. Aby došlo k masovějšímu rozšíření, byl tento software poskytován pro domácí a komerční účely zdarma.

Další rozvoj Google Maps a Google Earth umožnily tzv. mashupy (hybridní webové aplikace), které umožňovaly zobrazení dat třetích stran nad mapovými podklady Googlu. Bylo tak možné objevit síť restaurací, benzínových stanic a zobrazit data, která původně neměla geografický charakter. Google později dal uživatelům i možnost opravovat chyby a přidávat nová místa. [26] [24] [12]

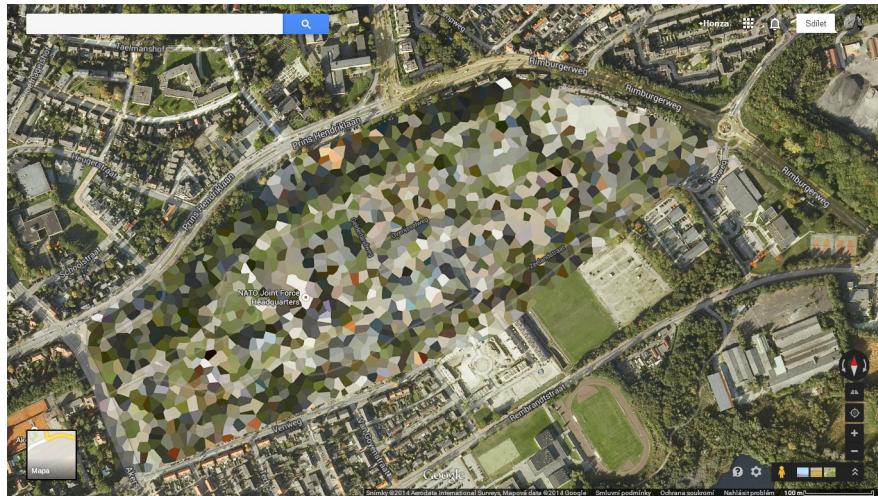
1.1.2 Zdroje dat

Mapy a satelitní snímky v aplikacích Google Maps a Google Earth jsou získávány z mnoha zdrojů a je nutné podotknout, že databáze obsahující vrstvu se satelitními snímky je společná pro obě tyto aplikace. V historii byly mapy pořizovány převážně od komerčních společností Tele Atlas a Navteq, zatímco satelitní snímky byly pořizovány jak od komerční společnosti DigitalGlobe, tak od americké vlády.

Původní satelitní snímky pořízené sately s nízkým rozlišením nebyly schopny jasné zobrazovat předměty menší než 60 cm. Původní snímky byly však postupně nahrazovány novými snímky ve středním rozlišení 15 metrů na pixel a snímky s vysokým rozlišením 70 cm na pixel. Tyto snímky pokrývaly pouze 1/3 světového obyvatelstva a byly již tak podrobné, že se na nich daly rozeznat jednotlivé domy a auta. Další rozvoj umožnil rozlišení 10 cm na pixel. Ještě podrobnější snímky přineslo využívání leteckého snímkování. Na rozdíl od satelitů, obíhajících Zemi ve výškách okolo 500 km, se letadla snímající zemský povrch pohybují ve výškách od 250 do 500 m. Spolu s rozvojem leteckého snímkování se do map dostaly i snímky pořízené z úhlu 45° . Pro takovéto fotografie se ustálilo označení „ptačí pohled“.

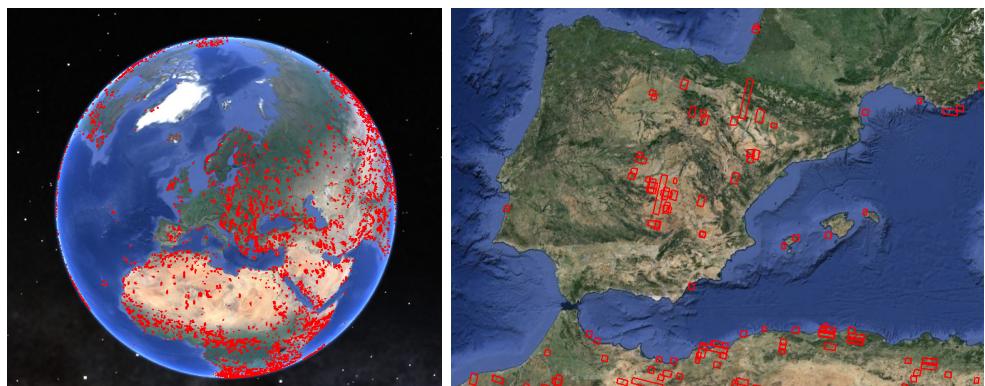
Google neustále shromažďuje snímky s nejvyšší možnou kvalitou a nejvyšším možným rozlišením, ale i přesto existuje několik oblastí, pro které nejsou data ve vysokém rozlišení k dispozici. Na druhou stranu snímky obsahující mraky nebo nesprávné barvy se mohou i přes vysoké rozlišení zdát rozmazané.

Se zvyšujícím se rozlišením snímků rostla přímou úměrou také obava veřejnosti a vládních organizací. Veřejnost měla obavy o své vlastní soukromí a vládní organizace měly strach z podrobnosti, s jakou byly zobrazovány vládní budovy a vojenská zařízení. Jejich obavy směřovaly především k hrozbe využití těchto snímků pro teroristické útoky. Proto jsou některé budovy či celé oblasti na satelitních snímcích zamaskovány tak, jak je ukázáno na obrázku 1.4, kde je zobrazena budova velitelství NATO v Nizozemí. [26]



Obrázek 1.4: Veleřadce NATO v Nizozemí

Databáze satelitních snímků není poskytována v aktuálním čase, ale aktualizuje se pravidelně dvakrát do měsíce, nejčastěji kolem šestého a dvacátého dne příslušného měsíce. V rámci každé aktualizace je obnovena pouze malá část světa. Spolu s ní je vydáván KML soubor obsahující přehled změněných dat, která jsou vizualizována jako červené obrysů. Aktuální soubor je možné stáhnout z webové stránky http://mw1.google.com/mw-earth-vectordb/Imagery_Updates/imagery_updates.kml. Předem není jasné, které snímky budou v následující aktualizaci obnoveny. Výjimku tvoří pouze přírodní katastrofy, kdy jsou snímky obnovovány co nejrychleji tak, aby mohly pomoci záchranařům a dotčeným obyvatelům v postižené oblasti. Google na svých stránkách uvádí, že většina satelitních snímků by neměla být starší tří let.[20]



Obrázek 1.5: Aktualizace snímků ze dne 21.1.2014

1.1.3 Kartografické zobrazení

Zobrazení je matematicky definovaný vztah používaný pro reprezentaci 3D povrchu Země do roviny mapy (2D). Při zobrazení z jedné referenční plochy do druhé dochází vždy k deformacím, které zkreslují jednu nebo více veličin - tvar, plochu, vzdálenost nebo směr. Existuje spousta mapových zobrazení a při volbě jednoho z nich je nutné brát ohled na účel výsledné mapy.

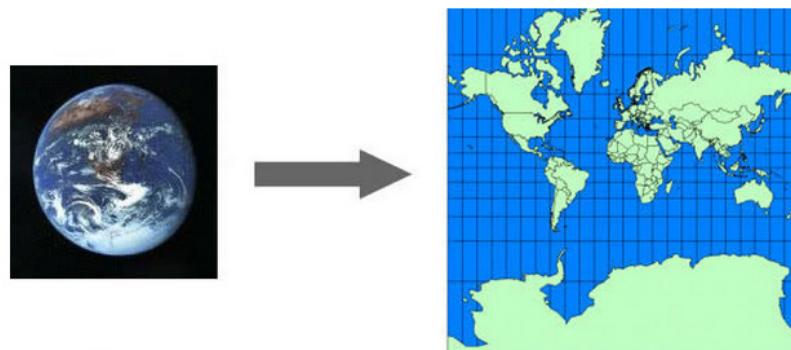
Google Maps pro účely výpočtu zeměpisných souřadnic a zobrazení zemského povrchu do roviny využívá Mercatorovo zobrazení. Jedná se o druh válcového úhlojevného zobrazení, jehož autorem je vlámský kartograf Gerhard Mercator. Přestože bylo zobrazení poprvé představeno v roce 1569, je pro svoje vlastnosti hojně využíváno i dnes.

Všechny rovnoběžky včetně rovníku se zobrazují jako síť rovnoběžných přímek a obrazy poledníků tvoří síť stejně odlehlých přímek kolmých na rovnoběžky. V důsledku různé odlehlosti rovnoběžek směrem od rovníku, se severní a jižní pól zobrazují v nekonečnu. Z toho také vyplývá, že délkové zkreslení se bude směrem od rovníku stále zvětšovat. Mapa je z tohoto důvodu a z důvodu, aby tvořila čtverec, uříznuta přibližně od 85° severní a jižní šířky.

Nejdůležitější vlastností Marcatorova zobrazení je především konformita. Ta zaručuje zachování tvaru i u menších objektů, což je především důležité pro zobrazení satelitních snímků, kde je požadavek na nezkreslení původního tvaru budov a náměstí. Nezkreslených úhlů (azimutů) využívá také loxodroma, která je v tomto zobrazení znázorněna přímkou. [3]

Pro jednodušší výpočty je pro zobrazení map Google používáno zobrazení z koule do roviny, nikoli z elipsoidu poskytující vyšší přesnost. To způsobí měřítkové zkreslení pouze 0,33% ve směru osy Y (směr ze západu na východ) a je tedy možné ho zanedbat.

Aplikace a doplnkový plugin Google Earth používá jednoduché válcové zobrazení, kde jsou rovnoběžky a poledníky zobrazeny jako pravidelná síť stejně odlehlých a vzájemně kolmých přímek. Toto zobrazení je také známo jako Marinovo zobrazení. [13]



Obrázek 1.6: Mercatorovo zobrazení

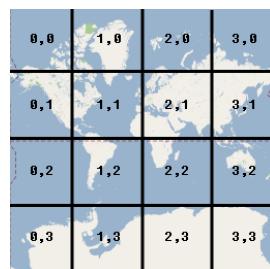
1.1.4 Souřadnicový systém

Ačkoliv Google pro zobrazení využívá Mercatorovo zobrazení a jeho sférickou variantu, výpočet souřadnic probíhá na elipsoidu WGS84 (World Geodetic System 1984). K vyjádření polohy bodu slouží zeměpisná délka a zeměpisná šířka. Stejný souřadnicový systém používá například satelitní navigace GPS. Pro vyhledání bodu v mapách Google je nutné zadat souřadnice ve formátu, který je podporován. Přehledný seznam podporovaných formátů je zobrazen níže.

- Stupně, minuty a sekundy: $50^{\circ}4'56.53''$, $14^{\circ}26'54.42''$
- Stupně a desetinné minuty: $50^{\circ}4.942'$, $14^{\circ}26.907'$
- Desetinné stupně: 50.08237 , 14.44845

Souřadnice mapových dlaždic jsou pro běžného uživatele nepodstatným pojmem, kterému ani není nutné věnovat pozornost, ale pro vývojáře webových mapových aplikací je jejich znalost téměř podmínkou. Jak již bylo zmíněno dříve v této kapitole, Google Maps stejně jako řada dalších poskytovatelů online mapových služeb využívá mapových dlaždic ke zrychlení načítání celé stránky. Mapové dlaždice jsou předem generované čtvercové rastrové obrázky o konstantní velikosti 256 x 256 pixelů, které jsou logicky uspořádány tak, aby jejich spojením vznikla souvislá mapa na každé úrovní přiblžení. Z předešlé věty je tedy patrné, že pro každou úroveň přiblžení musí existovat rozdílná sada mapových dlaždic. Při každé změně středu nebo měřítka mapy Maps API vyhodnocuje situaci a případně vrací souřadnice dlaždic, které mají být aktuálně zobrazeny. Souřadnice jsou vyjádřeny hodnotami x a y. Počátek souřadnic je umístěn v severozápadním rohu mapy, přičemž hodnota x vzhledem k mapě je směrem na východ a hodnota y vzhledem k mapě je směrem na jih.

Celý svět na úrovni přiblžení 0 pokryje pouze jedna jediná dlaždice, ale s každou úrovní přiblžení jejich počet roste. Lépe to bude pochopitelné na příkladu. Mapa na úrovni přiblžení jedna bude obsahovat 4 dlaždice (grid 2x2) a celkové rozlišení mapy bude 512 x 512 pixelů, na úrovni 2 už bude mapa složena z 16 dlaždic (grid 4x4) a celkové rozlišení naroste na 1024 x 1024 pixelů. Takto by bylo možné pokračovat dále až do nejvyššího přiblžení. [14]



Obrázek 1.7: Souřadnice mapových dlaždic na úrovni 2

Využití Google Maps API v3

Application Programming Interface (API) je označení pro programové rozhraní využívající proměnné, funkce a metody veřejné knihovny. Součástí tohoto rozhraní je dokument, který detailně popisuje definici jednotlivých tříd a jejich chování.

Síla Google Maps API spočívá ve vytváření interaktivních webových aplikací nad mapovými podklady nejpoužívanější mapové služby na světě. Nad tímto mapovým podkladem je možné vizualizovat vlastní data nebo data z rozsáhlé databáze míst udržované Googlem. Mimo to je možné využít veškeré služby známé z online map - satelitní snímky, snímky z pohledu 45 stupňů, Street View, trasy jízdy a výškové profily.

Google Maps API pro web jsou založeny na velmi jednoduché sadě tříd napsané v jazyce JavaScript. Mapové dlaždice jsou asynchronně načítány na pozadí stránky voláním Ajax příkazů a následně umístěny do elementu `<div>`. Při pohybu mapou nebo při oddalování či přiblížení API opět pomocí Ajax příkazů posílá informace o změnách a vrací nové dlaždice. Výhodou načítání dlaždic pomocí Ajax příkazů je odstranění nutnosti načítat a překreslovat celé webové stránky. Mimo Google Maps JavaScript API jsou dostupné i rozhraní pro mobilní zařízení používající operační systém iOS nebo Android a další API rozšiřující možnosti mapy. [8]

Název	Popis
Places API	detailní databáze míst
Direction API	trasa mezi body na mapě
Geocoding API	konvertor adresy na geografické souřadnice
Elevation API	výška bodu a výškový profil trasy
Static Maps API	statické snímky mapy
Street View Image API	statické snímky Street View
Google Earth API	digitální 3D glóbus

Tabulka 2.1: Rozšiřující API

Podle serveru *programmableweb* je Google Maps API mezi uživateli dlouhodobě nejpoužívanějším programovým rozhraním, které dnes používá více jak 800 000 webů. Díky takové rozšířenosti mezi uživateli a programátory existuje celá řada webů a internetových diskuzí, které se zabývají touto tématikou. Spolu s dobře zpracovanou dokumentací a referencí jednotlivých tříd na stránkách *Google Developers* tvoří tyto weby nepřeberné množství řešených příkladů a zdrojových kódů. [23]

Mimo aplikační programové rozhraní od Googlu nabízí možnost přizpůsobit a umístit mapu na webové stránky další hráči na trhu s online mapovými aplikacemi, například:

- Bing Maps API,
- Yahoo Maps API,
- ArcGIS API for JavaScript,
- MapQuest JavaScript API,
- Here Maps API,
- OpenLayer Javascript API,
- Mapy API (Seznam.cz).

