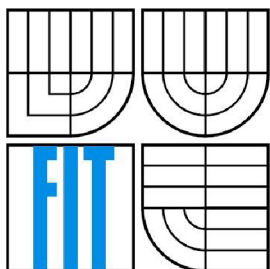


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMEDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# ZOBRAZENÍ VÝŠKOVÉHO PROFILU TRASY NA VEŘEJNÉM GISU

TRACK'S ELEVATION PROFILE VIEW ON A PUBLIC GIS

## BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

Štěpán Doubal

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. Michal Fapšo

BRNO 2009

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav počítačové grafiky a multimédií

Akademický rok 2008/2009

**Zadání bakalářské práce**

Řešitel: **Doubal Štěpán**

Obor: Informační technologie

Téma: **Zobrazení výškového profilu trasy na veřejném GISu**

Kategorie: Web

Pokyny:

1. Vyberte si libovolný veřejný GIS (mapy.cz, mapy.sk, ...) a zanalyzujte jeho zdrojový kód.
2. Navrhněte způsob získávání souřadnic z mapy, které uživatel nakliká pomocí nástrojů daného GISu. Souřadnice se budou získávat a dále zpracovávat pomocí GreaseMonkey rozšíření pro Firefox.
3. Naimplementujte bod 2 a zobrazení výškového profilu naklikané trasy pomocí služby <http://www.heywhatsthat.com/profiler.html>, nebo podobné.

Literatura:

- Dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Body 1, 2.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

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

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

Vedoucí: **Fapšo Michal, Ing.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2008

Datum odevzdání: 20. května 2009

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav počítačové grafiky a multimédií  
612 66 Brno, Žitná 2



doc. Dr. Ing. Jan Černocký  
vedoucí ústavu

## **Abstrakt**

Tato bakalářská práce se zabývá návrhem a implementací uživatelského skriptu pro GreaseMonkey (rozšíření pro Firefox), který umožňuje zobrazení výškového profilu zadané trasy. Trasa je zadána za použití nástrojů veřejného GISu [www.mapy.cz](http://www.mapy.cz). Skript získá souřadnice z DOM struktury webové stránky, dále je zpracovává a zobrazí výškový profil dané trasy. Vykreslení profilu je realizováno za pomoci služby serveru <http://www.heywhatsthat.com/profiler.html> [2] nebo podpůrného webu [www.profile.hy.cz](http://www.profile.hy.cz). Tento web byl vytvořen pro účely bakalářské práce. Umožňuje vykreslování výškového profilu v JavaScriptu za pomoci API [www.vyskopis.cz](http://www.vyskopis.cz) [4].

## **Abstract**

This bachelor's thesis describes the design and implementation of a user script for GreaseMonkey (plug-in for Firefox). This script adds a functionality to show height profile of a defined route. The route is defined using public GIS tools from [www.mapy.cz](http://www.mapy.cz). The script extracts coordinates from page DOM structure, processes it and draws a height graph. The graph is created using <http://www.heywhatsthat.com/profiler.html> [2] or [www.profile.hy.cz](http://www.profile.hy.cz), which provides height graph drawing using [www.vyskopis.cz](http://www.vyskopis.cz) JavaScript API [4].

## **Klíčová slova**

Geografický informační systém, referenční elipsoid, geoid, mapový server, DEM, souřadný systém, WGS84, UTM, S-JTSK, zeměpisná šířka, zeměpisná délka, nadmořská výška, NIMA, GTOPO, SRTM, API, GreaseMonkey, JavaScript, HTML, DHTML, PHP.

## **Keywords**

Geographic information system, reference ellipsoid, geoid, mapserver, DEM, system of coordinates, WGS84, UTM, S-JTSK, latitude, longitude, altitude, NIMA, GTOPO, SRTM, API, GreaseMonkey, JavaScript, HTML, DHTML, PHP.

## **Citace**

Štěpán Doubal: Zobrazení výškového profilu trasy na veřejném GISu, bakalářská práce, Brno, FIT VUT v Brně, 2009

# Zobrazení výškového profilu trasy na veřejném GISu

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Michala Fapša. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Štěpán Doubal  
Datum 25.05.2009

## Poděkování

Rád bych na tomto místě poděkoval vedoucímu práce Ing. Michalu Fapšovi za pomoc a odborné rady. Dále potom své rodině a přítelkyni, za jejich podporu.

© Štěpán Doubal, 2009

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



# Obsah

Obsah.....	1
1. Úvod.....	4
2. Geoinformatika a geografické informační systémy.....	5
2.1 Geoinformatika a co o ní víme.....	5
2.1.1 Geografický informační systém.....	5
2.2 GIS nástroje .....	5
2.3 Pojem mapový server.....	5
2.4 Fungování mapserveru.....	6
2.4.1 Protokol CGI.....	6
2.4.2 Rozhraní programovacích jazyků.....	6
2.5 Souřadnicové systémy.....	6
2.5.1 Související pojmy.....	6
2.5.1.1 Geoid.....	6
2.5.1.2 Referenční elipsoid.....	7
2.5.2 WGS84.....	7
2.5.2.1 Zeměpisná šířka a délka.....	8
2.5.3 UTM.....	8
2.5.4 S-JTSK.....	9
3. Analýza a návrh řešení.....	10
3.1 Dostupná výšková data.....	10
3.1.1 NIMA.....	10
3.1.2 GTOPO30.....	10
3.1.3 SRTM.....	10
3.1.4 Digitální výškový model Země.....	11
3.2 Možnosti získávání a zpracování topografických dat.....	11
3.2.1 Vytvoření vlastního mapového serveru.....	11
3.2.2 Využití veřejného serveru.....	12
3.2.2.1 Služba <a href="http://www.heywhatsthat.com/profiler.html">www.heywhatsthat.com /profiler.html</a> .....	12
3.2.2.2 API <a href="http://www.topocoding.com">www.topocoding.com</a> .....	13
3.2.2.3 API <a href="http://www.vyskopis.cz">www.vyskopis.cz</a> .....	13
3.2.2.4 Server <a href="http://latlontoelevation.com">latlontoelevation.com</a> .....	14
3.3 Server <a href="http://www.mapy.cz">www.mapy.cz</a> .....	14
3.3.1 API <a href="http://Mapy.cz">Mapy.cz</a> .....	15

4. Technologie použité k implementaci.....	16
4.1 JavaScript.....	16
4.2 DOM struktura webových stránek.....	16
4.3 GreaseMonkey.....	17
4.4 HTML a DHTML.....	18
4.5 PHP.....	18
4.6 Užívání API Mapy.cz .....	19
4.6.1 Licenční podmínky.....	19
4.6.2 Registrace.....	19
4.6.3 Vložení mapy do webové stránky.....	19
4.6.4 Souřadné systémy v API Mapy.cz.....	20
4.6.4.1 Převody mezi PP a UTM.....	20
4.6.4.2 Převody mezi PP a WGS84.....	20
4.6.4.3 Převody mezi UTM a WGS84.....	20
4.6.4.4 Souřadnice v pixelech.....	20
4.6.5 Přidání ovládacích prvků do mapy.....	21
4.6.6 Ovládání mapy pomocí myši.....	21
4.6.7 Nastavení středu mapy a přiblížení.....	21
4.6.8 Zobrazování mapových podkladů a vrstev.....	21
4.6.9 Práce s vektorovou mapovou vrstvou.....	23
5. Postup při implementaci.....	24
5.1 Uživatelský skript pro GM.....	24
5.1.1 Vytvoření nového skriptu.....	24
5.1.2 Odlišnosti GreaseMonkey skriptů.....	24
5.1.3 Přidání ovládacích prvků.....	24
5.1.4 Získání a zpracování zeměpisných souřadnic.....	25
5.2 Webová aplikace s mapou.....	27
5.2.1 Vložení mapy .....	27
5.2.2 Vykreslovací algoritmus.....	28
5.2.2.1 Zpracování parametrů.....	28
5.2.2.2 Kreslení profilu.....	28
5.2.3 Ukázky vykreslených výškových profilů.....	30
5.2.3.1 Porovnání vykreslovacích algoritmů.....	30
5.2.3.2 Různé velikosti obrázků.....	31
5.2.3.3 Porovnání různých přesností vykreslení.....	32

6. Závěr.....	33
Literatura.....	35
Seznam příloh.....	36

# 1. Úvod

V moderní době jsme svědky stále vzrůstajícího rozvoje informačních technologií, především internetu a webových služeb. Dalším odvětvím, které se také díky internetu čím dál více dostává do pořadí, jsou geografické informační systémy (GIS).

GIS je elektronický systém používaný ke zpracování, ukládání a zobrazování geografických dat [1]. Pracuje s prostorovými (geografická poloha objektu, nadmořská výška) i neprostorovými (zalidnění oblasti, nerostné bohatství, míra znečištění ovzduší) údaji.

Velkým přínosem pro rozšíření používání GISů je možnost práce s nimi pomocí webového rozhraní, není tedy nutné instalovat nákladný a složitý software, který by byl jinak nezbytný. O zpracování geografických dat se stará mapový server. Uživatel internetu má na svém počítači webový prohlížeč zajišťující zobrazení informací ze serveru na straně klienta. Téměř každý uživatel celosvětové počítačové sítě někdy využil služeb mapových serverů, jakými jsou [www.mapy.cz](http://www.mapy.cz), [www.maps.google.com](http://www.maps.google.com), [www.atlas.cz](http://www.atlas.cz) a mnohé další. Ačkoliv jsou tyto veřejné GISy na vysoké úrovni, zobrazení výškových dat, případně výškového profilu není tak obvyklé. Výškový profil můžeme zobrazit na volně dostupném serveru [2] nebo za pomoci API ze stránek [www.vyskopis.cz](http://www.vyskopis.cz) [3], které je po registraci také volně k dispozici.

U nás je jedním z nejpoužívanějších veřejných GISů server [www.mapy.cz](http://www.mapy.cz) od společnosti Seznam a.s. (dále jen Seznam).

Tato bakalářská práce se zabývá možností zobrazení výškového profilu pomocí služeb [www.mapy.cz](http://www.mapy.cz), [www.heywhatsthat.com/profiler.html](http://www.heywhatsthat.com/profiler.html) [2] a [www.vyskopis.cz](http://www.vyskopis.cz) [4]. Aby bylo možné zobrazit výškový profil trasy zadané pomocí nástrojů GISu, musíme získávat souřadnice zadaných bodů, dále je zpracovávat a využít je jako zdroj dat při vykreslení výškového profilu. K tomu nám poslouží Greasemonkey (GM) - rozšíření pro prohlížeč Firefox. GM umožní pomocí jazyka JavaScript pracovat s DOM objekty na webu [www.mapy.cz](http://www.mapy.cz). Především je nutné implementovat získávání souřadnic bodů zdaných pomocí nástrojů webového rozhraní. Známe-li souřadnice všech daných bodů, přistoupíme k vykreslení výškového profilu trasy. Ten můžeme získat pomocí [www.heywhatsthat.com/profiler.html](http://www.heywhatsthat.com/profiler.html) nebo implementovat vlastní vykreslovací algoritmus. Vytvoření aplikace schopné vykreslit výškový graf předpokládá znalost nejen geografických souřadnic bodů, jimiž trasa prochází, ale především jejich nadmořské výšky. Hodnoty nadmořských výšek nám poskytnou metody aplikačního rozhraní z webu [www.vyskopis.cz](http://www.vyskopis.cz). Součástí bakalářské práce bude i vytvoření webu umožňujícího práci s JavaScriptovým API (registrace se váže k doméně, na které je API používáno).

Oba uvedené způsoby získání výškového profilu (použití existující služby na internetu a vytvoření vlastního vykreslovacího algoritmu) jsou v práci popsány a implementovány.

## **2. Geoinformatika a geografické informační systémy**

### **2.1 Geoinformatika a co o ní víme**

GIT, neboli Geoinformatika je obsáhlý obor, ve kterém se snoubí několik oblastí informačních technologií. První pojem co nás napadne v souvislosti s GIT jsou Geografické informační systémy a mapové servery. To ale jistě není vše, čím se může a musí geoinformatika zabývat. Úzce souvisí s databázovými systémy, bez nichž by nemohla existovat, s počítačovými sítěmi, počítačovou grafikou, počítačovou simulací. Široké využití nalézáme ve stavebnictví pro vyměřování tras silnic a železnic, určování a mapování podloží a nerostného bohatství, v geodézii, v biologii (pro zkoumání lokalit výskytu rostlin a živočichů) a v každodenním životě lidí vůbec.

Princip vzájemné polohy objektů se uplatňuje také v počítačové kartografii (CAM) a v systémech pro kreslení výkresů a modelování jako jsou CAD systémy, vizualizační systémy a další.

#### **2.1.1 Geografický informační systém**

GIS je elektronický systém pro zpracování geografických informací [1].

Geografická informace je ucelený údaj o hmotném nebo nehmotném objektu, kde její nutnou součástí je údaj o geografické poloze objektu [1].

GIS tedy pracuje s geobjekty - objekty o nichž zpracovává a uchovává geografickou informaci.

## **2.2 GIS nástroje**

Existuje množství firem zabývajících se tvorbou GIS softwaru. Například produkt ArcGIS od firmy ESRI je jedním z nejpokročilejších nástrojů tohoto druhu na světě.

