

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

APLIKACE PRO CYKLISTY V PROSTŘEDÍ FLASH (AIR)

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JIŘÍ JONÁŠ

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

APLIKACE PRO CYKLISTY V PROSTŘEDÍ FLASH (AIR)

APPLICATION FOR BIKERS IN FLASH (AIR)

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JIŘÍ JONÁŠ

VEDOUCÍ PRÁCE

SUPERVISOR

ING. ZDENĚK VAŠÍČEK

BRNO 2008

Abstrakt

Předmětem této práce je návrh a implementace mapové aplikace, která umožní zobrazovat cyklistické trasy nad mapovými podklady. Aplikace umožní vytvoření vlastní trasy, pro kterou je možné generovat výškový profil. Trasu lze exportovat nebo importovat prostřednictvím GPX a KML souborů. Pomocí uživatelských účtů je možné vytvořenou trasu uložit. Aplikace je vytvořena v jazyce Haxe, což je open source varianta pro vývoj aplikací pro platformu Flash. Pro práci s uživatelskými účty je využit jazyk PHP a databáze MySql.

Abstract

The objective of this thesis is suggestion and implementation map application that allows display cycling routes over map layers. Application allows creating own route for which is possible generate elevation profile. Route can be exported or imported via GPX and KML files. Through user accounts is allowed approach to saved routes. Application is developed in programming language Haxe. It is open source option for developing applications for the Flash platform. For work with user accounts is used programming language PHP and database MySql.

Klíčová slova

Adobe Flash, ActionScript, Haxe, OpenStreetMap, Microsoft Bing Maps, KML, GPX, Mercatorovo zobrazení, Web, Databáze, PHP, MySql

Keywords

Adobe Flash, ActionScript, Haxe, OpenStreetMap, Microsoft Bing Maps, KML, GPX, Mercator's projection, Web, Database, PHP, MySql

Citace

Jonáš Jiří: Aplikace pro cyklisty v prostředí Flash (AIR), bakalářská práce, Brno, FIT VUT v Brně, 2011

Aplikace pro cyklisty v prostředí Flash (AIR)

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Zdeňka Vašíčka.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Jiří Jonáš

13. května 2011

Poděkování

Děkuji vedoucímu mé bakalářské práce Ing. Zdeňku Vašíčkovi za ochotu, užitečné rady, náměty a inspiraci k vytvoření této práce.

© Jiří Jonáš, 2011

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	2
2 Použité technologie	3
2.1 Platforma Flash	3
2.2 Programovací jazyk PHP	4
2.3 Databáze MySql	4
2.4 Programovací jazyk Haxe	5
2.5 Knihovna HaxeMaps	5
2.6 Výškový profil terénu - SRTM	6
2.7 Formáty pro ukládání geografických dat	7
2.8 Volně dostupné mapové podklady	8
3 Návrh aplikace	14
3.1 Návrh uživatelského rozhraní	14
3.2 Tvorba výškového profilu	14
3.3 Ukládání tras	17
4 Implementace	19
4.1 Objektový model	19
4.2 Navigační lišta	19
4.3 Výškový profil	19
4.4 Export a import XML souborů	20
4.5 Práce s uživatelskými účty	20
5 Testování	21
5.1 Testování přesnosti nadmořské výšky	21
5.2 Testování vytvořené trasy	22
5.3 Testování trasy z GPS přístroje	23
5.4 Zhodnocení testování	24
6 Závěr	25
Literatura	26
Seznam příloh	28

1 Úvod

Cyklistika je v České republice často vyhledávanou volnočasovou aktivitou. Velkou zásluhu má na tomto faktu poměrně rozsáhlá síť cyklostezek. Snad z každé obce v naší vlasti vede do sousední vesnice stezka, kterou by bylo možné zdolat prostřednictvím jízdního kola. Cyklistům jsou nakloněny i větší města, v jejichž dopravní infrastruktuře se cyklostezky stávají pevnou součástí. Není tedy divu, že při výběru dopravního prostředku na rodinný výlet se kolo stává častou volbou.

Správný cyklistický nadšenec stále objevuje nová místa, která by mohl se svým bicyklem prozkoumat. Je tedy nutné věnovat cestě, na které hodlá člověk objevovat nová zákoutí naší republiky, patřičnou přípravu. Ještě před několika lety by každý turistický fanda vybíral trasu svojí cesty prostřednictvím papírové mapy. Jenže včera není dnes. Ve světě, kde vládnu počítače a internet, již mapa pomalu ztrácí svoji pozici. Využijeme-li společně s počítačem „GPSku“, můžeme mapu prakticky odhodit do koše.

Stále častější variantou, jak navrhnout trasu, kterou by cyklista mohl absolvovat, je využit služeb mapových aplikací dostupných online. Rozmach mapových služeb byl odstartován aplikací Google Maps, která byla spuštěna roku 2005.

Tato práce se zabývá návrhem a tvorbou mapové aplikace, která by umožňovala tvorbu tras, kterým by bylo možné zobrazit výškový profil. Měla by také umožnit zobrazení trasy, která byla zaznamenána prostřednictvím GPS přístroje. Aplikace by měla umožňovat vytváření uživatelských účtů, prostřednictvím kterých by byl udržován seznam vytvořených tras.

Práce je členěna následovně. Ve druhé kapitole jsou postupně popsány nástroje, které jsou k implementaci projektu využity, a technologie, jejichž znalost je potřebná k vývoji aplikace. Třetí kapitola se zabývá návrhem aplikace. Čtvrtá kapitola popisuje implementaci jednotlivých částí projektu. V páté kapitole je provedeno testování programu.

Závěr obsahuje zhodnocení práce a dosažených výsledků. Jsou zde nastíněny možnosti, kterými by se vývoj mohl v budoucnu dále ubírat.

2 Použité technologie

Cílem tohoto projektu je vytvoření mapové aplikace na platformě Flash, pro jejíž vypracování bude využito open source nástrojů. Pro vytvoření vlastní aplikace byl proto vybrán multiplatformní jazyk Haxe, který umožňuje vytvořit aplikaci pracující na platformě Flash. Pro provádění operací na straně serveru je využit skriptovací jazyk PHP a k ukládání informací o uživatelských účtech byla uplatněna databáze MySql.

2.1 Platforma Flash

Flash je multimediální platforma. Umožňuje vytvářet vektorové i rastrové grafické objekty. Využívá se k vytváření jednoduchých 2D animací nebo k tvorbě interaktivních animací. Flash platforma vytváří aplikaci jako reprezentaci vektorových objektů, čímž podstatně snižuje velikost výsledné aplikace. Z tohoto důvodu je Flash oblíbenou technologií k vytváření jednoduchých webových her nebo reklamních bannerů. Flash podporuje práci s multimédií jako je streamování videa, obousměrný přenos zvuku i obrazu.

Autorem platformy Flash je firma Macromedia, která jej v roce 1996 vytvořila spojením programu FutureSplash Animator s vlastním projektem Shockwave. Vznika tak platforma s názvem Shockwave Flash. V roce 2005 koupila firmu Macromedia společnost Adobe Systems, která změnila název této technologie na Adobe Flash. [1]

2.1.1 Rich Internet Application

Technologie Adobe Flash se řadí mezi Rich Internet Application (RIA). Jedná se o aplikace, které jsou připojeny k webovým stránkám, ale samotný webový prohlížeč je není schopen interpretovat. Aby bylo možné tento nestandardní obsah webových stránek využívat, je nutné do webového prohlížeče nainstalovat zásuvný modul, který se postará o správné zobrazení této aplikace. Mezi RIA aplikace jsou dále řazeny platformy Java a Microsoft Silverlight. [2]

2.1.2 Flash Player vs. AIR

Aplikace vytvořené na platformě Flash jsou reprezentovány soubory ve formátu SWF (ShockWave Flash). Tyto soubory, mohou být spouštěny přehrávačem Flash Player, nebo pomocí běhového prostředí AIR.

Výslednou aplikaci je možné připojit k webové stránce, jenže samotný webový prohlížeč nedokáže tento program spustit. Je tedy nutné nainstalovat do webového prohlížeče program Flash Player, který zajistí správné chování aplikace.

Programy vytvořené pod platformou Adobe Flash, které běží na webových stránkách, mohou fungovat také jako desktopové aplikace. Pro tento účel bylo vytvořeno firmou Adobe prostředí AIR (Adobe Integrated Runtime). Jedná se o multiplatformní běhové prostředí, které je schopné interpretovat aplikace vytvořené pod platformou Adobe Flash, Adobe Flex, HTML a Ajax.

Výhodou použití prostředí AIR je daleko lepší přístup k souborům, které jsou uloženy na místním disku, zatímco aplikace spuštěné prostřednictvím webového prohlížeče, mají k externím souborům omezený přístup. Nevýhodou prostředí AIR je ovšem nutnost vytvoření instalačního balíčku, který musí být nainstalován na cílovém uživatelském počítači. [2]

2.1.3 Jazyk ActionScript

ActionScript je objektově orientovaný programovací jazyk, který je využíván k vývoji aplikací pod platformou Adobe Flash. Syntaxe jazyka ActionScript je odvozena od skriptovacího jazyka ECMAScript. Je určen primárně pro vývoj webových stránek a aplikací, které pracují na platformě Adobe Flash Player.

Kód jazyka ActionScript je kompilován do formátu bajtového kódu, který je vnořen do souboru SWF. Tento program je spouštěn prostřednictvím virtuálního stroje ActionScript Virtual Machine, který je součástí jak přehrávače Adobe Flash Player, tak programu AIR.

Jazyk ActionScript 3.0 zavádí koncept uzavřených tříd, kdy má třída fixně danou sadu vlastností a metod, které byly definovány v době kompilace, a přidávání dalších vlastností již poté není umožněno. Výhodou je, že je možné při kompilaci odhalit více chyb, které by vedly k pádu programu. Vytvářené programy jsou tak stabilnější. Ve třetí verzi jazyka ActionScript je také zavedena nativní podpora pro regulární výrazy, což umožňuje rychlé vyhledávání a manipulaci s řetězci. Implementace regulárních výrazů je zavedena podle specifikace jazyka ECMAScript.

