

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

## PROSTŘEDÍ PRO TVORBU MAP Z DAT PROJEKTU OPENSTREETMAP

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PETR OPLETAL

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

# PROSTŘEDÍ PRO TVORBU MAP Z DAT PROJEKTU OPENSTREETMAP

MAP CREATION ENVIRONMENT USING OPENSTREETMAP PROJECT DATA

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PETR OPLETAL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. TOMÁŠ NOVOTNÝ

BRNO 2014

## **Abstrakt**

Práce se zabývá tvorbou prostředí pro tvorbu map z vektorových dat projektu OpenStreetMap. Výsledná aplikace umožňuje měnit vzhled map a importovat data z externích zdrojů (např. vlastní trasy). V práci je popsán teoretický úvod, informace o projektu OpenStreetMap a informace o aplikaci Google My Tracks. Další část práce se již zabývá návrhem, vlastní implementací aplikace a jejím testováním. V závěru práce je shrnut dosažený výsledek a zmíněny některé návrhy na rozšíření.

## **Abstract**

This bachelor's thesis deals with the development of environment for creating maps using OpenStreetMap vector data. The final application allows to change appearance of maps and to import data from external sources (e.g. own tracks). The text contains a theoretical introduction, the information about the OpenStreetMap project and the information about the Google My Tracks application. Further, this text deals with the application design, implementation and testing. The conclusion summarizes achieved results and suggests some proposals how to extend the application.

## **Klíčová slova**

Maps, OpenStreetMap, Mapnik, Google Moje Trasy, tvorba map, kartografie, Qt toolkit.

## **Keywords**

Maps, OpenStreetMap, Mapnik, Google My Tracks, map creation, cartography, Qt toolkit.

## **Citace**

Petr Opletal: Prostředí pro tvorbu map z dat projektu OpenStreetMap, bakalářská práce, Brno, FIT VUT v Brně, 2014

# Prostředí pro tvorbu map z dat projektu Open-StreetMap

## Prohlášení

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

.....

Petr Opletal  
19. května 2014

## Poděkování

Chtěl bych poděkovat svému vedoucímu Ing. Tomáši Novotnému za odbornou pomoc při vypracování této práce.

© Petr Opletal, 2014.

*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

<b>1 Úvod</b>	<b>3</b>
<b>2 Mapy</b>	<b>4</b>
2.1 Obsah mapy . . . . .	4
2.1.1 Výškopis . . . . .	4
2.1.2 Polohopis . . . . .	4
2.1.3 Popis mapy . . . . .	5
2.1.4 Měřítko mapy . . . . .	5
2.2 Kartografické zobrazení . . . . .	5
2.2.1 Mercatorovo zobrazení . . . . .	5
2.3 Vznik mapy . . . . .	6
2.4 Klasifikace map . . . . .	6
2.4.1 Klasifikace map podle obsahu . . . . .	7
2.5 Souřadnicové systémy . . . . .	7
2.5.1 Typy souřadnicových systémů . . . . .	8
2.6 Geografické informační systémy (GIS) . . . . .	9
2.7 Globální polohovací systém (GPS) a Galileo . . . . .	9
<b>3 Projekt OpenStreetMap</b>	<b>11</b>
3.1 Formát dat . . . . .	11
3.2 Vlastnosti map . . . . .	12
3.3 OpenStreetMap API . . . . .	12
3.4 Dostupné vykreslovací nástroje a programy . . . . .	13
3.4.1 Mapnik . . . . .	13
3.4.2 Osmarender . . . . .	13
3.4.3 TileMill . . . . .	14
3.4.4 Maperitive . . . . .	14
3.4.5 OSM-3D . . . . .	14
<b>4 Aplikace MyTracks</b>	<b>15</b>
4.1 Keyhole Markup Language . . . . .	15
4.2 Google Chart API . . . . .	15
<b>5 Návrh aplikace</b>	<b>17</b>
5.1 Cíl tvorby aplikace . . . . .	17
5.2 Mapnik . . . . .	17
5.2.1 Mapnik XML Stylesheet . . . . .	18
5.2.2 Architektura zásuvných modulů . . . . .	19

5.2.3	Stínování kopců	20
5.2.4	Knihovna Qt	20
<b>6</b>	<b>Implementace</b>	<b>21</b>
6.1	Použité vývojové prostředí a třídy	21
6.2	Třída MainWindow	21
6.2.1	Načtení stylovacího souboru s mapou	22
6.3	Třída MapWidget	23
6.4	Třída TreeWidget	23
6.5	Třída Stylesheet	23
6.6	Třída StylesheetParser	24
6.7	Třída Rule	24
6.8	Třída EditLayerDialog	25
6.9	Třída EditStyleDialog	25
6.10	Třída EditSymbolizerDialog	25
6.11	Třída ExportMapDialog	25
6.11.1	Načtení a zpracování externích dat	26
<b>7</b>	<b>Testování</b>	<b>28</b>
7.1	Světová mapa rozložení populace	28
7.2	Mapa detailu centra města	29
7.3	Mapa s turistickým vzhledem	29
<b>8</b>	<b>Závěr</b>	<b>31</b>
8.1	Návrh rozšíření	31
<b>A</b>	<b>Ukázka okna výsledné aplikace</b>	<b>34</b>
<b>B</b>	<b>Ukázka exportovaných map</b>	<b>35</b>
<b>C</b>	<b>Obsah CD</b>	<b>39</b>

# Kapitola 1

## Úvod

Nálezy prvních map sahají až k počátkům civilizace. Již tehdy byly mapy důležitým společníkem a pomáhají lidem dodnes. S postupem času se nezměnil jejich význam, ale přibyly nové typy a směry využití o kterých se našim předkům ani nesnilo. Mapy tedy stále slouží ke stejnému primárnímu účelu, kterým je navigace v terénu, ale díky rozvoji kartografie a moderním informačním technologiím pronikají do dalších odvětví. Neboť je již celý svět objeven a do určité míry detailu zmapován, tak budoucnost našeho digitálního věku spočívá spíše ve tvorbě aplikací, které obsah těchto map zpřístupní v co nejjednodušší formě širokému okruhu lidí. S rozvojem internetu a mobilních technologií máme téměř neomezený přístup k různým mapám v digitální podobě. Technologie nám umožnila doplnit papírová vydání map o různé popisky, interaktivní vyhledávání, prostorové mapy, zobrazení trasy či tvorbu uživatelských map s vlastním obsahem.

Cílem této bakalářské práce bylo vytvořit program zpracovávající data z projektu OpenStreetMap, který umožňuje tvorbu vlastních uživatelských map. Lze tedy do jisté úrovně ovlivnit vzhled výsledných map a importovat do nich externí data, kterými jsou například vlastní trasy.

V následující kapitole je popsán teoretický základ map pojednávající o tom, co vlastně mapa je, jaké existují typy zobrazení, jak mapa vzniká. Je zde zmíněna i navigační technologie GPS. Další kapitola podává informace o komunitním projektu OpenStreetMap. Seznamuje čtenáře s cílem tohoto projektu, formátem dat a rozhraním pro programování aplikací (API) pro přístup k OpenStreetMap databázi. Čtvrtá kapitola má za cíl nastínit klíčové prvky aplikace „Moje trasy“ (My Tracks). Pátá kapitola se zabývá již vlastním návrhem aplikace a popisem vybrané knihovny a vykreslovacího nástroje. Následující kapitola popisuje již vlastní implementaci programu. Předposlední část tohoto dokumentu vysvětluje způsob testování. Závěrečná kapitola shrnuje vlastní výsledky dosažené tvorbou této práce.

## Kapitola 2

# Mapy

Prolistuje-li čtenář různé geografické publikace, atlasy či učebnice, tak nalezne více či méně se lišící definice a vymezení pojmu *mapa*. Ve své podstatě se jedná vždy o zmenšené a do určité míry abstrakce zjednodušené znázornění zemského povrchu zobrazené do roviny pomocí předem definovaných matematických vztahů. Důležitá je i dohodnutá forma zobrazení, která je dána pomocí smluvených znaků rozmístění a vlastnostmi skutečných objektů a jevů [14].

### 2.1 Obsah mapy

Obsahem mapy se rozumí všechny prvky, které jsou součástí mapy. Z hlediska kartografie jsou velice důležitými prvky: výškopis, polohopis a popis. Dalšími významnými prvky, které ovšem nejsou součástí vlastní kresby a leží tedy na okraji, jsou například: měřítko mapy, legenda, údaje o autorovi, druhu tisku atd [8].

#### 2.1.1 Výškopis

Výškopis poskytuje informace o výškovém obraze terénu. U rovinných map je v současnosti nejčastějším zobrazením výškopisu forma vrstevnic. Vrstevnice je spojnice, která spojuje body s totožnou nadmořskou výškou. Pro vlastní tvorbu mapy je nutné zvolit vhodný výškový interval mezi jednotlivými vrstevnicemi (*ekvidistanci*), která se odvíjí od měřítka mapy a členitosti povrchu (obvykle je však v řádech metrů). U prostorových map (např. digitální 3D model pohoří) je vlastní výška zobrazena v příslušném měřítku pomocí třetí dimenze. Oba tyto přístupy lze doprovázet vybarvením ploch barvou příslušné nadmořské výšky, odborně nazývané *barevná hypsometrie* [6].

#### 2.1.2 Polohopis

Popisuje horizontální situaci mapy, jedná se tedy o rozložení objektů na mapě. Pro znázornění těchto objektů využívá bodových, čárových a plošných značek.

Bodové značky slouží pro zobrazení budov, pramenů, výškových bodů... Dalším typem univerzálních mapových značek jsou čárové značky pro grafické zobrazení komunikací, vodních toků, železnic, elektrického vedení atd. Posledním typem jsou plošné značky, které jsou určeny například pro popis lesů, polí, luk nebo různých významných území.

Vysvětlení těchto značek je obsaženo v legendě, které bývá součástí mapy a nejčastěji se vyskytuje na jejím okraji [6].



### 2.1.3 Popis mapy

Podává nám důležité informace o mapovaném území. Popis jednotlivých prvků mapy rozlišujeme například podle velikosti, barvy či sklonu písma. Stejně prvky mapy by kvůli přehlednosti měly být popsány stejným stylem písma. K významným objektům na mapě zpravidla přísluší písmo větší či můžeme příslušný text zvýraznit podtržením. Některé barvy popisek obyčejně podléhají barvě cílového objektu, tedy popisek moře bývá modrý apod.

### 2.1.4 Měřítko mapy

Tento prvek mapy většinou již nebývá součástí vlastní kresby, ale pro vlastní práci s mapou je tento údaj velice důležitý. Měřítkem mapy se rozumí poměr (1 :  $m$ ), kde  $m$  udává zmenšení oproti skutečnosti. Nejčastěji bývá měřítko uvedeno v číselné podobě (např. 1 : 100 000), ale můžeme se setkat i s grafickým nebo slovním (1 cm na mapě odpovídá 10 km v terénu). Měřítko lze také dělit podle podrobnosti zobrazení na malá, střední a velká [14].

## 2.2 Kartografické zobrazení

Převedením zemského povrchu (či jiného nebeského tělesa) do roviny se zabývá matematická kartografie. Trojrozměrné těleso je za pomoci matematických rovnic převedeno do dvojrozměrného prostoru a vždy dochází k určitému zkreslení. Nezkreslují se buď délky (*ekvidistantní*), úhly (*konformní*) nebo plochy (*ekvivalentní*). Zkresluje-li se do určité míry všechno, tak hovoříme o klasifikaci *kompensační* [12].

Podle plochy na kterou těleso zobrazujeme rozlišujeme zobrazení následovně: azimutální (na rovinu), válcové (na plášť válce), kuželové (na plášť kužele) a obecné (odvozené od předcházejících). První tři typy z výše zmíněných nazýváme jednoduchá kartografická zobrazení. Mezi jejich společné rysy patří např. převod referenční plochy na jednoduché zobrazovací plochy či výskyt jediné proměnné v zobrazovacích rovnicích.

**Azimutální** zobrazení se používá při převodu referenčního tělesa přímo do roviny. Nejčastěji je užíváno pro území kulového charakteru, kterým je například oblast kolem pólů. Je-li tečná plocha kolmá na zemskou osu, tak hovoříme o normálním (polárním) typu azimutálního zobrazení. Obrazy poledníků v normální poloze tvoří trs polopřímek vycházejících z pólu a rovnoběžky jsou zobrazeny jako soustředné kružnice.

U **válcového** zobrazení se plocha glóbu nejprve převede na plochu pláště válce, která se poté rozvine do roviny. Dříve se používalo pro zobrazení mapy světa, ale později se od něj upustilo, protože docházelo k příliš velkému zkreslení ploch na pólech. Je vhodné použít pro pásy rozkládající se buď kolem rovníku nebo kolem některého z poledníků. Válcové zobrazení v normální poloze zobrazuje rovnoběžky a poledníky jako přímky navzájem na sebe kolmé.

**Kuželová** zobrazení vznikají zobrazením referenční plochy na vnitřní plášť kužele. Se zobrazovanou plochou mají společnou buď jednu nebo dvě vzájemně soustředné kružnice. V normální poloze je délkově zachována některá z rovnoběžek a poledníky tvoří trs polopřímek, které prochází počátkem souřadnicového systému. Nejčastěji se používají pro zobrazení oblastí kolem rovnoběžek [12].

