

Geolokalizační hra pro platformu Android

GPS-based game for Android platform

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Petr Sklenička**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Geolokalizační hra pro platformu Android
GPS-based game for Android platform**

Zásady pro vypracování:

Cílem práce je vytvořit herní server a mobilní aplikaci pro hru, využívající zjišťování polohy pomocí "Global Positioning System" (GPS) popř. prostřednictvím dat z bezdrátových sítí. Hra bude skloubením principů známého radioamatérského "honu na lišku" a deskové hry, známé v České republice jako "Fantom staré Prahy", tedy stíháním unikajícího "zločince" (jehož poloha je známa jen občas) skupinou "detektivů". Hra bude poskytovat různé úrovně obtížnosti a možnosti informování o poloze hráčů prostřednictvím centrálního serveru, s možností vizualizace polohy hráčů.

Pokyny k vypracování:

1. Nastudujte si problematiku tvorby her na mobilních zařízeních a problematiku her, využívajících určování polohy
2. Navrhněte vhodný způsob komunikace se serverem. V případě socketové komunikace popište navržený aplikační komunikační protokol.
3. Hru implementujte na mobilní platformě Android ve verzi 2.1 a vyšší, herní logiku oddělte od GUI, aby byl usnadněn případný přenos na další platformy.
4. Otestujte hru simulací pohybu hráčů a bude-li to možné, i v reálném provozu.

Seznam doporučené odborné literatury:

1. Burnette, E. Hello, Android: Introducing Google's Mobile Development Platform. Pragmatic Bookshelf, 2008. ISBN: 978-1-93435-617-3.
2. Meier, R. Professional Android 2 Application Development. Wrox Press Ltd., 2010. ISBN: 978-0-47056-552-0.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Pavel Moravec, Ph.D.**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 2. května 2012

Y. A. Čížek
.....

Na tomto místě bych rád poděkoval Ing. Pavlu Moravcovi, Ph.D. za jeho cenné rady a čas, který si na mě při tvorbě této práce udělal. Dále bych chtěl poděkovat Ing. Michalu Krumníkovi, který mě v rámci předmětu Tvorba aplikací pro mobilní zařízení II naučil mnoho věcí, které jsem při tvorbě aplikace použil.

Abstrakt

Tato práce se zabývá operačním systémem Android a dále vývojem hry s názvem Catch the Phantom, která je určena pro mobilní telefony a tablety s operačním systémem Android ve verzi 2.1 nebo vyšší. Hra využívá principu klient-server architektury a proto je složena ze dvou částí, konkrétně z herního serveru a samotné mobilní aplikace. Server je naimplementován ve skriptovacím programovacím jazyce PHP a mobilní aplikace je napsána v jazyce Java. Catch the Phantom je hra, která jako svůj stěžejní prvek využívá systém GPS. Podstata této hry je založena na skloubení principů her „Fantom staré Prahy“ a „Hon na lišku“.

Klíčová slova: Android, GPS, hra, Mapsforge, Google Maps, fantom

Abstract

The thesis deals with the Android operating system and the development of a game called „Catch the Phantom“, which is intended for the cell phones and tablets with Android operating system 2.1 or higher. The game uses the client-server architecture and consists of two parts - the game server and the mobile application itself. The server is implemented in PHP scripting and programming language and the mobile application in Java. „Catch the Phantom“ is a game which uses GPS and geolocation as a crucial element. The game was created by combining the principles of the games „Phantom of old Prague“ and Radio fox hunt.

Keywords: Android, GPS, game, Mapsforge, Google Maps, phantom

Seznam použitých zkratk a symbolů

CSS	– Cascading Style Sheets
DVM	– Dalvik Virtual Machine
GPS	– Global Positioning System
GUI	– Graphical User Interface
HTML	– HyperText Markup Language
HTTP	– Hypertext Transfer Protocol
JVM	– Java Virtual Machine
OS	– Operační systém
SDK	– Software Development Kit
SHA	– Secure Hash Algorithm
SQL	– Structured Query Language
TCP	– Transmission Control Protocol
UDP	– User Datagram Protocol
UML	– Unified Modeling Language
USB	– Universal Serial Bus
aapt	– Android Asset Packaging Tool
adb	– Android Debug Bridge
ddms	– Dalvik Debug Monitor Service
dx	– Dalvik Cross-Assembler

Obsah

1	Úvod	6
2	Operační systém Android	8
2.1	Obecný pohled	8
2.2	GPS hry a aplikace pro OS Android	9
2.2.1	Geocaching	9
2.2.2	Tourality	9
2.2.3	Google Latitude	10
2.2.4	Foursquare	10
2.3	Architektura systému Android	10
2.4	Vývoj aplikací pro platformu Android v jazyce Java	11
2.4.1	Android SDK	11
2.4.2	Android Manifest	12
3	Pravidla a popis hry Catch the Phantom	14
3.1	Základní princip hry	14
3.2	Vytvoření hry a volba fantoma	14
3.3	Připojení ostatních hráčů do hry	14
3.4	Průběh hry	15
3.5	Ukončení hry	16
4	Analýza	17
4.1	Herní server	17
4.1.1	Komunikace s mobilní aplikací	17
4.1.2	Funkčnost webového rozhraní	18
4.2	Mobilní aplikace	19
4.2.1	Uživatelské rozhraní	19
4.2.2	Herní logika	19
4.2.3	Komunikace se serverem	20
4.2.4	Volba mapového podkladu	21
4.2.5	Projekt Mapsforge	22
5	Implementace	23
5.1	Mobilní aplikace	23
5.1.1	Logické rozdělení zdrojového kódu	23
5.2	Server	24
5.3	Problémy při implementaci	25
5.3.1	Synchronizace herního času jednotlivých hráčů	25
5.3.2	Problém se změnou orientace displeje	26
5.3.3	Problém ztráty připojení k Internetu	26
6	Testování	27
6.1	Testování na reálných zařízeních	27

7 Závěr	29
8 Reference	30
Přílohy	30
A Uživatelská příručka	31
A.1 Nasazení serveru	31
A.2 Instalace a spuštění mobilní aplikace	32
A.2.1 Instalace aplikace na fyzické zařízení	32
A.2.2 Instalace aplikace do emulátoru	33
A.3 Popis aplikace	33
A.4 Popis herní obrazovky	35
B Obsah přiloženého CD	37

Seznam tabulek

1	Nejznámější verze Androidu	8
2	Podíl mobilních OS na trhu v listopadu 2011 [3]	9
3	Mobilní telefony k testování aplikace	27

Seznam obrázků

1	Architektura systému Android [2]	11
2	Výsledný třídní diagram herní logiky	20
3	Ukázka mapy při použití Google Maps	22
4	Ukázka mapy při použití Mapsforge	22
5	Přihlašovací obrazovka	34
6	Registrace uživatele	34
7	Hlavní menu aplikace	34
8	Vytvoření nové hry	34
9	Ukázka ze hry z pohledu fantoma	35
10	Ukázka hry z emulátoru s verzí Androidu 4.0	36

Seznam výpisů zdrojového kódu

1	Ukázka souboru <code>AndroidManifest.xml</code>	13
2	Metoda, která předá data na server a vrátí odpověď serveru	21
3	Soubor pro přidávání herních serverů	31

1 Úvod

V dnešní době zaznamenáváme enormní pokrok informačních technologií v různých oblastech. Jednou z těchto oblastí jsou mobilní telefony a přenosná zařízení obecně. Nahlédneme-li do historie o pár let zpátky, musíme zcela jistě souhlasit s tvrzením, že technologie mobilních telefonů velmi pokročila a dnes nám mobilní telefony nabízejí mnohem více funkcí a možností, než před několika lety. Jen pro představu se podíváme do roku 1983, kdy byl oficiálně představen první mobilní telefon DynaTAC 8000X. Jednalo se o mobilní telefon, jehož šířka byla 430 milimetrů, výška 330 milimetrů a tloušťka 890 milimetrů [1]. Pokud si určíme průměrnou tloušťku dnešních mobilních telefonů jako 1 centimetr, zjistíme, že bychom na sebe museli položit téměř devět takovýchto zařízení, aby jejich výsledná tloušťka byla stejná jako u telefonu DynaTAC 8000X. Tento krátký ilustrativní příklad jasně dokazuje, že ve směru vývoje přenosných zařízení je pokrok opravdu velký. Je ale nutno dodat, že i s ohledem na parametry telefonu DynaTAC 8000X, které se dnes mohou jevit jako úsměvné, slavilo toto zařízení ve své době nevídaný úspěch.

Začátkem devadesátých let se objevily SMS zprávy a telefony, které je uměly posílat. Jednalo se však o telefony, které kromě posílání SMS zpráv a volání neuměly nic výjimečného.

Vrátíme-li se po krátkém historickém náhledu do současnosti, můžeme tvrdit, že jsme již poměrně zvyklí používat mobilní telefony nejen k zasílání krátkých textových zpráv či k telefonování, ale i k mnoha dalším činnostem, jakými jsou například vyhledávání spojení hromadné dopravy, pohybování se po Internetu, poslouchání muziky, hraní her a využívání GPS. Tento výčet činností samozřejmě není a patrně ani nemůže být kompletní, jelikož každé dnešní zařízení nabízí nepatrně odlišné funkce. Jednotliví výrobci se snaží do svých zařízení přidávat stále nové prvky, díky nimž se od sebe různé typy telefonů liší, nicméně výše uvedené činnosti jsou dnes jakýmsi základem každého modernějšího telefonu. Za poslední dobu se pro takovéto telefony vžil název *chytré telefony*.

Vzhledem k tomu, že v současnosti je na trhu nepřehledné množství telefonů nabízejících různé funkce a možnosti, jsme mobilní telefony schopni dělit do skupin podle různých kritérií. Pokud se na telefony podíváme z hlediska jejich operačního systému, výsledkem budou telefony s operačním systémem Symbian, Windows Mobile, Bada, iOS, MeeGo, Android a případně ještě další méně známé operační systémy mobilních telefonů. Ze zmíněných platforem zažívá v poslední době největší vzestup operační systém Android. Díky tomu je i hra Catch the Phantom, kterou se tato práce zabývá, naimplementována pro tuto platformu.

Stěžejním prvkem této hry je GPS (Global Positioning System), což je globální družicový polohový systém, díky němuž je možno s poměrně velkou přesností určit polohu, kde se zařízení s tímto systémem nachází. Množství dostupných her pro OS Android, ať už placených či neplacených, je v dnešní době opravdu velké. Méně už je ale těch her, které využívají GPS, a proto má hra Catch the Phantom potenciál stát se něčím novým, nebo alespoň něčím, co nebude mít příliš velkou konkurenci v podobě stejných nebo velmi podobných her.

Cílem této hry, jak již její samotný název napovídá, je dopadení fantoma. Hra vychází ze skloubení principů her Fantom staré Prahy a Hon na lišku. Ani jedna z těchto her samozřejmě není určena pro platformu Android, nicméně převzetím některých základních principů těchto her a přidáním nápadu naimplementovat hru jako aplikaci pro mobilní zařízení vznikla myšlenka vyvinout hru Catch the Phantom.

