VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

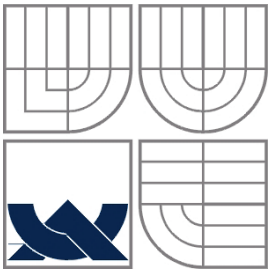
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

NAVIGACE PRO PARAGLIDING PRO PLATFORMU ANDROID

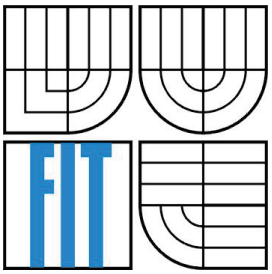
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE MARTIN IMRICH
AUTHOR

BRNO 2013



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

NAVIGACE PRO PARAGLIDING PRO PLATFORMU ANDROID

PARAGLIDING NAVIGATION FOR THE ANDROID PLATFORM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — MARTIN IMRICH
AUTHOR

VEDOUCÍ PRÁCE — Ing. ALEŠ LÁNÍK
SUPERVISOR

BRNO 2013

Abstrakt

Tato práce popisuje návrh a implementaci aplikace Navigace pro paragliding na platformě Android. Vzdušný prostor je rozdělen na velké množství území s různými pravidly a omezeními. Paraglidista ve vzduchu nepozná, jestli se již nachází v zakázaném vojenském prostoru nebo právě prolétává omezeným prostorem kolem letišť. Tato aplikace zobrazuje dané prostory a napomáhá celkové orientaci ve vzduchu. Pomůže paraglidistovi vypočítat dolet v závislosti na terénní situace kolem něj.

Abstract

This thesis describes design and implementation of an Android application for paragliding. Airspace is divided into several areas with different rules and limitations. When is the paraglider in the air, he can hardly know that he's already in any of these limited areas – like military area or airport. This application shows these areas and helps with orientation in the air. It also helps to calculate and show landing of the paraglider depending on the nearby terrain.

Klíčová slova

Paragliding, paragliding navigace, Android, mapy, Google mapy, navigace, Java, vzdušné prostory.

Keywords

Paragliding, Paragliding Navigation, Android, maps, Google Maps, navigation, java, airspace.

Citace

Martin Imrich: Navigace pro paragliding pro platformu Android, bakalářská práce, Brno, FIT VUT v Brně, 2013

Navigace pro paragliding pro platformu Android

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Aleše Láníka.
Další informace mi poskytl také pan Ing. Jozef Mlích.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Martin Imrich
15. května 2013

Poděkování

Tímto bych chtěl poděkovat vedoucímu práce Ing. Aleši Láníkovi za odborné konzultace, technickou korekturu a čas, který mi věnoval. Dále děkuji Ing. Josefu Mlíchovi, který mi poskytl k nahlédnutí jeho aplikaci databáze letišť. Poděkování patří i Zeměměřičskému úřadu, který mi zapůjčil výšková data České republiky.

Obsah

1 Úvod

Paragliding je létání, které vám umožní zažít něco nepopsatelného. Člověk si vždycky přál umět létat a vynalezl již několik prostředků, díky kterým může tento skvostný pocit zažít. První zmínky o pokusech o létání sahají až do 6. století našeho letopočtu a to vypovídá o tom, že lidstvo již dávno toužilo po té volnosti, která se dá zažít pouze ve vzduchu.

Na první pohled se to zdá být velmi atraktivní a prosté, ale realita je jiná. Ve vzduchu panuje mnoho pravidel a omezení, kterými se paraglidista musí řídit. Tato pravidla musela být nastolena k zajištění bezpečnosti a určitému řádu v tomto, nyní již velmi frekventovaném, prostoru. Paraglidista musí například respektovat omezené prostory, které mohou být v okolí letišť, vojenských zón, významných budov či staveb. Tyto prostory jsou sice definovány, ale v reálném světě vidět nejsou. Jak tedy může paraglidista vědět, zda se již nenachází v zakázaném prostoru v okolí letiště? Další problém se týká doletového prostoru. Paraglidista ve vzduchu nemůže vědět, jak daleko doletí, či případně přeletí vrcholek, který se nachází před ním.

Tato práce pojednává o návrhu a vytváření aplikace, která usnadňuje pohyb a orientaci v paraglidingu. Zobrazuje paraglidistovi jeho aktuální polohu, výšku a mapu pod ním. Do mapy zároveň vykresluje definované prostory a paraglidista tak vidí, kde může létat a kam již bez speciálních povolení nesmí. Dále mu dokáže vykreslit doletový prostor s ohledem na terénní situace a tím usnadňuje kompletní plánování a umocňuje celkový dojem z létání.

V druhé kapitole se čtenář dozví všechny potřebné informace k pochopení problému, souvislostí a spojitostí s aplikací. Je seznámen s platformou Android, získá základní povědomí o mapových aplikacích, jsou mu vysvětleny základní pojmy v paraglidingu a také se dozví, jak je rozdělen vzdušný letecký prostor České republiky a jaké jsou možnosti produktů výškopisu České republiky.

V další kapitole je čtenáři představen návrh aplikace na platformu Android. Je zde představen objektový návrh aplikace, použití Google Maps API, vykreslování vzdušných prostorů do mapy a návrhy řešení vykreslení doletového prostoru. K tomu účelu jsou v této kapitole popsány 3 algoritmy možného řešení.

V poslední kapitole je řešena konkrétní implementace aplikace Navigace pro paragliding. V této kapitole se postupně řeší veškeré moduly a funkce aplikace tak, jak byla postupně vyvíjena. Kapitola je doplněna o snímky z obrazovky a zaměřuje se na celkovou orientaci v mém řešení daného problému. Na konci kapitoly je také popsáno otestování aplikace a návrh na pokračování projektu.

2 Mapové aplikace na platformě android

V této kapitole si projdeme základní informace, které nám pomůžou pochopit základní fakta o vytváření mapových aplikací na platformě Android. Postupně si vysvětlíme k čemu slouží mapové aplikace, seznámíme se s platformou Android, vysvětlíme si základní terminologii v paraglidingu, rozdělíme a popíšeme vzdušný prostor a podíváme se na výškopis České republiky.

2.1 Mapové aplikace

Vytváření mapových aplikací se v poslední době velmi rozmáhá a vzniká velké množství těchto aplikací. Stále častěji totiž lidé vyhledávají různé informace v mapách a k tomu jim tyto aplikace mohou sloužit. Prostá mapa nám sice odhalí zmenšené objekty, povrch Země, či si díky ní můžeme naplánovat trasu našeho výletu, ale již nám nedokáže sdělit bližší informace. Např. o jaký objekt se jedná, kontakt na danou společnost, zda je daný areál veřejnosti přístupný nebo zda po trase našeho výletu nalezneme nějaké hotely. Tato metadata se dají zjistit a do mapové aplikace přidat např. pomocí značky, kterou uživatel v mapě vidí a po kliknutí si může tyto bližší informace o vybraném objektu zobrazit.

Mapové aplikace se začaly zprvu vyvíjet jako webové aplikace pro osobní počítače. Hlavním hráčem na trhu s online mapami je bezesporu společnost Google Inc. s jejich produktem Google Maps. Po vydání Google Maps v únoru roku 2005 odstartovala nová éra mapování našeho světa. Google si předsevzal, že vytvoří mapy, které budou obsáhlé, přesné a jednoduché k použití, což se jim povedlo na výbornou a mapy stále vylepšují. Za zmínku stojí technologie Google Street View. Google mapuje panoramatický pohled míst, kde se člověk pohybuje. Využívá k mapování speciálních kamer na autech, sněžných skútrech, kolech, vlacích a dává si tak za cíl zmapovat veškerý prostor na Zemi. Vytváří také kompletní 3D modelace měst a usnadňuje tak lidem orientaci v mapách a celkově usiluje o uživatelovo pohodlí v práci s mapami.

Pár měsíců po vydání Google Maps, přesněji v červnu 2005, posílá Google do světa Google Maps API a umožňuje tak vývojářům využívat tyto služby a integrovat Google Maps do webových stránek a či např. mobilních aplikací.

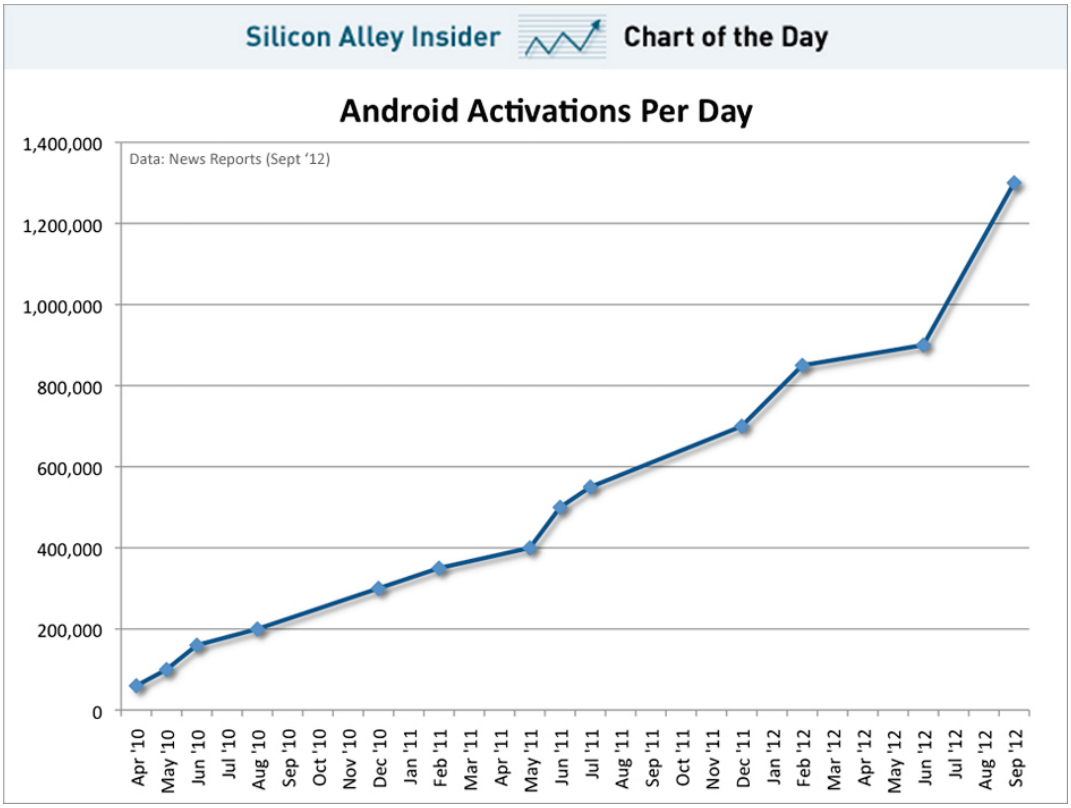
Mobilní zařízení nabízejí další široké uplatnění mapových aplikací. Mobilní mapové aplikace se dají jednoduše využít jako navigace nebo jako pomocník při poskytování v mapě uložených informací. V říjnu 2008 prvně spatří světlo světa Google Maps aplikace pro Android a lidé tak získávají velmi schopného pomocníka v orientaci ve městě či své zemi.

2.2 Seznámení se s platformou Android

Android je open source operační systém (založen na jádru Linuxu), vytvořen zejména pro mobilní zařízení. Android vyvíjí Open Handset Alliance společně se společností Google Inc., která Android Inc. koupila v srpnu roku 2005 [1].