### 2.2.1 Mercatorovo zobrazení

Je to úhlojevné (*konformní*) zobrazení, které bylo navrženo v roce 1569 kartografem a matematikem Gerhardem Mercatorem. Jedná se o válcové zobrazení v normální poloze, které

se dotýká glóbu na rovníku. Zkreslení tohoto zobrazení roste směrem od rovníku k pólům. Pro zobrazení polárních oblastí je kvůli plošným rozměrům nepoužitelné. Zobrazení rovníku je nerealizovatelné, jednalo by se o nekonečně dlouhý pás o šířce zobrazeného rovníku. Výhodou tohoto zobrazení je, že lokálně zachovává tvar objektů a dochází pouze k plošnému zkreslení. Tuto projekci využívají některé internetové mapové aplikace, např. Mapy Google (*Google Maps*) nebo projekt *OpenStreetMap* [14].

## 2.3 Vznik mapy

Mapy obecně vznikají dvojitým způsobem: původním mapováním v terénu nebo odvozením od již existujících map. Oba tyto přístupy jsou náročné na použité technologie a v dnešní době nám velice usnadňuje práci použití nejmodernějších informačních technologií. Celý proces vzniku mapy lze rozdělit do pěti vývojových etap [8, 4].

První z těchto etap jsou **astronomické práce**. Účelem těchto prací je určit pro mapované území co nejpřesněji zeměpisnou polohu (zeměpisnou šířku a délku) několika základních bodů. Tato část tvorby mapy je z hlediska času poměrně náročná.

Navazující etapou jsou **geodetické práce** [14]. Skládají se ze dvou částí doplňujících body vzniklé astronomickými pracemi: vytvoření trigonometrické a nivelační sítě. Trigonometrická síť je trojúhelníková síť, která vzniká z bodů naměřených teodolitem. Je základem pro polohopis mapy. V terénu jsou tyto body vyznačeny kamennými hranoly a nejdůležitější body bývají speciálně vyznačeny (např. základní trigonometrický bod Sněžka - 1602,02 m n.m.). Pro Českou republiku je výchozí trigonometrický bod Pecný ležící jihovýchodně od našeho hlavního města. Nivelační síť vzniká přesným měřením nadmořských výšek požadovaných bodů. Používá se pro výškopis mapy. U nás a v některých dalších zemích používáme výškový systém nazývaný „baltský po vyrovnání“. Výchozím bodem s nulovou nadmořskou výškou pro tento systém je mořský vodočet umístěný na břehu Baltského moře v Kronštadtu nedaleko Petrohradu. Základní nivelační bod pro Českou republiku se nachází v Lišově v jižních Čechách (564,7597 m n.m.).

Třetí etapou jsou **topografické práce**, které se dále dělí na práce v terénu a na vyhodnocování leteckých snímků (*fotogrammetrické metody*). Cílem topografů je zanezt do mapy veškerý její obsah, tedy všechny důležité prvky nacházející se na mapovaném území. Výsledkem prací v terénu je podrobné polohopisné a výškopisné měření, podrobný topografický obsah mapy určují až fotografické snímky.

Předposlední částí vzniku mapy jsou **kartografické práce**. Jejich úkolem je zpracovat výsledky topografických prací a dát mapě požadovanou podobu. Pokud lze předmět zobrazit celý v měřítku mapy, tak jej vyznačíme obrysovou čarou. Malé předměty jsou značeny vhodnou značkou. Když se nachází více objektů na malém území a nelze je všechny zobrazit, tak je na kartografech, které důležité objekty ponechají (*selekce*). Výsledkem kartografických prací je tzv. čistokresba výsledné mapy.

Ukončující etapou celého procesu tvorby mapy jsou **reprodukční práce**. Význam těchto prací je nezanedbatelný, protože se starají o vlastní tisk a vytváření kopií map. Obor, který se zabývá touto problematikou, se nazývá *kartografická polygrafie*.

## 2.4 Klasifikace map

Mapy jako základní kartografické dílo se můžou rozdělit do mnoha kategorií podle různých hledisek. Lze je klasifikovat podle rozsahu zobrazeného území, podle účelu, podle způsobu

vzniku, podle časového hlediska, podle měřítka, podle formy záznamu, podle obsahu a jistě lze nalézt další klasifikační kritéria. Tato práce si neklade za cíl popsat dopodrobna všechna tato dělení, ale zaměřit se především na dělení podle obsahu a speciálně tematické mapy, které jsou vlastním výstupem programu.

### 2.4.1 Klasifikace map podle obsahu

Podle obsahu lze mapy rozdělit na: mapy katastrální, topografické, všeobecně geografické a tematické [14].

**Katastrální mapy** slouží k vyobrazení pozemkového vlastnictví (snaží se zachytit a popsat nemovitosti a parcely). Obecně jsou to mapy velkého měřítka a v geografii jsou méně využívané, protože většinou obsahují jen polohopis. Není to však pravidlem a některé, zejména novější mapy, již obsahují i výškopis. Na území České republiky je pro katastrální mapy využíván souřadnicový systém *S-JTSK*.

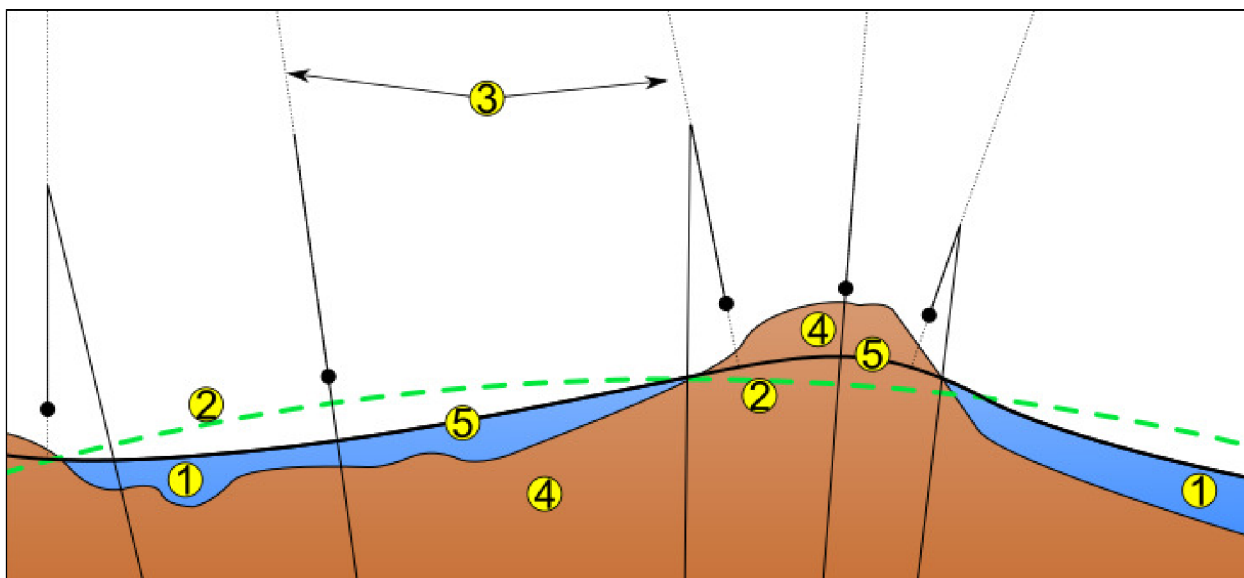
**Topografické mapy** jsou v geografii narozdíl od katastrálních map hojně využívány pro svůj bohatý obsah. Snaží se zachytit geografickou realitu co možná nejrealističtěji, často bývají velice podrobné. Obsahují všechny základní prvky z hlediska obsahu mapy, které jsou důležité pro orientaci: výškopis, polohopis a velké množství popisu. Výškopis bývá zpravidla vyobrazen pomocí vrstevnic doprovázených kótami a může být použita i barevná *hypsometrie*. Polohopis je založen na barevném podání, které se snaží co nejvíce odpovídat skutečnosti (vodní toky mají modrou barvu, lesy zelenou...). Příkladem této klasifikace map mohou být turistické mapy.

**Všeobecně geografické mapy** dosahují vysoké míry *generalizace*, tedy snaží se vyobrazit a popsat základní rysy mapovaného území bez zbytečných detailních informací. Zobrazují pouze nejdůležitější geografické objekty, charakteristiky zemského povrchu, vodstvo, sídla atd. Jsou to mapy malých měřítek zachycující celé státy, kontinenty či dokonce celý svět.

**Tematické mapy** [12] jsou mapy, které vznikly na podkladě všeobecně geografických nebo topografických map. Svým významem patří k velice důležitým mapám v geografické praxi. Neboť se tematické mapy zaměřují pouze na určitou problematiku (politická situace, chemický průmysl, potravinářský průmysl, počasí, zemědělství...), tak dochází k tzv. *kartografické generalizaci*. Jedná se o jisté zevšeobecnění a potlačení prvků, které s daným tématem nesouvisí a není je tedy potřeba na mapě zobrazovat. Pro tematickou mapu prodejů přípravků nepotřebujeme zobrazovat hornatost, vodstvo, lesy atd.

## 2.5 Souřadnicové systémy

Každý jistě má alespoň přibližnou představu o tvaru povrchu Země. Povrch Země je značně členitý, obsahuje různá vyvýšená místa (kopce, pahoraktiny, pohoří...) či naopak místa se zápornou nadmořskou výškou – prolákliny. Je tedy značně obtížné převést povrch Země do roviny. Proto je nutné pro použití souřadnicového systému povrch zjednodušit. Základním zjednodušením zemského povrchu je tzv. *geoid* (obrázek 2.1). Geoid je zjednodušená plocha zemského povrchu, která má v každém jejím bodě stejnou hodnotu zemské tíže. Tato plocha přibližně splývá se střední hladinou moří. Zde je důležité upozornit, že střední hladinou se rozumí výškový rozdíl mezi přílivem a odlivem a tato hodnota se také může na různých místech naší planety lišit. Povrch geoidu může dokonce i ležet pod skutečným povrchem Země (tato situace nastává především u různých vyvýšenin).



Obrázek 2.1: Aproximace povrchu Země (autor: Messer Volland) [11], popis: 1 - oceán, 2 - elipsoid, 3 - lokální kolmice, 4 - kontinent, 5 - geoid.

Jak si lze všimnout na obrázku 2.1, tak stále nelze tuto formu zjednodušení zemského povrchu snadno popsat matematicky. Je tedy nutné opět tuto aproximaci zemského povrchu zjednodušit. Dostáváme výsledné zjednodušení, které je nazývávané *rotační elipsoid* (na obrázku 2.1 je označen čárkovanou zelenou čarou). Jedná se o matematicky definované těleso snažící se co nejvíce přimyká geoidu. V geografii je rotační elipsoid nejčastěji nazýván tzv. *referenčním elipsoidem*. Tyto elipsoidy vznikají obvykle pro určitá menší území (národní elipsoidy).

**Besselův**, který je použit pro Českou republiku a Evropu. **Krasovského elipsoid** používaný na území Ruska. **NAD83** pokrývající Severní Ameriku a v neposlední řadě **WGS-84** - světový geodetický systém. Existuje ovšem i množství dalších elipsoidů [5].

### 2.5.1 Typy souřadnicových systémů

Souřadnicové systémy se dělí na globální (WGS 84, UTM) a lokální (S-JTSK pro ČR). Jejich vznik bývá velice často úzce spjatý s armádními účely [13].

#### 2.5.1.1 WGS-84 (World Geodetic System)

Jedná se o celosvětově uznávaný souřadný systém, který je někdy také nazýván systémem "šířka-délka". Používá tzv. *univerzální transversální Mercatorovo zobrazení* (UTM). Tento souřadný systém je *geocentrický*, protože jeho počátek leží ve středu Země. Používá stejnojmenný referenční elipsoid WGS-84. Číslo 84 značí rok 1984, kdy byl tento standard vydán ministerstvem obrany USA. Osy X a Y tohoto systému leží v rovině rovníku. Tento systém je využíván globálním polohovacím systémem (GPS).

#### 2.5.1.2 S-JTSK

Souřadnicový systém jednotné trigonometrické sítě katastrální (S-JTSK) je systém používaný na území České republiky. Je definován Besselovým referenčním elipsoidem za pomoci

tzv. *Křovákova zobrazení*. To bylo navrženo v roce 1922 Ing. Josefem Křovákem. Celá republika leží v prvním kvadrantu a tedy obě souřadnice X a Y nabývají pouze kladných hodnot. Kladný směr osy X směřuje k jihu a osa Y je na ni kolmá a směřuje na západ. Navíc platí, že hodnota souřadnice Y je vždy menší než hodnota souřadnice X.

### 2.5.1.3 UTM

Univerzální transversální Mercatorův systém souřadnic funguje na principu mřížky. Celá Země je rozdělena do 60 poledníkových zón (jsou číslovány od 1 do 60 směrem na východ). Každá tato zóna je dále rozdělena do 20 rovnoběžkových zón (značených písmeny abecedy směrem od jihu k severu). Tímto dělením nám vznikne pomyslná mřížka. Výhodou je, že v rámci jedné zóny lze pro výpočet vzdálenosti použít *Pythagorovu větu*. Stejně jako systém WGS-84 je i tento systém používán v GPS.

## 2.6 Geografické informační systémy (GIS)

Pohled na geografické informační systémy [10] se různí a nelze nálezt přesnou definici tohoto označení. Proto prohledáme-li literaturu zabývající se tímto tématem, tak nalezneme velké množství definic GIS. Z novějších definic lze uvést informaticky orientovanou definici:

„Streit, 1997: GIS je na počítačích založený informační systém na získávání, obhospodařování, analýzu, modelování a vizualizaci geoinformací. Geo data, která využívá, popisují geometrii, topologii, tematiku (atributy) a dynamiku (změny v čase) geoobjektů.“ [10]

Souhrně je GIS nějaký počítačový systém, který slouží pro ukládání, udržování a využívání údajů spojených s popisem míst na zemském povrchu.