Následující text je možno pomyslně rozdělit na tři části. V první části, kterou tvoří kapitola 2, si popíšeme operační systém Android a nastíníme si základní prvky, bez kterých se při implementaci aplikací pro tuto platformu neobejde žádný programátor. Zmíníme se i o aplikacích, které využívají GPS jako svůj stěžejní prvek. V další části, která se skládá z kapitol 3 až 5, se budeme zabývat vývojem aplikace Catch the Phantom. Poslední část textu je tvořena kapitolou 6 a 7. V této části budou představeny výsledky testování hry v reálném provozu a poté stručné shrnutí celé práce.

2 Operační systém Android

2.1 Obecný pohled

Když se řekne slovo Android, není tím myšlen pouze operační systém. Jedná se o rozsáhlou open source platformu, která je určena především pro mobilní telefony, nicméně Android nachází využití i v některých navigacích a tabletech. Zahrnuje v sobě operační systém, který je založen na linuxovém jádře verze 2.6, middleware a aplikace.

Původně byl Android vyvíjen společností Android Inc., která ale byla v srpnu roku 2005 koupena společností Google Inc. Dále bylo založeno konsorcium OHA (Open Handset Alliance). Cílem tohoto konsorcia je vytvářet nové standardy vývoje softwaru a hardwaru u mobilních zařízení. V současné době s OHA spolupracuje drtivá většina výrobců mobilních telefonů, výjimku tvoří pouze společnosti Nokia a Apple [2].

První verze Androidu byla oficiálně představena v září roku 2008. Od té doby až do současnosti prošel Android různými verzemi a nejznámější z nich zachycuje tabulka 1. Tabulka neobsahuje verzi 5.0, která v současné době ještě není dostupná, ale příchod této verze je očekáván ve druhém čtvrtletí roku 2012. Tato verze by měla nést kódové označení Jelly Bean.

Verze	Kódové označení	Měsíc, rok
1.5	Cupcake	Květen, 2009
1.6	Donut	Říjen, 2009
2.0	Eclair	Říjen, 2009
2.1	Eclair	Leden, 2010
2.2	Froyo	Květen, 2010
2.3	Gingerbread	Prosinec, 2010
3.0	Honeycomb	Únor, 2011
3.1	Honeycomb	Květen, 2011
3.2	Honeycomb	Červen, 2011
4.0	Ice Cream Sandwich	Říjen, 2011

Tabulka 1: Nejznámější verze Androidu

Porovnáme-li podíl Androidu na trhu s ostatními mobilními operačními systémy, můžeme sledovat, jak se jeho popularita za poslední dva roky zvyšuje. V posledním čtvrtletí roku 2010 vzrostl podíl Androidu na 22 %. V listopadu roku 2011 byl ale jeho podíl už 52,5 %, čímž se stal jedničkou na poli mobilních operačních systémů. Druhé místo obsadil Symbian a na třetím místě skončil iOS od společnosti Apple. Podrobnější informace o podílu jednotlivých mobilních systémů na trhu v listopadu 2011 zachycuje tabulka 2.

V současnosti existuje pro Android obrovské množství bezplatných i komerčních aplikací, a proto není divu, že velká část vývojářů se zaměřuje na vývoj aplikací právě pro tento mobilní operační systém.

Operační systém	Podíl na trhu (v procentech)
Android	52,5
Symbian	16,9
iOS	15.0
Research in Motion	11.0
Bada	2.2
Microsoft	1.5
Ostatní	0.9

Tabulka 2: Podíl mobilních OS na trhu v listopadu 2011 [3]

2.2 GPS hry a aplikace pro OS Android

Tato část kapitoly si klade za cíl seznámit čtenáře s některými GPS hrami a aplikacemi, které jsou dostupné pro platformu Android. Je nutno dodat, že existují i jiné GPS hry a aplikace než zde uvedené a mnohé z nich jsou určeny nejen pro OS Android. V této části si přiblížíme dvě hry, konkrétně velmi populární Geocaching a méně známou hru s názvem Tourality. Dále krátce zmíníme aplikace Google Latitude a Foursquare, což sice nejsou GPS hry, ale „pouze“ aplikace, které GPS využívají jako svůj stěžejní prvek.

2.2.1 Geocaching

V současné době se jedná o velmi rozšířenou GPS hru, jež v sobě kombinuje prvky turistických, navigačních a Internetových her. Její základní princip spočívá v tom, že někdo na neznámé místo ukryje schránku (tzv. *cache*). Následně na Internetu zveřejní její souřadnice nebo indicie vedoucí ke zjištění těchto souřadnic. Cílem ostatních hráčů je pak s použitím navigačních GPS přístrojů tuto schránku najít. Obsahem schránky bývá zpravidla tzv. *logbook*, který slouží jako návštěvní kniha. Hráč má možnost se do této knihy zapsat, čímž potvrdí, že danou cache opravdu našel.

Hlavním důvodem, proč je tato hra tak populární, je skutečnost, že většina schránek je umístěna na zajímavých místech a hráč má tak možnost spojit příjemnou procházku s nalezením schránky. Závěrem je ještě nutné dodat, že těchto schránek je po světě rozmístěno nepřehledné množství čili hru je možno hrát prakticky po celém světě.

2.2.2 Tourality

Jedná se o méně známou GPS hru, která se od geocachingu liší především tím, že v ní jde o rychlost. Hra funguje tak, že hráči jsou si navzájem soupeři a vyhrává ten, kdo se dříve dostane na dané místo. Pohyb hráčů není nijak limitován, je tedy jedno, zda se hráč pohybuje pěšky, na bruslích, na kole nebo zda používá motorové vozidlo. Hra je doplněna o další zajímavé prvky, jakými jsou například virtuální odměny a trofeje, které je možno získávat různými způsoby. Hráč má také možnost aktivovat si různé miniaplikace (tzv. *gadgets*), díky nimž může získat různé výhody proti svým soupeřům [4].

2.2.3 Google Latitude

Google Latitude je aplikace, kterou sice není možno zařadit do kategorie GPS her, nicméně i přesto je systém GPS stěžejním prvkem této aplikace. S pomocí Google Latitude je možno jednoduše sledovat polohu svých přátel. Uživatel si tedy v aplikaci přidá do svého seznamu osoby, které zná a které také používají tuto aplikaci, a hned může vidět, kde se jeho přátelé právě nacházejí. Aplikace má navíc různá další vylepšení, jakým je například funkce historie polohy. Díky ní může uživatel snadno zjistit, kolik času stráví týdně v práci, doma nebo venku.

2.2.4 Foursquare

Obdobně jako Google Latitude, ani Foursquare nespadá do kategorie GPS her. Jedná se o geolokační službu, která v sobě kombinuje i prvky sociální sítě. Foursquare nabízí uživateli možnost podělit se s ostatními o svoje aktivity a místa, kde uživatel tráví volný čas. Lze tedy říci, že Foursquare slouží k vytváření nových přátel, kteří navštěvují například stejnou kavárnu, restauraci, pizzerii a další. Je nutné podotknout, že v České republice není zatím tato geolokační služba příliš populární [5].

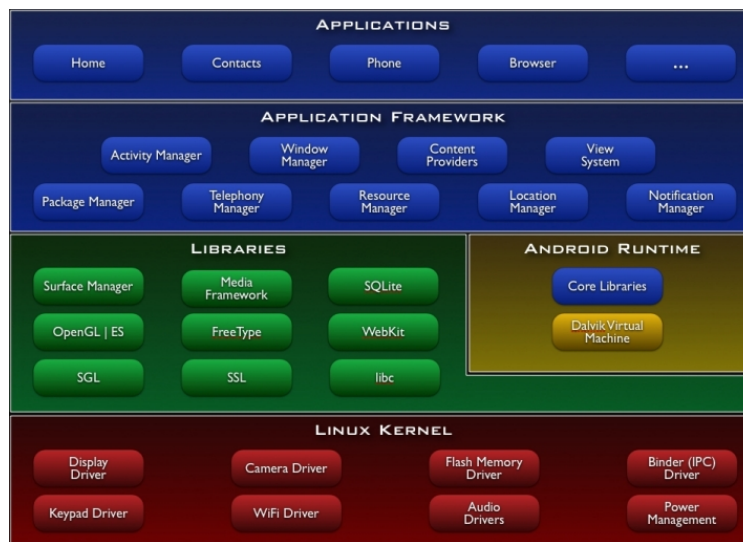
2.3 Architektura systému Android

Samotná platforma Android se mimo linuxového jádra skládá z velkého množství knihoven, napsaných v jazyce C/C++. Mezi tyto knihovny lze zařadit například knihovnu systémovou, knihovny pro práci s médii, dále knihovny enginu webového prohlížeče, SQLite knihovny a mnoho dalších. To, jak přesně vypadá architektura systému Android, znázorňuje obrázek 1.

Další věcí, kterou systém obsahuje, je *Dalvik Virtual Machine* - dále jen DVM. Jeho úkolem je zpracovávat soubory ve formátu `.dex`. Ty vznikají převodem z `.class` souborů, kterým se říká bytecode. Tento převod má na starost komponenta s názvem `dx`, která je obsažena v balíku Android SDK. Princip činnosti DVM je do jisté míry podobný činnosti Java Virtual Machine, která se využívá v aplikacích napsaných v programovacím jazyku Java. Hlavním rozdílem mezi činností JVM a DVM je tedy formát souborů, které zpracovávají - u DVM se jedná o `.dex` soubory, u JVM o `.class` soubory. Tyto dva typy souborů se od sebe liší především tím, že u `.dex` souborů jsou všechny třídy dané aplikace zabaleny v jednom souboru. U `.dex` souborů je také kladen důraz na malé požadavky na operační paměť. To je celkem pochopitelné, neboť u přenosných zařízení s OS Android nelze očekávat tak výkonné hardwarové vybavení jako u osobních počítačů, na nichž bývají spouštěny aplikace napsané v jazyce Java [6].

Navzdory některým rozdílům mezi JVM a DVM se aplikace pro Android programují primárně v jazyce Java. Je nutné ale podotknout, že Android podporuje zdrojový kód kompatibilní s JavaSE ve verzi 5 nebo nižší. Novinky obsažené v JavaSE 6 a 7 zatím nejsou podporovány, nicméně je možné, že toto se postupem času změní [7].

Druhým programovacím jazykem, který lze pro tvorbu aplikací pro OS Android použít, je C/C++. K tomu je třeba využít balík Android NDK (Native Development



Obrázek 1: Architektura systému Android [2]

Kit), který je zdarma dostupný pro všechny hlavní platformy. C/C++ se však využívá převážně pro tvorbu aplikací, u nichž klademe velký důraz na rychlost. Vzhledem k tomu, že aplikace Catch the Phantom je napsána v jazyce Java, nebudeme se dále vývojem aplikací v C/C++ zabývat.