Na platformách Linux můžeme použít volně dostupný software GRASS (Geographic Resources Analysis Support System). GRASS je původně projekt armády USA, který byl kompletně zveřejněn v 80. letech 20. století. Dnes díky vývojářům, kteří tento systém stále zdokonalují, můžeme využívat jeho obsáhlou knihovnu funkcí a nástrojů.

## **2.3 Pojem mapový server**

Mapové servery pracují na architektuře klient-server. Pracují na základě vstupních parametrů. Obsahují programy pro práci s geodaty a vlastní geodata ukládaná do speciálních databází.

Jsou provázány a komunikují s webovými servery. Potřebné parametry obrdží webový server (například z webových formulářů) a předá je mapovému serveru. Ten na základě těchto parametrů

provede příslušné operace nad geodaty a výsledek předá zpět webovému serveru, odkud jsou poslány zadavateli - uživateli webové aplikace.

## **2.4 Fungování mapserveru**

### **2.4.1 Protokol CGI**

Jednou z možností je komunikace pomocí Common Gateway Interface - CGI [6]. Program na mapovém serveru vrací na základě vstupních parametrů vygenerovanou mapu nebo HTML stránku, která danou mapu obsahuje. Webový server potom výsledek zobrazí.

### **2.4.2 Rozhraní programovacích jazyků**

Ke komunikaci s mapovým serverem můžeme použít řadu programovacích jazyků. Často jazyk Java, C/C++. Nebo skriptovací jazyky (PHP, MapScript, Perl, Python a také JavaScript).

## **2.5 Souřadnicové systémy**

Vhodný souřadnicový systém je nutnou podmínkou k určování topologie (vzájemné vztahy mezi objekty) a geometrie (velikosti a tvaru) objektů. Musí splňovat kriteria jednoznačnosti (objekty se stejnou polohou musí být identické) a definovat jednotky se kterými bude pracovat [1]. Bez souřadnicového systému bychom nemohli určovat nejdůležitější vlastnost objektů - jejich geografickou polohu a s tím související vzdálenosti.

Známe několik souřadných systémů mapujících celou Zemi i různé oblasti. Uplatňují se globální i regionální systémy. Zde se budeme krátce věnovat třem u nás nejpoužívanějším.

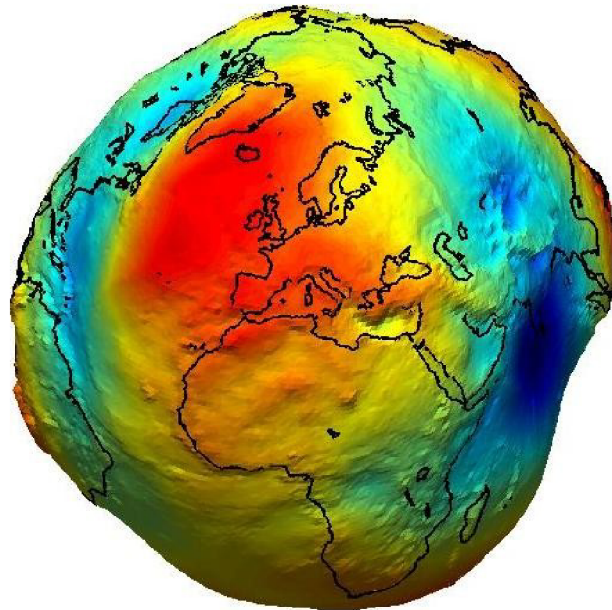
### **2.5.1 Související pojmy**

#### **2.5.1.1 Geoid**

Geoid je pomyslná plocha - idealizovaný tvar Země přibližně splývající se střední hladinou moří (střední hladina je průměrná výška hladiny mezi přílivem a odlivem). Je definován jako ekvipotenciální plocha vůči gravitaci. To znamená, že v každém bodě této plochy má tíhové zrychlení stejnou hodnotu a tíhový vektor je na plochu kolmý.

Aproximuje skutečný topografický povrch planety - zjednodušuje výškový popis zemského tělesa [5].

Jeho matematický popis je však stále příliš složitý, proto se zavádí další zjednodušení zemského povrchu - rotační (referenční) elipsoid.

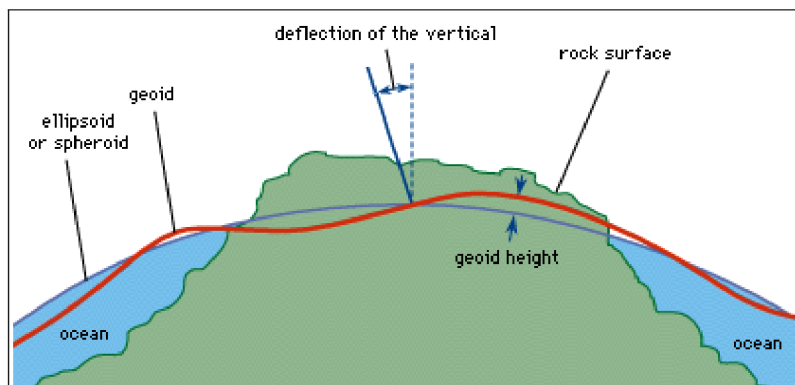


Obrázek 2.1: Geoid – počítačový model [6].

### 2.5.1.2 Referenční elipsoid

Rotační elipsoid je matematické těleso vytvořené rotováním elipsy kolem jedné z jejích os.

**Rotační elipsoid** přimykající se ke geoidu nazýváme **referenční elipsoid**. Plocha referenčního elipsoidu se označuje jako referenční plocha. Vytvoření globálního elipsoidu, který by popisoval povrch celé planety nebylo triviální. Dodnes se používá elipsoid WGS84, jehož podoba byla dokončena v roce 1984. Referenční elipsoid se od geoidu může lišit až o 100 metrů.



Obrázek 2.2: Skutečný reliéf, geoid a referenční elipsoid [6].

## 2.5.2 WGS84

World Geodetic System 1984 definovaný jako referenční elipsoid je velmi často používaný a dobře známý globální souřadnicový systém. Jeho výhodou je, že dokáže popsat komplexně povrch celé

Země, proto je široce používán v naprosté většině geografických systémů. Pro malá území, kde se vyžaduje vysoká přesnost, se častěji uplatňují lokální souřadné systémy .

Dalším důvodem rozšíření je používání systému zeměpisná šířka - zeměpisná délka.

### 2.5.2.1 Zeměpisná šířka a délka

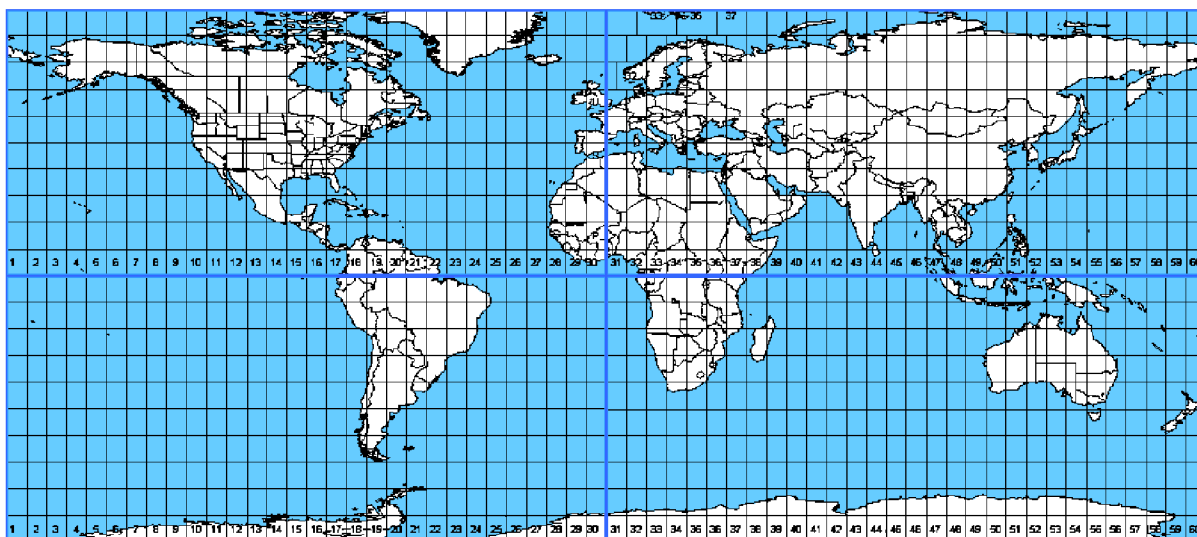
Země je rozdělena pomyslnými kružnicemi - poledníky a rovnoběžkami - tvořícími tzv. zeměpisnou síť. Rovnoběžky leží na rovinách kolmých k zemské ose rotace a určují zeměpisnou šířku. Nejdelší rovnoběžku nazýváme rovník. Zeměpisná šířka je úhel mezi spojnicí daného bodu se středem Planety a rovinou určenou rovníkem. Rozlišujeme jižní a severní šířku, pro jižní a severní polokouli. Polokoule jsou odděleny rovníkem ( $0^\circ$ ). Zeměpisná šířka (severní i jižní) může nabývat hodnot od  $0^\circ$  do  $90^\circ$  (dohromady  $180^\circ$ ).

Zeměpisnou délku určujeme pomocí poledníků - pomyslné kružnice kolmé na rovník a procházející oběma póly. Zeměpisná délka je tedy úhel, který svírá rovina základního ( $0$ ) poledníku s rovinou poledníku, který daným bodem prochází. Rozsah hodnot je od  $0^\circ$  do  $180^\circ$ . Základní poledník je určený dohodou. Z historických důvodů prochází městem Greenwich v Anglii.

## 2.5.3 UTM

Universal Transverse Mercator Geographic Coordinate System je založený na Gaus/Krügerově zobrazení. Zemský povrch se rozdělí na poledníkové pásy a promítne na válec. Válec se potom „rozvine“ do roviny a vznikne výsledný souřadný systém. Celá síť se skládá z 60 zón [7].

Střed souřadnic je pro každou zónu jiný a tvoří jej průsečík středového poledníku zóny s rovníkem. Od tohoto středu se měří vzdálenosti v metrech na ose x rostoucí od středového poledníku směrem na východ a na ose y rostoucí od rovníku směrem na sever [5]. Tento pravoúhlý souřadnicový systém značně zjednodušuje v rámci zóny výpočty vzájemné polohy jednotlivých bodů. Pokud leží dva body ve stejné zóně, lze jejich vzdálenost vypočítat pomocí Pythagorovy věty.



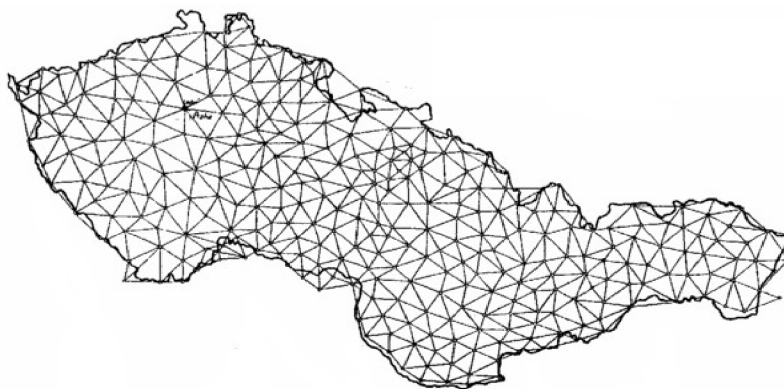
Obrázek 2.3: Souřadný systém UTM [6].



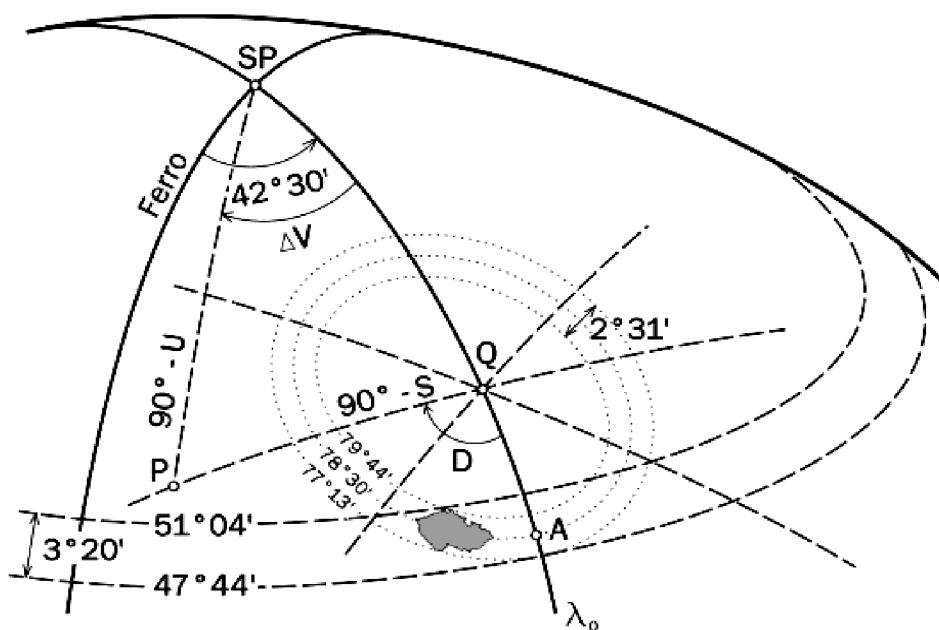
## 2.5.4 S-JTSK

Kromě globálních souřadných systémů umožňujících mapování rozsáhlých území známe několik lokálních systémů vhodných pro mapování určitých oblastí.