Android je poskytován pod licencí Apache a jeho zdrojový kód může být tedy volně modifikován a distribuován výrobci zařízení, mobilními operátory nebo vývojáři. Systém je navržen tak, aby co nejefektivněji pracoval na cílových zařízeních. Při vývoji tedy byly zohledněny faktory jako výdrž baterie, menší výkonnost a málo dostupné paměti. Zároveň bylo jádro Androidu navrženo pro běh na různém hardwaru a pro kompatibilitu s různými rozměry zobrazovacích displejů. Lze tedy snadno vytvořit jednu aplikaci, která může běžet na desítkách různých zařízení.

Android je nejpopulárnější mobilní platforma. Dokazuje to např. číslo 2 miliony aktivací za den, které uvádí ve zprávě Nathan Harbin [2], a údaj z března 2013, kdy se CEO společnosti Google Larry Page vyjádřil [3] o celkovém počtu 750 milionů aktivovaných zařízení. Podívejme se ještě na počet aktivací Android zařízení za den v období od dubna roku 2010 do září 2012 na následujícím grafu:



Graf 1: Vývoj aktivací Android zařízení za den. Graf převzatý z http://www.businessinsider.com/chart-of-the-day-android-activations-per-day-2012-9.

Android komunita je obrovská. Vidíme to z čísel a zásluhu na tom jistě má bezplatné poskytování systému, ale především široká nabídka aplikací nabízených přes Google Play. Uživatelé si mohou vybírat mezi více než 850 000 aplikací, ať již placených či neplacených.

Z těchto informací lze odvodit, že Android nabízí velmi rozsáhlý trh pro vývoj a prodej mobilních aplikací pro tuto platformu a jistě zajímavou budoucnost. To je lákadlo pro vývojáře Android aplikací, kteří tvoří také velmi rozsáhlou komunitu a vytvářejí právě obsah Google Play. V Google Play nalezneme také velké množství mapových aplikací či navigací.

2.3 Co je paragliding?

Paragliding je rekreačním, ale i závodním leteckým sportem. Paragliding se řídí zákony o civilním letectví a podléhá tedy jistým pravidlům. K provozování potřebuje pilot pilotní průkaz pro danou skupinu a podmínky k provozování u nás stanovuje Letecká amatérská asociace České republiky.

K provozování paraglidingu patří toto povinné vybavení:

- Padákový kluzák
- Postroj
- Přilba
- Výškoměr
- Záchranný padák

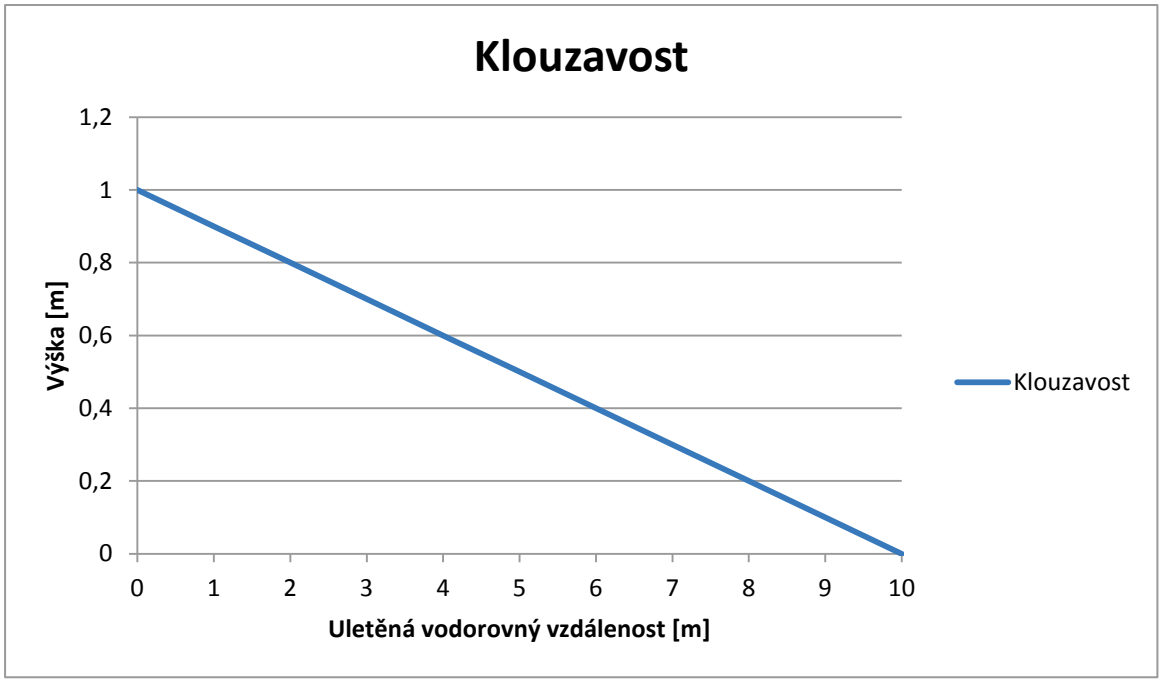
Hlavním příslušenstvím paraglidisty je bezpochyby padákový kluzák, který se skládá z křídla a šňůr, které jej spojují se sedačkou.

2.3.1 Klouzavost

K pochopení výpočtu teoretického doletu je nutné vyjasnit, co to znamená klouzavost kluzáku. Klouzavost je obecně jedním z velmi sledovaných technických parametrů paraglidingových křídel. Klouzavost můžeme definovat vztahem popsaným v článku od M. Ramšaka [4]:

$$klouzavost = \frac{L}{D}$$

kde L znamená vztlak (Lift) a D odpor (Drag). S tímto vzorcem operují většinou konstruktéři křídel. Klouzavost je jednotka bez veličiny a jedná se o vztah mezi uletěnou vzdáleností a rozdílem výšky. Tedy např. klouzavost 10 nám říká, že na 1 metr ztráty výšky uletíme 10 metrů vodorovné vzdálenosti. Prohlédnout si to můžeme na následujícím grafu.



Graf 2: Klouzavost.

Mnohem pochopitelnějším vzorečkem pro výpočet klouzavosti je pak tento vztah:

$$klouzavost = \frac{\Delta s}{\Delta h}$$

kde Δs je rozdíl (přírůstek) vodorovné vzdálenosti a Δh je rozdíl (úbytek) výšky. Je nutné si uvědomit, že klouzavost je měřena a ověřována v ideálních podmínkách. V realitě se tedy mohou vyskytnout jisté nuance od naměřených hodnot především díky proudění vzduchu, tuto skutečnost nebudu při výpočtu zohledňovat.

2.4 Zvyklosti a rozdělení leteckého vzdušného prostoru

Za vzdušný prostor považujeme jakoukoliv část atmosféry Země. Z hlediska létání je tento prostor rozdělen na řízený a neřízený. Toto rozdělení opět sahá do historie, kdy provoz v leteckém vzdušném prostoru nebyl zdaleka tak hustý, jako je tomu nyní. Piloti většinou létali pouze za dobré viditelnosti a díky nízké hustotě provozu nehrozily srážky ani na letišti. Když se podíváme na tuto situaci nyní, je zcela odlišná.

Letecká doprava v Evropě zaznamenává od počátku 90. let minulého století velký nárůst a představuje nejrychleji se rozvíjející dopravní odvětví. Tento nárůst znamenal také nutnost zavedení jistých pravidel pro zvýšení bezpečnosti a celkově vybudování systému. Proto bylo zřízeno ŘLP

(řízení letového provozu), které letadlům poskytuje pokyny, rady a informace pro zajištění bezpečného a ekonomického leteckého provozu, mj. také bezpečné rozestupy mezi letadly. V Česku tuto službu zajišťuje státní podnik Řízení letového provozu České republiky.

V dnešní době je vzdušný prostor rozdělen do několika vrstev, nás bude především zajímat ta nejnižší, tedy neřízená, kde se většinou pohybují paraglidisté. Čím vyšší vrstva, tím větší řízenost a tedy potřeba lepšího vybavení. Mimo toto rozdělení jsou vyhlášeny další prostory, jak jsem již naznačil v úvodu. Jsou to prostory kolem letišť, některých objektů na zemi, prostory pro potřeby vojenských letů apod. Každá vrstva a každý prostor patří do některé třídy.

V České republice rozeznáváme dle klasifikace vzdušného prostoru ATS publikované v letecké informační příručce AIP v části ENR [5] 4 třídy vzdušného prostoru:

- Třída C – řízený prostor, zahrnuje TMA (viz Typologie prostorů). Praha a vzdušný prostor od FL[1] 95 do FL 660
- Třída D – zahrnuje CTR/TMA všech letišť s výjimkou TMA PRAHA
- Třída E – prostor mimo CTR/TMA nad 1000 stop AGL[2] do FL 95
- Třída G – s výjimkou CTR vzdušný prostor od země do 1000 ft AGL

2.4.1 Typologie prostorů

V této podkapitole si vymezíme definice prostorů a jejich zkratek, se kterými budu dále pracovat. Tyto prostory jsou v aplikaci vykreslovány a každý typ má určitá pravidla. Následující informace vycházejí z letecké informační příručky [6].

Řízený okrsek

Z angličtiny Control Zone a zkratkou CTR je část vzdušného prostoru v těsném okolí letišť. Spodní hranice je ohraničena zemským povrchem, horní pak v rozmezí 9 až 20 km, ve směru vzletové dráhy je hranice nejvyšší.

Koncová řízená oblast

Z angličtiny Terminal Maneuvering Area a zkratkou TMA je část vzdušného prostoru v okolí letišť zpravidla navazujícím na CTR. Spodní hranice je většinou 1000 stop AGL a horní hranice 125 FL.

ATZ

Z angličtiny Aerodrome Traffic Zone je část vzdušného okolí neřízeného letiště. Hranice prostoru sahají od země do výšky 4000 metrů nad mořem. Provoz zde není řízen, ale letadla by měla navázat spojení se stanovištěm ATZ a řídit se jejich pokyny.

[1] Flight Level = letová hladina, je udávána ve stovkách stop a znamená nominální hodnotu nadmořské výšky
[2] Above Ground Level = nad zemí

Zakázaný prostor

Je část vzdušného prostoru sloužící k ochraně některých objektů v rámci České republiky. U nás jsou značeny LKP + pořadové číslo. Jedná se např. o prostory kolem Pražského hradu, jaderných elektráren Temelín a Dukovany, nebo továren na výrobu výbušnin. Tyto prostory chrání objekty před dopadem letadla při případné letecké havárii.

Omezený prostor

Je část vzdušného prostoru, kde je průlet omezen. V ČR jsou označeny LKR + pořadové číslo. Tyto prostory se zřizují pro ochranu před prováděním střelby, vojenských cvičení, nebo jako ochranná pásma v okolí jaderných elektráren.

