

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SKRÝŠOVEC - SOFTWARE PRO PODPORU GEOCACHINGU

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ANTONÍN KYSELÁK

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SKRÝŠOVEC - SOFTWARE PRO PODPORU GEOCACHINGU

SKRYSOVEC - SOFTWARE FOR GEOCACHING SUPPORT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

VEDOUCÍ PRÁCE

SUPERVISOR

ANTONÍN KYSELÁK

Ing. IGOR SZÖKE

BRNO 2008

Abstrakt

Práce se zabývá analýzou požadavků, návrhem a implementací softwaru pro podporu geocachingu. Zkoumá aplikace podobného typu a snaží se navrhnout podobnou, ale multiplatformní alternativu k těmto zavedeným nástrojům. Aplikace je využitelná pro běžné uživatele, hráče geocachingu, kteří chtějí mít data stále u sebe a mít možnost s nimi pracovat. Aplikace je do budoucna jednoduše rozšiřitelná a navržena tak, aby pokryla širokou oblast využití, díky její multiplatformnosti a multilingualitě.

Klíčová slova

GPS, geocaching, wxwidgets, návrh, implementace, objektově orientované programování, OOP, GPX, LOC, XML, GUI

Abstract

This work is focused on requirements analysis, design and implementation of the software for geocaching support. It studies applications of same type and tries to design similar, but multiplatform alternative to these used tools. Application is useful for common users, the geocaching players, which want to have data always with themselves and to be able to work with them. Application is simply extensible and designed to spread over wide sphere of use, thanks to its multiplatform and multilingual feature.

Keywords

GPS, geocaching, wxwidgets, design, implementation, object oriented programming, OOP, GPX, LOC, XML, GUI

Citace

Antonín Kyselák: Skrýšovec - software pro podporu geocachingu, bakalářská práce, Brno, FIT VUT v Brně, 2008

Skrýšovec - software pro podporu geocachingu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Igora Szöke

.....
Antonín Kyselák
14. května 2008

Poděkování

Zde bych rád využil příležitosti k poděkování mému vedoucímu práce, panu Ing. Igorovi Szöke, za vedení a rady k této práci, dále také panu Ing. Adamu Heroutovi Ph.D. za drobné konzultace.

© Antonín Kyselák, 2008.

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.

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2007/2008

Zadání bakalářské práce

Řešitel: **Kyselák Antonín**

Obor: Informační technologie

Téma: **Skrýšovec - software pro podporu geocachingu**

Kategorie: Softwarové inženýrství

Pokyny:

1. Seznamte se s dostupými aplikacemi (komerčními i nekomerčními) pro podporu geocachingu (např. Geoget a GSAK).
2. Navrhněte aplikaci s podobnými vlastnostmi. Zaměřte se na multiplatformnost. Součástí aplikace by měla být i podpora unknown-cache.
3. Implementujte aplikaci.
4. Otestujte funkčnost.

Literatura:

- Dle pokynů školitele.

Při obhajobě semestrální části projektu je požadováno:

- Obhajoba po prvním semestru body 1 a 2.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

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

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


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

Vedoucí: **Szőke Igor, Ing.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2007

Datum odevzdání: 14. května 2008

L.S.



doc. Dr. Ing. Pavel Zemčík
vedoucí ústavu

**LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Antonín Kyselák**
Id studenta: 78895
Bytem: Biskupská 266/6, 602 00 Brno
Narozen: 09. 04. 1986, Brno
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

**Článek 1
Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: Skryšovec - software pro podporu geocachingu
Vedoucí/školitel VŠKP: Szöke Igor, Ing.
Ústav: Ústav počítačové grafiky a multimédií
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě počet exemplářů: 1
elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užit, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.


Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....
Nabyvatel


.....
Autor

Obsah

1	Úvod	3
1.1	Geocaching	3
1.2	Popis jednotlivých kapitol	4
2	Současný stav	5
2.1	Aplikace pro práci s GPS daty	5
2.2	Aplikace pro podporu geocachingu	6
2.2.1	GSAK	6
2.2.2	GeoGet	7
2.2.3	Skrýšotlač	7
2.3	Zhodnocení a inspirace	8
3	Požadavky a cíle	9
3.1	Požadavky z pohledu uživatele	10
3.1.1	Nejčastější problém	10
3.1.2	Mystery, Multi cache	10
3.1.3	Prohlížení informací	10
3.1.4	Aktuálnost dat	10
3.1.5	Cesta	10
3.1.6	Výlet v terénu	10
3.1.7	Domácí příprava	11
3.1.8	GUI, rozhraní	11
3.1.9	Shrnutí	11
3.2	Požadavky z pohledu vývojáře	11
3.2.1	Import/export	12
3.2.2	Uložení dat	12
3.2.3	Výpočty souřadnic	13
3.2.4	GUI	13
3.2.5	Strukturovaný popis cache	13
3.2.6	Prostředky řízení aplikace	13
3.2.7	Shrnutí	14
4	Návrh aplikace	15
4.1	Systém jako celek	15
4.2	Popis jednotlivých modulů	16
4.2.1	Modul databáze – DB	16
4.2.2	Modul Import/Export – I/O	17
4.2.3	Modul Souřadnice WGS-84	18

4.2.4	Modul Grafické rozhraní – GUI	18
4.2.5	Modul Handler	19
4.3	Návrh databáze	19
4.3.1	ER diagram	19
4.3.2	Popis diagramu	20
4.4	Návrh grafického rozhraní	20
4.4.1	Struktura GUI	21
4.4.2	Popis struktury GUI	21
5	Implementace	23
5.1	Výběr implementačních nástrojů	23
5.1.1	wxWidgets	23
5.1.2	SQLite	23
5.1.3	WGS-84	24
5.1.4	Zdroje	24
5.2	Inicializace aplikace	24
5.2.1	Inicializace databáze	24
5.2.2	Inicializace GUI	25
5.2.3	Inicializace seznamu	25
5.3	Import, Export	25
5.3.1	Import cache	25
5.3.2	Export cache	25
5.3.3	Vlákna	26
5.4	Implementace seznamu	26
5.4.1	Vyhledání cache podle atributů	26
5.4.2	Vyhledání cache podél cesty.	26
5.5	Okno podrobností	26
6	Závěr	27

Kapitola 1

Úvod

Doba technických vynálezů, elektronických vymožeností všeho druhu a hravost lidí dala vznikat stále více populární turistické hře - **geocachingu** (1.1), jejíž centrum se nachází na globálním serveru www.geocaching.com.

To má zásadní výhody i nevýhody. Vzdálený centralizovaný přístup přes prohlížeč má samozřejmě výhodu především v nezávislosti na platformě, nízké datové vytíženosti uživatelů a vždy aktuálním obsahu. Nevýhodou je ale např. nemožnost držet data bez přístupu na internet nebo omezení daná rozhraním webové aplikace.

Toto je také motivací vzniku této práce, která má za úkol vytvořit multiplatformní aplikaci s prostředky pro import dat, jejich perzistentní uložení, následnou filtraci a práci s nimi. Aplikace podobného typu již samozřejmě existují, nabízí ale většinou jen nativní implementaci na MS Windows, což například uživatele linuxu staví do nevýhody.

1.1 Geocaching

Geocaching je relativně mladá celosvětově rozšířená hra, která se zakládá především na turistice, ale hlavně na technologii Global Positioning System (GPS), navigační systém vyvinutý Spojenými státy. Tento systém byl až do 1. května 2000 běžným uživatelům nedostupný, neboť obsahoval záměrnou odchylku kvůli ochraně např. před balistickými střelami. Po tomto datu byla díky objevení možnosti lokálního rušení signálu odchylka odstraněna a GPS začal být okamžitě využíván veřejností především jako námořní, automobilová, či turistická navigace.

Bezprostředně s tím také začal vznikat geocaching. Hra založená na uschování schránky (*cache*), představující *poklad*, jejím následným zaměřením a nakonec především nalezením ostatními hráči podle předložených souřadnic. Pokladem bývá často hlavně místo uschování, na které má schránka upozornit. Některé schránky vyžadují také delší putování, či požadavky na myšlení v podobě různých šifer, nebo jiných úkolů (typy schránek [4]). Ukázka schránky: 1.1.

Geocaching trefně popsal krátkým podtitulem Marek Lutonský ve svém článku *Geocaching: hra pro mozek, nohy a vaši GPS* [5], zde je také k nalezení více informací včetně obrazového materiálu.



Obrázek 1.1: Ukázka schránky (cache), zdroj: Wikipedia [7]

1.2 Popis jednotlivých kapitol

Kapitola 2 obsahuje popis nástrojů pro práci s GPS daty obecně a dále popis a srovnání několika nejpoužívanějších nástrojů pro podporu geocachingu. Diskutuje jejich funkce, výhody, nevýhody, využití a také (ne)závislost na různých platformách. V závěru je zhodnoceno srovnání a vyvozena inspirace těmito aplikacemi.