S-JTSK je zkratka pro „Jednotnou trigonometrickou síť katastrální“ někdy označovanou jako „Křovákovo zobrazení“. Je platné na území ČR a SR, pro tehdejší Československou republiku ho navrhl a vytvořil Ing. Josef Křovák a jeho tým. Základem je Besselův elipsoid (vhodný pro střední Evropu), ten je konformně zobrazen na Gaussovu kouli s poloměrem 6380,704 km. Poloměr koule se zmenší o jednu tisícinu, potom je koule zobrazena do roviny. Zobrazení je konformní a ekvidistantní, chyba způsobená transformací je snesitelná [1]. Vzniklý souřadnicový systém je pravouhlý, měří se v něm v metrech.



Obrázek 2.4: Jednotná trigonometrická síť katastrální 1. řádu (1936) [6].



Obrázek 2.5: Křovákovo zobrazení [6].

## 3. Analýza a návrh řešení

### 3.1 Dostupná výšková data

Abychom mohli vytvořit výškový profil nějakého území nebo trasy, potřebujeme nezbytně získávat a zpracovávat informace o nadmořských výškách.

Nejpřesnější tzv. „level 5“ (GTOPO) s přesností cca 1m má k dispozici americká armáda a orgány jejích spojeneckých armád. Výškové přesnosti jednoho metru dosahují u svých mapových produktů také komerční firmy (např. ESRI, u nás to je GEODIS, T-MAPY a další).

U nekomerčních dat začíná přesnost od 3 metrů (na území USA). U nás je maximální přesnost těchto volně využitelných dat 60 metrů.

Uvedeme zde tři nejobvyklejší způsoby získání volně dostupných výškových dat.

#### 3.1.1 NIMA

National Imagery and Mapping Agency - NIMA získává a spravuje výšková data primárně určená pro armádu. Podporuje ale i civilní klienty. Přesnost dat je rozdělena do kategorií (Level 0 – Level 5). Volně dostupná je na úrovni 0. S přesností cca 1km můžeme data stáhnout na [13]. Pomocí webového rozhraní, kde se objeví mapa Země, stáhneme požadovanou oblast.

#### 3.1.2 GTOPO30

Tato data jsou srovnatelná s NIMA, postup získávání a jejich přesnost (cca 1km) jsou si podobné. Jsou dostupné z internetu například na [12]. Pro Evropu mají velikost kolem 8MB.

#### 3.1.3 SRTM

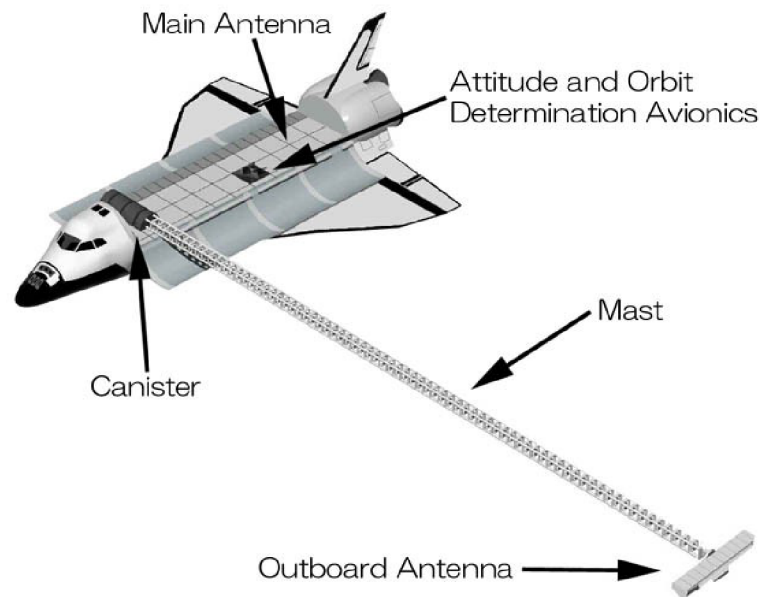
Shuttle Radar Topography Mission je výzkumný program NASA. Během mise STS-99 raketoplánu Endover trávící 11 dní měřící zařízení snímalo pomocí dvou antén povrch planety (metodou radarové interferometrie) [5]. Jedna byla umístěna na raketoplánu, druhá na 60 metrovém výsuvném rameni. Podařilo se nasnímat 80 % povrchu souše. Byl tak vytvořen soubor dat (několik terabytů) mapující zemský povrch ve vysokém rozlišení (až 1 úhlová vteřina na území USA, pro ostatní oblasti 3 vteřiny). Výškové hodnoty by se od skutečného reliéfu neměly lišit víc jak o  $\pm 16$  metrů [7].

Měření pomocí radarové interferometrie má i své nevýhody. Radiové vlny se od povrchu neodrážejí kolmo, ale pod určitým úhlem. Důsledkem toho je problém s měřením výškové polohy velkých vodních ploch, protože dopadající signál se od nich odráží jako od zrcadla a nevrátí se zpět k raketoplánu. Vznikají tak „hluchá místa“ bez dat (NODATA). Další nepřesnosti vznikají ve velmi složitých terénech. Nazýváme je „hroty“ a „studny“.

V novějších verzích SRTM2 a SRTM3 je snaha taková místa odstranit dodatečným měřením nebo speciálním softwarem. Byla vytvořena také vektorová vrstva mapující pobřežní linii, tzv. SWBD (SRTM Water Body Data) [7].

Všechna SRTM data jsou ke stažení z FTP NASA. Aktuální data jsou momentálně dostupná ve verzi SRTM3 [14]. V adresáři Eurasia můžeme najít data pro ČR. Každý jednotlivý soubor mapuje území 1 x 1 stupeň v zeměpisných souřadnicích. Maximální přesnost je 15 metrů horizontálně

a 12 metrů vertikálně. Data jsou v rastrové podobě s rozlišením 3 úhlové vteřiny, to znamená přesnost na rovníku cca 30 metrů a v našich zeměpisných šířkách mluvíme o rastru 60 x 90 metrů.



Obrázek 3.1: Snímací zařízení raketoplánu pro získávání SRTM dat [6].

### 3.1.4 Digitální výškový model Země

Na základě zpracování výškových dat lze následně prostřednictvím k tomu určenému softwaru (GRASS, ArcGIS, OZIExplorer a další) vytvořit DEM (Digital elevation model). Digitální model povrchu daného území.

## 3.2 Možnosti získávání a zpracování topografických dat

K implementaci aplikace umožňující zobrazení výškového profilu však samotná data nestačí. Jsou to binární soubory o nemalé velikosti, které je nutné zpracovávat pomocí speciálního softwaru. V zásadě máme dvě možnosti jak z těchto naměřených dat získávat informaci o nadmořské výšce daných bodů. Můžeme si vytvořit vlastní mapový server, který bude s daty pracovat a vytvářet výstupy. Nebo využít služeb již existujícího takového serveru, poskytujícího volně nebo po registraci potřebná data.

### 3.2.1 Vytvoření vlastního mapového serveru

Mapový server je poměrně složitý systém, k jehož fungování je zapotřebí nainstalování složitého softwaru. Na druhou stranu existuje několik kvalitních produktů s otevřenou licencí, jež můžeme k tomuto účelu použít.

Velice mocným nástrojem je systém GRASS, ten umí se SRTM daty pracovat, vytvářet mapové výstupy, 3D modely a má mnoho dalších možností. K použití GRASSu pro účely této bakalářské práce by bylo nutné zpřístupnit funkce GRASSu přes webového rozhraní, abychom mohli pomocí takto vytvořeného mapového serveru získávat informace o nadmořských výškách nebo přímo vykreslovat výškové profily. Tento způsob řešení by vyžadoval nainstalovat GRASS, stáhnout SRTM data, pro přístup a ovládání z internetu by byl nutností webový server.

Vlastní ovládání GRASSU z internetu lze realizovat za použití modulu Mapscrip, s jehož pomocí lze ve skriptovacích jazycích (PHP, Python, Perl a dalších) přistupovat k funkcím mapserveru.

Další možnost nabízí jeden z nejpoužívanějších mapserverů na světě – UNM Mapserver. Tento nástroj vyvinula Univerzita v Minesotě. Sice není uvolněn pod GNU licenci, je ale umožněno jeho volné využití [15]. Původně vznikl jako školní projekt, ten byl dále rozvíjen i ve spolupráci s renomovanými firmami a organizacemi (např. NASA).

Jedná se o CGI aplikaci ovládanou pomocí konfiguračních souborů. Potřebuje tedy komunikovat s webovým serverem, z něho bude získávat potřebné parametry pro provádění operací nad geodaty. Na základě obdržených parametrů z webového serveru provede příslušné mapové operace a vrátí požadovaný výsledek. Tím může být rastrová mapa, HTML stránka obsahující mapu.

Z uvedeného vyplývá, že vytvoření mapového serveru je uspokojivé řešení zobrazování výškového profilu na webu. Dokáže nám poskytnout široké možnosti pro práci a zobrazování výškových dat.

Podle zadání však budeme při získávání výškových informací o zadané trase využívat právě souřadnice zadané pomocí nástrojů veřejného a již existujícího mapového serveru s webovým rozhraním, jedná se tedy o rozšíření možností takového serveru pro uživatele.

Vytvoření vlastního serveru vyžaduje nainstalovat mapový software, webový a databázový server přístupný z internetu, potřebné místo pro uložení dat. Pro další používání aplikace by musel být server stále dostupný a udržovaný správcem.

Existuje ale jednodušší řešení – použít k získávání výškových dat a vykreslování výškového profilu již existující službu volně dostupnou na internetu.

## **3.2.2 Využití veřejného serveru**

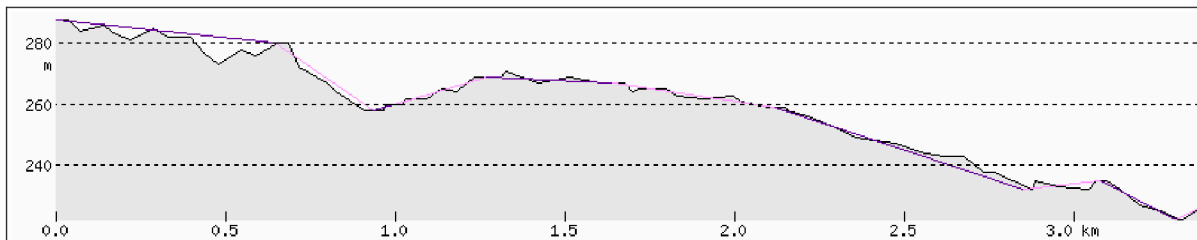
### **3.2.2.1 Služba [www.heywhatsthat.com/profiler.html](http://www.heywhatsthat.com/profiler.html)**

Na adrese <http://www.heywhatsthat.com/profiler.html> [2] nalezneme webové rozhraní, jehož prostřednictvím může uživatel naklikat jakoukoliv trasu na mapě a následně zobrazit její výškový profil.

O získávání výškových dat i vykreslování výškového grafu se stará server, uživatel zadává na straně klienta jednoduchými pohyby myši jednotlivé body (zapisované do URL stránky). Po kliknutí na příslušný odkaz se odešlou serveru metodou GET, klient dostane jako odpověď HTML stránku obsahující hotový obrázek profilu. Ten lze získat i samostatně (bez HTML stránky).

Výhodou tohoto způsobu komunikace se serverem je přenositelnost odkazů. Vytvořenou adresu můžeme poslat jako text kolegovi. Po zadání adresy do internetového prohlížeče obdrží náš kolega stejný obrázek, jaký vidíme na svém monitoru.

Tato služba má ale i jednu nevýhodu. Pokud chceme vytvořit vlastní algoritmus pro vykreslení výškového profilu, vygenerovaný obrázek nám jako zdroj dat poslouží jen těžko. Musíme vyhledat jinou aplikaci, umožňující zadat na vstup geografické souřadnice a vracející hodnotu nadmořské výšky.



Obrázek 3.2: Výškový profil trasy Koleční – Božetěchova vykreslený pomocí [2].

### 3.2.2.2 API [www.topocoding.com](http://www.topocoding.com)

Je nadstavba od firmy Google existujících mapových API (application programming interface). Poskytuje PHP API a JavaScriptové API pro získávání nadmořských výšek.

Původně bylo zamýšleno pracovat s PHP API [3]. Tedy implementovat PHP skript spuštěný na webovém serveru, který by používal toto API a fungoval jako prostředník mezi GreaseMonkey skriptem a zdrojovým serverem s výškovými daty. Tento PHP skript by byl volán pomocí JavaScriptu na straně klienta s určitými parametry. Potom by pomocí PHP API zjistil výškové údaje zadaných bodů, ty by dále zpracovával a vytrvářel by výškový graf.