Nebezpečný prostor

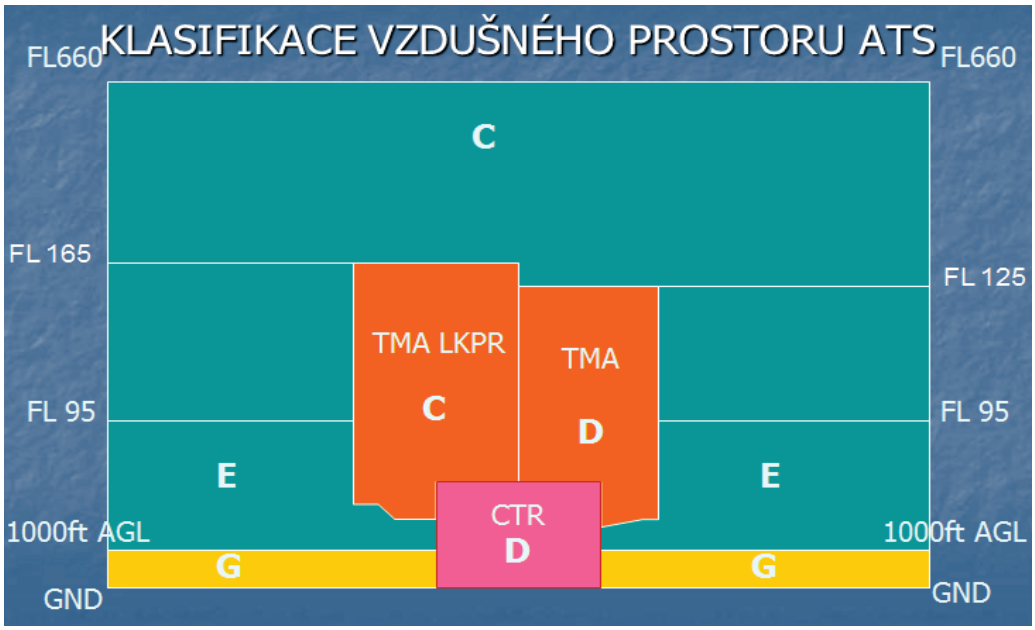
Značeny LKD + pořadové číslo. Tyto prostory se zřizují nad objekty, nad kterými může být nebezpečné prolétat. Nebezpečným prostorem je možné proletět, ale je doporučeno se mu vyhnout.

Dočasně vymezený prostor

Z angličtiny Temporary Reserved Area a zkratkou TRA je část vzdušného prostoru sloužící k ochraně letadel, kterým není možné zajistit rozestupy. Jedná se většinou o pracovní prostory pro pořádání leteckých soutěží nebo prostory pro vojenská letadla. Průlet je pouze na žádost po spojení se stanovištěm řízení letového provozu. Průlet nemusí být povolen.

Dočasně vyhrazený prostor

Z angličtiny Temporary Segregated Area a zkratkou TSA je část vzdušného prostoru sloužící k ochraně letadel, kterým není možné zajistit rozestupy. Jedná se o stejný typ jako je TRA, ale zde není zřízeno žádné řízení letového prostoru a je tedy průlet zakázán. Spodní hranice prostoru je 300 stop nad terénem a horní ve výšce 1000 stop nad terénem.



Obrázek 1: Shrnutí klasifikace vzdušného prostoru

Na předchozím obrázku je shrnutí rozdělení vzdušného prostoru v České republice. Paraglidisté se většinou nacházejí ve třídě G, která je v rámci dalších tříd nejníže položená a zahrnuje nejmenší rozsah vzdušného prostoru.

2.5 Výškopis České republiky

Jednou z funkcí mé aplikace je vykreslení teoretického doletu paraglidisty. Tento dolet musí zohlednit terénní situaci v daném okolí. Jelikož vytvářím aplikaci v rámci oblasti České republiky, potřeboval jsem sehnat výšková data.

Vytvářením výškopisu České republiky se zabývá Český úřad zeměměřický a katastrální (dále Čúzk) a tato data jsou v současné době vedená v rámci Základní báze geografických dat (ZABAGED®). Výškopisnou část ZABAGED® tvoří 3 typy objektů vrstevnic se základním intervalem 5, 2 nebo 1 m v závislosti na charakteru terénu a dále vybrané terénní hrany. Objekty jsou reprezentovány trojrozměrnou vektorovou prostorovou složkou. Výškopisná část ZABAGED® je doplněna o digitální model reliéfu v podobě pravidelné mříže (10 x 10 m) trojrozměrně vedených (3D) bodů, který je odvozený z vrstevnic a terénních hran ZABAGED®.

V roce 2008 byly dále zahájeny práce na novém digitálním modelu výškopisných dat v rámci České republiky s využitím technologií leteckého laserového skenování. Jedná se o produkty Digitální model povrchu České republiky 1., 4. a 5. generace.

Pro účely vytvoření této funkce v aplikaci jsem si musel zvolit vhodný produkt, pojďme se podívat na srovnání produktů.

Produkt	Pokrytí ČR	Chyba výšky (v metrech)	Cena za výdejní jednotku (v Kč)
ZABAGED® Grid 10x10 m	100 %	0,7 – 2,5	244
ZABAGED® Vrstevnice	100 %	0,7 – 2,5	244
DMR 1G	33 %	0,4 – 0,7	700
DMR 4G	67,9 %	0,3 - 1	500
DMR 5G	33,4 %	0,18 – 0,3	620

Tabulka 1: Srovnání produktů výškopisu ČR.

Z tabulky 1 vidíme, že produkty DMR 1G, 4G a 5G ještě nejsou dokončeny. Údaj na webových stránkách Čúzk udává, že 100% pokrytí se plánuje na rok 2015. Chyba výšky udává, jakou maximální chybu výšky může daný bod obsahovat. Dolní hranice je udána pro body v odkrytém terénu, horní hranice pak pro body v zalesněném terénu. Tady vidíme, že nejpřesnější je produkt DMR 5G. Cenově nejvýhodněji vycházejí produkty ZABAGED®. Produkt ZABAGED® Vrstevnice se však vydává pouze v grafickém formátu a já potřebuji textové údaje o souřadnicích bodu s jeho výškou.

Tento účel tedy nejlépe splní produkt ZABAGED® Grid 10x10 m, který poskytuje 100% pokrytí, relativně nízkou chybu výšky a především je vydáván v textovém formátu *[Souřadnice X, Souřadnice Y, Výška]*.

Čúzk poskytuje vybraná data pro účely bakalářských a diplomových prací po podání žádosti bezplatně. Tato data vydává však pouze v omezeném množství a tak se mi podařilo získat alespoň část dat v Brně, abych mohl vyzkoušet funkčnost mé aplikace. Má aplikace tedy nebude kompletně funkční pro celou Českou republiku, ukáže ale schopnost aplikování navržených principů na tomto vzorku a tyto principy by šly poté použít v plném rozsahu v případě dodání kompletních dat.

Problémem ukázkových dat byl jejich formát. Data nebyla v zeměpisném tvaru souřadnic, ale ve státním souřadnicovém systému JTSK. Musel jsem tedy ještě provést transformaci S-JTSK do ETRS 89 formátu, kterou poskytuje Čúzk na svém portále. Je to ale funkce velmi nevlídná pro uživatele, a pokud by bylo nutno převádět větší množství dat, stálo by za to implementovat vlastní transformaci. Je tam totiž omezení 1 MB na soubor a ještě musí mít vstupní soubor jiný formát, než ve kterém mi ho poskytli, což mi přišlo poněkud zvláštní.

3 Návrh aplikace

V předchozí kapitole jsme se dozvěděli, co jsou to mapové aplikace, seznámili jsme se s platformou Android a blíže se podívali, co je to paragliding. Dále jsme si popsali a rozdělili vzdušný prostor a vybrali jsme vhodná výšková data. Nyní již tedy známe veškeré záležitosti, které potřebujeme k vytvoření návrhu aplikace Navigace pro paragliding.

V této kapitole se podíváme na postup navrhování jednotlivých komponent, ale i celkové aplikace vyvíjené na platformě Android. Aplikace bude fungovat jako klasická mapová aplikace s přidanými vrstvami. Vrstvy se budou vykreslovat na standardní mapě, ale budou průhledné tak, aby uživatel viděl stále na mapu. Hlavní okno aplikace bude zobrazovat jen základní informaci v podobě aktuální výšky a bude obsahovat 3 tlačítka. Dvě standardní tlačítka z Google Maps API pro vycentrování polohy a ovládání přiblížení či oddálení mapy a třetí vlastní tlačítko pro vykreslení teoretického doletu. Aplikace bude obsahovat pouze základní nastavení, kde si uživatel bude moci nastavit klouzavost jeho padáku, nastavit ručně výšku a zobrazit satelitní nebo normální pohled mapy.

3.1 Objektový návrh v programovacím jazyku Java

Zdrojové kódy Android aplikací jsou psány v programovacím jazyku Java. Java je objektově orientovaný programovací jazyk, který vyvinula firma Sun Microsystems, a patří mezi nejpoužívanější programovací jazyky na světě. Aplikace bude vyvíjena primárně pro Android verze 2.3.3, cílem ovšem je, aby fungovala na široké škále zařízení a tedy i různých verzích systému.

Celé řešení problému jsem pojmul objektově. Jako reálné objekty jsem navrhnul samotnou mapu, vzdušné prostory, paraglidistu a výškové body. Každý objekt má své vlastnosti a metody, kterými komunikuje s okolními objekty. Některé objekty jsou standardní součástí Google Maps API, jiné si navrhnu sám.

Podle doporučení v knize Hello, Android: Introducing Google's Mobile Development Platform [9] je vhodné dodržovat jisté optimalizace kódu a zvýšit tak efektivitu aplikace na mobilních zařízeních. Není vhodné vytvářet zbytečné objekty, měly by se preferovat statické a finální konstanty, pokud to situace dovoluje atd. Můj návrh tedy bude řešit i optimalizaci zdrojů a kódu pro slabší výkon a menší paměť, s čím se musí u vývoje mobilních aplikací počítat. Více popíšu v kapitole Implementace aplikace Navigace pro paragliding.

3.2 Použití Google Maps API

Na použití mapového základu pro vyvíjenou aplikaci se dá nahlížet ze dvou pohledů. Jedním by bylo použití vektorových dat v „offline“ režimu, tím druhým využití Google Maps, tedy „online“ režim. Každé řešení má své výhody a nevýhody. Aplikace s vektorovými daty by nevyžadovala připojení k internetu, aplikace by tedy byla vhodná i pro uživatele bez internetu v mobilu. Velkou nevýhodou je na druhou stranu objemnost těchto dat, která by jistě byla velká a aplikace by tak vyžadovala hodně volného místa na externí kartě telefonu.

Na druhé straně je řešení s použitím Google Maps API, které sice vyžaduje připojení k internetu, ale nemá již takové vysoké nároky na volné místo a zároveň poskytuje již předdefinované a velmi užitečné funkce, díky čemu se tak aplikace stane uživatelsky velmi přívětivou a ocení to i vývojář.

Ze začátku jsem aplikaci navrhoval pro Google Maps API verze 1, protože jsem s ním měl již jisté zkušenosti a mapy fungovaly na mém Samsung Galaxy Mini telefonu rychleji. Aplikaci jsem tedy navrhoval a vyvíjel pro tuto verzi API a zpráva z Googlu o ukončení vývoje a poskytování API klíčů k verzi 1 mě nijak neznepokojovala. Aplikace vytvořené pro verzi 1 mají dále fungovat, jen již od 18. března 2013 Google nevydává klíče k této verzi API. To se mi stalo osudné, když jsem v dubnu přišel o data a tím i soubor s úložištěm klíčů a nemohl jsem tam pokračovat ve vývoji na verzi 1.

Tímto impulsem jsem byl tedy donucen předělat aplikaci do Google Maps API verze 2 a aplikace je nově navržena pro tuto vylepšenou verzi. Nová verze API poskytuje mnoho užitečných vylepšení. Nově využívá vektorové dlaždice a reprezentace dat je tak menší a mapa se zobrazuje rychleji. Z hlediska návrhu jsou mapy nyní zapouzdřeny v rámci třídy MapFragment a implementovat je můžeme rozšířením standardní Android Activity třídy [7].

3.3 Návrh vykreslování vzdušných objektů

Má aplikace bude umět vykreslovat 2 typy prostorů. Prvním typem jsou zakázané či omezené prostory. Tyto prostory budou neměnné a stačí je tak vykreslit pouze při startu aplikace. Druhý typ je samotný doletový prostor. Tento prostor se vykreslí pouze po kliknutí na tlačítko. Prostor zůstane vykreslen, dokud si uživatel nebude chtít vygenerovat nový doletový prostor. V tom případě se původní prostor vymaže a vypočítá se aktuální, který se zakreslí do mapy.