2.4 Vývoj aplikací pro platformu Android v jazyce Java

2.4.1 Android SDK

Programujeme-li pro Android aplikace v jazyce Java, je třeba použít balík Android SDK (Standard Development Kit). Tento balík je složen z několika důležitých částí. Jednou z nich je nástroj `dx`, který, jak již bylo výše zmíněno, slouží k převodu `.class` souborů do `.dex` souborů, jež jsou určeny pro DVM. Při tomto převodu se také konvertuje Java bytekód do Dalvik bytekódu. Je nutné podotknout, že tento převod není úplně stoprocentní, z čehož vyplývá, že ne všechny prvky z jazyku Java lze použít na platformě Android. Aplikace se tedy neprogramují v plnohodnotném jazyce Java.

Další důležitou komponentou balíku Android SDK je emulátor. Jedná se o samostatnou desktopovou aplikaci, v níž je možno testovat aplikace, aniž bychom potřebovali fyzické zařízení s OS Android. Ačkoliv nám emulátor nabízí široké možnosti, nevyrovná se reálnému zařízení a přináší sebou tedy několik omezení. Emulátor například nepodporuje [8]

- používání Bluetooth,
- připojení sluchátek,

- vstup z kamery a fotoaparátu,
- kontrolu nad stavem baterie,
- kontrolu nad vyjmutím a vložením SD karty,
- telefonní hovory (lze je ale simulovat skrze konzoli emulátoru),
- připojení přes USB.

Pokud bychom tedy chtěli otestovat, jak se naše aplikace zachová, pokud nastane některý z výše uvedených případů, nemáme jinou možnost než použít fyzické zařízení.

Emulátor a nástroj `dx` samozřejmě netvoří celý obsah balíku Android SDK. Alespoň za zmínku určitě stojí `aapt`, který slouží k vytváření a modifikaci aplikačních balíčků, dále program `ddms`, s jehož pomocí můžeme ladit aplikace, komunikovat s běžícími instancemi emulátoru a také s připojenými reálnými zařízeními. Poslední částí balíku Android SDK, kterou zde zmíníme, je program `adb`, pomocí něhož lze do zařízení nahrávat soubory či aplikace [9] [10].

Na závěr této části je ještě nutno podotknout, že stejně jako Android NDK, tak i Android SDK je volně dostupné pro všechny hlavní platformy. Nejsme tedy nijak omezeni ať už vyvíjíme aplikace na operačním systému Windows, Linux nebo Mac.

2.4.2 Android Manifest

Bezpochyby jednou z klíčových částí každé Android aplikace je soubor manifest. Tento soubor je vždy uložen v kořenové složce aplikace a má název `AndroidManifest.xml`. Jedná se o soubor s příponou `.xml`, ve kterém je uloženo několik druhů nezbytných informací. Mezi tyto informace patří například:

- verze použitého SDK
- seznam jednotlivých aktivit aplikace
- seznam oprávnění
- aktivita, která má být aktivní při startu aplikace
- název aplikace
- seznam použitých externích knihoven

Seznam oprávnění aplikaci umožňuje provádět některé specifické úkony. Jestliže například potřebujeme, aby naše aplikace mohla zjišťovat aktuální polohu prostřednictvím systému GPS, musíme toto oprávnění přidat do souboru manifest. Jako další činnosti, které vyžadují v manifestu svá vlastní oprávnění, lze uvést používání bluetooth, posílání krátkých textových zpráv, modifikování systémového času nebo třeba zápis na externí úložiště, kterým bývá zpravidla SD karta.

V případě, že uživatel instaluje na svoje zařízení aplikaci, která vyžaduje některá oprávnění, jsou mu tato oprávnění před instalací zobrazena a záleží jen na uživateli, zda se rozhodne aplikaci používat.

Ve výpisu 1 je ukázka jednoduchého manifest souboru, který obsahuje oprávnění pro zjišťování polohy přes systém GPS a dále oprávnění pro ukládání dat na externí úložiště.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="example.app"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="14" />

    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:label="@string/app_name"
            android:name=".ExampleApplicationActivity" >
            <intent-filter >
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter >
        </activity >
    </application>