Pro komunikaci mezi objekty a reakci na změny, které nastaly za běhu aplikace je v jazyku ActionScript využíváno generování a zpracovávání událostí. Koncept událostí je využíván při interakci s uživatelem, jako je třeba klepnutí myši nebo vyplnění formuláře. Dále jsou události využívány například při dokončení načítání dat ze serveru či při připojení kamery. Zpracovávání událostí je postaveno na objektovém modelu DOM3. [3]

2.2 Programovací jazyk PHP

Jazyk PHP (Personal Home Page) původně vznikl za účelem tvorby dynamických webových stránek. Význam zkratky PHP byl později změněn na Hypertext Processor. Jazyk PHP je vyvíjen celosvětovou komunitou na základě otevřeného kódu, a výsledný software je volně k dispozici. [4]

PHP je dynamicky typovaný skriptovací jazyk, který je platformně nezávislý. Primárně je určený k vývoji webových aplikací. PHP kód se vkládá přímo do zdrojového HTML dokumentu. Pokud se při zpracovávání webové stránky vyskytne kód v jazyce PHP, je tato část předána nejprve PHP procesoru, který vložený kód provede a klientovi zašle již výsledný HTML kód. Jedná se tedy o tzv. server-side jazyk, což znamená, že veškeré operace jsou provedeny na serverové stanici a uživateli je zaslán pouze výsledek skriptu. PHP podporuje přístup k většině databázových systémů jako je například MySQL, PostgreSQL, Oracle nebo MSSQL. Může také pracovat ve spolupráci s jazykem ActionScript, kde je využit například ke komunikaci s databázemi.

Syntaxe jazyka vychází z jazyků Pearl, C, Pascal a Java. Jazyk obsahuje velké množství knihoven pro zpracování řetězců, XML souborů, pro práci s databázemi, matematickými funkcemi atd. [5]

2.3 Databáze MySql

Jedná se o multiplatformní relační databázi, která je k dispozici pod licencí GPL, která nevyžaduje zakoupení komerční licence, pokud se nejedná o distribuci MySql jako produktu bez zdrojových kódů. Za těchto podmínek by zakoupení licence bylo nutné. [6]

Jak již z názvu vyplývá, komunikace s databází je zprostředkována pomocí dotazovacího jazyka SQL. Vývoj databáze MySql byl zaměřen především na rychlost, což se projevilo omezením

některých funkcí. Databáze podporuje jen jednoduché zálohování, a do verze 5.0 neumožňovala vytváření pohledů, triggerů a uložených procedur. [7]

2.4 Programovací jazyk Haxe

Haxe je open source programovací jazyk. Vychází z MTASC, což je překladač kódu jazyka ActionScript 2. Po ukončení vývoje druhé verze jazyka ActionScript byl vývoj MTASC rovněž ukončen. Po vydání jazyka ActionScript3 tedy přichází jeho nástupce Haxe. Haxe, oproti MTASC není zaměřen pouze na jazyk ActionScript. Jedná se totiž o multiplatformní jazyk. Umožňuje vytvářet aplikace v prostředí flash, multiplatformní klientské i serverové webové aplikace nebo desktopové aplikace pro různé operační systémy včetně aplikací pro mobilní telefony.

Kód jazyka Haxe má mnoho společného s jazykem AS3, ale disponuje mnohými vylepšeními a možnostmi vytvářet aplikace pro různé platformy, což ho řadí do zcela jiné kategorie než překladače Adobe. [9]

Různé příklady využití jazka Haxe a cílové platformy, jejichž kód je schopen generovat:

- Klientské webové aplikace – Flash a JavaScript
- Serverové webové aplikace – PHP, Neko
- Desktopové aplikace pro operační systémy Windows, Linux nebo Mac OS – C++, Neko
- Nativní kód pro mobilní aplikace pracující na platformě iOS, Android nebo webOS

2.5 Knihovna HaxeMaps

Jedná se o open-source knihovnu pro Flash, která poskytuje aplikační rozhraní k zobrazování mapových podkladů. Tato knihovna je napsaná v jazyce Haxe, který je možné převést do kódu jazyka ActionScript 3.0. Obsahuje jednoduché aplikační rozhraní, které je možné jednoduše rozšiřovat. Umožňuje využití rastrových i vektorových mapových podkladů a umožňuje připojení k jakékoli mapové službě. [10]

Základní třídy knihovny HaxeMaps: [10]

- *Canvas* – je použita jako vykreslovací plátno mapové aplikace a má na starosti zobrazení a správu mapových vrstev. Každá vrstva může být nastavena jako statická nebo interaktivní, což znamená, že bude reagovat na události, které přicházejí od uživatele. Jednotlivé vrstvy jsou přidávány tak, že novější vrstva překrývá starší. Vrstvy mohou být nastaveny jako neaktivní, přičemž tato vrstva není viditelná, ani není aktualizována při interakci s plátnem.
- *Layer* – bazová třída, která se stará o vykreslování mapového podkladu. Každé instanci této třídy, která představuje jednu vrstvu třídy *Canvas*, je možné nastavit, zda se při změně měřítko mapový podklad pouze přiblíží, což je vhodné u vektorových vrstev, nebo zda se musí obsah vrstvy načíst znovu. Obsahuje ukazatel na mapovou službu *map_service*, který se využívá k transformaci souřadnic z geodetických na rovinné a k vykreslování rastrových dat. Třída obsahuje funkce, pomocí kterých lze vypočítat polohu mapového podkladu vůči výřezu okna a nastavení přiblížení mapy.
- *MapService* – používá se k napojení na konkrétní mapovou službu. Prostřednictvím této vrstvy je poskytnuta URL adresa konkrétní dlaždice mapového podkladu a je provedena transformace z geodetických souřadnic na pravoúhlé. Tato třída implementuje rozhraní pro

načítání dlaždic OpenStreetMap a Bing Maps, jejichž licenční podmínky dovolují používat tyto mapové podklady ve vlastním programu.

- *LngLat* – třída, pomocí které jsou uchovávány geografické souřadnice bodu na mapě.
- *TileLayer* – třída rozšiřující třídu *Layer*. Instanci této třídy je předán ukazatel na mapovou službu, prostřednictvím které budou získány URL adresy jednotlivých mapových dlaždic. Třída dále definuje kolik dlaždic bude načítáno při zobrazení mapy a určuje chování aplikace při změně měřítka. Je možné nastavit, že se při zoomování všechny dlaždice odstraní a vyčká se na načtení nových dlaždic, nebo lze rozdělením dlaždice zajistit postupné překreslování. Při přibližování mapy je dlaždice rozdělena na čtyři části, které se překreslí novými dlaždicemi, nebo naopak při oddálení jsou čtyři dlaždice překresleny jednou novou dlaždicí.
- *TileLoader* – využívá se k načítání dlaždic a má na starosti, že se prioritně načtou dlaždice, které jsou aktuálně viditelné. Dále zajistí mazání neaktuálních požadavků ve frontě. Pokud se nezdaří načtení dlaždice, zajistí opakované načítání v intervalu pěti sekund, maximálně však třikrát. Vytváří si také frontu dlaždic, pomocí kterých vyplní dobu, kdy se při změně měřítka načítají nové dlaždice.

2.6 Výškový profil terénu - SRTM

SRTM neboli Shuttle Radar Topography Mission je název mezinárodního výzkumného programu, jehož cílem je získat co nej přesnější hodnoty nadmořských výšek k vytvoření topografické databáze Země v co nejlepším rozlišení. Mise SRTM, při které byla data získávána, se uskutečnila v únoru 2000 a trvala jedenáct dnů. Během této doby bylo dvakrát nasnímáno více jak 90% území celého světa, a více jak 50% zemského povrchu dokonce třikrát. Měření probíhalo pomocí speciální radarové soupravy, pracující na frekvencích, které umožňují provádět měření bez ohledu na denní dobu a meteorologické podmínky.

Naměřené hodnoty jsou distribuovány v souborech, které pokrývají území o rozměrech jeden stupeň zeměpisné šířky na jeden stupeň zeměpisné délky. V místě okolo rovníku toto území přibližně odpovídá čtverci o straně sto jedenáct kilometrů.

Data jsou dostupná ve dvou verzích. První verze – 1.0 je přímým výsledkem interferometrického měření. Verze 2.0 je výsledkem následných zpracovatelských operací, které se zaměřily na úpravu problematických oblastí, jako jsou vodní plochy, pobřežní oblasti, nebo oblasti okolo pohoří, kde se nacházejí velké sklony. [11]

2.6.1 Formát SRTM dat

SRTM data jsou uložena do souborů, které jsou pojmenovány podle zeměpisné souřadnice jihozápadního rohu dlaždice, která definuje obor hodnot nadmořské výšky v souboru.

Verze dat:

- SRTM1 – jedná se o soubor dat s nejvyšším rozlišením. Zemská plocha je vzorkována po jedné vteřině ve směru zeměpisné délky i šířky. Jedná se o vzorky, které se v oblasti rovníku nacházejí přibližně třicet metrů od sebe. Soubor, který je nosičem dat v tomto rozlišení, obsahuje v každém z 3601 řádků 3601 vzorků. Výškové hodnoty v tomto rozlišení jsou dostupné pouze pro území v USA.
- SRTM3 – data jsou vzorkována po třech vteřinách ve směru zeměpisné délky a šířky což v oblasti rovníku odpovídá devadesátimetrovým rozestupům bodů. Tato verze dat vznikla

zprůměrováním devíti okolních bodů, které byly uloženy v rozlišení SRTM1. V souboru je uloženo v každém z 1201 řádků 1201 vzorků. Zprůměrováním předchozích hodnot bylo dosaženo zhruba trojnásobného snížení chyby, která vznikla při měření. Hodnoty v tomto rozlišení jsou dostupné na území mezi 56° jižní zeměpisné šířky a 60° severní zeměpisné šířky.

- SRTM30 – hodnoty v tomto rozlišení jsou získány průměrem 10x10 hodnot z rozlišení SRTM3. Hodnoty jsou od sebe ve vzdálenosti 30 vteřin, což odpovídá zhruba devíti stům metrů v oblastech okolo rovníku. Data jsou dostupná v dlaždicích 5°× 5°. V tomto rozlišení je možné získat hodnoty nadmořské výšky téměř celého světa kromě Antarktidy a území mezi šedesátou severní rovnoběžkou a severním pólem.

Při měření docházelo v důsledku náhodného šumu k chybám. Tím vznikly zastíněná místa, u kterých nebylo možné nadmořskou výšku určit. V celkovém měřítku nepřesáhly tyto chyby 0,15% všech SRTM dat. Průměrováním dat se tato chyba snižovala podle druhé mocniny. Ve výsledku se může stát, že některá místa zůstala bez určené hodnoty. Počet těchto míst se vyskytuje v řádu jednotek.