Vykreslování bude probíhat pomocí polygonů a definovaných bodů. Oficiální souřadnice zakázaných a omezených prostorů můžeme najít v letecké informační příručce, já však využiji databázi letišť ze serveru www.aerobaze.cz. Soubor je ve formátu GPX a obsahuje základní informace o letištích a dále spodní vzdušný prostor od země do FL 95, což bohatě dostačuje pro účely mé práce.

Prostory budou vykreslovány průhlednou barvou tak, aby byl vidět mapový základ pod nimi. Prostory, kterými je zakázáno proletět, budou vykresleny barvou červenou. Jak jsem popsal v kapitole 2.4.1, jedná se o prostory CTR, CTA, LKP, LKR, TMA, TRA a TSA. Zahrnuji do nich tedy i prostory, kde je nutno se spojit s řídícím střediskem letového provozu. Pro tento projekt nepředpokládám, že paraglidista bude mít s sebou takové zařízení. Prostory, kterými je průlet pouze nedoporučen, budou vyznačeny modrou barvou. To jsou prostory LKD a ATZ. Výsledný doletový prostor bude ohraničen modrou linkou a uvnitř vybarven průhlednou zelenou.

3.4 Pohledy na výpočet doletové oblasti

Na funkce vykreslování teoretického doletu paraglidisty se dá nahlížet z různých úhlů pohledu. Musel jsem si nejprve uvědomit, jaké situace mohou nastat a jak nejlépe a nejefektivněji vyřešit tento problém. Kdyby totiž byla na Zemi všude stejná výška, bylo by to velmi jednoduché. Stačilo by výšku paraglidisty, označme jako *h,* dosadit do vzorečku na výpočet klouzavosti *k* a dostali bychom dráhu, což vlastně odpovídá poloměru výsledného doletového prostoru, byla by to totiž kružnice. Výška *h* se spočítá jako rozdíl aktuální výšky paraglidisty a_1 a výšky bodu terénu pod ním a_0. Dosadíme si smyšlené hodnoty:

$$k = \frac{s}{h}$$

$$k = \frac{s}{a_1 - a_0}$$

$$10 = \frac{s}{500 - 240}$$

$$s = 2600\,m$$

Jednoduchým dosazením získáme dráhu 2,6 km a mohl bych vykreslit kružnici s tímto poloměrem a můj úkol by byl hotový.

Já ale navrhuji aplikaci praktického využití a ta musí reflektovat terénní situaci v daném okolí. Vyskytuje se zde několik problémů. Velké množství dat (výškových bodů) a nepředvídatelný terén. Může se např. stát, že v dráze letu se nachází vysoký kopec, přes který bychom nepřeletěli, ale za ním může opět být nízká výška. Kdybych tedy nekontrolovat celou dráhu, mohl bych tento kopec lehce přeskočit a dolet by tak byl nesprávný. K řešení tohoto problému jsem navrhnul několik algoritmů, které si nyní projdeme.

3.4.1 Algoritmus paprsků

Algoritmus paprsků vychází v simulování vždy jednoho přímého směru (trajektorie) a kontrolování výškových bodů, jestli jsme již nenarazili na bod, za který by se paraglidista nedostal. Algoritmus by vypadal takto:

1. *Zjisti aktuální výšku.*
2. *Nastav směr vytvoření paprsku na jih.*
3. *Vytvoř paprsek v trajektorii letu paraglidisty.*
4. *Posuň se o 10 metrů a spočítej budoucí výšku paraglidisty v tomto bodě.*
5. *Podívej se na hodnotu výškového bodu v tomto místě.*
6. *Pokud je výška bodu menší než budoucí výška paraglidisty, vrať se na bod 4.*
7. *Ulož bod.*
8. *Nastav směr paprsku o 5 stupňů na východ.*
9. *Pokud je celkový počet otočení menší než 360 stupňů, běž na bod 3.*
10. *Spoj uložené body a ukonči algoritmus.*

Tento algoritmus je docela přesný, ale jelikož se paprsky rozbíhají, mohla by vzniknout nepřesnost při vysoké výšce paraglidisty a tedy vysokém rozptylu paprsků na konci jejich trajektorie. Algoritmus by se mohl adaptovat v závislosti na výšce paraglidisty. S rostoucí výškou by se snižoval stupeň otáčení paprsků. Tím by byl algoritmus přesnější, na druhou stranu pomalejší. V tomto případě by dále šla efektivita zvýšit nastavením větší vzdálenosti kontroly výškových bodů v bodě 4.

3.4.2 Algoritmus kružnic

Tento algoritmus řeší problém s nepřesností bodů. Kontroluje totiž všechny body a je tedy přesnější než algoritmus paprsků. Algoritmus vychází z myšlenky vytváření kružnic pod paraglidistou a kontrolou všech bodů. Střed kružnice odpovídá souřadnicím paraglidisty. Vypadal by následovně:

1. *Zjisti aktuální výšku.*
2. *Zmenši výšku pro vytvoření kružnice o 10 metrů.*
3. *Vypočítej poloměr kružnice v této výšce tak, aby trajektorie paraglidisty byla tečnou této kružnice.*
4. *Vytvoř kružnici.*
5. *Porovnej všechny výškové body, které se nachází na úrovni kružnice.*
6. *Ulož všechny body, které mají výšku větší nebo rovnu aktuální výšce kružnice.*
7. *Pokud mají všechny výškové body větší výšku, než je výška kružnice, běž na bod 9.*
8. *Vrať se na bod 2.*

9. *Spoj všechny body, které byly v rámci jedné trajektorie uložené jako první, vykresli tento prostor a ukonči algoritmus.*

Algoritmus kružnic je jistě přesnější než algoritmus paprsků, problém ale nastává při určování výsledných bodů, které mají být spojeny. Je to dáno tím, že se posuzují všechny výškové body a těch je po kružnici velké množství. Těžko tedy můžeme vyhodnotit, které body máme určit jako výsledné. Šlo by to řešit vybíráním prvních bodů v určitém pruhu vyslaném z pozice paraglidisty po trase trajektorie, což by bylo značně náročnější na výpočet.

3.4.3 Algoritmus vnitřních a vnějších bodů

Kompromisem a mým řešením je algoritmus vnitřních a vnějších bodů. Algoritmus vypočítá teoretický rádius doletu a kontroluje všechny body, které se nacházejí v tomto prostoru. Označuje body uvnitř, vně a na hranici. Výsledně spojí body na hranici a nejbližší vnitřní body, které jsou blízko dalším vnějším bodům, to znamená, že jsou nejblíže hranici doletu. Tento prostor výsledně vykreslí. Popsaný vypadá následovně:

1. *Zjisti aktuální výšku.*
2. *Vytvoř rádius teoretického doletu.*
3. *Vezmi bod a zjisti jeho výšku.*
4. *Porovnej tuto výšku s budoucí výškou paraglidisty v tomto bodě.*
5. *Pokud má bod nižší výšku, označ ho jako vnitřní.*
6. *Pokud má bod stejnou výšku, označ ho jako hraniční.*
7. *Pokud má bod větší výšku, označ ho jako vnější.*
8. *Dokud neprojdeš všechny body, vrať se na bod 3.*
9. *Najdi všechny dvojice sobě nejbližších vnějších a vnitřních bodů.*
10. *Vnitřní body z těchto dvojic označ jako hraniční body.*
11. *Spoj všechny hraniční body, vykresli doletový prostor a ukonči algoritmus.*

Tento algoritmus dosahuje přesných a jednoduchých výsledků. S rostoucí výškou však roste náročnost na výpočet.

4 Implementace aplikace Navigace pro paragliding

V této kapitole si popíšeme konkrétní implementaci navrhnuté aplikace v prostředí Eclipse SDK. Popíšeme si konkrétní implementační řešení problémů řešených v rámci této práce.

V kapitole je popsán celý proces od načítání dat do aplikace, přes vykreslování prostorů, až po výpočet doletového prostoru. Nakonec se ještě podíváme na otestování aplikace a navrhnutí dalšího pokračování a rozšíření projektu.

4.1 Načítání dat do aplikace

Načítání dat do aplikace je implementováno tak, aby šla aplikace aktualizovat. Stačí tedy nahrát aktuální data s prostory či výšková data ke zdrojovým souborům a aplikace pak data analyzuje.

Samotné načítání dat zajišťuje třída `FileReader.java`, která se vytváří konstruktorem v hlavní aktivitě a otevírá dané soubory. Soubor je načten funkcí `readFile()`, která načítá jako parametr název souboru a ukládá obsah do proměnné `file`, která je typu string.

4.1.1 Načítání a analýza vzdušných prostorů

Vzdušné prostory k vykreslení jsou zpracovány funkcí `getAirspaces()`, kde se pomocí regulárního výrazu `"<trk>(.*?)</trk>"` načítají veškeré informace o jednom prostoru. Tyto prostory jsou dále analyzovány funkcí `parseAirspace()`, která ukládá dané souřadnice, typ a název jako vlastnosti objektu `Airspace`.

4.1.2 Načítání a analýza výškových dat

Jak jsem popisoval na konci kapitoly 2.5, data jsem musel transformovat na zeměpisné souřadnice. Transformovaný soubor však opět nebyl vhodný pro zpracování, protože souřadnice byly ve stupních, minutách a sekundách zeměpisné délky a šířky. Pro můj program bylo tedy nutné převést datový soubor na stupně. K tomuto účelu jsem napsal skript v jazyce Python, abych pak ulevil zpracování již tak náročnému načítání a analyzování dat v aplikaci.

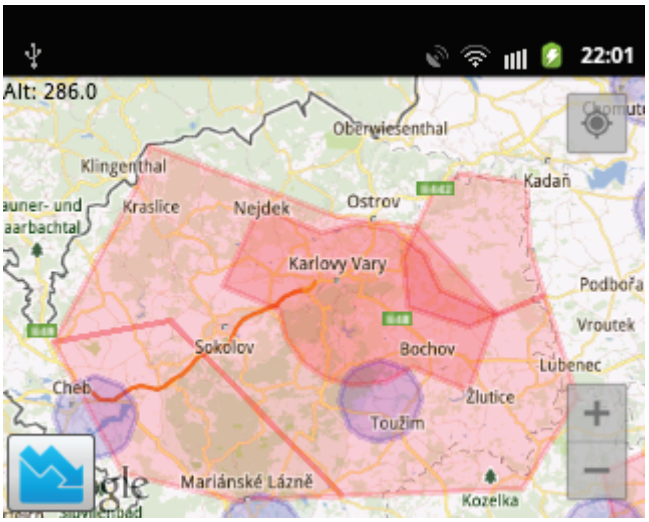
Skript tedy převede data do celých stupňů a připraví vhodný formát pro zpracování. Ke zpracování těchto dat jsem vytvořil funkci `parseElevation()`, která vyhledá pomocí jednoho regulárního výrazu `"S([0-9]+);([0-9]+);([0-9]+)E"` kompletní údaje o zeměpisné šířce, zeměpisné délce a zeměpisné výšce daného bodu. Tyto údaje jsou uloženy v číselném poli typu double.

4.2 Vykreslování objektů do mapového prostoru

Jak jsem popisoval v kapitole 3.3, rozlišuji v mé aplikaci dvojí vykreslování. Jedno se týká vykreslování leteckých vzdušných prostorů a druhé se týká vykreslování doletového prostoru. Vykreslení objektů probíhá stejným způsobem, princip je ale jiný.