V kapitole 3 je popsán postup získání specifikace z odlišných pohledů na věc, z pohledu uživatele a z pohledu vývojáře. její následné analýzy a vyvození příslušných závěrů, které budou sloužit jako podklady pro detailní návrh aplikace.

V textu kapitoly 4 je popsán detailní návrh aplikace jako celku, dále jednotlivých modulů, ze kterých je aplikace složena, návrh struktury databázových dat a nakonec také návrh struktury oken grafického rozhraní a rozložení ovládacích prvků. Texty jsou doplněny o výmluvné obrázky s diagramy.

Konečně kapitola 5 popisuje implementaci některých významných, či upozornění hodných postupů, nebo datových prvků. Zaměřím se zde na fyzické řešení problémů, které nelze zahrnovat do návrhu. Je zde zmíněn i výčet použitých prostředků a zdůvodnění jejich výběru.

Poslední kapitolou 6 je závěr, kde je popsáno zhodnocení výsledků práce, přínos této práce a možnost určitých návazností do budoucna.

Kapitola 2

Současný stav

Současná nabídka aplikací pro podporu geocachingu, pokrývající jak placenou sféru, tak i freeware, není příliš rozsáhlá, mluvíme-li o projektech přímo specializovaných pro geocaching.

2.1 Aplikace pro práci s GPS daty

Pokud budeme předpokládat i aplikace méně specializované (funkce jako správa geografických bodů, cest, tras, převod formátů souborů s body, zobrazování na mapě, atd.), nabídka se samozřejmě rozšíří. Tyto nástroje lze při geocachingu také využít, ale ne pro jeho plnou podporu.

- **Příklad: GPS Babel**, <http://www.gpsbabel.org/> Tento program slouží k převodu rozsáhlého počtu formátů souborů s geografickými daty (body, cesty, trasy). Pokrývá snad všechny známé formáty, tím se stává velice silným nástrojem, který využívá řada jiných aplikací pro svou potřebu. Pro geocaching jej lze využít při převodu bodů ve formátu ze serveru geocaching.com do formátu podporovaného vlastněnou GPS navigací. Je vydáván jako open-source.

Dále existují aplikace, které pro geocaching specializované jsou, ale jinými funkcemi, než offline správou dat, filtrováním, exportováním, atd. Jedná se hlavně a především o nástroje na výpočty statistik nálezů, jakými jsou např. frekvence nalezených keší v poměru k učitě době, počty nálezů v různých kategoriích, počet nálezů vzhledem k atributům cache, nebo suma nacestovaných kilometrů mezi cachemi.

- **Příklad: It's not about the numbers**, <http://www.itisnotaboutthenumbers.com/> Webový nástroj, umožňující po uploadování GPX souboru [2] s nalezenými cachemi vytvořit několik různých statistických souhrnů, např. počet nálezů v různých státech, počet nálezů vůči poměru obtížnost/terén či graf počtu nalezení ku měsícům.

Různých nástrojů pro práci s GPS daty je tedy na výběr dostatek, dále je však nemá smysl studovat, neboť se již netýkají tématu této práce.

2.2 Aplikace pro podporu geocachingu

V této sekci popíšeme některé často používané a citované nástroje přímo specializované pro podporu geocachingu. V komunitě českých geocacherů jsou tyto nástroje tři nejznámější. Jsou jimi **GSAK**, **GeoGet** a **Skrýšotlač**. Není vyloučena existence jiných aplikací tohoto typu, avšak zmiňovány jsou v drtivé většině tyto tři.

2.2.1 GSAK

Geocaching Swiss Army Knife (GSAK) je software pro správu geocache a jiných geografických bodů s širokou škálou funkcí a dobrou podporou ze strany poskytovatele. Je to ale placený produkt, což jej staví do poněkud jiné pozice ve srovnání s ostatními alternativami. To, že je produkt placený je ale samozřejmě vyváženo jeho vyšší kvalitou. Navíc je zde nevýhoda závislosti na operačním systému MS Windows.

Tato aplikace je jasně nejpoužívanějším softwarem pro podporu geocachingu a to z důvodu výše zmiňované vysoké funkcionality, množství podporovaných formátů a především použití maker, která dynamicky zvyšují obsah funkcí GSAKu.

Hlavní funkce aplikace

- Možnost správy více databází s bohatě strukturovanými záznamy
- Připojování mnoha atributů a poznámek k bodům
- Podrobné vyhledávání, filtrování a řazení podle řady požadavků
- Podpora mnoha formátů souborů s GPS daty (GPX, LOC, KML, WPT, ...)
- Přímá komunikace s některými podporovanými GPS přístroji
- Možnost automatizace některých úkonů pomocí maker (např. vytvoření statistik)
- Vestavěný komunikátor s Google Maps
- Desítky dalších drobných funkcí ...

Práce s GSAKem může být pro nezkušeného uživatele obtížnější, především začlenění tolika funkcí do grafického rozhraní s sebou nese jisté zmatení uživatele přílišným množstvím ovládacích prvků a datových oken. Po jisté době si ale uživatel program osvojí (ovládání je v celku intuitivní, začlenění drag & drop, kontextová menu).

Závěr: GSAK je jistě špičkou ve svém oboru poskytující téměř veškeré operace, které jsou pro práci s geocachingem potřeba. Tato práce si neklade za cíl dosáhnout podobné úrovně, pouze se inspirovat kladnými prvky aplikace. Na druhou stranu není GSAK multiplatformní SW, také je komerčním produktem.

WWW: <http://www.gsak.net/>

2.2.2 GeoGet

GeoGet je v komunitě českých geocacherů většinou označován jako *jednoduchá nekomerční alaternativa ke GSAKu*. Podle toho také vypadá. Má méně funkcí, některé detaily jsou řešeny jinak, ale program je celkově jednodušší. Funkcionalita je dle mého názoru dostatečná a aplikaci lze tedy bez problémů využívat.

Škála funkcí GeoGetu je podmnožinou funkcí GSAKu odfiltrovaná od nadstandardních drobných vymožeností. Neobsahuje ale především podporu maker, ani jiných skriptovacích prostředků.

V předchozích verzích obsahoval GeoGet také možnost přihlášení přes HTTP protokol na server `geocaching.com`, následné stažení množství HTML kódu, ze kterého program vydoloval informace o cache. Tato funkce již v nové verzi není, neboť odporuje licenci užívání služeb serveru `geocaching.com` (viz. [3, bod 5]).

Hlavní funkce aplikace

- Možnost správy více databází se záznamy různých bodů
- import GPX, LOC, HTML souborů
- Připojování poznámek k bodům
- Vyhledávání, filtrování a řazení podle řady požadavků
- Přidávání vedlejších bodů ke cache
- Podrobné ukázky bodů na serveru `mapy.cz`
- Možnost definování vlastního formátu exportu dat

Závěr: GeoGet je tedy vhodnou alternativou ke komerčnímu GSAKu, navíc má jednodušší ovládání, které je z pohledu uživatele většinou velice důležité. Aplikace má velké ambice do budoucna, je možné, že se stane rovnocenným konkurentem GSAKu a to s výhodou free licence. Nevýhodou GeoGetu je podobně jako u GSAKu jeho závislost na platformě MS Windows.

WWW: <http://geoget.ararat.cz/doku.php>

2.2.3 Skrýšotlač

Posledním studovaným softwarem byla aplikace Skrýšotlač. Tato aplikace je určena především pro stažení vybraných keší, jejich filtraci a následný export do několika formátů vč. HTML určeného pro tisk. Bohužel velkou nevýhodou je nemožnost perzistentního uložení dat ve vlastní databázi, jediná možnost, jak toto obejít, je exportovat připravený seznam do GPX souboru, který se později opět importuje.

Tento nástroj je tedy značně odlišný od předchozích dvou, které jsou založeny právě na perzistentně uložených datech a následné práci s nimi. Proto je tento program vhodný hlavně jako rozšířený filtr pro vyhledávání na serveru `geocaching.com` a jako příprava pro podrobný tisk informací o cache.

Další významnou odlišností je implementace v jazyce java, což aplikaci poskytuje určitou multiplatformnost. Je použitelná jak samostatně, tak jako applet v prohlížeči.

Hlavní funkce aplikace

- Načítání přímo ze serveru `geocaching.com`
- Velmi podrobná data o cache
- Generování HTML výstupu (pro tisk)
- Import/export několika formátů souborů s GPS daty
- Multiplatformní implementace v javě

Závěr: Skryšotlač poskytuje velmi podrobné informace o cache (pouze při stažení ze serveru `geocaching.com`) a jejich přípravě především pro tisk. Její velkou nevýhodou je nemožnost perzistentního uložení dat, což také vede k tomu, že veškerá načtená data drží v operační paměti, což při velkém objemu může způsobit komplikace v celém systému. Navíc je tento způsob značně pomalý. Je tedy vhodná jen pro menší objemy dat a jako pomůcka pro jednorázovou přípravu na cestu.