2.1 Úvod do JavaScriptu

Jak již bylo napsáno výše, Google Maps API jsou založeny na programovacím jazyce JavaScript. Tento skriptovací jazyk patří mezi objektově orientované programovací jazyky a využívá se při tvorbě webových stránek, do kterých umožňuje vkládat dynamické prvky a vytvářet tak vizuálně poutavé weby. JavaScript je závislý na internetovém prohlížeči a pro správné zobrazení obsahu stránky je nutné mít v prohlížeči JavaScript povolený - u většiny prohlížečů ve výchozím nastavení.

Běžným omylem bývá spojování JavaScriptu s programovacím jazykem Java. Tyto dva jazyky však spojuje pouze podobná syntaxe, která patří do stejné rodiny jako syntaxe jazyků C a C++. Pro pochopení následujících kapitol je nutné porozumět základům tohoto jazyka. Programovací jazyky C++ a Java odlišují proměnné pro číselné hodnoty, textové řetězce a další, zatímco JavaScript má pouze jednu proměnnou umožňující uchovávat všechny typy hodnot. Proměnná je deklarována klíčovým slovem *var*.

```
var number = 5.2;  
var text = 'JavaScript';  
var text = "JavaScript";  
var boolen = true;  
var boolen = false;
```

Dalším podstatným rozdílem v syntaxi je vytváření funkcí. V jazyce JavaScript se deklarace funkce skládá z klíčového slova *function* a názvu funkce, za kterým následují jednoduché a následně složené závorky. Hodnoty v jednoduchých závorkách vstupují do funkce jako parametry a složené závorky pak ohraničují vlastní funkci. Funkce je pak zavolána opsáním názvu funkce a vložením povinných parametrů funkce. Výsledkem volání následující funkce je okno s textem „Hello world“.^[27]

```
function message(text){alert(text);}
message("Hello world")
```

2.2 Vložení mapy na stránku

Tématem této práce je vytvoření webové mapové aplikace s využitím Google Maps API. Z tohoto důvodu jsem se rozhodl podrobně popsat jednotlivé kroky vedoucí od vložení jednoduché mapy na stránku, přes nastavení vlastního uživatelského rozhraní, až po vytvoření vlastního stylu mapy.

Aplikace využívající Google Maps API by měla obsahovat tzv. API key, který umožňuje sledovat aktivity na webových stánkách včetně čerpání limitů. Ty jsou nastaveny podle verze API. Při překročení limitů, které nabízí bezplatná verze, je možné dodatečně kvóty dokoupit nebo přejít na placenou verzi Google Maps API for Business.

Klíč může být přidělen pouze přihlášeným uživatelům, a proto je nutné již vlastnit nebo si vytvořit nový účet u společnosti Google. S tímto účtem je možné se přihlásit na stránkách *Google Developers Console*, vytvořit nový projekt a aktivovat Google Maps JavaScript API v3. Při vytváření projektu a při aktivaci API je nutné se seznámit s licenčními podmínkami využívaných služeb (Google APIs, Google Maps/Earth APIs) a souhlas s nimi potvrdit. Při generování nového klíče je také z bezpečnostních důvodů doporučeno vložit doménu, na které bude aplikace umístěna. Tím se zamezí zneužití klíče neoprávněnou osobou na jiných stránkách, než které jsou uvedeny v seznamu.^[9]

Služba	Limit
Geocoding Web Service	2 500 požadavků za den
Directions Web Service	2 500 požadavků za den + 10 bodů na požadavek
Distance Matrix Web Service	2 500 prvků za den + 100 prvků na dotaz
Elevation Web Service	2 500 požadavků za den
Static Maps API	Maximální rozlišení 640 x 640 pixelů
Street View Image API	Maximální rozlišení 640 x 640 pixelů

Tabulka 2.2: Limity pro využití Google Maps JavaScript API

```
<!DOCTYPE html>
<html>
  <head>
    <title>Orlické hory</title>
    <meta name="viewport" content="initial-scale=1.0,
          user-scalable=no" />
    <meta charset="utf-8">
    <style type="text/css">
      html { height: 100% }
      body { height: 100%; margin: 0; padding: 0 }
      #map-canvas { height: 100% }
    </style>
    <script type="text/javascript"
      src="https://maps.googleapis.com/maps/api/js?key=API_KEY&
           sensor=false"
    </script>
    <script type="text/javascript">
      function initialize() {
        var mapOptions = {
          center: new google.maps.LatLng(50.25955, 16.41699),
          zoom: 11
        };
        var map = new google.maps.Map(document.getElementById("map-canvas"),
          mapOptions);
      }
      google.maps.event.addDomListener(window, 'load', initialize
        );
    </script>
  </head>
  <body>
    <div id="map-canvas"/>
  </body>
</html>
```

Listing 2.1: Základní mapa

2.2.1 Rozbor kódu

Každý správný HTML soubor začíná řádkou `<doctype>`, která internetovému prohlížeči sděluje jak stránku správně interpretovat dle specifikace W3C¹. V případě žádného nebo špatně zadaného typu vykreslovacího režimu se stránka automaticky přepne do „quirk módu“, který tuto specifikaci záměrně porušuje, aby mohla být stránka i tak načtena. Tento mód je také často nazýván jako „adaptivní režim“. Deklarace starších HTML byly vícerádkové a nebylo jednoduché si je zapamatovat. S příchodem HTML5 se deklarace zjednodušila pouze na dvě slova, tak jak je uvedeno níže. I přestože se jedná o typ vztahující se k HTML5, je plně kompatibilní i s přechozími verzemi HTML.

```
<!DOCTYPE html>
```

Další obsah kódu je vhodné umístit mezi počáteční `<html>` a koncovou značku `</html>`, i přestože se jedná o značku nepovinnou. Pro značku je v některých českých a většině cizojazyčných literaturách používán pojmenování tag.

Hlavička souboru začíná a končí párovou značkou `<head>` resp. `</head>`. V hlavičce je umístěna ta část dokumentu, která se na stránce nezobrazuje. Výjimku tvoří prostý text, který se v některých prohlížečích zobrazí na začátku stránky.

V hlavičce je umístěno hned několik důležitých, avšak nepovinných elementů. Prvním z nich je element `<title>`, jenž obsahuje název stránky. Element `<meta charset>` je nutný pro správné jazykové kódování stránky. Já jsem pro svoji aplikaci použil kódování „utf-8“, které zajišťuje správnou interpretaci českých znaků. Dalšími elementy obsaženými v hlavičce, které budou podrobněji popsány níže, jsou style a script. Mimo jiné lze do hlavičky umístit popis stránky, klíčová slova nebo jméno autora.

```
<title>Orlické hory</title>
<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
<meta charset="utf-8">
```

Webové stránky se od sebe liší především obsahem, ale také svojí formou. Formou se v tomto případě rozumí jejich vzhled a uspořádání jednotlivých částí. Pro formátování stránky jsou v dnešní době nejběžněji používány kaskádové styly („Cascading Style Sheet“ nebo také CSS), které oproti formátování v HTML přináší rozsáhlejší možnosti. Mimo to oddělují obsahovou část dokumentu (HTML) od té vizuální (CSS). Pro zápis stylu v jazyce CSS se používá párová značka `<style>` s označeným typem stylu, tak jak je uvedeno na příkladu. V tomto konkrétním případě jsou nastavovány vlastnosti elementů html, body a oddílu div s názvem „map-canvas“, který byl vytvořen ve vlastním těle dokumentu mezi tagy `<body>` a `</body>`. Tento oddíl vyhrazuje místo pro vložení mapy. Atributy height případně width nastavují velikost tohoto oddílu, která může být udána

¹World Wide Web Consortium - mezinárodní konsorcium pro vývoj webových standardů.

v pixelech nebo v procentech. [15]

```
<style type="text/css">
    html { height: 100% }
    body { height: 100%; margin: 0; padding: 0 }
    #map-canvas { height: 100% }
</style>
```

Do takto vytvořené webové stránky je již možné načíst Google Maps API v podobě JavaScript souboru, který je umístěn na serverech Googlu. Odkaz na tento soubor v podobě adresy URL se umístí do atributu src, který je jedním ze dvou povinných atributů elementu <script>. Tím prvním je atribut type, který definuje programovací jazyk skriptu. I v adrese odkazu na JavaScript soubor se ukrývají dva povinné parametry, které musí být vyplněny. Parametr key obsahuje aplikační klíč nezbytný pro chod aplikace. Způsob získání tohoto klíče byl již podrobně popsán v úvodu kapitoly 2.2 na straně 14. Další požadovaný parametr sensor dává aplikaci informaci o tom, zda přístroj má či nemá zařízení pro určování polohy. To je důležité především pro mobilní zařízení se sensorem GPS. Parametr nabývá dvou hodnot, true a false. True v případě, že zařízení může pro určování polohy využívat sensor, a false v případě opačném. [4]

```
<script type="text/javascript"
        src="https://maps.googleapis.com/maps/api/js?key=API_KEY
            &sensor=false"
</script>
```

Po načtení Google Maps API zbývá nastavit pouze vlastnosti načítané mapy, jejichž deklarace je součástí objektu *Map options*. Základním a povinným nastavením každé mapy je střed mapy (center) a úroveň přiblížení (zoom):

- Center pomocí souřadnic definuje střed mapy. Nový objekt vytvořený metodou google.maps.LatLng(lat: number, lng: number), jejíž vstupem jsou souřadnice v pořadí zeměpisná šířka a zeměpisná délka.
- Zoom udává úroveň přiblížení. Nabývá pouze celočíselných hodnot od 0 do 21, dle oblasti.

```
var mapOptions = {
    center: new google.maps.LatLng(50.25955, 16.41699),
    zoom: 11
};

var map = new google.maps.Map(document.getElementById("map-canvas"), mapOptions);
```



Obrázek 2.1: Praha - zoom level 0, 10 a 17

Poslední řádek tohoto kódu umisťuje mapu s vlastním nastavením do oddílu s názvem „map-canvaſ“. Aby byla celá mapa bez nepředvídatelného chování načtena, je nutné veškeré nastavení a chování mapy zabalit do funkce, která bude spuštěna až po načtení celé stránky. [27]

```
google.maps.event.addDomListener(window, 'load', initialize);
```

2.2.2 Adresářová struktura

Základní mapa má pouze několik řádků zdrojového kódu a není tedy problém psát kód do jednoho souboru. S přibývajícími řádky by se však kód mohl stát nepřehledným a chaotickým. Proto je velice praktické oddělit JavaScript a kaskádové styly CSS od hlavního kódu do samostatných souborů. Výhodou tohoto oddělení je vysoká čitelnost i v případě složité webové aplikace. Oddělením se také zrychlí načítání celé stránky, neboť prohlížeč může do mezipaměti načítat JavaScript soubor nezávisle na HTML. Dobrým zvykem bývá umisťovat tyto soubory do odlišných adresářů.



Obrázek 2.2: Příklad adresářové struktury

Pro přehlednost zde uvádíم obsah všech souborů. Podle adresářové struktury nejprve soubor se styly *style.css*, poté JavaScript soubor s nastavením mapy *google_map.js* a na závěr HTML soubor *index.html*.

```
html { height: 100% }
body { height: 100%; margin: 0; padding: 0 }
#map-canvas { height: 100% }
```

Listing 2.2: style.css

```
function initialize() {
    var mapOptions = {
        center: new google.maps.LatLng(50.25955, 16.41699),
        zoom: 11,
    };
    var map = new google.maps.Map(document.getElementById("map-canvas"),
        mapOptions);
}
google.maps.event.addDomListener(window, 'load', initialize);
```

Listing 2.3: google_map.js

```
<!DOCTYPE html>
<html>
    <head>
        <title>Orlické hory</title>
        <meta name="viewport" content="initial-scale=1.0,
            user-scalable=no" />
        <meta charset="utf-8">

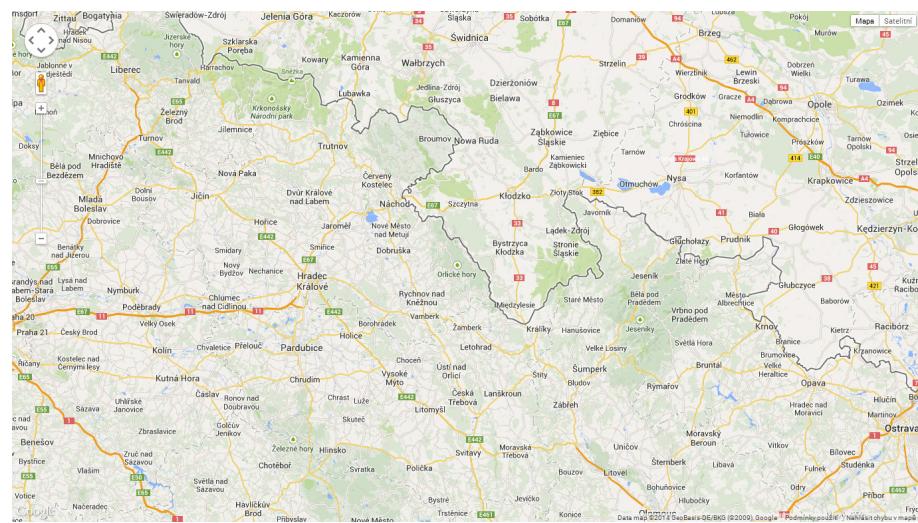
        <script type="text/javascript" src="https://maps.googleapis.
            com/maps/api/js?key=AIzaSyAUfPft30TYnx_LaRx0mW7L-H33eWzInrU
            &sensor=false">
        </script>
        <script type="text/javascript" src="js/google_map.js"></
            script>
        <link type="text/css" rel="stylesheet" href="css/style.css">

    </head>
    <body>
        <div id="map-canvas"/>
    </body>
</html>
```

Listing 2.4: index.html

2.2.3 Mobilní zařízení

Stále populárnější mobilní zařízení s internetovým připojením mají zpravidla menší velikost displeje než osobní počítače a také nabízejí jiné možnosti ovládání mapy. Proto se vývojáři z týmu Maps API při vývoji verze 3 zaměřili na funkčnost aplikace na mobilních zařízeních s operačním systémem iOS a Android. Pro správnou funkčnost aplikace na těchto zařízeních je pro umístění mapy doporučeno používat `<div>` s parametry `width` a `height` nastavenými na 100%. Existuje však metoda detekování zařízení, která umožňuje přizpůsobit nastavení mapy individuálně pro rozdílná zařízení. [4]



Obrázek 2.3: Základní mapa

2.3 Nastavení uživatelského rozhraní

Součástí každé vkládané mapy jsou základní ovládací prvky umožňující interaktivní ovládání. Do mapy s minimem nastavení, tak jak bylo podrobně popsáno v předešlé podkapitole, se automaticky vloží ovládací prvky pro volbu podkladové mapy a prvky pro práci s mapou (posun a přiblížení). Následuje kompletní výčet a popis ovládacích prvků a klávesových zkratek, které Google Maps API nabízí. [10]

- **MapTypeControl** umožňuje uživateli přepínat mezi jednotlivými typy map (satelitní snímky, terénní mapa). Ve výchozím stavu je tento ovládací prvek nastaven na hodnotu TRUE a je umístěn v pravém horním rohu mapy.
- **ZoomControl** slouží k přiblížení a oddálení mapy. Ve výchozím stavu je tento ovládací prvek nastaven na hodnotu TRUE a je umístěn v levém horním rohu mapy, v případě dotykových zařízení v levém dolním rohu.