Data jsou ukládána do .hgt souborů. Hodnoty nadmořské výšky jsou v souboru uloženy jako celé 16bitové číslo se znaménkem. Hodnoty jsou uloženy ve dvojrozměrném poli binárních hodnot. Soubor neobsahuje žádné počáteční ani ukončovací bajty, pouze samotná data. Všechny uložené hodnoty jsou v metrech. Bajty jsou řazeny podle standardu „big-endian“, kde se na prvním místě nachází nejvýznamnější bajt. Hodnoty nadmořské výšky, které je možné do souboru uložit, jsou v rozmezí od -32767 do 32767. Nadmořská výška je definována v metrech. Toto rozmezí spolehlivě vystačí na hodnoty, kterých je možné na Zemi dosáhnout.

Tyto údaje také obsahují příležitostná místa, ve kterých není hodnota v důsledku chyb při měření definována. Tato místa jsou označena hodnotou -32767. [11]

2.7 Formáty pro ukládání geografických dat

2.7.1 Formát KML

KML (Keyhole Markup Language) je univerzální formát na bázi XML pro uložení geografických dat. Formát byl vytvořen firmou Keyhole, která tento formát vyvinula pro použití v programu Keyhole Earth Viewer, který byl používán jako 3D mapa. Od roku 2004, kdy firmu Keyhole koupila společnost Google, se jméno této aplikace změnilo na Google Earth.

KML verze 2.2 se stal roku 2004 mezinárodním standardem zapsaným u standardizační organizace OGC. Formát může být využit k ukládání tras nebo bodů na mapě. Tyto údaje jsou využívány dvojdímenzionálními mapami. Programy na principu virtuálního globusu, jejímž představitelem je právě Google Earth, mohou do souboru navíc uložit prostorové objekty, které zobrazují například budovy, nebo mohou uložit dráhu letu letadla, která se nachází nad zemským povrchem.

Do souboru KML je možné uložit místa na mapě, značky, obrázky, trasy, 3D modely, textové popisky, a styly zobrazovaných objektů. Každé místo na mapě, které je v KML uloženo, má definovanou svoji zeměpisnou polohu. Dále je možné určit jeho nadmořskou výšku, popis, nebo umístění kamery, neboli polohu bodu, odkud bude mapový podklad pozorován. [13]

Pro lokalizaci prvků standardu KML se využívá celosvětově uznávaného geodetického standardu WGS84. Na jeho vývoji se podílelo ministerstvo obrany USA. Byl vytvořen na základě měření pozemních stanic družicového polohového systému a definuje souřadnicový systém, referenční elipsoid a geoid pro geodezii a navigaci. Zápis zeměpisných šířek a délek může být prováděn ve stupních, stupních a minutách, nebo ve stupních, minutách a vteřinách, vždy s jednotkami v desetinném tvaru. Standard KML ukládá souřadnice ve tvaru celých stupňů. [12]

KML soubory mohou být uloženy ve dvou formátech. Prvním je soubor KML, který obsahuje zápis geodetických dat ve formátu XML. Druhým typem je soubor KMZ. Jedná se o adresářovou strukturu, která ve svém kořenovém adresáři obsahuje KML soubor, který je pojmenovaný doc.kml. Dále může obsahovat doplňující obrázky, videa, zvuky nebo 3D modely. Z této adresářové struktury je vytvořen soubor, který je komprimovaný programem zip. Výsledný soubor se od klasického zip souboru liší pouze koncovkou souboru, která je změněna na .kmz. [13]

2.7.2 Formát GPX

GPX je formát pro výměnu geografických dat využívající XML, který se běžně využívá k ukládání dat vytvořených prostřednictvím GPS přístrojů a jejich výměně mezi aplikacemi a webovými službami. Do souboru je možné uložit geometrické tvary, jako jsou waypointy, trasy, nebo tracky. Waypoint je bod na mapě určený zeměpisnými souřadnicemi. Trasa se skládá z jednotlivých waypointů a slouží k postupné navigaci mezi nimi. Track představuje trasu, která je vytvořena například pomocí GPS přijímače, který projížděnou trasu zaznamenává. GPX byla vydána ve dvou verzích. První verze 1.0 vyšla v roce 2002. Druhá verze je označena 1.1, používá se od roku 2004 a není s verzí 1.0 kompatibilní. [14]

GPX soubor je tvořen kořenovým uzlem *gpx*. Tento element je v každém GPX souboru povinný. Obsahuje dva atributy *creator* a *version*, které jsou povinné. Soubor pak může obsahovat neomezený počet tras. Trasa je tvořena elementem *trk*. Jednotlivé body, kterými je trasa tvořena, jsou definované elementem *trkpt*, který má povinné atributy *lat* a *lon*. Pomocí těchto atributů jsou uloženy zeměpisné souřadnice daného bodu. Jednotlivé body na trase mohou obsahovat volitelné parametry jako je například *name* pomocí kterého je uloženo jméno bodu nebo *desc*, který definuje popis daného místa.

2.8 Volně dostupné mapové podklady

Programy, které slouží k zobrazení map, využívají mapové podklady. Jedná se o grafické zobrazení zemského povrchu, které je rozděleno na dlaždice. Každá dlaždice představuje určité místo na Zemi v daném přiblížení. Pro využití mapových služeb ve vlastní aplikaci je možné použít volně dostupné mapové podklady, které definují politiku *crossdomain.xml*. Mezi tyto služby se řadí projekt OpenStreetMap, nebo BingMap. Pokud licenční podmínky poskytovatele služby volné využití neposkytují, je nutné od něj získat potřebná povolení.

2.8.1 Mapové projekce

Matematické vyjádření, které reprezentuje pomocí dvojrozměrné mapy trojrozměrný zemský povrch, se nazývá mapová projekce. Tímto procesem vždy narušíme jednu, nebo více vlastností mapy. Převodem mapy můžeme zdeformovat například její měřítko, tvar nebo rozměr. Za tímto účelem byly vytvořeny stovky projekcí, které se přesně zaměřují na určitý prvek mapy. Mapová projekce je poté vybrána podle toho, kterou vlastnost pro daný úkol potřebujeme zobrazit přesněji. [20]

2.8.1.1 Mercatorova projekce

Jedná se o druh válcového mapového zobrazení, které bylo původně vytvořeno pro námořní účely. Základem zobrazení je válec, který se dotýká Zeměkoule na rovníku a je rovnoběžný se zemskou osou. Po převedení zemské plochy do roviny je vytvořena pravoúhlá síť poledníků a rovnoběžek. Z obrázku 2.5 je patrné, že poledníky jsou od sebe vzdáleny ve stejných rozestupech, zatímco vzdálenost rovnoběžek se směrem k pólům zvětšuje do nekonečna.



Obrázek 2.5: Mercatorovo zobrazení

Z obrázku 2.5 je možné pozorovat, že jsou všechny rovnoběžky zobrazeny jako úsečky o stejné délce, ovšem ve skutečnosti se délka rovnoběžek směrem k pólům zmenšuje. Tento fakt má za následek, že mapa, zobrazená Mercatorovou projekcí, dosahuje směrem k pólům stále většího zkreslení. V oblastech pólů je zkreslení tak velké, že není možné toto zobrazení vůbec použít. Mapy jsou tedy zobrazeny pouze do 86. rovnoběžky. I přesto, že je v oblasti rovníku chyba způsobená zkreslením zanedbatelná, není tato metoda vhodná pro přehledné světové mapy. Grónsko je zobrazeno téměř tak velké, jako celá Afrika, která je svojí rozlohou více jak desetkrát větší. V Mercatorově zobrazení je loxodroma, což je čára, která protíná všechny poledníky pod stejným úhlem, zobrazena jako přímka. Této skutečnosti lze dobře využít k udržování stálého směru plavby, nebo k udržení přímého směru letu.

Mercatorova projekce je využívána mapovými servery jako je Google maps, Bing Maps, nebo projekt OpenStreetMap [17], [18]. Důvodem proč je tato projekce využívána i přes vysokou míru zkreslení, je zachování úhlů. Tyto mapové služby se zaměřují spíše na zobrazení ulic a měst, kde je zobrazení pod správnými úhly nezbytnou záležitostí. Zobrazení měst se odehrává na vysoké úrovni přiblížení, při kterém není patrné zkreslení, které je projekcí vytvořeno. [19]

2.8.2 OpenStreetMap

OpenStreetMap (OSM) je společný projekt, zabývající se tvorbou volně dostupných, editovatelných mapových podkladů celého světa. K získávání geodetických dat, pro tvorbu těchto map jsou využívány záznamy z GPS zařízení, letecké snímky, data z jiných volně dostupných map, nebo místní znalosti uživatelů. Dalším, velmi cenným zdrojem, se v poslední době stávají vládní organizace. V USA federální úřady poskytly údaje, pokrývající celé území Spojených států. Do projektu OSM byla využita i data z katastrálních map České republiky. Mezi přispěvatele se řadí i komerční firmy, jako třeba Automotive Navigation Data, která poskytla kompletní síť Nizozemských silnic. Kvalitní

satelitní mapy byly uvolněny společností Yahoo. Příznivě se ke tvorbě těchto map postavila i firma HELP SERVICE – REMOTE SENSING, která dodala kompletní síť českých silnic 1. a 2. třídy.

Původní způsob shromažďování OSM dat je založený na principu wikipedie. Tento způsob funguje tak, že uživatel, který se na tvorbě map podílí, s pomocí GPS přístroje systematicky zmapuje určitou oblast. Naměřené hodnoty jsou poté převedeny do počítače, upraveny do OSM formy, případně popsány a uloženy do databáze. Tyto operace může provádět jakýkoli uživatel, který je zaregistrovaný na stránkách OSM. Registrovaným uživatelům je umožněna jednoduchá editace dat.

V současné době je projektem nejlépe zmapovanou oblastí západní Evropa, kde tento projekt vznikl, a území USA. Velký podíl na kvalitním zpracování map nad Evropou má velká uživatelská základna, která OSM nejen vyživá, ale v dostatečné míře i tvoří. Přestože je v USA podstatně menší počet uživatelů, kteří by se na tvorbě map podíleli, jsou údaje popisující tento stát velmi kvalitní. Velkou zásluhu na tomto faktu má projekt americké vlády nazvaný Topologically Integrated Geographic Encoding and Referencing (TIGER). Jedná se o soubor dat, které popisuje Zemi pomocí atributů, jako jsou například silnice, železnice, budovy, vodní toky, jezera atd. Tento projekt vznikl na základě sčítání lidí, které v Americe probíhá každých deset let. TIGER soubory neobsahují údaje o sčítání, ale pouze mapy. Data projektu TIGER jsou volně dostupná, protože se jedná o dílo, které bylo vytvořeno z veřejných prostředků.