WWW: <http://cacheprinter.bubbles.sk/>

2.3 Zhodnocení a inspirace

Vzhledem k tomu, že součástí zadání této práce bylo zkoumání zavedených aplikací pro podporu geocachingu a poté navržení a implementace aplikace podobné, bude produkt této práce čerpat inspiraci z výše citovaných programů se značnou mírou. Především první dva nástroje odpovídají zadání této práce, proto většina soustředění padne na ně.

Zhodnocením dospějeme k tomu, že nejdůležitějšími požadavky pro výslednou práci je možnost importu a perzistentnímu uložení dat a jejich následné filtraci, vyhledávání, řazení a nakonec také exportu. Návrh je tedy nutno soustředit především na tyto aspekty tvořící jádro a nejmasivnější zastoupení funkcí v aplikaci. Veškeré další funkce jsou více či méně závislé na výše zmíněných. Nejdůležitějším přínosem by mělo být zavedení multiplatformnosti.

Kapitola 3

Požadavky a cíle

Základními požadavky podle zadání práce je několik bodů, tvořící základní pilíře návrhu. Jsou jimi:

Seznámení s podobnými aplikacemi Prozkoumání jejich funkcí a způsobů provedení. Již diskutováno v kapitole 2.

Výběr vhodných funkcí Nalezení nejdůležitějších a nejpoužívanějších funkcí, které se dají očekávat od nástroje tohoto typu. Jsou jimi především: import souboru s geocache, ukládání do databáze, široké možnosti vyhledávání, filtrování a třídění, export v různých formátech, podpora tzv. *paperless geocachingu* [1].

Multiplatformní implementace Tento bod je důležitý především kvůli tomu, že je zvolen jako hlavní přínos této práce oproti jiným alternativám. Tento moderní přístup je třeba využít.

Tyto body tvoří pouze základní sortu požadavků, braných jako obecné cíle, nezabývají se detailnějšími záležitostmi. K získání podrobnějších požadavků je třeba se na problém podívat zejména ze dvou pohledů. Z pohledu **uživatele** a **vývojáře**. Zatímco v prvním případě vzniknou či vyjdou na povrch požadavky, které budou v budoucnu úzce spojené s rozhraním aplikace vůči uživateli, v druhém případě naopak dostaneme požadavky na uživateli skryté věci; vnitřní uspořádání, potřeby různých knihoven, studium určitých algoritmů atd.

V následujících podkapitolách si tedy přiblížíme specifikaci a analýzu požadavků, které mohou být vzneseny z těchto protipólů.

3.1 Požadavky z pohledu uživatele

Pokud chceme studovat požadavky z pohledu uživatele, není nic snadnějšího, než se jím stát. Mohu mluvit z vlastní zkušenosti, jakožto dostatečně dlouhou dobu působící geocacher, z toho vyplývá, že také jako plnohodnotný uživatel. Nejlepší způsob, jak odhalit potřeby běžného geocachera je rozdělit působení na jednotlivé akce, které geocacher zažívá a hledat v nich požadavky na funkce budoucí aplikace.

3.1.1 Nejčastější problém

Uživatel chce vyhledat cache v blízkosti nějakého bodu, kde se bude v nejbližší době nacházet. Individuálně si bude poté chtít zvolit různé atributy, kterými bude chtít výběr filtrovat (obtížnost, terén, typ, ...). Z toho vyplývají zatím pouze již výše zmiňované základní požadavky.

3.1.2 Mystery, Multi cache

U těchto typů schránek zadané souřadnice neukazují na cílový bod, ten je třeba spočítat a uchovat výsledné souřadnice. Z toho vzniká potřeba opatřovat cache poznámkami s výpočty či mezivýpočty a výslednými souřadnicemi. Nebo dokonce přímo možnost přidat uživatelský bod k dané cache.

3.1.3 Prohlížení informací

Uživatel, rozhodnutý vydat se do akce také potřebuje prohlížet tzv. *listing* (popis) cache, kde jsou uvedeny informace o místě a často také o způsobu nálezu, nebo vypočtení finálních souřadnic. Proto je třeba uvažovat přehledný výpis podrobností zvolené cache.

3.1.4 Aktuálnost dat

Každý běžný hráč geocachingu zná také fakt, že vydat se na lov bez zkontrolování aktuálnosti listingu není moudré, neboť změny se nevyhýbají ani této oblasti, proto je potřeba zachovávat informace co nejaktuálnější, popř. k nim vytvořit jednoduchý přístup. Z toho vyplývá potřeba možnosti obnovování starých záznamů pomocí importu, nebo dokonce přímé stažení informací o cache ze serveru.

3.1.5 Cesta

Hráč geocachingu také často vyráží na delší cesty, u kterých by rád jednoduše zjistil cache, nacházející se podél této cesty. Tato možnost by měla být také uvažována.

3.1.6 Výlet v terénu

Velmi často se stane, že geocacher zavítá na delší výlet do přírody, nemá dostatečnou podporu GPS přístroje v otázce uchování informací a chtěl by tedy s sebou vzít např. notebook, kde by získal všechny potřebné informace. Aplikace by proto měla být naprosto nezávislá na připojení k internetu.

3.1.7 Domácí příprava

Uživatel bude v době přípravy vybaven připojením k internetu, ale bude chtít pracovat raději s aplikací, ve které zůstávají data uložena. Nechce ale přijít o komfort interakce dané cache se zbytkem internetu. Tímto dostáváme požadavek na možnost rychlých odkazů např. na stránku cache na serveru, či lokace cache na některém z webových mapových serverů.

3.1.8 GUI, rozhraní

Každý uživatel jakékoliv aplikace vybírá často podle toho, jak se aplikace ovládá a jak vypadá. Pokud si bude vybírat ze softwaru s podobnou funkcí, vybere si uživatelsky přívětivější.

Toto je základní série požadavků, kladených od běžného uživatele (hráče geocachingu). Prostudováním výše psaných úkonů dostáváme níže uvedený přibližný výpis funkcí, které by měla aplikace z pohledu uživatele obsahovat.

3.1.9 Shrnutí

1. Import/export různých formátů souborů s daty cache (např. GPX)
2. Vyhledávání, filtrování a řazení podle řady kritérií
3. Možnost přidávat ke cache body, poznámky, či soubory
4. Přehledné zobrazení dostupných podrobností o dané cache
5. Update starších dat při importu, stažení aktuálních dat ze serveru
6. Hledání cache podél zadané cesty (formou souboru s cestou)
7. Nezávislost na serveru `geocaching.com`
8. Možnost interakce s `geocaching.com` či mapovými servery
9. **Především jednoduché a intuitivní ovládání**
10. Příjemný a přehledný vzhled aplikace

To byl pohled z uživatelského hlediska. Na ten navazuje pohled vývojáře, který se znalostí těchto požadavků vytváří požadavky další, a to na vnitřní záležitosti aplikace.

3.2 Požadavky z pohledu vývojáře

Výčet potřebných schopností aplikace z pohledu vývojáře se odvíjí od požadavků uživatele, který dává svými náměty návrháři podklady pro vytvoření požadavků na určité postupy a technologie. V následující podkapitole tedy probereme náměty uživatelů z předchozí podkapitoly, ze kterých vyvodíme zavedení určitých technologií do aplikace.

3.2.1 Import/export

Pro diskutování importu a exportu dat je nutná znalost použitých formátů vstupu či výstupu.

GPX Jako nejzajímavější možnost se jistě jeví formát souboru GPX. Tento formát je založen na technologii XML, schéma je popsáno na [2]. Umožňuje popis jak standartních informací určitého GPS bodu, tak i přidání vlastních informací na místo XML stromu k tomu určené. Takový formát tedy může obsahovat všechny potřebné informace o cache bez jakékoliv vaty, tudíž je z pohledu efektivity nejlepší. Tento formát je také podporován nejpoužívanějšími GPS přístroji. Je tedy nejvhodnější jak pro import, tak pro export.

LOC Mezi další formáty patří například původní formát pro geocaching, formát LOC, který je také založen na XML technologii, ovšem obsahuje pouze velmi základní údaje o cache jako název, stručný popis a souřadnice umístění. Je tedy vhodný pouze pro uchování cache jako bodu s případným pozdějším updatem.

WPT Často používaným formátem je také formát WPT určený pro navigátor a mapový SW OziExplorer. Obsah souboru má jednoduchý formát, kde každý řádek představuje jeden bod v daném formátu. Není tedy potřeba žádný složitý parser, ani jiná technologie. Tento formát není poskytován na serveru geocaching.com, ale je využíván v řadě navigačních přístrojů, je tedy vhodné jej zařadit jako exportní formát.

HTML Poslední možností importování dat je HTML stránka stažená přímo ze serveru geocaching.com a její následné zpracování do použitelných informací. Tento způsob je velmi nestabilní, pomalý a lze jej použít jen jako krajní možnost. Tento formát bude uvažován v aplikaci jen teoreticky.

Shrneme-li tyto požadavky, zjišťujeme nutnost zavedení technologie XML, jako prostředek a pro čtení a zápis některých import/exportních formátů.