- **PanControl** umožňuje uživateli posunovat mapou kliknutím na požadovaný směr posunu znázorněný šipkou. Ve výchozím stavu je tento ovládací prvek nastaven na hodnotu TRUE a je umístěn v levém horním rohu nad Zoom Control.
- **ScaleControl** zobrazuje aktuální měřítko mapy. Ve výchozím stavu je nastaven na hodnotu FALSE. Pokud by byl prvek explicitně nastaven na hodnotu TRUE, zobrazil by se v pravém dolním rohu.
- **StreetViewControl** je vizualizován pomocí žlutého panáčka, kterého je možné přesunout nad určité místo na mapě a aktivovat tak režim Street View. Ve výchozím stavu je nastaven na hodnotu TRUE a je umístěn mezi ovládací prvky přiblížení a posunu mapy.
- **RotateControl** je tvořen malou ikonou, která umožňuje uživateli rotovat s mapou. Tento ovládací prvek je spojen s 45° snímky a jeho zobrazení/skrytí závisí právě na povolení těchto snímků.
- **OverviewMapControl** zobrazuje v pravém dolním rohu zmenšenou a oddálenou mapu aktuálně zobrazovaného území. Ve výchozím stavu je však nastaven na hodnotu FALSE.
- **DisableDoubleClickZoom** nastavením na hodnotu TRUE zakáže přiblížení a nastavení středu mapy dvojklikem.
- **Draggable** nastavením na hodnotu FALSE zakáže pohybovat mapou tažením myši.
- **KeyboardShortcuts** nastavením na hodnotu FALSE zakáže ovládání mapy pomocí klávesových zkratek. Pro pohyb v mapě je možné využít navigačních tlačítek (šipek) a pro přiblížení nebo oddálení klávesy $+/-$.
- **Scrollwheel** nastavením na hodnotu FALSE zakáže přiblížování a oddalování mapy kolečkem myši.

Jak již bylo zmíněno výše, vložená mapa obsahuje základní prvky uživatelského rozhraní. Google Maps API dává programátorovi nástroj, jak toto automatické nastavení zrušit a vytvořit si nastavení vlastní. Zrušení nastavení se provede přiřazením hodnoty TRUE ke třídě disableDefaultUI, která je součástí objektu MapOptions.

```
disableDefaultUI: true
```

I když je výchozí uživatelské rozhraní vypnuto, je stále možné povolit nebo zakázat zobrazení ovládacích prvků individuálně.

```
mapTypeControl: boolean,
zoomControl: boolean,
panControl: boolean,
scaleControl: boolean,
streetViewControl: boolean,
overviewMapControl: boolean
```

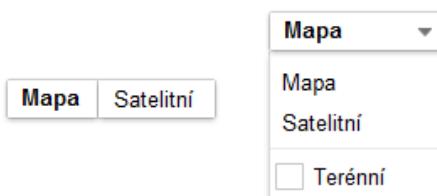
Většinu ovládacích prvků je možné přizpůsobit individuálním požadavkům aplikace. Vlastnosti prvků jsou vkládány spolu s ostatním nastavením mapy do objektu MapOptions. Následující tabulka shrnuje jednotlivé ovládací prvky a jejich možnosti úpravy. [27]

Prvek	Nastavení	Třídy
mapTypeControl	position	google.maps.ControlPosition
	style	google.maps.MapTypeControlStyle
	mapTypeId	google.maps.MapTypeId
zoomControl	position	google.maps.ControlPosition
	style	google.maps.ZoomControlStyle
panControl	position	google.maps.ControlPosition
scaleControl	position	google.maps.ControlPosition
streetViewControl	position	google.maps.ControlPosition
rotateControl	position	google.maps.ControlPosition
overviewMapControl	position	google.maps.ControlPosition

Tabulka 2.3: Nastavení ovládacích prvků

MapTypeControlOptions

Pokud je *MapTypeControl* aktivní, ovládá tento parametr vlastnosti tohoto prvku. První nastavení, style, definuje grafickou podobu. Hodnota *DEFAULT* automaticky nastavuje podobu v závislosti na velikosti okna a dalších faktorech. Dalším možným nastavením je *DROPODOWN_MENU* nebo *HORIZONTAL_BAR*. Druhé nastavení mapTypeId kontroluje zobrazené typy map, ze kterých má uživatel aplikace na výběr (*ROADMAP*, *SATELITTE*, *HYBRID*, *TERRAIN*).



Obrázek 2.4: Horizontal bar a Dropdown menu

ZoomControlOptions

U ovládacího prvku `ZoomControl` je možné navíc nastavit pouze podobu. Google Maps API nabízí programátorům hned tři podoby. Nastavením *DEFAULT* se podoba nastavuje s ohledem na velikost okna. Pokud je dostatek prostoru, je zvolena varianta *LARGE*. V opačném případě varianta *SMALL*, která obsahuje pouze malá tlačítka *+/-*.



Obrázek 2.5: ZoomControlStyle Large a Small

2.3.1 Pozice objektu na mapě

Z předešlé tabulky vyplývá, že každému prvku lze explicitně přidělit pozici na mapě. Ovládací prvky nepoužívají absolutní pozicování, ale v závislosti na velikosti mapy používají inteligentní rozložení ovládacích prvků kolem stávajících prvků. Možná je také kolize s prvky uživatelského rozhraní, kterým pozici měnit nelze (logo společnosti nebo panel s informacemi o datech). V takovém případě jsou ovládací prvky umístěny co nejblíže požadované pozici. Následujících několik řádků obsahuje kompletní seznam a popis možných pozic včetně chování v kolizních situacích. Poté následuje grafické znázornění prvků na mapě. [10]

TOP_LEFT pozice v levé části horního okraje mapy. V případě kolize se prvek posune do středu horního okraje mapy.

TOP_CENTER pozice podél horního okraje mapy.

TOP_RIGHT pozice v pravé části horního okraje mapy. V případě kolize se prvek posune do středu horního okraje mapy.

LEFT_TOP pozice v levé části horního okraje mapy, avšak pod prvkem s pozicí *TOP_LEFT*.

RIGHT_TOP pozice v pravé části horního okraje mapy, avšak pod prvkem s pozicí *TOP_RIGHT*.

LEFT_CENTER pozice podél levého okraje mapy.

RIGHT_CENTER pozice podél pravého okraje mapy.

LEFT_BOTTOM pozice v levé části dolního okraje mapy, avšak nad prvkem s pozicí *BOTTOM_LEFT*.

RIGHT_BOTTOM pozice v pravé části dolního okraje mapy, avšak nad prvkem s pozicí *BOTTOM_RIGHT*.

BOTTOM_LEFT pozice v levé části dolního okraje mapy. V případě kolize se prvek posune do středu dolního okraje mapy.

BOTTOM_CENTER pozice podél dolního okraje mapy.

BOTTOM_RIGHT pozice v pravé části dolního okraje mapy. V případě kolize se prvek posune do středu dolního okraje mapy.

TOP_LEFT	TOP_CENTER	TOP_RIGHT
LEFT_TOP		RIGHT_TOP
LEFT_CENTER		RIGHT_CENTER
LEFT_BOTTOM		RIGHT_BOTTOM
BOTTOM_LEFT	BOTTOM_CENTER	BOTTOM_RIGHT

2.3.2 Metody třídy Map

Nastavení mapy, které je součástí objektu MapOptions, je vhodné pouze při zakládání nové mapy (po načtení stránky již není možné toto nastavení měnit). Pokud programátor chce měnit nastavení i mimo tento objekt, musí využít jednu z členských metod, které patří mezi nejdůležitější částí Google Maps API. Existují dva typy těchto metod: komplexní metoda nastavující požadované vlastnosti nebo specifická metoda pro nastavení konkrétní vlastnosti. Následující tabulka shrnuje nejpoužívanější metody. Pro většinu vlastností lze ještě rozlišit dvě metody často označované jako „gettry“ a „settry“. Zjednodušeně řečeno „gettry“ se používají pro zjištění aktuálně použité vlastnosti a „settry“ se používají pro nastavení vlastnosti nové. [4]

Nastavení	Metoda	Popis
Options	setOptions	tato metoda umožňuje nastavení mapy včetně vypnutí/zapnutí ovládacích prvků a klávesových zkratek
Center	setCenter	nastavuje střed mapy, vstupem je proměnná uchovávající zeměpisné souřadnice
	getCenter	navrací zeměpisné souřadnice aktuálního středu mapy
Zoom	setZoom	nastavuje úroveň přiblížení, vstupem je celočíselná hodnota
	getZoom	navrací celočíselnou hodnotu aktuální úrovni přiblížení
MapType	setMapTypeId	nastavuje typ mapy (silniční, satelitní, hybridní, terénní)
	getMapTypeId	návratovou hodnotou je typ mapy v podobě textového řetězce
Pan	panBy	animovaný posun mapy o hodnoty udaných v pixelech
	panTo	animovaný posun mapy na pozici udanou v zeměpisných souřadnicích

Tabulka 2.4: Metody pro změnu nastavení

Třída Map obsahuje i další metody, které však během této práce nebyly použity, a proto zde nejsou uvedeny. Metody se stávají opravdu silným nástrojem až ve spojení s posluchači (event listeners) a událostmi (events), které reagují na nejrůznější podněty uživatele a umožňují tak vzájemnou interakci. První skupina tzv. user events je téměř totožná se standardními DOM událostmi, ale ve skutečnosti jsou tyto události součástí Maps API. Důvodem je vyšší kompatibilita napříč prohlížeči. Další skupinu tvoří události reagující na změny v mapě, například změna přiblížení nebo změna středu mapy. Následuje pouze seznam Maps API událostí, neboť jejich význam je patrný z názvu. [10]

click	center_changed	drag
dbclick	zoom_changed	dragend
rightclick	maptypeid_changed	dragstart
mousemove	bounds_changed	resize
mouseout	tilt_changed	idle
mouseover	projection_changed	tilesloaded

Tabulka 2.5: Události

2.4 Vlastní styl mapy

Novinka, která se objevila až ve verzi Google Maps JavaScript API v3, nabízí možnost upravit vizuální podobu jednotlivých prvků základní mapy, jako jsou například silnice, parky a zastavěné oblasti. Díky této funkci je možné mapu přizpůsobit designu webových stránek nebo konkrétnímu účelu použití.

K vytvoření vlastního stylu využívá API *featuresTypes*, *elementTypes* a *stylers*. První zmiňovaná část je používána pro výběr geografického objektu. Je možné volit z kategorií jako vodní plochy, krajina, silnice nebo body zájmu. Většina z těchto kategorií je ještě rozšířena o podkategoriю, které blíže specifikují rozsah použití. Geografické objekty jsou velmi často tvořeny geometrickou částí, názvem a v některých případech i značkou. *ElementTypes* umožňuje aplikovat styl pouze na některé z těchto zvolených částí.

Kategorie	Podkategorie
administrative	country, land-parcel, locality, neighborhood, province
landscape	man-made, natural
poi	attraction, business, government, medical, park
road	arterial, highway, local
transit	line, station (airport, bus, rail)
water	nemá žádnou podkategoriю

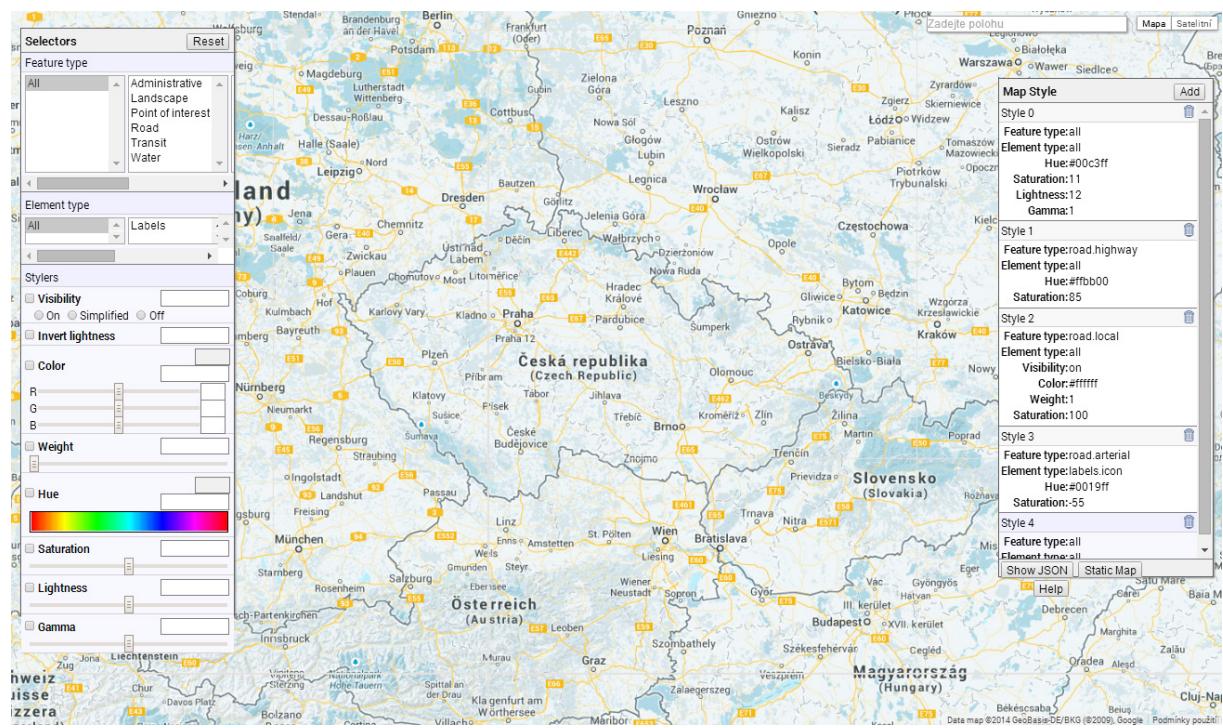
Tabulka 2.6: FeaturesTypes

Kategorie	Podkategorie
geometry	fill, stroke
labels	icon, text (fill, stroke)

Tabulka 2.7: ElementTypes

Stylers kombinací hodnot odstínu, sytosti, jasu a dalších vlastností definuje podobu jednotlivých geografických objektů a jejich částí. Barvu lze nastavit jednou jedinou hodnotou *color* nebo výše zmíněnou kombinací odstínu, sytosti a jasu. Tyto dvě různé metody nastavení barvy je však možné kombinovat, přičemž vlastnosti se projevují ve stejném pořadí, ve kterém jsou definovány. Vytvoření vlastního stylu mapy může být časově náročné a výsledek nemusí být uspokojivý. Naštěstí existuje online interaktivní nástroj Styled Map Wizard, který umožňuje projít všechny kroky nastavení a výsledek okamžitě zobrazovat v mapě. Pokud je uživatel s mapou spokojen, stačí vygenerovat JSON², vložit kód do inicializační funkce mapy a přidělit vlastní mapě identifikační název ID. [10]

²JavaScript Object Notation - JavaScriptový objektový datový formát



Obrázek 2.6: Styled Map Wizard - vlastní styl mapy

```

var styles = [
  {
    "featureType": "all",
    "elementType": "geometry.fill",
    "stylers": [
      { "hue": "#00c3ff" },
      { "visibility": "on" },
      { "saturation": 11 },
      { "lightness": 13 },
      { "gamma": 1 } ]
  }

var styledMap = new google.maps.StyledMapType(styles, {name: "Zimní mapa"});
map.mapTypes.set('winter_map', styledMap);

```

2.5 Přidání překryvné vrstvy

2.5.1 Vrstva KML a GeoRSS

Formát KML je používán pro zobrazování dvojrozměrných i trojrozměrných prostorových dat především v mapových aplikacích Google Earth a Google Maps. Původně byl tento formát vyvinut pro prohlížeč Keyhole Earth Viewer a odtud také pochází jeho původní název Keyhole Markup Language. Ten je však v současné době zastaralý a oficiálně používaným názvem je zkratka těchto tří slov KML.

