

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Mobilní klient pro systém publikování gamebooků

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 4. května 2015

Pavel Příbář

Poděkování

Rád bych poděkoval Ing. Ladislavu Pešíčkovi za vstřícnost, trpělivost, odborné rady a cenné připomínky, které mi pomohly tuto bakalářskou práci vypracovat.

Abstract

The purpose of this bachelor thesis is to create mobile client (application) for the Android platform, which can read gamebooks that are created in the system which is being developed. Gamebook is a book that allows reader to make decisions and partly influence the story of the book. In the first part of this work there is a comparison of applications that can read gamebooks. The second part describes the system for creating and publishing gamebooks which is being developed. Mobile client is part of this system. The last part contains design of the client and description of implementation including verification of functionality.

Abstrakt

Cílem této bakalářské práce je vytvořit mobilního klienta (aplikaci) pro platformu Android, který umožňuje procházet gamebooky vytvořené ve vyvíjeném systému. Gamebook je kniha, ve které se čtenář rozhoduje jak se bude příběh vyvíjet. V první části práce jsou porovnány některé dostupné aplikace, které umožňují procházení gamebooků. Druhá část práce popisuje vyvíjený systém, ve kterém je možné gamebooky vytvářet a publikovat. Součástí tohoto systému je i mobilní klient. Poslední část obsahuje návrh klienta a popis implementace včetně ověření jeho funkcionality.

Obsah

1	Úvod	1
1.1	Cíle práce	1
2	Existující aplikace	2
2.1	Charakteristika gamebooku	2
2.2	Vybrané mobilní aplikace	3
2.2.1	Aplikace The Forging of a Legend	3
2.2.2	Aplikace Choice of Broadsides	5
2.2.3	Aplikace Lone Wolf Saga	7
2.3	Srovnání testovaných aplikací	10
3	Systém pro publikování gamebooků	12
3.1	Program pro tvorbu gamebooku	13
3.2	Server	13
3.2.1	Služby pro editor	13
3.2.2	Uživatelské služby	14
3.2.3	Služby pro mobilního klienta	14
3.3	Mobilní klient	14
3.4	Reprezentace gamebooku	16
3.4.1	Struktura XML souboru	16
4	Návrh mobilního klienta	18
4.1	Návrh síťové komunikace	19
4.1.1	Možnosti síťové komunikace	19
4.1.2	Dostupné knihovny	20
4.1.3	Vybrané řešení síťové komunikace	22
4.2	Způsob uložení dat	22
4.2.1	Sdílené preference	22
4.2.2	Ukládání souborů	24
4.2.3	Databáze SQLite	25
4.2.4	Zvolený způsob pro uložení dat	25
4.3	Návrh UI	26
4.3.1	Aktivita MenuActivity	28

4.3.2	Aktivita <code>GameActivity</code>	28
4.3.3	Aktivita <code>LibraryActivity</code>	28
4.3.4	Ostatní aktivity	29
4.3.5	Příprava grafických souborů	30
5	Realizace mobilního klienta	31
5.1	Struktura aplikace	31
5.1.1	Soubor <code>AndroidManifest.xml</code>	31
5.2	Použité knihovny	32
5.2.1	Knihovna <code>logback-android</code>	32
5.2.2	Knihovna <code>ActionBarSherlock</code>	33
5.2.3	Knihovna <code>Nine Old Androids</code>	33
5.2.4	Knihovna <code>ViewPagerIndicator</code>	33
5.2.5	Knihovna <code>SlidingMenu</code>	33
5.3	Implementace <code>SQLite</code> databáze	34
5.3.1	Vytvoření databáze	35
5.3.2	Přístup k datům	35
5.4	Uživatelské funkce	35
5.4.1	Import gamebooků	36
5.4.2	Procházení gamebooku	36
5.4.3	Knihovna	38
5.5	Sít'ová komunikace	38
5.5.1	Implementace knihovny <code>Volley</code>	38
5.5.2	Získání seznamu dostupných gamebooků	38
5.5.3	Stažení gamebooku	39
5.5.4	Hodnocení gamebooku	40
5.6	Nastavení aplikace	40
6	Testování	42
6.1	Průběh testování	42
6.2	Testování nestandardních událostí	43
6.3	Výsledky testování	43
6.3.1	Nestandardní události	44
6.4	Možná rozšíření	44
7	Závěr	45
	Seznam použitých zkratk a výrazů	46
	Literatura	48
	Seznam příloh	50

A	Uživatelská příručka	51
A.1	Požadavky aplikace	51
A.2	Instalace	51
A.3	Ovládání aplikace	51
A.3.1	Procházení gamebooku	53
A.3.2	Knihovna	54
A.3.3	Nastavení aplikace	56
B	Obsah CD	60
C	Použité knihovny	61

1 Úvod

V dnešní době vlastní téměř každý tzv. „chytrý“ telefon, který kromě telefonování a psaní zpráv poskytuje mnoho jiných funkcí. Výrobci a programátoři se snaží mnoho činností, úkonů a nástrojů běžného života přetransformovat do mobilních aplikací tak, aby tyto funkce a nástroje byly kdekoliv a kdykoliv snadno přístupné pomocí mobilního zařízení, např. mapy, digitální fotoaparát, budík, rádio a mnohé další. Jinak tomu není ani u gamebooků.

Gamebook je kniha, ve které se čtenář rozhoduje, jak se bude příběh dále vyvíjet a do jisté míry si tak příběh knihy určuje sám. Po přečtení určité části textu jsou hráči nabídnuty možnosti, pro které se může rozhodnout. Výběrem jedné z těchto možností je hráč odkázán na jinou část textu, která na rozhodnutí navazuje. Tímto stylem je vytvořena celá kniha. Některá rozhodnutí mohou záviset také na náhodě, např. hráč musí hodit kostkou a podle výsledku je odkázán dále. Spolu s dalšími herními prvky, např. souboji hlavní postavy, je gamebook možné označit za textovou hru.

1.1 Cíle práce

Jedním z cílů této práce je prozkoumat vybrané mobilní aplikace, které umožňují procházení gamebooků. Dalším cílem této práce je popsat vyvíjený systém, který umožňuje gamebooky vytvářet a publikovat (číst, hrát). Smyslem popisu systému je vysvětlit jeho základní principy fungování. Součástí vyvíjeného systému bude mobilní klient (mobilní aplikace), který umožňuje gamebooky vytvořené pomocí tohoto systému procházet. Návrh a implementace mobilního klienta je hlavním cílem této práce. Po popisu vyvíjeného systému bude vybrána vhodná funkcionalita, kterou by mobilní klient měl obsahovat. Následně bude popsán návrh této funkcionality pro platformu Android. V poslední části práce bude ukázána a vysvětlena implementace mobilního klienta spolu s popisem ověření jeho funkcionality.

Po přečtení této práce by měl čtenář pochopit, jakým způsobem vyvíjený systém bude fungovat a jaké jsou jeho součásti. Dále by se měl dozvědět jakou funkcionalitu mobilní klient obsahuje a jak je tato funkcionalita navržena. Z popisu implementace by měl být pochopen princip, jakým je mobilní klient (aplikace) implementován na platformě Android.

2 Existující aplikace

Důvodem k prozkoumání vybraných mobilních aplikací (umožňujících procházení gamebooku) je získat obecný přehled o těchto aplikacích (popř. systémech pro vývoj nebo publikování gamebooku), jejich funkcnostech a možnostech, které nabízejí. Ze získaných informací je pak možné se při implementaci vlastního mobilního klienta (popř. celého systému) inspirovat. Mobilní klient vyvíjeného systému bude realizován na platformě Android, proto byl výběr existujících aplikací, které umožňují procházení gamebooků, zaměřen výhradně na aplikace spustitelné na platformě Android.

2.1 Charakteristika gamebooku

Gamebook je kniha, která umožňuje čtenáři při čtení příběhu provádět rozhodnutí, která ovlivní vývoj příběhu knihy. Zpravidla je kniha členěna do odstavců a každý odstavec je očíslován. Po přečtení odstavce má čtenář na výběr několik možností, pro které se může rozhodnout. Každá možnost je očíslována. Toto číslo označuje odstavec, ve kterém příběh pokračuje pro danou volbu. Čtenář najde daný odstavec a pokračuje v četbě. Příběh se tímto způsobem větví a vzniká orientovaný graf. Orientovaný graf [4] je konečná množina, jejíž prvky nazýváme vrcholy (uzly) grafu a hrany grafu. Hrany spojují jednotlivé vrcholy, každá hrana má určený směr a mezi dvěma vrcholy existuje v jednom směru pouze jedna hrana. Protože čtenář vytváří rozhodnutí a děj se větví, tak kniha může mít více alternativních konců. Při rozhodování může být také použit náhodný jev. Čtenář při čtení může narazit na pokyn od autora, aby pro další volbu hodil hrací kostkou. Podle toho jaké číslo na kostce padne, pak čtenář pozná, ve kterém odstavci má pokračovat. Hrací kostka může být standardní šestistěnná nebo vícečetná, záleží na autorovi.

V úvodu knihy je čtenář seznámen se zápletkou, prostředím a hlavní postavou. V některých knihách je možné vlastnosti postavy na začátku knihy upravit. Vlastnostmi jsou myšleny atributy, např. zdraví postavy, síla, inteligence nebo schopnost, kterou postava ovládá. Používané vlastnosti nejsou jednoznačně určeny a v různých knihách se mohou lišit. Vlastnosti mohou být upraveny náhodně pomocí hrací kostky a hozené hodnoty budou postavě přičteny k dané vlastnosti. Druhou možností je, že hráč dostane konečný počet bodů, které může mezi jednotlivé vlastnosti rozdělit. V průběhu čtení, se vlastnosti postavy mohou na základě děje měnit. Příběhy obvykle také obsahují souboje s nepřáteli, kteří se v příběhu vyskytují. Zda čtenář (hráč) v souboji uspěje záleží na jeho vlastnostech a na náhodě (např. hodu kostkou). Proto se těmito knihám říká gamebook z anglického slova game, tedy hra. Gamebook je tedy možné označit za textovou hru. Žánr gamebooku je většinou psán

ve stylu fantasy nebo sci-fi, ale není to podmínkou. Autor může zvolit jakýkoliv jiný žánr.

2.2 Vybrané mobilní aplikace

V následujících podkapitolách budou popsány mobilní aplikace, které umožňují procházení gamebooků na platformě Android. Vybrané aplikace pro srovnání jsou následující:

1. The Forging of a Legend dostupné z <https://play.google.com/store/apps/details?id=com.gamebook>,
2. Choice of Broadsides dostupné z <https://play.google.com/store/apps/details?id=com.choiceofgames.broadsides>,
3. Lone Wolf Saga dostupné z <https://play.google.com/store/apps/details?id=com.GDVGames.LoneWolfBiblio>.

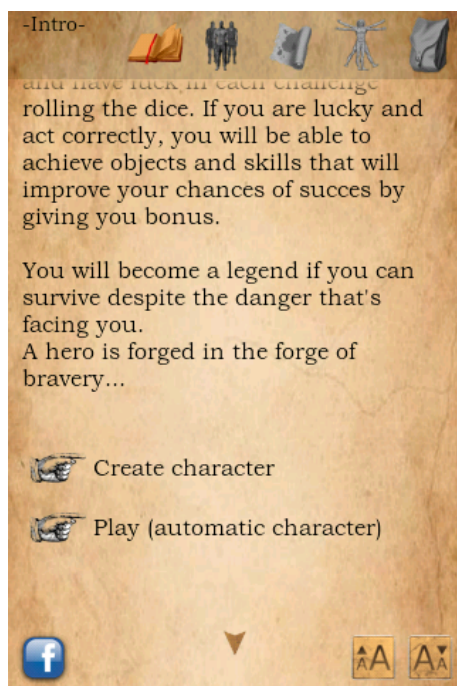
Výše uvedené aplikace jsou dostupné z obchodu Google play (dostupný z <https://play.google.com>), který je oficiálním nástrojem platformy Android pro distribuci aplikací. Všechny aplikace obsahují příběh v angličtině, popř. navíc obsahují i jiné jazykové verze, ale nepodařilo se nalézt gamebook aplikaci s příběhem v českém jazyce. Aplikace byly testovány na zařízeních Samsung GT-S5839i a Samsung GT-S6500 s verzí 2.3.6 (Gingerbread) platformy Android a na zařízení Sony Xperia M obsahující verzi 4.3 (Jelly Bean). Testování bylo především zaměřeno na pohoďnost ovládání, grafickou stránku aplikace a herní možnosti hráče. Příběh a žánr, který byl v dané aplikaci, nebyl posuzován, protože to není cílem této práce.

2.2.1 Aplikace The Forging of a Legend

První testovaná aplikace se jmenuje **The Forging of a Legend**. Velikost instalace je **10 MB** a minimální vyžadovaná verze Androidu je **1.6 (Donut)**. Tento gamebook je možné číst (hrát) v katalánštině, španělštině a angličtině. Aplikace je dostupná zdarma.

Ovládání aplikace nebylo bezchybné. Po kliknutí na tlačítko nebylo poznat, zda tlačítko bylo opravdu stisknuto, chyběla zde reakce na stisk, např. změna barvy tlačítka. Tlačítka byla rozmístěna blízko sebe a několikrát se stalo, že bylo stisknuto tlačítko jiné, než bylo zamýšleno. Při čtení delšího textu bylo nutné přecházet na

další obrazovku (otáčet stránku), ale aplikace občas na přetažení prstu nereagovala. Bohužel aplikace neumožňovala čtení při tzv. **landscape** orientaci obrazovky (zobrazení obrazovky na šířku), ale pouze v **portrait** orientaci, tj. zobrazení obrazovky na výšku (viz [1]). Výběr možnosti rozhodnutí je umístěn na konci stránky pod textem, viz obr. 2.1. Velmi praktická byla možnost zmenšit nebo zvětšit písmo textu knihy.



Obrázek 2.1: Ukázka z aplikace The Forging of a Legend



Obrázek 2.2: Ukázka animace z aplikace The Forging of a Legend

Grafický vzhled aplikace působí příjemně a přehledně, význam jednotlivých ikon na hrací ploše je snadno rozpoznatelný. Aplikace obsahuje jednoduché animace, například hod kostkou (obr. 2.2) nebo efekt otočení stránky při přechodu na jinou obrazovku. Tyto animace jsou doprovázeny zvukovým efektem. U některých částí textu jsou zobrazeny obrázky související s textem, které jsou umístěny na konci stránky pod textem nebo zabírají celou obrazovku. Tyto obrázky čtení velmi oživí. Většina postav v tomto gamebooku má svůj obrázek, který se zobrazí když postavu poprvé potkáte, viz obr. 2.3. Herní obrazovka, viz obr. 2.1 nebo 2.3, je rozdělena na tři části, v horní liště jsou umístěny ikony, které slouží pro uložení hry, procházení obrázků objevených postav apod. Kromě ukládání hry nemají tyto funkce ve hře žádný praktický význam a slouží pouze jen jako doplňky a dodatečné informace pro hráče. Uprostřed obrazovky se zobrazuje text a obrázky. Na dolní liště jsou umístěny dvě ikony, pomocí kterých lze zvětšovat nebo zmenšovat text a ikonka sociální sítě facebook, která umožňuje sdílet některé události ze hry na této sociální síti.



Obrázek 2.3: Ukázka obrázku postavy z aplikace The Forging of a Legend

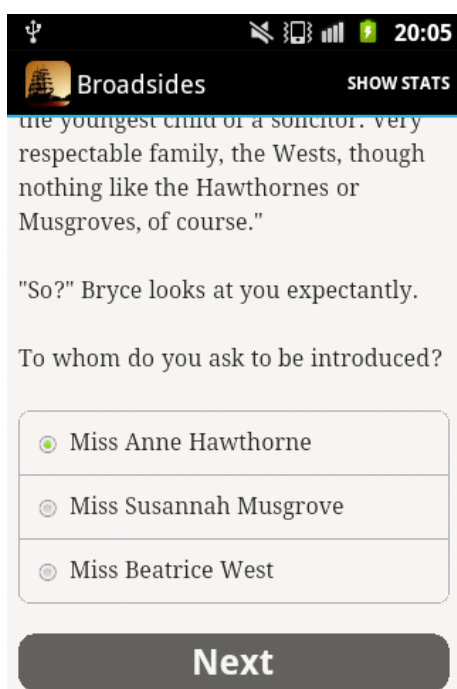
Na počátku hry má hráč možnost nechat si atributy postavy automaticky náhodně vygenerovat nebo může pro každý atribut házet kostkami. Ve hře jsou vždy používány dvě šestistěnné kostky. Součet těchto dvou hozených hodnot je hráči přičten k danému atributu. V průběhu hry může hráč získávat předměty, které tyto atributy vylepší, a nebo mohou být tyto atributy ve hře vylepšeny po skončení nějaké události, např. po skončení boje. Při soubojích se využívá pouze kostek. Hráč má lichý počet pokusů a musí vyhrát vždy více než polovinu z nich. Pokus je vyhrán tak, že hráč musí hodit hodnotu stejnou nebo vyšší než je zadaná. Pokud hodí menší, pokus je neúspěšný. V průběhu hry bylo nabyto dojmu, že získané atributy postavy nemají na souboje žádný vliv a jsou pouze o náhodě. Hru je možné během čtení ukládat, ale pouze třikrát za celou hru. Hra je automaticky uložena před každým soubojem a tři možnosti uložení tak nemají větší význam. Pokud by se hra neukládala automaticky a mohl ji ukládat pouze hráč jen třikrát za celou hru, pak by hra byla mnohem obtížnější a především zajímavější.