3.2.2 Uložení dat

Aplikace, jak již bylo řečeno, musí umožňovat persistentní uložení dat, důležitý je také efektivní přístup k těmto datům a jejich jednoduché a rychlé zpracování. S těmito znalostmi dostáváme nejvhodnější možnost – zavedení databáze do aplikace.

Databázové prostředky jsou rozděleny především na dvě kategorie. Jsou jimi: **interní** databázová logika, která je součástí aplikace jako knihovna a **externí** databázový proces, na který se aplikace připojuje prostřednictvím komunikačního kanálu. Vzhledem k tomu, že software by měl být především nezávislý a vezmeme-li v potaz také zjednodušení přímou integrací databázového systému do aplikace, jeví se jako lepší druhá varianta.

Databáze také nabízí jednoduchou strukturalizaci objemnějších dat pro jejich efektivnější vyhledání a zpracování. Umožňuje indexaci a velmi efektivní vyhledávání, filtraci, řazení. Je to celkově nejlepší způsob perzistentního uložení heterogenních tabulkových dat. Musí být tedy nedílnou součástí aplikace.

3.2.3 Výpočty souřadnic

V aplikaci je také důležitá schopnost počítat WGS-84 (viz. [8]) souřadnice, vzdálenosti, úhly a další geografická orientační data. Jelikož databáze většinou neoplývá funkcemi pro tyto výpočty a navíc je bude třeba využít i přímo v aplikaci. To si žádá buďto knihovnu, která obsahuje tyto funkce, nebo vytvoření vlastního modulu.

Jako jednodušší a jistější se samozřejmě jeví první možnost, ale ani po delším hledání takovéto knihovny jsem byl stále neúspěšný. Zbývá tedy pouze druhá varianta. Ta zahrnuje vyhledání daných algoritmů a jejich implementaci v rámci vlastního modulu aplikace.

Výpočet oblasti podél cesty (ad 6) nevyžaduje žádné jiné výpočty, nežli výše uvedené. Algoritmus pro jejich výpočet je triviální a vystačí z těmito elementárními operacemi.

3.2.4 GUI

Grafické uživatelské rozhraní (GUI) je velmi důležitým prvkem softwaru, jenž se má stát pro uživatele lehce a příjemně ovladatelným. Pro uživatele je důležitá především přehlednost ovládacích prvků, jejich snadné a především intuitivní ovládání.

Jako prostředek lze samozřejmě využít přímo knihovny k daným GUI okenním systémům, je to ale značně neefektivní a navíc by to mohlo ohrozit multiplatformnost aplikace, což by porušilo základní požadavek. Je tedy lepší zvolit multiplatformní nádstavbu, která bude obsahovat vše potřebné pro vytváření a aplikaci GUI.

Vzhled GUI by měl zůstat ve správě uživatelova okenního systému, neboť je na něj zvyklý a často si jej také dotváří podle svých potřeb a svého vkusu.

3.2.5 Strukturovaný popis cache

Pro takovýto výpis plný podrobností již nedostačují běžné prvky standartních GUI rozhraní, tato úloha vyžaduje prostředek umožňující jednoduše strukturovat a formátovat data, také s nenáročnou možností změny tohoto formátu.

Jako vhodná volba se jeví např. použití nějakého XML, nebo HTML vykreslovacího enginu, který jako vstup používá daný značkovací jazyk. Dnes se tyto formáty zavádějí i do dříve jinak kódovaným (většinou binárně) formátů souborů, určených pro dokumenty, což jistě znamená existenci určitých výhod. To je především čistě textová reprezentace.

Takovýto prvek je tedy v aplikaci nepostradatelný, ušetří mnoho práce vývojářům a zlepší kvalitu pro uživatele.

3.2.6 Prostředky řízení aplikace

Důležitou částí pro vývoj aplikace by měl být také systém, který podporuje vnitřní řízení aplikace, zprostředkující komunikaci se systémem, naprosto zapouzdřenou a tím také nezávislou na dané platformě. Následuje výčet hlavních požadavků, kladených na takovýto nástroj:

- Možnosti operačního systému (procesy, komunikace s jinými aplikacemi, ...)
- Lokalizace, jazykové mutace, podpora různých kódování
- Znalost standartních adresářů
- Komunikace s ostatními aplikacemi

3.2.7 Shrnutí

Z předchozí analýzy uživatelských požadavků vzniká tedy přinejmenším tato řada technologií, knihoven a podpůrných nástrojů, které je třeba do aplikace zavést.

- Zavedení technologie XML formou knihovny obsahující parser
- Databázový systém jako knihovna + případná nadstavba s ohledem k jazyku
- Knihovna, nebo modul pro základními výpočty souřadnic WGS-84
- Nadstavba GUI systému umožňující multiplatformní implementaci
- Podpora HTML engine, nejlépe zasazená do knihovny pro GUI
- Knihovny pro řízení aplikace, komunikaci se systémem a ostatními aplikacemi

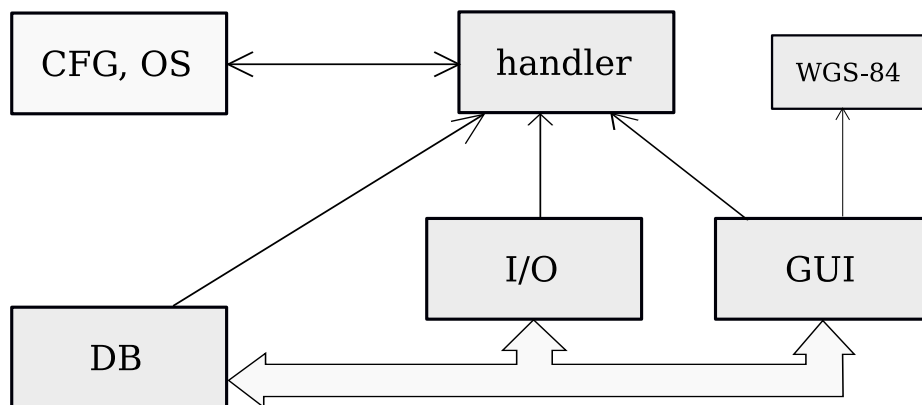
Kapitola 4

Návrh aplikace

Tato kapitola popisuje detailní návrh systému, jeho objektový návrh, dále návrh toku a větvení programu a také návrh struktury databáze. Jsou zde také zmíněny použité algoritmy k některým úlohám. Popis návrhu postupuje od návrhů obecných a aplikace jako celku po návrhy jednotlivých modulů.

4.1 Systém jako celek

Systém je založen na dekompozici jednotlivých funkčních celků do modulů. Každý modul plní jinou specifickou činnost a obsahuje třídy charakteristické pro tuto činnost. Princip uspořádání vysvětluje obrázek 4.1.



Obrázek 4.1: Grafické zpracování návrhu uspořádání modulů aplikace

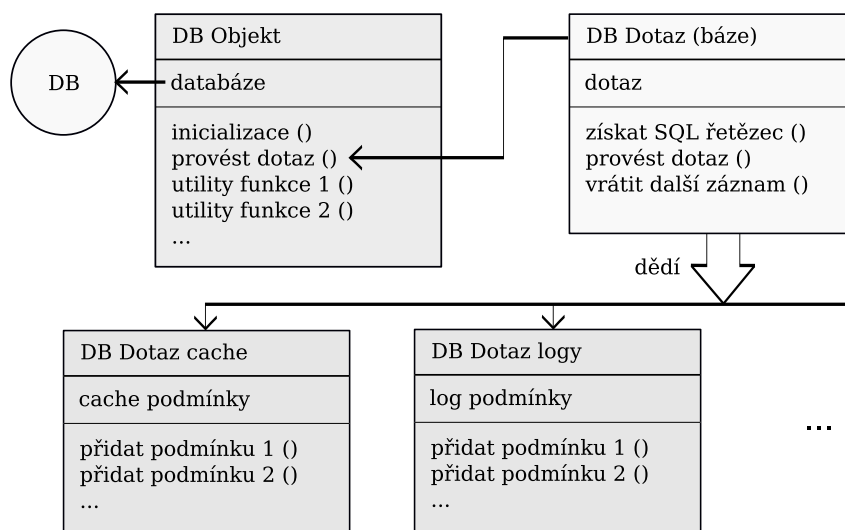
Jak je patrné z obrázku, moduly plnící specifické funkce, kterými jsou moduly **Databáze** (DB), **Import/Export** (I/O) a **GUI**, jsou instancovány do centrálního modulu **handler**, který se stará o většinu komunikace mezi ostatními moduly prostřednictvím předávání odkazů na určité vnitřní objekty jednotlivých modulů. Přes tyto otevřené komunikační kanály poté moduly mohou vyměňovat data a příkazy mezi sebou. Na globální sféře je modul, přístupný všem ostatním modulům, který se stará o konfiguraci aplikace a prostředí, také komunikaci se systémem.

4.2 Popis jednotlivých modulů

V následujících podkapitolách bude uveden podrobný návrh každého modulu, budou popsány objektové vztahy a také naznačeny vstupy a výstupy daného modulu.