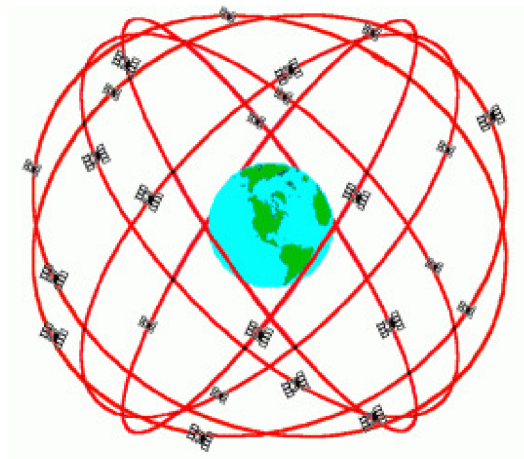
Existují i různé názory na funkčnost GIS. Pohled jako na prostředek zabývající se mapami, databázový pohled a analytický pohled. Tyto pohledy se liší především podle potřeb a zaměření uživatelů. U osob s informatickým vzděláním převládá databázový pohled.

Historie počítačových GIS sahá do 60. let minulého století. GIS jsou výsledkem paralelního vývoje v mnoha oborech pracujících s prostorovými informacemi a na jejich vzniku se podílelo velké množství vědních disciplín a lidských činností. Z oborů pracujících s prostorovými informacemi lze zmínit např. CAD projektování a počítačovou grafiku, kartografii a kvalitní kreslení, technologii dálkového průzkumu Země (DPZ), mapování a fotogrammetrii... Jako příklad vědních disciplín formujících GIS se může zmínit geografie, projektování, kartografie, prostorová matematika atd.

## 2.7 Globální polohovací systém (GPS) a Galileo

Původní název systému je Navstar GPS (*Navigation Signal Timing and Ranging Global Positioning System*). Je to vojenský navigační systém provozovaný Ministerstvem obrany Spojených států amerických, který vzniknul roku 1973. Slouží pro určování relativně přesné polohy a času nejen na Zemi, ale i ve vzduchu.

Tento systém je tvořen 24 družicemi (obrázek 2.2) obíhajícími kolem planety Země ve výšce cca 20 000 km. Signál je vysílán na takových frekvencích, aby byl co nejméně ovlivněn meteorologickými vlivy. Vlastní použití lze rozdělit do dvou kategorií: na vojenský a civilní sektor. Vojenské přijímače dokážou naproti civilním přijímat i dekodovat tzv. P(Y) kód



Obrázek 2.2: Rozmístění a dráhy 24 družic GPS, autor: Peter H. Dana, Ph.D.

a díky tomu dosahovat vyšší přesnosti. Tato vysoká přesnost je pro civilní sektor nežádoucí z hlediska možného zneužití pro teroristické útoky.

Projekt Galileo je plánovaný autonomní evropský družicový navigační systém. Je obdobou amerického GPS. Má být tvořen 30 družicemi (27 operačních + 3 záložní) obíhajícími planetu Zemi ve výšce asi 23 500 km nad zemským povrchem [7, 8].

## Kapitola 3

# Projekt OpenStreetMap

*OpenStreetMap* (*OSM*) [3] je projekt, který si klade za cíl vytvořit volně editovatelnou mapu celého světa. Založil jej v červenci 2004 Steve Coast z Velké Británie, který byl inspirován úspěchem Wikipedie. Vlastní geodata jsou sbírána za pomoci GPS přijímačů, z dostupných licenčně kompatibilních map a satelitních snímků.

Podobně jako známá internetová encyklopedie Wikipedie je i tento projekt založen na dobrovolné činnosti uživatelů. Umožňuje tedy velice snadnou editaci dat, uchovává historii změn a veškerá jeho data jsou volně přístupná široké veřejnosti. V současnosti komunita čítá přes 1 milion registrovaných uživatelů a jejich počet od roku 2006 exponenciálně roste. Pro zapojení do projektu se stačí pouze registrovat.

Data jsou dostupná pod licencí *Open Database Licence* (*ODbL*) a celá stránka je podporována neziskovou organizací *OpenStreetMap Foundation* registrovanou ve Velké Británii.

### 3.1 Formát dat

Formát uložení vektorových dat je odvozen od serializačního formátu XML. Použitým souřadnicovým systémem je globální systém WGS-84 (viz sekce 2.5.1.1) a projekce odvozená od Mercatorovy projekce (viz sekce 2.2.1).

OpenStreetMap využívá topologickou datovou strukturu se čtyřmi klíčovými elementy (datová primitiva):

- **Uzly** (*nodes*) - Jsou to body s přesnou geografickou polohou uchovávanou dvojicí atributů *lat* (latitude - zeměpisná šířka) a *lon* (longitude - zeměpisná délka) v souřadném systému WGS-84. Mimo použití v cestách označují většinou bezrozměrné objekty map (body zájmu, zastávky autobusu, veřejné hodiny...)
- **Cesty** (*ways*) - Je to seznam 2 až 2000 uzlů reprezentující křivku či v případě stejného počátečního a koncového bodu polygon. Slouží jak k reprezentaci lineárních objektů (chodníky, silnice, řeky...), tak k reprezentaci plošných objektů (lesy, parky, jezera apod.). Oblasti s dírami nebo seznamem uzlů přesahujícím horní hranici 2000 uzlů bývají reprezentovány pomocí multipolygonálních relací.
- **Relace** (*relations*) - Jsou to víceúčelové datové struktury vyjadřující vztah mezi dvěma nebo více elementy (uzly, cestami nebo relacemi). Tyto elementy nazýváme členy relace (*relation's members*). Každý element může mít volitelný atribut *role*, určující další vlastnosti člena relace. Jako příklad lze uvést využití relací pro různé

zákazy v dopravě, při zákazu zahnutí doleva budou mít členové role „from“ a „to“ popisující odkud kam nesmíme zahnout. Vlastní popis významu relace se provádí pomocí atributů.

- **Atributy** (*tags*) - Atributy jsou tvořeny dvojicemi klíč-hodnota. Jsou to textové řetězce, které mohou obsahovat až 255 znaků. Používají se k uchování *metadat* o jednotlivých elementech (uzly, cesty, relace) či členech relací. Samotné tedy nemají význam, vždy se k něčemu vztahují. Podrobněji je tato problematika rozebrána v následující podkapitole 3.2.

Struktura, kterou je nutné pro práci s tímto formátem souboru dodržet, je následující:

- Hlavička XML souboru s typem kódování UTF-8
- osm element obsahující verzi API a název generátoru ze kterého tento soubor vzniknul (např. editační nástroj)
  - blok uzlů (nodes) obsahující lokaci v systému WGS-84
    - ▷ atributy náležející ke každému uzlu
  - blok cest (ways)
    - ▷ reference na uzly ze kterých se cesta skládá (elementy nd)
    - ▷ atributy pro každou cestu
  - blok relací (relations)
    - ▷ reference na členy pro každou relaci (elementy member)
    - ▷ atributy pro každou relaci

## 3.2 Vlastnosti map

Nedílnou součástí každé mapy je její popis (viz sekce 2.1.3). V projektu OSM je veškerý popis řešen pomocí atributů (tags), které se vážou na uzly, cesty a relace. Tento systém je nazýván *OpenStreetMap's free tagging system*<sup>1</sup> a umožňuje vkládat neomezené množství těchto atributů. U každého atributu je uveden klíč, hodnota, informace o tom na které se váže elementy, popis, vizuální podoba na mapě a ve skutečnosti.

Pro popis mnoha prvků mapy stačí pouze velmi malé množství těchto atributů. Jako příklad lze uvést chodník, který si vystačí se dvěma atributy *highway=footway* (určujícím typ silnice - chodník) a *name=\** (určujícím název).

Atributů existuje velké množství a stále nové přibývají, některé se upravují či někdy je vhodné nepoužívané odebrat. Dělí se do několika základních kategorií: primární vlastnosti (budovy, místa, silnice...), odkazy (veřejná doprava, obchody, turismus...) a dodatečné vlastnosti (adresy, jména, omezení...).

## 3.3 OpenStreetMap API

Rozhraní pro programování aplikací projektu OSM slouží pro stahování a editaci dat. Je založeno na myšlenkách *RESTful API*<sup>2</sup>. Využívá architektury server-klient a požadavky

<sup>1</sup>[http://wiki.openstreetmap.org/wiki/Map\\_Features](http://wiki.openstreetmap.org/wiki/Map_Features)

<sup>2</sup>REST (Representational State Transfer) je architektura rozhraní navržená pro distribuované prostředí (vyvinuta souběžně s HTTP/1.1 protokolem, autorem je Roy Fielding)



jsou ve formě protokolu HTTP GET (získat), PUT (aktualizovat), POST (vytvořit) a DELETE (smazat). Odpovědi od serveru jsou ve formátu MIME typu "text/xml" s použitým UTF-8 kódováním. Požadavky na editování databáze používají základní HTTP autentizaci či doporučený OSM mechanismus *OAuth*. V současné době je nejnovější OSM API verze 0.6. Limitujícím faktorem tohoto API je maximální počet 50 000 prvků pro danou oblast a pro stažení většího množství dat je nutné použít *OSM Extended API* či externí zdroje dat.

**OSM Extended Api** (OSM XAPI) je postaveno na základním OSM API a poskytuje pokročilé dotazy a vyhledávání. Narozdíl od původní API slouží pouze pro čtení. Používá opět rozhraní *REST* a navíc přidává podporu pro *X-path* výrazy. Maximální limit elementů pro danou oblast je zde stanoven na 10 000 000. Existují i různé další implementace myšlenky XAPI (jXAPI, Overpass API, Xappy.js).

## 3.4 Dostupné vykreslovací nástroje a programy

Vykreslovací nástroje slouží pro převod vektorových OSM dat do podoby, která je srozumitelná uživateli - výsledné vykreslené mapy. Většina těchto aplikací využívá tzv. stylovací soubory (*stylesheets*) pomocí kterých lze definovat požadovaný vzhled mapy. Těchto nástrojů existuje velké množství a liší se mnoha faktory (jednoduchost, multiplatformnost, vzhled atd.). Mezi dva populární vykreslovací nástroje ve dvourozměrném prostoru patří *Mapnik* (viz sekce 3.4.1) a *Osmarender* (viz sekce 3.4.2).

### 3.4.1 Mapnik

Mapnik [9] je open source projekt pro vykreslování (*rendering*) map. Je implementován v programovacím jazyce C++ a je multiplatformní. Pro vykreslování map používá AGG vykreslovací knihovnu (*AGG rendering library*), která umožňuje použití technologie *anti-aliasing*. Vstupními formáty mohou být např. ESRI shapefile, PostGis, TIFF rastry, soubory osm, CSV soubory, GDAL... Nabízí také rozmanité výstupní formáty: PNG, JPEG, SVG nebo PDF. Pro určování vzhledu výsledných map používá stylovací soubory ve formátu XML (*XML Mapnik Stylesheet*). Je v současné době použit jako hlavní vykreslovací nástroj projektu OpenStreetMap.

### 3.4.2 Osmarender

Osmarender<sup>3</sup> je na pravidlech založený vykreslovací nástroj pro generování obrázků ve vektorovém formátu SVG z OSM dat. Jedná se tedy o převod vektorového souboru OSM XML na SVG formát pomocí XSLT transformace. Vlastním vstupem tohoto programu je XML soubor obsahující veškerá pravidla a stylovací XSL soubor popisující XSLT transformaci. Pro zpracování transformace je nutné použít nějaký dostupný *XSLT procesor* (Xalan-J, XRAY2, Saxon, XML Starlet, MSXML, NXSLT2), který provede XSLT transformaci do výsledného SVG formátu. Pokud chceme výslednou mapu v rastrovém formátu, tak je nutné použít nějaký externí program na editaci obrázku pro export do požadovaného formátu.

Až do března 2012 byl Osmarender používán jako jeden z hlavních vykreslovacích nástrojů pro projekt OpenStreetMap. V současné době je jeho vývoj je pozastaven.

---

<sup>3</sup><http://wiki.openstreetmap.org/wiki/Osmarender>

### 3.4.3 TileMill

TileMill<sup>4</sup> je prostředí pro tvorbu map využívající pro vykreslování výše zmíněný Mapnik. Stylovací soubory tohoto nástroje jsou ve formátu *CartoCSS*<sup>5</sup>. Na vstupu si poradí se soubory ve formátu *shapefile* či se dokáže připojit na *PostGIS* databázi pro vykreslování. V základním nastavení vykresluje do vlastního formátu *MBTiles*, ale výstupem může být i formát PNG či PDF.

### 3.4.4 Maperitive

Maperitive<sup>6</sup> je napsán v jazyce C# a je multiplatformní (Windows, Linux, Mac). Je to prostředí, které je zaměřené na jednoduchost na úkor výkonnosti. Není jej tedy vhodné použít pro zpracování velkého množství dat. Vhodným výstupem může být např. mapa města. Přesto nám nabízí zajímavé prvky funkcionality: stínování kopců (*hillshading*), skripty pro automatizované generování map, vytváření mapových dlaždic (*tiles*)... Program je stále ve vývoji a v současné době se ovládá přes konzolové rozhraní.