4.2.1 Vykreslování leteckých vzdušných prostorů

Všechny letecké vzdušné prostory jsou uloženy v poli `Airspace` objektů, které je celé procházeno v cyklu a v každé iteraci se přistupuje k objektu dle jeho typu. Každý objekt má vlastnost `type`, kde je uložen typ prostoru dle popsané typologie v kapitole 2.4.1. Podle typu prostoru jsou pak definovány vlastnosti výsledného vykreslovaného polygonu. Do vlastností polygonu jsou uloženy všechny souřadnice prostoru, barva ohraničení, síla ohraničení a barva výplně. Tyto prostory jsou tedy v závislosti na typu vykreslovány průhlednou červenou nebo modrou barvou. Průhlednost je dosahována přidáním tzv. alfa složky v ARGB režimu. Průhledný kód červené barvy je pak definován takto `Color.argb(40, 255,0, 0)`, kde první číslo je alfa složka a následuje RGB kód. Na výsledné vykreslení se můžeme podívat na následujícím snímku obrazovky.



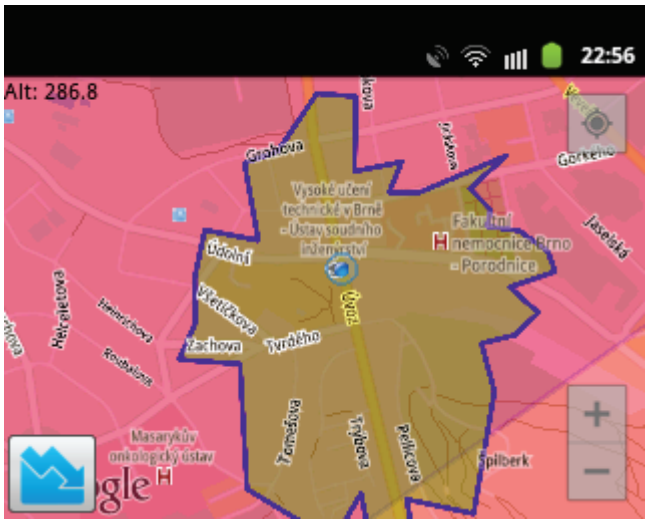
Obrázek 2: Ukázka vykreslených prostorů.

Hlavní červená oblast kolem Karlových varů je CTR Karlovy Vary, větší červená oblast je pak TMA 1 Karlovy Vary a TMA 2 Karlovy Vary a modré prostory jsou prostory kolem neřízených letišť ATZ. K vykreslování těchto prostorů jsem vytvořil funkci `drawAirSpace()`.

4.2.2 Vykreslování doletového prostoru

Výpočet doletového prostoru budu řešit až v kapitole 4.5. Nyní popíši pouze samotné vykreslení prostoru do mapy.

K vykreslení výsledného doletového prostoru jsem vytvořil funkci `drawMarkersFromSortedList()`. Funkce bere jednotlivé prvky seřazených výsledných hraničních bodů z algoritmu vnitřních a vnějších bodů a ukládá jejich souřadnice jako objekt LatLng[3] do vlastností polygonu, který bude výsledně vykreslen. Ohraničení prostoru je definováno modrou neprůhlednou barvou a výplň průhlednou zelenou, aby se prostor odlišil v případě překreslení vzdušného prostoru. Na ukázku se můžeme podívat na snímku obrazovky.



Obrázek 3: Ukázka vykreslení doletového prostoru.

Jedná se o vykreslení teoretického doletu blízko centra Brna. Na mapě je tedy vidět, že se zároveň nacházíme v zakázaném prostoru, konkrétně v CTR Tuřany, a také v oblasti ATZ.

4.3 Optimalizace kódu pro rychlejší běh aplikace

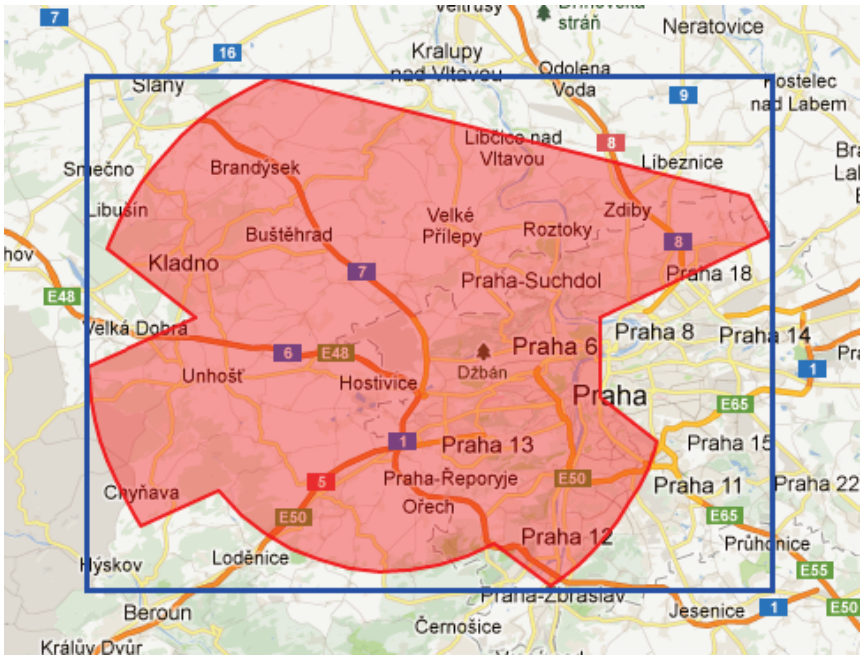
Při vytváření aplikací pro platformu Android musí vývojář také myslet na výkon a dostupné prostředky cílových zařízení. Nové mobilní zařízení již mají sice výkon srovnatelný s přenosnými počítači, ale pro účely mé aplikace chci aplikace optimalizovat tak, aby byla vhodná i pro starší mobilní zařízení s platformou Android.

4.3.1 Vytvoření Bounding Boxu

První problém nastal při vykreslování vzdušných prostorů. Prostory jsou definovány souřadnicemi v textovém souboru a těchto údajů obsahuje soubor opravdu velké množství. Zpracování a překreslování prostorů bylo tedy velmi náročné na výpočet a zpomalovalo celou aplikaci. Jako řešení jsem implementoval tzv. bounding box, který funguje jako hranaté ohraničení daného prostoru a

[3] Zkratky anglických výrazů Latitude a Longtitude znamenající zeměpisnou šířku a délku.

v podmínce pak kontroluji, jestli je tento prostor na obrazovce. Podívejme se, jak vypadá např. bounding box CTR Ruzyně prostoru.



Obrázek 4: Ukázka bounding boxu prostoru CTR Ruzyně.

Z obrázku vidíme, že bounding box je definován 4 body prostoru: nejmenší a největší x-ová a nejmenší a největší y-ová souřadnice. Souřadnicový systém displeje má hodnotu [0,0] v levém horním rohu. Jestli se daný bouding box nachází mimo displej, můžeme tedy zjistit, pokud bude platit jedna z následujících podmínek:

1. *Největší x-ová souřadnice prostoru je menší než 0.*
2. *Nejmenší x-ová souřadnice prostoru je větší než hodnota šířky displeje.*
3. *Nejmenší y-ová souřadnice prostoru je menší než 0.*
4. *Největší y-ová souřadnice je větší než výška displeje.*

Pokud je jedna podmínka pravdivá, prostor musí být mimo displej a v tom případě ho nezpracovávám. Tímto dosáhnu procházení pouze prostorů, které jsou uživateli zobrazeny, a tím zrychlím aplikaci.

4.3.2 Použití vláken

Aplikace při startu zpracovává velké množství dat. Musí naráz zpracovat všechny vzdušné prostory, nahrát soubor s výškovými daty a získat aktuální polohu s výškou. Při výpočtu teoretického doletu pak rychle utřídit seznam bodů a dále získat informace o výšce a poloze. Tyto operace jsou velmi

náročné a vzhledem k povaze aplikace bychom chtěli, aby mohl uživatel hned po startu interagovat s mapou a při výpočtu nemusel příliš dlouho čekat. K tomuto účelu jsem implementoval 4 vlákna.

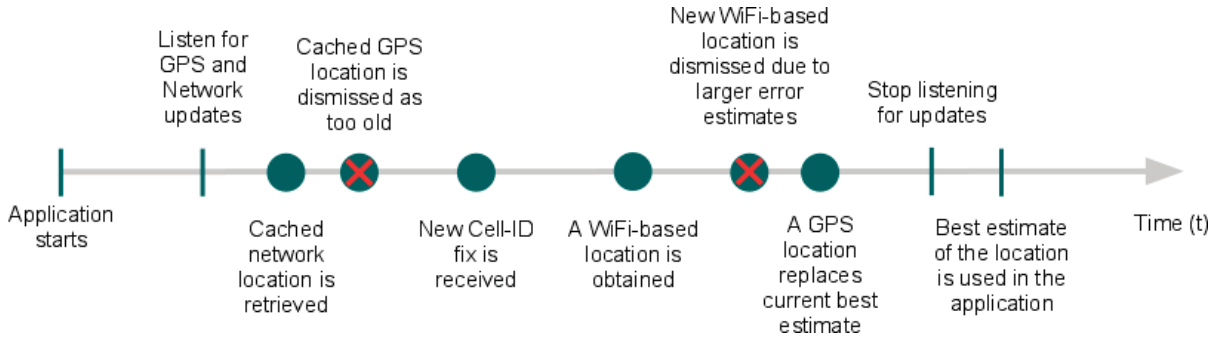
- ParseFileThread – vlákno pro zpracování dat ze souboru a následné vykreslení prostorů
- LoadElevationThread – vlákno pro nahrání a zpracování výškových dat ze souboru
- SortThread – vlákno využité při výpočtu teoretického doletu pro seřazení výsledných hraničních bodů
- GetAltitudeThread – vlákno pro získání aktuální výšky

Podle knihy Android Application Development [8] jsou právě vlákna klíčem k plynulému chodu aplikace, kdy je vyžadováno více akcí naráz. Často hrozí, že se aplikace zasekne vzhledem ke zpracování více požadavků na hlavní vlákno aplikace většinou označované UI[4] Thread. V horším případě může aplikace i spadnout nebo se dostat do stavu „Neodpovídá“. To může nastat, pokud je blokována kolem 5 vteřin.

4.4 Implementace sledování pohybu pomocí GPS

Pro sledování pohybu a pozice zařízení jsem vytvořil třídu `MyLocation.java`. Aplikace musí sledovat a zobrazovat polohu zařízení a dále zjišťovat jeho aktuální výšku. Ke správné funkčnosti je tedy zapotřebí GPS senzor, kterým můžeme zjistit aktuální výšku. Přibližnou polohu můžeme zjistit i z údajů mobilní sítě, výšku ale nikoli.

Pro účely této aplikace jsou však data z mobilní sítě také velmi důležitá. Podívejme se na grafiku, jak vypadá naslouchání získání aktuální polohy v čase.