2.2.2 Aplikace Choice of Broadsides

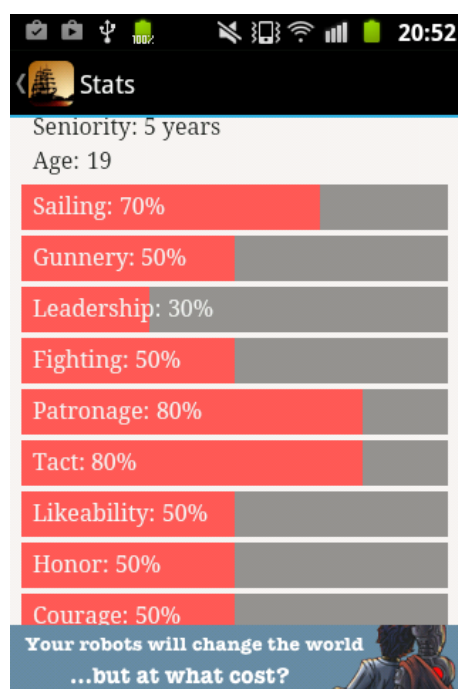
Druhá testovaná aplikace se jmenuje **Choice of Broadsides**. Velikost její instalace je **4.6 MB** a minimální potřebná verze Androidu je 2.3 (Gingerbread). Aplikace je dostupná v angličtině a byla vyvinuta společností **Choice of Games LLC**.

Tato společnost už vydala přes 20 podobných gamebookových aplikací. Aplikace jsou v podstatě stejné, liší se pouze tématem, žánrem a obsahem textů. Společnost pro psaní příběhů vyvinula vlastní skriptovací jazyk **ChoiceScript**, více informací viz [5]. Aplikace je dostupná zdarma, ale obsahuje reklamy, které je možné odstranit zaplacením poplatku.

Ovládání této aplikace je velmi jednoduché a umožňuje pouze číst text a rozhodovat se. Pod textem pro danou část odstavce jsou umístěny volby, které hráč může provést, viz obr. 2.4. Uživatel musí nejdříve kliknout na danou volbu, a poté kliknutím na tlačítko pod těmito volbami výběr potvrdit. Poté je hráči zobrazena obrazovka s novým textem. V aplikaci není možné provádět žádná nastavení, např. velikost textu. Aplikace se skládá z hlavní (herní) obrazovky, dále z obrazovky, na které jsou zobrazeny statistiky postavy a z kontextového menu (na zařízeních s verzí 3.0 (**Honeycomb**) a vyšších je nahrazeno **Action Barem**). **Action Bar** je grafická komponenta (lišta) v horní části obrazovky (případně i dolní), která by uživateli měla usnadnit navigaci v aplikaci a poskytnout přímý přístup k důležitým nástrojům a funkcím aplikace. Toto menu obsahuje tlačítko pro restart hry, odkazy na podobné aplikace od této firmy, zaplacení za odebrání reklam z aplikace a další odkazy. Za výhodu se dá považovat to, že aplikace jako jediná z testovaných umožňuje čtení textu v landscape orientaci obrazovky.



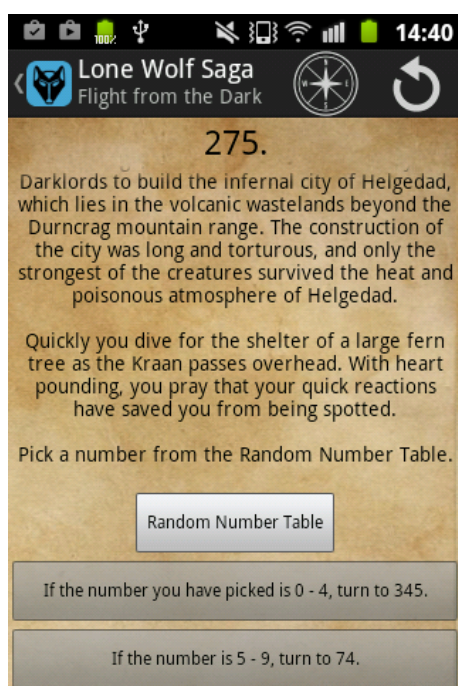
Obrázek 2.4: Ukázka z aplikace Choice of Broadsides



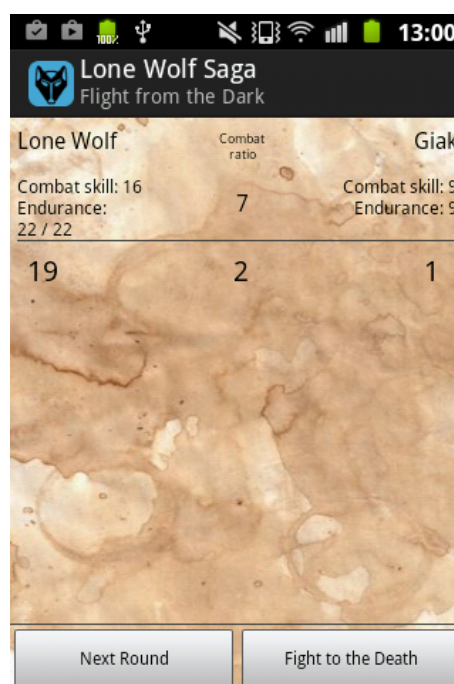
Obrázek 2.5: Ukázka statistik v aplikaci Choice of Broadsides

Grafika, tzn. obrázky nebo ikony, nejsou v této aplikaci použity a aplikace neobsahuje žádné animace ani zvukový doprovod. Text je členěn do přehledných odstavců a čtení delšího textu je tak snadnější. Bohužel zde chybí možnost zvětšovat nebo zmenšovat velikost písma jako u první testované aplikace.

Vytvoření postavy včetně statistik probíhá na začátku tak, že hráči jsou položeny otázky a nabídnuty možné odpovědi, ze kterých si může vybrat a podle vybrané odpovědi se zlepší určitá vlastnost (atribut) postavy. Uživatel si také musí vybrat svoji slabinu, tzn. daná vlastnost se zhorší. Schopnosti jsou ohodnoceny procenty, viz obr. 2.5. Vybrané vlastnosti pak v příběhu přímo ovlivňují šanci úspěchu při akci, na které daná vlastnost závisí.



Obrázek 2.6: Ukázka z aplikace Lone Wolf Saga



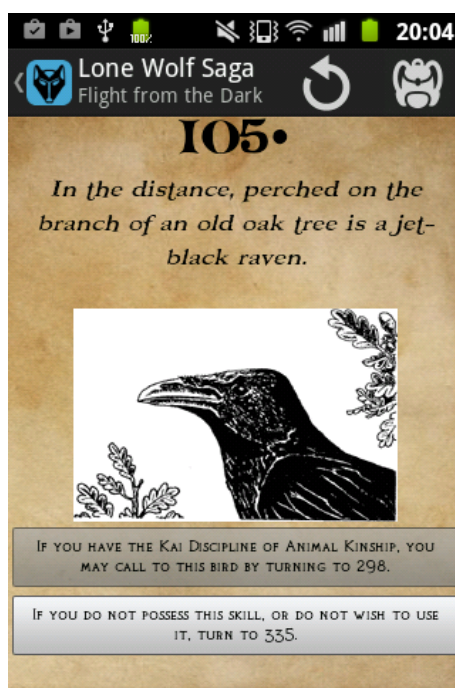
Obrázek 2.7: Ukázka boje z aplikace Lone Wolf Saga

2.2.3 Aplikace Lone Wolf Saga

Další testovanou aplikací je **Lone Wolf Saga**. Velikost instalace je **7.1 MB**, ale po instalaci aplikace je nutné ještě stáhnout databázi obsahující jednotlivé gamebooky a obrázky. Tato databáze má velikost přibližně **20 MB**. Aplikace je lokalizovaná pouze do anglického jazyka a je určena pro zařízení s operačním systémem Android od verze 2.1 (Eclair). Aplikace je zdarma, ale je možné finančně podpořit vývojáře a autora gamebooku nebo si koupit sloty pro ukládání průběhu hry.

Gamebooky, které jsou obsaženy v této aplikaci existují také jako fyzické knihy (<http://lonewolf.mytago.cz/>), které jsou na rozdíl od aplikace přeloženy i do českého jazyka.

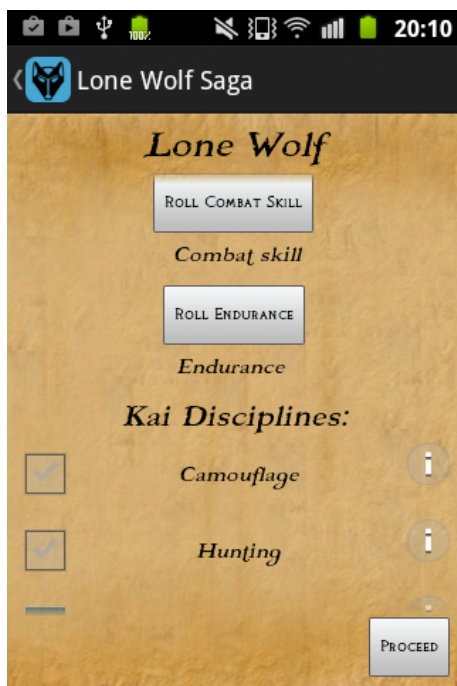
Aplikace se ovládá pomocí tlačítek rozmístěných na obrazovce. Četba gamebooku probíhá podobně jako u ostatních testovaných aplikací. Text je umístěn na obrazovce a na konci textu jsou tlačítka s možnými volbami. Hod kostkou (nebo jakýkoliv jiný náhodný jev) zde probíhá pouze pomocí tlačítka (viz obr. 2.6) bez jakékoliv animace. Po stisknutí tlačítka je hráči oznámena vygenerovaná náhodná hodnota a podle ní se hráči zpřístupní akce, kterou může provést. Podobně je implementován boj, viz obr. 2.7, který je možné procházet po jednotlivých kolech nebo nechat boj odsimulovat celý najednou. Aplikace obsahuje také nabídku (**Action Bar**), ve které jsou tlačítka pro zobrazení mapy, restart hry, uložení hry a otevření statistik a inventáře postavy. Pořadí těchto tlačítek lze upravit v nastavení tak, že některé funkce budou zobrazeny přímo na obrazovce. Pořadí sice nemá vliv na funkčnost, ale hráči může umožnit pohodlnější přístup k dané vlastnosti.



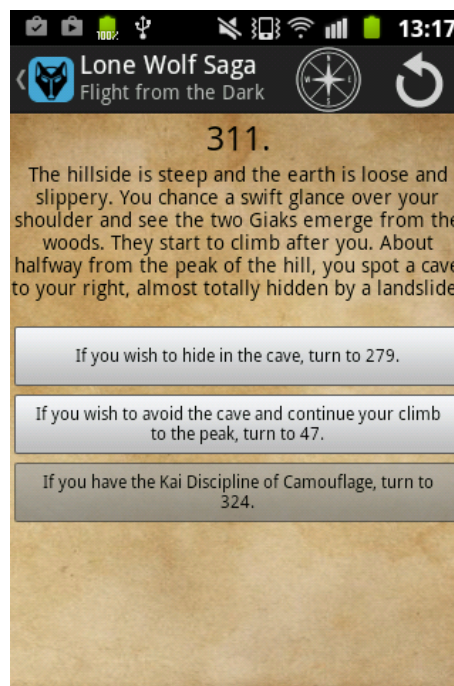
Obrázek 2.8: Ukázka umístění obrázku z aplikace Lone Wolf Saga

Grafika je v aplikaci obsažena ve formě obrázků, které se objevují mezi textem, viz obr. 2.8, a mapy oblasti, v níž se příběh odehrává. Obrázky jsou obsaženy v databázi, kterou je potřeba před zahájením hry stáhnout. Aplikace umožňuje nastavit různá barevná schémata, která změní barvu pozadí a barvu písma. Je také možné měnit font písma. Uživatel si může vybrat, zda bude text zobrazen v systémovém fontu nebo v jednom ze dvou fontů aplikace. Text se ale u těchto dvou fontů velmi špatně

čte, což je v aplikaci, která je založena na četbě velká nevýhoda. Velikost textu je možné také nastavovat, ale pouze v nastavení aplikace. Určitě by bylo vhodnější měnit velikost textu přímo ve hře jako u první testované aplikace. Text také není možné číst v landscape orientaci obrazovky.



Obrázek 2.9: Ukázka vytváření profilu hráče v aplikaci Lone Wolf Saga



Obrázek 2.10: Ukázka, jak schopnost umožňuje více voleb v aplikaci Lone Wolf Saga

Při prvním spuštění aplikace je uživatel vyzván, aby stáhl databázi obsahující gamebooky a obrázky. Dokud databáze není stažena není možné gamebooky procházet (hrát). Stažení databáze se provádí v nastavení aplikace. Databázi je možné uložit na interní úložiště zařízení i na paměťovou kartu. Dalším krokem je vytvoření profilu postavy. Profilů je možné vytvořit více. Při vytváření profilu, viz obr. 2.9, je uživatel vyzván k zadání jména a vygenerování náhodných hodnot pro dvě vlastnosti a to *Combat skill* (bojová dovednost) a *Endurance* (výdrž). Tyto vlastnosti jsou poté využívány v boji a boj přímo ovlivňují. Jako další si musí hráč zvolit 5 schopností, které bude ovládat. Vybrané schopnosti pak hráči při některých akcích odemykají volbu navíc, jak zobrazuje obr. 2.10, ze kterého je vidět, že postava danou vlastností nedisponuje, a proto nemůže vybrat poslední volbu (tlačítko). V průběhu hry se hráč může dostat do fáze hry, ve kterém je nucen sníst jídlo, které má v inventáři. Pokud jídlo nemá, jsou mu odečteny 3 body z výdrže. Aplikace obsahuje více gamebooků. Tyto gamebooky se postupně otevírají po úspěšném dokončení předchozích. Pokud je aplikace ukončena, postup ve hře je zachován, ale při smrti hlavní postavy musí

být daný gamebook hrán od začátku. Pouze v případě, že je navíc zakoupen slot pro ukládání, je možné načíst i uloženou pozici před smrtí.

2.3 Srovnání testovaných aplikací

Z porovnávaných aplikací lze ohodnotit jako nejpropracovanější a nejpovedenější aplikaci **Lone Wolf Saga**, která nabízí mnoho nastavení (velikost písma, použitý font, barevné schéma). Velikost aplikace je oproti ostatním aplikacím poměrně větší, což je dáno tím, že obsahuje tři gamebooky, které jsou postupně zpřístupňovány po dokončení předchozích. Vlastnosti postavy skutečně ovlivňují průběh příběhu. Například při boji atributy a vlastnosti postavy určují šanci na výhru. To je rozdíl oproti aplikaci **The Forging of a Legend**, ve které se zdá, že výsledek boje záleží pouze na náhodě. Jediné negativum u této aplikace je nečitelnost některých použitých fontů.

Druhá testovaná aplikace **Choice of Broadsides** je po grafické i herní stránce velmi jednoduchá a nelze jí nic vytknout, i když aplikace neobsahuje žádnou grafiku ani žádné možnosti nastavení (možná právě pro jednoduchost ji nelze nic vytknout). Jednotlivé volby jsou velmi dobře zpracovány (mnoho možností, podrobně popsány). Jako jediná obsahuje možnost procházet gamebook v **landscape** orientaci obrazovky.

Grafika v aplikaci **The Forging of a Legend** četbu velmi oživuje a aplikace tak vypadá velmi dobře. Na rozdíl od ostatních vybraných aplikací umožňuje sdílení nebo vazbu na sociální síť, což je v dnešní době u mobilních aplikací obvyklé.