4.2.1 Modul databáze – DB

Tento modul slouží pro komunikaci s databází, zastupovanou zapouzřeným objektem, pro účely popisu návrhu jej můžeme označit jednoduše jako *databáze*. Obrázek 4.2 ukazuje objektový návrh modulu.



Obrázek 4.2: Grafické zpracování objektového návrhu modulu databáze

Modul databáze obsahuje dvě důležité třídy. Jsou jimi **DB Objekt**, což je třída zprostředkující upravený přístup k databázi podle požadavků aplikace a **DB Dotaz**, což je bázová třída pro vytváření obecných dotazů na databázi.

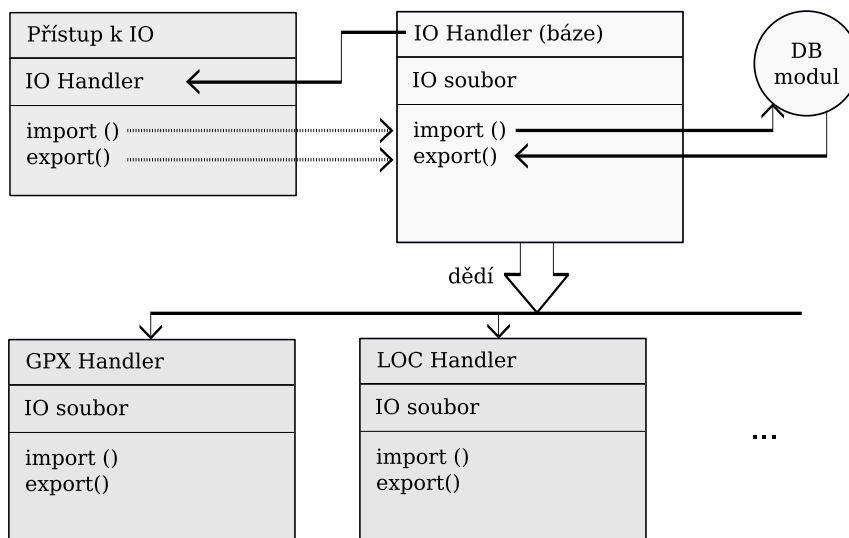
DB Objekt Musí existovat vždy pouze v jediné instanci, neboť fyzicky představuje spojení s DB, které nemůže být obsluhováno více objekty. Při inicializaci otestuje databázi a provede potřebné operace. Třída obsahuje především metodu `provést dotaz`, kdy se zpracuje předaný objekt dotazu, definovaný některou ze zděděných variant třídy DB dotaz a naplní ji požadovanými daty. Třída DB Objekt dále poskytuje řadu drobných funkcí, zapouzdřujících některé speciální operace v databázi.

DB dotaz Slouží jako abstraktní bázová třída, umožňující jednotnou obsluhu různých dotazů, vracejících tabulkové výsledky. Dotaz vždy obsahuje metodu na svoje spuštění v DB, virtuální metodu vytvoření SQL dotazu a také virtuální metodu pro iterační průchod výsledku dotazu. Děděné třídy si poté vytvoří ve vlastní režii tabulku podmínek a metody pro jejich editaci.

Datové vstupy a výstupy modulu tedy tvoří především třída DB dotaz, ale také některé ze specializovaných metod objektu DB Objekt.

4.2.2 Modul Import/Export – I/O

Je to modul, sloužící jako rozhraní pro import a export cache do a z databáze. Přistupuje k souborům na disku a podle pokynů zadaných uživatelem importuje, či exportuje dané cache. 4.3 ukazuje objektový návrh modulu.



Obrázek 4.3: Grafické zpracování objektového návrhu modulu import/export

V zásadě je tento modul podobný DB modulu, ale používá odlišnou metodu přístupu. Na rozdíl od DB modulu, zde je vždy nutno vytvořit objekt **Přístup k IO**, kterému jsou předány parametry importu/exportu a také typ handleru, který má být použit. Ten poté rozhodne, kterou dědenou třídu od základní třídy **IO Handler** má použít a předat jí dále dané parametry.

Přístup k IO Je třídou představující univerzálné rozhraní k vstupní, nebo výstupní operaci nad jednotlivými typy formátů. Je zásadně jediným přímým přístupovým a datovým kanálem k tomuto modulu.

IO Handler Představuje základní třídu pro vstupně výstupní operace, z níž jsou děděny specializované třídy pro určité formáty. Jako datový vstup pro IO operace lze použít objekt DB dotaz z 4.2.1, nebo objekt seznamu načtených cache z 4.4. Specializované třídy mají ve své režii veškeré zpracování předaných dat.

Jediným datovým kanálem, který lze pro tento modul použít je třída Přístup k DB.

4.2.3 Modul Souřadnice WGS-84

Tento modul je spíše pomocnou knihovnou obsahující třídy objektů souřadnic a výpočty spojené s těmito objekty. Objekty pracují v souřadném systému WGS-84, protože je to používaný systém pro GPS navigace.

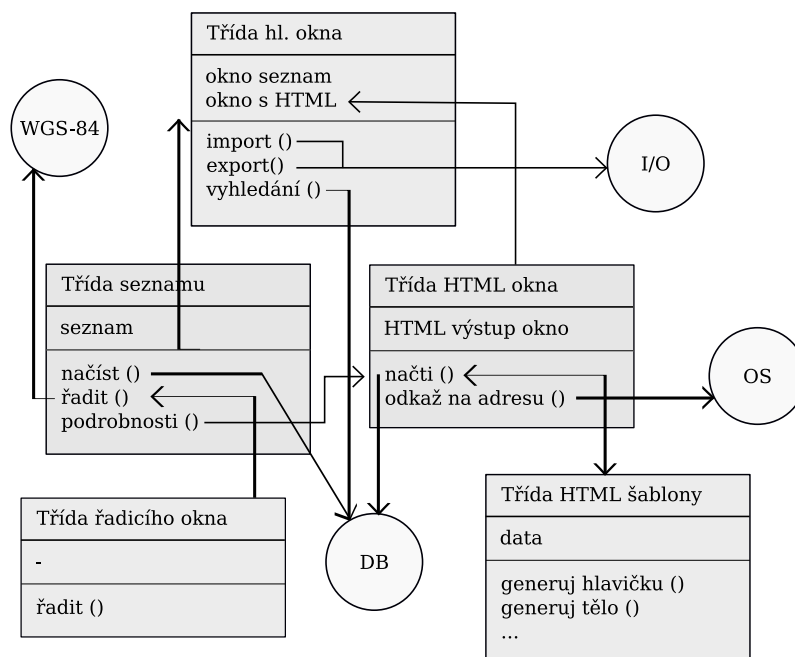
Stupeň Je třídou popisující polovinu GPS souřadnice. Zapouzdřuje inicializační a převodní operace.

Souřadnice Třída, sloužící k uchování GPS souřadnice a obsahující metody pro výpočty jejich vzáleností mezi sebou, úhlů, či výpočtu nových souřadnic podle zadaných parametrů. Samozřejmě je také metoda pro tisk souřadnic.

Bounding Box Dokáže uchovat čtveřici souřadnic značící rohy určité oblasti, lze jej poté použít pro detekci výskytu souřadnic uvnitř.

4.2.4 Modul Grafické rozhraní – GUI

Modul GUI slouží ke správě grafického rozhraní aplikace. Je tvořen třídami oken, které spolu složitě komunikují. Zde nelze jednoznačně hovořit o jasném objektovém návrhu modulu, neboť již ve velké míře využívá objektového modelu zavedené knihovny pro GUI. Obrázek 4.4 ukazuje strukturu tříd a spojení s ostatními moduly.



Obrázek 4.4: Grafické zpracování objektového návrhu modulu GUI

Zde nebudu popisovat objektové uspořádání, neboť je závislé na objektovém návrhu a možnostech dané knihovny a také není produktem této práce. Více o návrhu GUI se dozvíte v sekci 4.4.

4.2.5 Modul Handler

Modul handler obsahuje jedinou třídu pro řízení a komunikaci mezi moduly, a to třídu **Handler**. Tato třída uchovává jednotlivé objekty, instancované a registrované v průběhu inicializace aplikace. Obsahuje metody pro registraci vybraných objektů, které existují po celou dobu a naopak poskytuje také metody pro výdej těchto odkazů. Tímto se navazuje komunikace mezi moduly, které na sebe jiným způsobem nemohou vidět.