Výše uvedený způsob (pomocí podpůrného PHP skriptu) nebyl implementován, protože PHP API se ukázalo jako (dočasně) nefunkční.

Další postup práce se zaměřil na použití API v JavaScriptu. K tomuto API existuje dokonce dokumentace v češtině. Nalezneme ji na adrese [www.vyskopis.cz](http://www.vyskopis.cz) (viz. níže). Na českých stránkách je dokumentace jen k JavaScriptovému API (PHP API zde již není zmíněno).

### 3.2.2.3 API [www.vyskopis.cz](http://www.vyskopis.cz)

Vyskopis.cz [4] je česká verze webu [www.topocoding.com](http://www.topocoding.com). Poskytuje aplikační rozhraní umožňující zjišťování nadmořských výšek v JavaScriptu. Používá souřadný systém WGS84. Výšková data jsou kombinací GTOPO a SRTM3.

Je nutná registrace domény na které chceme funkce API používat. Nejdůležitější poskytované funkce jsou:

`topoGetAltitude(lat, lon, action, context, timeout)` – Funkce umožňující získání nadmořské výšky bodu se souřadnicemi „lat“ - zeměpisná šířka, „lon“ - zeměpisná délka. Parametr „action“ určuje událost, které se předá zjištěná hodnota nadmořské výšky daného bodu. Volitelný parametr „context“ je předán jako druhý parametr události „action“. „Timeout“ určuje maximální dobu čekání na odpověď ze serveru.

Příklad použití funkce `topoGetAltitude(lat, lon, action, context, timeout)`. Následující kód vypíše nadmořskou výšku zadaného bodu:

```
topoGetAltitude( lat, lon , function(altitude){  
    alert ("Nadmořská výška je: "+altitude+"metrů");  
});
```

Pro větší počet použití je tato funkce bohužel nevhodná. Zjištění výšky jednoho bodu je třeba jeden dotaz na server, což pro několik stovek nebo tisíců bodů trvá příliš dlouhou dobu.

`topoGetAltitudes`(*quadruples*, *timeout*): Provede hromadný dotaz na server. "Quadruples" je pole polí [lat lon action context] - viz popis parametrů `topoGetAltitude()`. Tato funkce umí získat v jednom dotazu na server až 280 hodnot nadmořských výšek, což představuje výrazné urychlení oproti předchozí funkci.

`topoComputeDistance`(*lat1*, *lon1*, *lat2*, *lon2*): Vypočítá v JavaScriptu vzdálenost v kilometrech mezi místem o souřadnicích (*lat1*, *lon1*) a místem o souřadnicích (*lat2*, *lon2*).

`topoComputeIntermediate`(*lat1*, *lon1*, *lat2*, *lon2*, *fraction*) vrátí pole obsahující souřadnice místa, které se nachází na cestě mezi místem o GPS souřadnicích (*lat1*, *lon1*) a místem o souřadnicích (*lat2*, *lon2*).

`topoDrawGraph`(*divElement*, *coordinates*, *width*, *height*, *english*): Vykreslí výškový profil cesty procházející místy, jejichž souřadnice (*lat*, *lon*) jsou prvky pole "coordinates". Tato funkce nebyla v projektu využita.

#### 3.2.2.4 Server [latlontoelevation.com](http://latlontoelevation.com)

Ze serveru je možné získat výšková data SRTM kombinovaná s NIMA. Jako v předchozím případě v celosvětovém pokrytí.

Přístup k těmto datům je ale nevhodný a vyžaduje interakci uživatele. Pomocí webového rozhraní lze nahrát na server konfigurační soubor v předepsaném formátu. Server provede výpočet a po jeho dokončení umožní stáhnutí výsledku. Opět je za potřebí interakce uživatele. Služba tedy není vhodná pro automatizované získávání dat.

## 3.3 Server [www.mapy.cz](http://www.mapy.cz)

Podle zadání je cílem této bakalářské práce vytvořit aplikaci umožňující uživateli používajícímu veřejný GIS zobrazit výškový profil zadané trasy. Trasa má být zadána pomocí nástrojů tohoto GISu a dále zpracována tak, aby bylo možné profil vykreslit.

GISem stěžejním pro celou práci byla vybrána služba [www.mapy.cz](http://www.mapy.cz), jeden z nejpoužívanějších mapových veřejných serverů v ČR. Důvodem, proč byl GM skript implementován právě pro [www.mapy.cz](http://www.mapy.cz), jsou také podrobné mapové podklady. Čeští uživatelé používající tuto službu provozovanou a vytvořenou firmou Seznam budou mít tedy možnost vykreslit výškový profil za pomoci webového rozhraní, které je jim dobře známé. Máme zde k dispozici kvalitní mapové podklady (vlastníkem je firma GEODIS) a širokou nabídku poskytovaných funkcí a nástrojů, jež můžeme použít.

Generování map je zajišťováno ze strany severu. Ovládání GISu je přístupné webovým rozhraním, používajícím takzvané API [Mapy.cz](http://www.mapy.cz). Jedná se o aplikační rozhraní v jazyce JavaScript implementující operace nad mapami. Většina nástrojů pro práci s mapou na [www.mapy.cz](http://www.mapy.cz) jakými jsou např. „měření“, „GPS“, „zoom“, přepínání mapových vrstev a další používají funkce tohoto API (viz. bližší popis níže).

### **3.3.1 API Mapy.cz**

API Mapy.cz [8] je obchodní označení pro aplikační rozhraní implementující funkce pro vytvoření a ovládání mapy a práci s ní na webové stránce. Je to výpočetní knihovna GIS serveru [www.mapy.cz](http://www.mapy.cz). Použití API není ale omezené jen pro tuto webovou adresu.

Umožňuje vytvořit mapu na libovolné webové stránce. K tomu budeme potřebovat existující doménu a registrační klíč. Klíč obdržíme po zaregistrování domény, na které chceme mapu zobrazovat.

## 4. Technologie použité k implementaci

### 4.1 JavaScript

Stěžejním programovacím jazykem celé práce je skriptovací jazyk JavaScript. Je to interpretovaný programovací jazyk se základními objektově orientovanými schopnostmi [9]. Používá se při dynamickém vytváření uživatelského rozhraní WWW stránek. Vytvořila ho firma Netscape, kde byl původně vyvíjen pod názvy Mocha, později LiveScript. Některé další firmy vyvinuly vlastní verze (např. Microsoft a jeho JScript podporovaný platformou .NET). V roce 1997 byl JavaScript standardizován organizací ECMA (European Computer Manufacturers Association), v roce 1998 potom ISO (International Organization for Standardization). Následovaly další standardizace a implementace jazyka.

Syntaxe JavaScriptu vychází z jazyků typu C/C++ a Java. Jeho podobnost s javou ale na úrovni syntaxe končí (jinak s ní nemá mnoho společného).

Existuje i serverová verze JavaScriptu, ale většinou je užíván jako interpretovaný jazyk na straně klienta. To má mnoho výhod. Umožňuje vytvářet kód webové stránky reagující na akce uživatele bez nutnosti dotazů na server. Lze přidávat do stránky text, HTML kód, tlačítka, formuláře, obrázky, vytvářet animace, kreslit obrázky (např. pomocí DHTML tagu `canvas`), vypisovat hlášky, reagovat na pohyb kurzoru po obrazovce, zpracovávat další události. Na straně klienta je rozvňž možné provádět různé výpočty, zpracovávat zadané hodnoty, zobrazovat výsledky uživateli, používat timeout, dotazovat se serveru, získávat data z databáze, otevírat a zavírat okna, pracovat s DOM strukturou stránky a další.

Provádění kódu na straně klienta má i nevýhody. Ne všechny prohlížeče striktně dodržují standardizaci a interpretace na jiném prohlížeči může být rozdílná. Někdy se liší DOM struktura stránky, jindy zase funkce nebo metody v různých prohlížečích nebudou fungovat, jiné jsou naopak podporovány právě jedním prohlížečem.

### 4.2 DOM struktura webových stránek

V minulosti každý prohlížeč měl své specifické aplikační rozhraní pro práci s XML a HTML dokumenty v JavaScriptu. Vznikla tak potřeba standardizace. World Wide Web Consortium (W3C) vytvořilo standardizaci Document Object Model (W3C DOM).

Aplikační rozhraní DOM umožňuje přístup k dokumentu jako ke stromu objektů. Jednotlivé XML, XHTML objekty jsou seřazeny do hierarchické struktury. Prostřednictvím JavaScriptu potom lze přistupovat ke každému objektu a pracovat s ním (např. měnit hodnoty atributů, přidávat text, měnit css styly, vytvářet nové objekty, měnit odkazy atd.). Každý objekt může mít podobjekty, vlastnosti, metody.

Podobjekt je podřazený objekt svému předchůdci podle stromu objektů. Například `document.location` - objekt „document“ je nadřazený objektu „location“.

Vlastnost (atribut) objektu má nějakou hodnotu, která se obvykle vztahuje k danému objektu. Například `document.title` může mít textovou hodnotu „Název stránky“.



Metoda je - zjednodušeně řečeno - funkce přidělená objektu. Metody spustíme jejich zavoláním (mohou mít parametry a mohou vrátit návratové hodnoty). Například `document.location.replace('http://www.adresa.cz')` přesměruje stránku na novou adresu.

## 4.3 GreaseMonkey

GreaseMonkey je rozšíření pro známý internetový prohlížeč Mozilla Firefox. Tento plugin umožňuje uživateli internetu přidávat do již existujících webových stránek a aplikací vlastní funkcionalitu prostřednictvím JavaScriptu. Přidaný kód se provádí v prohlížeči až po úplném načtení webové stránky ze serveru. GreaseMonkey pracuje s klientkým JavaScriptem, který se jen mírně liší od „klasického“ JavaScriptu.

Pomocí GM (GreaseMonkey) a klientského JavaScriptu lze tedy přímo manipulovat s DOM strukturou stránky a téměř libovolně přidat jakýkoliv obsah. Pokud z nějakého důvodu nejsme spokojeni s webovou aplikací, chybí nám funkce, které známe z jiných webů a chceme je mít i na webové stránce, kde nejsou. Nebo se nám nelíbí vzhled, použité styly a barvy. Můžeme přidat libovolné ovládací prvky a měnit barvy a vzhled, jak se nám bude líbit. Můžeme si například na stránkách svého oblíbeného internetového obchodu zobrazovat za jakou cenu má stejné zboží konkurence. Často se GM využívá ke zdokonalení navigace a přidání odkazů do webových stránek pro lepší provázanost.

Plugin je pouze pro Firefox. V poslední době již začínají i jiné prohlížeče podporovat podobná rozšíření.

Nejnovější verze GreaseMonkey je momentálně k dispozici na webové adrese <https://addons.mozilla.org/cs/firefox/addon/748>. K nainstalování GM potřebujeme mít již na svém počítači prohlížeč Firefox a nějaký textový editor. Při instalaci se nás průvodce zptá, jaký textový editor chceme pro úpravu a tvorbu GM skriptů používat. Do menu Firefoxu „nástroje“ je přidána položka „Greasemonkey“. Ta dále odkazuje na možnost vypnout nebo zapnout GM, na vytvoření nového GM skriptu a na nástroj pro správu uživatelských skriptů.

Uživatelské skripty můžeme nainstalovat (již existující skript stažený z webu nebo jinak získaný) nebo napsat vlastní. To nám umožní průvodce „Nový uživatelský skript“. Otevře se první okno. Do položky „name“ vyplníme název skriptu, do položky „description“ napíšeme stručný popis. Do „Includes“ a „Excludes“ vyplníme masky adres stránek, na kterých chceme náš skript spouštět.

Po potvrzení se vytvoří šablona nového uživatelského skriptu s příponou `.user.js` (například `muj_skript.user.js`) a otevře se v editoru, který jsme zadali při instalaci GM jako výchozí. V meta informacích jsou uvedeny údaje o skriptu jako název, popis, masky adres stránek, na kterých se má skript spouštět.

Příklad meta informací:

```
// ==UserScript==
// @name           muj_skript
// @namespace      none
// @description    Ukázkový skript pro GM
// @include        http://www.mapy.cz/*
// ==/UserScript==
```

Nyní již můžeme psát vlastní kód skriptu.

## 4.4 HTML a DHTML

HyperText Markup Language je jedním z jazyků W3C. Umožňuje vytváření a publikování hypertextových dokumentů na internetu. Každá webová stránka je primárně napsána v HTML. Patří do rodiny jazyků SGML (Standard Generalized Markup Language), což je univerzální značkovací jazyk pro definici značkovacích jazyků.

Pomocí DHTML (dynamické HTML) vytváříme „dynamické“ stránky, to znamená stránky, které po načtení mohou měnit svůj vzhled nebo obsah (bez nutnosti dotazu na server). Takové změny jsou prováděny na straně klienta. V DHTML se kombinují technologie jako HTML, CSS (kaskádové styly popisující vzhled dokumentů), JavaScript a DOM. Opakem je statické HTML, kdy se vzhled a obsah stránek po načtení již nemění.