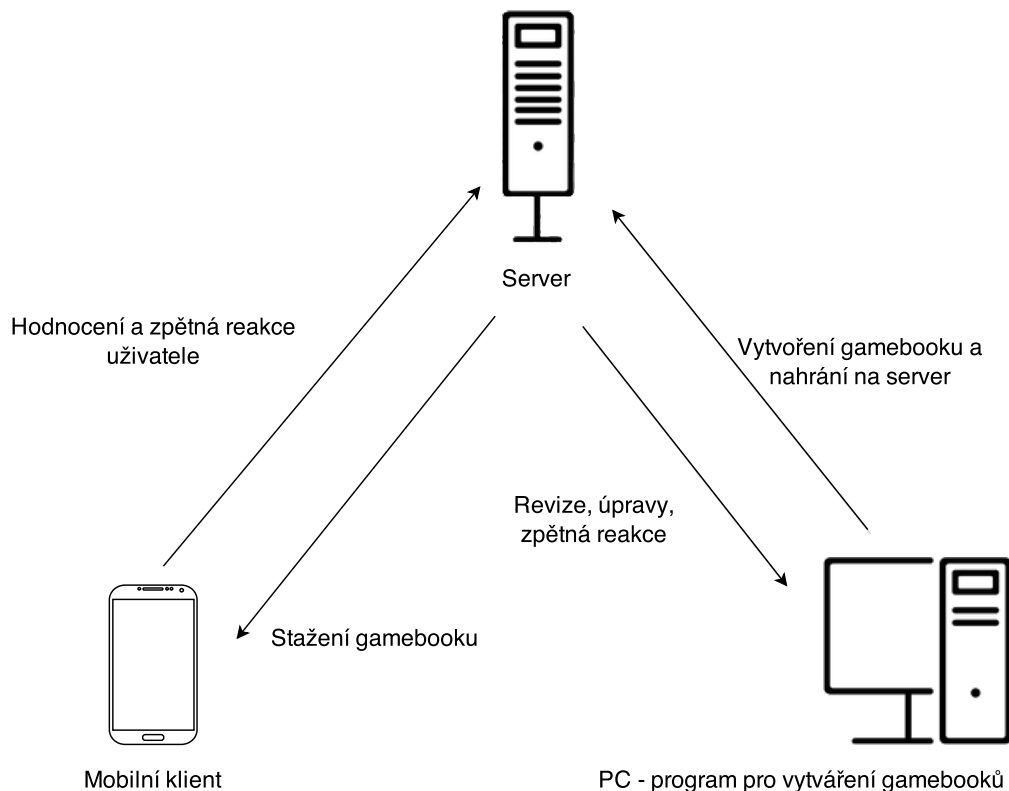
Díky porovnání těchto aplikací, jsme získali přehled o aplikacích podobných té, která bude realizována jako mobilní klient. Při implementaci je tak možné vyhnout se negativům, která byla v testovaných aplikacích nalezeny. A naopak se inspirovat užitečnými funkcemi z těchto aplikací.

Překvapující bylo zjištění, že se nepodařilo dohledat žádnou aplikaci s gamebookem v českém jazyce, ani najít veřejně dostupný systém, který se podobá vyvíjenému systému, a zároveň by v něm bylo možné napsat gamebook a vydat ho, ať už jako samostatnou aplikaci nebo jako část aplikace. Společnost (Choice of Games LLC), která vytvořila aplikaci **Choice of Broadsides**, umožňuje pomocí svého skriptovacího jazyka **ChoiceScript** vytvářet gamebooky, viz [5]. Vytvořený gamebook, ale musí být odeslán emailem této firmě a až poté z něj firma vytvoří aplikaci, kterou je možné publikovat, ovšem vytvořený gamebook musí být pouze v anglickém jazyce. Díky tomu zde vzniká prostor pro vytvoření systému, který by umožňoval publikovat gamebooky i v jiných jazycích než v anglickém. Vytváření gamebooků pomocí tohoto systému by mohlo být uživatelsky příjemnější (např. pomocí programu s gra-

fickým rozhraním) a vydání gamebooku by mohlo probíhat automatizovaně. Protože vyvíjený systém by byl první svého druhu, mohl by být úspěšný.

3 Systém pro publikování gamebooků

Vyvíjený systém pro publikování gamebooků (dále jen systém) se skládá ze tří hlavních částí a je vyvíjen týmem, který se zatím skládá ze dvou členů, tj. Pavel Příbáň a Jakub Sido. První částí systému je program (editor), který umožňuje tvorbu gamebooku. Druhou částí je server, na kterém jsou gamebooky uloženy. Poslední část představuje mobilní klient (mobilní aplikace), který umožňuje gamebooky procházet (hrát, číst) a jeho realizace je předmětem této práce. Jednotlivé části systému spolu mohou komunikovat prostřednictvím serveru, viz obr. 3.1, na kterém jsou znázorněny jednotlivé části systému a naznačena základní komunikace. V následujících podkapitolách budou jednotlivé části systému popsány podrobněji.



Obrázek 3.1: Schéma systému pro publikování gamebooků

3.1 Program pro tvorbu gamebooku

Program (dále jen editor) umožňuje registrovaným uživatelům vytvářet a publikovat gamebooky. Aby uživatel mohl začít psát gamebook, musí se nejprve registrovat a stáhnout editor. Při vytváření gamebooku je uživatel v aplikaci ověřen pomocí údajů zadaných při registraci. Editor je programován v jazyce **JAVA**, stejně jako mobilní klient. Obě aplikace (mobilní klient a editor) využívají sdílenou knihovnu. Sdílená knihovna obsahuje některé třídy a metody (např. načítání souboru s gamebookem), jež by bylo třeba implementovat v obou aplikacích. Je tedy výhodné využít stejný zdrojový kód. Použití sdílené knihovny je možné, díky tomu, že SDK platformy Android je z části založené na JDK. Sdílená knihovna je programována pouze s využitím vybraných funkcí a nástrojů, které jsou průnikem SDK a JDK. Díky tomu je možné knihovnu využít v editoru i v mobilním klientovi. SDK (Software Development Kit) je tvořen sadou nástrojů a API umožňující vývoj aplikací (zde pro platformu Android). JDK (Java Development Kit) obsahuje sadu základních nástrojů pro vývoj na platformě **JAVA**. API (Application Programming Interface) je rozhraní pro programování aplikací.

3.2 Server

Server poskytuje služby mobilnímu klientovi, editoru a uživatelům, kteří systém využívají k tvorbě nebo editaci gamebooků. Server je implementován jako webový server. Pro komunikaci se serverem se využívá protokol **HTTP**. Na serveru je také umístěna **MySQL** databáze, ve které jsou uloženy jednotlivé gamebooky, jejich hodnocení, uživatelé, překlady a korektury gamebooků a další informace. Přístup k těmto datům, tj. čtení, zápis a úprava, je poskytován jednotlivým částem systému pomocí níže popsaných služeb.

3.2.1 Služby pro editor

Hlavní službou, kterou server poskytuje editoru, je možnost nahrát vytvořený gamebook na server a uložit jej do databáze. Služby poskytované editoru zde nebudou uvedeny podrobně, protože jsou určeny pro potřeby editoru, jehož detailní popis není součástí této práce.

3.2.2 Uživatelské služby

Služby pro uživatele budou implementovány formou webových stránek, které jsou vyvíjené, ale zatím nejsou veřejně dostupné. Po jejich dokončení budou dostupné z adresy <http://malpal.cz/>. Stránky jsou programovány pomocí jazyka PHP a využití frameworku *Nette*. Framework pomáhá usnadnit a urychlit vývoj softwaru tak, že se vývojáři snaží poskytnout řešení rutinních činností a problémů. Framework může být obecně knihovnou nebo šablonou, která se dále rozšiřuje, popř. je využíváno jejích služeb. Aby uživatel mohl začít vytvářet gamebooky, bude se muset nejprve na stránkách zaregistrovat. Mezi činnostmi, které budou moci registrovaní uživatelé vykonávat, patří:

- vytváření gamebooku,
- provádění korektury před vydáním gamebooku,
- překládání gamebooku do jiného jazyka.

Pro vytváření gamebooku je potřeba mít nainstalovaný editor na vlastním počítači. Překládat nebo provádět korekturu bude možné přímo na webových stránkách. Korektura gamebooku se provádí před jeho vydáním. Korektor má za úkol gamebook zkontrolovat, a případně opravit gramatické a pravopisné chyby.

3.2.3 Služby pro mobilního klienta

Tyto služby jsou poskytovány mobilnímu klientovi a jsou realizovány pomocí HTTP protokolu a dotazů prováděných na webový server. Webový server na požadovaný dotaz vygeneruje HTML stránku, ze které klient získá informace, o něž v dotazu žádal. Služby, které server poskytuje pro mobilního klienta jsou:

- vygenerovat seznam dostupných gamebooků,
- vygenerovat odkaz pro stažení gamebooku,
- přijmout hodnocení daného gamebooku od klienta.

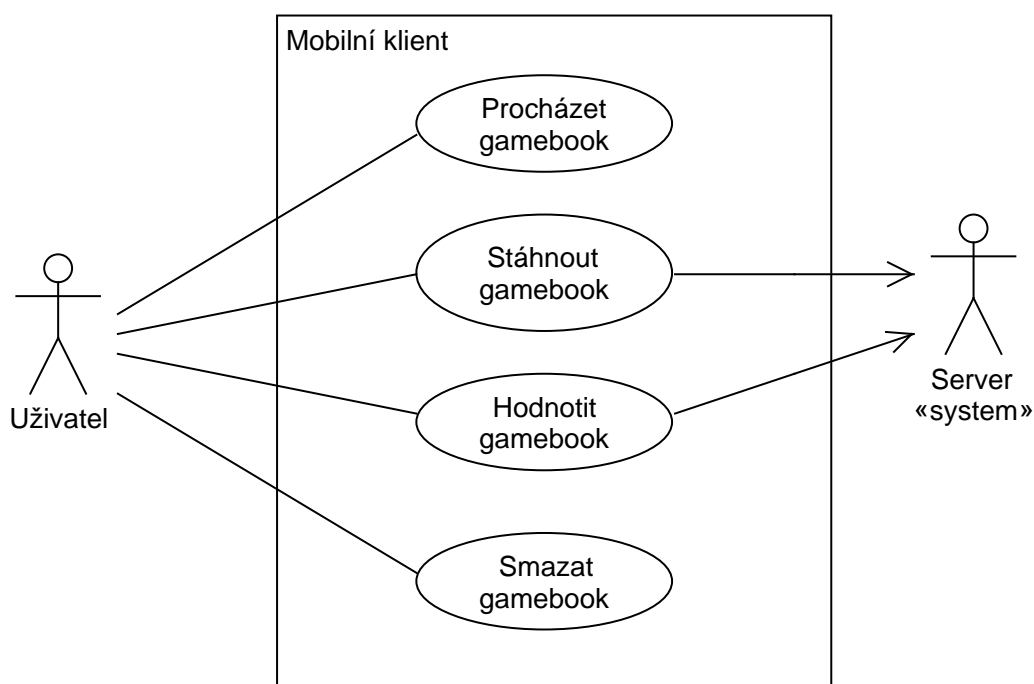
3.3 Mobilní klient

Úkolem mobilního klienta je umožnit uživatelům (nejen registrovaným v systému) procházení vytvořených gamebooků na platformě Android. Při instalaci klienta (apli-

kace) by měla aplikace obsahovat některé již hotové gamebooky. Tyto gamebooky budou distribuovány spolu s aplikací, protože jejich autoři se podíleli na testování systému v jeho počátcích a byli prvními autory, proto budou takto zvýhodněni. Při vydání (publikování) nového gamebooku by měla být aplikace schopna nový gamebook do zařízení stáhnout. Základní funkcionalitu klienta tedy budou tvořit tyto dva body:

- procházení gamebooků,
- stažení nově vydaných gamebooků.

Dále by aplikace měla uživateli umožnit zpětnou vazbu, např. možnost ohodnotit vytvořený gamebook. Na obrázku 3.2 je umístěn UML diagram použití, který znázorňuje základní funkcionalitu mobilního klienta. Realizace mobilního klienta je předmětem této bakalářské práce. Klient bude programován v jazyce JAVA, a jak bylo zmíněno v kap. 3.1, využívá funkcionalitu sdílené knihovny. Podrobný návrh mobilního klienta bude popsán v kap. 4. UML (Unified Modeling Language) je „univerzální jazyk pro vizuální modelování systémů“ [16], který se používá především v softwarovém inženýrství. Při vytváření UML diagramů v této práci jsem čerpal z [16] a [17].



Obrázek 3.2: UML diagram použití mobilního klienta

3.4 Reprezentace gamebooku

Gamebook v knižní (tištěné) podobě je uchováván jako klasická kniha, ale uložit vytvořený elektronický gamebook pouze jako text není příliš vhodné a to především z důvodu pozdějšího zpracování a procházení gamebooku na jiné části systému (např. mobilním klientovi). Je proto potřeba použít sofistikovanější přístup. Proto byl pro reprezentaci gamebooku zvolen orientovaný graf, který je možné použít, jak bylo vysvětleno v kapitole 2.1. Vrcholem grafu je zde stav, neboli část textu, po které následuje rozhodnutí hráče. Hrany mezi vrcholy grafu představují jednotlivé akce (rozhodnutí), pro které se může hráč rozhodnout. Graf je reprezentován XML souborem, který je popsán v kapitole 3.4.1. Protože je graf ukládán do souboru typu XML, který je podporován mnoha programovacími jazyky, bylo docíleno toho, že soubor je nezávislý na platformě, na které je zpracováván.

3.4.1 Struktura XML souboru

Kořenovým elementem XML souboru je element `ROOT`, který obsahuje element `Info`. V tomto elementu jsou umístěny počáteční hodnoty atributů postavy, se kterou bude hráč gamebook procházet. Na stejné úrovni jako `Info`, je obsažen element `Graph`, jehož atributy obsahují informace o gamebooku (`id`, jméno autora, název gamebooku atd.), viz ukázka kódu 3.1, na které je pro ilustraci zobrazena zjednodušená ukázka souboru (neobsahuje všechny elementy a atributy). Tento element (`Graph`) obsahuje všechna data a informace daného gamebooku. Uvnitř jsou vnořeny elementy `State`, které představují jednotlivé stavy v grafu gamebooku (tj. část textu, po které se může hráč rozhodnout, jak bude pokračovat). Každý takový stav obsahuje elementy `Action`, tedy akce, které může hráč v daném stavu provést. V elementu `Action` jsou uloženy informace potřebné pro vyhodnocení dané akce. Atribut `id` je jedinečný identifikátor akce v gamebooku a atribut `finalID` udává, do jakého dalšího stavu se má příběh přesunout po vykonání této akce.

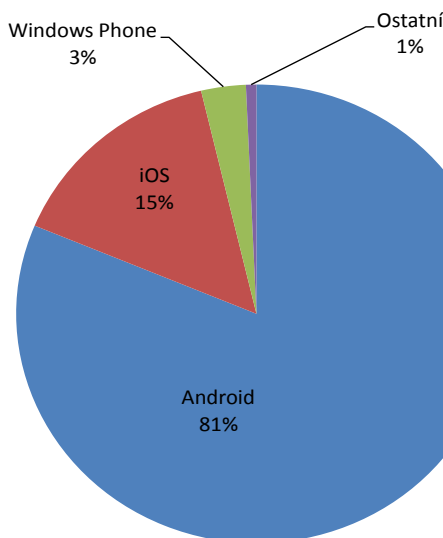
```
<Graph
  id="103"
  author="Jindra"
  title="Příběh Benjamina Davise" >
  <State
    id="1"
    desc="Píše se rok 1963. Do vesmíru se právě dostala
      1. žena.">
    <Action
      id="154"
      desc="Dám na jistotu a pojedu autobusem"
      finalID="2"
      type="AKCE_PROSTA" />
    <Action .... />
  </State>
  ...
</Graph>
```

Ukázka kódu 3.1: Zjednodušená ukázka struktury XML souboru

4 Návrh mobilního klienta

Mobilní klient je předmětem této práce a měl by obsahovat základní funkcionalitu, tj procházení gamebooku, jeho hodnocení a stažení nově vydaného gamebooku, také viz kap. 3.3. Mobilní klient bude navržen pro platformu Android od verze 2.3 (Gingerbread). Hlavním důvodem pro výběr platformy Android je především její většinový podíl na trhu s chytrými telefony, viz obr. 4.1 (data převzata z [2]). Jako nejnižší podporovaná verze byla zvolena verze 2.3, protože podíl mobilních zařízení s operačním systémem s verzí 2.3 a výše je také většinový, viz tab. 4.1 (data převzata z [13]). Dalšími platformami, které přicházely v úvahu pro vývoj byly platformy iOS a Windows Phone.