Mapy OSM jsou tvořeny pomocí jednoduchých objektů. Prvním z nich je bod (node), dalším je cesta (way), kterou tvoří spojnice dvou nebo více bodů. Speciálním případem cesty je plocha (area), kterou tvoří minimálně tři body na mapě a tím vytváří uzavřenou oblast. Posledním objektem, kterým jsou mapy tvořeny, je oblast, která má uvnitř prázdné místo. Tento objekt může popisovat například hranice, trasy autobusových linek atd.

Mapy, které uživatel vytvoří, jsou uloženy v databázi ve formě geoprostorových vektorových dat. Aby bylo možné mapy zobrazovat, je nutné je z této textové podoby převést do podoby grafické. Tento proces, pomocí kterého se mapy vykreslují, se nazývá rendering. Jedná se o převod vektorových dat do podoby rastrových dlaždic, které mají rozměry 256 × 256 pixelů. OSM používá pro renderování několik programů. Mezi nejznámější patří Mapnik a Osmarender. Mapnik je open source nástroj. Mapový podklad vytvořený tímto renderem je vytvořen pro 18 různých přiblížení. Generování map probíhá jednou týdně. Příklad mapy vytvořené tímto nástrojem je na obrázku 2.1. Osmarender umožňuje generování mapy na vyžádání, takže je možné okamžitě zkontrolovat, jak se provedené změny zobrazily na mapě. Vytváří 17 úrovní přiblížení. Příklad mapy vygenerované renderem Osmarender je na obrázku 2.2.



Obrázek 2.1: Mapnik



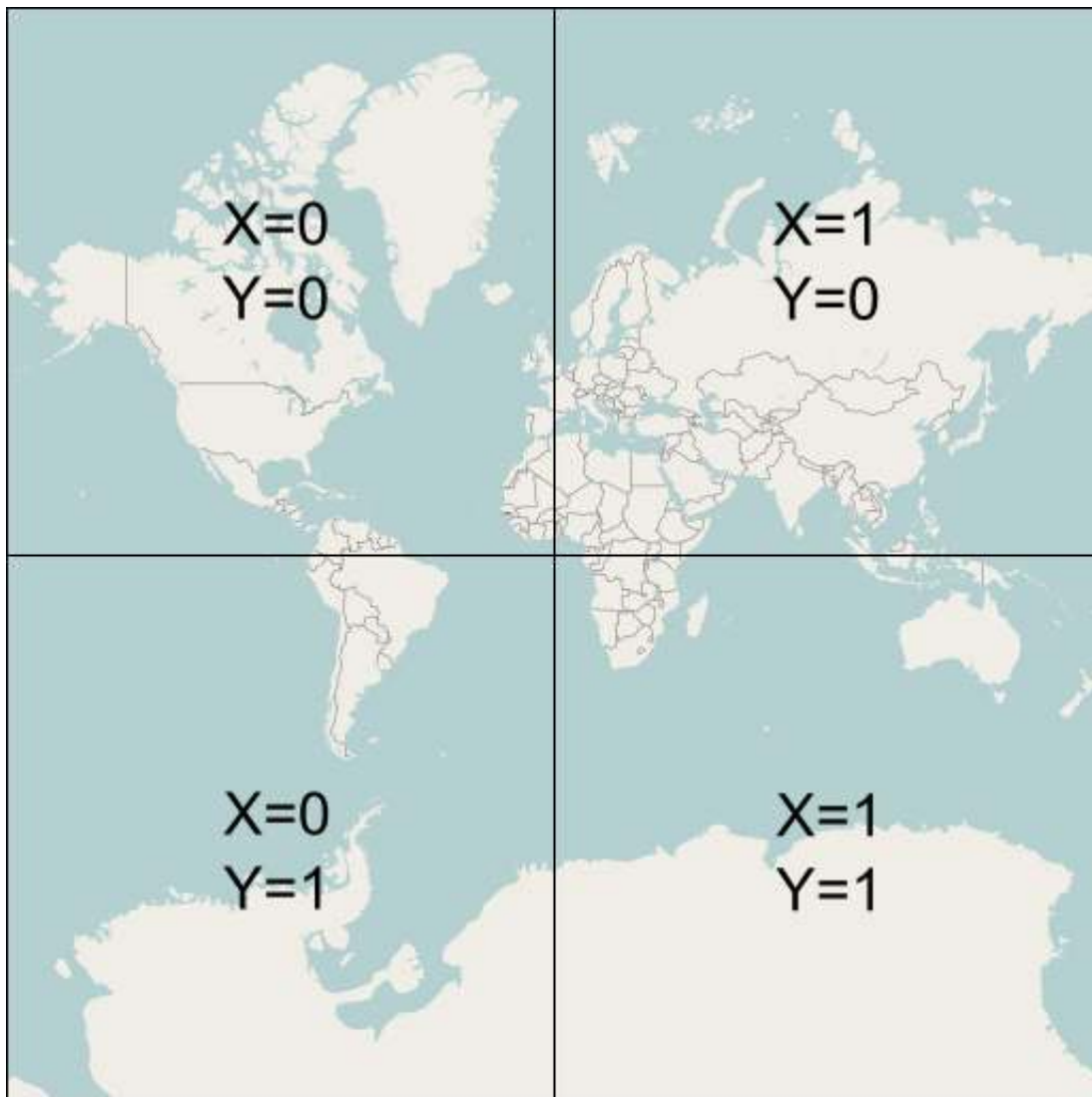
Obrázek 2.2: Osmarender

Mapy v rastrové podobě jsou uloženy na webovém serveru, a jsou přístupné široké veřejnosti. Obrázky jsou rozděleny do adresářů podle přiblížení, x-ové a y-ové souřadnice. Počet dlaždic pro jednotlivá přiblížení popisuje tabulka 2.1.

Zoom	Popis	Počet dlaždic
0	1 dlaždice pokrývá celý svět	1
1	2 × 2 dlaždice	4
2	4 × 4 dlaždice	16
n	$2^n \times 2^n$ dlaždic	2^{2n}
12	4096 × 4096 dlaždice	16.777.216
17	Maximální přiblížení pro vrstvu Omarender	17.179.869.184
18	Maximální přiblížení pro vrstvu Mapnik	68.719.476.736

Tabulka 2.1: Počet dlaždic v daném přiblížení

Dlaždice, z kterých je výsledná mapa tvořena, jsou uloženy jako dvojrozměrné indexované pole. X-ová souřadnice je indexována od 0 což představuje poledník na 180° západní délky do $2^{\text{zoom}} - 1$ (poledník 180° východní délky). Y-ová souřadnice je indexována ve stejném rozmezí od 0 do $2^{\text{zoom}} - 1$, což odpovídá zeměpisným souřadnicím od 85,0511° severní šířky do 85,0511° jižní šířky. Indexování mapových podkladů je naznačeno na obrázku 2.3. [15]



Obrázek 2.3: Indexování mapových dlaždic OSM

2.8.3 Bing Maps

Bing Maps je webová mapová služba, poskytovaná jako součást sady vyhledávačů Bing. Tato služba je vyvíjena firmou Microsoft. Jejím předchůdcem je program Live Maps, který byl s příchodem nového vyhledávače Bing, který postupně sjednotil všechny vyhledávací služby Microsoftu, přejmenován do dnešní podoby.

Společnost Microsoft nastavila službě Bing Maps takové licenční podmínky, které umožňují využívat její mapové podklady ve vlastní aplikaci. Bing mapy využívají, stejně jako OSM, mapové podklady tvořené dlaždicemi o rozměru 256×256 pixelů. Bing Maps dlaždice poskytují mapové podklady ve formě klasické silniční mapy, hybridní foto mapy, nebo ve formě ptačí perspektivy. Pohled z ptačí perspektivy vytvářejí šikmé letecké snímky, které jsou pořízeny z nízké výšky.

K Bing Maps dlaždicím, které jsou uloženy na serveru, se přistupuje podle speciálního jména, které je obrázkem vygenerované podle požadovaného podkladu, polohy a přiblížení dané dlaždice. První znak jména, kterým je soubor pojmenován, je volba mapového podkladu a může nabývat hodnot „r“, „h“ nebo „o“, kde znak „r“ představuje silniční mapu, „h“ hybridní mapu a „o“ pohled

z ptáčí perspektivy. Dalšími znaky jsou čísla od 0 do 3. Číslice představuje kvadrant, ze kterého je zobrazovaná dlaždice vybrána. Počet číslic určuje míru přiblížení mapového podkladu. Princip dělení mapy na příslušné dlaždice je zobrazen na obrázku 2.4. Dlaždici, která je na tomto obrázku červeně zvýrazněná, by identifikoval kód „r12“. [16]



Obrázek 2.4: Princip pojmenování Bing Maps dlaždic

3 Návrh aplikace

3.1 Návrh uživatelského rozhraní

Okno aplikace je pokryto mapovým podkladem, který zobrazuje vybranou část mapy v požadovaném přiblížení. Jedná se o interaktivní plochu, na kterou je možné kliknutím tlačítka myši vložit body, které určují tvar požadované trasy na mapě. Pokud je při kliknutí na bod trasy přidržena klávesa *Ctrl*, je příslušný bod smazán. V okně je umístěno několik komponent, které slouží k zobrazení různých údajů uživateli. Rozložení jednotlivých částí aplikace je zobrazeno na obrázku 3.1.

První komponentou, která uživateli nabízí ovládací prvky je navigační lišta. Je zobrazena ihned po spuštění aplikace a obsahuje tlačítka pro generování výškového profilu trasy, přepnutí na celoobrazovkový režim, import a export XML souborů, změnu měřítka mapy, volbu mapového podkladu a tlačítka pro založení nebo přihlášení se k uživatelskému účtu.

Další komponenty nejsou po spuštění aplikace viditelné. První je okno s výškovým profilem, které se zobrazí po stisknutí příslušného tlačítka na navigační liště. Podmínkou je, že na mapě musí být vytvořeny alespoň dva body, které vytvoří trasu, pro kterou se bude profil generovat. Další komponenty jsou zaměřeny na práci s uživatelským účtem. Pro přihlášení, k účtu, nebo vytvoření účtu nového je zobrazena přihlašovací nabídka, která uživateli umožní vyplnit přihlašovací údaje. Po úspěšném přihlášení je vytvořeno uživatelské menu, které umožňuje ukládání nově vytvořených tras, nebo manipulaci s uloženými položkami.