Každá verze HTML je charakterizována množinou značek a jejich atributů definovaných pro danou verzi. Každá značka neboli tag začíná úhlovou závorkou „<“ a končí „>“. Značky mohou být párové a nepárové (párové jsou ukončeny koncovou značkou). Tagy uzavírají části dokumentu a určují tím jejich sémantiku (význam) v dokumentu. Část textu dokumentu, která je uzavřena pomocí tagů se nazývá **element**.

Vytvořený skript pro zobrazení výškového profilu pracuje za pomoci GreaseMonkey a JavaScriptu s HTML elementy webu [www.mapy.cz](http://www.mapy.cz).

HTML je použit také pro vytvoření podpůrné webové aplikace umožňující používání registrovaného JavaScriptového API.

Samotné vykreslování výškového profilu probíhá prostřednictvím JavaScriptových funkcí umožňujících kreslení pomocí DHTML tagu `canvas`.

Canvas je HTML element umožňující volné kreslení do webové stránky. Můžeme vytvářet, kreslit a vyplňovat grafická primitiva (úsečky, kružnice, obdélníky, plygony), vkládat externí obrázky, vytvářet animace. K elementu canvas přistupujeme a kreslíme v JavaScriptu.

## 4.5 PHP

Pro vytvoření podpůrné webové aplikace pro uživatelský skript použijeme jazyka DHTML v kombinaci s PHP. Hypertext Preprocessor nebo také Personal Home Page je oblíbený a hojně používaný skriptovací jazyk pro tvorbu dynamických webových stránek. Syntaxe je podobná jazykům Pascal, C/C++, Perl, nebo Java.

PHP skripty mohou být psány jako součást HTML kódu nebo samostatně (způsob jakým skript napíšeme se odvíjí od jeho účelu). Skript se spouští na serveru, klientovi je odeslán výsledek skriptu (nejčastěji webová stránka). PHP je multiplatformální jazyk, to znamená že kód lze použít na různých operačních systémech. Jsou podporovány různé knihovny. Například pro generování obrázků, práci se soubory, práci s internetovými protokoly (HTTP, FTP, LDAP, POP3, IMAP, SMTP, SNMP a další.). Velmi využívané je PHP pro přístup k databázovým systémům. Podporuje například: MySQL, PostgreSQL, Oracle a další DB technologie.

V této bakalářské práci je PHP využit jen okrajově - k vytvoření webových podpůrných stránek projektu.

## 4.6 Užívání API Mapy.cz

Toto API je soubor metod, které používá GIS [www.mapy.cz](http://www.mapy.cz). Umožňuje ale i vkládání mapy do libovolné webové stránky, jejíž adresu zaregistrujeme. API poskytuje mnoho metod pro vkládání ovládacích prvků, práce s mapou, převody souřadnic, vkládání značek do mapy, centrování mapy na libovolné souřadnice, přepínání mapových vrstev a mnoho dalších.

V této kapitole je popsáno použití metod a funkcí API, které jsou pro projekt důležité. Ostatní metody, jež nebyly použity v projektu, jsou zmíněny jen okrajově nebo vůbec. Úplnou dokumentaci najdeme na stránkách API [8].

### 4.6.1 Licenční podmínky

API lze volně používat pro nekomerční účely. Má omezené podklady oproti službě [www.mapy.cz](http://www.mapy.cz). Zobrazení mapy je omezeno na 1000 za den.

Použití vyžaduje vyplnění [www](http://www.mapy.cz) adresy, na které budeme mapy zobrazovat. Je vyžadována aktivace (aktivační odkaz je zaslán na zadaný e-mail). Kompletní licenční podmínky najdeme na adrese: <http://api.mapy.cz/keygen>.

### 4.6.2 Registrace

Abychom mohli používat API, musíme naši webovou stránku zaregistrovat. Registraci provedeme na stránkách [8]. Máme-li registrační klíč, vložíme do stránky následující kód:

```
<script type="text/javascript"
src="http://api.mapy.cz/main?key=REG_KLIC&ver=3&encoding=utf-8">
</script>
```

Od této chvíle již můžeme API plně využívat.

### 4.6.3 Vložení mapy do webové stránky

Nejprve vytvoříme HTML element, ve kterém budeme chtít mapu zobrazit.

```
<div id="mojeMapa"></div>
```

Do tohoto bloku potom vložíme mapu. Vytvoříme objekt „**mapa**“ pomocí API konstrukturu `SZN.MapEngine(element)`:

```
<script type="text/javascript">
if(SZN.isSupported){
var mapa = new SZN.MapEngine(document.getElementById("mojeMapa"));
}
</script>
```

Vytvořený objekt `mapa` je instancí mapy va API [mapy.cz](http://www.mapy.cz). Metody API voláme: `mapa.metoda()`.

## 4.6.4 Souřadné systémy v API Mapy.cz

GIS server `www.mapy.cz` používá pro manipulaci s mapou svůj vlastní souřadný systém. Tento pravouhlý systém „PP“ je podobný souřadnému systému UTM. Jednotky PP je možné jednoduše přepočítat na metry. K tomu slouží v API metoda `mapa.ppToMeter(ppDiff)`. Argument „ppDiff“ je vzdálenost v metrech. Metoda vrací číslo – vzdálenost v PP. Pokud chceme převést metry na jednotky PP, použijeme metodu `mapa.meterToPP(meterDiff)`. Vracející vzdálenost v metrech. K ověření platnosti PP souřadnic bodu můžeme použít metodu `mapa.checkPP(x, y)`. Pokud je daný bod v zobrazovaném PP světě, metoda vrací **true**, jinak vrací **false**.

V API jsou k dispozici metody pro převody mezi systémy PP, UTM a WGS84.

### 4.6.4.1 Převody mezi PP a UTM

Oba tyto souřadné systémy jsou pravouhlé. Převod souřadnic mezi nimi je pouhá transformace z jednoho systému do druhého.

Pro převod z PP do UTM slouží metoda `mapa.ppToUTM(x, y)`. Argumenty „x“ a „y“ jsou souřadnice daného bodu v PP. Metoda vrací objekt s atributy „east“ a „north“, které jsou ekvivalentními souřadnicemi v systému UTM.

K převodu z UTM do PP slouží metoda `utmToPP(east, north, zone)`, která vrací objekt s atributy „x“ a „y“. Argument „zone“ určuje v jaké zóně systému UTM se daný bod nalézá. Systém UTM je rozdělen do 60 zón.

### 4.6.4.2 Převody mezi PP a WGS84

K převádění souřadnic ze systému PP do WGS84 je určena metoda `mapa.ppToWGS(x, y)`, která přepočítá zadané souřadnice v PP a vrátí objekt s atributy „lat“ a „lon“. Atribut „lat“ představuje zeměpisnou šířku, atribut „lon“ zeměpisnou délku daného bodu v souřadném systému WGS84.

Z WGS84 do PP převedeme souřadnice metodou `mapa.wgsToPP(lat, lon)`.

### 4.6.4.3 Převody mezi UTM a WGS84

Převod mezi pravouhlým souřadným systémem UTM a systémem šířka-délka WGS84 nám umožní metoda `utmToWGS(east, north, zone)` vracející objekt s atributy „lat“ a „lon“.

Ze souřadného systému WGS84 do UTM převedeme souřadnice metodou `wgsToUTM(lat, lon)`. Metoda vrací objekt s atributy „east“, „north“ a „zone“.

### 4.6.4.4 Souřadnice v pixelech

Při publikování dokumentů na webu jsou velmi často používanými jednotkami k určení polohy elementu na stránce pixely.

Souřadnice bodů je třeba znát nejen v zeměpisném souřadném systému, ale také v pixelech. To má význam například při určování zeměpisných souřadnic bodu, kam ukazuje kurzor myši.

V API Mapy.cz použijeme k určení souřadnic bodu v pixelech metodu `mapa.ppToPixel(pp, zoom)`. Tato převádí vzdálenost v PP na pixely při daném přiblížení (argument „zoom“).

Opačnou metodou je `mapa.pixelToPP(pixel, zoom)` převádějící vzdálenost v pixelech na jednotky pp při daném přiblížení.

## 4.6.5 Přidání ovládacích prvků do mapy

Nejprve je třeba vytvořit `layoutBox`, který bude ovládací prvky obsahovat.

```
var layoutBox = mapa.getDefaultLayoutBox();
```

Ovládací prvek pro oddalování a přibližování mapy vytvoříme pomocí API konstruktoru:

```
var zoomControl = new SZN.Visual.ZoomControl('full');
```

Následně prvek přidáme do `layoutBoxu`

```
mapa.addControls(zoomControl, layoutBox, x_position, y_position);
```

Nástroj pro ovládání polhy mapy vytvoříme podobně.

```
var moveControl = new SZN.Visual.MoveControl();
```

A opět vložíme do `layoutBoxu`:

```
mapa.addControls(moveControl, layoutBox, x_position, y_position);
```

## 4.6.6 Ovládání mapy pomocí myši

Možnosti ovládání mapy pomocí myši nastavíme metodou `mapa.mouseSet(setNum)`.

Argument určuje stupeň ovládání, které pomocí myši uživateli povolíme. Může nabývat hodnot od 0 (ovládání myši není povoleno) až po 7 (je povolen pohyb mapou pomocí myši, zoomování kolečkem myši i klikáním).

## 4.6.7 Nastavení středu mapy a přiblížení

Střed mapy určíme metodou `mapa.setCenter(x, y)`, kde „x“ a „y“ jsou souřadnice v souřadném systému PP.

Stupeň přiblížení lze nastavit pomocí `mapa.zoomSet(zoom)`. Parametr „zoom“ je číslo od 3 do 16.

## 4.6.8 Zobrazování mapových podkladů a vrstev

API `Mapy.cz` umožňuje zobrazovat více mapových podkladů (mapu základní, turistickou, fotografickou a historickou). Poklady jsou zobrazovány jako jednotlivé mapové vrstvy, které můžeme mezi sebou přepínat.

Mapové podklady a názvy příslušných mapových vrstev:

- **base** – základní mapa
- **ophoto** – fotomapa satelitní nebo letecká (podle stupně přiblížení)
- **turist** – turistická mapa
- **army2** – historická mapa (druhé vojenské mapování)

K přepínání vrstev slouží metoda `mapa.switchLayer(ukazat, [vypnout])`. Parametr „ukazat“ bude jméno mapové vrstvy, kterou chceme zobrazit, druhým volitelným parametrem je obsahující názvy vrstev, které vypneme. Například zobrazení turistické mapy docílíme zápisem: `mapa.switchLayer("turist");`

Příklad použití metody s více argumenty – následující zápis zobrazí vrstvu „base“ a ostatní vypne:

```
mapa.switchLayer("base", ["base", "ophoto", "turist", "army2"]);
```

Přepínáním podkladů ale možnosti práce s mavovými vrstvami v API nekončí. Máme zde ještě k dispozici několik dalších vrstev s dodatečnými informacemi:

- **hybrid** – hybridní mapa, obsahuje silnice a sídla
- **ttur** – vrstva označuje turistické trasy
- **ctur** – značí cyklostezky
- **relief-h** – stínování reliéfu - tmavé
- **relief-l** – stínování reliéfu - světlé

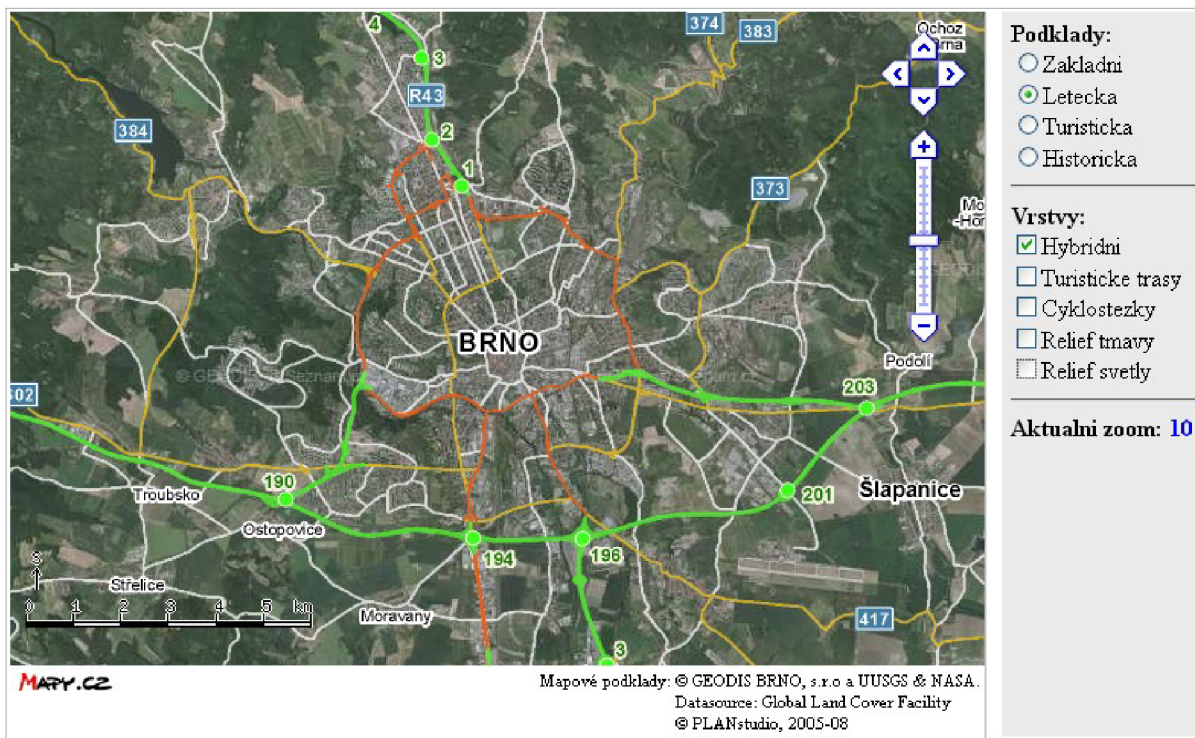
Tyto jsou zobrazeny nad vrstvami reprezentujícími mapové podklady. Vrstvu povolíme zobrazit pomocí metody `mapa.enableLayer(vrstva)`, zakážeme metodou `mapa.disableLayer(vrstva)`, zrušíme metodou `mapa.removeLayer('vector')`.