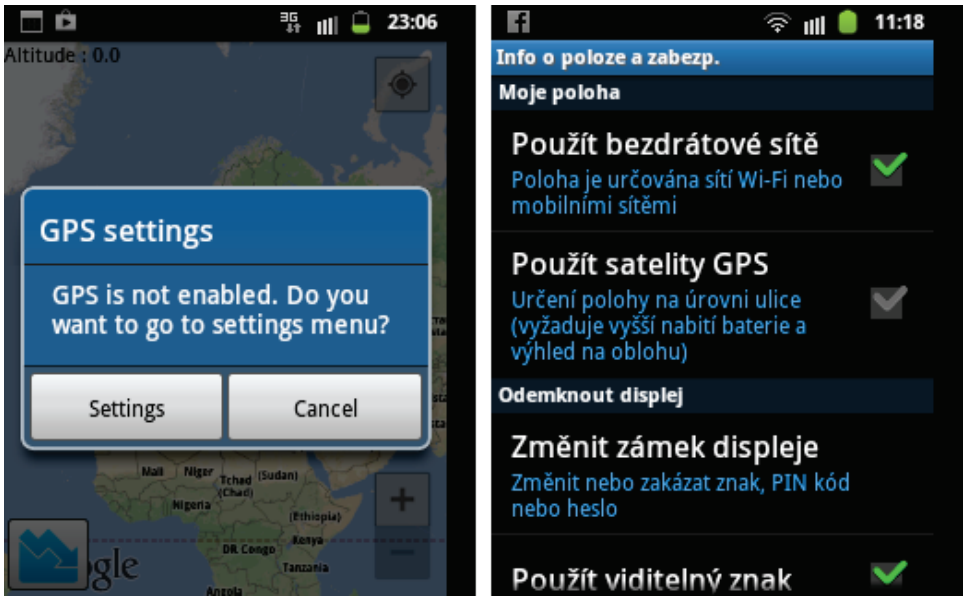
Obrázek 5: Naslouchání pro získání polohy v čase. Obrázek dostupný z http://developer.android.com/guide/topics/location/strategies.html.

Z obrázku vidíme, že první údaj o přibližné poloze získáme právě z mezipaměti mobilní sítě. Na druhé straně údaj o přesné poloze ze senzoru GPS až jako poslední. Mezi aktivací GPS a obdržením správné hodnoty je velká časová prodleva. Z toho důvodu využiji nejdříve přibližný údaj pozice

[4] User Interface = uživatelské prostředí

z hodnot mobilní sítě. Tato pozice je uložena v mezipaměti jako naposled známá poloha a tím získáme přibližnou pozici zařízení. Ve chvíli, kdy se spojíme s GPS a dostaneme přesný údaj, aktualizuji pozici a nadmořskou výšku z těchto obdržených hodnot. Logika naslouchání polohy v mé aplikaci tedy vypadá takto:

1. *Pokud není aktivován senzor GPS v telefonu, aplikace se zeptá na aktivaci v nastavení (viz obr. 6).*
2. *Je aktivována GPS a zaslán dotaz na získání údajů o poloze.*
3. *Pokud není ještě známa poloha z GPS, použije se hodnota z mezipaměti mobilní sítě.*
4. *Pokud není dostupná mobilní síť a ještě neznáme hodnotu pozice z GPS, vypíše aplikace zprávu, že nelze zjistit aktuální polohu.*
5. *Když dorazí správný signál z GPS, aktualizují se hodnoty pozice a na displej se přidá aktuální výška.*
6. *Pokud je výška menší než 115, použije se hodnota z nastavení aplikace.*



Obrázek 6: Dialog pro nastavení GPS.

Měření aktuální nadmořské výšky je občas velmi nepřesné. Pokud se tedy naskytne chyba v údaji o výšce, použije se výška z hodnoty nastavení, která lze zadat ručně. V bodu 6 předchozího algoritmu používám hranici 115 m nad mořem, což je nejnižší nadmořská výška v České republice a lze tedy předpokládat, že pokud je údaj z GPS menší, bude chybný. Uživatel si také může zadat výšku ručně přes nastavení aplikace, pokud má podezření, že mu GPS měří výšku špatně.

4.5 Výpočet doletového prostoru

Výpočet doletového prostoru probíhá v rámci hlavní aktivity a vytvořil jsem k němu několik funkcí, které nyní popíši.

Ve funkci `loadMapwithElevationData()` jsou vyhodnocovány vnitřní body. Vyhodnocení probíhá procházení výškových bodů a kontrolou jejich výšky s teoretickou výškou paraglidisty v tomto bodě. Teoretickou výšku paraglidisty v tomto bodě vypočítáme takto:

```
double futAltide = Currentalt - (results[0] / gliding_value);
```

kde `Currentalt` je aktuální výška paraglidisty, `results[0]` je vzdálenost mezi paraglidistou a výškovým bodem, a `gliding_value` je hodnota klouzavosti. Jestli je bod uvnitř pak vyhodnotím podmínkou `if (futAltide >= newaltitude)`, kde `newaltitude` je výška výškového bodu. Pokud tato nerovnice platí, daný bod je považován za vnitřní a uložen do seznamu vnitřních bodů.

Funkce `makeCircle(double radius)` dostává jako argument rádius a vyhledává nejvíce vzdálené vnitřní body. Hlavní činnost funkce spočívá v nalezení bodů uvnitř rádiu a uložení těchto bodů do pole `points`, které je typu objektu LatLng. Body kolem aktuální pozice se hledají uvnitř těla funkce. Aktuální pozice je převedena na radiány a v cyklu od 0 do hodnoty 2π s přírůstkem t = 0.5 se procházejí a ukládají X a Y souřadnice bodu. Zde vidíme ukázku výpočtu bodu zeměpisné šířky `double latPoint = lat + (radius / EARTH_RADIUS) * Math.sin(t)`, kde konstanta `EARTH_RADIUS` je poloměr zeměkoule s hodnotou 6378100 a `lat` je naše souřadnice zeměpisné šířky.

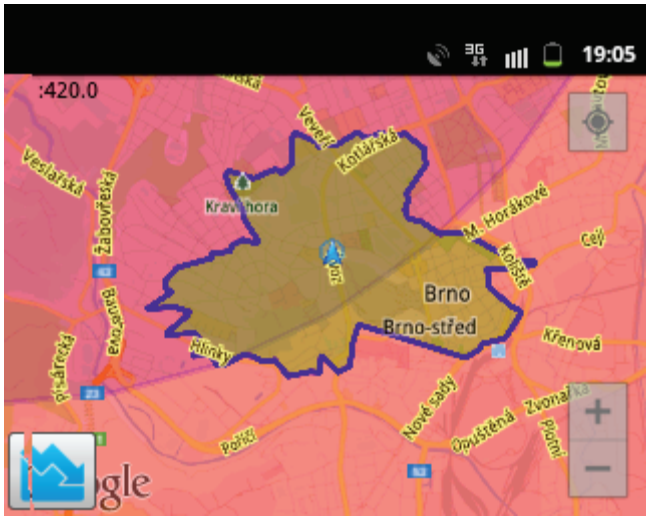
K výpočtu doletového prostoru potřebujeme znát i nadmořskou výšku terénu pod námi. K tomuto účelu bych mohl využít výšková data, ale to by bylo celkem náročné na výpočet a zbytečně by to zatěžovalo výkon aplikace. Rozhodl jsem se tedy pro variantu dotazu na online službu Google Elevation API. Zasílám tedy HTTP GET metodou dotaz na aktuální výšku terénu na moji pozici. Vytvořil jsem funkci `getElevationFromGoogleMaps()`, která má 2 argumenty: zeměpisnou šířku a délku aktuální pozice. Tyto argumenty se posílají GET metodou následovně:

```
String url = "http://maps.googleapis.com/maps/api/elevation/"
                + "xml?locations=" + String.valueOf(latitude)
                + "," + String.valueOf(longitude)
                + "&sensor=true";
HttpGet httpGet = new HttpGet(url);
```

Odpověď dostávám v XML formátu a nadmořská výška je analyzována ze značky <elevation> a uložena.

Body uložené v seznamu je potřeba setřídit tak, aby mohly být poté spojeny a prostor vykreslen. Na setřídění seznamu jsem vytvořil funkci `SortLatlng()`.Funkce seřadí body v seznamu po směru hodinových ručiček. K tomuto seřazení využívá hodnoty funkce `atan2` z knihovny Math a v zanořeném cyklu prochází všechny body v rámci rádiu.

Po uložení a setřídění všech bodů můžeme seznam bodů spojit a vykreslit doletovou oblast ohraničenou těmito body. K vykreslení oblasti slouží funkce `drawMarkersFromSortedList()`, kterou jsem popisoval v kapitole 4.2.2.

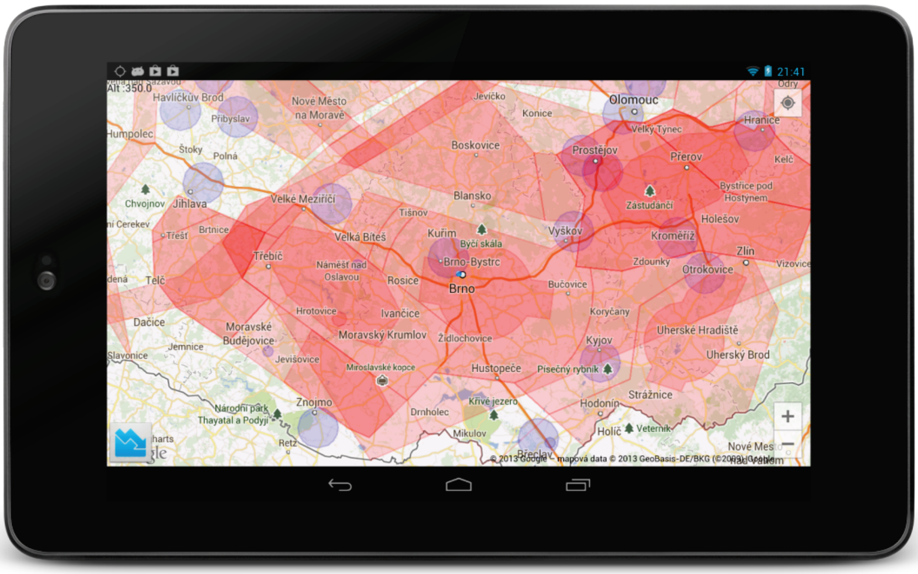


Obrázek 7: Ukázka vykreslení teoretického doletu pro výšku 420 metrů nad mořem a hodnotou klouzavosti 9.

4.6 Uživatelské prostředí mapové aplikace

Uživatelské prostředí je vyvíjeno jako klasická mapová aplikace s prvky z klasické aplikace, aby byla zajištěna přehlednost a celkové uživatelovo pohodlí s ohledem na podmínky využití aplikace. Paraglidista mívá většinou ve vzduchu rukavice a tím se znatelně omezuje použití klasických uživatelských prvků aplikace.

Kromě nastavení hodnoty klouzavosti, kterou si uživatel nastaví na začátku dle svého kluzáku, již pak nemusí zadávat žádné údaje. Vše je počítáno automaticky a paraglidista si může posouvat, oddalovat nebo přibližovat mapu a když si chce vykreslit doletový prostor, jednoduše klikne na tlačítko umístěné v dolním rohu aplikace.



Obrázek 8: Ukázka uživatelského prostředí na tabletu Nexus 7.

4.6.1 Hlavní okno aplikace

Uživatelské prostředí hlavního pohledu aplikace je implementováno jako xml soubor v Eclipse SDK prostředí. Tedy standardní cestou, jak se vytváří rozložení uživatelského rozhraní na platformě Android. Soubor main.xml je řešen skrze relativní pozicování kořenové párové značky `<RelativeLayout></RelativeLayout>` jednotlivých komponent uživatelského prostředí. Pozice každého prvku může být specifikována jako relativní k sourozeneckému elementu, jako např. umístění vlevo nebo pod tímto elementem, nebo naopak jako relativní k rodičovskému elementu (např. umístění dole či vlevo od středu pohledu).