### 3.4.5 OSM-3D

*OSM-3D.org*<sup>7</sup> je projekt spravovaný vědeckou skupinou *GIscience* na univerzitě v Heidelbergu. Je vyvíjen jako ukázkový příklad potenciálu prostorových map. Jde především o prostorové zobrazení budov a terénu. Vzhledem k jednoduchosti modelů budov nelze očekávat dokonalé realitě odpovídající modely. Příkladem může být proporcčně složitá Eiffelova věž v Paříži, která je zobrazena formou jednoduchého mnohostěnu. Projekt pracuje s rozsáhlou databází, která čítá přes 1 TB dat.

Jedná se o velice zajímavý projekt udávající směr, kterým by se měly ubírat vykreslovací nástroje v budoucnu.

---

<sup>4</sup><https://www.mapbox.com/tilemill/>

<sup>5</sup>CartoCSS je preprocesor pro stylovací soubory používané Mapnikem vyrobený firmou MapBox

<sup>6</sup><http://maperitive.net>

<sup>7</sup><http://www.osm-3d.org/>

## Kapitola 4

# Aplikace MyTracks

*MyTracks*<sup>1</sup> je aplikací od firmy Google napsaná v jazyce Java pro operační systém Android. Účelem této aplikace je zaznamenávat pomocí GPS zařízení souřadnice průběhu nějaké trasy. Lze ji tedy použít například pro zaznamenání trasy při turistice, cykloturistice, běhání, v automobilu, při zimních sportech...

Aplikace je schopna na základě získaných dat z GPS přijímače určit u trasy cestu, rychlost, vzdálenost či nadmořskou výšku. Dále je schopna tato data zobrazovat v reálném čase a dokáže se synchronizovat s různými biometrickými senzory (např. pro zaznamenání srdeční činnosti v průběhu aktivity). Výsledné trasy lze exportovat do následujících formátů: GPX, KML, CSV nebo TCX.

### 4.1 Keyhole Markup Language

*Keyhole Markup Language*<sup>2</sup> (KML) je aplikací metajazyka XML a slouží pro uchování geodat. Původně byl vytvořen firmou Keyhole, Inc., kterou později koupila firma Google. V roce 2008 byl mezinárodně standardizován jako *OpenGIS® KML Encoding Standard* (OGC KML) standardizační organizací *Open Geospatial Consortium*.

Tento formát nachází největší využití v produktech firmy Google, kterými jsou mapové aplikace Google Earth, Google Maps a Google Mobile. Vzhledem k tomu, že byl standardizován, nalézá své uplatnění i v jiných geografických projektech.

Standard jazyka KML popisuje způsob jakým jsou geodata reprezentována (geometrie, abstraktní pohledy, popisy atd.).

### 4.2 Google Chart API

*Google Chart API*<sup>3</sup> je nástroj sloužící pro tvorbu snadno vkladatelných grafů do webových prezentací. V počátcích bylo API určeno primárně pro aplikace firmy Google. V prosinci roku 2007 bylo uvolněno pro všechny vývojáře.

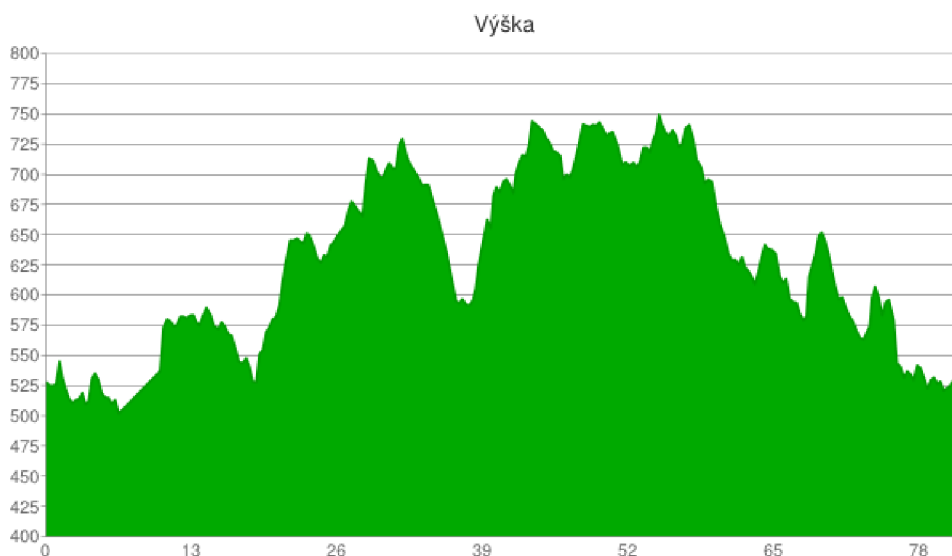
Nejdříve sloužilo pouze pro tvorbu obrázků na základě parametrů uvedených v řetězci URL (*Google Image Chart API*), nyní lze grafy vkládat do webových prezentací pomocí

---

<sup>1</sup><http://www.google.com/mobile/mytracks/>

<sup>2</sup><https://developers.google.com/kml/>

<sup>3</sup><https://developers.google.com/chart/image/>



Obrázek 4.1: Výškový profil trasy vygenerovaný na základě URL řetězce.

JavaScriptu. API nabízí velké množství typů grafů (sloupcové, výsečové, Vennovy diagramy...). Grafy jsou vykreslovány za pomoci technologie *HTML5/SVG* pro zajištění podpory ze strany různých prohlížečů a systémů.

V současné době původní systém tvorby grafů z URL stále funguje, ale má ze strany firmy ukončenou podporu. Obecný formát URL řetězce má následující formát:

```
https://chart.googleapis.com/chart?cht=<typ_grafu>&chd=<
  data_grafu>&chs=<velikost_grafu>&...dalsi_parametry...
```

Ukázkou výsledného grafu vygenerovaného na základě URL řetězce (vzhledem ke svojí délce zde není zdrojové URL uvedeno) je graf 4.1 vytvořený aplikací Google MyTracks (viz kapitola 4).

## Kapitola 5

# Návrh aplikace

Tato kapitola se zabývá vlastním návrhem aplikace. Popisuje jak má vlastní aplikace vypadat, jaké byly použité nástroje pro její vývoj či objasňuje základní principy fungování využívaného vykreslovacího nástroje.

### 5.1 Cíl tvorby aplikace

Cílem této práce bylo vytvořit multiplatformní aplikaci pro vytváření uživatelských map z dat projektu OpenStreetMap. Musí být možné v aplikaci přidávat do mapy data z jiných zdrojů než je projekt OSM (např. kilometrová mřížka, vrstevnice, vlastní trasy atd.).

Výsledný program musí být použitelný pro obvyčejného uživatele a tedy musí být do jisté míry jednoduchý na ovládání (v rámci možností) a uživatelsky přívětivý. Vlastní rozhraní aplikace je zpracováno jako grafické uživatelské rozhraní (*GUI*).

Hlavní okno aplikace je rozvrženo tak, že má vzhled velice podobný různým grafickým editorům. V pravé části tohoto okna nalezneme náhled právě vytvářené mapy, který proporčně zaujímá největší část okna aplikace. Tento náhled je průběžně překreslován na základě voleb uživatele. V levé části lze nalézt panel se správou aktuálního stylovacího souboru. V horní části okna se nachází panel nástrojů s často používanými tlačítky.

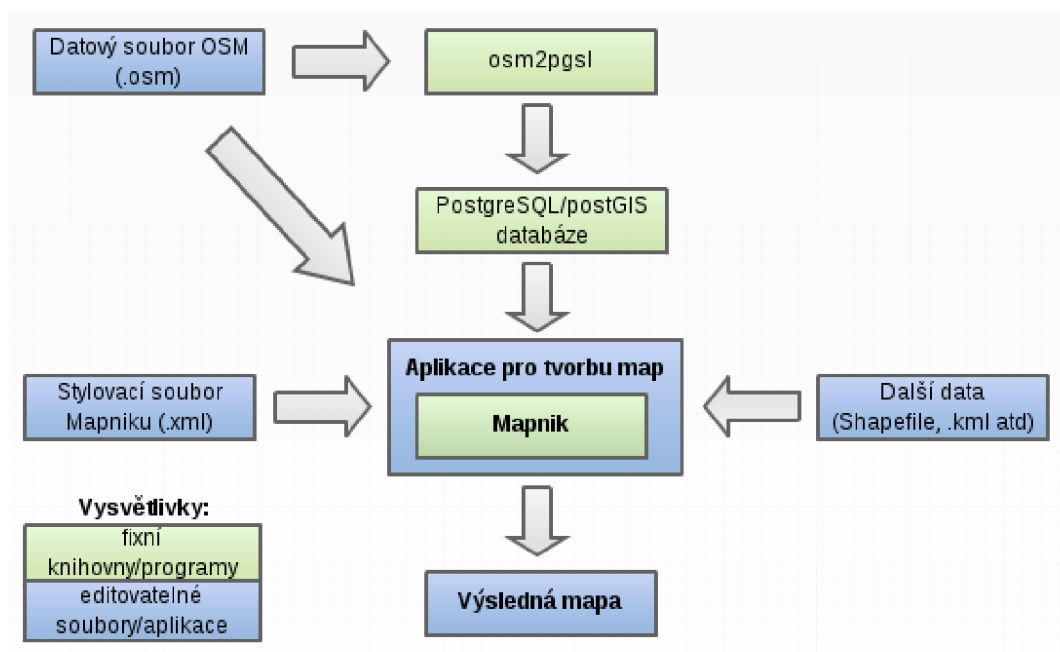
Další funkcionalitou je import vlastních tras. Aplikace podporuje import vlastních tras pomocí souborů ve formátu KML (viz podkapitola 4.1), které jsou výstupem exportu z aplikace MyTracks (viz kapitola 4). Výsledné mapy mohou obsahovat graf výškového profilu importované trasy získaný za pomoci *Google Chart API* (viz podkapitola 4.2).

Vlastním výstupem programu jsou exportované uživatelské mapy, které jsou ve formátu připraveném pro tisk. Jak bude vypadat výsledná mapa záleží čistě na preferencích uživatele a požadovaného vzhledu lze dosáhnout pomocí rozmanitých nastavení v aplikaci. Výsledná aplikace umožňuje zobrazení náhledu finální mapy před jejím exportem do souboru.

### 5.2 Mapnik

Neboť cílem této práce je vytvořit prostředí pro tvorbu map, tak je nezbytně nutné mít nějaký vykreslovací (*renderovací*) nástroj. Možnosti jsou prakticky pouze dvě. První možností je vytvořit si vlastní vykreslovací nástroj a druhou je použití již vytvořeného. Vzhledem k rozsahu práce, která je spojena s vytvořením vlastního vykreslovacího nástroje, je nutné použít již nějakou hotovou a odladěnou knihovnu.

Na výběr bylo poměrně velké množství různých nástrojů. Vzhledem ke skutečnosti, že



Obrázek 5.1: Schéma zobrazující koncept výsledné aplikace (vytvořeno v Creately).

jej potřebujeme použít uvnitř vlastního programu, tak se výběr zúžil. Nešlo tedy použít dříve zmíněný TileMill (viz sekce 3.4.3) či Maperitive (viz sekce 3.4.4), které jsou vlastními programy s využitím nějakého stávajícího či ryze vlastního vykreslovacího nástroje. Nabízelo se zvolit například Osmarender (viz sekce 3.4.2), ale vzhledem k ukončené podpoře by se nejednalo o zrovna nejvhodnější volbu.

Po porovnání různých nástrojů a knihoven byla zvolena knihovna Mapnik [9], která je v současné době i jako hlavní vykreslovací nástroj projektu OpenStreetMap. Mimo jiné je schopna pracovat i s velkým množstvím vstupních formátů a je napsána v jazyce C++, tedy ve stejném programovacím jazyce jako vyvíjené prostředí.

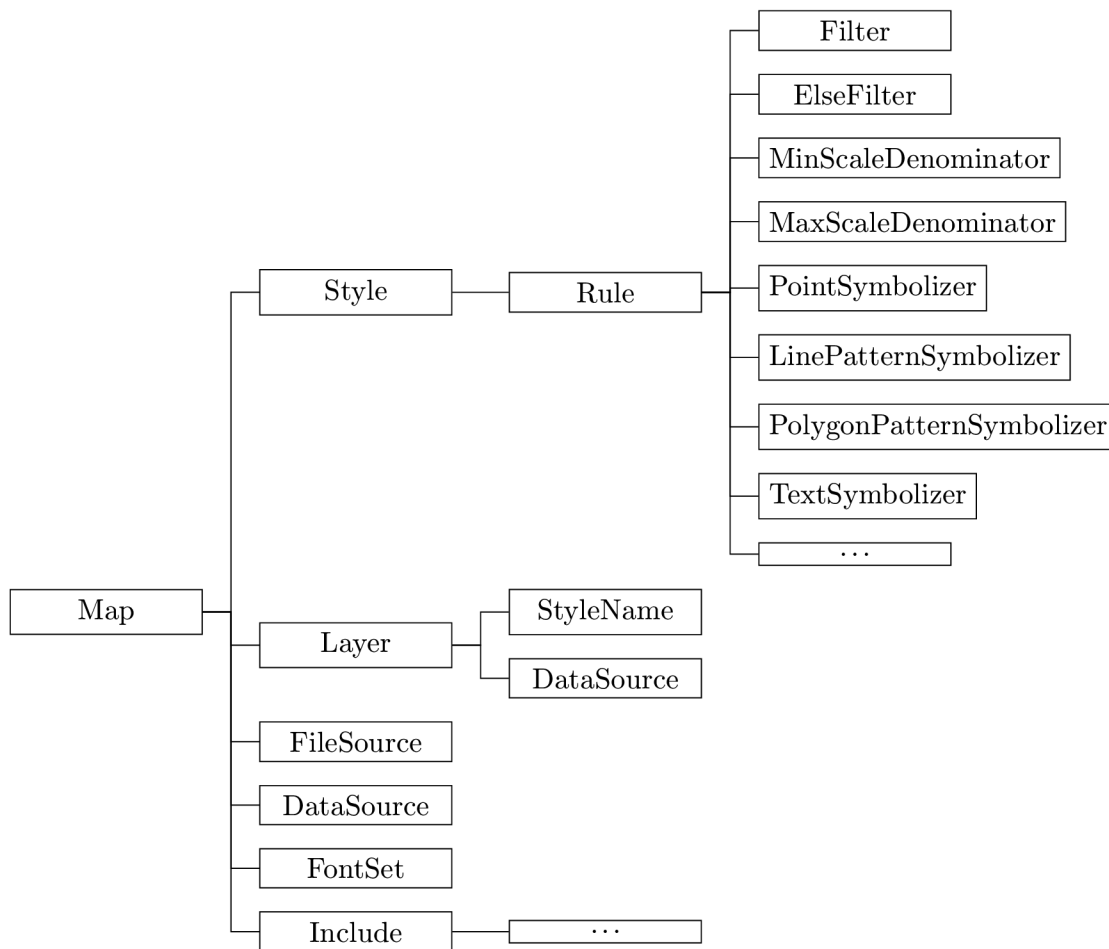
### 5.2.1 Mapnik XML Stylesheet

Objekt mapy je centrálním objektem, který poskytuje metody pro vykreslení do výsledného souboru. Samotná mapa je tvořena vrstvami (*layers*). Každá vrstva má zdroj dat (*datasource*). Pro načtení a zpracování těchto dat je nutné použít konkrétní zásuvný modul (*plugin*).