Obrázek 3.1: Rozložení prvků grafického rozhraní

3.2 Tvorba výškového profilu

K vytvoření výškového profilu jsou využity hodnoty, které byly naměřeny v projektu SRTM. Tyto hodnoty jsou uloženy do souborů o velikost 2,8MB. Každý soubor definuje nadmořskou výšku pro plochu čtverce o rozměrech 1° zeměpisné šířky a 1° zeměpisné délky. Jelikož je tento projekt

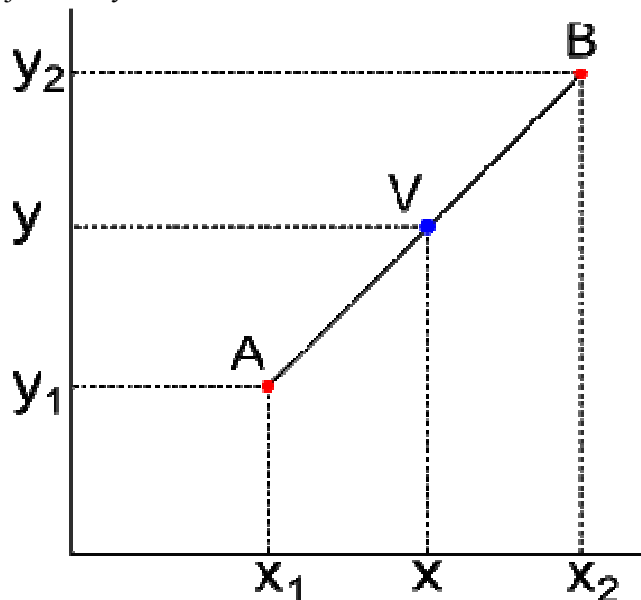
vytvořen jako aplikace v prostředí Flash, která běží na klientském počítači, bylo by nutné kopírovat celý soubor na klientskou stanici. Přenos takto velkého množství dat by uživateli s nižší rychlostí připojení k internetu mohl výrazně snížit rychlost běhu aplikace, nehledě na to, že při výpočtu nadmořské výšky mohou být požadována data z několika takovýchto souborů. Řešením je provést výpočet nadmořské výšky přímo na straně serveru a klientskému počítači poslat pouze požadované hodnoty. Pro tuto operaci je vhodné použít PHP skript.

V případě, že je výpočet prováděn na straně serveru, jsou po síti přenášeny pouze body, které definují trasu, pro kterou je výškový profil počítán, a následně jsou hodnoty nadmořské výšky pro danou trasu zaslány zpět. Podstatně se tak omezí objem přenášovaných dat, která je nutné přenést mezi klientem a serverem.

3.2.1 Prokládání trasy

Pro výpočet nadmořské výšky má skript k dispozici polohu trasy, která je zadána seznamem bodů. Tyto body jsou zaslány z aplikace běžící na klientovi, prostřednictvím protokolu HTTP. Počet souřadnic, které má skript k dispozici, není dostatečný pro určení profilu trasy po celé její délce, proto je nutné navzorkovat trasu dostatečným počtem bodů, aby byla přesnost zobrazených dat co nejlepší. Pro aplikaci byla zvolena hodnota 500 vzorků. Tento počet byl zvolen z toho důvodu, protože okno, ve kterém je zobrazován výškový profil, je široké právě 500 pixelů, a větší počet vzorků by stejně nebyl pro uživatele pozorovatelný.

Vzorkování probíhá tak, že je vypočítán krok na trase, který je úměrný pěti setině celkové délky trasy. Jednotlivé souřadnice, pro které je zjištěna nadmořská výška, jsou vypočítány pomocí vzorců 3.1 a 3.2, které se odkazují na obrázek 3.2. V obrázku představuje vzdálenost $|AV|$ krok vzorkování a vzdálenost bodů A a B reprezentuje celkovou vzdálenost trasy. Hodnoty x a y jsou zeměpisné souřadnice jednotlivých bodů.



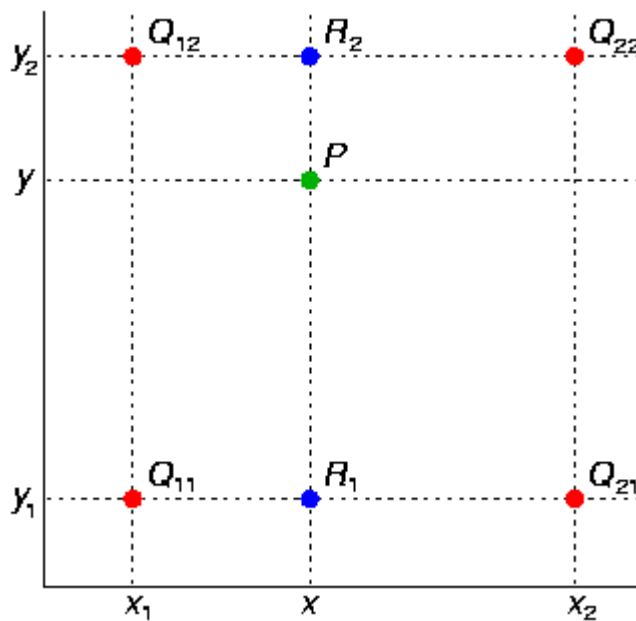
Obrázek 3.2: Výpočet souřadnic bodu

$$x = (x_2 - x_1) \cdot \left(\frac{|AV|}{|AB|} \right) + x_1 \quad (3.1)$$

$$y = (Q_2 - y_1) \cdot \left(\frac{|AV|}{|AB|}\right) + y_1 \quad (3.2)$$

3.2.2 Bilineární interpolace

Zjištění nadmořské výšky probíhá tak, že je podle zadaných souřadnic vybrána hodnota ze SRTM souboru, která definuje, nadmořskou výšku pro daný bod. Nadmořská výška, jejíž hodnoty jsou uloženy v SRTM souborech je vzorkována přibližně po 90 metrech. Pro zpřesnění výsledku se využívá bilineární interpolace, pomocí které je výsledná hodnota spočítána ze čtyř bodů, které jsou nejblíže požadované souřadnici, jak je naznačeno na obrázku 3.3.



Obrázek 3.3: Bilineární interpolace

$$R_1 = \frac{x_2 - x}{x_2 - x_1} \cdot Q_{11} + \frac{x - x_1}{x_2 - x_1} \cdot Q_{21} \quad (3.3)$$

$$R_2 = \frac{x_2 - x}{x_2 - x_1} \cdot Q_{12} + \frac{x - x_1}{x_2 - x_1} \cdot Q_{22} \quad (3.4)$$

$$P = \frac{y_2 - y}{y_2 - y_1} \cdot R_1 + \frac{y - y_1}{y_2 - y_1} \cdot R_2 \quad (3.5)$$

Výpočet výsledné hodnoty nadmořské výšky je proveden pomocí rovnic 3.3 – 3.5, kde Q_{11} , Q_{12} , Q_{21} a Q_{22} představují hodnoty nadmořské výšky, které jsou získány ze SRTM souboru, proměnné x_1 , x_2 , y_1 a y_2 představují zeměpisné souřadnice, pro které jsou hodnoty nadmořské výšky známy a x , y jsou souřadnice bodu, pro který je nadmořská výška interpolována. Proměnná P po provedení těchto výpočtů obsahuje výslednou nadmořskou výšku pro požadovaný bod.

3.2.3 Výpočet vzdálenosti dvou bodů

Pro výpočet vzdálenosti bodů je využit algoritmus *Haversine*. Jedná se o metodu, která počítá vzdálenost bodů ze zadaných zeměpisných souřadnic. Pro určení vzdálenosti se v GPS přístrojích využívá spíše algoritmus *GreatCircle*. Jedná se o rychlejší a výpočetně jednodušší algoritmus, ale při výpočtu malých vzdáleností je zatěžkán chybou, která vzniká zaokrouhlováním [21]. Tyto algoritmy provádí výpočet, jako by se jednalo o výpočet délky trasy na kouli. Země ovšem nemá tvar koule, ale geoidu, který lze jen těžko matematicky vyjádřit. Jako náhrada geoidu se využívá rotačního elipsoidu. Pro výpočet vzdálenosti dvou bodů na rotačním elipsoidu bývá používán algoritmus *Vincenty*, který je schopen určit vzdálenost dvou bodů s přesností 0,5 milimetru v rámci celého elipsoidu.

Algoritmus *Haversine* byl vybrán, protože se jedná v porovnání s *GreatCircle* o přesnější algoritmus. Rozdíl mezi algoritmy *Vincenty* a *Haversine* je maximálně 50m na 100km, což je při výpočtu vzdálenosti prakticky zanedbatelná hodnota [21].

3.3 Ukládání tras

Tento projekt umožňuje export nebo import tras, prostřednictvím KML nebo GPX souborů. Dále je také možné vytvořit si v samotné aplikaci uživatelský účet, prostřednictvím kterého si může uživatel vytvořené trasy uložit přímo v aplikaci a při příští návštěvě aplikace k nim bude mít pohodlný přístup

3.3.1 Export trasy

V případě exportu trasy do XML souborů je nutné nejprve vytvořit příslušný soubor na serveru, který poté může být uživateli odeslán. Pro vytvoření XML souboru je využit PHP skript na straně serveru, kterému je předán seznam zeměpisných souřadnic, které danou trasu definují. Skript z těchto dat vytvoří XML soubor, který uloží na server. Aplikace poté zajistí odeslání vytvořeného souboru.

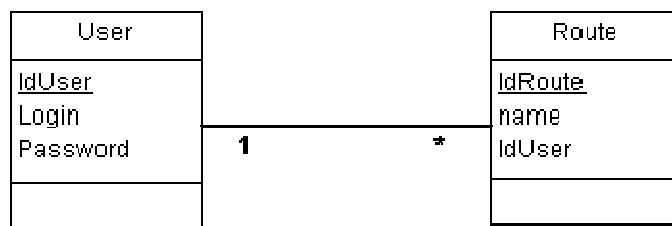
3.3.2 Import trasy

Aplikace umožňuje zobrazení dat, které má uživatel uložené v externím souboru ve formátu KML nebo GPX. Tyto formáty jsou oblíbeným prostředkem pro výměnu dat mezi počítačem a GPS přístroji. Je tedy možné zobrazit trasu, kterou uživatel zaznamenal prostřednictvím svojí navigace.