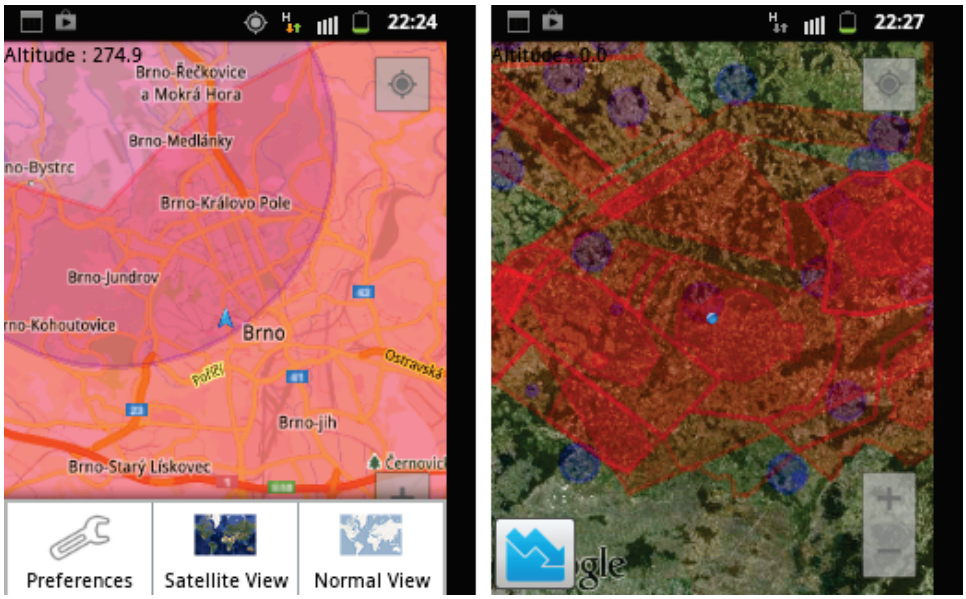
V mém souboru využívám vnořených elementů RelativeLayout, tlačítka Button a jednoho textového pole TextView. První vnořený RelativeLayout element obsahuje tlačítko s ID `"@+id/drawLanding"`, které slouží pro vykreslení teoretického doletu. Toto tlačítko je zarovnáno relativně vzhledem k rodičovskému elementu příznakem `android:layout_alignParentBottom="true"`. V tlačítku je pak využita ikonka, která je uložena ve standardní složce pro ukládání zdrojů: res/drawable.

Textový pohled pro zobrazení aktuální výšky paraglidisty je nastaven příznakem `android:layout_alignParentLeft="true"` a je tedy umístěn v levém horním rohu. Další tlačítka jsou standardní tlačítka Google Maps API a jsou řešeny v kódu hlavní aktivity. Jedná se o tlačítka pro vycentrování obrazovky animováním na aktuální polohu a kontrolního panelu pro přiblížení a oddálení mapy.

Obsluha tlačítek a hodnot z tlačítek pak zajišťuje hlavní aktivita `MainActivity.java`. Pomocí ID z xml souboru je nalezeno příslušné tlačítko a v aktivitě je mu nastavena buď příslušná funkce, nebo uložena hodnota klouzavosti z uživatelova vstupu.

4.6.2 Nabídka nastavení aplikace

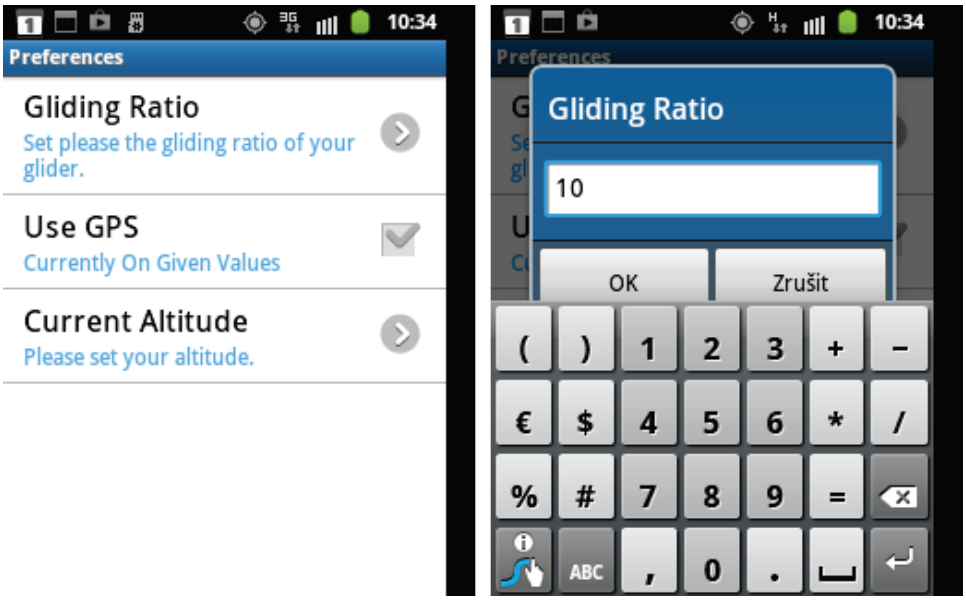
Aplikace poskytuje také menu pro nastavení a konfiguraci. Nabídka se vyvolá tlačítkem Menu a obsahuje 3 tlačítka: Preferences, Satellite View a Normal View. Tlačítkem Preferences se uživatel dostane do nové aktivity, kde si může nastavit hodnotu klouzavosti jeho kluzáku, vybrat zda chce používat hodnoty z GPS nebo nastavit výšku ručně. Řešení novou aktivitou jsem zvolil pro kompatibilitu s novými verzemi Android. Nová zařízení již totiž nemívají tlačítko Menu a tak si uživatelé na těchto zařízeních mohou nabídku vyvolat ikonkou Preferences vedle hlavní spouštěcí ikony aplikace. Tlačítka Satellite a Normal View pak přepínají zobrazení Google Maps do satelitního nebo normálního pohledu.



Obrázek 9: Ukázka vyvolání menu a satelitního pohledu.

Jednotlivé položky jsou uloženy v souboru menu.xml, který se nachází ve standardním umístění res/menu. V kořenovém párovém elementu `<menu></menu>` jsou uloženy položky v nepárovém elementu `<item/>`. Základ jednotlivých položek menu tvoří opět ikonky uložené ve zdrojích. Popisky jsou odkazovány přes soubor strings.xml. Soubor strings.xml obsahuje samotné řetězce a umožňuje tak snadnou lokalizaci řetězců.

Po kliknutí na tlačítko Preferences se zobrazí menu s nastavením hodnoty klouzavosti a vybráním automatického či ručního ukládání výšky.



Obrázek 10: Snímek obzrazovky Preferences a nastavení hodnoty klouzavosti.

Toto menu je vytvořeno v souboru preferences.xml umístěném ve složce res/xml. Využil jsem standardního předdefinovaného elementu `<PreferenceScreen> </PreferenceScreen>`. Pro nabídku vložení hodnoty klouzavosti jsem využil element `<EditTextPreference>`, který umožňuje zadat pouze číselnou hodnotu a obsahuje předdefinovanou hodnotu 10. Pro přepínač mezi automatickým ukládáním výšky z GPS a ručním zadáváním je vytvořen element `<CheckBoxPreference/>`. Poslední nabídkou je možnost vložení hodnoty výšky, která je vytvořena stejně jako dotaz na klouzavost. Hodnoty jsou také předávány a ukládány v hlavní aktivitě MainActivity.java a dále použity při výpočtu teoretického doletu.

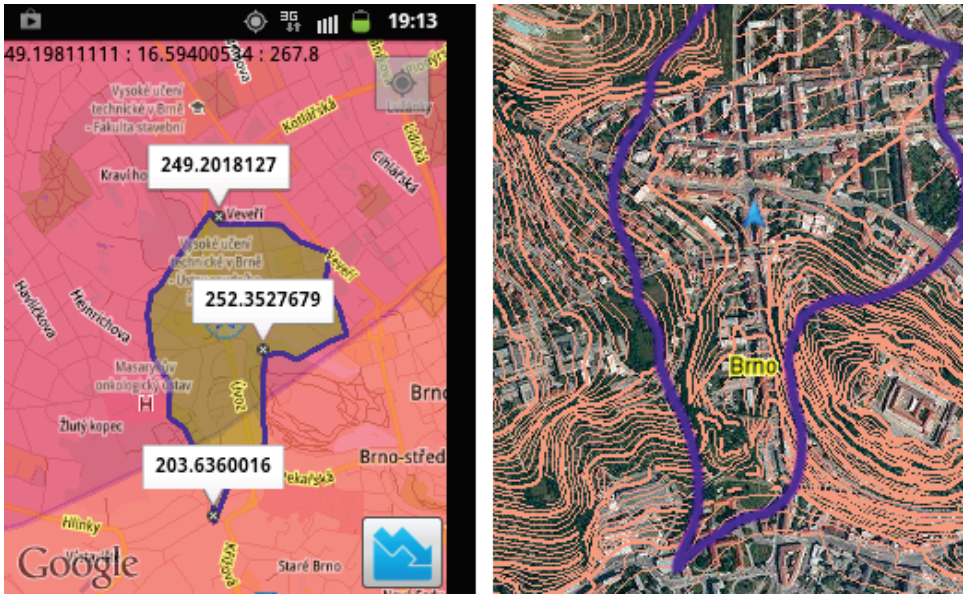
4.7 Otestování aplikace

Aplikace byla otestována na různých zařízeních a Android verzích. Na všech zařízeních vykazovala stejnou funkčnost, pouze rozdílnou výkonnost vzhledem k dostupným prostředkům cílového zařízení. Aplikace byla vyvíjena pro Android verze 2.3.3 a se zajištěním kompatibility pro vyšší verze. Testována byla na verzích 2.3.3 a 2.3.6 s pojmenováním Gingerbread na zařízení Samsung Galaxy Mini a Samsung i9003 Galaxy SL, na verzi 4.2.2 s pojmenováním Jelly Bean na mobilním telefonu Samsung Note 2 a tabletech Nexus 7 a Acer A100.

4.7.1 Správnost výpočtů doletového prostoru

Správnost výsledků výpočtu teoretického doletu vyplývá z implementace, pro názornost jsem ještě měřil pomocí vzdálenosti na Google Maps. Podívejme se nyní na testovací snímek obrazovky, kde

jsem do aplikace přidal vyznačení některých výškových bodů, abychom mohli porovnat správnost naměřených údajů.



Obrázek 11: Porovnání výsledků doletu s mapou vrstevnic na stejném území.

Na levém snímku obrazovky vidíme testovací verzi aplikace, kde jsem si nechal vykreslit některé výškové body a jejich hodnoty výšky. Na obrázku vedle vidíme stejné území ve vrstevnicích (zdroj http://geoportal.cuzk.cz/geoprohlizec/) a přibližně označené vykreslené území s naší polohou. Snímek z aplikace je vytvořen pro výšku 290 metrů nad mořem a klouzavosti kluzáku 10. Z měření vzdálenosti na Google Maps jsem dostal ke spodnímu bodu vzdálenost 857 m, k prostřednímu 365 m a k hornímu 398 m. Když si nyní vypočítáme možnou dráhu k těmto bodům a výšku paraglidisty v tomto bodě, mělo by nám vyjít přibližně stejné číslo nebo větší číslo. Jedná se o vnitřní body, takže k nim by se měl paraglidista určitě dostat, díky nepřesnostem by mohl doletět i o pár metrů dále.
Rovnice pro dolní bod:

$$k = \frac{\Delta s}{\Delta h}$$

$$10 = \frac{s}{290 - 203{,}64}$$

$$s = 863{,}6\ m$$

Pro prostřední bod již přímo dosazení:

$$s = 10 * (290 - 252{,}35)$$

$$s = 376{,}5\ m$$

Pro horní bod to samé:

$$s = 10 * (290 - 249{,}2)$$
$$s = 408\,m$$

Z výpočtu vyplývá, že paraglidista se dostane na danou výšku jednotlivých bodů za uraženou dráhu *s*. Pokud tedy chceme, aby paraglidista doletěl dále nebo přesně na definovaný výškový bod, musí být dráha *s* větší nebo rovna vzdálenosti od paraglidisty k bodu, označme s_0. Vzhledem k možnosti výskytu chyb jako jsou nepřesnosti při získávání výšky z GPS, nedostatečné množství výškových dat a výsledky vyhlazovací funkce, musíme počítat s drobnými odlišnostmi od reálné situace. Pojďme se podívat na výsledky testu. Dosazuji hodnoty do nerovnice $s \geq s_0$.