Pro definování vizuální podoby mapy se používají styly (*styles*). Každé vrstvě může být přiřazen jeden nebo více těchto stylů. Každý styl obsahuje jedno nebo více pravidel (*rules*). Tato pravidla se skládají z volitelných filtrů (*filters*) a tzv. symbolizérů (*symbolizers*). Filtry umožňují vybrat pouze některé prvky map na které se bude dané pravidlo vztahovat a symbolizéry určují vlastní vzhled těchto prvků.

Je důležité zvolit vhodné pořadí vrstev a pravidel, protože je zde využit tzv. malířův algoritmus (*Painter's algorithm*) a vykreslení výsledné mapy se odvíjí právě od tohoto pořadí.

Stromovou hierarchii stylovacího souboru XML lze vidět na obrázku 5.2 (jsou zobrazeny jen nejdůležitější prvky této hierarchie).



Obrázek 5.2: XML schéma stylovacího souboru Mapnik.

### 5.2.2 Architektura zásuvných modulů

Architektura zásuvných modulů (*plugin architecture*) je systém, který umožňuje snadno přidávat nové moduly do Mapniku. Každý modul má na starosti zpracování nějakého vstupního formátu souboru. V současné době je v knihovně Mapnik k dispozici asi 12 zásuvných modulů. Je důležité upozornit, že ne všechny moduly jsou stabilní a některé jsou teprve v rané fázi vývoje. Některé moduly jsou sestavovány spolu s knihovnou Mapnik a některé je v případě potřeby nutno sestavit manuálně.

Přehled některých dostupných zásuvných modulů:

- Postgis - podpora pro přístup do postrelační databáze *PostgreSQL* s geografickou nadsatvbou *PostGIS*
- Raster - podpora pro obrázky v souborovém formátu TIFF
- Shape - modul pro parsování *ESRI Shapefile* souborového formátu
- Gdal - podpora pro souborový formát GDAL
- Ogr - podpora pro souborový formát OGR
- Osm - podpora pro vektorové mapy ve formátu OSM XML

### 5.2.3 Stínování kopců

Stínování kopců (*hillshading*)<sup>1</sup> je technika, kterou lze použít v mapách pro zdůraznění výškopisu (viz sekce 2.1.1). Výsledkem této techniky je, že mapa vypadá jako by byla plastická. Dává nám tedy jistou prostorovou iluzi. Pro aplikaci této techniky je nutné mít knihovnu GDAL.

GDAL (*Geospatial Data Abstraction Library*) [1] je knihovna určená pro práci s rastrovými GIS formáty. Používá jednoduchý abstraktní model pro práci s těmito formáty. Pomocí konzolového rozhraní lze provádět různé konverze a zpracování těchto dat. GDAL ve verzi 1.9.0 podporuje cca 120 rastrových GIS formátů.

Vlastní postup stínování je následující:

1. Stáhneme výšková data ve formátu *\*.hgt* a případně rozbalíme archiv zip.
2. Pomocí nástroje `gdal_translate` převedeme souborový formát *\*.hgt* na *\*.tiff* a zároveň se ujistíme, že je v souboru definovaná projekce.
3. V dalším kroku pomocí nástroje `gdal_warp` převedeme obrázek do projekce ve které je mapa (ve většině případů sférická Mercatorova).
4. Na závěr vytvoříme stínování za pomoci `gdal_dem`.
5. Zahrneme stínovací obrázek do Mapnik XML stylovacího souboru pro výslednou mapu.

### 5.2.4 Knihovna Qt

Pro tvorbu aplikace byla zvolena knihovna jazyka C++ Qt [2]. Jedná se o velice populární knihovnu pro vytváření multiplatformních aplikací primárně s grafickým uživatelským rozhraním (*GUI*). Podporuje nejen desktopové platformy (Windows, Linux, OS X), ale v současné době se zaměřuje i na mobilní segment (Android, iOS...).

Součástí této knihovny je několik nástrojů usnadňujících vývoj samotných aplikací. Prvním z nich je *Qt Creator*. Jedná se o vývojářské prostředí umožňující editovat zdrojové kódy v jazyce C++ či QML. Další aplikací je *Qt Designer*, který slouží pro tvorbu grafického uživatelského rozhraní. Snaží se o co možná nejjednodušší práci s těmito prvky a je možné vidět jejich výsledné rozmístění bez nutnosti spouštění výsledné aplikace. V neposlední řadě je důležité zmínit *qmake*. Tento nástroj slouží pro automatické generování souborů *Makefile* při vývoji projektu pro různé platformy.

---

<sup>1</sup>[http://wiki.openstreetmap.org/wiki/Hillshading\\_with\\_Mapnik](http://wiki.openstreetmap.org/wiki/Hillshading_with_Mapnik)



# Kapitola 6

## Implementace

Další část této práce je tvořena popisem vlastní implementace navržené aplikace v předchozí kapitole 5. V počáteční podkapitole jsou vyjmenovány a stručně popsány jednotlivé třídy. V dalších podkapitolách jsou již rozebrány konkrétní implementační problémy.

### 6.1 Použité vývojové prostředí a třídy

Celá aplikace byla naprogramována v objektově orientovaném jazyce C++ a při tvorbě bylo použito vývojové prostředí *Qt Creator* verze 3.0.3 (založené na Qt 5.2.1). Jako šablona nového projektu byla zvolena možnost *Qt Widgets Application*, protože se jedná o aplikaci s grafickým uživatelským rozhraním. Návrh dialogových oken pro zadávání dat od uživatele byl proveden pomocí vestavěného nástroje *Qt Designer* ve vývojovém prostředí.

Seznam tříd se stručným popisem:

- `MainWindow` - třída hlavního okna aplikace
- `MapWidget` - stará se o vykreslování mapy ve hlavním okně (pravá část hl. okna)
- `TreeWidget` - zobrazení použitých vrstev a stylů (levá část hl. okna)
- `Stylesheet` - uchovává data aktuálního vzhledu
- `StylesheetParser` - třída pro načtení stylovacího souboru do interní datové reprezentace programu (třída *Stylesheet*)
- `Rule` - uchovává data pravidel stylovacího souboru
- Dialogové třídy - `EditLayerDialog`, `EditStyleDialog`, `EditSymbolizerDialog`, `ExportMapDialog` a `NewMapDialog`

### 6.2 Třída `MainWindow`

Tato třída je základním stavebním kamenem celé aplikace, protože představuje hlavní okno aplikace. Jak lze vidět na obrázku v příloze A.1, tak se hlavní okno skládá z několika prvků. Nejdůležitějším z nich je *Widget*, starající se o vlastní vykreslování mapy, implementovaný třídou *MapWidget*. Druhým důležitým prvkem je *TreeWidget*, který nám umožňuje pracovat se stylovacím souborem a jehož implementace je popsána třídou *TreeWidget*. Nedílnou

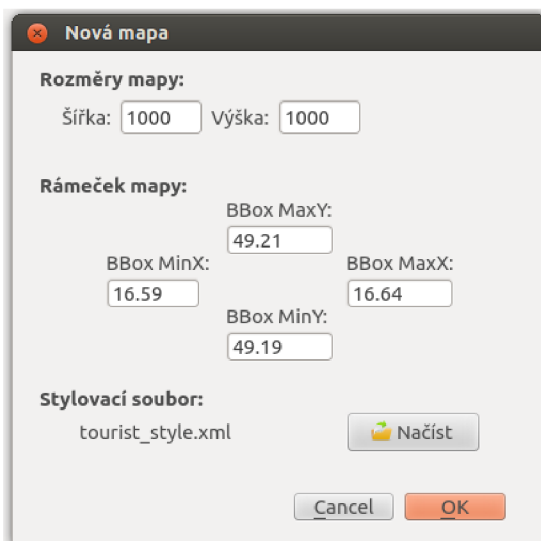
součástí hlavního okna aplikace jsou také ovládací tlačítka prvku *TreeWidget* (přidání stylu/vrstvy, změna pořadí, editace či smazání). Posledními částmi jsou stavový řádek a nástrojová lišta s tlačítky pro práci s mapou (načtení/uložení stylovacího souboru, export mapy, překreslení mapy, posuvy do všech směrů a změny velikosti).

Konstruktor této třídy je tvořen z velké části pouze funkcemi *connect*, které spojují signály s příslušnými sloty. Za zmínku stojí především použití funkcionality Qt *signalMapper*. To nám umožňuje namapovat dva signály podobných tlačítek na jeden slot přebírající parametrem referenci na *QString*. Podle stisknutého tlačítka je potom automaticky předán požadovaný parametr.

### 6.2.1 Načtení stylovacího souboru s mapou

Práce s programem je vždy zahájena načtením stylovacího souboru. Tento soubor musí být umístěn ve složce *data*, která se nachází na stejné úrovni jako je spouštěcí soubor. Není-li tato složka k dispozici, tak je automaticky vytvořena při spuštění programu. Stylovací soubor obsahuje název souboru s mapou se kterou se bude pracovat. Jedná se o předem stažený soubor s vektorovými daty popisujícími danou oblast. Tento soubor má příponu *\*.osm* a rovněž musí být ve složce *data*.

Stylovací soubor lze vybrat v dialogovém okně, které je implementováno třídou *NewMapDialog*.



Obrázek 6.1: Ukázka dialogu načtení mapy.

Jak lze vidět na obrázku 6.1, tak při vytváření nové mapy je nutno zvolit její šířku a výšku udávanou v pixelech. Dále je nutno zvolit oblast, kterou chceme vykreslit. Tu zadáme pomocí tzv. ohraničovacího rámečku (*Bounding Box*), který nám v souřadnicích ohraničuje vykreslovanou oblast a je udáván ve stupních. Vlastní stylovací soubor vybereme pomocí tlačítka načíst. Jeho výběr je implementován pomocí Qt třídy *QFileDialog* a její metody *getOpenFileName*.

Po potvrzení dialogového okna dojde k načtení mapy do interní reprezentace dat knihovny Mapnik funkcí *mapnik::load\_map*. Dále je mapa z těchto dat vykreslena, je vytvořena pracovní kopie stylovacího souboru *tmp.xml* (Veškeré další operace jsou prováděny právě s touto kopií tak, aby nedošlo k poškození původního stylovacího souboru.) a nakonec je načtena

pomocí objektu třídy *StylesheetParser* do interní datové reprezentace programu (objekt třídy *Stylesheet*) a je naplněn položkami vrstev a stylů *TreeWidget*.

### 6.3 Třída MapWidget

Hlavní úlohou této třídy je vykreslovat mapu v pravé části hlavního okna aplikace, ale stará se i o její posuny či různé změny velikosti. O vykreslení mapy se stará funkce *renderAgg()*, která používá standardní vykreslovací nástroj Mapniku, kterým je *Anti-Grain Geometry Renderer*<sup>1</sup>. Vykreslení mapy v programu se provede následovně:

```
// vytvoření bufferu pro mapu o velikosti vykreslované mapy
image_32 buf(map->width(), map->height());

/* vytvoření objektu třídy agg_renderer,
   přijímá dva parametry - mapu a buffer */
agg_renderer<image_32> ren(*map, buf);

// vykreslení mapy
ren.apply();

// konverze do formátu se kterým pracujeme v Qt
QImage img((uchar*)buf.raw_data(), map->width(),
           map->height(), QImage::Format_ARGB32);
```

Nutno dodat, že tato ukázka kódu je zjednodušená, protože při volání funkce *apply()* může dojít k výjimkám, které je nutno ošetřit.

### 6.4 Třída TreeWidget

Tato třída je odvozena od Qt třídy *QTreeWidget* a pomocí stromové hierarchie nám zobrazí vrstvy a styly načteného stylovacího souboru. Kořenovými prvky jsou vrstvy a jejich potomky jsou styly náležící dané vrstvě. Jednotlivými položkami jsou zaškrťovací políčka (*checkbox*). Jejich zaškrtnutím jsou vybrány vrstvy a styly, které chceme v mapě vykreslit. Výsledný vzhled lze pomocí příslušného tlačítka uložit. O uložení do souboru se stará funkce *writeStylesheetToFile*.