Na příkladech map níže vidíme použití metod pro práci s mapou v praxi. Metody jsou volány prostřednictvím jednoduchých ovládacích prvků na stránce. Jsou zde prvky pro ovládání posunutí a přiblížení mapy. Dále je možné zobrazit výše uvedené mapové vrstvy.



Obrázek 4.1: Stínování reliéfu na pokkladu základní mapy [3].





Obrázek 4.2: Zobrazení hybridní vrstvy nad fotomapou [3].

## 4.6.9 Práce s vektorovou mapovou vrstvou

API podporuje kreslení úseček. Kreslí se do zvláštní mapové vektorové vrstvy. Zde popíšeme, jakým způsobem vytvoříme vektorovou vrstvu a jak s ní budeme pracovat.

Njprve vytvoříme zdroj vektorové vrstvy:

```
var vecSrc = new SZN.MapEngine.VectorSource();
```

Další potřebný objekt bude vrstva bodů, které chceme úsečkami propojit. Vytvoříme prázdný zdroj této vrstvy a vrstvu samu:

```
var pois = {source:"", data: []};
```

```
SZN.MapEngine.MapLayers.PoiLayer("body", false, 1, pois, null);
```

Metdou `mapa.addLayer(poiLayer)` přidáme vytvořenou vrstvu do mapy. A přistoupíme k vlastnímu kreslení. Určíme výchozí bod kreslení: `vecSrc.moveTo(ppX1, ppY1)`. A vytvoříme další bod: `vecSrc.lineTo(ppX2, ppY2, barva_usecky)`. Třetím parametrem metody bude barva, jakou chceme úsečku nakreslit. Tloušťka čáry je úměrná zvětšení mapy.

Pokud máme zadávání bodů hotové, vytvoříme vlastní vrstvu, přidáme ji do mapy a povolíme její zobrazení.

```
var vecLayer = new SZN.MapEngine.MapLayers.VectorLayer("vector",  
vecSrc, true) //vytvoření vrstvy  
mapa.addLayer(vecLayer); //přidání vrstvy do mapy  
mapa.enableLayer('vector'); //zobrazení vrstvy
```

## 5. Postup při implementaci

Pro zvládnutí implementace bylo nutné seznámit se s výše uvedenými technologiemi a jejich použitím. Implementace je rozdělena na dvě hlavní části:

- 1) Uživatelský skript pro GreaseMonkey a prohlížeč Firefox.
- 2) Podpůrný web umožňující použití JS API.

### 5.1 Uživatelský skript pro GM

Pro implementaci uživatelského skriptu bylo zapotřebí prostudovat DOM strukturu webu [www.mapy.cz](http://www.mapy.cz) a API [Mapy.cz](http://www.mapy.cz). Zjistit možnosti získávání informací o nadmořských výškách a možnosti vykreslování výškového profilu pomocí existujících služeb jako jsou [www.heywhatsthat.com](http://www.heywhatsthat.com), [www.topocoding.com](http://www.topocoding.com) a [www.vyskopis.cz](http://www.vyskopis.cz).

Implementace vlastního vykreslovacího algoritmu a získávání nadmořských výšek za použití JavaScriptového API si vyžádala vytvoření pomocného webu (z něho je možné stáhnout GM skript).

#### 5.1.1 Vytvoření nového skriptu

Pokud máme nainstalované GM do Firefoxu, můžeme napsat první skript. Vytvoření nového skriptu je popsáno v předchozí kapitole.

#### 5.1.2 Odlišnosti GreaseMonkey skriptů

Skripty pro GM mají některé odlišnosti oproti „klasickému“ JavaScriptu psaném přímo do HTML kódu. Uvedeme několik příkladů.

- Nelze přistupovat přímo k event-handlerům. Místo `element.onclick="func()"` použijeme `element.addEventListener("click", func, true)`. Nebo lze při vytváření prvků přidat atribut `onclick` do HTML kódu. Např.: `<p onclick="func()">`.
- Nelze použít cyklus `for (var element in elements) { ... };`. Použijeme konstrukci: `for (var i=0;i<5;i++){ ... };`.
- Nelze `document.getElementsByClassName("class");`, musíme použít objekt `unsafeWindow`:  
`unsafeWindow.document.getElementsByClassName("class");`
- Místo `element.setAttribute("style","color:blue");` použijeme `element.style.color="blue";`.

#### 5.1.3 Přidání ovládacích prvků

Prvním krokem při psaní skriptu bylo přidání vlastního ovládacího prvku do DOM struktury webu. DOM struktura webu [www.mapy.cz](http://www.mapy.cz) je poměrně složitá. Elementy obsahující části mapy jsou obsaženy v bloku s `id="MapBox"`, který obsahuje další bloky s HTML elementy.

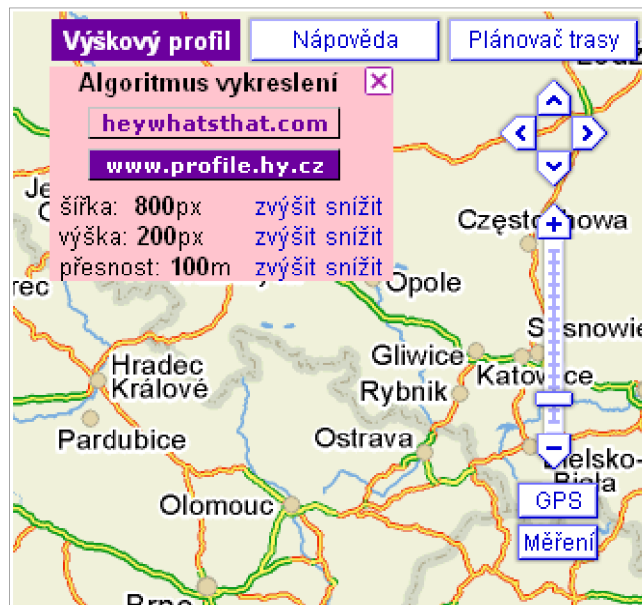
Ovládací prvky přidáme do bloku třídy „port“, jak ukazuje příklad:



```

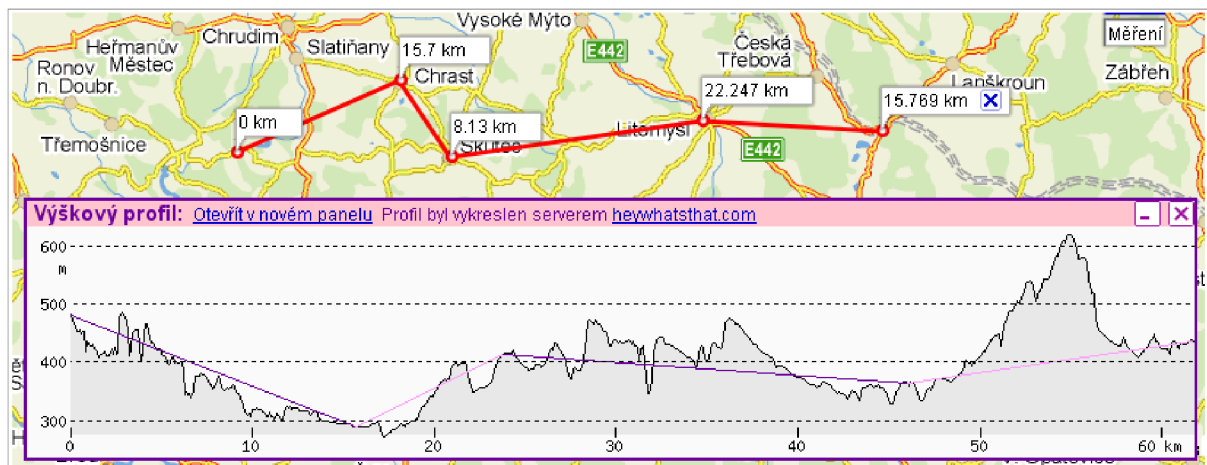
var HTML_kod_prvku = "<div>Ovládací prvek</div>";
var divs =unsafeWindow.document.getElementsByClassName("port");
divs[0].innerHTML+= HTML_kod_prvku;

```



Obrázek 5.1: Začlenění ovládacího prvku do webové stránky

Podobným způsobem přidáme do DOM i výsledný výškový profil.



Obrázek 5.2: Začlenění výškového profilu do webové stránky

### 5.1.4 Získání a zpracování zeměpisných souřadnic.

Pomocí nástroje „Měření“ může uživatel zadávat požadovanou trasu. Souřadnice naklikaných bodů se zapisují do URL stránky. Lze je tedy získat z objektu `document.location`.

Služba <http://www.heywhatstthat.com/profiler.html> používá souřadný systém WGS84. Zeměpisné souřadnice přijímá ve stupních v desítkové soustavě. Vykreslovací algoritmus je spoutěn

na serveru. Pomocí URL jsou mu předány potřebné souřadnice. Ty je třeba získat z `document.location` webu `www.mapy.cz` a pomocí regulárních výrazů a převést je ze souřadného systému PP na WGS84 a do desítkové soustavy.

Vytvoříme-li ve správném formátu URL adresu, aby server správně přečetl pomocí metody GET souřadnice trasy, dostaneme po odeslání požadavku odpověď ve formě HTML stránky s obrázkem nebo přímo vygenerovaný obrázek profilu. Ten začleníme pomocí klientského JavaScriptu do webového dokumentu. Příklad vidíme na obrázku 5.2.

Převody mezi souřadnými systémy realizujeme metodami `mapa.wgsToPP(lat, lon)` a `mapa.ppToWGS(x, y)`. Práce s nimi je popsána v předchozí kapitole. URL pro zobrazení požadovného profilu tedy vytvoříme ve forcyklu přes `unsafeWindow.document.location` ze kterého regulárními výrazy získáme souřadnice.

Podobně vygenerujeme URL potřebnou k vykreslení profilu pomocí algoritmu vytvořeném pro účely této práce. Tento ale přijímá PP souřadnice bodů a umožňuje zadat další parametry k vytvoření grafu. Jsou jimi šířka obrázku v pixelech ( $w$ =šířka), výška v pixelech ( $h$ =výška) a přesnost s jakou bude rozdělovat trasu určenou naklikanými body na části ( $d$ =presnost). Parametry lze zadávat pomocí jednoduchých ovládacích prvků přidanych do webového dokumentu nebo přímo změnou v odkazu na výsledný dokument s vytvořeným profilem. Příklad níže udává URL, podle níž skript vytvoří obrázek o šířce 750px, výšce 250px s přesností na 100m.

```
http://www.profile.hy.cz/?api=GM&w=750&h=250&d=100&i=1&x0=139329536  
&y0=133824512...
```

Protože nešlo použít PHP API pro získávání výškových dat ze serveru [3], získávání těchto dat bylo realizováno za pomoci JS API. Registrace API se váže k doméně, na které je používáno. Z toho důvodu nelze vykreslovací algoritmus implementovat přímo v GM skriptu, musí být vytvořeno podpůrné webové rozhraní se zaregistrovanou doménou. Vlastní začlenění profilu do DOM struktury webu `www.mapy.cz` se tím komplikuje.

Bylo použito HTML tagu `<iframe src='http://www.profile.hy.cz/...'>`. Tento způsob, kdy do existujícího dokumentu začleníme jiný cizí dokument, má jistě nevýhody, ale řeší nastalý problém se zobrazením profilu vykresleného za podpory JS API.

Implementace podpůrné webové aplikace a vykreslovacího algoritmu je popsána v následující kapitole.

Jako další způsob získávání zdrojových souřadnic pro vykreslování profilu bylo zamýšleno použít i nástroje „**Plánovač trasy**“. Tento nástroj umožňuje vyhledávání trasy mezi zadanými místy. Trasa je generována na straně serveru a zobrazována jako vrstva mapy. API `mapy.cz` neumožňuje přístup ke všem souřadnicím této trasy. Lze ze získat souřadnice významných bodů trasy (cílový a počáteční bod a křižovatky na kterých je nutné odbočit při jízdě autem po trase). Těchto bodů je jen malé množství. Vykreslený profil by neodpovídal vygenerované trase.

Získání všech bodů trasy by bylo možné při používání přímo algoritmů k vytvoření trasy a získávání jejich výsledků v podobě souboru souřadnic. Což není ze strany klienta možné, bylo by třeba spolupráce s firmou Seznam.