KML je strukturovaný značkovací jazyk používající vnořené elementy a atributy založené na standardu XML (Extensible Markup Language). V roce 2008 Open Geospatial Consortium (OGC), mezinárodní standardizační organizace spolupracující na vývoji a implementaci standardů pro geoprostorová data a služby, přijalo formát KML ve verzi 2.2 jako mezinárodně uznávaný otevřený standard, což zaručuje harmonizaci dalších verzí s normami OGC. KML se tak stal doplňkem k existujícím standardům GML (Geography Markup Language), WFS (Web Feature Service) a WMS (Web Map Service). Právě z výše uvedeného standardu GML odvozuje reprezentaci geometrických prvků jako je bod, přímka, linie nebo polygon. Ty jsou definovány jako souřadnice lomových bodů v souřadnicovém systému WGS84 vložených do elementu *<coordinates>*.

Aplikace podporující formát KML zpracovávají soubory s příponou názvu souboru .kml a .kmz (komprimovaná verze souboru KML) obdobným způsobem, jako webové prohlížeče zpracovávají soubory HTML nebo XML. [21]

Formát GeoRSS rozšiřuje stávající kanály RSS o prostorovou informaci. RSS je formát založený na standardu XML a jeho hlavním cílem je jednoduše zprostředkovávat a sdílet aktuální novinky na webových stránkách, kde často dochází ke změně obsahu (zpravidelské servery). Polohu lze prostřednictvím GeoRSS interpretovat dvěma způsoby. GeoRSS Simple se pokouší o co nejjednodušší zápis geografické informace, který umožní rychlé nasazení v rámci webu. Druhá varianta GeoRSS GML dodržuje standardy formátu GML a nabízí širší možnosti. [5]

Do mapy je možné geografická data ve výše popsaných formátech vložit s využitím objektu *KmlLayer*, jehož konstruktor má povinný pouze jeden parametr a to adresu nebo cestu k tomuto souboru. Vlastnosti vrstvy jsou součástí souboru a není možné je ve zdrojovém kódu aplikace ovlivnit.

2.5.2 Dynamické tabulky

Google Fusion Tables (v češtině používaný výraz „Dynamické tabulky“) je experimentální webová aplikace, která umožňuje zacházet s velkým objemem prostorových dat.

Data je možné vytvářet v grafickém prostředí aplikace nebo importovat vlastní soubory s maximální velikostí 100MB. Celkový limit dat na jednoho uživatele je omezen na 250MB s tím, že je možné potřebný prostor dokoupit. Fusion Tables podporuje běžné tabulky ve formátu .csv, .tsv., .xlsx nebo .ods a samozřejmostí je podpora Google Spreadsheets. Geografická data je prozatím možné importovat pouze z KML souborů. I zde platí některá omezení, která stanovují maximální počet znaků v buňce. Při překročení jednoho milionu znaků v popisu KML se import souboru nezdaří a skončí varovnou hláškou.

Importovaná nebo vytvořená data je možné dále kombinovat, sdílet s ostatními uživateli, provádět nad daty analýzy a dotazy nebo data graficky vizualizovat. K vizualizaci lze využít grafy nebo každému řádku databáze přiřadit informaci o geografické poloze a prvky vizualizovat v mapě. Pro běžného uživatele, kterému stačí prostor 250MB, je Fusion Tables velice výhodnou alternativou ke klasickým databázovým systémům, jejichž pořizovací náklady nebývají nejlevnější.

Data lze také použít jako vlastní vrstvu v aplikaci využívající Google Maps API. Komunikaci s Fusion Tables obstarává jednoduchý objekt *FusionTablesLayer*. Velikou výhodou je zobrazení dat na základě specifických dotazů, které jsou součástí *FusionTablesLayerOptions* a mají velmi podobnou syntax jako příkazy jazyka SQL (select, from a where). Na základě těchto podmínek je možné jednotlivé prvky graficky odlišit a vytvářet tak jednoduché tematické mapy. Na rozdíl od ostatních vrstev nejsou geografická data zobrazována jako jednotlivé prvky, ale na straně serveru jsou vytvářeny rastry, ze kterých jsou následně generovány mapové dlaždice odpovídající schématu dlaždic, které používají podkladové mapy Google Maps.

Při vkládání tabulky do mapy je důležité dodržet hned několik pravidel a omezení. Hlavní pravidlo říká, že tabulka musí být veřejně sdílena a data musí mít vloženy informace o poloze. Poloha může být určena zeměpisnými souřadnicemi, adresou nebo spojením se souborem KML. Omezení se týkají především počtu vkládaných tabulek, počtu sloupců a jejich velikosti. [11]

2.5.3 Další vrstvy

Mapa je také vhodná pro vizualizaci aktuální dopravní informace, síť městské hromadné dopravy, předpovědi počasí nebo oblačnosti. Toho si je Google vědom, a tak uživatelům využívající Maps API umožňuje tato data vkládat jako překryvné vrstvy nad podkladovou mapou. Následující text zmiňuje pouze ty vrstvy, které byly v této aplikaci použity. [4]

Vrstva *Traffic layer* zobrazuje aktuální dopravní situaci, dopravní nehody nebo plánované dlouhodobé uzavírky. Tato vrstva je však v rámci České Republiky omezena pouze na několik hlavních silnic a dálnic, ale i přesto může znatelně zkrátit dojezdové časy. Její implementace je jen jednořádková a je založena na vytvoření objektu *new google.maps.TrafficLayer()* a vložení tohoto objektu do mapy.

Vrstva *počasí a oblačnosti* vyžaduje přidání knihovny „weather“ na konec reference odkazující na Google Maps API. Vytvořením objektu *new google.maps.weather.CloudLayer()* bude do mapy přidána vrstva oblačnosti viditelné v rozsahu přiblížení 0 až 6. Aktuální teploty včetně ikon se do mapy vloží vytvořením objektu *new google.maps.weather.WeatherLayer()*. Kliknutím na jednu z ikon se otevře nové okno obsahující doplňkové informace o vlhkosti a směru větru, ale také čtyřdenní předpověď počasí pro dané místo. Zdrojem pro aktuální stav a předpověď počasí je server www.weather.com.

Vrstva *Panoramio* je tvořena fotografiemi, jejichž součástí jsou souřadnice místa pořízení. Přidávání geografických metadat k fotografiím či jiným médiím je často označováno cizím termínem „geotagging“. Podobně jako u vrstvy počasí je nutné nejdříve načíst knihovnu Panoramio a poté vytvořit objekt *new google.maps.panoramio.PanoramioLayer()*. Informace o použití služby Panoramio jsou dostupné na stránkách www.panoramio.com. Na těchto stránkách je možné opatřit svoje vlastní fotografie geotagem, zobrazit v mapě a sdílet s ostatními uživateli využívající tuto službu.

Realizace aplikace

Při vytváření aplikace byla použita celá řada programovacích jazyků, které spolu s vývojovým prostředím Google Maps API vytváří celkový vzhled stránky a přívětivé uživatelské rozhraní aplikace. Použité programovací jazyky jsou HTML, CSS, JavaScript a JavaScriptová knihovna jQuery. Psaní zdrojového kódu může zjednodušit správný výběr editoru. Nejjednodušší variantou je editace v poznámkovém bloku, který je součástí každé instalace Windows. Tato varianta se hodí spíše pro drobné úpravy, než pro psaní celého projektu, a proto je vhodnější zvolit speciální editor, jehož velikou výhodou je zvýraznění syntaxe podporovaných programovacích jazyků. Na internetu je ke stažení velké množství volně dostupných editorů pro nejpoužívanější operační systémy. Po přečtení některých diskuzí a rad jsem zvolil editor Sublime Text, se kterým jsem byl velmi spokojen, a mohu ho proto začínajícím programátorem doporučit. V následující kapitole se podrobněji zaměřím na jednotlivé programovací jazyky včetně praktického využití v aplikaci. Součástí kapitoly jsou také informace o datových vrstvách a jejich zpracování.

3.1 Specifikace a historie HTML

HyperText Markup Language, ve zkratce HTML, je označení pro značkovací jazyk používaný pro tvorbu webových stránek a publikaci dokumentů na internetu. Jazyk HTML je charakteristický množinou značek (tzv. tagů), mezi které se uzavírají části textu, a tím se určuje jejich sémantika. Vývoj a definice tohoto jazyka byla ovlivněna především vývojem webových prohlížečů. Historie jazyka HTML je pro pochopení samotné tvorby webových stránek celkem nepodstatná a dá se bez její znalosti obejít, ale podle mého názoru je velmi zajímavá a každý programátor stránek by měl mít minimálně povědomí o vývoji tohoto značkovacího jazyka.

Za zakladatele jazyka HTML jsou považovány zaměstnanci výzkumného centra CERN Tim Berners-Lee a Robert Caillau, kteří pro CERN vyvíjeli jednodušší alternativu ke složitým programovacím jazykům, určeným k vytváření dokumentů (TeX, PostScript nebo SGML). První koncept jazyka HTML a protokolu pro přenos hypertextu HTTP se zrodil v roce 1989 spolu s projektem WWW a o rok později s prvním webovým prohlížečem WorldWideWeb pro počítač NeXT v CERNu. Ten mimo jiné obsahoval možnost samotné editace dokumentu. Webový prohlížeč a hypertext měl sloužit vědcům a vědeckým insti-

tucím po celém světě ke sdílení výzkumných prací. První návrh jazyka HTML byl silně založen na architektuře jazyka SGML, který dodržuje mezinárodně dohodnuté metody členění textu do konstrukčních celků, jako jsou například odstavce, nadpisy nebo položky seznamu. Výraznou novinkou oproti SGML byly tehdy například hypertextové odkazy.

V roce 1991 se po zveřejnění specifikací jazyka HTML zvedla vlna diskusí o budoucnosti a využití HTML pro širokou veřejnost. Prostřednictvím elektronických diskuzí si nadšenci z řad akademických a výzkumných pracovníku vyměňovali nejrůznější nápady na vylepšení. Takovým nadšencem byl například Dave Raggett ze společnosti Hewlett-Packard, který po návratu z návštěvy u Tima Bernerse-Leea začal pracovat na obohacené verzi s označením HTML+ a později také na svém vlastním prohlížeči Arena. V té době se také o HTML začala zajímat organizace NCSA (National Center for Supercomputer Applications), která pověřila programátora Marca Andreessena vytvořením webového prohlížeče s implementací HTML. Prohlížeč byl zveřejněn v roce 1993 a nesl název Mosaic. Byl to první prohlížeč s grafickým uživatelským rozhraním, ale již bez možnosti editace. Marc Andreessen se později stal zakladatelem společnosti Netscape, která díky úspěšnému prohlížeči byla často mylně pokládána za tvůrce webu.

Nově vznikající webové prohlížeče přidávaly do HTML své vlastní prvky, což samozřejmě vedlo mezi programátory a uživateli k chaosu. Jako reakci na tento problém vytvořil Berners-Lee, pod záštitou skupiny IETF (Internet Engineering Task Force), návrh standardu HTML 2.0, který shromažďoval nejpoužívanější HTML tagy napříč prohlížeči. V roce 1994 vznikla, v tomto dokumentu již několikrát zmiňovaná, organizace World Wide Web Consortium, v jejímž čele stál Tim Berners-Lee. Cílem organizace je kontrolovat otevřené webové standardy pro World Wide Web a sjednotit rozdílné verze HTML. První verze standardu pod záštitou W3C byla vydána v roce 1997 a nesla označení HTML 3.2. Ta kombinovala prvky z verze HTML 2.0 a 3.0, která vznikla z HTML+ a přinesla spoustu nových prvků, ale jejich úplnou podporu nabízel pouze prohlížeč Arena. Následovala verze HTML 4.0, která pouze obohacovala předchozí verzi o některé prvky a rámy. Drobné chyby a nedostatky byly řešeny aktualizací na verzi 4.01.

Vývoj jazyka HTML měl být ukončen a nahrazen novým jazykem XHTML odvozeným od standardu XML, ale tento krok se později ukázal jako špatný. Prohlížeče tento nový jazyk většinou nepodporovaly a samotní výrobci založili pracovní skupinu WHATWG (The Web Hypertext Application Technology Working Group), která měla za cíl rozšiřovat jazyk HTML. Sloučením této skupiny a organizace W3C vznikla nová pracovní skupina, která se od roku 2007 podílí na vzniku nového standardu HTML 5. [31] [18]

3.2 HTML5 a Geolocation API

HTML5 je nejnovější specifikace značkovacího jazyka HTML, která oproti svým předchůdcům nabízí možnost vytvářet bohatší webové stránky bez nutnosti načítání přídavných pluginů. Nová specifikace obohacuje HTML o nové tagy, které zvyšují

přehlednost zdrojového kódu (article, header, footer), možnost spouštět webové aplikace i bez nutnosti stálého připojení k internetu, podporu audia i videa bez použití flashe a další novinky. HTML5 je multiplatformní a je navržen tak, aby byla podporována veškerá zařízení přes počítače, tablety, chytré telefony až po stále populárnější Smart TV. HTML5 také podporuje širší spolupráci s rozšířenými kaskádovými styly CSS3 a skriptovacím jazykem JavaScript, ke kterému přistupuje skrze Document Object Model, zkráceně DOM. Verze HTML5 není stále ještě dokončena a v současnosti je ve stádiu označovaném jako *Candidate Recommendation*. U specifikace na stránkách konsorcia je uvedeno datum zveřejnění 4. ledna 2014. V této fázi je jazyk již dokončen a ke konečnému doporučení konsorcia *Recommendation* zbývá jen několik kroků v podobě testování kompatibility na různých prohlížečích a doladování některých detailů. [33]

Nově představenou novinkou, které se budu v této části dokumentu podrobněji zabývat, je geolokační rozhraní Geolocation API. Pomocí tohoto rozhraní je možné zjistit a sledovat aktuální pozici uživatele, respektive polohu jeho zařízení, přímo v prostředí webového prohlížeče. Dochází tak ke zjednodušení a sjednocení metod pro získávání souřadnic ve webových aplikacích. Geolocation API podporuje většina současných desktopových i mobilních prohlížečů. Podporu konkrétních verzí prohlížečů lze dohledat na oficiálních stránkách specifikace konsorcia W3C. [29]

Geolokační rozhraní samo o sobě nezajišťuje aktuální pozici, ale pouze prostřednictvím webové aplikace a prohlížeče s podporou HTML5 zprostředkovává komunikaci se zařízením, které získá a odešle zpět do aplikace aktuální souřadnice ve světovém souřadnicovém systému WGS84. Odeslání požadavku na zjištění aktuální pozice musí uživatel většiny prohlížečů nejdříve povolit, čímž je zajištěno soukromí uživatele a vyvráceny veškeré spekulace o nechtěném sledování. [2]



Obrázek 3.1: Povolení sledování polohy v prohlížeči Google Chrome

3.2.1 Zjištění polohy

Zdrojů, pomocí kterých získávají rozdílná zařízení informace o poloze, je hned několik a v některých případech se tyto zdroje vzájemně kombinují. Nejjednodušší a zároveň nejméně přesnou metodou, jak získat aktuální souřadnice, je lokalizace přes IP adresu. Tato metoda využívá informace o fyzické adrese poskytovatele, která může být od aktuální polohy uživatele až stovky kilometrů vzdálená. Další metoda lokalizace spočívá ve výpočtu polohy zařízení na základě triangulace z vysílacích stanic mobilních operátorů a Wi-Fi přijímačů. Metoda triangulace vrací překvapivě přesné výsledky, ale pouze v případě, že je v okolí dostatek stanic a přijímačů. Posledním a také nejpřesnějším způsobem je lokalizace přes GPS satelity na oběžné dráze Země. Přesnost této metody se pohybuje v řádech metrů, ale je nutné vlastnit zařízení s integrovaným GPS přijímačem a mít

přímý výhled na oblohu. Nevýhodou se může zdát delší časová prodleva před spuštěním samotného přijímače a následným spojením se satelity GPS. Tento problém Geolocation API řeší tak, že nejprve určí polohu na základě jedné z předchozích metod a tu následně po inicializaci GPS zpřesňuje. [22]

3.2.2 Aplikace lokalizace

Každá aplikace, využívající Geolocation API, by měla začínat jednoduchým testem na přítomnost objektu knihovny *navigator.geolocation*, jehož inicializace je závislá na internetovém prohlížeči a svolení uživatele tato data využívat. Tím se vyvarujeme nechtěným chybám při samotném běhu aplikace. Při kladném výsledku tohoto testu může být v aplikaci využita jedna z metod tohoto objektu, v případě opačném může být vypsána varovná hláška informující uživatele o nemožnosti sledování polohy.