### 6.5 Třída Stylesheet

Účelem této třídy je uchovávat načtená data ze stylovacího souboru v interní datové reprezentaci programu. Obsahuje tři privátní proměnné:

```
QHash<QString, QString> mapAttributes;
QList<TStyle> styles;
QList<TLayer> layers;
```

---

<sup>1</sup>[http://en.wikipedia.org/wiki/Anti-Grain\\_Geometry](http://en.wikipedia.org/wiki/Anti-Grain_Geometry)

První proměnná uchovává atributy a jejich hodnoty elementu *Map* ve stylovacím souboru. Jak lze vidět z deklarace, tak klíč i hodnota jsou datového typu *QString*. Styly a vrstvy jsou uchovávány pomocí dvou uspořádaných seznamů. Datovými typy těchto seznamů jsou struktury, které mají svou deklaraci v hlavičkovém souboru. Jelikož se jedná o uspořádané seznamy, které udávají v jakém pořadí se budou vykreslovat jednotlivé prvky mapy, tak je nutné mít funkce pro změnu pořadí prvků seznamu. Dalšími funkcemi, které lze nalézt v této třídě jsou především funkce pro získání či smazání stylu nebo vrstvy požadovaného jména.

## 6.6 Třída StylesheetParser

Neboť v poslední stabilní verzi 2.2.0 knihovna Mapnik (viz podkapitola 3.4.1) neumožňuje modifikovat data uložená v její interní datové reprezentaci, tak bylo nutné tento problém vyřešit tvorbou vlastní třídy. Tato třída slouží jak pro načítání dat ze stylovacího souboru *\*.xml* do vlastní datové reprezentace, tak pro ukládání dat z vlastní reprezentace do stylovacího souboru. S takto načtenými daty již lze pracovat pomocí funkcí této třídy v požadovaném rozsahu.

Existuje několik možností jak lze za pomoci knihovny Qt pracovat s XML souborem:

1. Využít třídu knihovny Qt *QXmlStreamReader*.
2. Využít třídu knihovny Qt *QDomDocument*.
3. Načíst celý soubor ručně pomocí vlastní implementace.

Z těchto tří možností byla pro implementaci zvolena možnost číslo dvě. Třetí možnost byla vyloučena jako první, protože vlastní implementace je zbytečně zdlouhavá a odváděla by od vytyčeného cíle. Ze zbývajících možností byla zvolena možnost s využitím knihovny Qt *QDomDocument*, která má sice oproti první větší paměťové nároky, ale za cenu lepšího přístupu k načteným elementům ve stromové struktuře.

Při načítání či ukládání do *\*.xml* souboru narazíme na mnoho opakujících se algoritmických problémů, jako je např. zpracování atributů elementu a jejich hodnot. Pro každý tento problém byla implementována příslušná funkce. Jako příklad lze uvést: *parseAttributes*, *parseRules*, *writeStyles*, *writeSymbolizer*, *writeDataSourceToLayer*...

## 6.7 Třída Rule

Tato třída slouží pro uchovávání dat veškerých pravidel. Symbolizéry jsou uchovány pomocí seznamu objektů třídy Qt *QHash*, jehož deklarace vypadá následovně:

```
QList<QHash<QString, QString> > symbolizerList;
```

Jednotlivé symbolizéry v tomto seznamu lze rozlišit podle speciálního haše s klíčem „#“ pomocí kterého přistoupíme k hodnotě uchovávanému názvu symbolizéru. Toto chování je implementováno funkcí *getSymbolizerByName* přijímající jako parametr název symbolizéru, který je datového typu *QString*.

V této třídě se nachází i kopírovací konstruktor, který je využit v dialogové třídě *EditStyleDialog* (pravidla musí být zkopírována, aby se dala obnovit při nepotvrzení změn v dialogovém okně).

## 6.8 Třída `EditLayerDialog`

Jedná se o dialogovou třídu, které je odvozena od Qt třídy `QDialog`. Dialog slouží k úpravě a zadávání důležitých atributů vrstvy. Jako příklad lze uvést atributy: `name`, `status`, `srs`, `clear-label-cache`, `minzoom`...

Nedílnou součástí tohoto dialogového okna je i editor stylů, který je implementován pomocí Qt třídy `QTreeWidget`. Ten nám umožňuje přidávat, odebírat či měnit pořadí stylů příslušících dané vrstvě. Pořadí stylů ve vrstvě určuje vzhled výsledné mapy. Čím je styl v tomto seznamu níže, tím později je vykreslen.

Ve spodní části dialogového okna se nachází editor elementu `Datasource`. Tento editor nám primárně umožňuje zvolit zdroj dat (`osm`, `ogr`, `shape`, `postgis`, `raster` a `gdal`). Na základě zvoleného zdroje dat se aktualizuje nabídka rozbalovacího seznamu atributů. Atributy lze přidávat či odebírat a měnit u nich hodnoty.

## 6.9 Třída `EditStyleDialog`

Tato třída má stejnou bazovou třídu jako třída `EditLayerDialog`. V levé horní části tohoto dialogového okna se nachází textové pole pro editaci názvu stylu. Pod ním je umístěn `QTreeWidget`, který slouží pro správu pravidel. Umožňuje vytvářet nová pravidla, odebírat pravidla či měnit jejich pořadí. V pravé části se nachází výčet všech možných elementů pravidla (`Filter`, `ElseFilter`, `MinScaleDenominator`, `PointSymbolizer`, `LinePatternSymbolizer`, `TextSymbolizer` a další symbolizéry). Pokud to velikost dialogového okna dovolila, tak lze příslušné atributy symbolizérů editovat přímo v tomto okně. U symbolizérů `TextSymbolizer`, `ShieldSymbolizer` a `MarkersSymbolizer` je nutno kliknout na tlačítko „Nastavit“, které otevře další modální dialogové okno třídy `EditSymbolizerDialog` pro nastavení atributů.

Ukázku tohoto dialogového okna s vybraným pravidlem lze vidět na obrázku 6.3. Filtr (`[landuse]='quarry'`) omezuje množinu osm prvků na které bude pravidlo použito. `MaxScaleDenominator` s hodnotou 400000 určuje horní hranici měřítka do které bude pravidlo aplikováno. `PolygonPatternSymbolizer` určuje obrázek, kterým bude polygon dlaždicově vyplněn. A poslední nejmenovaný element je `LineStyleSymbolizer` se zadanou barvou pomocí atributu `Stroke` a tloušťkou čáry, ohraničující polygon, nastavenou parametrem `Stroke-width`.

## 6.10 Třída `EditSymbolizerDialog`

Smyslem této třídy je umožnit přidávání, odebírání a editaci atributů u symbolizérů, které jich obsahují velké množství. Atributy pro přidání lze vybrat z rozbalovací nabídky.

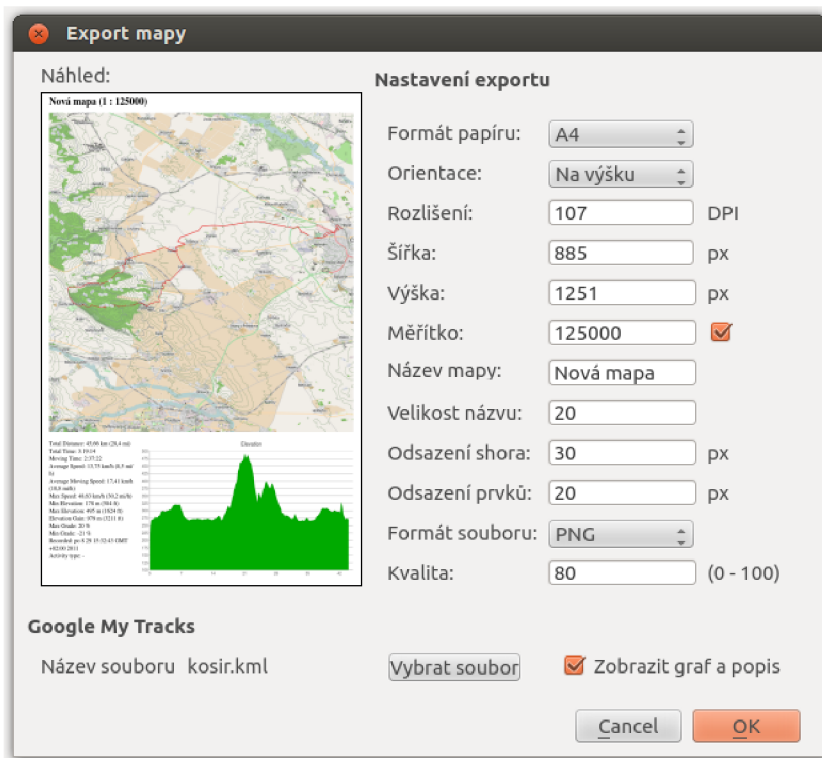
## 6.11 Třída `ExportMapDialog`

Jak lze vidět na obrázku 6.2, tak je okno této dialogové třídy pomyslně rozděleno na tři části. Levá část obsahuje zmenšený náhled výsledné mapy, který je průběžně překreslován na základě hodnot zadaných uživatelem v pravé části. Jedním ze základních nastavení je volba formátu papíru. Lze zvolit orientaci buď na šířku (vhodné zejména pro mapy bez doplňujících dat z externích zdrojů, protože obsáhnou větší část papíru.) nebo na výšku. Velikost výsledné mapy lze ovlivnit: zadáním rozměrů, rozlišení (DPI) či změnou měřítka. Tyto hodnoty spolu úzce souvisí a zbývající se vždy dopočítávají v závislosti na ostatních. Dále lze zadat název, jeho velikost a odsazení od horního okraje papíru. Dalším nastavením

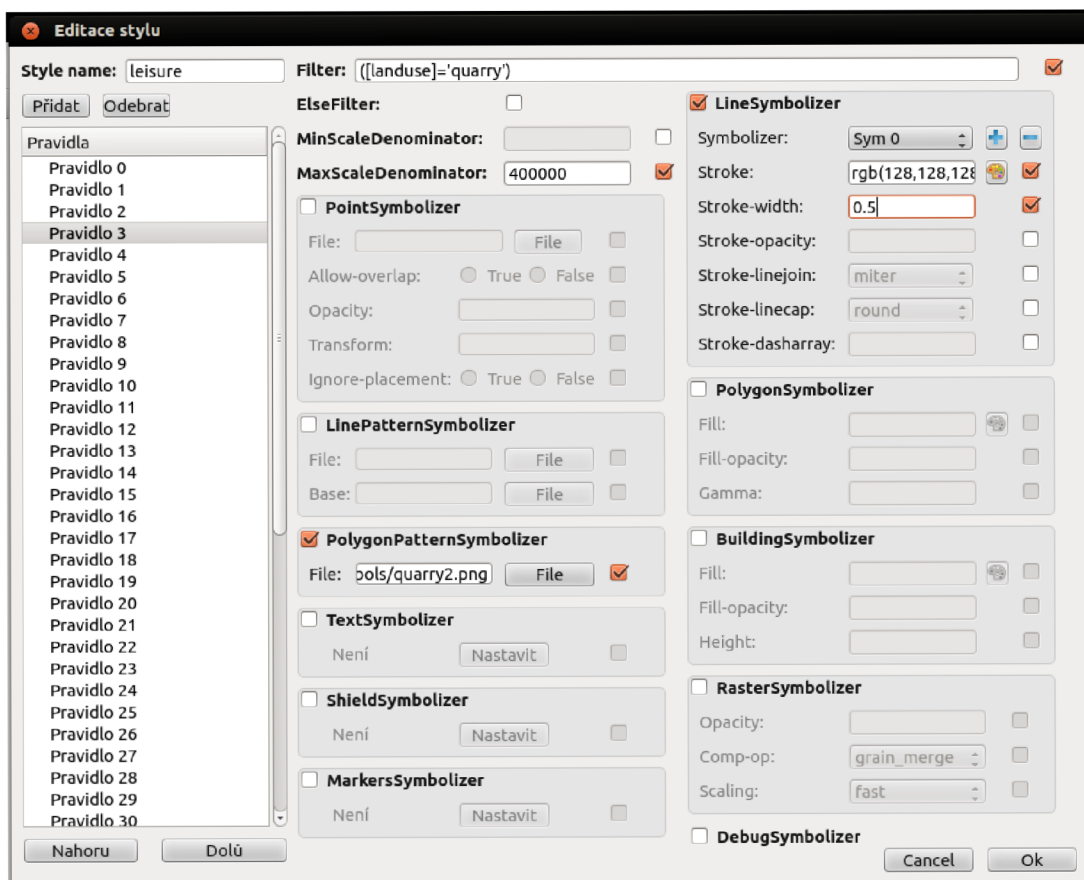
je odsazení prvků, které určuje mezery od okraje papíru a mezi jednotlivými prvky. Jako formát souboru lze zvolit *\*.bmp*, *\*.jpg* či *\*.png* s udávanou kvalitou exportu na stupnici od 0 do 100 (-1 pro výchozí nastavení). Nejnižší číslo je vhodné zvolit pro malé komprimované soubory a nejvyšší analogicky opačně. Poslední volbou je výběr souboru *\*.kml* získaný z aplikace *Google My Tracks* (viz kapitola 4). Zpracováním a vykreslením těchto dat se zabývá následující podkapitola.