1. Spodní bod: $863{,}3 \geq 857$
2. Prostřední bod: $376{,}5 \geq 365$
3. Horní bod: $408 \geq 398$

Vidíme, že nerovnice platí a výsledky přibližně odpovídají. V tomto případě by tedy paraglidista doletěl o pár metrů dále za hranici vyznačeného prostoru a to je očekávané chování.

4.7.2 Náročnost aplikace na výpočet

Časová náročnost aplikace narůstá s výškou paraglidisty. Čím výše je paraglidista, tím se prohledává větší prostor a řadí, vyhodnocuje a zpracovává se větší množství bodů. Zvolil jsem tedy stejnou výšku a lokalitu a vypočítal časovou náročnost na různých zařízeních. Výšku jsem zvolil 500 metrů nad mořem, lokalitu na adrese Úvoz, Brno.

Zařízení	Android verze	Procesor	Vykreslovací čas
Samsung Galaxy Mini	2.3.6	600 Mhz	26 s
Samsung Galaxy SL	2.3.6	1 GHz Cortex-A8	15,6 s
Nexus 7	4.2.2	1.3 GHz Cortex-A9, čtyři jádra	3,6 s
Samsung Note 2	4.2.2	1,6 Ghz Cortex-A9, čtyři jádra	3,1 s

Tabulka 2: Srovnání testovacích zařízení.

Aplikace tedy úspěšně funguje na různých zařízeních s různou verzí Android systému, ale jak vidíme z tabulky, není vhodná pro starší typy telefonů s nízkým výpočetním výkonem. Čas 26 s je příliš

dlouhý na komfortní práci s aplikací. Podpora starších zařízení s nízkým výkonem by se mohla vyladit v rámci dalších rozšíření aplikace.

4.8 Možnosti rozšíření a pokračování projektu

Pro možnost pokračování tohoto projektu je prvně potřeba sehnat kompletní výšková data České republiky. Pro účely této práce mi byla poskytnuta ukázka výškových dat a tím pádem je aplikace pouze ukázkou možného řešení, nejedná se o kompletně funkční aplikaci. S kompletními daty by tedy byla aplikace vhodná pro použití v rámci České republiky.

Možné rozšíření aplikace by mohlo např. spočívat v ukládání informací o letu a zveřejňování těchto informací na stránkách komunity. Informace jako celková doba letu, průměrná rychlost, maximální výška, průměrná výška paraglidisty nebo by aplikace mohla ukládat i celkovou dráhu letu a vykreslit ji do mapy. Tyto informace by aplikace ukládala do osobního profilu na webových stránkách komunity paraglidistů, kde by mohli uživatelé vkládat k aktivitě komentáře apod. Uživatel by mohl svůj let sdílet na sociálních sítích. V aplikaci by si pak uživatel mohl zobrazit na mapě např. své přátele z komunity a sledovat jejich let. Užitečným vylepšením by mohlo být ovládání hlasem. To by uživateli poskytovalo větší komfort při používání aplikace.

Dále by šla aplikace rozšířit pro další letecké prostředky jako ultralehká letadla, menší letadla či soukromé helikoptéry. V nastavení by si uživatel vybral svůj letecký prostředek a podle toho by se aplikace chovala. Pro letadla by se mohla implementovat funkce, která by letadlo intuitivně naváděla na přistávací dráhu, která je u každého letiště jinak orientovaná. Tato dráha by mohla být zakreslena do mapy a přidán k ní směr, kterým se má letadlo pohybovat.

Jako velký projekt by se mohlo pojmout rozšíření aplikace pro globální použití. Bylo by potřeba sehnat výšková data v zemích, pro které by byla aplikace vyvíjena. Data by se uložila na server a uživatel by si při instalaci vybíral, pro kterou zemi bude aplikaci používat. Ze serveru by se pak příslušná data stáhla a používala. Další alternativou by byl převod aplikace k využívání dat z Google služeb. Místo uložených výškových dat by se prováděl webový dotaz na výšku a veškerá komunikace by tedy probíhala online.

5 Závěr

Výsledkem této práce je funkční aplikace pro platformu Android. Aplikace Navigace pro paragliding umožňuje sledovat paraglidistův pohyb, vykreslovat prostory, kam nesmí letět nebo kde je průlet omezen, a dokáže reflektovat terénní situace v daném okolí a vykreslit doletový prostor.

Před návrhem a implementací aplikace předcházelo nastudování vytváření mapových aplikací na platformě Android a zvyklostí leteckého vzdušného prostoru. Zjistil jsem, jak funguje řízení letového provozu v České republice a jak je definována typologie leteckých vzdušných prostorů. Tyto poznatky jsem přenesl do aplikace, díky které uživatel může na mapě rozeznat zakázané a omezené prostory.

Aplikace také poskytuje funkci pro vypočítání a vykreslení teoretického doletového prostoru. Tato funkce je omezena pouze pro určitou lokalitu, konkrétně střed Brna. Je to dáno chybějícími výškovými daty. Tento vzorek dat pro Brno mi poskytl Český úřad zeměměřický a katastrální pro účely bakalářské práce zdarma. Pro úplnou funkčnost aplikace v rámci České republiky by bylo zapotřebí zakoupit kompletní výšková data.

Aplikace byla otestována na více mobilních zařízeních a více verzích platformy Android. Na každém zařízení vykazovala správnou funkčnost a byla by tedy připravená k ostrému provozu

Po implementaci některých popsaných rozšíření by se dala aplikace prodávat přes Google Play. Výhodou je unikátnost této aplikace, na marketu jsem nenašel podobnou aplikaci. Nevýhodou je relativně úzká cílová skupina, která by šla však rozšířit navrhovaným rozšířením a pokračováním projektu. Vývoj aplikací pro platformu Android má jistě velkou budoucnost a s neustálým vývojem technologií se pro vývojáře naskytují nové možnosti a výzvy, které s radostí přijmeme.

Literatura

[1] TABOOLA. Google Buys Android for Its Mobile Arsenal. *Businessweek* [online]. 2005 [cit. 2013-04-05]. Dostupné z: http://www.businessweek.com/stories/2005-08-16/google-buys-android-for-its-mobile-arsenal

[2] HARBIN, Nathan. Global Android Activations Rocket to 2 Million Per Day. In: *Android Headlines* [online]. 2013 [cit. 2013-04-05]. Dostupné z: http://androidheadlines.com/2013/03/global-android-activations-rocket-to-2-million-per-day.html

[3] PAGE, Larry. Update from the CEO. In: *Google Official Blog* [online]. 2013 [cit. 2013-04-05]. Dostupné z: http://googleblog.blogspot.co.uk/2013/03/update-from-ceo.html

[4] RAMŠAK, Matjaž. Radio Controlled Sailplane Flight: Experimental and Numerical Analysis. *Strojniški vestnik - Journal of Mechanical Engineering* [online]. 2012, roč. 58, č. 3, s. 147-155 [cit. 2013-04-08]. ISSN 00392480. DOI: 10.5545/sv-jme.2009.153. Dostupné z: http://en.sv-jme.eu/data/upload/2012/03/01_2009_153_Ramsak_04.pdf

[5] Česká republika. Klasifikace vzdušného prostoru ATS. In: *Letecká informační príručka*. 2006. Dostupné z: http://lis.rlp.cz/ais_data/www_main_control/frm_cz_aip.htm

[6] Česká republika. Vzdušný prostor letových provozních služeb. In: *Letecká informační príručka*. 2010. Dostupné z: http://lis.rlp.cz/ais_data/www_main_control/frm_cz_aip.htm

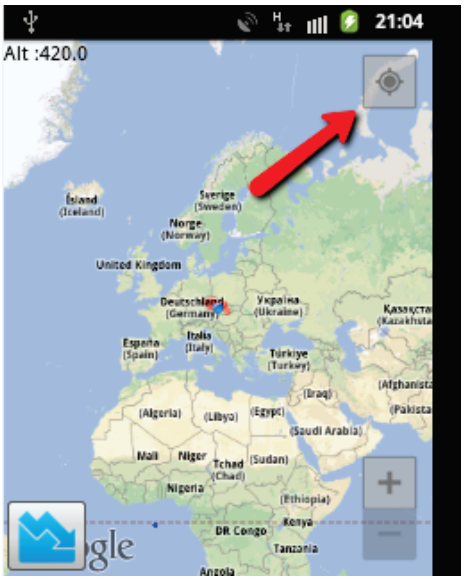
[7] Google Maps Android API v2. GOOGLE INC. *Google Developers* [online]. 2013, 26. 2. 2013 [cit. 2013-04-11]. Dostupné z: https://developers.google.com/maps/documentation/android/

[8] ROGERS, Rick. *Android application development*. 1st ed. Sebastopol, Calif.: O'Reilly, c2009, xiii, 318 p. ISBN 05-965-2147-2.

[9] BURNETTE, Ed. *Hello, Android: introducing Google's mobile development platform*. 3rd ed. Raleigh, N.C: Pragmatic Bookshelf. ISBN 19-343-5656-5.

Seznam příloh

Příloha 1: Uživatelský manuál



Po startu aplikace se zobrazí mapa světa s vykreslenými prostory v České republice. Pro správnou funkčnost je potřeba mít zapnutou GPS a aktivovanou mobilní datovou síť. Aplikace funguje i pro Wi-Fi připojení, ale zajišťování polohy funguje lépe s mobilní datovou sítí. Pro automatické zaměření na aktuální polohu klikněte na tlačítko umístěné v horním pravém rohu (viz obrázek).

Do nastavení aplikace se dostanete přes menu tlačítko a kliknutí na Preferences. Na novějších verzích systému Android lze nastavení Preferences spustit přes ikonu Preferences vedle hlavní spouštěcí ikony aplikace. V nabídce lze pak zadat hodnota klouzavosti, vybrat, zda se má používat výška z hodnoty GPS nebo hodnota ručního zadání. Pokud je tedy nabídka Use GPS nezaškrtnutá, bere se jako výšková hladina hodnota z pole Altitude v Preferences. Kdykoli po kliknutí na levé tlačítko se vykreslí aktuální doletový prostor do mapy. V případě výšky nad 500 metrů může trvat výpočet déle především na zařízeních s nižším výkonem.

Pokud se na displeji objeví zpráva „There is no elevation data in your location.“, znamená to, že se nacházíte mimo prostor, pro který jsem dostal výšková data. Pokud se objeví zpráva „You are too low, there is not enough elevation points“, znamená to, že aktuální výška je příliš nízká k vykreslení doletové oblasti. Pokud se zobrazí popisek „Count: 0“, tak je problém s lokalizováním vašeho zařízení. Tento problém nastává při použití aplikace pouze s GPS a Wi-Fi připojením, kdy se nedá načíst poloha z mobilní datové sítě a GPS signál je chybný.

Příloha 2: Obsah CD

Obsah CD je rozdělen do adresářů

- apk – zkompilovaná aplikace připravená k nainstalování
- src – zdrojové souboru k projektu v Eclipse SDK a soubor readme.pdf s návodem ke kompilaci aplikace
- text – technická zpráva v PDF formátu
- word – zdrojový soubor technické zprávy ve formátu MS Word