Základní metodou pro zjištění aktuální polohy uživatele je metoda *getCurrentPosition()*, jejímž vstupem je jeden povinný a dva nepovinné parametry.

```
getCurrentPosition(succesCallback, errorCallback, options);
```

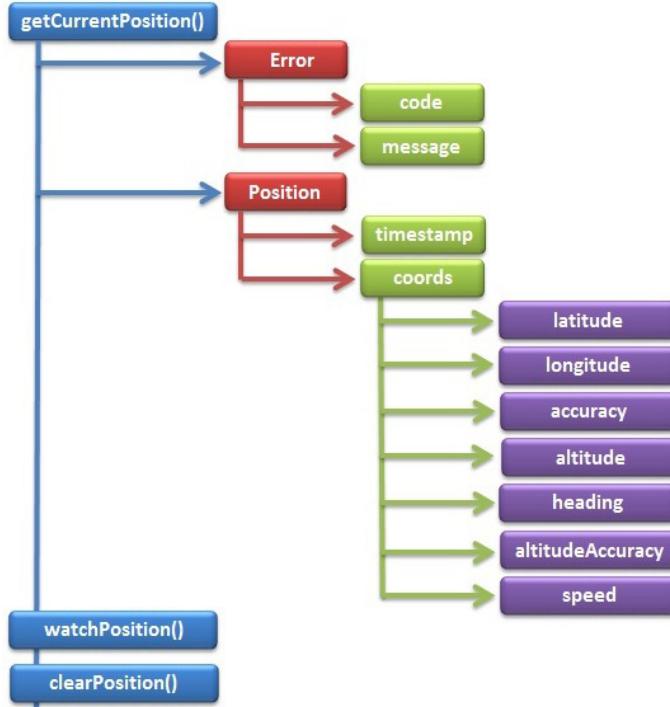
První povinný parametr *succesCallback* definuje návratovou metodu, které je hned po zjištění pozice předán objekt se dvěma atributy - *coords* a *timestamp*. Tento objekt je mimo jiné potomkem objektu, který implementuje rozhraní *position*. Atribut *timestamp* (*časové razítko*) obsahuje pouze informaci o čase zjištění polohy, zatímco atribut *coords* obsahuje všechny důležité informace o poloze. Ty spolu s jednotlivými metodami a objekty přehledně zobrazuje schéma na obrázku 3.2¹.

Informace o zeměpisných souřadnicích jsou reprezentovány dvojicí *latitude*, *longitude* a atributem *accuracy*, udávající přesnost určení polohy v metrech. Souřadnice jsou udávány ve formátu desetinného čísla. Atribut *altitude* vrací nadmořskou výšku v metrech s přesností určenou *altitudeAccuracy*. Ke zjištění aktuální rychlosti v metrech za sekundu nebo směru pohybu zařízení slouží atributy *speed* respektive *headings*. Směr pohybu je udáván azimutem, což je úhel mezi myšlenou přímkou k severu a směrem pohybu. Úhel je počítán po směru hodinových ručiček. Takže například při směru pohybu na východ bude hodnota *headings* 90 stupňů.

První nepovinný parametr *errorCallback* určuje návratovou metodu, která je zavolána v případě chyby nebo nesplnění některé podmínky funkčnosti lokalizace. Poslední a zároveň druhý nepovinný parametr je objekt obsahující následující možnosti:

1. *enableHighAccuracy* je hodnota typu boolean, která v případě nastavení na hodnotu true zapíná zjišťování polohy nejpřesnějším dostupným způsobem, například využitím signálu z GPS družic. Nejedná se tedy o zpřesnění výsledku, jak by mohlo být z názvu špatně pochopeno.

¹<http://imdavidwebb.files.wordpress.com/2012/10/structureofgeoapi.jpg>



Obrázek 3.2: Schéma Geolocation API

2. *timeout* nastavuje maximální časový limit, během kterého se prohlížeč snaží získat informace o poloze. Čas je udáván v milisekundách. Při neúspěšném pokusu se po vypršení intervalu vyvolá definovaná chybová hláška.
3. *maximumAge* specifikuje, jak staré měření může být v aplikaci použito. Například nastavením na hodnotu 600 000 milisekund je zaručeno, že pozice bude maximálně 10 minut stará.

Výše popsaná metoda `getCurrentPosition` je dostatečná u statických zařízení jako jsou například stolní počítače, ale pro mobilní zařízení, která mohou průběžně měnit svoji polohu, je vhodnější využít metodu `watchPosition()`. Tyto dvě metody jsou téměř stejné s jedním jediným rozdílem, že druhá zmínovaná metoda vrací aktuální souřadnice při každé změně polohy. Zrušení sledování polohy obstarává metoda `clearPosition()`. [32]

3.2.3 Použití s Google Maps API

K vizualizaci aktuální polohy uživatele v prostředí Google Maps API jsem využil třídu `google.maps.Marker`, jejíž konstruktor má na vstupu pouze dva povinné parametry - `position` a `map`. První parametr určuje počáteční umístění značky, zatímco druhý parametr určuje mapu, na které se značka zobrazí.

```

var position = new google.maps.LatLng(position.coords.latitude,
    position.coords.longitude);
devicePositionMarker.setOptions({
    position: position,
    map: map,
    icon: {
        url: "css/icon/gps_button.png",
        size: new google.maps.Size(34, 34),
        scaledSize: new google.maps.Size(17, 17),
        anchor: new google.maps.Point(8, 8)
    });
    
```

Obsahem tohoto zdrojového kódu je praktická ukázka využití Geolocation API k vytvoření nové proměnné uchovávající zeměpisnou šířku a zeměpisnou délku aktuální pozice uživatele a následnému umístění značky do mapy. Třída marker byla vytvořena globálně tak, aby k ní byl umožněn přístup z jakékoli části kódu. To je důležité především při sledování polohy metodou *watchPosition()*, která k této třídě přistupuje a kontinuálně, na základě aktuální polohy uživatele, nastavuje novou pozici značky v mapě. [10]

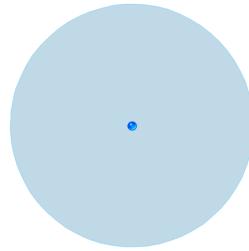
Parametr *icon* umožňuje nahradit výchozí styl ikony stylem vlastním, který pomocí různých nastavení definuje vizuální chování značky. V nejzákladnějším tvaru je nutné uvést pouze odkaz na novou ikonu, která nahradí klasickou podobu připínáčku. Velikost ikony je nastavena automaticky, nebo je možné velikost nastavit ručně. Součástí nastavení je také měřítko a takzvaný kotvící bod, který je závislý na tvaru značky. V tomto konkrétním případě je kotvící bod posunut od levého horního rohu na souřadnice odpovídající středu značky.

Použití jednoduché značky pro zobrazení polohy, jejíž nepřesnost může být v některých případech i několik kilometrů, by mohlo být pro uživatele značně matoucí. Bylo by tedy velmi užitečné nepřesnost určení polohy nějakým způsobem vizualizovat. V tomto případě se přímo nabízí využít třídu *google.maps.Circle*, která nedělá nic jiného, než jen vykresluje kruh s určitým středem a poloměrem. Právě tyto dva parametry, reprezentovaný souřadnicemi aktuální polohy a přesnosti jejího určení, jsou pro vizualizaci nejdůležitější. Doplňující parametry, jako je například barva výplně, průhlednost nebo tloušťka obvodové čáry, definují grafickou podobu kruhu. Ze stejných důvodů jako třída marker byla i tato třída vytvořena globálně.

```

devicePositionCircle.setOptions({
    center: position,
    map: map,
    radius: position.coords.accuracy,
    strokeColor: "#1bb6ff",
    strokeOpacity: 0.4,
    strokeWeight: 1,
    fillColor: "#61a0bf",
    fillOpacity: 0.4 });
    
```

Vizualizace byla dále doplněna o informační okno třídy *google.maps.InfoWindow*, které informuje uživatele o aktuálních souřadnicích, rychlosti pohybu a přesnosti určení polohy.



Obrázek 3.3: Vizualizace polohy a její přesnosti

3.3 Vzhled stránky

Během vývoje jazyka HTML vznikaly rozdílné metody jak text formátovat. Starší a dnes již téměř nepoužívaný způsob formátoval text definovanými HTML tagy, které nepopisovaly obsah, ale způsob jeho zobrazení. Tato metoda nebyla velmi praktická, a proto většina moderních webů využívá pro nastavení vzhledu stránky jazyk CSS, tedy „Cascade Style Sheets“.

Autorem tohoto jazyka je norský vývojář Håkon Wium Lie, spolupracovník Tima Bernerse-Leea, který na prvním návrhu začal pracovat z podnětu standardizační organizace W3C. Cílem bylo vytvořit jazyk, který by odděloval vzhled dokumentu od jeho obsahu. To je dnes možné třemi způsoby. Přímo v textu u formátovaného elementu, v hlavičce zdrojového kódu nebo v externím souboru, což je z hlediska přehlednosti nejpraktičtější. Kaskádové styly jsou s elementy propojeny takzvanými selektory, které na základě třídy nebo identifikátoru přiřazují elementu definovaný styl. [15]

Nejnovější verze kaskádových stylů nese označení CSS3 a oproti svým předchůdcům přináší celou řadu nových vlastností, jako jsou kulaté rohy, stíny, pohyby nebo rotace prvků. Vývoj nové verze kaskádových stylů stále pokračuje a může se tedy stát, že nové vlastnosti nebudou některými webovými prohlížeči a jejich staršími verzemi podporovány. Na druhou stranu podpoře napomáhá konkurenční souboj prohlížečů, které se vzájemně předhánějí v rychlosti implementace nově představených vlastností jak CSS3 tak HTML5. Jednotlivé specifikace a aktuální návrhy CSS3 je možné dohledat na internetových stránkách pracovní skupiny „Cascading Style Sheet working group“, působící pod organizací W3C. [34]

```
.map_inner{  
-webkit-border-radius: 2px;  
-moz-border-radius: 2px;  
border-radius: 2px; }
```

Praktický příklad ukazuje využití vlastnosti *border-radius*, která určuje poloměr zakulacení rohů elementů třídy map_inner. Hodnota zakulacení může být udána v pixelech nebo v procentech šířky elementu. Zadáním posloupnosti čtyř hodnot je možné zakulcovat každý roh zvlášť, nebo vložením jedné hodnoty nastavit jednotné zakulacení pro všechny rohy. V případě prvním se postupuje po směru hodinových ručiček v pořadí levý horní roh, pravý horní roh, dolní pravý roh a dolní levý roh. Většina novějších prohlížečů podporuje vlastnost *border-radius*, nicméně pro zpětnou kompatibilitu je dobré uvádět testovací vlastnosti lišící se prefixem *moz* pro Mozilla a *webkit* pro prohlížeče Chrome, Opera a Safari.

3.4 JavaScript a knihovna jQuery

JavaScript je objektově orientovaný skriptovací jazyk, jehož autorem je americký vývojář Brendan Eich, bývalý zaměstnanec společnosti NetScape. Při jeho představení v roce 1995 nebyl znám pod označením JavaScript, ale pod původním označením Mocha. Název JavaScript získal až o necelý rok později. Nyní se nejčastěji používá jako interpretovaný programovací jazyk pro tvorbu webových stránek. Jazyk je interpretován na straně klienta, což znamená, že všechny skripty jsou po načtení webové stránky dočasně staženy do prohlížeče, kde se spustí (interpretují). V aplikaci sportovních aktivit v Orlických horách byl programovací jazyk JavaScript použit výhradně pro vložení mapy a práci s prostředím Google Maps API, a proto byl podrobněji popsán v kapitole 2.1 Úvod do JavaScriptu. [28]

Práci s JavaScriptem výrazně usnadňuje JavaScriptová knihovna jQuery, která klade důraz na jednoduchost a rychlosť tvorby dynamického webu. Mimo to je obohacena o další funkce pro vytváření událostí, animací a manipulaci s kaskádovými styly. Myšlenku vytvořit knihovnu, která by používala CSS selektory k propojení JavaScriptu a HTML, publikoval poprvé na svém blogu John Resig. Knihovna jQuery byla oficiálně představena na BarCampu² o půl roku později, v lednu 2006. První stabilní verze se uživatelé dočkali až o dalších 6 měsíců později. [16]

3.4.1 Použití knihovny jQuery

Knihovna jQuery je poskytována zdarma v souladu s licenčními podmínkami MIT, které umožňují bezplatné použití knihovny i v komerčních projektech. Jedinou podmínkou je zachování autorských práv v hlavičce souboru. Pro použití ve vlastním projektu je při prvním kroku nutné vybrat správnou verzi knihovny a stáhnout potřebný komprimovaný či nekomprimovaný soubor. Alternativní variantou může být přidání odkazu na knihovnu nacházející se na CDN (Content Delivery Network) serverech. Tato metoda umožňuje ke

²Mezinárodní síť otevřených konferencí zaměřujících se na nové trendy v celé řadě oborů

knihovně rychlejší přístup, než v případě umístění na vlastním serveru nebo jiné hostingové službě.

```
<script type="text/javascript" src="https://ajax.googleapis.com/
    ajax/libs/jquery/1.11.0/jquery.min.js"></script>
```

Stejně jako u jazyku CSS bývá dobrým zvykem oddělit funkčnost stránky, tedy kódy psané v JavaScriptu, od vlastního obsahu a zapisovat funkce do externího souboru s koncovkou *.js*. V některých dokumentech bývá toto chování označováno jako princip nevtíráváho JavaScriptu. Výběr prvku v HTML stránce probíhá přes selektory, před nimiž musí být umístěn znak dolaru, vymezující jmenný prostor knihovny. Typická syntax obsahuje nejdříve element, který chceme vybrat, poté název události a nakonec to, co chceme udělat. Následující příklad demonstruje praktické použití některých metod. [16]

```
$( "#one" ).click(function() {
    $( "#pan_one" ).slideToggle("linear");
    $(this).toggleClass("toggle_active");
    $( "#one .fa-plus" ).toggleClass("fa-minus");
});
```

Interpretace kódu je následující: HTML prvku s identifikátorem one je přiřazena metoda *click*. Po kliknutí na tento element se spustí metoda *slideToggle*, která za použití jednoduché animace skryje či zobrazí element s identifikátorem pan_one. Metoda *toggleClass* mění jednotlivým elementům třídu a tím umožňuje měnit vzhled.