</manifest>
```

Výpis 1: Ukázka souboru `AndroidManifest.xml`

3 Pravidla a popis hry Catch the Phantom

3.1 Základní princip hry

Tato hra je určena pro dva až deset hráčů. Jeden z hráčů je v roli fantoma, ostatní hráči jsou v roli detektivů. Jak již samotný název hry napovídá, cílem hry je dopadnout fantoma. Detektivové se tedy musí pohybovat tak, aby se dostali co nejbližší k fantomovi. Jeho aktuální pozice je jim však oznámena vždy jen jednou za stanovený časový interval, fantom vidí pozice detektivů stále. Pokud se některému z detektivů podaří dostat k fantomovi na určitou vzdálenost předtím, než vyprší časový limit hry, vyhrává tým detektivů. V opačném případě je vítězem hry fantom.

3.2 Vytvoření hry a volba fantoma

K tomu, aby vůbec bylo možné hrát hru, je nejprve nutné ji vytvořit, což spočívá v nastavení parametrů hry. Pokud tedy chce hráč vytvořit novou hru, musí u ní specifikovat následující údaje:

- název hry
- počet hráčů (maximum je deset)
- časový limit hry, do jehož vypršení musí některý z detektivů fantoma najít¹, jinak fantom vyhraje
- vzdálenost detektiva od fantoma, při které bude fantom vyhodnocen jako chycený
- časový interval, jak často bude detektivům oznamována fantomova pozice
- heslo pro připojení ostatních hráčů do hry (není povinné)

Po vyplnění těchto informací má hráč ještě možnost zvolit, zda chce být fantomem. Pokud se rozhodne hrát v roli fantoma, ostatní hráči, kteří se budou připojovat k této hře, budou automaticky v roli detektivů. V opačném případě bude každý další hráč dotázán, zda chce být fantomem. Pokud nikdo z hráčů nebude chtít být fantomem, stane se fantomem poslední připojený hráč. Tím je tedy zajištěno, že nevznikne hra, v níž by byli pouze detektivové.

3.3 Připojení ostatních hráčů do hry

V předchozí části bylo řečeno, jaké informace musí hráč vyplnit, aby bylo možné vytvořit novou hru. U každé hry jsou však evidovány i jiné informace, mezi které patří i přesná pozice hráče v době, kdy hru vytvořil. Tuto pozici je možno chápat jako *centrální bod hry*. Ostatní hráči se pak mohou připojovat pouze k těm hrám, jejichž centrální bod je od jejich pozice do určité vzdálenosti. Bylo otázkou analýzy, zda tuto vzdálenost bude

¹v tomto případě se slovem najít rozumí přiblížit se na určitou vzdálenost

moci nastavit hráč při vytvoření hry, nebo zda bude vzdálenost určena již předem. Byla vybrána druhá varianta - v aplikaci je tedy nastavena fixní vzdálenost, která je stanovena na 30 kilometrů. Toto omezení je v aplikaci implementováno z toho důvodu, aby se předešlo situacím, kdy bude fantom od detektivů vzdálen mnoho kilometrů, neboť pak by nebylo možné v reálném časovém intervalu fantoma dopadnout a celá hra by pozbyla svého smyslu. Stejně tak detektiv, který by se ocitl ve velké vzdálenosti od centra hry, by byl zbytečný a pro fantoma bezpředmětný. Hry, jejichž centrální bod je od daného hráče velmi daleko (tedy více jak 30 kilometrů), nejsou v aplikaci vůbec zobrazovány. Dále pochopitelně nejsou zobrazeny ty hry, které již probíhají nebo byly již ukončeny. Hráč tedy má možnost připojit se ke všem hrám, které se mu v aplikaci zobrazí. Pokud je hra, do níž se hráč chce připojit, chráněna heslem, musí toto heslo zadat, jinak se připojení do hry nezdaří. Následně, pokud ještě není zvolen fantom, má hráč možnost vybrat si roli, v jaké bude hrát. To platí ale pouze v případě, že není posledním hráčem, neboť pak se stane fantomem. Jestliže je fantom již určen, je jasné, že hráč bude hrát v roli detektiva.

Tímto způsobem probíhá připojování hráčů do hry. Jakmile budou připojeni všichni hráči, hra automaticky začne.

3.4 Průběh hry

Hra začíná automaticky poté, kdy se připojí všichni hráči. Tím se tedy rozumí to, kdy počet připojených hráčů je roven nastavenému počtu hráčů dané hry. V části 3.2 bylo uvedeno, že u každé hry je nastaven časový limit, před jehož vypršením musí být fantom dopaden. Součástí tohoto intervalu je i tzv. *freeze time*. Jedná se o interval dlouhý tři minuty, který začne ubíhat ihned po zahájení hry. Během tohoto intervalu není možné dopadnout fantoma a detektivům ani není zobrazována fantomova poloha. Toto omezení je implementováno z toho důvodu, aby fantom nebyl okamžitě polapen ve hře, v níž by počáteční vzdálenost mezi některým detektivem a fantomem byla menší, než nastavená vzdálenost mezi detektivem a fantomem, při které je fantom vyhodnocen jako chycený. Skutečný maximální čas hry, během kterého může být fantom dopaden, je tedy o tři minuty kratší, než čas, který nastaví hráč při zakládání nové hry.

Začne-li běžet regulérní herní čas, detektivům je v pravidelných intervalech zobrazována fantomova pozice. Aby se detektivové mezi sebou mohli domlouvat, jak fantoma dopadnou, mohou si mezi sebou posílat zprávy. Posílání zpráv je naimplementováno jako hromadný chat - zprávu, kterou detektiv pošle, obdrží všichni detektivové. Detektivové mají možnost stahovat si zprávy manuálně, nicméně v aplikaci probíhá kontrola nových zpráv automaticky každou minutu a v případě, že jsou k dispozici nové zprávy, detektivovi se na displeji zobrazí číslo, které udává počet nových zpráv. Fantom o těchto zprávách není nijak informován. Má pouze neustálý přehled o aktuálních pozicích detektivů.

Mohlo by se zdát, že posílání zpráv mezi detektivy přináší detektivům jistou herní výhodu oproti hráči, který hraje v roli fantoma. Aby byla tato mírná nevýhoda fantoma vykompenzována, má fantom během hry na svém displeji zobrazen údaj, který jej informuje o nejmenší vzdálenosti mezi ním a některým z detektivů.

Každý hráč, bez ohledu na to, zda je detektiv či fantom, vidí i čas, který již uběhl a může si tedy snadno dopočítat, kolik minut chybí do konce hry.

3.5 Ukončení hry

Je celkem pochopitelné, že hra končí tehdy, dopadne-li některý z detektivů fantoma (vítězí tým detektivů), nebo vyprší herní čas a fantom nebude dopaden (vítězí fantom). Toto však nejsou všechny možné důvody, proč může být hra ukončena.

Detektivové mohou vyhrát i tak, že se fantom odpojí z probíhající hry. To je možné chápat i tak, že se fantom vzdal. Dojde-li k odpojení některého z detektivů, hra pokračuje dále. Hra by byla ukončena pouze tehdy, pokud by se odpojili všichni detektivové a fantoma by tak neměl kdo dopadnout.

4 Analýza

Předchozí kapitola popisuje, jak má hra fungovat, k jakým situacím může během hry dojít a především popisuje hru z pohledu uživatele. Aby bylo možné dodržet všechny výše uvedené aspekty, bylo nutno zvážit požadavky na celou aplikaci a na základě těchto požadavků pak sestavit způsob, jakým bude celá hra naimplementována.

4.1 Herní server

Vzhledem k tomu, že hra je určena pro více hráčů, přičemž každý hráč hraje na svém mobilním telefonu, bylo nutno stanovit, jakým způsobem si spuštěné hry budou předávat potřebná data - pozice jednotlivých hráčů, pozice fantoma, zprávy mezi detektivy apod. Pro vyřešení tohoto problému byla použita klient-server architektura, což je síťová architektura, která odděluje server a klienty, kteří spolu komunikují přes počítačovou síť.

Veškerá funkcionální serveru zajišťující komunikaci s mobilní aplikací je naimplementována ve skriptovacím programovacím jazyce PHP. Analýzu a následnou implementaci serveru je možno rozdělit do tří částí, které spolu velmi úzce souvisí:

- část, která se stará o komunikaci s mobilní aplikací
- část pro zpracování logiky webového rozhraní, prostřednictvím kterého může uživatel zpětně nahlížet na odehrané hry, zobrazovat si informace o svém profilu apod.
- grafické uživatelské rozhraní

Jak bylo již výše zmíněno, pro implementaci části serveru pro komunikaci s mobilní aplikací je použit jazyk PHP. Stejně tak je tento jazyk využit pro zpracování logiky webového rozhraní. Je nutné ale dodat, že ke tvorbě uživatelského rozhraní byl použit značkovací jazyk HTML, dále CSS a speciálně pro zpětné vykreslení pohybu fantoma v dané hře je použit programovací jazyk JavaScript.

4.1.1 Komunikace s mobilní aplikací

Tato část serveru má na starost výměnu dat mezi serverem a mobilní aplikací. V podstatě existují dvě varianty komunikace - buď aplikace potřebuje od serveru obdržet nějaká data, nebo chce na server data odeslat. V obou případech platí, že komunikaci zahajuje klient čili mobilní aplikace. Průběh komunikace serveru s aplikací lze zjednodušeně popsat těmito třemi kroky:

1. Aplikace naváže spojení se serverem a případně předá serveru potřebná data
2. Server zpracuje požadavek (ten je závislý na datech, které předá aplikace)
3. Server vrátí odpověď a komunikace je ukončena

Většina herních dat, jako například pozice hráčů, seznam vytvořených her apod., je uložena v databázi. Jedná se o relační databázi MySQL. Tento typ databáze byl vybrán jednak proto, že komunikace s ní prostřednictvím PHP je velmi jednoduchá a zároveň je pro účely aplikace postačující, neboť se nejedná o žádný rozsáhlý informační systém a není tedy potřeba využívat databáze známějších firem, jakými jsou například Oracle či Microsoft. Díky tomu, že některá data zůstávají v databázi uložena i po skončení hry, je možno hru zpětně rekonstruovat a hráči se tak mohou podívat, jak se například během hry pohyboval fantom.

4.1.2 Funkčnost webového rozhraní

Dalším úkolem při analýze serveru bylo určení, jaké možnosti bude uživateli nabízet webové rozhraní. Společně s tímto návrhem byl úzce spjat i návrh databáze, jelikož velká většina informací, které budou uživateli k dispozici i po ukončení hry, zůstane v databázi uložena. Výjimkou jsou v tomto případě pouze údaje o pohybu fantoma v dané hře. Ty jsou po skončení hry uloženy do `.xml` souboru a z databáze následně vymazány. Po skončení každé hry je tedy automaticky vygenerován soubor s názvem `gameHistory[ID hry].xml`, z něhož je možno rekonstruovat a vizualizovat pohyb fantoma.

Veškerá ostatní data, která zůstávají k dispozici po ukončení hry, z databáze odstraněna nejsou. Mezi taková data však nepatří například zprávy zasílané v průběhu hry nebo jednotlivé pozice detektivů. Cílem při analýze tedy bylo najít kompromis mezi množstvím funkcí webového rozhraní a mezi množstvím dat, která budou trvale uložena v databázi.

Nakonec vznikl seznam funkcí, které bude mít přihlášený uživatel na serveru k dispozici. Tyto funkce je možno pomyslně rozdělit do dvou kategorií:

- funkce související s profilem uživatele
- funkce pro zpětnou rekonstrukci hry

V rámci svého profilu bude mít uživatel možnost změnit si své přihlašovací heslo nebo si v případě jeho ztráty či zapomenutí může vygenerovat heslo nové, které mu bude zasláno emailem. Dále má uživatel možnost zobrazit si statistiku, ve které uvidí počet her, které odehrál, počet vyhraných her apod.

Co se týče možností souvisejících s odehranými hrami, přihlášený uživatel má možnost nechat si vypsat hry, které odehrál. U těchto her uvidí počet hráčů a jejich jména, kteří tuto hru odehráli. Dále bude samozřejmě k dispozici informace, jak hra dopadla. Pokud například hra skončí chycením fantoma, je u hry zobrazeno, že vyhrál tým detektivů, dále jméno detektiva, který fantoma dopadl a také čas dopadení. K dispozici jsou samozřejmě i informace, které se zadávají při vytváření hry - herní čas, interval, jak často se zobrazovala fantomova pozice apod. Stejně tak budou zobrazeny souřadnice centrálního bodu hry - tedy místa, kde byla hra vytvořena. Poslední funkcí bude vizualizace polohy fantoma během hry. Uživatel tak zpětně uvidí všechna místa, po kterých se fantom pohyboval.

4.2 Mobilní aplikace

4.2.1 Uživatelské rozhraní

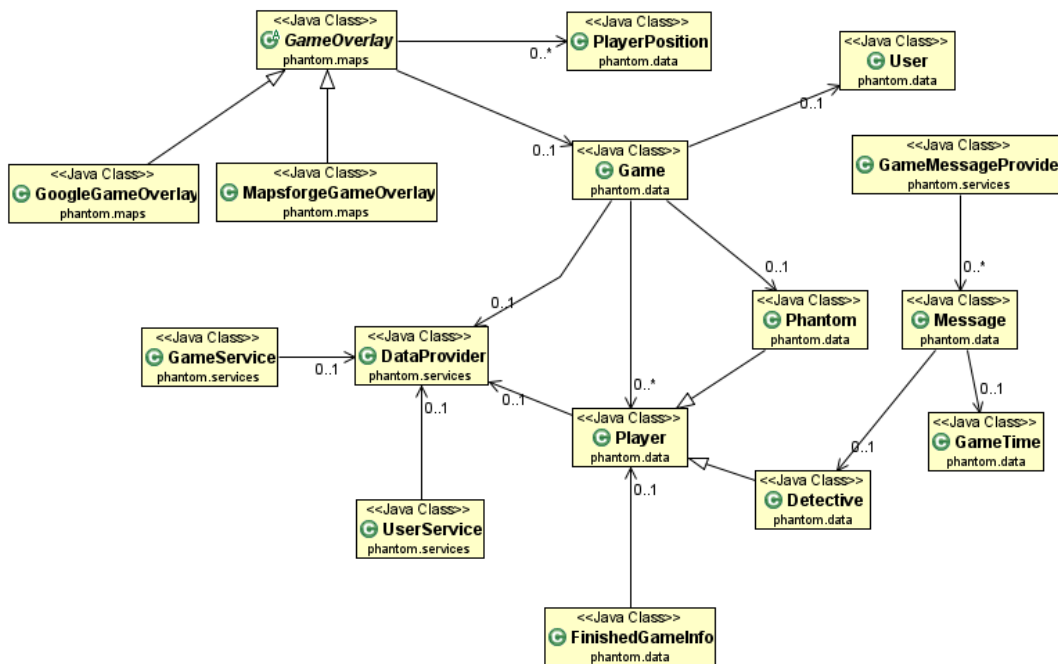
Při návrhu aplikace bylo nutno zvážit hned několik věcí, mezi které bezesporu patří i to, jak bude aplikace vypadat. Jinými slovy bylo nutno navrhnout GUI, což je zkratka z anglického Graphical User Interface. Vzhled aplikace je tím prvním, co uživatel při prvním spuštění uvidí, a proto je třeba dbát na to, aby vzhled byl na pohled příjemný, dále aby rozmístění ovládacích prvků bylo intuitivní, přičemž je ale nutné zároveň dodržet skutečnost, že GUI musí uživateli poskytovat možnost využívat veškerou funkcionalitu a možnosti, které má aplikace nabízet. Obecně totiž platí, že i kdyby aplikace disponovala velkým množstvím funkcí, uživatel ji patrně nebude používat, bude-li její rozmístění ovládacích prvků nepřehledné nebo případně špatně barevně sladěné. Z toho tedy jasně vyplývá, že návrh GUI je činnost, která by se rozhodně neměla podceňovat a měl by jí být věnován dostatečný čas. Výhodou při návrhu aplikací pro platformu Android je fakt, že vývojář má k dispozici velké množství ovládacích prvků, které je možno dále upravovat dle potřeby aplikace. Ukázka výsledného grafického uživatelského rozhraní je k dispozici v příloze tohoto textu.

4.2.2 Herní logika

Patrně nejdůležitější částí celé aplikace je její herní logika, proto jí také při analýze bylo věnováno velké množství času. Cílem bylo navrhnout jednotlivé třídy a vazby mezi nimi tak, aby byla herní logika co nejvíce oddělena od grafického uživatelského rozhraní, což přináší výhodu, že výsledný kód je čitelnější a v případě potřeby lze aplikaci snadněji rozšiřovat o nové funkce. Dále je díky oddělení herní logiky od GUI usnadněn případný přenos aplikace na další platformy.

Před samotnou implementací aplikace byl tedy navržen třídni diagram, který reprezentoval celou herní logiku. V průběhu vývoje se tento digram nepatrně měnil, nicméně základ zůstal stejný. Výsledný třídni diagram je zobrazen na obrázku 2. V tomto diagramu nejsou zahrnuty třídy reprezentující jednotlivé aktivity, dialogy apod.

V souvislosti s herní logikou bylo dále nutné zamyslet se, zda bude stačit, pokud aplikace poběží pouze v jednom vlákne. Již ze samotného zadání bylo celkem jasné, že jedno vlákno nebude dostačující. V průběhu hry je totiž nutné provádět několik na sobě nezávislých činností. Jde především o vykreslování pozic hráčů na mapě, odesílání pozice hráče na server a běh herního času. Kromě toho, že tyto činnosti na sobě nemohou být závislé, musí se ještě v pravidelných časových intervalech opakovat. V prvotní fázi analýzy bylo zamýšleno použití rozhraní `Runnable`, které by zajistilo rozdělení aplikace do vláken a dále použití třídy `Handler`, pomocí níž by se zajistilo automatické opakované provádění výše uvedených činností. Později v průběhu implementace se však ukázalo, že mnohem lepší a zároveň jednodušší bude použití tříd `Timer` a `TimerTask`. Výhodou použití těchto tříd je především fakt, že není nutné starat se o rozdělení činností do vláken. Tříde `Timer` stačí pouze předat potřebné metody společně s časovým intervalem, který určuje, po jaké době bude daná metoda opět zavolána. Každá metoda pak bude spouštěna v samostatném vlákne, aniž by to bylo nutné v kódu nějakým způsobem uvádět.



Obrázek 2: Výsledný třídní diagram herní logiky

4.2.3 Komunikace se serverem

Nedílnou součástí aplikace je také třída, která zajišťuje přenos dat mezi serverem a klientem. Bylo tedy nutné stanovit, jak bude celý přenos dat probíhat a hlavně jaký síťový protokol čtvrté vrstvy referenčního modelu OSI bude použit. Nabízely se tedy protokoly UDP a TCP. Tyto protokoly se od sebe liší především tím, že TCP je tzv. spojově orientovaný, UDP ne. To v praxi znamená, že přenos dat nad protokolem TCP je spolehlivý, UDP ne. Protokol UDP nezaručuje korektní přenos dat, oproti TCP je však rychlejší, což je způsobeno tím, že neobsahuje žádný mechanismus, který by zajišťoval potvrzování přenášených dat. Rozdíl v rychlosti TCP a UDP je však patrný až při přenosu většího objemu dat, což není případ hry Catch the Phantom. Proto pro komunikaci se serverem byl vybrán spolehlivý TCP protokol.

Výběr protokolu čtvrté vrstvy však problém komunikace se serverem vyřešilo pouze částečně. Stále zbývalo najít odpověď na otázku, zda bude použita komunikace pomocí socketů. Po zvážení všech požadavků kladených na funkčnost se ukázalo, že použití komunikace prostřednictvím socketů nebude nutné - jako dostačující se ukázala komunikace prostřednictvím HTTP protokolu. Pokud tedy aplikace potřebuje serveru předat nějaká data, jsou tato data předána jako parametry v URL.

Jednoduchá metoda, která na server odešle potřebná data a vrátí odpověď serveru, se nachází ve výpisu 2.