Podíl operačních systémů chytrých telefonů za rok 2014



Obrázek 4.1: Podíl operačních systémů chytrých telefonů za rok 2014

Nevýhodou při vývoji aplikací pro platformu iOS a následném publikování aplikace v oficiálním obchodě (App store) je nutnost vlastnit zařízení s operačním systémem OS X (dříve Mac OS X), na kterém probíhá vývoj aplikace. Nevýhodou platformy Windows Phone je zatím její poměrně nízké rozšíření mezi uživateli oproti platformě Android.

Jako výhodu při vývoji pro platformu Android lze považovat rozsáhlou dokumentaci, návody, ukázky a doporučení pro vývojáře, viz [1]. Díky velkému rozšíření této platformy a široké komunitě vývojářů je možné nalézt řešení mnoha problémů i na neoficiálních fórech, např. stackoverflow viz [11].

Verze	Kodové jméno	API	Podíl	Součet
2.2	Froyo	8	0,4%	0,4%
2.3.3 - 2.37	Gingerbread	10	6,9%	99,6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	5,9%	
4.1.x	Jelly Bean	16	17,3%	
4.2.x		17	19,4%	
4.3		18	5,9%	
4.4	KitKat	19	40,9%	
5.0	Lollipop	21	3,3%	

Tabulka 4.1: Procentuální zastoupení jednotlivých verzí platformy Android k 2. březnu 2015

4.1 Návrh síťové komunikace

Pro návrh síťové komunikace mezi mobilním klientem a serverem je nejdříve potřeba definovat jaká a jak velká data budou přenášena. Mezi serverem a klientem budou probíhat tyto síťové komunikace:

1. stažení gamebooku ze serveru,
2. odeslání hodnocení gamebooku uživatelem,
3. stažení seznamu dostupných gamebooků ze serveru.

Předpokládaná velikost textu gamebooků je v řádech stovek kB až jednotek MB. U těchto textů by ale v budoucnu měla přibýt ještě grafika (obrázky). Velikost výsledného souboru, který bude stahován, tak může dosáhnout až desítek MB. Tato (potenciální) velikost musí být při návrhu zohledněna. Soubor bude stahován ve formátu XML, pokud nebude obsahovat grafiku. Pokud bude spolu s gamebookem distribuována grafika, budou všechny soubory zkomprimovány a zabaleny do archivu, který bude přenášen jako jediný soubor. Pro body 2 a 3 by mělo platit, že jsou to krátké textové soubory s velikostí v řádech jednotek kB.

4.1.1 Možnosti síťové komunikace

Android API obsahuje standardní balík `java.net` z JDK, který slouží pro síťovou komunikaci. Základní třídou pro síťovou komunikaci pomocí protokolu HTTP (resp.

HTTPS) je `HttpURLConnection`. Komunikovat lze také pomocí tzv. socketů. Pro protokol TCP existuje třída `Socket` a protokol UDP využívá třídu `DatagramSocket`. Sockety jsou nástroje (API), které se využívají v síťové komunikaci zpravidla pro model **klient - server**. Oba protokoly (TCP, UDP) jsou zařazeny do 4. (transportní) úrovně referenčního modelu ISO/OSI. HTTP je protokol 7. (aplikační) vrstvy modelu ISO/OSI, viz [6].

4.1.2 Dostupné knihovny

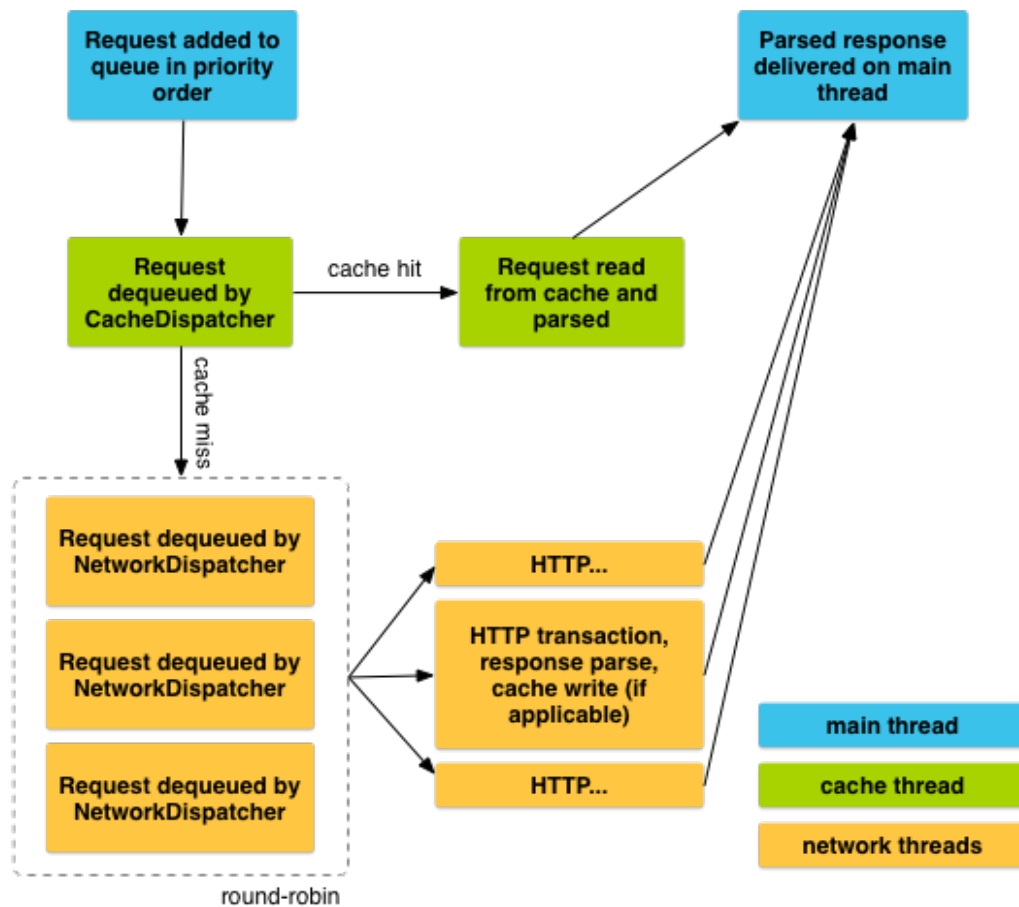
Pro zjednodušení síťové komunikace pomocí protokolu HTTP lze využít knihovnu Volley (dostupná z <https://android.googlesource.com/platform/frameworks/volley>). Podle oficiální dokumentace [12] by knihovna měla implementaci síťové komunikace zjednodušovat a urychlovat. Mezi některé výhody této knihovny patří:

- asynchronní zpracování a vykonání síťových dotazů,
- implementovaná cache paměť pro opakující se dotazy,
- podpora více souběžných síťových spojení,
- podpora priority jednotlivých dotazů.

Zjednodušený scénář použití knihovny pak může probíhat tak, že nejprve je vytvořen tzv. Request (požadavek, dotaz), tj. instance třídy `Request`, ve kterém je definovaná URL adresa serveru a další parametry HTTP dotazu (např. výběr mezi HTTP metodami GET a POST). Dále jsou při vytváření objektu předány instance dvou tříd (tzv. posluchačů), které implementují rozhraní `Response.Listener` a `Response.ErrorListener`. Metody těchto rozhraní jsou zavolány po asynchronním dokončení požadavku. Metoda `onResponse()` posluchače `Response.Listener` je zavolána při úspěšném dokončení požadavku a metoda `onErrorResponse()` druhého posluchače je zavolána při neúspěšném dokončení. Vytvořený objekt požadavku je přidán do fronty požadavků, která je reprezentována instancí třídy `RequestQueue`. Poté je požadavek asynchronně zpracován knihovnou a je vyvolána jedna z výše zmíněných metod, viz obr. 4.2 (převzato z [12]).

Dalšími funkčně podobnými knihovnami pro komunikaci na platformě Android jsou `Retrofit` (<http://square.github.io/retrofit/>) nebo také `OkHttp` (<http://square.github.io/okhttp/>).

Pro dlouhotrvající stahování poskytuje Android tzv. `DownloadManager`, což je systémová služba. Služba v systému Android je aplikace, která běží na pozadí a nemá vlastní uživatelské rozhraní (UI). Tato služba zajistí stažení souboru i při výpadku



Obrázek 4.2: Životní cyklus síťového požadavku v knihovně Volley

spojení nebo restartování zařízení. Instance třídy `DownloadManager` se získá voláním metody `getSystemService()` s parametrem `Context.DOWNLOAD_SERVICE`. Postup pro stažení souboru je obdobný jako u knihovny Volley, nejdříve je potřeba vytvořit objekt reprezentující požadavek, předat mu adresu souboru a nastavit potřebné parametry (např. místo uložení souboru). Poté je objekt požadavku předán `DownloadManageru`, který se pokusí zajistit jeho stažení. Aby služba mohla aplikaci informovat o průběhu (popř. dokončení, neúspěchu), musí aplikace zaregistrovat tzv. `BroadcastReceiver` (dále receiver). Receiver má za úkol reagovat na systémové události nebo na události jiných aplikací. Pokud nastane vybraná událost, jsou informováni všechny receivers, kteří mají registrovanou tuto událost. V receiveru je implementována reakce na danou událost v případě `DownloadManageru`, tedy např. reakce na úspěšné dokončení, neúspěšné dokončení nebo na zahájení stahování.

4.1.3 Vybrané řešení síťové komunikace

Z výše popsaných možností byla pro implementaci bodů 2 (odeslání hodnocení) a 3 (stažení seznamu dostupných knih) z kap. 4.1 vybrána knihovna `Volley`. Důvodem pro výběr této knihovny je především zjednodušení implementace síťových požadavků oproti implementaci pomocí třídy `URLConnection`. Od verze 3.0 (`Honeycomb`) musí být všechny síťové operace prováděny mimo hlavní vlákno aplikace. Knihovna vykonává požadavky automaticky asynchronně a vývojář tak nemusí tento problém řešit.

Velikost stahovaného gamebooku může v budoucnu dosahovat až desítek MB, proto bude stahování gamebooků implementováno pomocí systémové služby (`DownloadManager`). `DownloadManager` je podle [14] přímo určen pro stahování velkých souborů. Velkou výhodou `DownloadManageru` je, že sám dokáže řešit výjimečné situace jako je přerušení síťového spojení nebo restart systému. Na obr. 4.3 je zobrazen UML diagram aktivit znázorňující chování aplikace při stahování gamebooku.

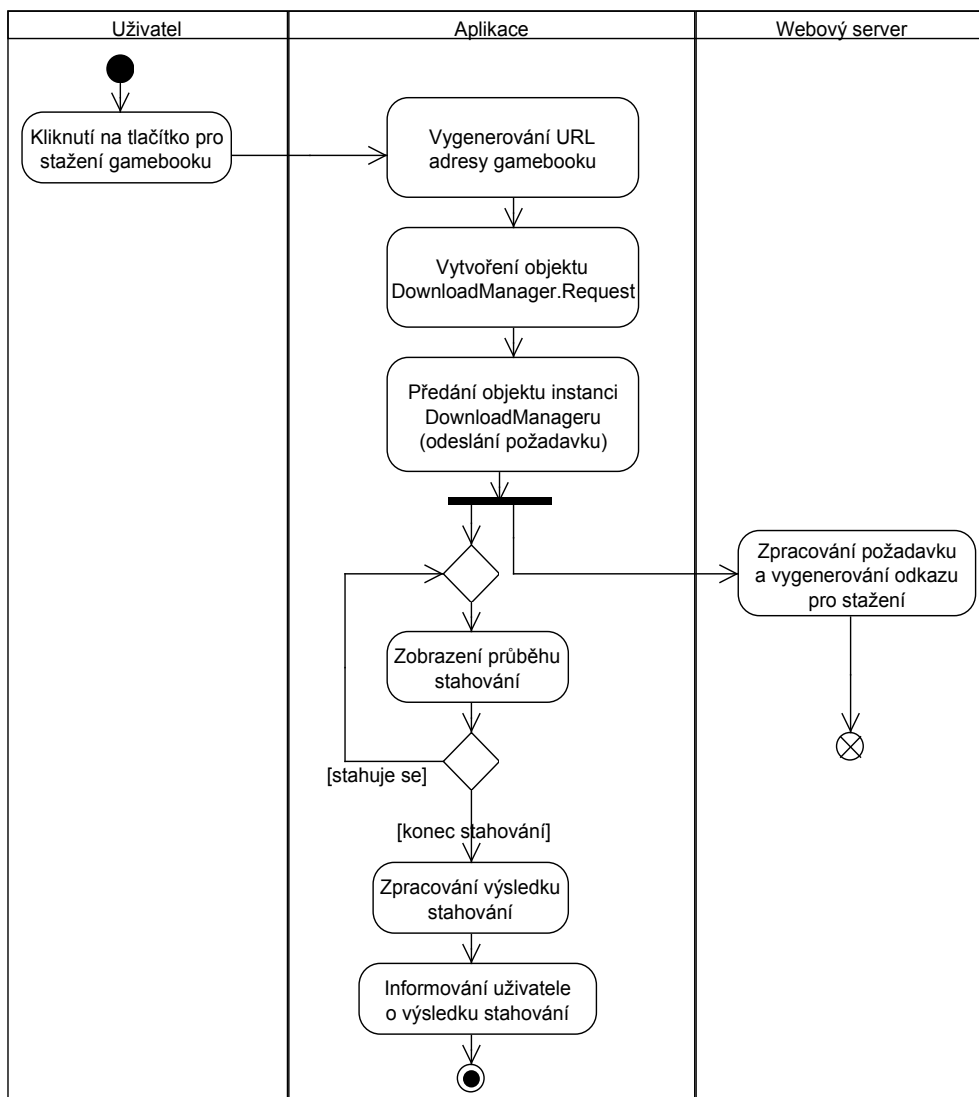
4.2 Způsob uložení dat

Aplikace bude pracovat s daty, která reprezentují jednotlivé gamebooky (např. XML soubor), s grafickými soubory a s lokálními daty reprezentující nastavení aplikace. Tato data je potřeba v aplikaci nějak uložit. Pro perzistentní uložení dat v zařízení, viz [15], nabízí Android tyto tři možnosti:

1. uložení základních datových typů ve formátu **Klíč - Hodnota** pomocí třídy `SharedPreferences`,
2. uložení souboru do vnitřního nebo vnějšího úložiště,
3. uložení dat do SQLite databáze.

4.2.1 Sdílené preference

Sdílené preference jsou určeny pro ukládání relativně malých dat, která jsou reprezentována základními datovými typy (`boolean`, `long`, `int`, `float`), navíc je možné ukládat objekt třídy `String` a kolekci této třídy (množinu) reprezentovanou rozhraním `Set<String>`. Data jsou ukládána do souboru uloženého na privátním



Obrázek 4.3: UML diagram aktivit znázorňující chování aplikace při stahování gamebooku

úložišti aplikace (jiné aplikace ani uživatel se k těmto datům nemohou dostat¹).

Pro uložení dat je potřeba získat instanci třídy `SharedPreferences`. Třída se získá voláním metody `getSharedPreferences()` nad objektem třídy `Context` nebo jejich podtříd. Parametry metody jsou jméno souboru, do kterého se budou data ukládat a mód, v jakém bude soubor otevřen (např. přepsání pokud už soubor existuje). Následným zavoláním metody `edit().putInt(klic, hodnota)` nad získaným objektem třídy `SharedPreferences` uložíme proměnnou datového typu `int`,

¹Neplatí pro zařízení, které jsou tzv. „rootnuté“, kde uživatel má práva roota jako v klasickém linuxovém systému.

kde "klic" je jedinečný řetězec a "hodnota" je ukládaná proměnná. Načtení proměnné se provede zavoláním metody `getInt(klic, defHodnota)`, kde argument "defHodnota" je hodnota, která bude vrácena, pokud číslo pod zadaným klíčem nebude nalezeno. Pro ostatní datové typy je postup obdobný pouze se liší názvy metod a ukládané datové typy. Výhodou sdílených preferencí je jednoduchá implementace ukládání a načítání dat, ale nevýhodou je možnost ukládat jen základní datové typy.

4.2.2 Ukládání souborů

Práce se soubory na platformě Android probíhá stejně jako u jazyka JAVA, tj. pomocí tříd z balíku `java.io`. Soubory je možné ukládat na vnitřní (internal) nebo vnější (external) úložiště. Vnitřním úložištěm je myšlena vestavěná non-volatile paměť. Non-volatile paměť je druh paměti, která dokáže uchovávat data i po odpojení napájení. Vnitřní úložiště není možné ze zařízení vyjmout, je jeho součástí. Vlastnosti vnitřního úložiště jsou:

- je vždy dostupné,
- soubory uložené ve vnitřním úložišti jsou přístupné pouze aplikaci, které vnitřní úložiště náleží,
- při odinstalaci aplikace jsou systémem odstraněna všechna uložená data v tomto úložišti, která náleží odinstalovávané aplikaci.