3.4.2 Kalendar plugin

Kalendář v aplikaci slouží k přehlednému zobrazení chystaných akcí v Orlických horách. K vizualizaci a správě kalendáře je použit jQuery plugin, který napsal švédský programátor Eric Wennerberg. Kalendář je možné přizpůsobit vzhledu vlastních webových stránek, nastavit české názvy dnů a měsíců, přidávat jedno i vícedenní události nebo integrovat události z vlastního kalendáře Google. Pro poslední možnost je nutné použít klíč Google Calendar API. Dostupná je verze Free a verze Pro, která je poskytovaná za drobný poplatek 10 dolarů, podporující další vývoj kalendáře. [30]

Pro použití kalendáře na vlastních stránkách stačí vložit řádek s adresou skriptu do hlavičky zdrojového kódu a kalendář inicializovat, určit tedy element *div*, kde se kalendář zobrazí.

```
<script src="http://ericwenn.se/kalendar/free/"></script>

<script type="text/javascript">
$(container).kalendar();
</script>
```

Kalendář má pouze jeden nepovinný parametr určující jeho vlastnosti (barva, první den v týdnu, názvy) a další možnosti stylování včetně přidávání událostí. Následující příklad ukazuje integraci s kalendářem Google, který byl pro aplikaci sportovních aktivit v Orlických horách speciálně vytvořen.

```
var options = {
    googleCal: [{ 
        calendar: "sportsactivityoh@gmail.com",
        apikey: "AIzaSyCPSWAKdmUqiPY4SezRbhvFp0SRaMnyseg"
    }]
};
```

3.5 Příprava a import turistických tras

Vektorová data turistických, cyklistických a běžkařských tras poskytlo pro tvorbu a prezentaci této bakalářské práce kartografické vydavatelství SHOCart ve spolupráci se společností T-Mapy, která mi pomohla konkrétní data zpracovat, a se kterou byla následně podepsána licenční smlouva. Veškerá poskytnutá data, použitá v této práci, byla do vektorové podoby převedena z grafických prvků tematických náplní map v měřítku 1:50 000. Jednotlivé vrstvy byly navíc transformovány ze souřadnicového systému S-42, využívající Gauss Krügerovo zobrazení na Krasovského elipsoidu, do souřadnicového systému JTSK s orientací a označením os (-y, -x) vůči S-JTSK (y, x). Liniové i bodové vektorové vrstvy byly uloženy ve formátu Esri shapefile.

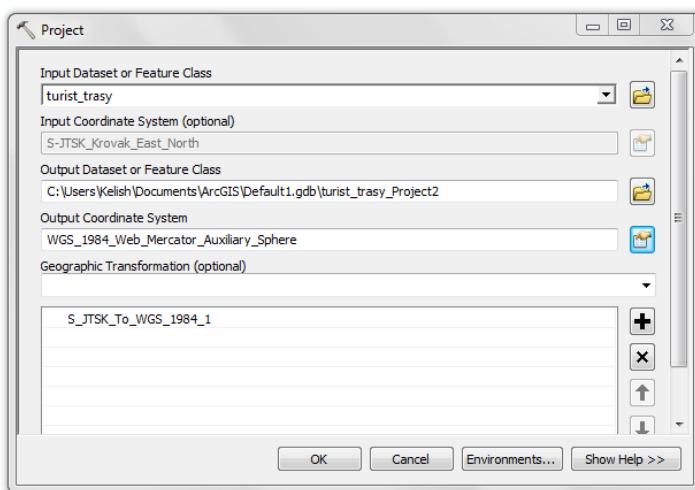
Pro další práce s daty byly použity produkty americké společnosti Esri, která dlouhá léta patří mezi největší producenty softwaru určeného pro práci s geografickými informačními systémy, zkráceně GIS. Tyto produkty jsou známy pod souhrnným označením ArcGIS a nabízí desktopové, serverové i mobilní řešení. Desktopová verze ArcGIS for Desktop pomáhá data organizovat, vytvářet, upravovat a analyzovat, zatímco úlohou serverového řešení ArcGIS for Server je data publikovat a zpřístupnit funkcionality GIS po internetu. Pro využití služeb ArcGIS serveru není nutné vlastnit specializované softwary ani výkonné počítače, neboť celé zpracování probíhá na serveru. Toho využívají webové a mobilní aplikace, pro které se vžilo spojení „lehký klient“. [1]

3.5.1 Úprava dat

Pro publikování a přidání vrstev do prostředí Google Maps API bylo zapotřebí data drobně upravit. Poskytnutá data vznikla ořezáním původních dat podle hranic zájmového území, čímž ale na hranicích tohoto území vznikly velmi krátké liniové prvky, pozůstatky tras. Tyto prvky byly vybrány a z liniové vrstvy odstraněny.

Aby byla zajištěna kompatibilita s mapovými podklady Googlu, bylo potřeba data transformovat do souřadnicového systému Web Mercator. Tento systém nevyužívá pouze Google, ale většina populárních poskytovatelů mapových obsahů, jako je Microsoft® Bing™ nebo ArcGIS Online. V podstatě se ale dá konstatovat, že většina těchto společností na Google navazuje.

V programu ArcGIS Desktop je pro transformaci z jednoho do druhého souřadnicového systému nejvhodnější zvolit nástroj *Project*, který se ukrývá v ArcToolboxu - Data Management Tools - Projections and Transformations - Feature. Nástroj vytvoří kopii stávajících dat a původní souřadnicový systém nahradí systémem novým. V tomto konkrétním případě S-JTSK systémem, který je v ArcGIS označovaný jako WGS1984 Web Mercator Auxiliary Sphere.

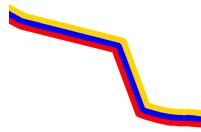


Obrázek 3.4: Transformace souřadnic - ArcGIS Project Tool

3.5.2 Publikační služby ArcGIS Serveru a Google Maps API

Po úpravě, transformaci a kontrole správné funkčnosti byl výstup publikován jako dynamická mapová služba ArcGIS Serveru. Pro tento způsob publikování dat jsem se rozhodl proto, abych při zobrazování v mapové aplikaci zachoval u souběžných tras nastavené posuny, takzvané offsety. Dynamická mapová služba se vykresluje při každé změně zobrazení, zatímco její alternativa, statická mapová služba, načítá předem připravené dlaždice a tím zrychluje komunikaci mezi klientem a serverem. Pro překryvnou vrstvu turistických tras je však tato varianta nevhodná.

Je škoda, že Google Maps API dodnes začlenění webové mapové služby přímo nepodporuje, ale na druhou stranu je možné využít JavaScriptové knihovny třetích stran. První takovou knihovnou, o které se zde zmíním, je knihovna s výstižným názvem ArcGIS Extension for the Google Maps API. Rozšíření nabízí možnost využít služeb ArcGIS serveru pro vizualizaci a analýzu dat nad mapovými podklady Googlu. JavaScriptová knihovna a



Obrázek 3.5: Posun u souběžných tras

její oficiální dokumentace obsahuje velmi mnoho užitečných funkcí a metod, které jsou ale postaveny na starší a dnes již nepodporované verzi Google Maps API. Společnost ESRI se o další aktualizaci stále nevyjádřila, a tak se můžeme domnívat, že vývoj této knihovny pokračovat nebude.

Další možností je využití knihovny ArcGIS Server Link for Google Maps JavaScript API V3, která je vydávána v rámci open source projektu široké skupiny vývojářů. Tato knihovna podporuje přidání mapových zdrojů přes ESRI ArcGIS Server REST API. Skripty, kompletní dokumentace a praktické ukázky využití jsou na oficiálních stránkách tohoto projektu. [6]

K přidání webové mapové služby vedou dva jednoduché kroky. V prvním kroku je nutné přidat stažený skript nebo odkaz na hostovanou verzi do hlavičky dokumentu, vždy až po odkazu na Google Maps API. Druhý krok již zahrnuje přidání dynamické mapové služby poskytované ArcGIS Serverem. Pro implementaci této služby je určen objekt *MapOverlay*. Ten se velmi podobá oficiálnímu objektu *google.maps.GroundOverlay*, který nad mapu přidává statický snímek.

```
var url = "http://geo103b.fsv.cvut.cz:6080/arcgis/rest/services/
  klima/data/MapServer"
layer = new gmaps.ags.MapOverlay(url);
layer.setMap(map);
```

3.6 Virtuální prohlídka a Google Earth API

Prostřednictvím pluginu Google Earth a jeho vývojářskému rozhraní JavaScript API je velice snadné vytvářet poutavé webové stránky s možností procházet a zkoumat zeměpisná data na 3D zeměkouli. Pro zobrazení virtuálního glóbu ve webovém prostředí je nutné použít jeden z podporovaných internetových prohlížečů a mít nainstalovaný plugin Google Earth. Aktuální seznam podporovaných prohlížečů a návod na instalaci pluginu jsem již podrobně popsal v kapitole 1.1 na straně 5. Zajímavým srovnáním dostupných technologií pro vizualizaci prostorových modelů se ve své diplomové práci zabývá Miroslav Kopecký. [17]

Pomocí Google Earth API je možné využít téměř všechny nástroje desktopové verze a zobrazovat tak na stránkách 3D modely budov, nebo si vytvořit svoji vlastní virtuální

prohlídku. V následujícím textu se podrobněji zaměřím na integraci okna Google Earth a virtuální prohlídky do své vlastní aplikace.

3.6.1 Vytvoření okna s prohlídkou

Stejně jako u Google Maps API je nutné do hlavičky stránky vložit odkaz na JavaScript soubor, který definuje jednotlivé metody a vlastnosti. Vlastní zdrojový kód lze psát do samostatného souboru a tento soubor v hlavičce opět načíst, nebo kód umístit do hlavního dokumentu mezi tagy `<script>` a `</script>`. Nejdůležitější metoda, `google.load`, načítá jednotlivé funkce rozhraní a má tři povinné parametry - co se má načíst, použitou verzi API a zda je pro určení polohy povoleno používat sensor GPS. Nad třetím parametrem mám trochu otazník, neboť Google Earth plugin není podporován žádným mobilním zařízením. Dále je nutné vytvořit instanci pluginu a dvě funkce. První funkce je volána po úspěšném načtení pluginu, a funkce druhá v případě, že se načítání nezdaří. Další nastavení ovlivňující vzhled a chování stránky se umisťují právě do první funkce `initCB()`. Je možné přidávat datové vrstvy, například 3D budovy, stromy nebo silniční síť, a upravovat ovládací prvky pro práci s mapou. [7] Základní skript obsahující nově přidané vrstvy a ovládací prvky vypadá následovně:

```

<script type="text/javascript">
    var ge;
    google.load("earth", "1", {"other_params": "sensor=false"});

    function init() {
        google.earth.createInstance('map3d', initCB, failureCB);
    }

    function initCB(instance) {
        ge = instance;
        ge.getWindow().setVisibility(true);
        ge.getNavigationControl().setVisibility(ge.VISIBILITY_AUTO);
        ge.getTime().getControl().setVisibility(ge.VISIBILITY_SHOW);
        ge.getSun().setVisibility(true);
        ge.getLayerRoot().enableLayerById(ge.LAYER_BUILDINGS, true);
        ge.getLayerRoot().enableLayerById(ge.LAYER_TERRAIN, true);
        ge.getLayerRoot().enableLayerById(ge.LAYER TREES, true);
    }

    function failureCB(errorCode) {"Something is wrong"}
    google.setOnLoadCallback(init);
</script>

```

Virtuální prohlídku, spustitelnou v pluginu, je nutné nejprve vytvořit v aplikaci Google Earth a následně exportovat do souboru KML. Při vytváření prohlídky, která sleduje trasu, lze nastavit úhel naklonění kamery, rozsah kamery a rychlosť prohlídky. Tento soubor již může být vstupem pro metodu `ge.getTourPlayer`, obstarávající načtení a ovládání prohlídky. Důležité je, aby byl tento soubor umístěn na veřejně přístupné adrese. [7]Inicializace tohoto objektu je velmi snadná:

```
var href = 'http://www.sportsactivityoh.wz.cz/data/tours/view0.kml';
google.earth.fetchKml(ge, href, kmlFinishedLoading);
function kmlFinishedLoading(object) {
    ge.getTourPlayer().setTour(object);
}
```

Takto vytvořená prohlídka se v aplikaci zobrazuje pomocí jQuery knihovny FancyBox.

3.6.2 Zobrazení prohlídek v mapě

Pro tvorbu prohlídek jsem zvolil pouze některé úseky z původní sítě turistických a cyklistických tras, přičemž celková délka jednoho úseku se pohybuje v rozmezí od tří do pěti kilometrů. Pro vizualizaci těchto úseků bylo nezbytné využít třídu `google.maps.Polyline`, která jako jediná nabízí události `mouseover` a `mouseout`. Ty jsou důležité pro změnu grafické podoby linie po najetí myši, čímž se odliší právě vybraný úsek. Vytváření jednotlivých polylinií jsem si zjednodušil cyklem, který přes Fusion Tables prochází KML souborem a separuje dílčí souřadnice lomových bodů. [10]

```
/*FUSION TABLES*/
var script = document.createElement('script');
var url = ['https://www.googleapis.com/fusiontables/v1/query?'];
url.push('sql=');
var query = 'SELECT type, geometry FROM ' + '1
WvbxFNqRitgCqFeZHCJjVENN3aoa-taJe4sZxn0i';
var encodedQuery = encodeURIComponent(query);
url.push(encodedQuery);
url.push('&callback=drawPolyline');
url.push('&key=AIzaSyBkGpteF7vhC9Y_qMWTOKAZVVNVYPQwC8');
script.src = url.join('');
var body = document.getElementsByTagName('body')[0];
body.appendChild(script);
index = 0;

function drawPolyline(data) {
    var rows = data['rows'];
    for (var i in rows) {
```

```
var geometries = rows[i][1]['geometries'];
if (geometries) {
    for (var j in geometries) {
        var newCoordinates = constructNewCoordinates(
            geometries[j]);
        layers [index]= new google.maps.Polyline({
            path: newCoordinates,
            strokeColor: "#19A3FF",
            strokeOpacity: 0.4,
            strokeWeight: 5,
            geodesic: true
        });
    }
} else {
    var newCoordinates = constructNewCoordinates(rows[i]
        ][1]['geometry']);
    layers [index] = new google.maps.Polyline({
        path: newCoordinates,
        strokeColor: "#19A3FF",
        strokeOpacity: 0.4,
        strokeWeight: 5,
        geodesic: true
    });
}
index = index + 1;
}

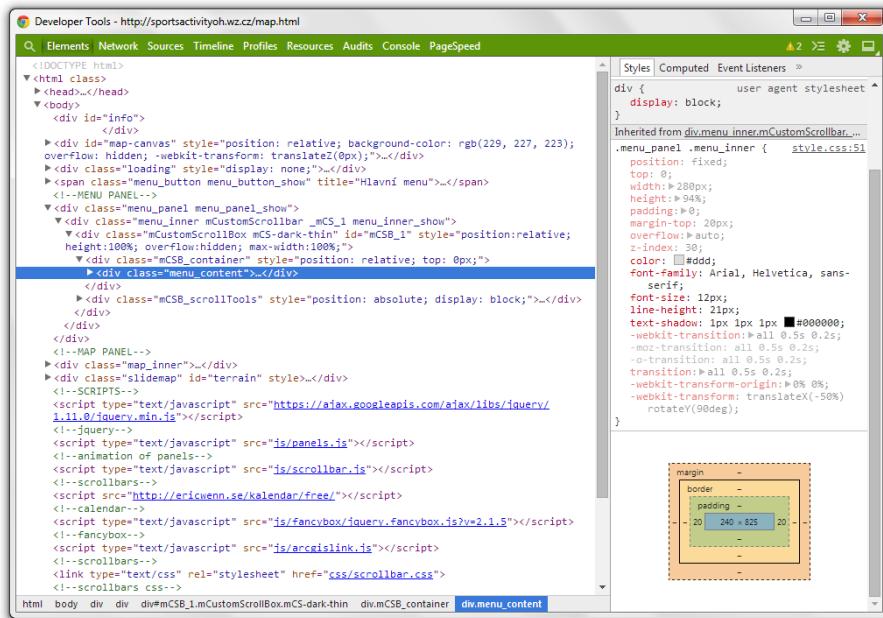
function constructNewCoordinates(polyline) {
    var newCoordinates = [];
    var coordinates = null;
    if (polyline['coordinates'])
        coordinates = polyline['coordinates'];
    if (coordinates.length == 1) {
        coordinates = coordinates[0];
        alert("length = 1");
    };
    for (var i in coordinates) {
        newCoordinates.push(
            new google.maps.LatLng(coordinates[i][1], coordinates[i]
                [0]));
    };

    return newCoordinates;
};
```

3.7 Testování aplikace

Většina dnešních moderních prohlížečů nabízí integrované nástroje pro ladění webových aplikací, což vývojářům znatelně ulehčuje práci. Nástroje se v různých prohlížečích liší, ale funkce pro testování a ladění webu zůstávají téměř stejné a při jejich používání jde tedy jen o zvyk. Osobně jsem zůstal věrný prohlížeči Google Chrome a nebyl důvod tohoto kroku litovat.