Další možností získání souřadnic na vygenerované trase je jejich zpětná rekonstrukce z obrázku. Toto řešení by bylo také značně nepřesné při malém přiblížení nebo velmi složité při vysokém přiblížení mapy.

Od záměru získávání souřadnic z vygenerované trasy bylo upuštěno. Práce se ubírala směrem vytvoření vlastního vykreslovacího algoritmu naklikané trasy.

## 5.2 Webová aplikace s mapou

Podpůrná webová aplikace byla nutná k používání JavaScriptového aplikačního rozhraní. Je dostupná na adrese `http://www.profile.hy.cz`.

### 5.2.1 Vložení mapy

Aby tento web nesloužil jen pro samotné vykreslování výškového profilu, je přidána mapa za použití metod API mapy.cz. Postup přidání mapy a k tomu potřebné metody jsou popsány v kapitole 4.6.3. Byly vytvořeny i jednoduché ovládací prvky pro přepínání mapových podkladů a mapových vrstev (kapitola 4.6.8).

API Mapy.cz neumožňuje přímo vložit nástroj k vytvoření trasy. Ten bylo nutné implementovat.

Postup jeho vytvoření je následující:

- Je nutné na globální úrovni udržovat údaje o pozici kurzoru na stránce.
- Nástroj na kreslení trasy musí být možno vypnout, zapnout a přerušit. Přerušení kreslení určuje hodnota logické proměnné. Zrušení a opětovné zapnutí musí být realizováno za pomoci metod `mapa.addLayer(vecLayer); mapa.enableLayer("vector");` při zapínání kreslení, `mapa.disableLayer(vrstva);` a `mapa.removeLayer("vector");` při zrušení kreslení.

- Je nutné znát přesnou polohu prvku obsahující mapu. A jeho přesný střed. Pozici určíme následovně:

```
var x = mojeMapa.offsetLeft;  
var y = mojeMapa.offsetTop;
```

- Po zapnutí nástroje na kreslení musíme přepočítávat polohu kurzoru vůči mapě při každém pohybu kurzoru. Definujeme funkci, která bude při pohybu kurzoru přepočítávat jeho polohu na stránce vůči bodu, na který ukazuje na mapě určením středu mapy metodou `mapa.getCenter()` a převedením na pixely pomocí `mapa.ppToPixel(pixel, zoom)`. Máme-li střed mapy v pixelech, vztahujeme ho ke středu prvku obsahujícímu mapu. A pomocí vzdálenosti kurzoru od tohoto středu přepočteme vzdálenost bodu na který kurzor uazuje od středu mapy. Definice funkce pracující s polohou kurzoru:

```
function Trace_mouse(evnt) {  
X_pos=(event.clientX + document.body.scrollLeft);  
Y_pos=(event.clientY + document.body.scrollTop);  
... }  
...  
document.onmousemove = Trace_mouse;
```

- Posledním krokem k vytvoření kreslicího nástroje je přidání události `onclick="KlikNaMapu()"` prvku obsahujícímu mapu. Funkce `KlikNaMapu()` vykreslí úsečku do mapy. Použije metody pro práci s vektorovou vrstvou (kapitola 4.6.9).

## 5.2.2 Vykreslovací algoritmus

Registrace JS API `www.vyskopis.cz` [4] vztahuje k doméně „`http://www.profile.hy.cz`“. Použití na jiné adrese např. „`http://www.profile.hy.cz/skript.php`“ by nefungovalo. Web je tvořen souborem `index.php`, který pokud nemá v požadavku GET definovanou proměnnou `GET_api`, zobrazí webové rezraní s mapu. Pokud je `GET_api == 'GM'` vykreslí profil na základě parametrů v URL. Toto větvení je realizováno pomocí jazyka PHP. Vykreslování je realizováno pomocí Javascriptu.

Při zobrazování do DOM struktury `www.mapy.cz` v GreaseMonkey má tento způsob vykreslování jednu nevýhodu. Na stránce se zobrazuje reklamní lišta. Po uvážení bylo rozhodnuto tuto lištu nezakrývat z důvodů licenčních podmínek serveru `www.hy.cz` a snadné přenositelnosti zdrojového kódu webu na jiný server. Protože pro uživatele nepředstavuje výrazné omezení nebo překážku.

Zdrojové kódy jsou snadno přenositelné, stačí jen změnit příslušné URL adresy a registrační klíče k aplikačním rozhraním.

### 5.2.2.1 Zpracování parametrů

Parametry získává skript pomocí regulárních výrazů z objektu `document.location`. Ty zpracovává a ukládá je do objektu použitému jako asociativní pole. Příklad zpracování souřadnic:

```
var pointarr = new Object();
for (var i=0;i<=maxindex;i++){
pointarr["x"+i] = getxpp(i); /*dynamicky vytváříme argumenty objektu
pointarr["y"+i] = getypp(i); *pointarr */
}
```

Body procházíme dvakrát. Při prvním průchodu získáváme souřadnice z `document.location` a vypočteme celkovou vzdálenost. Cyklus začíná indexem 0 a končí indexem určeným proměnnou „`i`“ v URL.

V druhém průchodu rozdělíme trasu na úseky, jejichž velikost je určena proměnnou „`d`“ získanou z `document.location`. Zároveň vypočítáváme každému takto vytvořenému bodu jeho vzdálenost od počátku trasy.

Známe-li souřadnice a vzdálenosti všech bodů na které byla trasa rozdělena, vytvoříme zdrojové pole pro použití funkce `topoGetAltitudes(quadriples,timeout)`. Popis parametrů této funkce najdeme v kapitole 3.2.2.3 nebo na [4].

Funkce `topoGetAltitudes()` musí být často volána opakovaně. Závisí to na počtu bodů ležících na trase. Tato funkce zavolá při jednom dotazu maximálně 280 krát událost „`action`“ (viz. popis funkce), které předá parametr „`altitude`“. Předaná nadmořská výška je pro každý bod ukládána objektu „`alt`“, požívaném jako asociativní pole (např. `alt["index"] = altitude`).

Pokud známe hodnoty nadmořských výšek všech požadovaných bodů, můžeme přistoupit ke kreslení výškového grafu.

### 5.2.2.2 Kreslení profilu

V předchozí kapitole jsme si ukázali, jak získat požadované souřadnice trasy, jak trasu rozdělit na menší úseky o dané velikosti a následně získat nadmořské výšky bodů na trase.

Vykreslování profilu je realizováno prostřednictvím DHTML tagu `<canvas>` a JavaScriptových funkcí umožňujících kreslit do canvasu jednoduché grafické objekty [10].

Nejprve získáme požadovanou šířku a výšku celého obrázku. Od šířky a výšky odečteme okraje potřebné k vykreslení souřadných os a jednotek na nich. Vypočítáme poměr mezi celkovou délkou trasy v metrech a šířkou obrázku. Zjistíme maximální a minimální nadmořskou výšku trasy a vypočteme stejný poměr pro výšku obrázku.

Cyklem procházíme celou trasu. Tentokrát nás zajímá u každého bodu vzdálenost od počátku trasy a jeho nadmořská výška. Vzdálenost od počátku trasy a nadmořskou výšku bodu (od které odečteme minimální nadmořskou výšku) vynásobíme těmito poměry a můžeme vykreslovat hraniční úsečky vyplňovaného polygonu. Při posledním průchodu trasy je vypočítána i skutečná délka trasy pomocí Pythagorovy věty a je zobrazena pod výškovým profilem. Výškový profil vykreslujeme jako vyplněný polygon.

Zjednodušený postup při vykreslování:

```
var elem = document.getElementById("myCanvas");

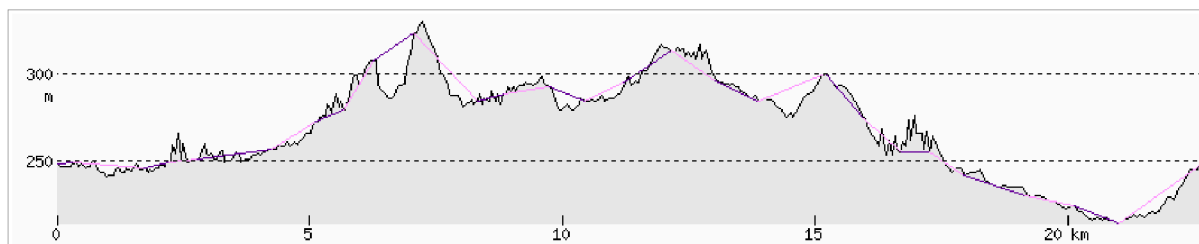
if (elem && elem.getContext) {
    var context = elem.getContext('2d');
    if (context)
    {
        context.fillStyle = "#e0ffff"; //barva výplně
        context.strokeStyle = "#660099"; //barva čáry
        context.lineWidth = 0.5; //tloušťka čáry
        context.beginPath(); // Začíná se v levém horním rohu
        context.moveTo(x0, y0); //výchozí bod
        context.lineTo(x1, y1); //další bod obrysu
        context.lineTo(x2, y3);
        ... //určení dalších bodů polygonu
        context.lineTo(x0, y0); //zpět do počátku
        context.fill(); //vyplnění polygonu
        context.stroke(); //vykreslení obrysu
        context.closePath();
    }
}
```

Text (jako jsou popisky os) nelze psát do canvasu přímo, canvas podporuje vykreslování textu jen v nejnovější verzi Firefoxu. Text je třeba emulovat prostřednictvím naprogramovaných funkcí umožňujících vykreslení textu. V projektu bylo použito skriptu dostupného na [11]. Ten umožňuje vykreslovat jednoduchý text do canvasu.

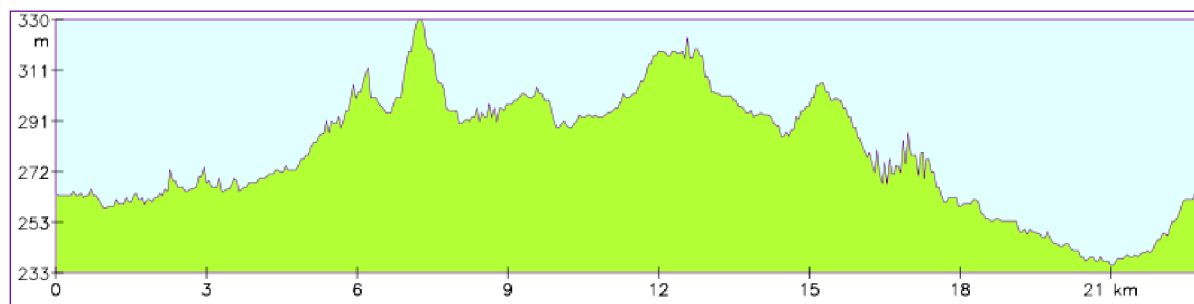
## 5.2.3 Ukázky vykreslených výškových profilů

Vytvořený algoritmus má několik výhod. Jednou z nich je volitelná šířka a výška vytvořeného obrázku. Můžeme vytvořit obrázek například 150 x 3000 pixelů. Profily delších tras tak budou vypadat „reálněji“. Dále můžeme zvolit různou přesnost s jakou se má profil vykreslit. Při vykreslování je vypočtena skutečná délka trasy.

### 5.2.3.1 Porovnání vykreslovacích algoritmů



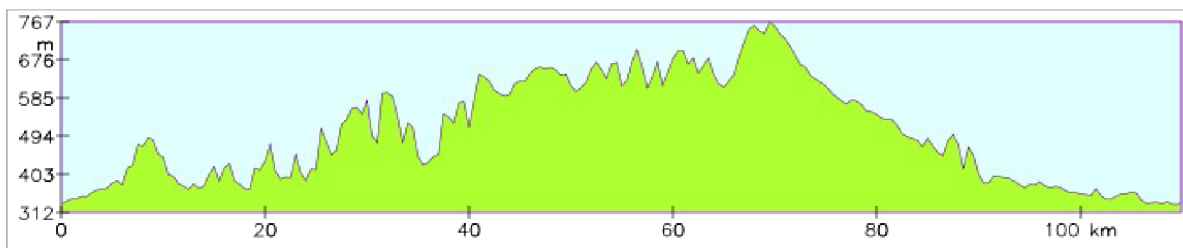
Obrázek 5.3: Profil trasy Tišnov – Brno vykreslený pomocí služby <http://www.heywhatsthat.com/profiler.html>



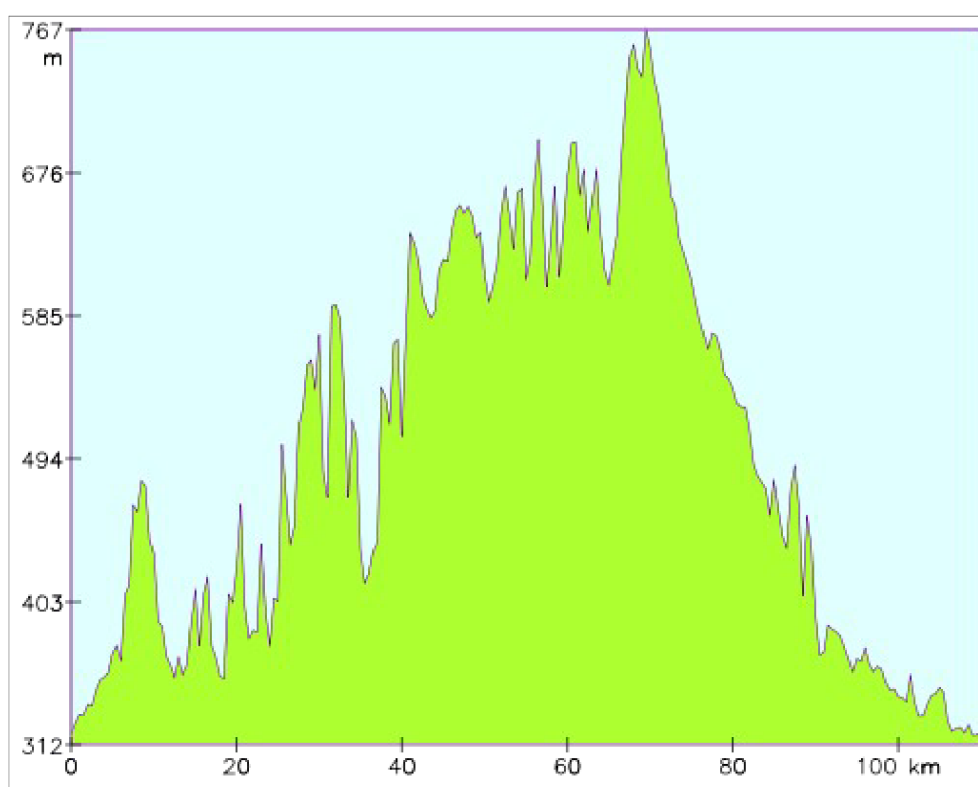
Skutečná délka trasy je: 22760 metrů

Obrázek 5.4: Profil trasy Tišnov – Brno vykreslený pomocí vytvořeného algoritmu (přesnost 50 m).