Aby mohl program externí soubor zpracovat, je nutné jej nejprve uložit na server. Za tímto účelem se uživateli zobrazí Open dialog, prostřednictvím kterého vybere soubor s uloženou trasou v jednom z dvojice formátů GPX a KML. Přijatá data jsou předána PHP skriptu, který zajistí uložení souboru na serverovém disku. Z tohoto souboru jsou získány souřadnice, ze kterých je vygenerovaná příslušná trasa.

3.3.3 Ukládání tras prostřednictvím uživatelských účtů

Uživatelské účty jsou ukládány do databáze MySQL. Za tímto účelem byla vytvořena jednoduchá databáze, jejíž relační model je zobrazen na obrázku 3.4. Tabulka *User* uchovává přihlašovací jméno a heslo uživatele. Druhá tabulka je nazvaná *Route*. Ukládá název trasy, kterou si uživatel uložil a cizí klíč *IdUser*, který identifikuje autora této trasy. Tabulky jsou v relaci 1:N, kde jeden uživatel může mít N uložených tras. Databáze tedy uchovává pouze přihlašovací údaje uživatele a název trasy.



Obrázek 3.4: Relační model – Ukládání tras

Samotná data o vytvořených trasách se uchovávají na serveru v adresářové struktuře, kde každému uživatelskému účtu odpovídá jeden adresář, který je pojmenovaný podle přihlašovacího jména uživatele. Do těchto adresářů jsou uživatelům ukládány jednotlivé trasy prostřednictvím souborů ve formátu GPX.

Pokud program vyžaduje vytvoření uživatelských účtů, přihlašování se k nim, nebo jedná-li se o manipulaci s vytvořenými trasami, je nutné využít PHP skript, který zde představuje komunikační rozhraní mezi aplikací běžící v prostředí Flash a databází. Prostřednictvím PHP skriptu je možné přidávat údaje do databáze, ověřovat přihlašovací údaje uživatele, nebo vypsát trasy, které si daný uživatel vytvořil.

4 Implementace

Tato kapitola se zabývá implementací projektu. Jsou zde popsány jednotlivé třídy a funkce, které tvoří výslednou aplikaci. Aplikace byla vytvořena v prostředí jazyka Haxe. Jedná se o open source programovací jazyk, pomocí kterého je možné vytvořit aplikace pracující na různých platformách.

4.1 Objektový model

Základní třída aplikace, která obsahuje funkci *main()* se nazývá *Maps*. Funkce *main()* je v aplikacích vytvořených v jazyce Haxe vyžadována a plní funkci vstupního bodu do aplikace. Ve funkci *main* je vytvořen objekt třídy *Maps*. V této třídě jsou vytvořeny instance tříd, které jsou součástí knihovny HaxeMaps. Jedná se o objekty *ToolBar*, *Canvas*, *TileLayer* a *InteractiveLayer*.

Objekt třídy *ToolBar* představuje navigační lištu na horním okraji obrazovky, na kterou jsou vkládány řídicí prvky aplikace. Třída *TileLayer* je odvozena z třídy *Layer* a jedná se o vrstvu, pomocí které se zobrazí mapové podklady aplikace. *InteractiveLayer* je třída, která je využívána pro zobrazení bodů a trasy na mapě. *Canvas* představuje plátno, na něž jsou vkládány jednotlivé vrstvy typu *Layer*, které jsou pomocí této třídy zobrazeny.

V konstruktoru třídy *Maps* je kromě vytvoření instancí právě popsaných tříd provedeno několik dalších kroků. Nejprve je provedena inicializace navigační lišty v metodě *initToolBar()*, což představuje vytvoření tlačítek, které umožní využívat další funkce aplikace. Po inicializaci *ToolBaru* jsou vytvořeny objekty *TileLayer*, které představují jednotlivé mapové podklady. Pomocí metody *addLayer()* jsou mapové podklady předány objektu typu *Canvas*, který umožní jejich zobrazení, přičemž aktivní je pouze jeden z nich. Volbu mapového podkladu je možné provést prostřednictvím tlačítek na navigační liště. Objekty *ToolBar* a *Canvas* jsou nakonec pomocí metody *addChild()* uloženy do kontejneru pro zobrazení. Jedná se o kontejner, do kterého jsou vkládány grafické objekty, které tvoří grafické rozhraní aplikace.

4.2 Navigační lišta

Navigační lišta je tvořena objektem typu *ToolBar*. Tento objekt je součástí knihovny HaxeMaps, ovšem pro využití v tomto projektu byl rozšířen o některé funkce. Mezi původní metody, které jsou v projektu využity, patří *addButton()* pro přidávání tlačítek na lištu, *setText()* k zobrazení textu a metoda *addSeparator()*, která se využívá jako oddělovač mezi skupinou tlačítek. Přidané funkce jsou *disableButton()* a *enableButton()*, jejichž úkolem je skrývání a zobrazování tlačítek.

4.3 Výškový profil

Pro vytvoření výškového profilu je na navigační liště vytvořeno tlačítko *GraphButton*. Po stisknutí tohoto tlačítka je zavolána funkce *createEW()*, která se nachází ve třídě *Maps*. V této funkci je vytvořena instance třídy *ElevationWindow*. Jedná se o třídu, která umožňuje vytvářet grafické objekty. Obsahuje metodu *drawGraph()*, která provede volání skriptu *getElevation.php*, jenž zajistí hodnoty nadmořské výšky pro danou trasu. Seznam bodů, které tuto trasu definují, je skriptu odeslán prostřednictvím protokolu HTTP pomocí metody POST. Následně je nastaven posluchač události *Event.Complete*, která signalizuje, že jsou přijata výsledná data od skriptu. Když tato událost nastane,

je zavolána funkce *loadEValues()*, jejíž úkol je zpracovat přijaté hodnoty, které jsou odděleny dvojtečkou. Hodnoty jsou rozděleny pomocí funkce *split(„:“)* a uloženy do pole *elevations*. Následně je zavolána funkce *paint()*, která vytvoří objekt *eCanvas*, na který je vykreslen graf výškového profilu. Pokud je nad objektem *eCanvas*, který představuje graf s výškovým profilem, zachycena událost pohybu myši, je zavolána metoda *mouseMove()*, která zobrazí nad kurzorem hodnotu nadmořské výšky v daném bodě. Kliknutím na objekt *eCanvas* se provede funkce *mouseClick()*. Její činnost spočívá ve výpočtu místa na trase, podle polohy kurzoru v grafu. Dále je pomocí události zavolána metoda *routeFocus()*, která na trase vyznačí pomocí objektu typu *Shape* místo, odpovídající poloze kurzoru na grafu.

4.4 Export a import XML souborů

Pro práci s externími soubory jsou využívány funkce *exportXML()* a *importXML()*, které jsou implementovány ve třídě *Maps*. Metoda *exportXML()* nejprve zavolá prostřednictvím objektu typu *InteractiveLayer* funkci *saveRoute()*. Tato funkce zajistí provedení skriptu *saveXML.php*, jemuž jsou předány parametry *fileName*, který určuje jméno výstupního souboru a *xmlType*, jehož obsah určuje, zda se má vytvořit soubor ve formátu GPX, nebo KML. Dalšími parametry jsou jednotlivé souřadnice bodů trasy a jejich počet, který reprezentuje parametr *length*. Ve skriptu je pomocí funkce *fopen()* vytvořen soubor, do kterého jsou uloženy jednotlivé souřadnice trasy ve formátu KML nebo GPX. Po dokončení skriptu je ve funkci *exportXML()* vytvořen objekt třídy *FileReference*. Prostřednictvím jeho funkce *download()* je zobrazeno dialogové okno, které umožní uživateli stáhnout soubor ze vzdáleného serveru.

Načítání dat z externího souboru zajišťuje funkce *importXML()*. V této funkci je vytvořena instance třídy *FileReference*, jejímž prostřednictvím je volána funkce *browser()*, která zobrazí dialogové okno, pomocí něhož je možné vybrat soubor pro odeslání. Jakmile je soubor vybrán, je prostřednictvím události *Event.SELECT* zavolána funkce *importSelect()*, která zajistí provedení skriptu *uploadFile.php*, jehož úkolem je přesunout načítaný soubor z dočasné složky do adresáře *XMLfiles*. Po uložení uploadovaného souboru na serveru je volanou funkcí *loadRoute()*, která tento soubor načte prostřednictvím skriptu *loadXML.php*. Načtený XML soubor je zpracován ve funkci *completeLoadKML()* nebo *completeLoadGPX()*, podle toho o jaký typ souboru se jedná. Pro práci s daty ve formátu XML je využita třída *XML*, pomocí níž jsou získány jednotlivé souřadnice trasy. Tyto body jsou funkcí *addMark()* přidávány do pole *marks_lng* a *marks_lat*. Po načtení všech bodů je trasa prostřednictvím funkce *updateContent()* zobrazena na mapě.

4.5 Práce s uživatelskými účty

Registrace uživatelů je provedena prostřednictvím funkce *createReg()*. V této funkci je vytvořen grafický objekt typu *Login*, který představuje formulář pro registraci. Po odeslání formuláře jsou přihlašovací údaje nového uživatele předány skriptu *registration.php*. Pokud není zadáno přihlašovací jméno zatím v databázi uloženo, je nový uživatel do databáze zaregistrován. Následně jsou uživateli prostřednictvím funkce *mail()* zaslány přihlašovací údaje na e-mail, který byl při registraci zadán.

Pro přihlášení uživatele je zavolána funkce *createLogin()*, která rovněž vytvoří instanci třídy *Login*, pomocí níž jsou zadány přihlašovací údaje uživatele. Odesláním formuláře je zavolán skript *login.php*, který v databázi ověří platnost zadaného přihlašovacího jména a hesla. Prostřednictvím

skriptu je následně vrácena hodnota *true*, která signalizuje správnost přihlašovacích údajů, nebo hodnota *false*, která značí, že uživatelem zadané údaje nejsou správné. Odchytávání návratové hodnoty skriptu je prováděno funkcí *loadVerifyLogin()*, která zajistí volání funkce *createMenu()*, pokud byla návratová hodnota skriptu *true*.

Funkce *createMenu()* zajistí, že je vytvořen objekt typu *Menu*, který představuje uživatelské menu, pomocí kterého je možné ukládat vytvořené trasy. Tato třída využívá k manipulaci s trasami skript *routeOperation.php*. V tomto skriptu jsou implementovány funkce *loadRoutes()*, která má za úkol načíst všechny trasy, které má uživatel uložené v databázi, *addRoute()* implementuje přidávání tras do databáze a funkce *removeRoute()* umožňuje trasu z databáze vymazat. Volba operace, která se má pomocí skriptu provést, je zajištěna prostřednictvím parametru *command*, který je skriptu předán. Pokud je na server ukládána nová trasa, je nejprve uložen do databáze její název, a následně je pomocí funkce *saveRoute()* uložena na server do adresáře daného uživatele. K načítání trasy je využita funkce *loadRoute()*. Obě tyto funkce jsou popsány v kapitole 4.4, kde jsou využívány k exportu a importu souborů.