```
// PAGE – adresa serveru
// nameValuePairs – parametry, které chceme předat serveru
public InputStream getResponse(final String PAGE, ArrayList<NameValuePair> nameValuePairs)
    throws IOException
{
    HttpClient httpClient = new DefaultHttpClient();
    HttpPost httpPost = new HttpPost(PAGE);

    String entityValue = URLEncodedUtils.format(nameValuePairs, HTTP.UTF_8);
    entityValue = entityValue.replaceAll("&", "%26");

    StringEntity entity = new StringEntity(entityValue, HTTP.UTF_8);
    entity.setContentType(URLEncodedUtils.CONTENT_TYPE);
    httpPost.setEntity(entity);

    httpPost.setEntity(new UrlEncodedFormEntity(nameValuePairs));
    HttpResponse httpResponse = httpClient.execute(httpPost);
    HttpEntity httpEntity = httpResponse.getEntity();
    return httpEntity.getContent();
}
```

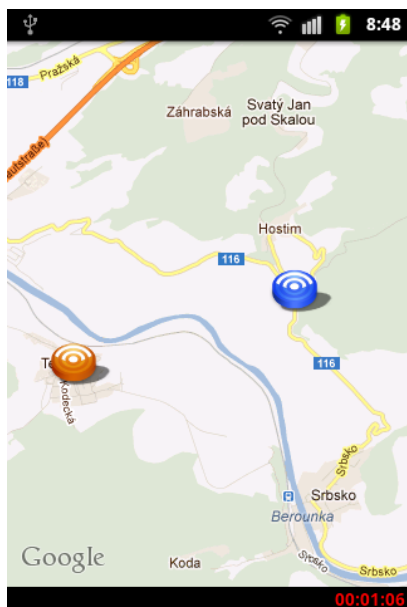
Výpis 2: Metoda, která předá data na server a vrátí odpověď serveru

4.2.4 Volba mapového podkladu

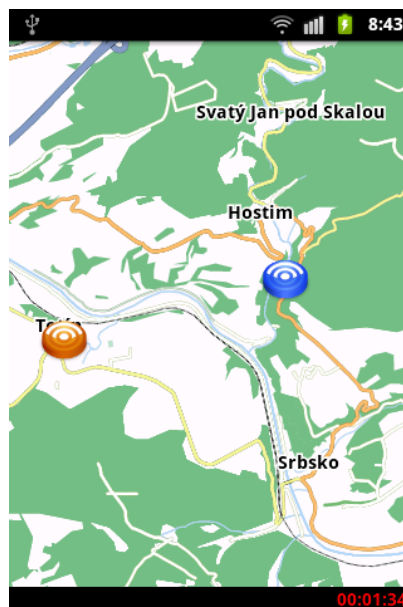
V kapitole 3 bylo uvedeno, že hráčům jsou v pravidelných intervalech zobrazovány pozice ostatních detektivů a fantoma. Tyto pozice jsou samozřejmě vykreslovány na nějaký mapový podklad - a právě výběr mapového podkladu byla další věc, jež bylo nutno ujasnit, jelikož se nabízely dvě možnosti:

1. stahovat potřebnou část mapy z Internetu - použití Google Maps
2. uložit do telefonu (respektive na SD kartu) externí mapový soubor (přípona `.map`)

Je potřeba říci, že každá z možností má své klady i zápory. Při používání Google Maps je velkou výhodou to, že se uživatel po nainstalování aplikace nemusí starat ještě o stažení externí mapy, která navíc k tomu zabírá relativně dost místa - například mapa České republiky zabírá na paměťové kartě přibližně 271 MB. Na druhou stranu, neustálé stahování mapy z Internetu je náročné z hlediska objemu přenášených dat. Vezmeme-li pak v potaz skutečnost, že aplikace potřebuje v pravidelných intervalech získávat ze serveru pozice ostatních hráčů včetně dalších informací, dojdeme k závěru, že použitím Google Maps by se zbytečně zvýšil objem přenášených dat. Otázkou při výběru mapového podkladu tedy bylo dilema mezi pohodlím uživatele a menším množstvím přenesených dat. Nakonec byla aplikace naimplementována tak, že si uživatel v nastavení aplikace sám může vybrat, jakou mapu chce použít.



Obrázek 3: Ukázka mapy při použití Google Maps



Obrázek 4: Ukázka mapy při použití Mapsforge

4.2.5 Projekt Mapsforge

Aby bylo možno v aplikaci Catch the Phantom dát uživateli na výběr, zda chce používat Google Maps nebo mapy, které jsou k dispozici i bez připojení k Internetu, bylo nutno využít projekt Mapsforge. Tento projekt je volně dostupný a slouží pro práci s vektorovými mapami. V současnosti je k dispozici ve verzi 0.3.0, nicméně v aplikaci Catch the Phantom je použita starší verze, konkrétně 0.2.4. Tato verze je použita jednak proto, že vyšla teprve nedávno² čili již nebyl dostatek času změnit aplikaci tak, aby používala nejnovější verzi. Druhým důvodem použití verze 0.2.4 je skutečnost, že nejnovější verze zatím obsahuje drobné chyby, které mají být odstraněny až v další verzi [11].

Použití Mapsforge je velmi jednoduché. Z webových stránek tohoto projektu si stačí stáhnout knihovnu, kterou je následně nutno přidat do aplikace. Poté je již možné využívat všechny možnosti, které tato knihovna nabízí. Podmínkou pro správný chod aplikace je samozřejmě to, že uživatel bude mít ve svém zařízení jeden nebo více souborů s mapami.

Funkcionalita, kterou Mapsforge nabízí, je velmi podobná jako u Google Maps. Navíc velká většina jednotlivých metod je pojmenována stejně čili způsob implementace se příliš neliší. Na obrázku 3 je znázorněna ukázka, jak vypadá mapa, pokud hráč používá Google Maps, na obrázku 4 je pak ukázka ze hry, kdy se hráč rozhodl používat offline mapy projektu Mapsforge.

²konkrétně dne 19. března 2012

5 Implementace

5.1 Mobilní aplikace

Implementace mobilní aplikace probíhala podle předem vytvořeného UML třídního diagramu, reprezentujícího vnitřní logiku celé aplikace. Došlo sice k nepatrným změnám, nejednalo se však o zásadní změny, které by razantním způsobem změnilly podobu mezi navrženým a výsledným třídním diagramem. Byly tedy vytvořeny odpovídající třídy, které byly postupně rozšiřovány o nové potřebné metody.

Současné s vývojem mobilní aplikace bylo nutno implementovat i server, respektive část serveru, která má na starost komunikaci a výměnu dat mezi aplikací a serverem. Zpočátku byly naimplementovány pouze odpovídající metody pro výměnu potřebných dat a jejich správné fungování bylo testováno výpisy do logu. Teprve ve chvíli, kdy bylo zřejmé, že výměna dat probíhá bezchybně, bylo dotvářeno uživatelské rozhraní.

5.1.1 Logické rozdělení zdrojového kódu

Během implementace byl zároveň kladen důraz na oddělení herní logiky od grafického uživatelského rozhraní a samozřejmě také na rozdělení zdrojového kódu na menší logické celky. Kód byl tedy rozdělen do celkem šesti balíčků³, které si nyní ve stručnosti popíšeme. Jedná se o tyto balíčky:

1. `phantom.activities`
2. `phantom.data`
3. `phantom.dialogs`
4. `phantom.exception`
5. `phantom.maps`
6. `phantom.services`

Jak již samotný název napovídá, v prvním balíčku jsou obsaženy pouze jednotlivé aktivity aplikace. Těchto aktivit je celkem devět a patří mezi ně například herní aktivita, hlavní menu aplikace, přihlašovací obrazovka, seznam vytvořených her apod.

Nejrozsáhlejším balíčkem je balíček s názvem `phantom.data`. Obsahem tohoto balíčku je celkem dvanáct tříd, které v podstatě obstarávají velkou část herní logiky. Do tohoto balíčku patří třída `Player`, z níž dědí třídy `Phantom` a `Detective`, dále sem spadá třída `GameTime`, která reprezentuje herní čas a dále například třída `FinishedGameInfo`, jež v sobě zapouzdřuje informace o ukončení hry.

Balíček `phantom.dialogs` obsahuje čtyři třídy, které slouží jako složitější dialogy, které se hráči mohou v průběhu hry objevit. Patří sem dialog, který se zobrazuje při ukončení hry a dále dialogy pro posílání a čtení zpráv, zasílaných během hry jednotlivými

³tzv. package

detektivy. Společně s jednotlivými aktivitami vytváří tento balíček prezentační vrstvu aplikace.

V balíčku `phantom.exception` se nachází tři třídy, které reprezentují specifické druhy výjimek. Níže jsou uvedeny jejich konkrétní názvy, společně s vysvětlením, za jakých okolností může dojít k jejich vyvolání.

- `InputValueException` - k vyvolání této výjimky dojde v momentě, kdy uživatel vyplňuje textová pole (např. při zakládání nové hry nebo při registraci nového uživatele) a některá vyplněná položka není ve správném formátu, případně není vyplněna vůbec.
- `NoRecordsFoundException` - tato výjimka je vyvolána, pokud aplikace požaduje od serveru nějaká data, která se ale serveru nepodaří najít. Příkladem může být případ, kdy aplikace potřebuje od serveru získat data o uživateli podle jeho jedinečného čísla, avšak server zjistí, že uživatel s daným jedinečným číslem neexistuje.
- `PhpScriptException` - pokud server zjistí nějaký problém během provádění určité akce, vrátí mobilní aplikaci popis problému, který je dále propagován prostřednictvím této výjimky. Jako příklad lze uvést registraci nového uživatele - pokud server zjistí, že uživatel chce použít uživatelské jméno, které je již registrováno, v aplikaci dojde k vyvolání výjimky `PhpScriptException`.

Obsah balíčku `phantom.maps` je tvořen několika třídami, které slouží k provádění metod, spojených s vykreslováním pozic jednotlivých hráčů na mapu. Nachází se zde třídy jak pro práci s Google mapami, tak pro práci s offline mapami projektu Mapsforge.

Posledním balíčkem v aplikaci je `phantom.services`. Obsahem tohoto balíčku je mimo jiné velmi důležitá třída `DataProvider`, která obsahuje metody a atributy potřebné pro komunikaci s herním serverem. Pokud v aplikaci probíhá jakákoli komunikace se serverem, vždy se této komunikace účastní tato třída, lépe řečeno instance této třídy. Dále balíček obsahuje třídy pro práci s uživateli, hrami a herními zprávami. Pomocí těchto tříd je možno vytvářet nové uživatele nebo hry, ověřovat existenci uživatele, posílat herní zprávy apod. Poslední třídou v tomto balíčku je třída s názvem `Cryptography`. Tato třída je velmi triviální a obsahuje pouze jednu statickou metodu, která na vstupu očekává textový řetězec. Výstupem z metody je tzv. hash vstupního řetězce. K tomuto účelu je použit hashovací algoritmus SHA. Tato metoda je určena pro případ, kdy aplikace posílá na server citlivá data, jakými jsou přihlašovací hesla uživatelů. Díky tomu, že se tato hesla přenáší v zašifrované podobě, je znemožněno odposlechnutí a následné zneužití hesla potenciálním útočníkem.

5.2 Server

Server byl implementován ve dvou fázích. První fází se rozumí vytvoření části, jež obstarává komunikaci s klientem. Tato část byla pochopitelně vyvíjena paralelně s mobilní aplikací, jelikož její správnou funkcionální bylo možno otestovat pouze při komunikaci s

klientem. Teprve ve chvíli, kdy mobilní aplikace byla ve stavu, který umožňoval alespoň základní hratelnost hry, což znamená, že i na serveru byly vytvořeny patřičné metody, následovala implementace logiky webového rozhraní.

Implementace této části byla mnohem jednodušší jednak proto, že oproti části komunikující s klientem nevyžaduje tak velké množství funkcí, ale také proto, že některé její části nebylo nutno implementovat, jelikož je už obsahovala část pro komunikaci s mobilní aplikací.

5.3 Problémy při implementaci

Během implementace se objevilo několik problémů, které nebyly při analýze a tvorbě třídních diagramů hned patrné. Některé z těchto problémů si nyní ve stručnosti popíšeme.

5.3.1 Synchronizace herního času jednotlivých hráčů

Jedním z problémů, který se vyskytl při implementaci mobilní aplikace, byla synchronizace herního času. Problém spočívá v nepatrné odchylce časů, který běží jednotlivým hráčům připojeným do téže hry. Mobilní aplikace obsahuje třídu, která obsahuje metody a atributy potřebné pro správný běh herního času. Dále obsahuje fragment kódu, který má na starost přechod na herní aktivitu. K provedení této části kódu dojde v momentě, kdy aplikace zjistí, že do hry jsou připojeni všichni hráči a je tedy možné zahájit hru. Samotný problém však spočívá v tom, že u každého hráče dojde k přechodu na herní aktivitu a tedy i ke spuštění běhu herního času v jiný okamžik. To je způsobeno tím, že aplikace v pravidelném dvousekundovém intervalu kontroluje počet připojených hráčů. V případě, že proběhne kontrola, ihned poté se připojí poslední hráč, v aplikaci nastane téměř dvousekundová prodleva, během které už samotná hra může být u jiných hráčů spuštěna.

Bylo otázkou, zda tuto nepatrnou odchylku časů vůbec řešit. V případě ukončení hry jiným způsobem než vypršením herního času je velká pravděpodobnost, že hráči by o této odchylce ani nevěděli. Problém by mohl nastat v případě, že hru vyhraje fantom díky vypršení herního času, během kterého nebude chycen. Mohlo by se tak stát, že se některým hráčům ukončí hra s informací, že časový limit vypršel, ale jejich údaj o aktuálním čase bude ukazovat dvě sekundy do konce hry. Proto bylo nutné tento problém vyřešit.

Problém synchronizace herního času byl vyřešen poměrně jednoduchým, ale zároveň dobře fungujícím způsobem. V případě, že některý z hráčů obdrží od serveru odpověď, že hra může být zahájena a zároveň je tento hráč první, který to zjistil, server si uloží aktuální čas zahájení hry. Ostatním hráčům je pak v okamžiku jejich dotazu na zahájení hry předána informace o tom, kolik času uběhlo od zahájení hry. Díky této informaci je pak možné snadno čas synchronizovat. O celý tento proces se na straně mobilní aplikace stará metoda `synchronize()`, která je součástí třídy `GameTime`.

5.3.2 Problém se změnou orientace displeje

Velmi závažným problémem při implementaci bylo špatné chování aplikace v případě, že uživatel změnil v průběhu hry orientaci displeje. Pokud v operačním systému Android dojde k otočení displeje, standardním chováním je to, že daná aktivita je opětovně vytvořena a dojde tedy k volání metod `onCreate()`, `onStart()` a `onResume()`. V herní aktivitě se v metodě `onResume()` nachází fragment kódu, jenž zajistí, aby docházelo k automatickému volání některých potřebných metod. Jednou z těchto metod je i metoda, která je automaticky volána každou sekundu a zajišťuje správný běh herního času. Je nutné si uvědomit, že tato metoda běží v samostatném vlákne. Při otočení displeje tedy došlo k dalšímu vytvoření vlákna a metoda se tak volala dvakrát za jednu sekundu, což způsobilo, že herní čas běžel dvakrát rychleji. Při dalším otočení displeje se již za jednu sekundu čas posunul o tři sekundy. Vzhledem k tomu, že se jednalo o velmi závažný problém, bylo třeba jej vyřešit.

Jako nejlepší a zároveň patrně i nejjednodušší se ukázalo řešení, jež spočívalo v zabránění volání výše uvedených metod, které jsou standardně volány při změně orientace displeje. Toho se docílilo tak, že bylo třeba na dané aktivitě překrýt metodu `void onConfigurationChanged(Configuration c)` a dále v souboru manifest bylo třeba k dané aktivitě dopsat níže uvedený řádek.