Vnější úložiště je zpravidla reprezentováno vyměnitelnou kartou typu SD a liší se těmito rozdíly:

- nemusí být vždy dostupné (uživatel může vyjmout SD kartu),
- přístup k souborům není obvykle nijak omezen a každý (uživatel, jiná aplikace) je může číst a měnit,
- při odinstalaci nejsou soubory vytvořené aplikací odstraněny.

Popsané vlastnosti mohou být chápány jako výhody i nevýhody, proto se každé úložiště hodí pro jiný typ dat. Na interní úložiště je vhodné ukládat data, ke kterým chceme, aby přistupovala pouze naše aplikace a ostatním aplikacím zůstala skrytá. Naopak vnější úložiště je vhodné pro zpřístupnění dat uživateli nebo jiným aplikacím.

4.2.3 Databáze SQLite

SQLite je Open Source databáze. Open Source znamená, že software má otevřený zdrojový kód (zdrojový kód je volně dostupný). Více informací o SQLite je dostupných z <http://www.sqlite.org/>. SQLite databáze je součástí operačního systému Android a každá aplikace ji může využívat. Pro manipulaci s databází (vytváření tabulek, dotazů atd.) je potřeba vytvořit třídu která dědí od třídy `SQLiteOpenHelper`. V této třídě je potřeba překrýt metody `onCreate()` a `onUpgrade()`.

Metoda `onCreate()` je automaticky volána při vytvoření databáze. V této metodě by měly být provedeny DDL příkazy jazyka SQL pro vytvoření tabulek databáze. DDL (Data Definition Language) jsou příkazy pro vytváření a upravování struktury databáze. Pokud databáze neexistuje, je při získání reference na databázi metodou `getWritableDatabase()` nebo `getReadableDatabase()` zavolána metoda `onCreate()` a vytvořena databáze. DML i DDL příkazy jazyka SQL, jsou prováděny pomocí metod třídy `SQLiteDatabase`. DML (Data Manipulation Language) jsou příkazy jazyka SQL pro manipulaci s daty v databázi, např. vkládání, mazání nebo změna dat.

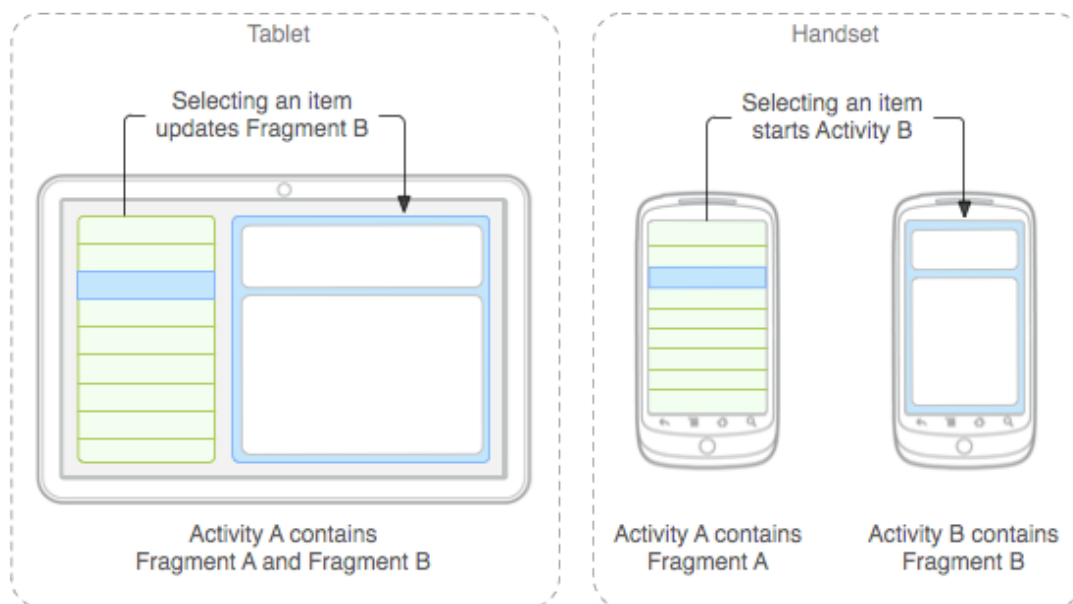
4.2.4 Zvolený způsob pro uložení dat

Sdílené preference budou v aplikaci použity pro uložení pomocných dat (např. informace o aktuálně čtené knize). Do souborů budou dočasně ukládány stažené gamebooky před tím než budou zpracovány a importovány do databáze.

Pro uložení gamebooku v aplikaci bude použita SQLite databáze. Jedním z důvodů pro použití databáze k uložení gamebooku je struktura XML souboru, ve kterém je uložen gamebook při stahování ze serveru (viz. kap. 3.4.1). Jednotlivé elementy XML souboru budou reprezentovat tabulky v databázi a atributy těchto elementů (popř. některé vnořené elementy) budou tvořit sloupce tabulek. Dalším důvodem pro použití databáze je, že není potřeba při procházení gamebooku načítat celý soubor s gamebookem do paměti. Při procházení gamebooku pak budou v paměti načtena jen data, která jsou právě používána (např. jen text k aktuálnímu stavu, ve kterém se hra nachází). ER diagram znázorňující strukturu databáze je vidět na obr. 4.4.

Důležitou tabulkou databáze je tabulka `Books`, ve které jsou uloženy informace o stažených gameboocích. Tabulka obsahuje informace, tj. jednotlivé sloupce o daném gamebooku jako je id knihy (gamebooku), název knihy, jméno autora, id počátečního stavu a další. Primárním klíčem této tabulky je id knihy, což je číslo, které je v celém vyvíjeném systému unikátní a jednoznačně určuje gamebook. Toto id je

aplikace. Mobilní zařízení, která využívají operační systém Android se velmi často liší velikostí obrazovky, např. tablet a mobilní telefon. Proto byla do Android API přidána možnost využívat tzv. fragmenty. Fragmenty se zpravidla používají jako část UI, kdy se uživatelské rozhraní vytváří podle velikosti obrazovky. Jednotlivé fragmenty se pak skládají podle potřeby. Na zařízení s malou obrazovkou se zobrazí pouze jeden fragment, zatímco ve stejné aplikaci na zařízení s velkou obrazovkou jsou najednou zobrazeny dva fragmenty, což velmi názorně ilustruje obr. 4.5 (převzat z [18]). Zařízení s velkou obrazovkou budou v tomto textu považována za zařízení, která jsou podle [18] označena jako *large* (min. 640dp x 480dp) a *xlarge* (min. 960dp x 720dp). Jednotka dp (Density-independent pixel) představuje virtuální pixel obrazovky, který je nezávislý na hustotě pixelů dané obrazovky, detailněji viz [18]. Každý fragment musí být umístěn v hostitelské aktivitě, ale aktivita může existovat nezávisle na fragmentech.



Obrázek 4.5: Použití fragmentů na zařízeních s různou velikostí obrazovky

Velkou výhodou při vývoji aplikace pro platformu Android je, že dovoluje oddělit logiku aplikace (zdrojový kód jazyka `JAVA`) od definice vzhledu aplikace. Vzhled aplikace se definuje v `XML` souboru, ve kterém jsou obsaženy jednotlivé grafické komponenty. Tyto komponenty jsou reprezentovány jednotlivými elementy v `XML` souboru. Při manipulaci s UI ve zdrojovém kódu jsou jednotlivé elementy reprezentovány třídami, které jsou potomky třídy `View`. UI je možné definovat i programově, ale takový přístup není příliš častý, použije se např. při dynamickém vytváření UI.

Dále budou popsány základní a nejdůležitější aktivity aplikace a jejich vlastnosti, tzn. co by měly umožňovat a jak by měly vypadat. Při návrhu bude u některých aktivit zohledněna i podpora zařízení s větší velikostí obrazovky (tablety). Aktivity budou reprezentovány třídami, které jsou potomky třídy `FragmentActivity`. V implementaci nebudou dědit přímo od třídy `FragmentActivity`, ale od tříd z různých knihoven třetích stran, které dědí od `FragmentActivity`. Tyto knihovny budou popsány v kap. 5.

4.3.1 Aktivita `MenuActivity`

Tato aktivita bude sloužit jako menu, které bude zobrazeno po spuštění aplikace. Aktivita by měla umožňovat snadný pohyb v aplikaci, tj. pokračovat v rozečteném gamebooku, spustit nové procházení, otevřít knihovnu (viz kap. 4.3.3) a otevřít nastavení aplikace. Pro přehlednost bude umístěna část tlačítek (akcí) na hlavní obrazovku a část jich bude přesunuta do komponenty `Action Bar`, viz obr. 4.6.

4.3.2 Aktivita `GameActivity`

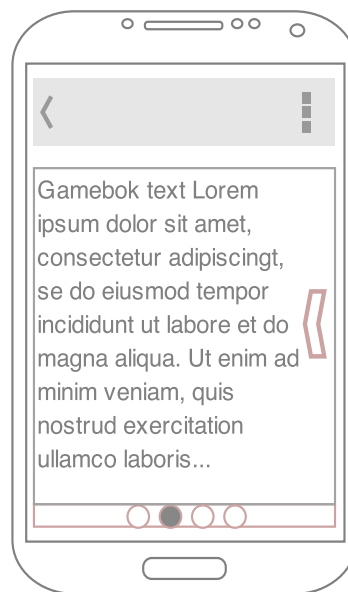
V této aktivitě bude probíhat prohlížení (čtení) gamebooku. Na obrazovce se zobrazí text, který bude rozdělen do stránek. Uživateli bude umožněno se mezi stránkami pohybovat táhnutím prstu po obrazovce. Počet stránek a indikace aktuálně zobrazované stránky bude reprezentována indikátorem na dolním okraji obrazovky, viz obr. 4.7. Na okraji obrazovky bude umístěno tlačítko pro vysunutí nabídky s akcemi, které je možné provést. Nabídka s akcemi a zobrazovaný text budou umístěny do fragmentů, ale bude použita vždy jen jedna aktivita. Na zařízeních s velkou obrazovkou a v režimu `landscape` bude nabídka vždy zobrazena vedle textu, jinak se bude nabídka vysouvat z pravého okraje obrazovky.

4.3.3 Aktivita `LibraryActivity`

Aktivita `LibraryActivity` reprezentuje knihovnu umožňující zobrazit seznam stažených a seznam dostupných gamebooků, které je možné stáhnout. Tato aktivita bude využívat fragmenty. Na velkých obrazovkách v `landscape` orientaci obrazovky bude v této aktivitě zobrazen fragment se seznamem gamebooků a vpravo bude umístěn fragment s detailem vybraného gamebooku (viz obr. 4.5). Na menších obrazovkách a na všech obrazovkách v `portrait` orientaci bude nejprve zobrazen pouze fragment se seznamem a po kliknutí na daný gamebook bude spuštěna aktivita `DetailBookActivity` s detailními informacemi o gamebooku. Fragment s detailem gamebooku



Obrázek 4.6: Grafický návrh aktivity MenuActivity



Obrázek 4.7: Grafický návrh aktivity GameActivity

bude obsahovat tlačítka pro stažení, smazání a vybrání gamebooku v závislosti na tom, zda se zobrazují stažené gamebooky nebo gamebooky ke stažení. Fragment bude také umožňovat hodnocení gamebooku.

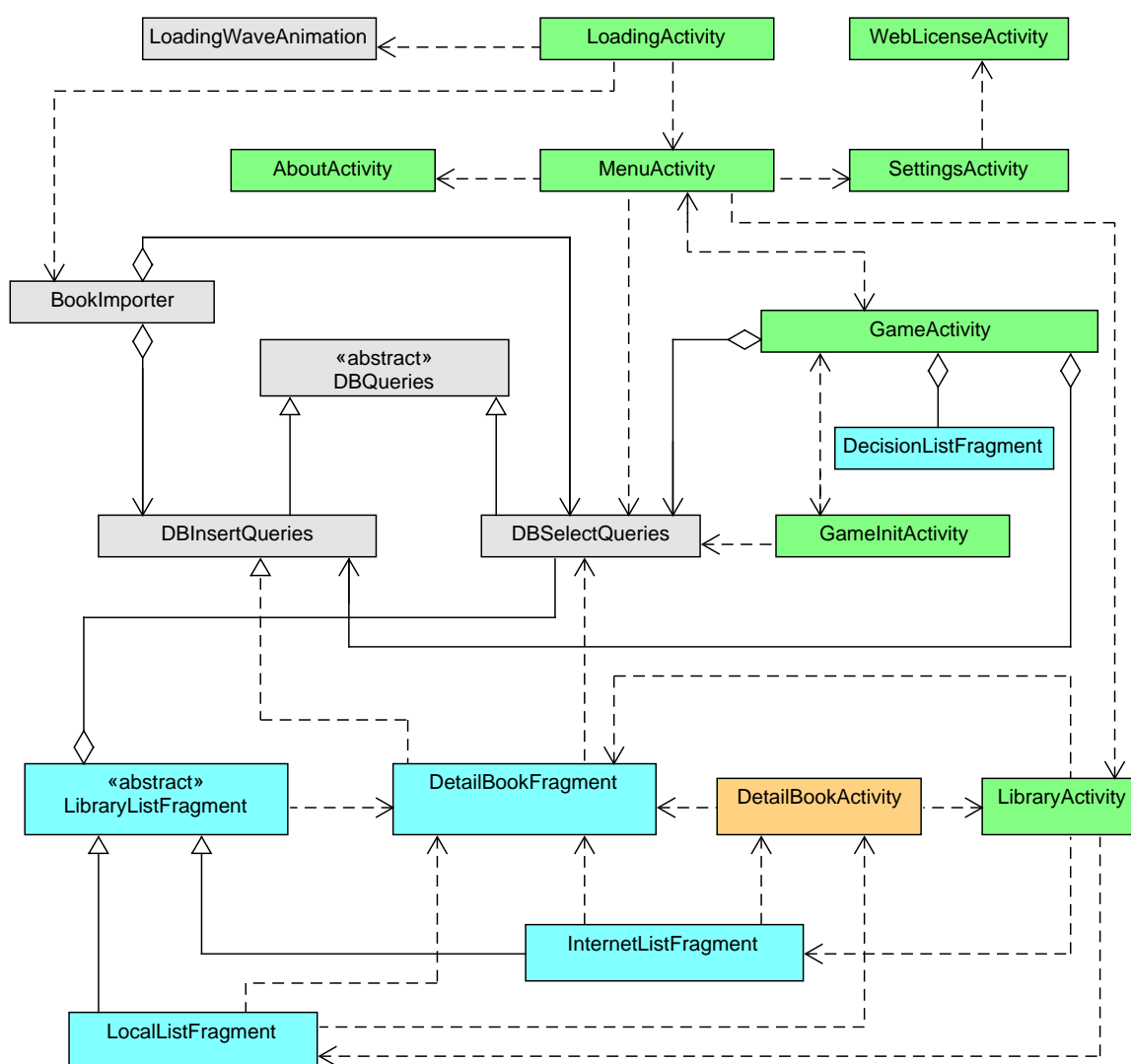
4.3.4 Ostatní aktivity

V aplikaci budou také implementovány další aktivity. Aktivita **AboutActivity** zobrazí obecné informace o programu. **GameInitActivity** bude určena k nastavení hry před jejím začátkem. **LoadingActivity** bude aktivita, která bude spuštěna jako první aktivita aplikace a bude mít za úkol naimportovat do databáze gamebooky distribuované s aplikací a poté spustit **MenuActivity**. **SettingsActivity** bude umožňovat úpravu nastavení aplikace. Ve **WebLicenseActivity** budou vyjmenovány použité knihovny třetích stran a licence, pod kterými jsou distribuovány.

Na obr. 4.8 je vidět UML diagram tříd s aktivitami, fragmenty a dalšími důležitými třídami, které budou implementovány (neobsahuje všechny třídy aplikace). Pro lepší orientaci v diagramu mají třídy aktivit zelené pozadí, fragmenty modré a ostatní třídy šedé pozadí.

4.3.5 Příprava grafických souborů

V aplikaci bude potřeba použít obrázky a jinou grafiku např. jako pozadí nebo ikony. Proto byly vytvořeny obrázky, které budou ve výsledném projektu uloženy v adresářích `/res/drawable`, `/res/drawable-hdpi`, `/res/drawable-mdpi`, `/res/drawable-xhdpi`. V adresářích se jmény `/res/drawable-xxxx` jsou uloženy stejné obrázky, ale v různých rozlišeních, protože jak již bylo zmíněno, platforma Android je používána na mnoha zařízeních s různou velikostí a kvalitou (hustotou pixelů) obrazovky. Z tohoto důvodu je potřeba některé obrázky distribuovat ve více rozlišeních a až na konkrétním zařízení dle velikosti obrazovky je rozhodnuto, jaké rozlišení obrázku bude použito.