5 Testování

Projekt se zabývá zobrazováním a vytvářením tras na mapě, generováním výškového profilu a výpočtem délky trasy. Experimentování se tedy bude zabývat, přesností zobrazovaného výškového profilu a délky trasy. Cílem je tedy zjistit, zda jsou hodnoty vypočítané prostřednictvím tohoto projektu reálné.

Pro testování jsou využita data ze serveru *obec.cr* [23], na kterém jsou uloženy polohy jednotlivých měst a obcí a jejich nadmořská výška. Pro porovnávání délky trasy a výškového profilu je využit server *cykloserver.cz*, který umožňuje načítat trasy prostřednictvím formátu GPX a aplikace Google Earth, již je testovaná trasa předávána prostřednictvím formátu KML.

5.1 Testování přesnosti nadmořské výšky

Testování nadmořské výšky bylo provedeno za pomoci hodnot ze serveru *obec.cr*. Z tohoto serveru byly zjištěny geografické souřadnice města Třebíč a obce Výčapy, které byly využity jako koncové body trasy. Následně byl vygenerován výškový profil této trasy, a byly odečteny hodnoty nadmořské výšky koncových bodů. Výsledky testování shrnuje tabulka 5.1.

Obec	Zeměpisná šířka	Zeměpisná délka	NV <i>obec.cr</i>	NV projekt
Třebíč	49.21305°	15.87833°	405 m n. m.	401 m n. m.
Výčapy	49.14305°	15.87944°	478 m n. m.	473 m n. m.

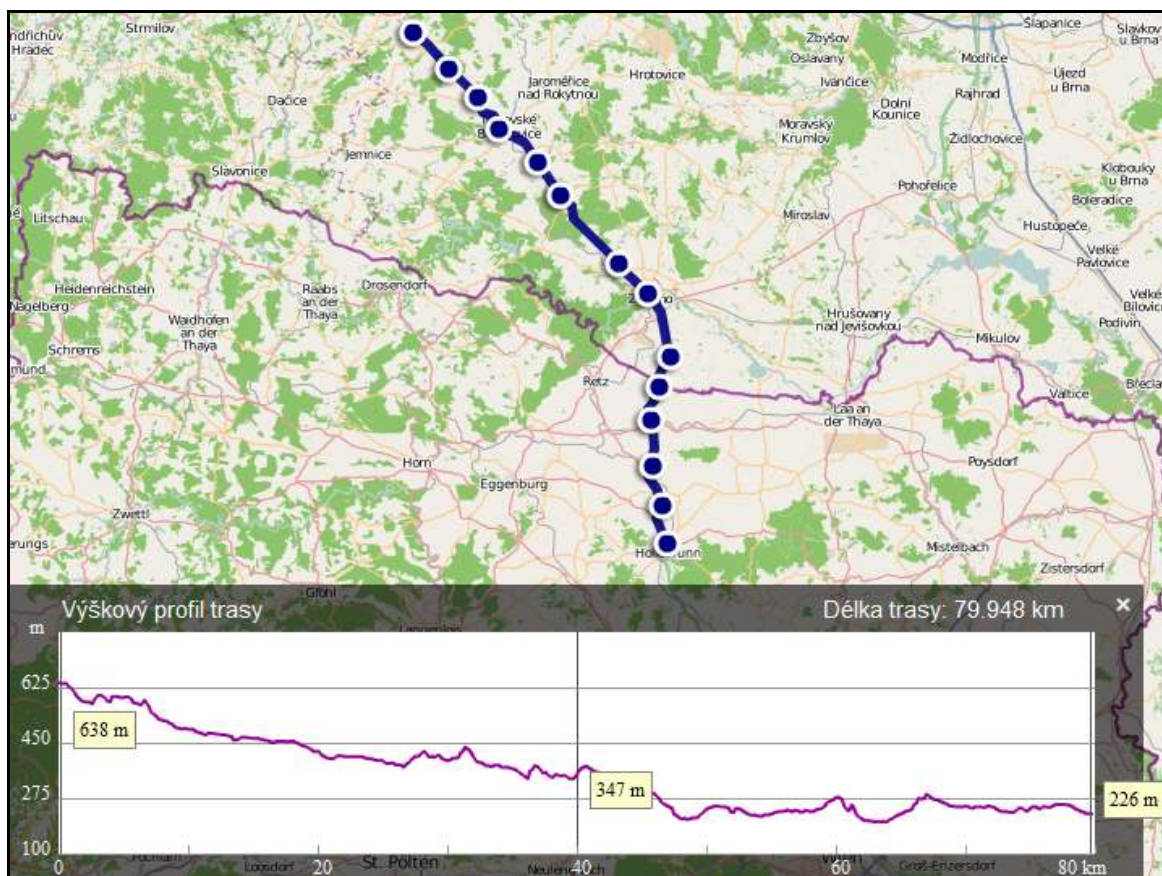
Tabulka 5.1: Porovnání nadmořských výšek

Tabulka 5.1 zobrazuje geografické souřadnice pro jednotlivé obce. Sloupec *NV obec.cr* obsahuje nadmořskou výšku, která byla zjištěna ze serveru *obec.cr* a ve sloupci *NV projekt* jsou hodnoty získané z tohoto projektu. Z tabulky je vidět, že se hodnoty liší v řádu metrů, což může být způsobeno tím, že jsou hodnoty nadmořské výšky v SRTM souborech vzorkovány po 90 metrech. I přesto, že je výsledná nadmořská výška počítána pomocí bipolární interpolace ze čtyř okolních hodnot, může být měření jednotlivých bodů zatěžkáno chybou.

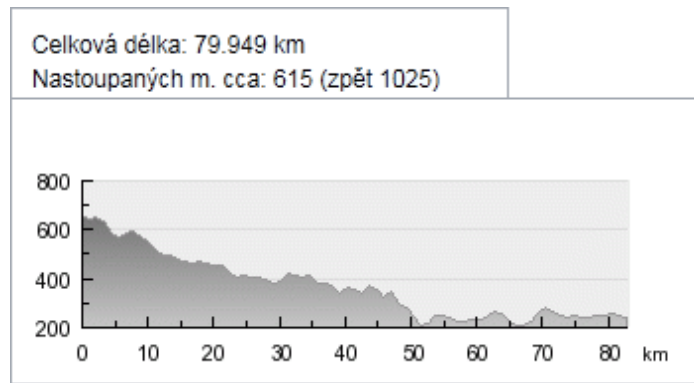
5.2 Testování vytvořené trasy

Pro druhý test byla vytvořena jednoduchá trasa, která je složena z 36 bodů. Jako zdroj referenčních dat byl použit server *cykloserver.cz*, který umožňuje tvorbu tras, vytvoření jejího výškového profilu a importování a exportování trasy ve formátu GPX. Toto testování je zaměřeno na ověření správnosti výpočtu délky trasy a na porovnání výškového profilu se severem *cykloserver.cz*. Data na tento server byla přenesena prostřednictvím souboru ve formátu GPX.

Testování bylo nejprve zaměřeno na ověření správnosti celkové délky trasy. Hodnota vypočtená pomocí tohoto projektu byla 79,948 km. *Cykloserver.cz* udává hodnotu 79,949 km, což znamená, že se výsledná hodnota liší pouze o jeden metr. Dále bylo testování zaměřeno na ověření správnosti výsledného výškového profilu. Na *cykloserveru* není bohužel možné zobrazit konkrétní hodnotu nadmořské výšky na daném kilometru. Nezbyvá než vystačit si s vygenerovaným grafem. Na obrázku 5.1 je zobrazena trasa, pro kterou je toto testování prováděno, a její výškový profil vygenerovaný v tomto projektu. V grafu výškového profilu jsou zobrazeny hodnoty nadmořské výšky. Pokud tyto hodnoty porovnáme s výsledným grafem, který je na obrázku 5.2, vytvořeným na *cykloserveru* zjistíme, že hodnoty nadmořské výšky jsou velmi podobné.



Obrázek 5.1: Zobrazení trasy a výškového profilu



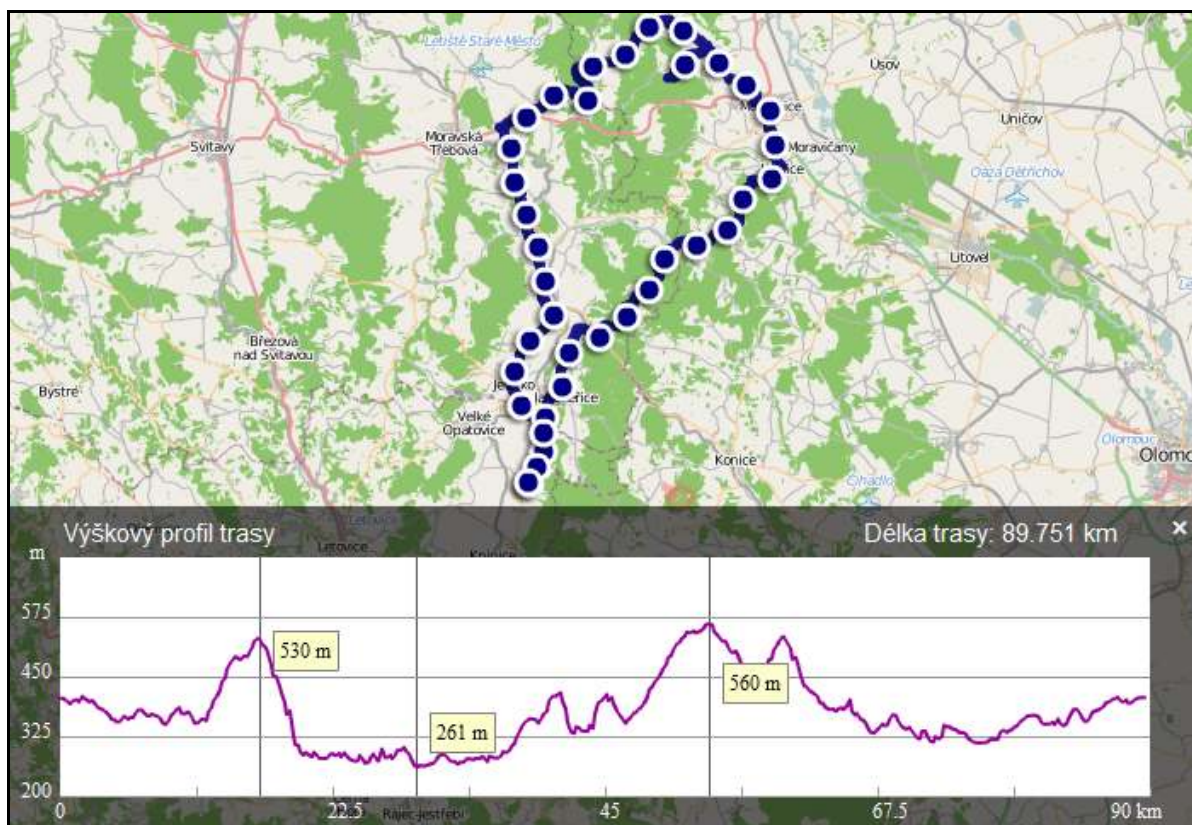
Obrázek 5.2: Výškový profil trasy vytvořený na serveru *cykloserver.cz*

5.3 Testování trasy z GPS přístroje

Data pro třetí test jsou získána z GPS přístroje. Jedná se o trasu, která je tvořena 3266 body. Pro porovnání výsledků byla použita aplikace Google Earth. Trasa, která byla předmětem testování, byla z GPS exportována ve formátu KML. Porovnání hodnot prostřednictvím serveru *Cykloserver.cz* nebylo možné, neboť umožňuje načtení pouze 800 hodnot.