```
android:configChanges="orientation"
```

Díky této jednoduché změně se celý problém vyřešil, jelikož nedocházelo při otočení displeje k volání metody `onResume()` a nebylo tedy vytvářeno další vlákno, jež by obstarávalo běh herního času.

5.3.3 Problém ztráty připojení k Internetu

Dalším problémem při implementaci mobilní aplikace byla situace, kdy hráč hraje hru a přijde o připojení k Internetu. Pokud k této situaci dojde, znamená to, že hráč nemůže odesílat na server svou polohu a zároveň nemůže stahovat ze serveru informace o poloze ostatních hráčů. Server však o ztrátě připojení není nijak informován a z jeho pohledu je hráč stále připojený do hry. Na serveru bylo tedy nutné napsat část kódu, která zkontroluje, zda jsou stále všichni hráči aktivní.

Kód, který tuto kontrolu obstarává, je relativně jednoduchý. Vždy, když nějaký hráč požaduje od serveru informace o pozicích ostatních hráčů, server zkontroluje časy, kdy byly tyto polohy na server odeslány. Tyto časy porovná s aktuálním herním časem a zjistí-li, že nějaká poloha je starší než dvě minuty, odpojí ze hry hráče, jenž tuto polohu zaslal. Pokud se tedy stane, že některý z hráčů přijde o připojení k Internetu, server jej za dvě minuty korektně odpojí ze hry. Pokud se bude jednat o hráče v roli detektiva, má tento hráč možnost se při obnovení připojení k Internetu opět připojit do hry. Odpojí-li server fantoma, bude hra okamžitě ukončena s informací, že se odpojil fantom a vyhraje tým detektivů.

6 Testování

Ze začátku vývoje mobilní aplikace bylo testování prováděno na mobilním telefonu Samsung Galaxy Ace s operačním systémem Android 2.3.3. V okamžiku, kdy aplikace byla ve fázi, která umožňovala základní hratelnost hry, byl k účelům testování společně s reálným zařízením použit i standardní emulátor, který je součástí balíku Android SDK. Emulátor samozřejmě nemá systém GPS, proto bylo na emulátoru nutné simulovat polohu.

Postupem času, během kterého byly do aplikace přidávány všechny funkce a možnosti, které má hra nabízet, bylo zřejmé, že testování na emulátoru a jednom fyzickém zařízením nebude dostačující. Proto ve chvíli, kdy se aplikace blížila finální podobě, bylo nutno otestovat aplikaci pouze na reálných mobilních telefonech bez toho, aniž by byla pozice některého z hráčů simulována.

6.1 Testování na reálných zařízeních

Ve chvíli, kdy byla aplikace v téměř finální verzi, došlo na testování hry v reálných zařízeních. K testovacím účelům byly použity tři mobilní telefony, které společně s jejich verzí operačního systému Android a rozlišením displeje zachycuje tabulka 3.

Typ telefonu	Verze OS Android	Rozlišení displeje
Samsung Galaxy Ace	2.3.3	320x480
LG P500	2.2	320x480
Vodafone 945	2.1	240x400

Tabulka 3: Mobilní telefony k testování aplikace

Z tabulky je patrné, že aplikace byla testována na třech různých verzích systému Android. Bohužel v tabulce chybí zařízení s verzí 4.0, jelikož se nepodařilo sehnat například nějaký tablet, který by tuto verzi Androidu obsahoval. V příloze této práce je tedy alespoň ukázka aplikace, spuštěné v emulátoru s verzí Androidu 4.0.

V rámci testování bylo odehráno několik her podle předem domluveného scénáře tak, aby byly vyzkoušeny všechny možnosti, při kterých dojde k ukončení hry. Konkrétně se jedná o tyto čtyři možnosti:

1. fantom byl dopaden některým z detektivů
2. fantom vyhrál tak, že vypršel herní čas, během kterého nebyl chycen
3. vyhrál tým detektivů, jelikož se fantom odpojil ze hry
4. vyhrál fantom, jelikož se postupně odpojili oba detektivové

Během her samozřejmě bylo vyzkoušeno i posílání zpráv mezi detektivy. Ukázalo se, že posílání a přijímání zpráv funguje správně. Co se týče odpojování se hráčů ze hry, bylo vyzkoušeno jak korektní odpojení⁴, tak automatické odpojení, kdy hráč úmyslně

⁴to znamená, že hráč byl připojen k Internetu a použil tlačítko *Disconnect from game*

vypnul připojení k Internetu. V obou případech aplikace fungovala dle očekávání. Dále bylo vyzkoušeno i znovupřipojení hráče do hry, ze které se odpojil, ať už úmyslně, nebo vinou ztráty připojení k Internetu. Ani v tomto případě nebyly shledány žádné chyby. Je nutné podotknout, že odpojení ze hry tlačítkem *Disconnect from game* a následné opětovné připojení do hry mohl provést pouze hráč v roli detektiva, jelikož při úmyslném odpojení fantoma je hra okamžitě ukončena.

7 Závěr

Cílem této práce bylo podat čtenáři ucelené informace o dnes velmi populárním operačním systému Android a dále popsat pravidla hry Catch the Phantom, analýzu požadavků kladených na hru, následnou implementaci a nakonec i výsledky testování hry v reálném provozu.

V současné době se operační systém Android drží na první příčce v pomyslném žebříčku mobilních operačních systémů. Ve třetím čtvrtletí roku 2011 bylo podle průzkumu zjištěno, že 52,5 % prodaných smartphonů obsahuje OS Android [12]. Proto lze zcela jistě tvrdit, že vývoj aplikací pro tuto platformu je z pohledu programátora velmi perspektivní. Je otázkou, jak dlouho se Android na první příčce udrží. V úvodu této práce jsme zmínili, že tento systém je možno nalézt v mobilních telefonech, tabletech a některých GPS navigacích. Velmi zajímavé je, že dnes lze Android najít i ve speciálních lyžařských brýlích. Jedná se o brýle, které v sobě mají zabudovaný inteligentní modul, který obsahuje systém GPS a také Bluetooth, pomocí něhož se brýle dokáží spojit s mobilním telefonem a dokáží na sklo brýlí promítat různé informace, například aktuální rychlost, polohu, apod. Vezmeme-li navíc v potaz skutečnost, že se blíží příchod verze 5.0, lze očekávat, že Android své prvenství na poli mobilních operačních systémů ještě dlouho uhájí. Velký zlom však může přinést příchod operačního systému Windows Phone 8 s rozhraním Metro. Zdá se, že toto by mohl být vážný konkurent Androidu.

Co se týče aplikace Catch the Phantom a jejího dalšího možného vývoje, určitě najdeme několik možností, o které by se tato hra dala rozšířit. Jednou z těchto možností je například kontrola rychlosti pohybu hráčů. Díky této možnosti by pak bylo možné zajistit, že některý z hráčů nebude používat při hře například osobní automobil či prostředky městské hromadné dopravy. V případě velkého zájmu ze strany uživatelů by bylo určitě zajímavé přenést aplikaci na další mobilní platformy, jakými jsou například Windows Mobile od společnosti Microsoft či iOS od společnosti Apple. Díky tomu by bylo možné, aby jednu hru hráli hráči na zařízeních s různými operačními systémy.

Podíváme-li se na možná rozšíření webového rozhraní, uživatelé by jistě ocenili nějaké rozsáhlejší statistiky či lépe propracovanou funkci zpětné rekonstrukce hry. Aktuálně webové rozhraní nabízí pouze základní informace o odehraných hrách a základní informace o hráčích. Bylo by přínosné vytvořit například bodovou statistiku, kdy by každému hráči byly po odehrané hře podle speciálního algoritmu přiděleny body, podle nichž by se určovala úroveň, jak moc je hráč dobrý.

8 Reference

- [1] Galaxie: Vývoj mobilních telefonů (1. díl) [online]
Dostupné z <http://www.galaxie.name/index.php?clanek=vyvoj-mobilnich-telefonu-1-dil>
- [2] Srovnání mobilních systémů - OSMZ [online]
Dostupné z <http://osmz.wikidot.com/android>
- [3] Gartner Says Sales of Mobile Devices Grew 5.6 Percent in Third Quarter of 2011; Smartphone Sales Increased 42 Percent. [online]. Dostupné z: <http://www.gartner.com/it/page.jsp?id=1848514>
- [4] What is Tourality?: The ultimate Location Based Game. *Tourality* [online]. 2012. Dostupné z: <http://www.tourality.com/what-is-tourality/>
- [5] Foursquare. In: *Wikipedia: the free encyclopedia* [online]. San Fransisco (CA): Wikimedia Foundation, 2001
Dostupné z: <http://cs.wikipedia.org/wiki/Foursquare>
- [6] EHRINGER, David. The Dalvik Virtual Machine Architecture. 2010, s. 8. Dostupné z: <http://davidehringer.com/software/android/The.Dalvik.Virtual.Machine.pdf>
- [7] BURNETTE, Ed. *Hello Android: Introducing Google's mobile development platform*. Version 2008-12-5. Raleigh, N.C.: Pragmatic Bookshelf, c2008, 218 s. ISBN 9781934356173 (PBK.).
- [8] Android Developers [online]
Dostupné z <http://developer.android.com/index.html>
- [9] Vyvíjíme pro Android – úvod. *Svět Androida* [online]. Dostupné z: <http://www.svetandroida.cz/vyvijime-pro-android-1-uvod-201103>
- [10] KRUMNIKL, Michal. *Přednášky pro předmět Tvorba aplikací pro mobilní zařízení II* [online]
Dostupné z <http://tamz2.mrl.cz/download/>
- [11] Mapsforge: Free mapping and navigation tools [online]
Dostupné z <http://code.google.com/p/mapsforge/>
- [12] Každý druhý prodaný smartphone má v sobě Android. *Svět Androida* [online]. Dostupné z: <http://www.svetandroida.cz/kazdy-druhy-prodany-smartphone-ma-v-sobe-android-201111>

A Uživatelská příručka

A.1 Nasazení serveru

Aby mohla aplikace správně fungovat, je třeba uvést do provozu herní server. K tomu je potřeba mít k dispozici server s podporou skriptovacího programovacího jazyku PHP ve verzi 5.3.3 nebo vyšší. Na tento server je třeba nahrát všechny soubory a složky, které se nachází v adresáři **Server** na příloženém CD. Následně je třeba povolit zápis do složky `GameHistory`, jelikož se do ní po skončení hry generuje xml soubor pro zpětnou rekonstrukci pohybu fantoma.

V současné době je server spuštěn na adrese