Obrázek 4.8: UML diagram tříd aplikace

5 Realizace mobilního klienta

V této kapitole bude popsána implementace mobilního klienta. Klient byl programován v Jazyce JAVA, což je primární jazyk pro programování na platformě Android. Nejnížší verze platformy Android, na které lze aplikaci spustit je 2.3 (Gingerbread). Aplikace je lokalizována do anglického a českého jazyka, ale gamebooky jsou zatím napsány pouze v českém jazyce. K aplikaci je dostupná uživatelská příručka, která je umístěna v příloze A a zdrojové soubory aplikace jsou umístěny na příloženém CD (viz příloha B). Při implementaci jsem čerpal z těchto zdrojů: [1], [7], [8], [9] a [10]. Při hledání řešení problémů jsem využíval především [11]. Aplikace byla vytvořena ve vývojovém prostředí Eclipse s nainstalovaným pluginem ADT (Android Development Tools). Při implementaci bylo využíváno verzovacího systému Git.

5.1 Struktura aplikace

Zdrojové kódy aplikace jsou umístěny v jednotlivých balíčcích v adresáři `/src`. Třídy jsou rozděleny do balíčků podle funkčnosti (typu) třídy, např. aktivity jsou v samostatném balíčku a fragmenty jsou také ve svém samostatném balíčku, i když spolu nemusí třídy v jednom balíčku logicky souviset. Toto rozdělení se ukázalo při programování jako velmi praktické pro orientaci při větším počtu zdrojových souborů.

V adresáři `/assets` je umístěn soubor `licenses.html`, který obsahuje seznam použitých open-source knihoven a licence, pod kterými je možné je používat. Dále je zde umístěn konfigurační soubor `logback.xml` pro knihovnu `logback-android`, která je v projektu použita pro logování, viz kap. 5.2.1. Adresář také obsahuje soubory s příponou `TTF`, které popisují font písma.

Poslední složkou, ve níž programátor upravuje a přidává soubory je `/res`, zde jsou k nalezení v jednotlivých adresářích všechny definice UI ve formě XML souborů, veškerá použitá grafika, textové řetězce a další potřebné soubory (zdroje). V adresáři `/res/raw` jsou umístěny soubory gamebooků, které jsou distribuované spolu s aplikací.

5.1.1 Soubor `AndroidManifest.xml`

Soubor `AndroidManifest.xml` musí být obsažen v každé aplikaci pro platformu Android. Obsahuje všechny důležité informace o aplikaci - jako jsou komponenty aplikace (aktivity, služby, content providery atd.), podrobnější obsah viz [19]. Dů-

ležitý je element `uses-sdk`, do kterého se zapíše číslo minimální verze platformy Android, pro kterou je aplikace určena, zde verze devět. Číslo označuje API pro danou verzi Androidu. Devítka znamená verzi 2.3 (Gingerbread). Dalším důležitým elementem je element `uses-permission`, který označuje, jaká oprávnění aplikace vyžaduje. Mobilní klient vyžaduje tato oprávnění:

1. přístup k internetu - `android.permission.INTERNET`,
2. zápis do externího úložiště - `android.permission.WRITE_EXTERNAL_STORAGE`,
3. zjištění stavu sítě - `android.permission.ACCESS_NETWORK_STATE`.

5.2 Použité knihovny

Pro rychlejší a snadnější implementaci některých funkcí a zpětnou kompatibilitu aplikace na starších verzích Androidu, konkrétně verze 2.3 – 2.3.7, bylo využito knihoven třetích stran. V této kapitole nebudou popisovány knihovny, které jsou součástí **Android Open Source Project** [3]. Pro přehlednost jsou stránky, ze kterých jsou knihovny dostupné, umístěny v tab. C.1 v příloze C. Knihovny **ActionBar-Sherlock** a **Nine Old Androids** jsou od prosince 2014 deprecated a nebudou svými autory dále vyvíjeny a udržovány (při zahájení vývoje nebyly deprecated).

5.2.1 Knihovna logback-android

Knihovna `logback-android` je využívána pro logování a umožňuje pokročilejší nastavení logovacích zpráv oproti nativnímu logování pomocí třídy `android.util.Log` z Android API. Knihovna je upravenou verzí knihovny `logback`, která je vytvořena tvůrci populární knihovny `log4j`. Pro použití je potřeba stáhnout JAR soubory knihoven (`logback` a `logback-android`) a přidat je k projektu. V adresáři `/assets/` vytvořit soubor se jménem `logback.xml`, ve kterém je konfigurace logování (výstup logů, vzor logů atd.). Do třídy, ve které chceme použít logování, pak jednoduše vložíme kód 5.1 a logovací zprávu pak zapíšeme pomocí kódu 5.2.

```
static private final Logger logger =  
    LoggerFactory.getLogger(Trida.class);
```

Ukázka kódu 5.1: Vytvoření statické proměnné `logger` pro logování

```
logger.debug("Obsah logovací zpravy");
```

Ukázka kódu 5.2: Zapsání logovací zprávy

5.2.2 Knihovna ActionBarSherlock

Knihovna je rozšířením `Support Library` z [3], ale není její součástí. Knihovna je určena k tomu, aby bylo možné v aplikaci použít komponentu `Action Bar` již od verze 2.x a výše, protože `Action Bar` byl přidán až ve verzi 3.0. Knihovna automaticky použije nativní implementaci `Action Baru` na zařízeních s verzí 4.0 nebo vyšší. Pro předešlé verze, které neobsahují `Action Bar` použije vlastní implementaci založenou na zdrojových souborech verze 4.x. Knihovna je využita ve všech aktivitách aplikace. Implementace aktivity s touto knihovnou pak probíhá tak, že aktivita dědí od třídy `SherlockActivity` nebo `SherlockFragmentActivity`. Pro získání instance `ActionBar` se použije metoda `getSupportActionBar()`. S touto instancí je pak možné provádět úpravy a nastavení `Action Baru`.

5.2.3 Knihovna Nine Old Androids

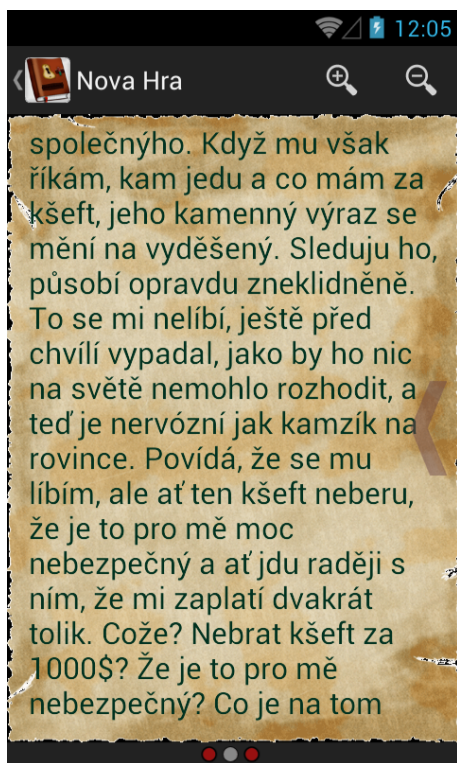
Tato knihovna umožňuje použít nové API pro animace zavedené ve verzi 3.0, na všech verzích platformy Android, tzn. od verze 1.0. V aplikaci je využita třídou `LoadingWaveAnimation`, která vytváří animaci při spuštění aplikace.

5.2.4 Knihovna ViewPagerIndicator

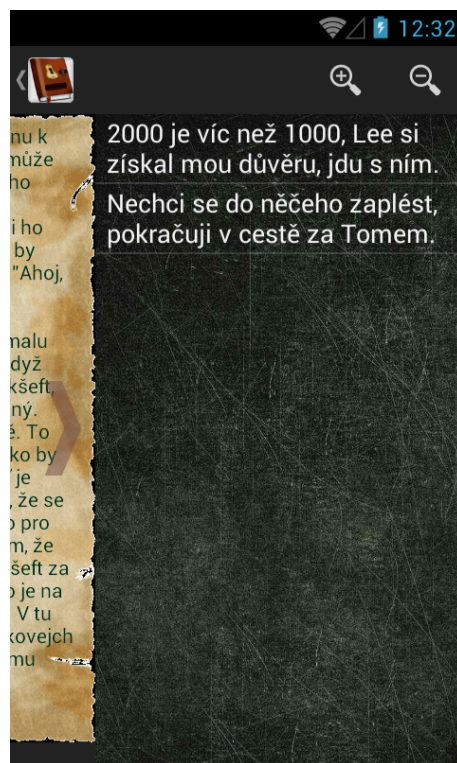
Knihovna se používá společně s třídou (layoutem) `ViewPager`, která umožňuje uživateli pohybovat se mezi více obrazovkami v jedné aktivitě. `ViewPagerIndicator` zobrazuje, na jaké obrazovce se uživatel nachází. V aplikaci je této vlastnosti využito v aktivitě `GameActivity`, kde indikuje aktuální stránku a počet stránek, viz obr. 5.1.

5.2.5 Knihovna SlidingMenu

Knihovna umožňuje snadnou implementaci vysouvací nabídky z okraje obrazovky, která je v aplikaci použita a obsahuje nabídku s akcemi, mezi kterými se uživatel při procházení gamebooku rozhoduje, viz obr. 5.2.



Obrázek 5.1: Ukázka herní obrazovky s použitím ViewPagerIndicator



Obrázek 5.2: Ukázka nabídky akcí implementované pomocí SlidingMenu

V aplikaci dědí aktivita `GameActivity` od třídy `SlidingFragmentActivity`. Při vytváření aktivity je v metodě `onCreate()` zavolána metoda `initSlidingMenu()` a je získána reference na instanci `SlidingMenu`, která reprezentuje nabídku. V `initSlidingMenu()` je nabídce nastaven layout `decision_fragment_view.xml` a jsou provedena další nastavení. Pokud je zařízení v `landscape` orientaci a velikost obrazovky je `large`, není vysouvací nabídka použita a nabídka je nahrazena klasickým fragmentem.

5.3 Implementace SQLite databáze

Pro uložení gamebooků je v aplikaci implementovaná SQLite databáze. Do databáze se ukládají i informace o rozehraných hrách. Struktura databáze je implementována podle návrhu ER diagramu z kap. 4.2.4, viz obr. 4.4. Databáze je vytvářena při prvním spuštění aplikace a manipulace s daty databáze se provádí pomocí potomků abstraktní třídy `DBQueries`.

5.3.1 Vytvoření databáze

Databáze je vytvářena při prvním spuštění aplikace pomocí kódu 5.3 v metodě `onCreate()` třídy `GameBookApplication`, která dědí od třídy `Application`. Tato třída je inicializována při vytváření procesu aplikace.

```
//získání třídy, která slouží pro komunikaci s databází
GameBookDataBaseHelper ghp = GameBookDataBaseHelper.
    getInstance(this.getApplicationContext());

//získání reference na databázi čímž dojde k jejímu vytvoření
ghp.getWritableDatabase();
```

Ukázka kódu 5.3: Vytvoření databáze ve třídě `GameBookApplication`

Třída `GameBookDataBaseHelper` dědí od třídy `SQLiteOpenHelper` a je implementována podle návrhového vzoru jedináček (Singleton). Návrhový vzor jedináček byl použit pro usnadnění správy instance databáze, protože v počátku vývoje docházelo při přístupu z více instancí k problémům s databází, např. souběžným přístupem z různých vláken. Třída přepisuje metodu `onCreate()`, která je automaticky volána při vytváření databáze (když neexistuje). V metodě jsou provedeny jednotlivé DDL příkazy jazyka SQL vytvářející tabulky. Třída obsahuje veškeré řetězce potřebné k sestavení příkazu pro vytvoření databáze. Třídou využívají především třídy, které dědí od `DBQueries` a manipulují s daty databáze.

5.3.2 Přístup k datům

Přístup k datům je implementován třídami, které dědí od abstraktní třídy `DBQueries`, jedná se o: `DBSelectQueries`, `DBInsertQueries`, `DBUpdateQueries` a `DBDeleteQueries`. Třídy obsahují `protected` proměnou třídy `GameBookDataBaseHelper`, pomocí níž otevřou databázi a získají instanci databáze. S touto instancí a voláním metody `query()` provádějí jednotlivé dotazy do databáze.

5.4 Uživatelské funkce

Mobilní klient obsahuje funkcionalitu popsanou v kap. 3.3, tj. procházení gamebooku, stažení gamebooku a některé další funkce. V této kapitole bude popsána implementace těchto funkcí.

5.4.1 Import gamebooků

Import gamebooků do databáze implementuje třída `BookImporter`. Import gamebooku je rozdělen na tyto dvě části:

1. zpracování souboru typu XML parserem,
2. zapsání zpracovaného gamebooku do databáze.

Parsování (zpracování) XML souboru s gamebookem zajišťuje metoda `nactiZeSouboruV2_02()` třídy `libFiles` ze sdílené knihovny, která vrací instanci třídy `Graph<Stav>`. V této instanci je obsažen celý gamebook, reprezentovaný jednotlivými třídami ze sdílené knihovny. Parser je implementován jako DOM parser. DOM (Document Object Model) parser nejprve načte celý XML soubor do paměti a vytvoří z něj stromovou strukturu, ke které se pak přistupuje. Stejný parser využívá i editor (program pro tvorbu gamebooků). V počátku vývoje byl pro mobilního klienta využíván samostatný SAX parser, ale při každé změně struktury XML souboru s gamebookem musely být upraveny oba parsery. I přesto, že je DOM parser paměťově náročnější, na výkonu aplikace se to neprojevalo. SAX (Simple API for XML) parser zpracovává XML dokument proudově, tzn. že není možné se vracet k již zpracovaným částem souboru a v paměti je uložena pouze část XML souboru. V projektu je stále uchováována původní verze SAX parseru (třída `XMLPullParser`) pro případ, že by v budoucnu z důvodu výkonnosti bylo potřeba tento parser znovu použít.

Zapsání gamebooku do databáze je následně provedeno ve třídě `DBInsertQueries` zavoláním metody `writeWholeBook()`. Zápis gamebooku je proveden jako jedna transakce, tzn. pokud by došlo k chybě při zápisu nějaké části, gamebook nebude zapsán, tím je zajištěno, že v databázi bude vždy celý gamebook.

5.4.2 Procházení gamebooku

Procházení gamebooku je umožněno uživateli v aktivitě `GameActivity`. V metodě `onCreate()` je při vytváření aktivity provedeno načtení rozehrané hry z databáze a sdílených preferencí, případně je spuštěna aktivita `GameInitActivity`, ve které uživatel provede základní nastavení hry.

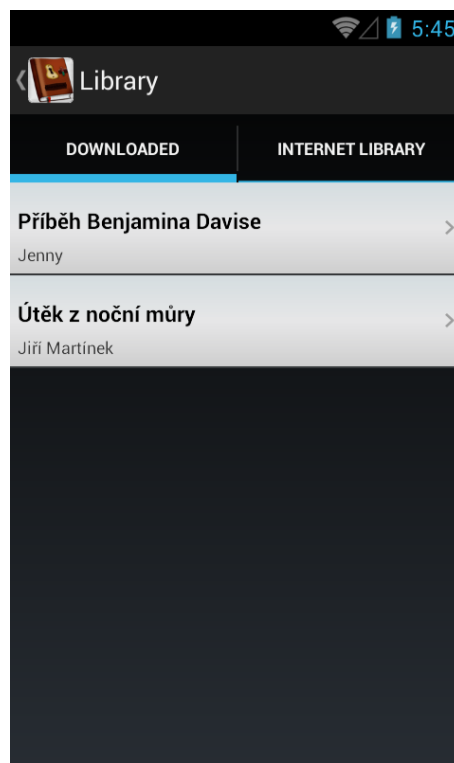
Vyhodnocování akcí, které si uživatel vybírá z vysouvací nabídky, viz obr. 5.2, probíhá ve třídě `Simulator` ze sdílené knihovny. Při vytváření objektu simulátoru je předáno id gamebooku, aktuální id stavu, instance hráče (třída `Postava`) a dvě rozhraní `IGraphDataAccessible` a `IActionDisplayable`. `IGraphDataAccessible` obsahuje metody, které vracejí data o gamebooku. Metody druhého rozhraní jsou

volány po vyhodnocení vybrané akce a mají uživateli zobrazit výsledek akce (např. zobrazení nového textu), toto rozhraní implementuje `GameActivity`. `IGraphData-Accessible` implementuje třída `DBSelectQueries`. Protože třída simulátoru je využívána také v editoru, jsou použita rozhraní. Rozhraní umožňují přistupovat k datům a zobrazovat prohlížení gamebooku nezávisle na aplikaci, tzn. umožňují použití třídy simulátoru v mobilním klientovi i v editoru.