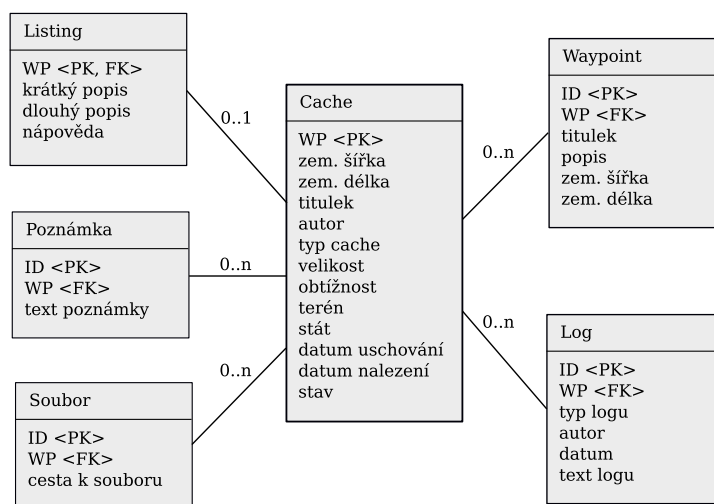
Ostatní globální třídy Návrh zahrnuje také existenci několika globálních objektů, využívaných všemi moduly podle potřeby. Jsou jimi především **Konfigurace**, **Komunikace s OS** a **Lokalizace**. Všechny tyto třídy ale spadají pod celek používané knihovny, nebudu je tedy zařazovat do vlastního objektového návrhu, více se o nich zmíním v kapitole 5.

4.3 Návrh databáze

Pro databázi je použit klasický relační model, neboť všechny dostupné databázové systémy tento model podporují, je také nejpoužívanějším. Tabulky jsou strukturovány takovým způsobem, aby byly rozděleny prioritní a méně prioritní informace o cache. To je výhodné např. při načítání seznamu, kde jsou potřeba pouze základní informace. Ostatní lze poté vyžádat dodatečně a nezneefektivňovat přístup.

Obrázek 4.5 popisuje strukturu databáze, jednotlivé základní tabulky a jejich vztahy pomocí ER diagramu.

4.3.1 ER diagram



Obrázek 4.5: Grafické znázornění návrhu struktury databáze pomocí ER diagramu

4.3.2 Popis diagramu

Cache Hlavním prvkem databázového návrhu je samozřejmě cache se svými důležitými atributy, vhodnými pro vyhledávání, řazení a filtrování. Tabulka pro cache tedy obsahuje pouze sloupce s těmito atributy, veškeré podrobnější informace jsou rozděleny do dílčích datových celků, kvůli menší datové zátěži při dotazování.

Listing Tato tabulka obsahuje položky uchováující podrobný popis (*listing*) cache. Cache může mít přiřazen pouze jediný záznam v tabulce listing, cizí klíč WP tabulky listingu tedy musí být unikátní.

Log Obsahuje data *logů*, neboli záznamů uživatelů, autorů i posuzovatelů cache (*reviewerů*). Uchovává informace o autorovi, datu záznamu, typu záznamu a jeho textový obsah. Na každou cache se může vázat teoreticky neomezené množství logů, tudíž vzniká vazba 1..n.

Waypoint Je tabulka geografických bodů, které mohou být použity jako přídavné body k určité cache, ale i jako samostatné body. Návrh s touto možností počítá, v této práci se jí ale nedostane využití. Díky tomu, že bod může být samostatný (nezávislý na cache), položka cizího klíče může obsahovat prázdnou hodnotu. Počet vázaných bodů je podobně jako u logů neomezený.

Poznámka a soubor Tyto tabulky slouží jako shromaždiště textů a souborů k určitým cache. Tyto data jsou oddělená především kvůli tomu, že jsou nestandardní. Takovýto typ dat není na geocaching.com podporován. Slouží jako utříděný seznam uživatelských poznatků k dané cache.

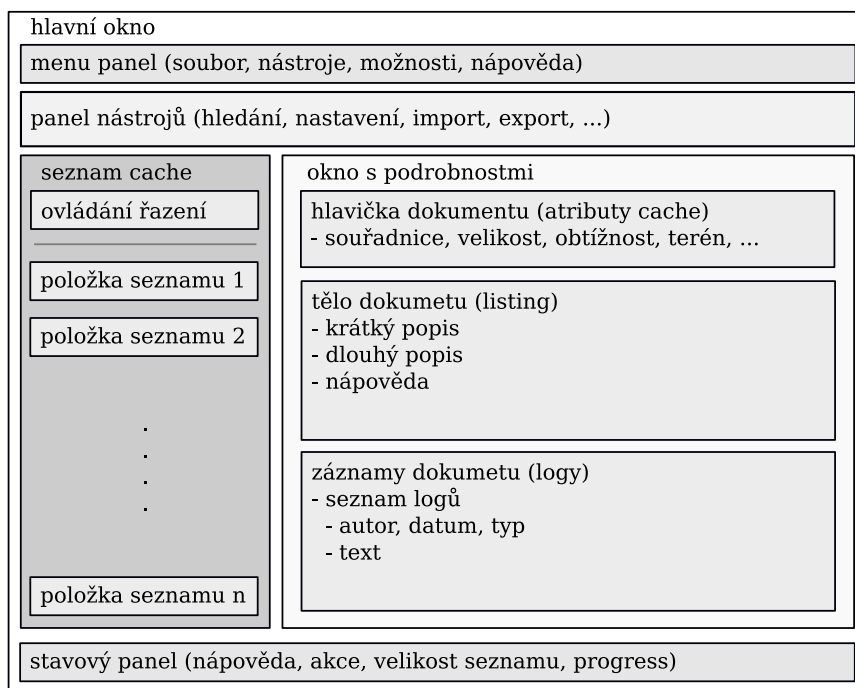
Databáze je v praxi obsluhována zvoleným databázovým systémem, v rámci aplikace je pak přístupná přes modul DB (4.2.1).

4.4 Návrh grafického rozhraní

Návrh vzhledu a uspořádání grafického rozhraní vychází z uživatelských požadavků (sekce 3.1), jejichž znalost je pro schopnost správně navrhnout a uspořádat strukturu grafického rozhraní s ovládacími a informačními prvky velmi důležitá. Je to přímé spojení s uživatelem, musí tedy plně vyhovovat zaběhnutým způsobům ovládání a zároveň také obsáhnout všechny potřebné prvky, nutné pro ovládání aplikace a zobrazování informací.

Na obrázku 4.6 je znázorněn vzhled a struktura grafického rozhraní, navrženého pro tuto aplikaci. Veškeré prvky (podokna) hlavního okna grafického prostředí jsou zde v návrhu zatím zmíněny jako abstraktní objekty, které dostanou přesnější podobu až v kapitole 5. Význam grafických prvků je zřejmý, není tedy třeba legendy, ani jiných pomocných informací. Obrázek 4.7

4.4.1 Struktura GUI



Obrázek 4.6: Grafické znázornění struktury a uspořádání GUI

4.4.2 Popis struktury GUI

Základní kámen grafického rozhraní tvoří objekt hlavního okna, které je rodičem všech ostatních oken, na něj umístěných.

Menu a nástroje Základem většiny aplikací jsou panel menu a panel nástrojů, standardně umístěny vodorovně pod horním okrajem okna. Panel menu obsahuje kategorizovaný seznam téměř veškerých akcí, které jsou v aplikaci k dispozici. Panel nástrojů je osazen některými nepoužívanějšími akcemi formou tlačítka s ikonou, vyjadřující její smysl.

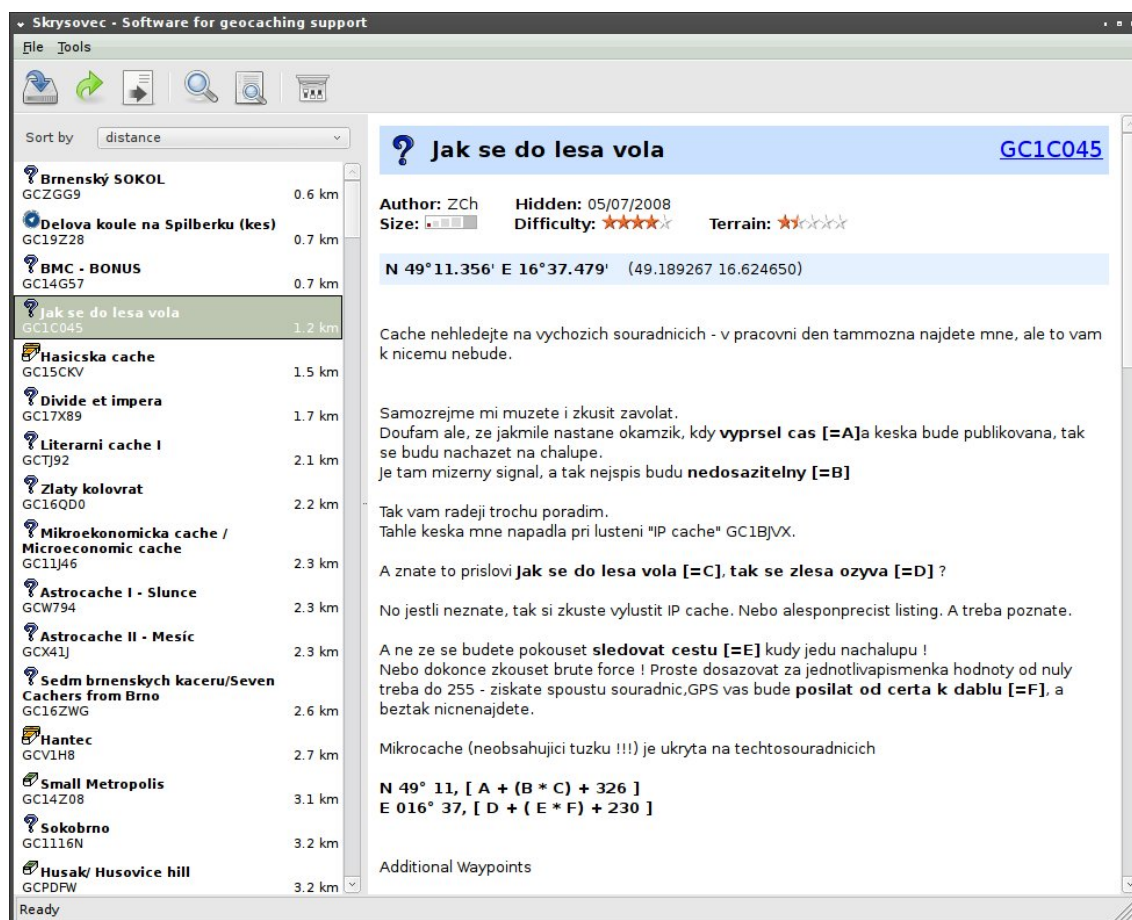
Stavový panel Taktéž velmi základním prvkem je stavový panel, který je standardně umístěn vodorovně nad spodním okrajem hlavního okna. Tento panel slouží jako ukazatel průběhu některých akcí, či ukazatel stavu nějaké hodnoty. Standardně je také využíván jako zobrazovač nápovědy k ovládacím prvkům menu a panelu nástrojů.