Nástroje pro vývojáře je možné spustit klávesovou zkratkou *Ctrl + Shift + I* nebo výběrem z nabídky přes *Nástroje - Nástroje pro vývojáře*. Všechny dostupné funkce jsou přehledně zobrazeny v horní liště vývojářské konzole a hned první záložku *Elements* jsem využíval asi nejčastěji. Tato funkce umožňuje přímo v prohlížeči manipulovat s HTML DOM elementy a měnit jim v CSS styl. Tyto změny se projeví pouze v prohlížeči a to bez nutnosti stránku znova nahrávat, přičemž původní zdrojový kód zůstane nedotčen. Takto jsem například testoval rozložení a grafickou podobu prvků. Dále jsem ocenil nástroj skrývající se pod záložkou *Sources*. Tento nástroj umožňuje procházet JavaScriptové funkce krok po kroku a jednodušeji tak odhalit chybu. Z dalších nástrojů jsem pro závěrečná ladění využil grafické znázornění časové složitosti jednotlivých požadavků aplikace a informace o celkovém objemu přenesených dat. [19]



Obrázek 3.6: Nástroj vývojáře Google Chrome

Aplikace z pohledu běžného uživatele

Tato kapitola má za účel seznámit běžného uživatele s prostředím aplikace mapující sportovní aktivity v Orlických horách. Obsahuje podrobný popis uživatelského rozhraní včetně popisu jednotlivých nástrojů a ovládacích prvků, které umožňují interaktivní práci s mapou. Pro používání aplikace doporučuji využít nejnovější verze webových prohlížečů Google Chrome a Opera, na kterých byla aplikace testována, a na kterých je zaručena stoprocentní funkčnost. V případě starších verzí či jiných prohlížečů se může stát, že funkčnost aplikace bude z části omezena. V některých situacích pomůže doinstalování požadovaných pluginů, jako je například plugin Google Earth. Aplikaci je také možné spustit na mobilních zařízeních s připojením k internetu. Zde je ale bohužel nedostupná funkce virtuálních prohlídek, neboť Google Earth plugin podporu mobilních zařízení dosud nenabízí.

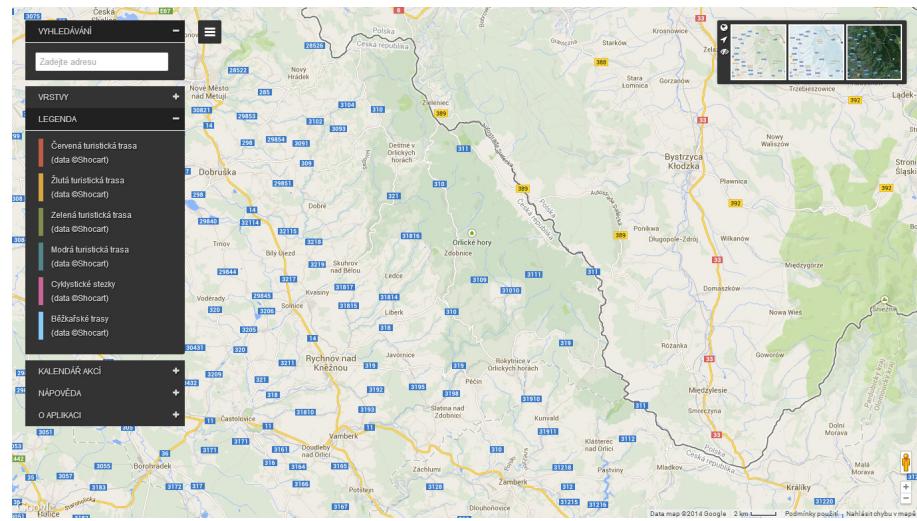
4.1 Seznámení se s uživatelským rozhraním

Při každém spuštění aplikace je nutné vyčkat na načtení mapových podkladů. Tato akce by však v závislosti na rychlosti internetového připojení neměla trvat déle, než několik málo sekund. Po úspěšném načtení stránky se zobrazí hlavní okno aplikace s mapou a dvěma ovládacími panely. V levém ovládacím panelu se pod vysouvacími lištami nachází následující nástroje:

- Vyhledávání,
- Vrstvy,
- Legenda,
- Kalendář akcí,
- Ná pověda,
- O aplikaci.

Pokud používáte pouze jeden nástroj a zbylé vás matou, je možné tyto nástroje kliknutím na znaménko minus – skrýt. Opětovné otevření nástroje provedete kliknutím na

znaménko plus +. Pro pohyb mezi nástroji je k dispozici posuvná lišta, která se automaticky zobrazí a skryje, v závislosti na počtu otevřených nástrojů. Posunovat lze přímo lištou nebo kliknutím na šipky znázorňující požadovaný směr posunu. Pokud nevyužíváte žádný z nástrojů a raději chcete pracovat s mapou, můžete celý levý panel schovat kliknutím na tlačítko, které se nachází v pravém horním rohu panelu.



Obrázek 4.1: Hlavní okno aplikace

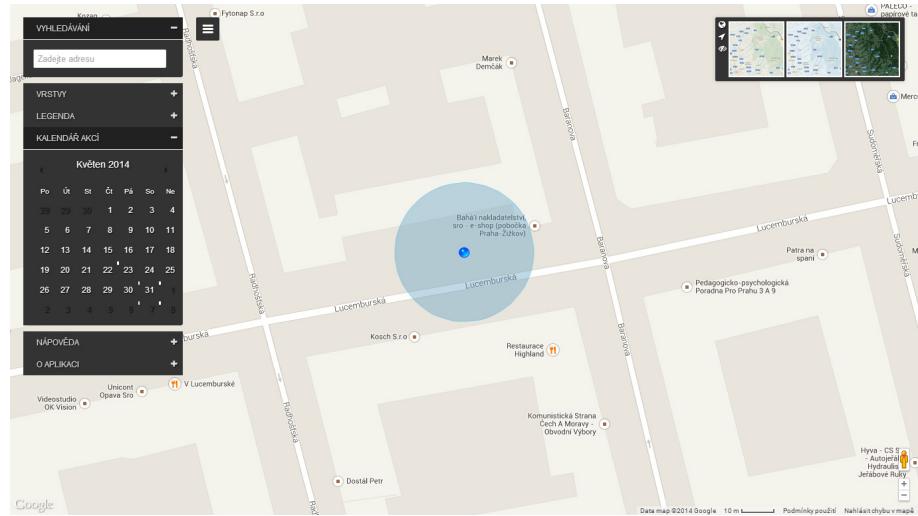
Druhý ovládací panel umístěný v pravém horním rohu okna je určen pro výběr mapového podkladu. Uživatel má na výběr hned ze čtyř možností:

- Běžná mapa,
- Zimní mapa,
- Terénní mapa,
- Satelitní snímky.

4.2 Sledování polohy

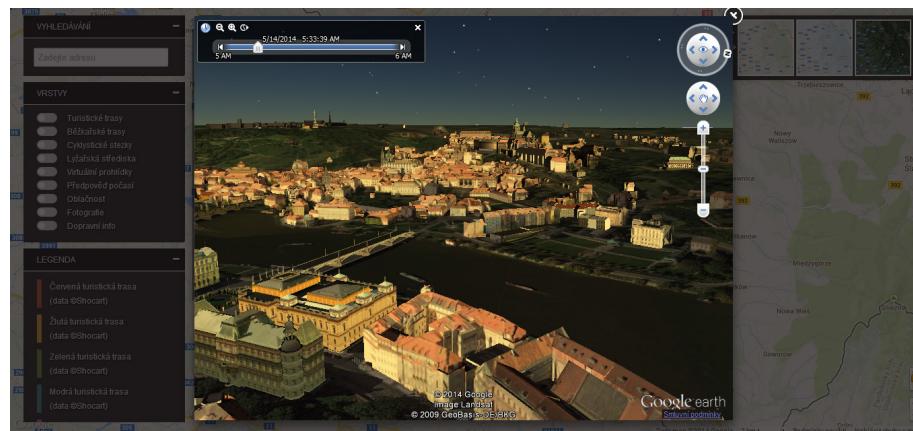
Součástí tohoto panelu jsou i nástroje pro spuštění služby sledování polohy nebo aktivaci okna s 3D virtuálním glóbem. Pro aktivaci sledování polohy je určena ikona ve tvaru šipky. Po kliknutí na tuto ikonu se mapa automaticky přesune na vaši pozici. Pro využití této služby je nutné používat aktuální verze internetových prohlížečů a sledování polohy povolit - žádost o povolení se automaticky objeví u horního okraje okna. Po zíce je v mapě vizualizována modrou tečkou, kolem které je modrá průhledná kružnice, znázorňující přesnost určení polohy. Podrobnosti o poloze, včetně zeměpisných souřadnic

a přesnosti jejího určení, vyvoláte kliknutím na modrou tečku. Mapou můžete dále pochybovat, a pokud byste se chtěli vrátit zpět na svou pozici, stačí opět stisknout tlačítko šipky. Sledování polohy lze vypnout ikonou připomínající přeškrtnuté oko.



Obrázek 4.2: Sledování polohy

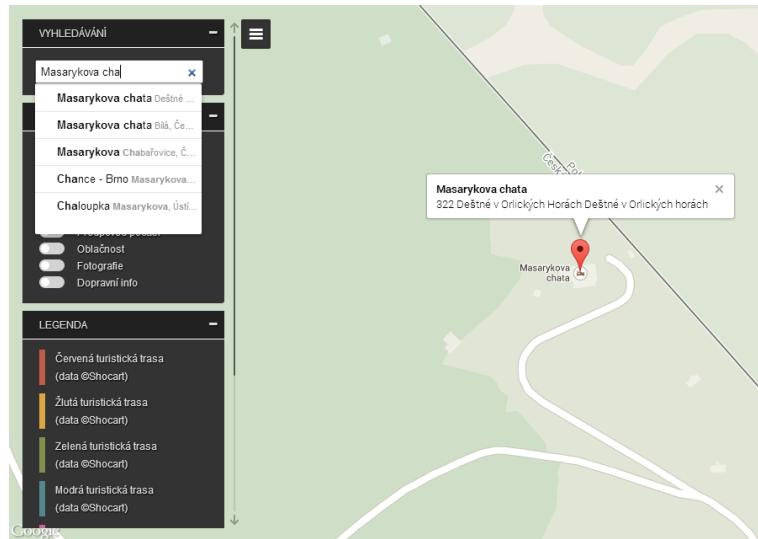
Poslední z trojice ikon slouží pro otevření překryvného okna s 3D modelem Země. Při otevření se automaticky přesunete na vaši aktuální pozici. Prostřednictvím tohoto okna je možné si prohlížet 3D modely budov, procházet se terénem, pozorovat noční oblohu nebo cestovat v čase. Ovládání je velmi intuitivní a lze využít jak ovládání pomocí myši, tak pomocí integrovaných ovládacích prvků. Tento mód se vypíná tlačítkem X v pravém horním rohu.



Obrázek 4.3: 3D model Země

4.3 Vyhledávací pole

Vyhledávací pole je určeno pro vyhledávání nejrůznějších míst, adres, stravovacích a ubytovacích zařízení. Při psaní do vyhledávacího pole se automaticky zobrazuje nabídka míst, která jsou podobná napsanému dotazu. Po zvolení požadovaného místa se do mapy umístí značka s dalšími podrobnostmi. Pro vymazání výrazu stačí kliknout na křížek.

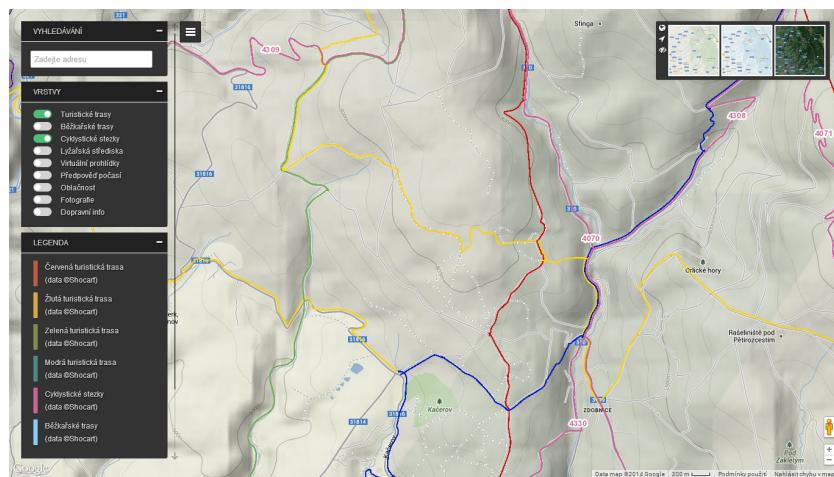


Obrázek 4.4: Vyhledávání míst a adres

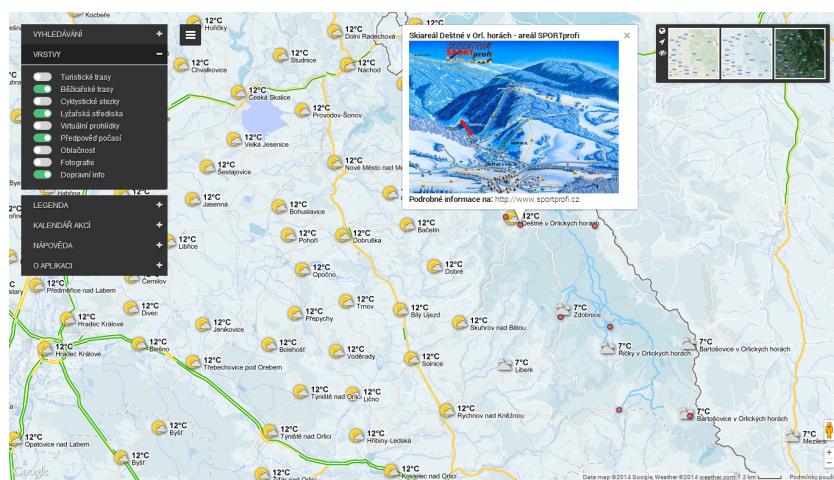
4.4 Přepínání vrstev

Dalším nástrojem na ploše je panel umožňující vypínání či zapínání překryvných vrstev, jako jsou turistické trasy, běžkařské trasy a cyklistické stezky nebo vrstvy s aktuálním stavem a předpovědí počasí na 4 dny dopředu. Vrstva s fotografiemi zobrazuje fotografie ze služby Panoramio. Tyto fotografie jsou opatřeny souřadnicemi místa pořízení, takzvanými „geotagy“. Pokud byste chtěli, aby se i vaše fotografie v mapě zobrazovaly, přejděte na stránky www.panoramio.com a nahrajte pořízené fotografie. Jednotlivé vrstvy lze aktivovat kliknutím na jezdce nalevo od názvu vrstvy. Ten se plynule posune a jeho pozadí se zbarví do zelená.