Po kliknutí uživatele na vybranou akci ze seznamu, který je implementován fragmentem `DecisionListFragment`, je zavolána metoda `executeAction()` instance simulátoru. Zde je akce vyhodnocena a zavolána jedna z metod rozhraní `IAction-Displayable`, kterou implementuje `GameActivity`. V případě typu akcí `AkceNahoda` a `AkceDovednost`, je vrácena i instance třídy `AkceVyhodnoceni`, která obsahuje detailnější informace o výsledku. Tyto informace jsou pak použity a uživateli jsou zobrazeny kostky (viz obr. 5.3), které představují náhodu. Stiskem a držetím obrázku kostek se spustí animace hodu kostkami. Po skončení animace je uživateli zobrazen výsledek hodu, který byl spočítán z předaných informací. Při vyhodnocení akce `AkceBoj` je uživateli pouze zobrazen dialog s výsledkem boje.



Obrázek 5.3: Ukázka zobrazení hodu kostkou



Obrázek 5.4: Ukázka obrazovky v knihovně

5.4.3 Knihovna

V knihovně jsou uživateli zobrazeny gamebooky dostupné ke stažení i gamebooky, které již má v zařízení stažené, viz obr. 5.4. V aktivitě `LibraryActivity` jsou obsaženy dvě záložky se seznamy gamebooků, tj. dostupných ke stažení a již stažených. Po kliknutí na daný gamebook je zobrazen jeho detail fragmentem `DetailBookFragment`, kde je možné tento gamebook stáhnout, smazat nebo vybrat pro hraní v závislosti na tom, z jaké záložky byl gamebook vybrán. Na zařízení s velkou obrazovkou je detail gamebooku zobrazen vedle seznamu. Na zařízení s malou obrazovkou a v režimu `portrait` je spuštěna nová aktivita `DetailBookActivity`.

5.5 Sít'ová komunikace

Aplikace využívá pro sít'ovou komunikaci knihovnu `Volley` a pro stahování gamebooku systémovou službu `DownloadManager`. Aplikace provádí sít'ové operace popsané v kapitole 4.1. Dále bude popsána jejich implementace.

5.5.1 Implementace knihovny `Volley`

Pro využívání knihovny `Volley` byla implementována třída `VolleySingleton` podle návrhového vzoru jedináček (Singleton) a třída `RequestFactory`. Třída `RequestFactory` obsahuje metody pro vytváření požadavků na sít'ové operace (seznam a hodnocení gamebooků) reprezentované instancemi třídy `StringRequest`. Při vytváření požadavku jsou předávány instance tříd `Response.Listener` (úspěch) a `Response.ErrorListener` (neúspěch), jejichž metody jsou po dokončení zavolány v závislosti na výsledku dotazu. Třída `VolleySingleton` přidává vytvořené požadavky do fronty, ze které jsou automaticky vybírány a asynchronně zpracovávány. Výběr a zpracování požadavku je prováděn knihovnou `Volley` a programátor ho nemusí řešit.

5.5.2 Získání seznamu dostupných gamebooků

Získání seznamu dostupných gamebooků využívá fragment `InternetListFragment`, který tento seznam také zobrazí. Fragment dědí od abstraktní třídy `LibraryListFragment` a implementuje metodu `fillList()`, ve které probíhá zaslání požadavku na server o seznam dostupných gamebooků.

Nejprve je zkontrolována dostupnost sítě, poté je třídou `URLFactory` vygenerována URL adresa, kde je dostupný seznam gamebooků. URL (Uniform Resource Locator) je řetězec, který jednoznačně určuje umístění (adresu) zdroje dostupného na internetu. Následně se zjistí, zda požadavek na danou URL adresu není uložen v paměti, viz 5.4. Pokud ano, je zpracován a zobrazen seznam, jinak je vytvořen požadavek, který je přidán do fronty třídou `VolleySingleton` a následně zpracován.

```
// Vygenerování url pro seznam
String url = URLFactory.generateURL4bookReqMalPal();
// Získání cache paměti
Cache cache = VolleySingleton.getInstance(getActivity()).
    getRequestQueue().getCache();
// Získání záznamu pro danou url adresu
Entry entry = cache.get(url);
if(entry != null){
    // Zpracování záznamu z cache paměti
```

Ukázka kódu 5.4: Vytvoření databáze ve třídě `GameBookApplication`

Při přidávání požadavku do fronty jsou také předány reference na objekty tříd, ve kterých je zpracována přijatá odpověď a při úspěšném přijetí zprávy je zobrazen seznam gamebooků, které je možné stáhnout. Přijatý seznam je odfiltrován a jsou zobrazeny pouze gamebooky, které ještě nejsou staženy.

5.5.3 Stažení gamebooku

Stažení gamebooku je implementováno ve fragmentu `DetailBookFragment` a je implementováno s využitím systémové služby `DownloadManager`. Stažení probíhá na stejném principu jako u knihovny `Volley`, nejdříve je vytvořena adresa pro stažení a požadavek, viz 5.5. Dále je požadavku nastaveno kam má být soubor s gamebookem uložen a nakonec je požadavek zařazen do fronty pro stažení (začne stahování). Při získávání instance `DownloadManageru` je registrován `BroadcastReceiver`, který reaguje na zprávy zaslané `DownloadManagerem`. Zpracováním těchto zpráv fragment reaguje na výsledek dokončení stahování, ať již úspěšný nebo neúspěšný. Po úspěšném dokončení je uložený gamebook importován do databáze a soubor smazán.

Pokud uživatel během stahování opustí `DetailBookFragment`, je `BroadcastReceiver` odregistrován a není tedy možné reagovat na zprávy od `DownloadManageru`, ale stahování pokračuje automaticky dále. Po dokončení stahování není gamebook, v tomto případě automaticky importován do databáze, ale je jen stažen a uložen. Pro import je potřeba, aby se uživatel znovu dostal do fragmentu s daným gamebookem. Při vytváření fragmentu je kontrolováno, zda není soubor s gamebookem

již stažen. Pokud je soubor už stažený, potom je gamebook automaticky importován do databáze.

```
// Vytvoření adresy podle id a jazyka požadované knihy
Uri uri = Uri.parse(URLFactory.generateURL4DownloadBook(book.id,
    book.language));
// Vytvoření požadavku
DownloadManager.Request request = new DownloadManager.Request(uri);
```

Ukázka kódu 5.5: Vytvoření požadavku pro stažení gamebooku

5.5.4 Hodnocení gamebooku

Hodnocení gamebooku probíhá ve fragmentu `DetailBookFragment`. Odeslání hodnocení se provádí v metodě `onRating()`. Odeslání probíhá podobně jako stažení seznamu dostupných gamebooků. Nejprve je vygenerována URL adresa na základě id hodnoceného gamebooku a vytvořen požadavek (instance třídy `StringRequest`). Následně je požadavek opět přidán do fronty požadavků třídou `VolleySingleton`, ze které je vybrán a zpracován.

5.6 Nastavení aplikace

Aplikace umožňuje uživateli provést tato nastavení:

- používat kostku při procházení gamebooku, tzn. nechat uživatele házet kostkou (zobrazit animaci), při vypnutí není uživateli animace zobrazena a je přímo poslán do příslušného stavu,
- změnit font písma,
- nastavit základní velikost písma při startu nové hry,
- možnost zobrazení šipky (tlačítka) na herní obrazovce pro vysunutí menu s nabídkou.

Implementace nastavení je obsažena v aktivitě `SettingsActivity`, která dědí od třídy `SherlockPreferenceActivity`. Z důvodu kompatibility pro zařízení s verzí nižší než 3.0 je UI s nastaveními vkládáno přímo do aktivity. Pro zařízení od verze 3.0 je použit fragment, jenž je reprezentován statickou vnitřní třídou `PrefsFragment`, která dědí od třídy `PreferenceFragment`. Implementace nastavení není díky

Android API složitá. Je třeba vytvořit XML soubor ve adresáři `/res/xml` s definicí dostupných nastavení. Následně v aktivitě nebo fragmentu, který dědí od příslušného rodiče a umožňuje implementaci nastavení, přidat tento XML soubor voláním metody `addPreferencesFromResource()`. Systém automaticky zaznamenává vybrané položky uživatelem a ukládá je. Každá položka definovaná v XML souboru má svůj unikátní klíč, atribut `android:key`, pod tímto klíčem je následně nastavená hodnota přístupná kdekoliv v aplikaci, viz 5.6. Klíče jsou uloženy ve třídě `PrefTags`. Pro výběr základní velikosti fontu bylo potřeba implementovat třídu `NumberPickerPrefDialog`, která umožňuje vybrat číslo.

```
// Získání instance pro přístup k nastavení
SharedPreferences settingsPref =
    PreferenceManager.getDefaultSharedPreferences(this);
/*Získání konkrétní hodnoty pod klíčem (String) prefKey, pokud
   hodnota není nalezena bude vrácena hodnota defValue */
int value = settingsPref.getInt(prefKey, defValue);
```

Ukázka kódu 5.6: Získání hodnoty z nastavení aplikace

6 Testování

Pro ověření funkcionality vytvořené aplikace bylo provedeno testování na vybraných mobilních telefonech a tabletu. Aplikace byla testována na zařízeních, která jsou uvedena v tab. 6.1. V tabulce je vždy uveden název zařízení, verze Androidu a rozlišení jeho obrazovky. Testování probíhalo na reálných i na virtualizovaných zařízeních. Virtualizovaným zařízením je myšlen emulátor zařízení, který emuluje dané zařízení s operačním systémem Android. Testování bylo prováděno autorem práce, ale i uživateli, kteří se s aplikací setkali poprvé.

Zařízení	Verze	Rozlišení obrazovky v px
Samsung S6500 Galaxy Mini II	2.3.6	480 x 320
HTC One V	4.0.3	800 x 480
Sony Xperia M Dual	4.3	854 x 480
Custom Tablet (virtualizace)	5.0.0	2560 x 1600
Samsung Galaxy S2 (virtualizace)	4.1.1	800 x 480
LG P710	4.1.2	800 x 480
LG Optimus L3 II (virtualizace)	4.1.1	320 x 240
Samsung Nexus S	4.1.2	800 x 480
Samsung S5690 Galaxy Xcover	2.3.6	480 x 320

Tabulka 6.1: Zařízení, na kterých byla aplikace testována

6.1 Průběh testování

Aplikace byla při vývoji průběžně testována autorem. Pro zápis logů byla použita knihovna `logback-android`, která umožňovala pokročilejší nastavení logovacích zpráv (formát, výstup). Testeré (běžní uživatelé) dostali již hotovou aplikaci a měli za úkol aplikaci prozkoumat a vyzkoušet všechny funkce aplikace nebo alespoň ty nejdůležitější, mezi které patří:

- procházení gamebooku,
- změna velikosti písma,
- hod kostkami,
- přechod mezi jednotlivými obrazovkami textu,

- stažení nového gamebooku,
- vybrání a prohlížení staženého gamebooku,
- hodnocení gamebooku.

6.2 Testování nestandardních událostí

Dále bylo na vybraných zařízeních provedeno několik testů, ve kterých byla aplikace vystavena nestandardním situacím, tzn. takovým situacím, které mohou nastat nezávisle na aplikaci při jejím používání. Mezi tyto testované situace při používání aplikace patří:

- přijetí SMS (Short Message Service) zprávy,
- příchozí hovor,
- fungování aplikace při nedostupné síti,
- smazání lokálních dat aplikace a její nové spuštění.

6.3 Výsledky testování

Při testování bylo uživateli zjištěno, že při změně orientace obrazovky v seznamu gamebooků, které je možné stáhnout, docházelo k pádu aplikace z důvodu neinicializované reference na objekt a vyvolání výjimky `NullPointerException`. Problém byl vyřešen otestováním reference na hodnotu `null` před jeho použitím.

Druhým zjištěným problémem bylo nezobrazení seznamu stažitelných gamebooků při pomalém připojení k internetu. Rychlost internetu byla v zařízení omezena operátorem po vyčerpání limitu. I když připojení fungovalo, knihovna `Volley` nedokázala požadavek zpracovat a výsledek vyhodnotila jako nepodařený (zavolána metoda pro obsluhu výsledku, který skončil chybou). Uživateli pak byl zobrazen prázdný seznam. Tento problém se projevil pouze u jednoho testovaného zařízení, ale ne vždy, po více pokusech se seznam podařilo stáhnout. Takové chování aplikace není možné jednoduše ovlivnit, protože požadavek zpracovává knihovna. V případě neúspěchu je proto uživatel pouze informován, aby se pokusil znovu obnovit seznam pomocí akce v `Action Baru`. Jiné závažné problémy objeveny nebyly. Testeři měli pouze připomínky k UI, ale tyto připomínky lze považovat za subjektivní, protože se lišily. Lze

očekávat, že po dlouhodobějším používání aplikace (po její distribuci) se na základě reakcí uživatelů ukáže, které části UI by měly být optimalizovány.

Z výkonnostního hlediska nebyl testery nalezen žádný závažný problém. Aplikace fungovala bez problémů s plynulostí zobrazení. Pouze při testování aplikace na tabletu, který byl virtualizovaný, aplikace nefungovala zcela plynule, ale to bylo způsobeno velkými systémovými nároky na virtualizaci zařízení. Obdobné problémy mají na tomto zařízení i jiné aplikace.

6.3.1 Nestandardní události

Při testování nestandardních událostí nebyl nalezen žádný problém. Aplikace zvládala všechny situace uvedené v kapitole 6.2. Příchozí hovor je možné přijmout a po jeho skončení je aplikace vrácena do popředí. Aplikace nepotřebuje stálé připojení k síti, pouze při stahování gamebooku nebo seznamu dostupných gamebooků. Pokud uživatel chce zobrazit seznam stažitelných gamebooků a není dostupná síť, je o tom informován dialogem. Pokud dojde k výpadku spojení během stahování gamebooku, tak tento problém řeší sama systémová služba `DownloadManager`. Při smazání lokálních dat jsou smazány všechny stažené knihy a veškerý postup ve hře, ale po opětovném spuštění aplikace je znovu vytvořena databáze a naimportovány gamebooky distribuované s aplikací a aplikace může opět fungovat.

6.4 Možná rozšíření

Aplikace samozřejmě obsahuje části, které je možné dále rozšiřovat a vylepšovat. K implementované funkcionalitě by mohla být přidána i nová funkcionalita (a v budoucnu by i měla být přidána). Mezi možná rozšíření, která by mohla být implementována patří:

- filtrování gamebooků, které je možné stáhnout (podle jazyka, žánru atd.),
- ozvučení aplikace,
- přidání obrázků, které souvisí s příběhem gamebooku přímo do textu,
- propojení aplikace se sociálními sítěmi (facebook), např. možnost sdílet dosažené úspěchy ve hře,
- možnost volit různé vzhledy aplikace,
- implementace zobrazení průběhu boje.

7 Závěr

Gamebook je kniha, ve které je možné provádět rozhodnutí, jak se bude příběh dále vyvíjet a do jisté míry si tak příběh knihy může čtenář určovat sám. Po přečtení dané části textu knihy jsou nabídnuty možnosti, pro které je možné se rozhodnout. Zvolením jedné z těchto možností je čtenář odkázán na jinou část, která na předchozí navazuje a takto pokračuje až do konce.

Cílem této bakalářské práce bylo navrhnout, realizovat a ověřit funkcionalitu mobilního klienta pro platformu Android, který umožňuje procházet gamebooky vytvořené ve vyvíjeném systému. Tento cíl práce byl splněn. Při implementaci bylo potřeba spolupracovat (spoluvytvářet sdílenou knihovnu) s autorem programu pro tvorbu gamebooků (editoru).

V rámci práce jsem nejprve prozkoumal vybrané mobilní aplikace, které umožňují procházení gamebooků. Při porovnání existujících aplikací jsem nenalezl žádnou aplikaci, která by obsahovala gamebook v českém jazyce. Dále jsem popsal vyvíjený systém pro publikování gamebooků a navrhl mobilního klienta umožňujícího gamebooky vytvořené v tomto systému procházet. Navrženého mobilního klienta jsem následně implementoval na platformě Android a ověřil jeho funkcionalitu. Mobilní klient umožňuje procházení gamebooků a stažení nově vydaných gamebooků.

Mobilní klient bude dále rozšiřován a vyvíjen o funkcionalitu popsanou v kap. 6.4. Aplikace je připravena na publikování v obchodě **Google play**. Jakmile bude dokončen dostatek gamebooků, které budou spolu s aplikací publikovány, bude aplikace v tomto obchodě dostupná.