Plocha hlavního okna Je rozdělena svislým tahačem na dva odlišné celky, kterými jsou seznam cache a okno zobrazující podrobnosti.

Seznam cache Tento panel je umístěný v levé polovině plochy hlavního okna. Obsahuje vertikálně řazený seznam cache, kde každá položka obsahuje název, *waypoint* (unikátní označení pro každou cache) a také vzdálenost a směr od domovských souřadnic. V horní části tohoto okna je také umístěn ovládací prvek s volbou klíče řazení. Akce v tomto okně řídí vedlejší okno s podrobnostmi.

Podrobnosti Toto okno má za úkol zobrazovat formátovaný dokument s podrobnostmi zvolené cache. Obsahuje také prvky spojené s úkony týkající se otevírání dovolených adres v standartním prohlížeči.

Grafické rozhraní je řízeno zprávami zasílanými operačním systémem, jako klasická jiná aplikace s GUI a v reakcích na tyto zprávy komunikuje s odpovídajícími moduly.



Obrázek 4.7: Ukázka praktické realizace grafického rozhraní aplikace

Kapitola 5

Implementace

V této kapitole budou především popsány vybrané postupy nějakým způsobem vyjimečné, nebo důležité. Před tímto výčtem je také vhodné zastavit se nad výběrem použitých nástrojů a knihoven a zdůvodněním jejich použití.

5.1 Výběr implementačních nástrojů

V sekci 3.2.7 byly shrnuty požadavky na externí nástroje, které je třeba do aplikace zahrnout. Předtím, než začnu popisovat vybrané nástroje, je důležité zmínit programovací jazyk, použitý při implementaci. Tímto jazykem je objektově orientovaný (OO) jazyk C++ poskytující drtivou většinu výhod OO modelu. Od toho se tedy bude dále odvíjet i výběr knihoven a jiných nástrojů, nutných k realizaci aplikace. Samozřejmostí při výběru nástrojů byl fakt, že daný nástroj musí být multiplatformní. Následuje výčet použitých knihoven, komentář k jejich výhodám a zdůvodnění výběru.

5.1.1 wxWidgets

Tato knihovna byla jasnou volbou již od počátku, neboť poskytuje veškeré funkce multiplatformně a také pokrývá velmi široké spektrum schopností. Od implementace grafického rozhraní, přes řadu datových struktur, konfigurace, souborové systémy až po XML parsery, HTML prohlížeče, nebo například kompletní systém lokalizace.

Tato knihovna odvádí za programátora opravdu kvanta práce, což má výhody jak charakteru časové úspory, tak také větší stability a kompatibility. Výčet funkcí této knihovny je opravdu předlouhý, uvedu tedy nejdůležitější prvky, použité v této aplikaci. Patří mezi ně: GUI, přístup k souborům, řetězce a jejich kódování, lokalizace, vlánka, prostředky operačního systému, XML parser.

5.1.2 SQLite

Je knihovnou, mající v režii databázi, obsahuje prostředky pro její vytváření, zápis, úpravy, atd. Je to tedy plnohodnotný databázový systém s výhodou přímé integrace do aplikace jako knihovna. SQLite není tak vybaveno jako robustnější databázové nástroje. Obsahuje však všechny potřebné funkce a schopnosti, které se od databázového systému očekávají, včetně transakcí atd. Knihovna sama o sobě nabízí pouze rozhraní pro jazyk C, což je v tomto projektu nevyhovující. Přímé použití by bylo zbytečně komplikované, navíc by to rušilo obraz objektového zpracování aplikace.

Bylo tedy vhodné najít vhodnou nastavbu nad C rozhraní SQLite pro jazyk C++. Pro tyto účely byla zvolena knihovna **SQLite wrapped**, což je velmi jednoduchá a nenáročná nastavba nad SQLite pro C++.

5.1.3 WGS-84

Pro tento účel nebyla nalezena žádná vyhovující knihovna, přistoupil jsem tedy ke studiu problematiky uveřejněné na internetu, kde jsem objevil několik algoritmů pro výpočty ve WGS-84. Nakonec jsem použil volně šiřitelné algoritmy z [6]. Studium těchto algoritmů v rámci této práce je nesmyslné, neboť se jí týká pouze nepřímo.

5.1.4 Zdroje

Při používání cizích knihoven je nutné uvést zdroj, zde je tedy seznam použitých knihoven spolu s jejich zdroji:

- **wxWidgets**, licence *LGPL*
<http://www.wxwidgets.org/>
- **SQLite**, licence *public domain*
<http://www.sqlite.org/>
- **SQLite wrapped**, licence *GPL*
<http://www.alhem.net/project/sqlite/index.html>
- **WGS-84 algoritmy**, licence *LGPL*
<http://www.movable-type.co.uk/scripts/latlong-vincenty.html>

5.2 Inicializace aplikace

Vstupní bod aplikace a tedy i místo prvotní inicializace je metoda třídy aplikace `OnInit()`. Nejprve jsou nastaveny určité parametry, na kterých je závislý další průběh. V první řadě se nastaví instalační adresář (určený v době kompilace) do globálního objektu správy standardních adresářů (*wxWidgets*). Z něj se pak odvíjí defaultní cesty k adresářům, využívaných aplikací. Bezprostředně poté je vytvořen a zveřejněn konfigurační objekt, starající se o načtení, čtení, zápis a následné uložení konfigurace po skončení aplikace.

5.2.1 Inicializace databáze

Dalším krokem je vytvoření databázového objektu `DbObject`, který je následně registrován do globálního objektu handleru. Inicializace databáze proběhne v rámci konstruktoru databázového objektu. Nejprve je otevřeno spojení s databází, která se stará o čtení a také existenci souboru s databází. Po připojení zkontroluje `DbObject` existenci vyžadovaných tabulek v databázi metodou `testTables()`, kde zjistí, zda nejsou v databázi chybějící tabulky. Pokud nějaké zjistí, voláním metody `createTables()` tyto chybějící tabulky v databázi vytvoří. Po tomto bodu je již databáze správně inicializována.

5.2.2 Inicializace GUI

Prvním bodem inicializace je vytvoření hlavního okna, které poté prostřednictvím svého konstrukturu vytváří a inicializuje dceřinná okna. Polohovatelná okna přitom využívají konfiguračního objektu, ze kterého čtou poslední známou polohu. Při svém zániku naopak svou polohu ukládají.

Jako první jsou vytvořeny panely menu, nástroje a stavu, všechny jejich položky také a jsou usazeny na své pozice. Následuje vytvoření oken seznamu cache a podrobností cache a jejich usazení do *splitteru* (Objekt dělí plochu panelu na dvě podokna s možností upravit šířek).

5.2.3 Inicializace seznamu

Při vytváření okna seznamu cache se toto okno inicializuje přes svůj konstruktorem, ve kterém vytvoří objekt `DbCacheRequest`, který předá databázi na zpracování, následně z něj metodou `loadCaches()` vytáhá seznam cache, který je již nachystán pro zobrazení a další operace s ním. Pro každou cache také spočítá vzdálenost a směr od domovských souřadnic, tyto hodnoty uloží jako součást dočasného záznamu v seznamu (kvůli zobrazení a řazení).