V aplikaci je také integrována velmi populární služba StreetView nabízející panoramatické pohledy. Chcete-li zobrazit snímky StreetView, klikněte na ikonu panáčka v pravém dolním rohu a přetáhněte ji na požadované místo v mapě.



Obrázek 4.5: Ukázka turistických a cyklistických tras



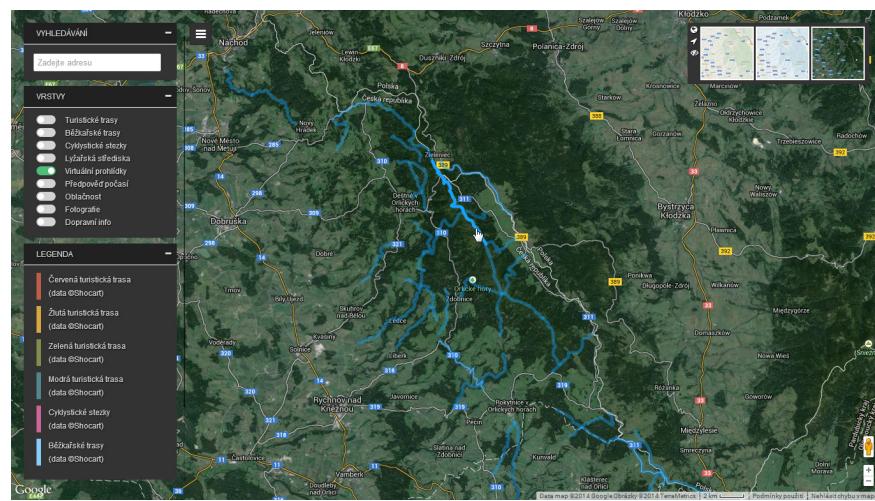
Obrázek 4.6: Lyžařské areály, běžkařské tratě, počasí a dopravní info nad zimní mapou



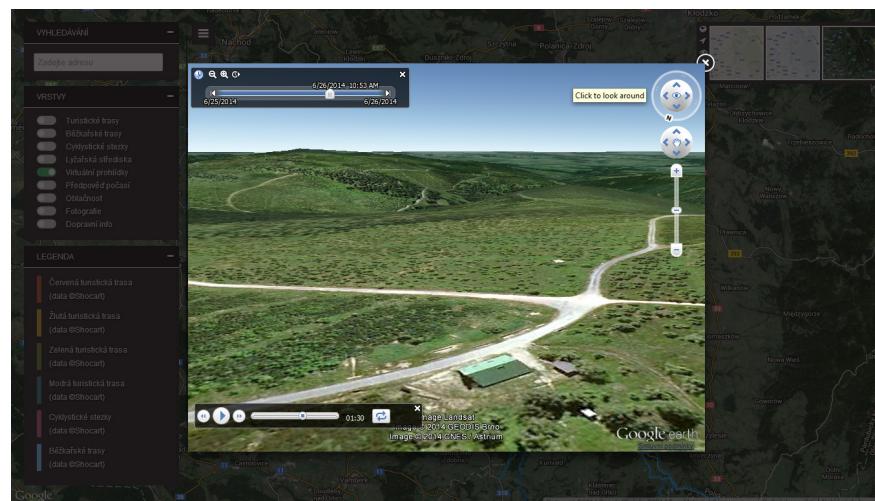
Obrázek 4.7: Google StreetView

4.4.1 Virtuální prohlídky

Virtuální prohlídky vám umožní plynule přelétnout z jednoho místa na jiné, zobrazení terén a libovolně se rozhlížet po okolí. Po zapnutí vrstvy *Virtuální prohlídky* a vybrání trasy můžete prozkoumat detaily terénu z pohodlí vašeho domova. V levém dolním rohu 3D prohlížeče jsou umístěny ovládací prvky, díky nimž můžete průběh prohlídky ovlivňovat. Prohlídku spusťte nebo pozastavte kliknutím na tlačítko Pozastavit/Přehrát. Prohlídku můžete také zrychlit a zpomalit opakovaným stisknutím tlačítka šipek, nebo se pomocí posuvníku přesunout na jakoukoliv část prohlídky. Během prohlídky také můžete měnit úhel a směr pohledu, ale také se pomocí nástroje v levém horním rohu přesunout v čase a pozorovat východ slunce či zdánlivý pohyb hvězd po obloze.



Obrázek 4.8: Výběr virtuální prohlídky



Obrázek 4.9: Virtuální prohlídka

4.5 Kalendář akcí

V kalendáři je možné vyhledávat plánované výstavy, slavnosti a sportovní události v Orlických horách. Je možné procházet jednotlivé měsíce i dny. Den s událostí je v kalendáři zvýrazněn bílým obdélníčkem. Po rozkliknutí se zobrazí podrobnější informace obsahující název události, termín konání a místo konání. Pro návrat na měsíční přehled stačí kliknout na číslo dne.



Obrázek 4.10: Kalendář akcí

ZÁVĚR

Cílem této bakalářské práce bylo popsat možnosti vývojového rozhraní Google Maps JavaScript API a s jeho využitím vytvořit funkční webovou mapovou aplikaci. V rámci práce byly také představeny základní principy tvorby webových aplikací se zaměřením na programovací jazyky HTML, CSS a JavaScript. Více jsem se pak zaměřil na popis geolokačního rozhraní Geolocation API, které je součástí nově představené specifikace HTML5. Krátce jsem se také zmínil o experimentálním nástroji Google Fusion Tables a jeho využití pro vizualizaci geoprostorových dat v mapě.

Výsledná aplikace by mohla sloužit sportovním nadšencům a ostatním návštěvníkům Orlických hor, kteří si chtějí z pohodlí domova naplánovat směr své cesty. To jim usnadní síť značených turistických, cyklistických a běžkařských tras, ale také virtuální prohlídky, díky kterým se budou moci vydat přímo na hřebeny hor. Aplikace je plně kompatibilní s mobilními zařízeními mající stálé připojení k internetu, což činí tuto aplikaci použitelnou i v terénu. Nepostradatelným nástrojem by pak pro mnohé mohlo být sledování polohy a zobrazení aktuální pozice v mapě.

Pro účely testování a prezentace webové mapové aplikace byla vytvořena internetová stránka www.sportsactivityoh.wz.cz, využívající hostingové služby serveru webzdarma. Z důvodu zamezení neoprávněného použití poskytnutých dat a dodržení licenčních podmínek bylo nutné tyto stránky zabezpečit přihlašovacím jménem:xxxxxxxxx a heslem:xxxxxxxxxxxxxx .

Ačkoliv vývojové prostředí Google Maps nabízí širokou škálu funkcí a metod, byl jsem v některých konkrétních příkladech nucen využít alternativní řešení. Takovým problémem bylo například vložení značených turistických tras, které jsou v některých úsecích vzájemně souběžné. Při jejich exportu do souboru KML a následném vložení do mapy vznikala velmi chaotická situace, kdy se jednotlivé trasy překrývaly. Dlouhou dobu jsem nemohl najít řešení, které by toto odsazení souběžných tras zachovalo. Situace byla nakonec vyřešena publikováním dat na ArcGIS server a použitím knihovny ArcGIS Server Link for Google Maps JavaScript API V3, která do map umožňuje přidat webové mapové služby tohoto serveru. Další komplikaci přineslo vytváření liniové vrstvy, která by reagovala změnou grafické podoby, při každém přejetí myši. Řešení tohoto problému jsem včetně zdrojového kódu popsal v kapitole 3.6.2 na straně 44.

Během tvorby webové mapové aplikace vznikaly další nápady na vylepšení, které se však v rámci této bakalářské práce nepodařilo uskutečnit. V budoucnu bych chtěl například obohatit aplikaci o možnost zobrazení výškového profilu trasy. Dalším plánem je vytvořit uživatelské rozhraní pro editaci a přidávání vlastních tras exportovaných z přenosných GPS přístrojů.

Literatura

- [1] ARCDATA PRAHA: *Produkty ArcGIS* [online]. 2014 [cit. 2014-04-19]. Dostupné z: <http://www.arcdata.cz/produkty-a-sluzby/software/arcgis/>
- [2] ŠŤASTNÝ, Jiří: *HTML5 geolokační rozhraní* [online]. 2012 [cit. 2014-04-11]. Dostupné z: <http://programujte.com/clanek/2011052400-html5-geolokacni-rozhrani/>
- [3] BUCHAR, Petr: *Matematická kartografie*. Nakladatelství ČVUT, 2007, ISBN 978-80-01-03765-2.
- [4] DINCER, Alper a URAZ, Balkan: *Google Maps JavaScript API Cookbook*. Packt Publishing, 2013, ISBN 978-1-84969-882-5.
- [5] GEORSS: *Geographically Encoded Objects for RSS feeds* [online]. 2013 [cit. 2014-04-05]. Dostupné z: <http://www.georss.org/>
- [6] GOOGLE CODE: *ArcGIS Server Link for Google Maps API V3* [online]. 2014 [cit. 2014-04-19]. Dostupné z: <http://google-maps-utility-library-v3.googlecode.com/svn/trunk/arcgislink/docs/examples.html>
- [7] GOOGLE DEVELOPERS: *Google Earth API* [online]. 2012 [cit. 2014-04-24]. Dostupné z: <https://developers.google.com/earth/>
- [8] GOOGLE DEVELOPERS: *Hello map* [online]. 2014 [cit. 2014-02-25]. Dostupné z: <https://developers.google.com/maps/>
- [9] GOOGLE DEVELOPERS: *Google Maps API licensing* [online]. 2014 [cit. 2014-02-28]. Dostupné z: <https://developers.google.com/maps/licensing>
- [10] GOOGLE DEVELOPERS: *Google Maps Javascript API v3* [online]. 2014 [cit. 2014-03-02]. Dostupné z: <https://developers.google.com/maps/web/>
- [11] GOOGLE Inc.: *Fusion Tables* [online]. 2013 [cit. 2014-04-06]. Dostupné z: <https://support.google.com/fusiontables/>
- [12] GOOGLE Inc.: *O společnosti Google* [online]. 2014 [cit. 2014-02-10]. Dostupné z: http://www.google.cz/intl/cs_cz/about/
- [13] GOOGLE Inc.: *Nápověda Google Earth* [online]. 2014 [cit. 2014-02-15]. Dostupné z: <https://support.google.com/earth/>
- [14] GOOGLE Inc.: *Nápověda Google Maps* [online]. 2014 [cit. 2014-02-16]. Dostupné z: <https://support.google.com/maps/>

- [15] JANOVSKÝ, Dušan: *Jak psát web: o tvorbě internetových stránek* [online]. 2014 [cit. 2014-03-01]. Dostupné z: <http://www.jakpsatweb.cz/>
- [16] JQUERY: *The jQuery Foundation* [online]. 2014 [cit. 2014-04-13]. Dostupné z: <https://jquery.org/>
- [17] KOPECKÝ, Miroslav: *Možnosti interaktivní prezentace prostorových modelů na internetu*. Praha, 2013. Diplomová práce. ČVUT v Praze, Fakulta stavební.
- [18] KOSEK, Jiří: *Historie a vývoj HTML* [online]. 2013 [cit. 2014-04-10]. Dostupné z: <http://htmlguru.cz/uvod-historie.html>
- [19] MAŇÁK, Michal: *Ladíme webové stránky přímo v prohlížeči Google Chrome* [online]. 2010 [cit. 2014-05-01]. Dostupné z: <http://www.manakmichal.cz/blog/prohlizece/ladime-webove-stranky-primo-v-prohlizeci-google-chrome/>
- [20] MELLEN, Mickey: *How often does Google update the imagery in Google Earth* [online]. 2010 [cit. 2014-02-18]. Dostupné z: http://www.gearthblog.com/blog/archives/2010/10/how_often_does_google_update_the_im.html
- [21] OPEN GEOSPATIAL CONSORTIUM: *KML standards* [online]. 1994 - 2014 [cit. 2014-04-05]. Dostupné z: <http://www.opengeospatial.org/standards/kml>
- [22] PILGRIM, Mark: *Dive into HTML5* [online]. 2009 - 2011 [cit. 2014-04-11]. Dostupné z: <http://diveintohtml5.info/geolocation.html>
- [23] PROGRAMMABLEWEB: *API news* [online]. 2005 - 2014 [cit. 2014-02-25]. Dostupné z: <http://www.programmableweb.com/>
- [24] SCHNEIDER, F.; BLACHMAN, N.; FREDRICKSEN, E.: *How to Do Everything with Google*. McGraw Hill Professional, 2004, ISBN 0-07-223174-2.
- [25] STATCOUNTER: *Globals Stats* [online]. 1999 - 2014 [cit. 2014-02-10]. Dostupné z: <http://gs.statcounter.com/>
- [26] STROSS, Randall: *Planeta Google*. Computer Press, 2009, ISBN 978-80-251-2412-3.
- [27] SVENNERBERG, Gabriel: *Beginning Google Maps API 3*. Apress, 2010, ISBN 978-1-4302-2803-5.
- [28] W3C: *A Short History of JavaScript* [online]. 2012 [cit. 2014-04-13]. Dostupné z: https://www.w3.org/community/webed/wiki/A_Short_History_of_JavaScript
- [29] W3RESOURCE: *Web development tutorials* [online]. 2013 - 2014 [cit. 2014-04-11]. Dostupné z: <http://www.w3resource.com/>
- [30] WENNERBERG, Eric: *Kalendar* [online]. 2014 [cit. 2014-04-15]. Dostupné z: <http://www.ericwenn.se/kalendar/>
- [31] WESLEY, Addison: *A history of HTML* [online]. 1998 [cit. 2014-04-10]. Dostupné z: <http://www.w3.org/People/Raggett/book4/ch02.html>

- [32] WORLD WIDE WEB CONSORTIUM: *Geolocation API Specification* [online]. 2012 [cit. 2014-04-12]. Dostupné z: <http://dev.w3.org/geo/api/spec-source.html>
- [33] WORLD WIDE WEB CONSORTIUM: *HTML5 Candidate Recommendation* [online]. 2014 [cit. 2014-04-11]. Dostupné z: <http://www.w3.org/TR/html5>
- [34] WORLD WIDE WEB CONSORTIUM: *Standards and Drafts of CSS* [online]. 2014 [cit. 2014-04-13]. Dostupné z: http://www.w3.org/TR/#tr_CSS

Příloha A

Elektronické přílohy

Přiložené CD obsahuje elektronickou verzi tohoto dokumentu a zdrojové soubory webové mapové aplikace. Zdrojové kódy jsou dle účelu rozděleny do adresářů.

- icon - složka se styly ikon Font Awesome
- scrollbar.css - kaskádové styly pro posuvnou lištu
- style.css - kaskádové styly pro layout aplikace
- data - složka s KML soubory a HTML soubory prohlídek
- fancybox - funkce pro zobrazení ve Fancyboxu
- arcgislink.js - funkce pro použití s ArcGIS serverem
- google_maps.js - základní funkce pro práci s mapou a vrstvami
- panels.js - funkce pro dynamické efekty
- scrollbar.js - funkce pro práci s posuvnou lištou
- map.html - hlavní HTML dokument aplikace
- BP.pdf - bakalářská práce ve formátu PDF