### 6.11.1 Načtení a zpracování externích dat

O vlastní načtení a zpracování *\*.kml* souboru se stará funkce *parseKML*, která přijímá jako parametr cestu k souboru. Pomocí funkcí třídy Qt *QString* je z tohoto souboru vyextrahován popis trasy a je uložen do privátní proměnné *description*. Obdobně je získána URL grafu (viz podkapitola 4.2) a uložena do proměnné *graph*. Následně je pomocí objektu Qt třídy *QNetworkAccessManager* zavolána metoda *get*, které je parametrem předáno URL grafu. Tato metoda slouží k odeslání HTTP požadavku na server. Odpovědí na tento požadavek je uložena do nového objektu třídy *QNetworkReply* a funkce *get* na něj vrací ukazatel. Požadovaný graf profilu trasy ve formátu *\*.png* lze získat pomocí třídy *QImageReader*, která slouží pro načítání obrázků z různých zdrojů. Takto získaný popis a výškový profil trasy již lze vykreslit do náhledu exportované mapy.



Obrázek 6.2: Ukázka dialogu pro export mapy.



Obrázek 6.3: Ukázka dialogu pro editaci stylů s vybraným pravidlem pro vzhled lomu.

# Kapitola 7

## Testování

Výslednou aplikaci bylo potřeba podrobit řádnému testování, které si kladlo za cíl odhalit nejen drobné nedostatky, ale i chyby závažnějšího charakteru. Výstupem ze samotného testování jsou tři mapy, každá jiného vzhledu i zaměření. Veškeré testování probíhalo na operačním systému Linux - konkrétně Ubuntu ve verzi 12.04 LTS (Precise Pangolin).

### 7.1 Světová mapa rozložení populace

První mapa graficky znázorňuje množství obyvatel v různých státech. S větším počtem obyvatelstva je polygon tvořící hranice státu vyplněn tmavší barvou (Grónsko - světlá modrozelená, Čína - tmavá červená). Jako zdroj dat byla použita vektorová data ze serveru *Natural Earth*<sup>1</sup> a jsou ve formátu *ESRI Shapefile*<sup>2</sup>. Skládá se z několika souborů: hlavní, indexový a databázový. V hlavním jsou popsány vlastní hranice států. Indexový propojuje hlavní s databázovým, který obsahuje data od organizací (např. OSN) s doplňujícími tématickými informacemi. Neboť se jedná o mapu malého měřítka, tak zdrojová data byla stažena v měřítku 1:110 000 000 (1 cm = 1 100 km).

Ve stylovacím souboru pak definování různých barev podle populace vypadá následovně:

```
<Style name="population">
  <Rule>
    <Filter>[POP_EST] > -1 and [POP_EST] < 15000</Filter>
    <PolygonSymbolizer fill="#c7e9b4"/>
    <LineSymbolizer stroke="black" stroke-width=".1"/>
  </Rule>
  ...
  <Rule>
    <Filter>[POP_EST] >= 25650000 and [POP_EST] < 1134000000</Filter>
    <PolygonSymbolizer fill="#122F7F"/>
  </Rule>
  <Rule>
    <ElseFilter/>
    <PolygonSymbolizer fill="darkred"/>
    <LineSymbolizer stroke="black" stroke-width=".7"/>
  </Rule>
</Style>
```

<sup>1</sup><http://www.naturalearthdata.com>

<sup>2</sup>Vektorový formát pro ukládání prostorových dat v geografických informačních systémech, který je vyvinut a řízen firmou Esri.



</Rule>  
</Style>

Je definován vzhled *population* s příslušnými pravidly. V pravidlech je za pomoci elementu *Filter* vymezen populační interval (např. u prvního pravidla od -1 do 15 000). *PolygonSymbolizer* pak určuje barvu výplně státu, případně se tam vyskytuje *LineStyleSymbolizer* udávající barvu a tloušťku čáry, kterou je vykreslena hranice státu. Nespadá-li počet obyvatel do žádného intervalu, tak je aplikováno pravidlo obsahující *ElseFilter*. Tento případ nastává jen u Indie a Číny.

Formát papíru byl zvolen A4 na šířku. Rozměry mapy jsou  $1239 \times 877$  px v rozlišení 106 DPI. Měřítko u této mapy není uváděno, protože se jedná o mapu celého světa a směrem k pólům dochází k velkému zkreslení. Výsledná mapa byla exportována ve formátu *\*.png* a nastavené kvalitě 50, výsledný soubor má velikost 220,4 kB.

Mapu si lze prohlédnout v příloze B.1 či na přiloženém CD pod názvem *mapa\_1.png*.

## 7.2 Mapa detailu centra města

Druhá mapa má poměrně jednoduchý vzhled, jehož základ je tvořen silnicemi, budovami a značkami. Tento vzhled je použitelný především pro městské části a u map malých měřítek by nedošlo k plnému využití jeho potenciálu. Praktické použití by tato mapa jistě našla pro základní orientaci ve městě. Tomu napomáhají především názvy ulic a různé značky restaurací, kaváren či významných budov.

Zdrojovými daty této mapy je *\*.osm* soubor obsahující oblast s městem. Tento soubor byl předem stažen a vytvořený za pomoci exportovacího nástroje vektorových OSM dat - *Overpass API*<sup>3</sup>.

Formát této mapy je A4 s orientací na výšku. Rozměry mapy jsou  $935 \times 1321$  px v rozlišení 113 DPI. Jedná se o mapu velkého měřítka 1:25 000. Mapa byla exportována ve formátu *\*.bmp* a nastavené kvalitě 75, výsledný soubor má velikost 3,7 MB.

Výsledek exportu lze vidět v příloze B.2 nebo na CD pod názvem *mapa\_2.bmp*.

## 7.3 Mapa s turistickým vzhledem

Poslední mapa má svým vzhledem připomínat turistickou mapu. Zdrojovým souborem této mapy je opět soubor vektorových dat *\*.osm*. Před exportem byly do mapy přidány informace z externích zdrojů: vrstevnice a trasa z aplikace *Google MyTracks* (viz kapitola 4).

Data vrstevnic byla stažena z webových stránek *NASA* a bylo potřeba je převést programem *gdal\_contour* do formátu *ESRI Shapefile*, který je *Mapnikem* nativně podporován. Při převodu lze parametrem *-i* nastavit rozestupy vrstevnic. Byly vytvořeny tři soubory s rozdílnými rozestupy (po 10, 50 a 100 m). Vrstevnice je dále nutné přidat do stylovacího souboru jako nové vrstvy typu *shape* s uvedením požadovaného souboru.

Trasa je do mapy zanesená vytvořením nové vrstvy se zdrojem dat typu *gdal*, který se používá pro načítání dat z *\*.kml* souboru. Soubor s trasou bylo nutné vybrat i v exportovacím dialogu a pod mapu byl automaticky doplněn popis trasy a její výškový profil (viz podsekcce 6.11.1). Pokud je požadována mapa bez těchto doplňujících informací, tak není nutné tento soubor vybírat nebo je lze odstranit odškrtnutím zaškrtnutí políčka v dialogu exportu mapy.

---

<sup>3</sup><http://overpass-api.de/>

Formát této mapy je A4 s orientací na výšku. Rozměry mapy jsou  $901 \times 1274$  px v rozlišení 109 DPI. Jedná se o mapu velkého měřítka 1 : 120 000. Mapa byla exportována ve formátu *\*.png* a nastavené maximální kvalitě (hodnota 100), výsledný soubor má velikost 3,5 MB.

Výsledek exportu lze vidět v příloze **B.3** nebo na CD pod názvem *mapa\_3.png*.

# Kapitola 8

## Závěr

Cílem této bakalářské práce bylo implementovat program pro tvorbu map z vektorových dat projektu *OpenStreetMap*. Dalším požadavkem zadání bylo, aby aplikace obsahovala možnost měnit vzhled u výsledné mapy a přidávat do ní data z externích zdrojů. Před samotnou implementací bylo nutné nejprve nabýt teoretického základu studiem příčné literatury a seznámit se s projektem *OSM*. Aplikaci se podařilo implementovat tak, aby splňovala veškeré body zadání. Testování výsledného programu pomohlo odhalit některé nedostatky, které byly následně opraveny.

Neboť aplikace pro vzhled mapy používá externí knihovnu *Mapnik*, která definuje vzhled pomocí vlastního stylovacího souboru, tak je pro práci s aplikací nutné mít alespoň základní přehled o způsobu, jakým je mapa pomocí něj tvořena. Výsledné grafické uživatelské rozhraní programu umožňuje jednoduchou editaci stylovacího souboru pomocí různých dialogových oken.

### 8.1 Návrh rozšíření

Vzhledem ke zvolenému tématu zpracování map se nabízí spousta možností, jak výslednou aplikaci rozšířit. Mnoho nápadů samo vyplynulo na povrch v průběhu samotné implementace.

Jako nejdůležitější rozšíření se jeví automatizace tvorby některých externích dat importovaných do mapy. V současné stavu je například nutné vrstevnice nejprve stáhnout z příslušného serveru a převést je do požadovaného formátu. Po automatizaci by byly vrstevnice staženy na základě vybrané oblasti mapy a příkazem externímu programu převedeny do výsledného formátu.

Velice zajímavým rozšířením se jeví možnost doplnit aplikaci o možnost exportu dlaždic dané oblasti, které slouží jako zdroj dat pro v dnešní době velice populární mapy ve webových prohlížečích.

Vzhledem k poměrně jednoduchému návrhu aplikace by dalším možným rozšířením bylo exportování map na základě příkazů v konzoli.

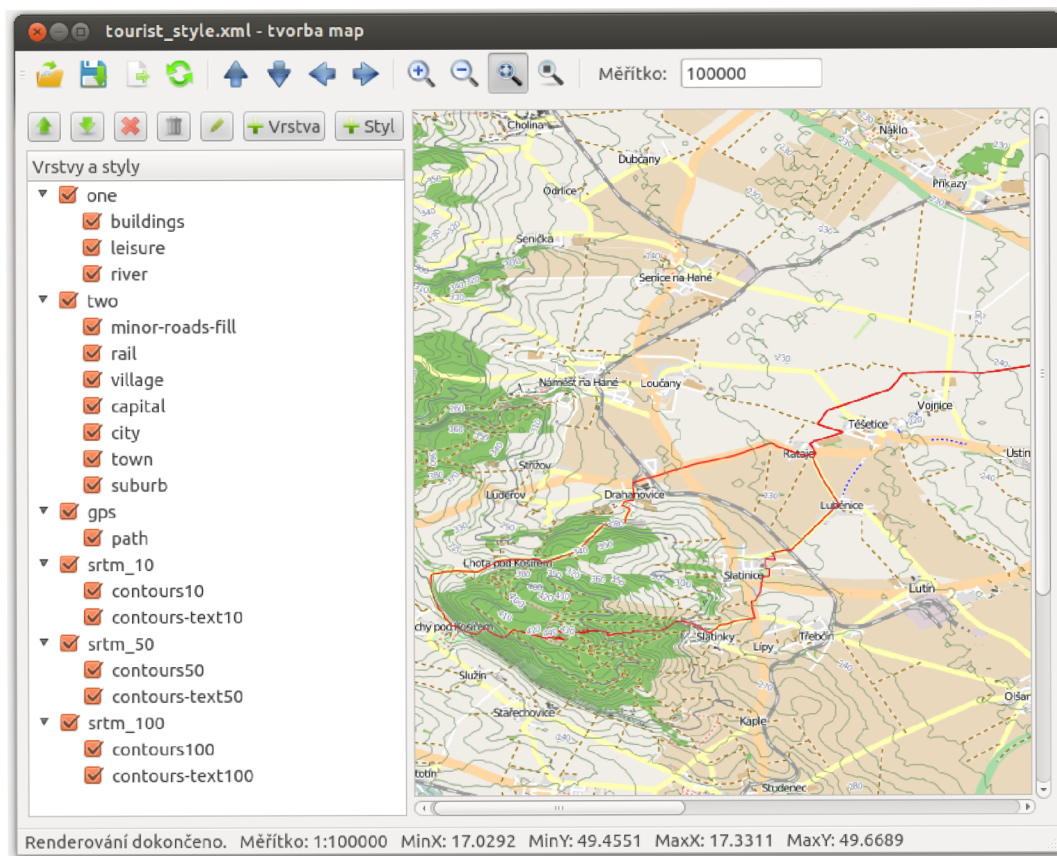
# Literatura

- [1] GDAL - Geospatial Data Abstraction Library [online]. 2013 [cit. 2014-01-28], URL <http://www.gdal.org/>.
- [2] Qt Project [online]. 2013 [cit. 2014-01-28], URL <http://qt.digia.com/>.
- [3] Coast, S.: Projekt OpenStreetMap (The Free Wiki World Map) [online]. 2013 [cit. 2014-01-27], URL <http://www.openstreetmap.org>.
- [4] Demek, J.; Voženílek, V.; Vysoudil, M.: *Geografie pro střední školy 1 (Fyziogeografická část)*. SPN, 1997, ISBN 80-85937-73-5.
- [5] Hrubý, M.: *Studijní opora - Geografické Informační Systémy (GIS)*. FIT VUT, Brno, 2006.
- [6] Konečný, M.; Kaplan, V.; Keprtová, K.; aj.: Multimediální učebnice kartografie a geoinformatika [online]. 2012 [cit. 2014-01-19], URL <http://oldgeogr.muni.cz/ucebnice/kartografie/obsah.php>.
- [7] Kvapil, J.: Kosmický segment GPS a jeho budoucnost. *Aldebaran Bulletin*, 2005, URL [http://www.aldebaran.cz/bulletin/2005\\_02\\_gps.php](http://www.aldebaran.cz/bulletin/2005_02_gps.php).
- [8] Ondráček, J.; kol.: Učební text turistika I [online]. 2006 [cit. 2014-01-18], URL <http://is.muni.cz/elportal/estud/fspjs/js07/turistika/index.html>.
- [9] Pavlenko, A.: Mapnik - Free Toolkit for developing mapping applications [online]. 2013 [cit. 2014-01-28], URL <http://www.mapnik.org>.
- [10] Tuček, J.: *Geografické informační systémy - Principy a praxe*. Computer Press, 1998, ISBN 80-7226-091-X.
- [11] Volland, M.: Wikipedia.org - obrázek geoidu [online]. 2013 [cit. 2014-02-03], obrázek ve formátu SVG. URL <http://commons.wikimedia.org/wiki/File%3AGeoida.svg>.
- [12] Voženílek, V.: *Aplikovaná kartografie I*. Univerzita Palackého v Olomouci, 2001, ISBN 80-244-0270-X.
- [13] Čada, V.: Přednáškové texty z Geodézie [online]. 2007-05-17 [cit. 2014-01-21], URL <http://www.gis.zcu.cz/studium/gen1/html/>.