5.3 Import, Export

Z charakteru těchto operací jasně vyplývá, že budou pracovat s většími objemy dat, lze předpokládat, že dokážou vzít také velké množství výpočetního času. Měly by být tedy implementovány relativně efektivně.

Při požadavku uživatele o import, nebo export vytvoří GUI objekt `MainIO`, kterému jsou předány údaje o souboru. Ten poté podle typu rozhodne, který IO handler použije. Zde se silně uplatňuje dědičnost a polymorfismus, vytvářející jednotné rozhraní ke vstupu/výstupu. Objekt `MainIO` obsahuje dvě metody, `importCaches()` a `exportCaches()`, umožňující pracovat s daným handlerem jako s objektem jediného typu¹.

5.3.1 Import cache

Jak bylo řečeno výše, import cache může být často časově náročný. Kromě toho je nutné zvážit skutečnost, že operace může být uprostřed nějakým způsobem přerušena, což vede k nutnosti závest tuto operaci v celku jako transakci, které knihovna SQLite podporuje. K použití transakce slouží metody třídy `DBObject`. Metodou `beginTransaction()` je transakce započata a pokud proběhl import bez problémů, transakci lze potvrdit metodou `commitTransaction()`.

5.3.2 Export cache

Pro export cache existují v aplikaci dvě možnosti, co se týče předané datové struktury. Za prvé aplikace umožňuje export aktuálního seznamu cache, nacházejícím se v okně seznamu. Za druhé lze použít objekt `DbCacheRequest`, naplněný požadovanými podmínkami (např. jako výstup dialogu vyhledávání). Provedení této operace opět zajišťuje objekt `MainIO`, který obsahuje přetíženou metodu `exportCaches()`, jíž lze použít pro obě dvě varianty. Export již není třeba uzavírat do transakce, neboť nemění databázová data, ale pouze je čte.

¹V aplikaci implementována zatím pouze podpora GPX formátu

5.3.3 Vlákna

Kvůli zmíněné časové náročnosti se nabízí možnost, použít pro import/export operace samostatných vláken, vracejících hlavnímu procesu v pravidelných intervalech informace o stavu dané operace. Kvůli jednoduchosti a lepší stabilitě je zvolena varianta zobrazení uživatelem neuzavřitelného modálního dialogu, který informuje o stavu operace, jiná práce po tuto dobu není možná.

5.4 Implementace seznamu

V sece 5.2.3 je popsáno chování seznamu při inicializaci, je vytvořen čistý dotaz a cache z něj načteny do seznamu. Takto vytvořený objekt dotazu `DbCacheRequest()` je v seznamu uchován pro případné znovunačtení seznamu. Pokud uživatel vyplní a potvrdí nějaký vyhledávací filtr, jeho dialog vrátí nový objekt dotazu, kterým je přepsán ten stávající a seznam je načten znovu podle nového filtru.

5.4.1 Vyhledání cache podle atributů

Základní vyhledávací operací je hledání podle atributů. Ta je představována dialogem se vstupními prvky jednotlivých podmínek. Po potvrzení dialogu je vytvořen objekt `DbCacheRequest`, který je poté naplněn zadanými podmínkami a předán objektu seznamu cache `SkListBox`, který jej učiní novým objektem tohoto seznamu.

5.4.2 Vyhledání cache podél cesty.

Tento vyhledávací postup potřebuje poněkud složitější logiku. Vstupem této operace je soubor s body cesty, které od sebe mohou být různě vzdálené. Dále dostaneme od uživatele přibližnou vzdálenost od cesty x , kterou máme uvažovat. Pro tento případ se nabízí iterační postup, kdy mezi dva sousední body b a $b+1$, které jsou od sebe vzáleny více než uvažovaná vzdálenost x , vkládáme postupně nové body, vždy na přímku mezi těmito dvěma body v každém kroku po vzdálenosti x . Tím docílíme toho, že prohledávaná plocha bude přibližně odpovídat ploše vedoucí podél cesty. Pro všechny body jsou pak spočítány *bounding boxy* s danou šířkou $2x$, které jsou poté vloženy do vytvořeného objektu `DbCacheRequest`. Odsud je postup již stejný, jako u předchozí varianty.

5.5 Okno podrobností

Toto okno `SkHtmlView` je v aplikaci v praxi reprezentováno HTML vykreslovacím oknem, jehož vstupem je kód v HTML jazyce, okno se poté chová jako prohlížeč. To dovoluje velmi jednoduchou a efektivní práci s formátováním výstupu. Ve chvíli, kdy obdrží seznam cache zprávu o požadavku na zobrazení podrobností vybrané cache, je předán tomuto HTML oknu index dané cache. Okno si poté vybere veškeré informace o cache z databáze a naformátuje. Formátování provádí za pomoci šablonovací třídy `SkDescTemplate`. Vrácený HTML dokument poté předá přímo oknu za účelem vykreslení.

Třída okna `SkHtmlView` má implicitně nastaveno načítání obrázků a otevírání hypertextových odkazů do stejného okna, což je pro stabilitu nebezpečné. Proto před zobrazením jsou všechny přímo ukazované obrázky pomocí regulárních výrazů přepsány na hypertextové odkazy. Odkazy na jiné zdroje jsou blokovány úplně. Obsluha hypertextových odkazů z okna podrobností je poté přesměrovaná do standartního prohlížeče systému.

Kapitola 6

Závěr

Cílem bylo vytvořit aplikaci, která hráčům hry geocaching umožní jednoduchou správu dat, získaných ze serveru `geocaching.com`. Požadavkem bylo obsažení základních vyhledávacích, filtrovacích, import/exportních operací, která by tato aplikace měla zvládat. Nejdůležitějším a zároveň nejpřírodnějším požadavkem byla multiplatformní implementace, která zatím ve sféře softwaru pro podporu geocachingu chybí.

V návrhu jsem se snažil vytvořit prostředí, kde jednotlivé moduly, či objekty mají definovaná jasná a jednoduchá rozhraní, přes která navzájem komunikují. Důraz jsem kladl také na to, aby byla aplikace jednoduše rozšiřitelná, tzn. bez zásahů do původního návrhu a s minimálními zásahy do původních zdrojových kódů. Toto je zaručeno především využitím principů objektového programování, jako jsou dědičnost a vazba rodičovských a synovských tříd. Tento aspekt lze využít např. při požadavku integrace dalších typů souborů, nebo vyhledávacích dalších tříd databáze.

Aplikace nenavazuje na jiné projekty, je projektem původním, který ale čerpá inspiraci z podobných aplikací tohoto typu, které jsou však vesměs určeny pro jedinou platformu.

Tento software obsahuje pouze základní výčet funkcí, připadajících na nástroj tohoto typu. Jak je ale psáno výše, model aplikace je přímo navržen s ohledem na budoucí rozšiřování. Vzhledem k široké paletě požadavků, kladených uživateli se tato aplikace přímo nabízí pro budoucí práci na něm a rozšiřování jeho funkcí. Jedná se především o různé vyhledávací, statistické a třídící operace, či ovládací prvky grafického rozhraní.

Hlavním přínosem této práce je multiplatformní alternativa k zavedeným nástrojům pro podporu geocachingu, neboť aplikace tohoto typu jsou zásadně určeny pro platformu MS Windows. Tato práce tedy přináší podobné výhody i uživatelům jiných platforem, zejména linuxu.

Literatura

- [1] Cacheopedia: Paperless geocaching. [online], march 2008, rev. 10. march 2004, [cit. 2008-05-10].
URL http://cacheopedia.com/wiki/Paperless_geocaching
- [2] Foster, D.: GPX: the GPS Exchange Format. [online], may 2008, rev. 10. may 2008, [cit. 2008-05-10].
URL <http://www.topografix.com/gpx.asp>
- [3] Groundspeak: Geocaching.com site terms of use agreement. [online], 2003, rev. 5. april 2004, [cit. 2008-05-10].
URL <http://www.geocaching.com/about/termsofuse.aspx>
- [4] Groundspeak: Cache Types. [online], may 2008, rev. 10. may 2008, [cit. 2008-05-10].
URL http://www.geocaching.com/about/cache_types.aspx
- [5] Lutonský, M.: Geocaching: hra pro mozek, nohy a vaši GPS. [online], únor 2007, rev. 1. únor 2007, [cit. 2008-05-10].
URL <http://navigovat.mobilmania.cz/clanky/AR.asp?ARI=112930>
- [6] Veness, C.: Vincenty formulas. [online], květen 2002, rev. 11. květen 2008, [cit. 2008-05-11].
URL <http://www.movable-type.co.uk/scripts/latlong-vincenty-direct.html>
- [7] Wikipedia: Geocaching. [online], may 2008, rev. 5. may 2008, [cit. 2008-05-10].
URL
<http://en.wikipedia.org/w/index.php?title=Geocaching&oldid=210233859>
- [8] Wikipedie: World Geodetic System. [online], duben 2008, rev. 2. duben 2008, [cit. 2008-05-10].
URL http://cs.wikipedia.org/w/index.php?title=World_Geodetic_System&oldid=2431428