```
http://homel.vsb.cz/~skl0013/phantom/
```

Lze říci, že toto je hlavní herní server. Pokud by bylo potřeba spustit herní server i na jiné adrese, je třeba na hlavním serveru v souboru `app_server.php` přidat odkaz na nový herní server. Při spuštění mobilní aplikace ihned dojde ke zjištění dostupných herních serverů a uživatel má možnost přes hardwarové tlačítko Menu zvolit herní server, který chce používat. Díky tomu je možno hru přesunout na více serverů bez zásahu do zdrojového kódu mobilní aplikace. Soubor `app_server.php` vypadá nyní takto:

```
<?php

/**
 * When you want to add game server, add line into the $servers array.
 * Important: Line must be in this format:
 * server name (anything you want);url of server
 */

$servers = Array();

/**
 * Primary game server
 * Name: homel.vsb.cz
 * URL: http :// homel.vsb.cz/~skl0013/phantom/app
 */
$servers[0] = "homel.vsb.cz;http:// homel.vsb.cz/~skl0013/phantom/app";

for ($i = 0; $i < count($servers); $i++)
{
    echo $servers[$i] . "<br>";
}
?>
```

Výpis 3: Soubor pro přidávání herních serverů

Pokud by bylo tedy potřeba přidat například server s názvem `secondServer`, který by byl spuštěn na adrese `http://secondaryServer.cz/phantom/app`, bylo by třeba přidat následující řádek:

```
$servers[1] = "secondServer;http://secondaryServer.cz/phantom/app";
```

Dále je pro správný běh serveru nutno mít k dispozici databázi MySQL verze 5.5.15 nebo vyšší. Tato databáze se vytvoří SQL skriptem, který se nachází ve složce **SQL** na příloženém CD.

V současné době se databáze nachází na serveru `win-edu.cs.vsb.cz`. Pokud by bylo potřeba změnit umístění databáze, je zároveň nutné v souboru `dtbConnect.php`, jež se nachází ve složce **Server/services** na příloženém CD, změnit cestu a případně i jméno a heslo k databázi. Soubor je nutno upravit tak, aby obsahoval správně nastavený následující řádek:

```
$connectionId = mysql_connect('[adresa serveru]', '[jméno  
databáze]', '[heslo]');
```

A.2 Instalace a spuštění mobilní aplikace

Aplikaci *Catch the Phantom* je možno spustit na fyzickém zařízení, které má operační systém Android ve verzi 2.1 nebo vyšší (mobilní telefon, tablet), nebo je možné ke spuštění využít PC s nainstalovaným emulátorem.

A.2.1 Instalace aplikace na fyzické zařízení

K tomu, aby mohla být aplikace nainstalována na fyzické zařízení, je nejprve třeba toto zařízení propojit s PC. To lze udělat buď pomocí USB kabelu, nebo pomocí technologie Bluetooth. Po připojení zařízení jsou dvě možnosti jak aplikaci nainstalovat:

- přenést soubor `CatchThePhantom.apk` na paměťovou kartu, kde po jeho vybrání dojde k instalaci
- využít nástroj `adb` a použít příkaz

```
adb -s [sériové číslo zařízení] install [cesta k aplikaci]
```

Ve výše uvedeném příkazu je sériové číslo zařízení nepovinný parametr. Je nutné jej zadat pouze v případě, že je k počítači připojeno více zařízení a my potřebujeme určit, na které z těchto zařízení má být aplikace nainstalována. Sériová čísla všech připojených zařízení je možno zjistit následujícím příkazem:

```
adb devices
```

Vzhledem k tomu, že aplikace nepochází z Google Play⁵, je nutno při instalaci z paměťové karty v menu zařízení povolit instalaci aplikací z jiných zdrojů. Pokud bude aplikace instalována přes nástroj `adb`, není nutné instalaci z jiných zdrojů povolovat.

⁵dříve pod názvem Android Market

A.2.2 Instalace aplikace do emulátoru

Instalace aplikace do emulátoru se provádí stejným příkazem jako instalace do fyzického zařízení přes nástroj `adb`, jen s tím rozdílem, že je třeba správně zadat sériové číslo emulátoru, do kterého má být aplikace nainstalována.

Pro správné hraní hry v emulátoru je třeba simulovat polohu, jelikož emulátor nedisponuje systémem GPS. Fiktivní poloha se do emulátoru zašle následujícím příkazem:

```
geo fix [latitude] [longitude]
```

A.3 Popis aplikace

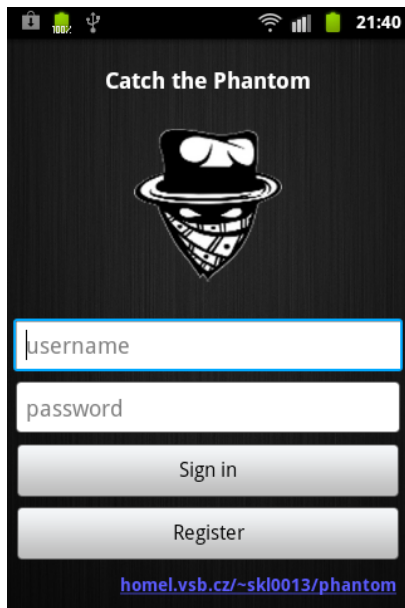
Jakmile je aplikace nainstalována v reálném zařízení nebo v emulátoru, je možno ji spustit. Před spuštěním je třeba zajistit, že zařízení či emulátor má k dispozici připojení k Internetu. Zároveň je nutné, aby byl spuštěn herní server, jinak se aplikace ukončí.

Po spuštění se objeví přihlašovací okno (obrázek 5), ve kterém je nutno zadat uživatelské jméno a heslo. Pokud ještě uživatel nemá vytvořen účet, je možné přejít na aktivitu pro registraci nového uživatele (obrázek 6).

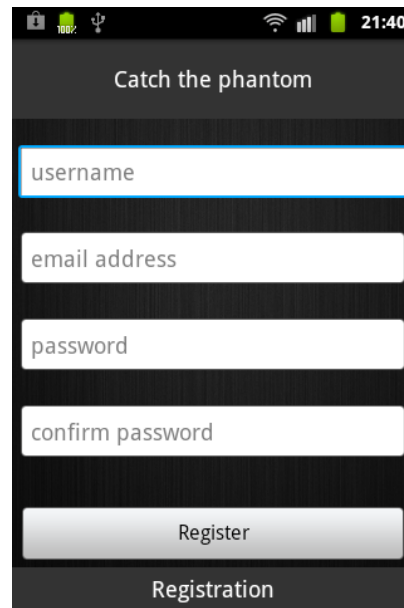
Po úspěšném přihlášení se objeví hlavní menu celé aplikace (obrázek 7). V tomto menu má uživatel možnost založit buďto novou hru (tlačítko *Create new game*), nebo se připojit k již vytvořené hře, která čeká na připojení všech hráčů (tlačítko *Join a game*). Pokud se rozhodne vytvořit novou hru, je třeba vyplnit potřebné údaje o hře (herní čas, počet hráčů apod.) (obrázek 8). Pokud se chce hráč připojit k již vytvořené hře, musí znát heslo, pokud je jím hra chráněna. Dále mu bude oznámeno, zda bude hrát v roli detektiva nebo fantoma. Pokud nebude posledním připojícím se hráčem a ve hře ještě nebude fantom, bude vyzván, aby si herní roli vybral sám. Poté přejde aplikace do stavu čekání, dokud se nepřipojí zbylí hráči. Jakmile se tito zbylí hráči připojí, hra automaticky začne.

Kdykoliv během hry je možno stisknout hardwarové tlačítko Menu, čímž dojde k vyvolání herního menu. V tomto menu má uživatel možnost vybrat si ze tří možností:

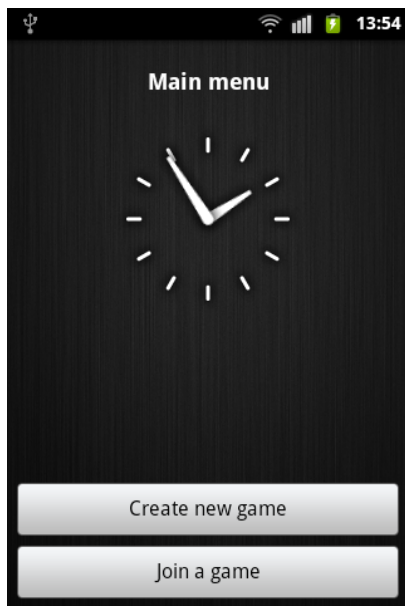
- Read messages - zobrazí okno s příchozími zprávami (funguje pouze tehdy, hraje-li hráč v roli detektiva)
- Send message - zobrazí okno pro napsání nové zprávy (funguje pouze tehdy, hraje-li hráč v roli detektiva)
- Disconnect from game - slouží k odpojení hráče ze hry



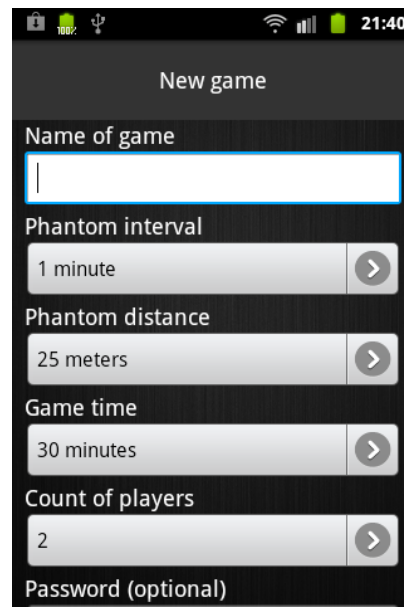
Obrázek 5: Přihlašovací obrazovka



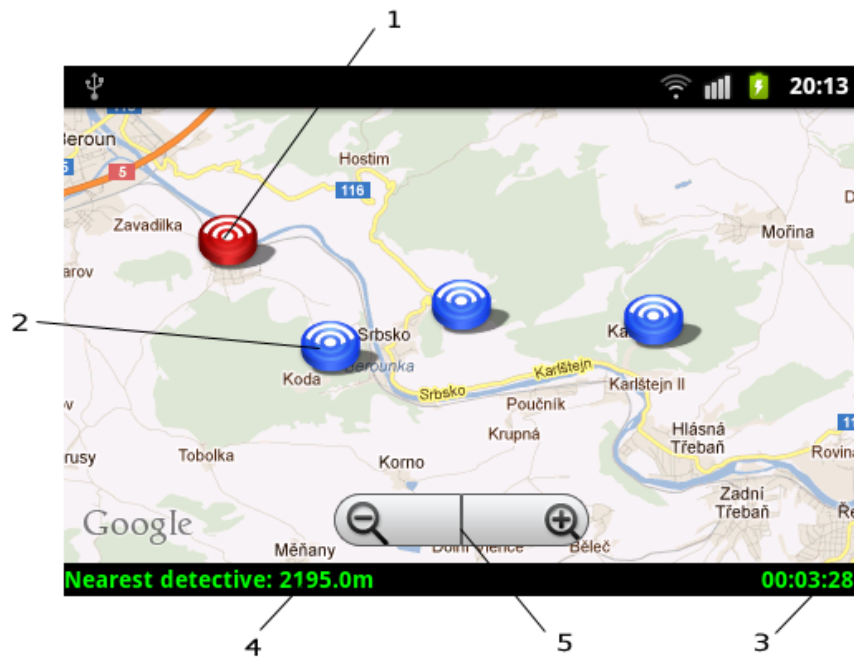
Obrázek 6: Registrace uživatele



Obrázek 7: Hlavní menu aplikace



Obrázek 8: Vytvoření nové hry



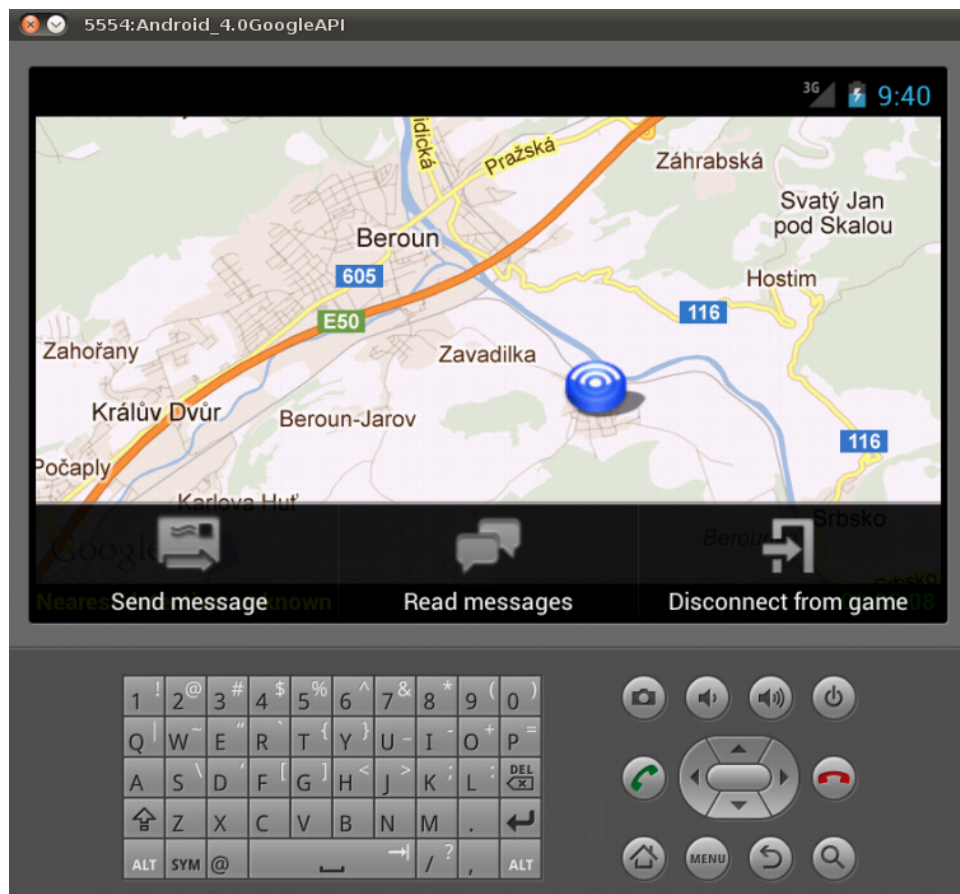
Obrázek 9: Ukázka ze hry z pohledu fantoma

A.4 Popis herní obrazovky

Na obrázku 9 se nachází ukázka ze hry z pohledu fantoma.

Legenda k obrázku

1. Pozice fantoma
2. Pozice detektivů
3. Herní čas
4. Vzdálenost nejbližšího detektiva
5. Tlačítka pro přiblížení a oddálení mapy (místo nich lze ale také pro přiblížení a oddálení použít „zoom dvěma prsty“)



Obrázek 10: Ukázka hry z emulátoru s verzí Androidu 4.0

B Obsah přiloženého CD

Na přiloženém CD se nachází následující složky:

- **MobilniAplikace** - instalační soubor s příponou apk
- **MobilniAplikaceProjekt** - zdrojové kódy jako zkomprimovaný projekt Eclipse
- **Server** - zdrojové kódy serveru
- **SQL** - SQL skript pro vytvoření databáze
- **Text** - text bakalářské práce ve formátu pdf
- **Dokumentace** - dokumentace zdrojového kódu mobilní aplikace