### 5.2.3.2 Různé velikosti obrázků

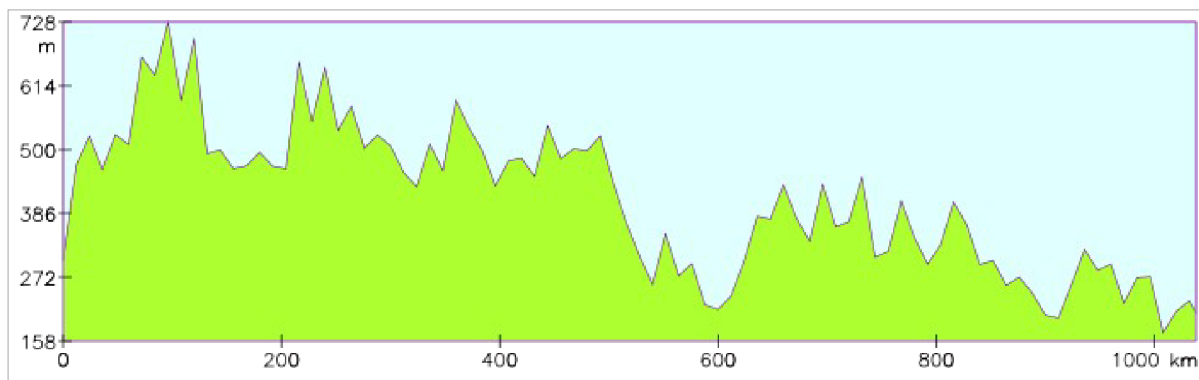


Obrázek 5.5: Profil trasy Brno – Česká Třebová 700 x 150 pixelů.

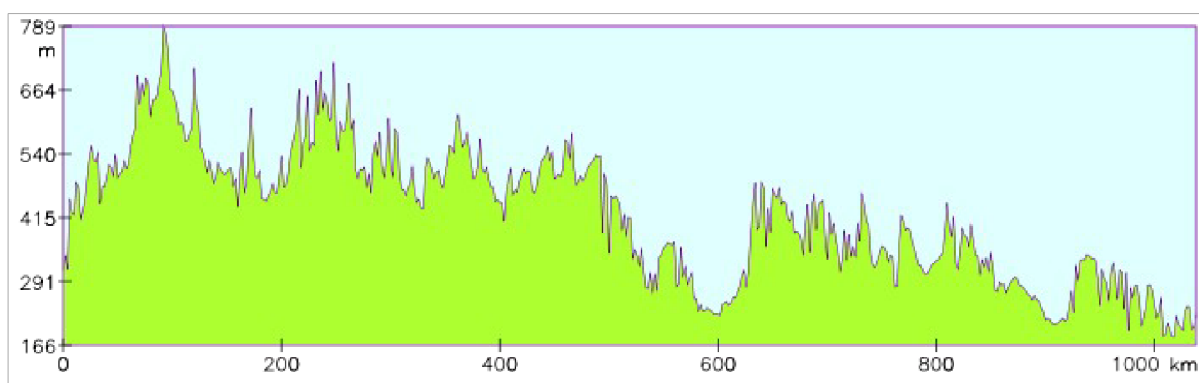


Obrázek 5.6: Profil trasy Brno – Česká Třebová 400 x 400 pixelů.

### 5.2.3.3 Porovnání různých přesností vykreslení



Obrázek 5.7: Profil trasy Brno – Paříž (přesnost 12.000 metrů)



Obrázek 5.8: Profil trasy Brno – Paříž (přesnost 2.000 metrů)



## 6. Závěr

Podařilo se navrhnout a implementovat získávání souřadnic z mapy a vykreslování výškového profilu trasy pomocí <http://www.heywhatsthat.com/profiler.html> na veřejném GISu [www.mapy.cz](http://www.mapy.cz) podle zadání. Souřadnice jsou získávány a dále zpracovávány pomocí GreaseMonkey (rozšíření pro Firefox).

GreaseMonkey umožňuje pomocí JavaScriptu pracovat s objekty DOM struktury webového dokumentu. Byla zanalyzována DOM struktura webového GISu [www.mapy.cz](http://www.mapy.cz) a vytvořen skript pro GreaseMonkey realizující získávání souřadnic trasy a vykreslování jejího výškového profilu.

Nejprve bylo nutné seznámit se s DOM strukturou webu [www.mapy.cz](http://www.mapy.cz) a s možností práce s ní pomocí JavaScriptu a GreaseMonkey. Byl implementován skript, který se spouští po načtení stránky [www.mapy.cz](http://www.mapy.cz) prohlížečem (Firefox), přidá do DOM struktury stránky vlastní ovládací prvek, umožní získání souřadnic naklikané trasy a vykreslí výškový profil. Souřadnice bodů, které uživatel zadá pomocí nástroje „měření“, jsou získávány z objektu `document.location`. Získávání souřadnic trasy vytvořené pomocí nástroje „plánovač trasy“ by bylo příliš složité nebo nepřesné, proto nebylo implementováno. Práce se ubírala směrem k vytvoření vlastního algoritmu pro vykreslení výškového profilu.

Původně bylo zamýšleno informace o nadmořských výškách bodů na trase získávat pomocí podpůrného PHP skriptu a PHP API ze stránek [www.topocoding.com](http://www.topocoding.com). PHP API se ukázalo jako nefunkční. Jeho alternativou je JavaScriptové API služby [www.vyskopis.cz](http://www.vyskopis.cz). Použití je podmíněno registrací adresy webu, na kterém budeme API používat. Abychom mohli vykreslovat výškový profil JavaScriptového API a začlenit ho do DOM struktury stránky [www.mapy.cz](http://www.mapy.cz), bylo nutné vytvořit podpůrnou webovou aplikaci.

Vykreslení výškového profilu je implementováno v JavaScriptu. Skript je začleněn do pomocného webu na adrese [www.profile.hy.cz](http://www.profile.hy.cz). Souřadnice bodů na trase skript získává z URL. Kromě souřadnic jsou mu předány i parametry určující šířku a výšku obrázku a požadovanou přesnost vykreslení. Hodnoty nadmořských výšek bodů na trase jsou získávány vhodným použitím funkcí JS API z [www.vyskopis.cz](http://www.vyskopis.cz). Samotné vykreslení grafu je realizováno pomocí kreslicích funkcí pro DHTML element `<canvas>`[10]. Pro zobrazení takto vykresleného profilu do DOM struktury webu [www.mapy.cz](http://www.mapy.cz) pomocí GM byl použit HTML tag `<iframe>`. Tento způsob začlenění do stránky má své nevýhody, řeší ale problém s možností použití JS API jen na registrované doméně.

Uživateli jsou tedy k dispozici dva způsoby vykreslování profilu pomocí GM na webu [www.mapy.cz](http://www.mapy.cz). Lze použít již existující služby <http://www.heywhatsthat.com/profiler.html> nebo vlastního vykreslovacího algoritmu umístěného na webu [www.profile.hy.cz](http://www.profile.hy.cz).

Na tuto práci je možné navazovat a vytvářet další funkcionalitu. Potenciál je například v rozšíření GUI o další možnosti volby vlastností grafu (např. barva vykreslení, barva pozadí, druh a velikost písma na popiscích os). Nebo ve vytvoření neobvyklých ovládacích nástrojů pro práci s mapou. Například výběr oblasti určené vzdáleností od daného bodu, výpočet plochy ohraničené oblasti nebo implementovat způsob získávání souřadnic trasy podobný „plánovači trasy“. Zajímavé by bylo navrhnout způsob vykreslování 3D profilu vybrané oblasti.

Dalším možným rozšířením by bylo vytvořit vlastní zdroj výškových dat. Data jsou volně dostupná. Je třeba zprovoznit mapový server který bude na základě vstupních parametrů vytvářet výškový profil nebo vracet soubor dat potřebný k vykreslení profilu.

Jako alternativu mapového severu by bylo možno vytvořit databázi výškových dat. SRTM data by bylo nutné zpracovat pomocí GIS softwaru (například GRASS) a následně je uložit do databáze. Nevýhodou při použití vlastního datového zdroje je velikost SRTM dat (v celosvětovém měřítku je v řádu terabytů). Toto řešení by bylo reálně možno pracovat jen s omezenou oblastí (např. jen pro ČR).

# Literatura

- [1] Hrubý, M. *Geografické Informační Systémy, Studijní opora*. [online]. Dostupné na URL: <<http://perchta.fit.vutbr.cz:8000/vyuka-gis>>. [cit. 2009-05-23].
- [2] *HeyWhatsThat Path Profiler*. [online]. Dostupné na URL: <<http://www.heywhatsthat.com/profiler.html>>. [cit. 2009-05-23].
- [3] *Topocoding API, Documentation of PHP API* [online]. Dostupné na URL: <<http://www.topocoding.com>>. [cit. 2009-05-23].
- [4] *Výškopis České republiky, Dokumentace*. [online]. Dostupné na URL: <<http://www.vyskopis.cz>>. [cit. 2009-05-23].
- [5] *Wikipedia - otevřená encyklopedie*. [online]. Dostupné na URL: <<http://cs.wikipedia.org>>.
- [6] *Google - internetový vyhledáveč*. [online]. Dostupné na URL: <<http://www.google.cz>>.
- [7] *Arcdata Praha - geografické informační systémy*. [online]. Dostupné na URL: <<http://www.arcdata.cz/uvod/>>. [cit. 2009-05-23].
- [8] *API Mapy.cz* [online] Dostupné na URL: <<http://api.mapy.cz/>>. [cit. 2009-05-23].
- [9] FLANAGAN, D.: *JavaScript: Kompletní průvodce*. Praha 4: Computer Press, druhé vydání, 2002, ISBN 80-7226-626-8.
- [10] *Drawing Graphics with Canvas* [online] Dostupné na URL: <[https://developer.mozilla.org/en/drawing\\_graphics\\_with\\_canvas](https://developer.mozilla.org/en/drawing_graphics_with_canvas)>. [cit. 2009-05-23].
- [11] *Primitive Text Functions for the HTML5 <canvas> Element* [online] Dostupné na URL: <<http://jim.studt.net/canvastext/>>. [cit. 2009-05-23].
- [12] *Earth Resources Observation and Science*. [online]. Dostupné na URL: <<http://edc.usgs.gov/products/elevation/gtopo30/gtopo30.html>>. [cit. 2009-05-23].
- [13] *NGA Raster Roam*. [online] Dostupné na URL: <[http://geoengine.nga.mil/geospatial/SW\\_TOOLS/NIMAMUSE/webinter/rast\\_roam.html](http://geoengine.nga.mil/geospatial/SW_TOOLS/NIMAMUSE/webinter/rast_roam.html)>. [cit. 2009-05-23].
- [14] *ftp NASA*. [online]. Dostupné na URL: <<ftp://e0srp01u.ecs.nasa.gov/srtm/version2/SRTM3/>>. [cit. 2009-05-23].
- [15] *Mapový server snadno a rychle*. [online]. Dostupné na URL: <<http://www.root.cz/clanky/mapovy-server-snadno-a-rychle-1/>>. [cit. 2009-05-23].

# Seznam příloh

Příloha 1. CD:

Příložené CD obsahuje:

- Text práce ve formátu PDF a ODT
- Skript pro GreseMonkey: `vyskovy_profil.user.js`
- Návod na instalaci a použití skriptu ve formátu PDF
- Zdrojové soubory webu `www.profile.hy.cz`