- [14] Čapek, R.; Mucha, L.; Mikšovský, M.: *Geografická kartografie*. SPN, 1992, ISBN 80-04-25153-6.

## Příloha A

# Ukázka okna výsledné aplikace



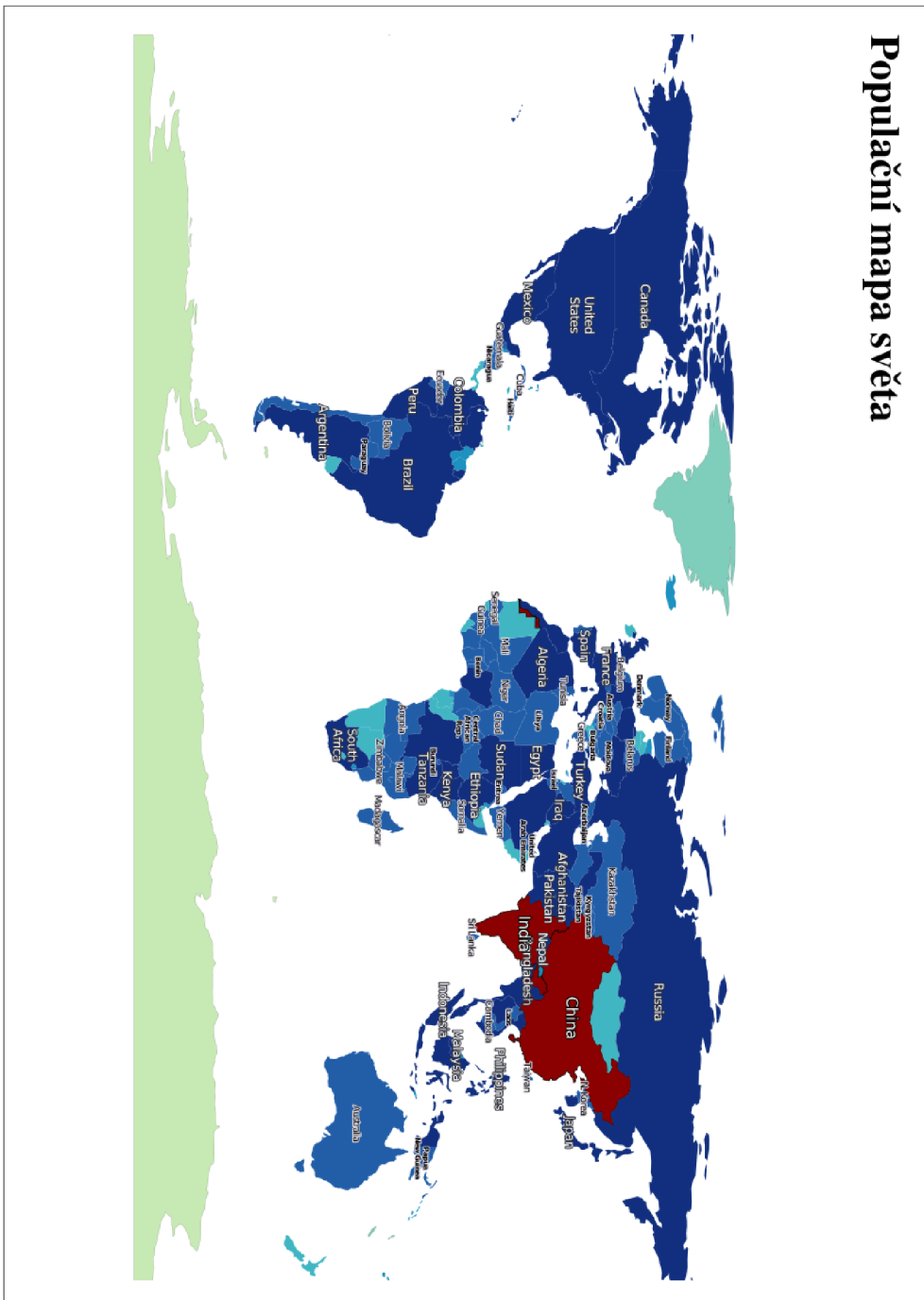
Obrázek A.1: Ukázka výsledného okna aplikace s vlastní trasou.

## Příloha B

# Ukázka exportovaných map

Na následujících stranách jsou přiloženy ukázky exportovaných map. Vložení map do této práce došlo k jejich zmenšení, a tudíž uvedené měřítko není zcela přesné.

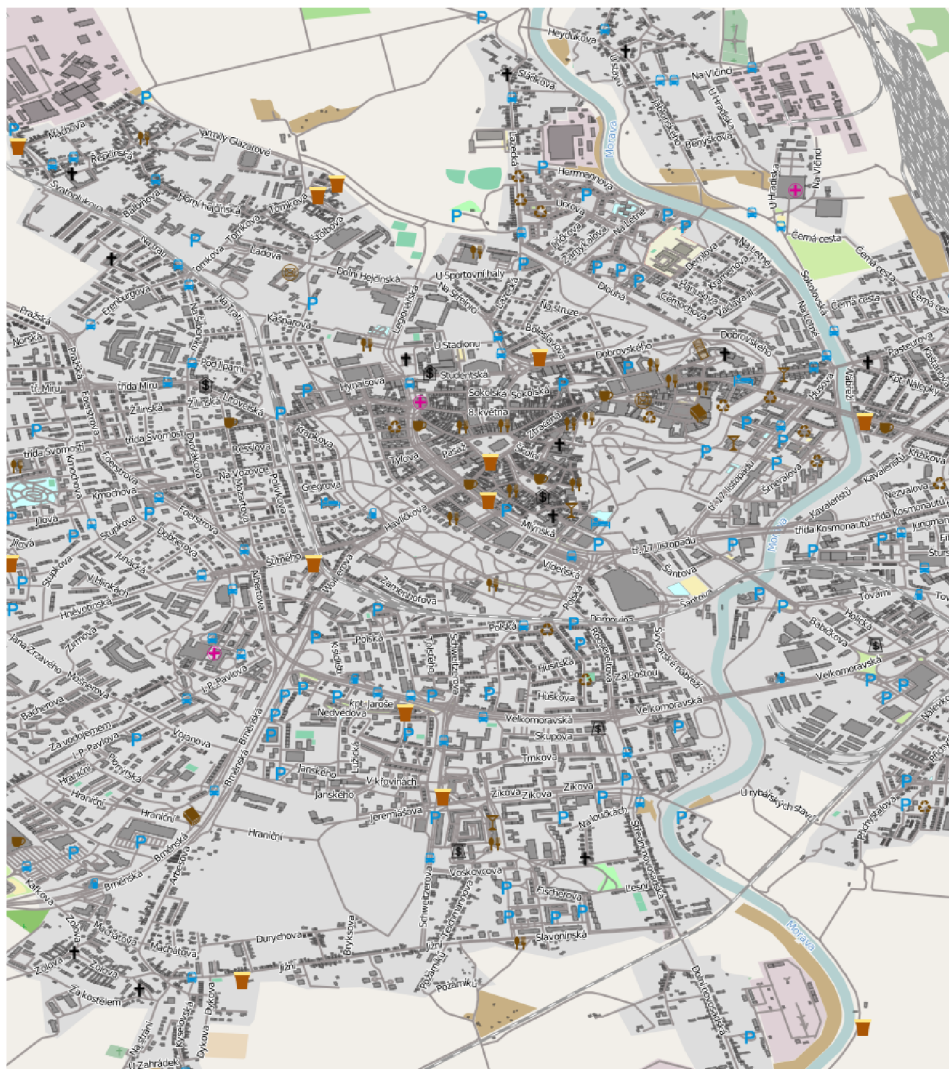
## Populační mapa světa



Obrázek B.1: Exportovaná populační mapa.

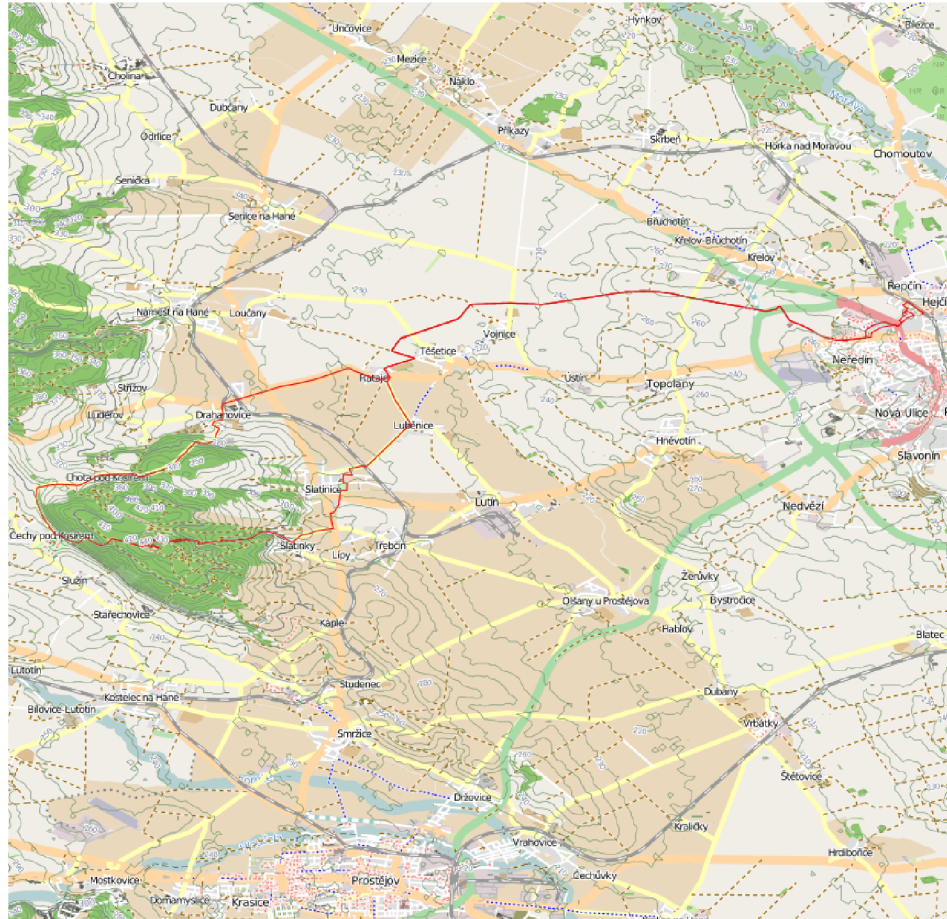


**Město Olomouc (1 : 25000)**

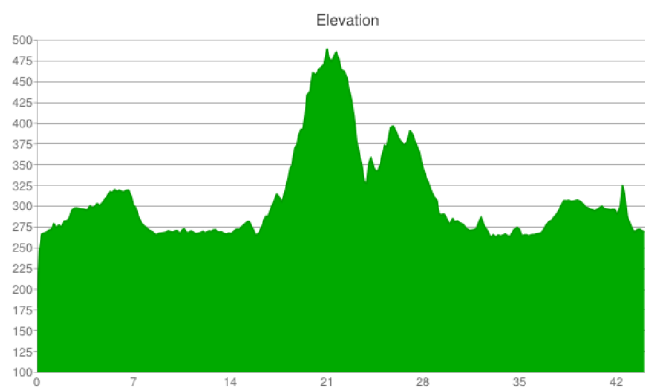


Obrázek B.2: Exportovaná mapa Olomouce.

### The map of the Velky Kosir track (1 : 120000)



Total Distance: 45,66 km (28,4 mi)  
 Total Time: 3:19:14  
 Moving Time: 2:37:22  
 Average Speed: 13,75 km/h (8,5 mi/h)  
 Average Moving Speed: 17,41 km/h (10,8 mi/h)  
 Max Speed: 48,63 km/h (30,2 mi/h)  
 Min Elevation: 178 m (584 ft)  
 Max Elevation: 495 m (1624 ft)  
 Elevation Gain: 979 m (3211 ft)  
 Max Grade: 20 %  
 Min Grade: -21 %  
 Recorded: Mon 8 29 15:32:43 GMT  
 +02:00 2011  
 Activity type: -



Obrázek B.3: Exportovaná turistická mapa s vlastní trasou.

# Příloha C

## Obsah CD

Na přiloženém datovém nosiči k této práci se nacházejí následující soubory a složky:

- složka *examples/* - exportované mapy a jejich stylovací soubory
- složka *doc/* - dokumentace aplikace ve formátu doxygen
- složka *src/* - zdrojové soubory aplikace
- složka *src-text/* - zdrojové soubory textu práce vytvořené pomocí  $\text{\LaTeX}$
- soubor *README.txt* - doplňující informace ke spuštění aplikace