Testování bylo opět zaměřeno na ověření délky trasy a správnosti zobrazení výškového profilu. Délka trasy byla v aplikaci Google Earth vyčíslena na hodnotu 89,912 kilometru a v tomto projektu na 89,751. Rozdíl mezi těmito hodnotami je 161 metrů, což není při daném počtu bodů a délky trasy zas tak velký rozdíl.

Při zobrazení výškového profilu v aplikaci Google Earth, byla tímto programem zobrazena jiná vzdálenost, než jaká byla udávána ve vlastnostech trasy. Není proto možné porovnat, jestli je hodnota dané nadmořské výšky ve správné části trasy. Pomocí grafu výškového profilu z aplikace Google Earth je ovšem možné ověřit správnost hodnoty nadmořské výšky v extrémních bodech. Porovnání vybraných bodů je zobrazeno na obrázku 5.3, kde je zobrazen výškový profil vytvořený v tomto projektu, a na obrázku 5.4, který zobrazuje graf vygenerovaný v Google Earth. Z obrázků je patrné, že obě aplikace udávají v těchto bodech totožné hodnoty.



Obrázek 5.3: Zobrazení trasy a výškového profilu



Obrázek 5.4: Výškový profil trasy vytvořený na aplikaci Google Earth

5.4 Zhodnocení testování

Testováním bylo zjištěno, že aplikace vytvořená v tomto projektu umožňuje zobrazit poměrně přesné údaje o výškovém profilu vytvořené trasy a vypočítat reálnou hodnotu délky trasy. Testováním nebylo ověřeno jen generování výškového profilu a výpočet trasy, ale také načítání a generování externích souborů, které byly mezi aplikacemi vyměňovány. Bylo také ukázáno, že je pomocí aplikace možné načíst trasy, které byly vytvořeny pomocí GPS přístroje. Aplikace, pomocí které byly testy prováděny je dostupná z webové stránky <http://www.stud.fit.vutbr.cz/~xjonas04/ibp/>. Trasy, které byly při testování využity, je možné zobrazit prostřednictvím uživatelského účtu s přihlašovacím jménem *student* a heslem *aaa*.

6 Závěr

Cílem práce bylo seznámit se s tvorbou mapových aplikací a běžnými formáty pro výměnu mapových dat, jako jsou KML a GPX. Za pomoci těchto poznatků navrhnout a naimplementovat aplikaci, která by umožňovala zobrazení mapy a vytváření vlastní trasy. Pro trasu je generován výškový profil. Součástí implementace je výměna externích souborů s uloženou trasou. Je možné vytvářet uživatelské účty a udržování seznamu tras.

Práce splnila všechny body zadání. Ve druhé kapitole je nastíněna problematika tvorby mapových aplikací pomocí knihovny HaxeMaps a využití volně dostupných mapových podkladů. Je zde také pojednáno o souborech KML a GPX. Třetí kapitola se zabývá návrhem funkcí aplikace a uživatelského rozhraní. Ve čtvrté kapitole je naznačeno, jak byla aplikace implementována. Pátá se zabývá testováním činnosti aplikace a zhodnocením.

Prostřednictvím této práce jsem si osvojil práci s mapovými aplikacemi a seznámil jsem se s možnostmi, jak takovou aplikaci vytvořit. Díky projektu OpenStreetMap jsem se dozvěděl, jakými způsoby je možné vytvářet mapové podklady pro tvorbu mapových aplikací. Také jsem si osvojil práci s open source jazykem Haxe, s kterým jsem dosud neměl žádné zkušenosti.

Do budoucna by bylo vhodné aplikaci rozšířit o možnost zobrazovat více tras současně a následně takto vytvořené trasy uložit do externích souborů. Aplikace by se také v blízké době měla dočkat rozšíření, které by umožňovalo vyhledat nekratší možnou trasu z jednoho místa do druhého. Tato trasa by měla vést po silnicích a cestách, které jsou na mapě zobrazeny.

Literatura

- [1] HONZÍK, M.: Úvod do platformy Adobe Flash [online]. Vydáno 6.2.2011 [citováno 2011-04-20]. Dostupné z: <http://flash.jakpsatweb.cz/adobe-flash/>.
- [2] FLORES, A.: Adobe AIR: Browser vs. application [online]. 3/2011 [citováno 2011-04-18]. Dostupné z: <http://www.adobe.com/products/air/comparison/>.
- [3] ADOBE CREATIVE TEAM. Programování v jazyce Adobe ActionScript 3.0. San Jose: Adobe System Incorporated, 2008.
- [4] ZAJÍC, P.: PHP – Historie a budoucnost [online]. Vydáno 27.5.2004 [citováno 2011-04-23]. Dostupné z: http://www.linuxsoft.cz/article.php?id_article=171.
- [5] HRUŠKA, T., BURGET, R.: Internetové aplikace (WAP) IV., část Programování serveru (PHP). Brno: FIT VUT v Brně, 2007.
- [6] KOCAN, M.: S licencemi opatrně [online]. Vydáno 12.6.2002 [citováno 2011-04-24]. Dostupné z: <http://www.dbsvet.cz/view.php?cislocianku=2002061205>.
- [7] ZAJÍC, P.: MySQL – pestrý svět databází [online]. Vydáno 1.3.2005 [citováno 2011-04-24]. Dostupné z: http://www.linuxsoft.cz/article.php?id_article=731.
- [8] PONTICELLI, F., McCOLL-SYLVESTER, L.: Professional haXe and Neko. 1. vyd. Indianapolis, 2008. 649 s. ISBN: 978-0-470-122135-6.
- [9] CANNASSE, N.: haXe 2.07 Released! [online]. Vydáno 30.1.2011 [citováno 2011-04-24]. Dostupné z: http://ncannasse.fr/blog/haxe_207.
- [10] VAŠÍČEK, Z.: HaxeMaps [online]. c2010 [citováno 2011-04-27]. Dostupné z: http://www.fit.vutbr.cz/~vasicek/docs/map_index.htm.
- [11] FARR, T. G., KOBRICK, M.: Shuttle Radar Topography Mission produces a wealth of data. Pasadena: California institute of Technology, 2000.
- [12] KML Documentation Introduction: KML Reference [online]. c2011 [citováno 2011-05-03]. Dostupné z: <http://code.google.com/intl/cs/apis/kml/documentation/kmlreference.html>.
- [13] KML Documentation Introduction: KMZ Files [online]. c2011 [citováno 2011-05-03]. Dostupné z: <http://code.google.com/intl/cs/apis/kml/documentation/kmzarchives.html>.
- [14] GPX 1.1 Schema Documentation [online]. 19.7.2007 [citováno 2011-05-03]. Dostupné z: <http://www.topografix.com/gpx/1/1/>.
- [15] RATHOUSKÝ, J.: OpenStreetMap – svobodné mapy světa [online]. Vydáno 11.3.2009 [citováno 2011-05-04]. Dostupné z: <http://www.abclinuxu.cz/clanky/ruzne/openstreetmap-svobodne-mapy-sveta>.

- [16] McGovern, R.: Building Your Own Tile Server [online]. c2011 [citováno 2011-05-03]. Dostupné z: <http://msdn.microsoft.com/en-us/library/bb545006.aspx>.
- [17] Jenny, B., Weber, A.: Manual-OpenStreetMap [online]. Vydáno 5.4.2011 [citováno 2011-05-09]. Dostupné z: <http://mapanalyst.cartography.ch/man/openstreetmap.html>.
- [18] Schwartz, J.: Bing Maps Tile System [online]. c2011 [citováno 2011-05-09]. Dostupné z: <http://msdn.microsoft.com/en-us/library/bb259689.aspx>.
- [19] Headley, J.: Why does Google maps use the inaccurate, ancient and distorted Mercator projection? [online]. Vydáno 8.5.2009. [citováno 2011-05-09]. Dostupné z: <http://www.google.com/support/forum/p/maps/thread?tid=075eb10962e00cc5&hl=en>.
- [20] Google earth [online]. Vydáno 20.3.2011. [citováno 2011-05-03]. Co je to mapová projekce. Dostupné z: <http://earth.google.com/support/bin/answer.py?hl=cs&answer=148111>.
- [21] Path.cz [online]. Vydáno 10.3.2009. [citováno 2011-05-03]. Jak Garmin počítá přímou vzdálenost mezi body. Dostupné z: <http://www.path.cz/forum/viewtopic.php?f=20&t=602&start=0>.
- [22] Obec.cr [online]. c2011 [cit. 2011-05-14]. Obec Výčapy. Dostupné z: <http://www.obec.cr/vycapy/591939/>.
- [23] Obec.cr [online]. c2011 [cit. 2011-05-14]. Město Třebíč. Dostupné z: <http://www.obec.cr/trebic/590266/>.

Seznam příloh

Příloha A: Obsah přiloženého CD

A Obsah příloženého CD

Zdrojové kódy aplikace, technická zpráva, manuál, instalační soubor překladače Haxe. Podrobný popis je v souboru README, který se nachází v kořenovém adresáři CD.