Aplikace byla otestována na vybrané množině uživatelů a představuje vhodný přechod od knižní podoby gamebooků k elektronické formě.

Seznam použitých zkratek a výrazů

Action Bar	Grafická komponenta (lišta) na platformě Android umístěná v horní části obrazovky (případně i dolní), která uživateli usnadňuje navigaci v aplikaci.
ADT	Android Development Tools - plugin pro Eclipse IDE umožňující vývoj pro platformu Android.
API	Application Programming Interface - rozhraní pro programování aplikací.
DDL	Data Definition Language - příkazy jazyka SQL pro vytváření a upravování struktury databáze.
DML	Data Manipulation Language - příkazy jazyka SQL pro manipulaci s daty databáze.
DOM	Document Object Model – objektový model dokumentu umožňující přístup k jednotlivým objektům dokumentu.
dp	Density-independent pixel - virtuální pixel obrazovky, který je nezávislý na hustotě pixelů dané obrazovky.
Framework	Nástroj (obecně) pro usnadnění a urychlení vývoje aplikace.
Google play	Služba primárně určená pro distribuci aplikací pro platformu Android, umožňuje ale distribuovat i jiný digitální obsah.
HTML	HyperText Markup Language - značkovací jazyk používaný k tvorbě webových stránek.
HTTP	HyperText Transfer Protocol - internetový protokol pro přenášení HTML dokumentů.
JDK	Java Development Kit - základní nástroje pro vývoj aplikací na platformu JAVA.
Landscape	Zobrazení obrazovky na šířku na platformě Android.
Non-volatile	Paměť, která dokáže uchovat data i bez napájení.

Open Source	Software s otevřeným zdrojovým kódem.
Portrait	Zobrazení obrazovky na výšku na platformě Android.
SAX	Simple API for XML - API umožňující proudové zpracování XML souboru.
SDK	Software development kit - množina nástrojů určená pro vývoj aplikací.
SMS	Short Message Service - služba umožňující posílání krátké (160 znakové) textové zprávy.
SQL	Structured Query Language - jazyk pro práci s relačními databázemi.
UI	User Interface - uživatelské rozhraní.
UML	Unified Modeling Language - jazyk pro vizuální modelování systémů.
URL	Uniform Resource Locator - řetězec jednoznačně určující umístění (adresu) zdroje dostupného na internetu.
XML	Extensible Markup Language - rozšiřitelný značkovací jazyk určený pro přenos dat.

Literatura

- [1] GOOGLE. *Android Developers* [online]. 2015 [cit. 2015-04-01]. Dostupné z: <http://developer.android.com/>
- [2] Strategy Analytics: Android Shipped 1 Billion Smartphones Worldwide in 2014. In: *PR Newswire* [online]. 2015 [cit. 2015-04-01]. Dostupné z: <http://www.prnewswire.com/news-releases/strategy-analytics-android-shipped-1-billion-smartphones-worldwide-in-2014-300027707.html>
- [3] GOOGLE. *Android Open Source Project* [online]. 2015 [cit. 2015-04-03]. Dostupné z: <https://source.android.com/>
- [4] ČADA, Roman, Tomáš KAISER a Zdeněk RYJÁČEK. *Diskrétní matematika*. Plzeň: Západočeská univerzita, 204. ISBN 80-7082-939-7
- [5] CHOICE OF GAMES LLC. *Choice of Games* [online]. 2015 [cit. 2015-04-03]. Dostupné z: <https://www.choiceofgames.com/>
- [6] SOSINSKY, Barrie A. *Mistrovství – počítačové sítě*. Vyd. 1. Brno: Computer Press, 2010, 840 s. Mistrovství (Computer Press). ISBN 978-80-251-3363-7.
- [7] ALLEN, Grant. *Android 4: průvodce programováním mobilních aplikací*. 1. vyd. Překlad Jakub Mužík. Brno: Computer Press, 2013, 656 s. ISBN 978-80-251-3782-6.
- [8] MURPHY, Mark L. *Android 2: průvodce programováním mobilních aplikací*. Vyd. 1. Brno: Computer Press, 2011, 375 s. ISBN 978-80-251-3194-7.
- [9] TUTORIALS POINT. *Android Programming Tutorial* [online]. 2014 [cit. 2015-04-04]. Dostupné z: <http://www.tutorialspoint.com/about/index.htm>
- [10] VOGELLA COMPANY. *Android Development* [online]. 2015 [cit. 2015-04-04]. Dostupné z: <http://www.vogella.com/tutorials/android.html>
- [11] STACK EXCHANGE INC. *Stackoverflow* [online]. 2015 [cit. 2015-04-04]. Dostupné z: <http://stackoverflow.com/>

-
- [12] Transmitting Network Data Using Volley. GOOGLE. *Android Developers* [online]. 2015 [cit. 2015-04-18]. Dostupné z: <https://developer.android.com/training/volley/index.html>
- [13] Dashboards. GOOGLE. *Android Developers* [online]. 2015 [cit. 2015-04-18]. Dostupné z: https://developer.android.com/about/dashboards/index.html?utm_source=suzunone
- [14] DownloadManager. GOOGLE. *Android Developers* [online]. 2015 [cit. 2015-04-19]. Dostupné z: <http://developer.android.com/reference/android/app/DownloadManager.html>
- [15] Saving Data. GOOGLE. *Android Developers* [online]. 2015 [cit. 2015-04-19]. Dostupné z: <http://developer.android.com/training/basics/data-storage/index.html>
- [16] ARLOW, Jim a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. 2. akt. a dopl. vyd. Překlad Bogdan Kiszka. Brno: Computer Press, 2007, 567 s. ISBN 978-80-251-1503-9.
- [17] HANA, Kanisová a Müller MIROSLAV. *UML srozumitelně*. Vyd. 1. Brno: Computer Press, 2004, 157 s. ISBN 80-251-0231-9.
- [18] Fragments. GOOGLE. *Android Developers* [online]. 2015 [cit. 2015-04-19]. Dostupné z: <http://developer.android.com/guide/components/fragments.html>
- [19] App Manifest. GOOGLE. *Android Developers* [online]. 2015 [cit. 2015-04-19]. Dostupné z: <http://developer.android.com/guide/topics/manifest/manifest-intro.html>

Seznam příloh

Uživatelská příručka - Příloha A

Obsah CD - Příloha B

Použité knihovny - Příloha C

A Uživatelská příručka

Uživatelská příručka obsahuje základní informace o aplikaci, tj. co aplikace umožňuje (její funkce), popis její instalace a ovládání.

A.1 Požadavky aplikace

Pro správné fungování aplikace je potřeba, aby zařízení, na které se bude aplikace instalovat, obsahovala minimálně verzi 2.3 (Gingerbread) operačního systému Android a externí SD kartu. Při instalaci jsou vyžadována tato oprávnění:

1. přístup k internetu,
2. zápis do externího úložiště,
3. zjištění stavu sítě.

A.2 Instalace

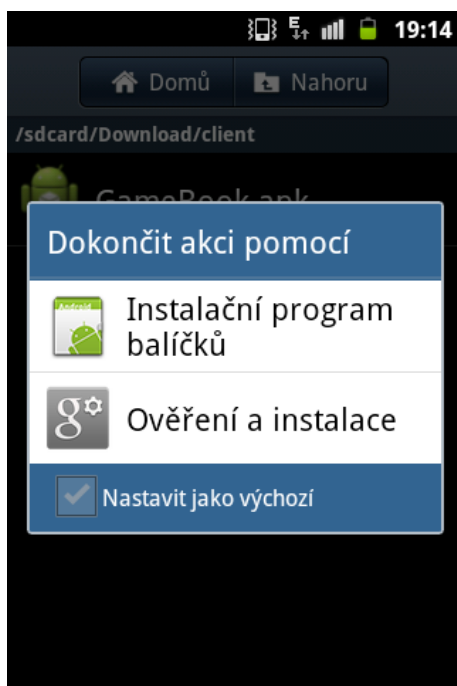
K nainstalování aplikace je potřeba zkopírovat přiložený soubor **GameBook.apk** do zařízení (na externí SD kartu nebo na vnitřní paměť). Po zkopírování naleznete pomocí souborového manažera soubor v zařízení.

Před instalací je potřeba na některých verzích Androidu v nastavení povolit instalaci z neznámých zdrojů. Pro povolení jděte do / *Nastavení* / *Zabezpečení* / *Neznámé zdroje* a zaškrtněte položku *Neznámé zdroje*, viz obr. A.2.

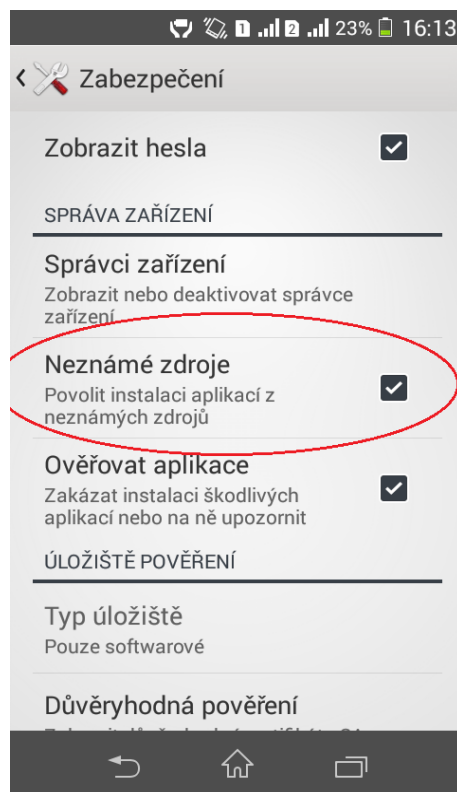
Soubor vyberte stiskem. Pokud se zobrazí nabídka *Dokončit akci pomocí*, viz obr. A.1, pak stiskněte volbu *Instalační program balíčků*. Dále se objeví nabídka, zda chcete tuto aplikaci nainstalovat, viz obr. A.3. Stiskněte tlačítko *Instalovat* a aplikace se začne instalovat. Po úspěšném nainstalování je možné aplikaci spustit

A.3 Ovládání aplikace

Po spuštění aplikace je zobrazena obrazovka s animací, viz obr. A.4, na této obrazovce by uživatel měl vyčkat než se spustí obrazovka s menu, viz obr. A.5. Obrazovka



Obrázek A.1: Ukázka z instalace aplikace

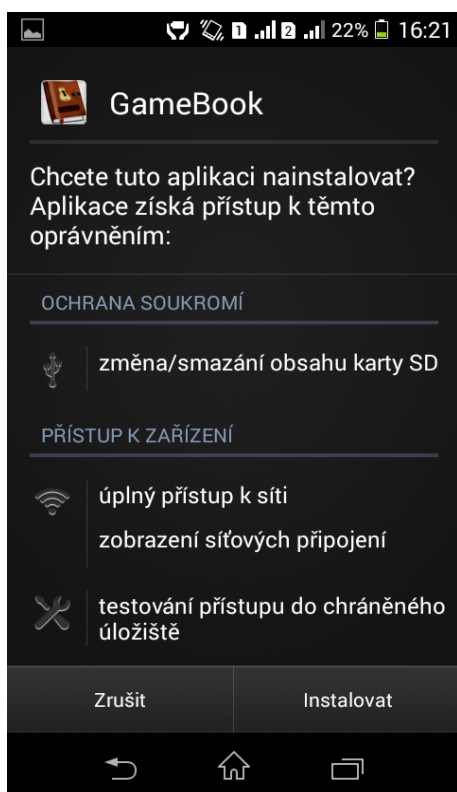


Obrázek A.2: Povolení instalace z neznámých zdrojů

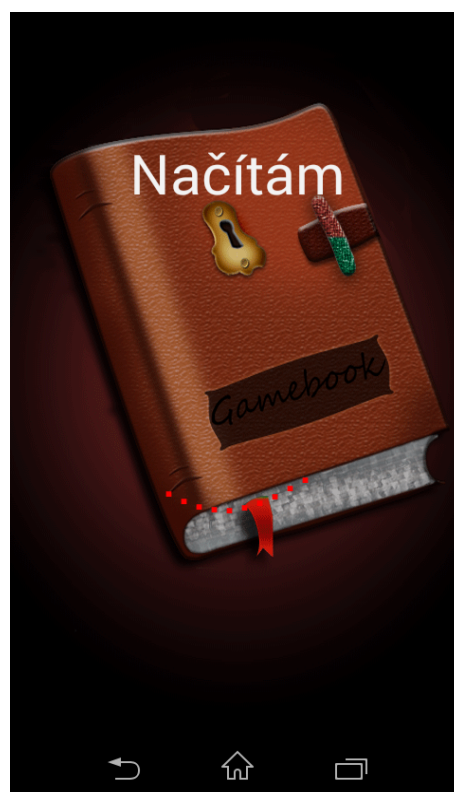
s animací se nemusí objevit vždy, zpravidla pouze při prvním spuštění aplikace.

V menu jsou umístěna tlačítka *Nová Hra*, *Pokračovat* a *Knihovna*. Kliknutím na tlačítko *Nová Hra*, se spustí procházení aktuálně vybraného gamebooku. Jméno vybraného gamebooku je zobrazeno v levém dolním rohu obrazovky. V případě, že není vybrán žádný gamebook je spuštěn gamebook distribuovaný s aplikací. Tlačítko *Pokračovat* je možné stisknout pokud je nějaký gamebook už procházen. Pokud se klikne na novou hru, je zobrazen dialog, zda má být skutečně spuštěna nová hra a přepsat původní. V případě, že hra není rozehraná, dialog se nezobrazí. Kliknutím na tlačítko *Knihovna* se spustí knihovna s gamebooky, které jsou v zařízení již staženy a s gamebooky, které je možné stáhnout.

V menu je umístěn také **Action Bar** (lišta v horní části obrazovky), ve které jsou umístěny akce pro zobrazení informací o programu, nastavení aplikace a ukončení aplikace. Rozložení se může na různých verzích platformy Android lišit, tzn. že na nižších verzích Androidu (verze nižší než 3.0) budou nějaké akce umístěny v kontextové nabídce vyvolané hardwarovým tlačítkem, viz obr. A.6.



Obrázek A.3: Ukázka z instalace aplikace



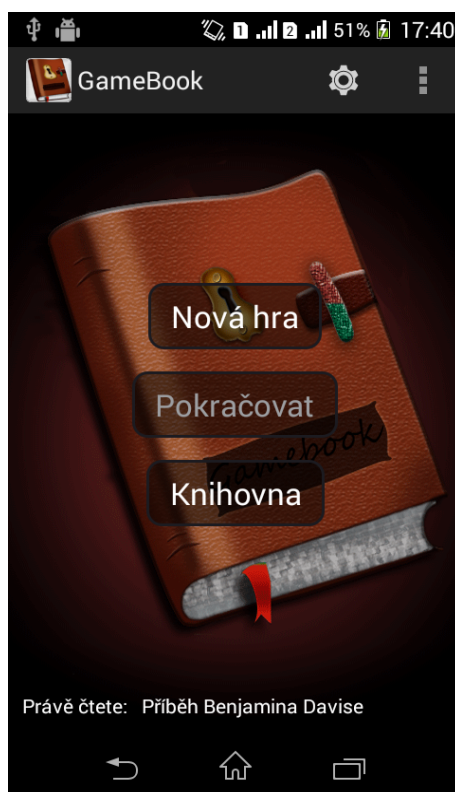
Obrázek A.4: Ukázka úvodní obrazovky po spuštění

Po kliknutí na akci *O Programu* (ikona s písmenem *i*) je zobrazen popis aplikace a odkaz na webové stránky celého systému.

A.3.1 Procházení gamebooku

K procházení gamebooku je nejprve třeba vybrat gamebook z knihovny a kliknout na tlačítko *Nová hra* v hlavním menu aplikace. Pokud nebude gamebook vybrán z knihovny bude použit gamebook distribuovaný s aplikací. Po kliknutí na novou hru bude uživateli zobrazena obrazovka, viz obr. A.7, na které provede základní nastavení hry, tj. jméno a případně si vybere vlastnosti postavy, které pak ovlivňují příběh. Po stisknutí tlačítka *Začít hru* se zobrazí úvodní stav gamebooku, viz obr. A.8. Pokud text obsahuje více stránek je možné táhnutím prstu přecházet mezi nimi. V *Action Baru* je možné kliknout na akce pro zmenšení nebo zvětšení textu a také restartovat aktuální hru.

U velkých obrazovek a v režimu obrazovky na šířku je vedle textu zobrazena nabídka s možnostmi, které je možné vykonat, viz obr. A.10. Na ostatních konfigu-



Obrázek A.5: Ukázka hlavního menu

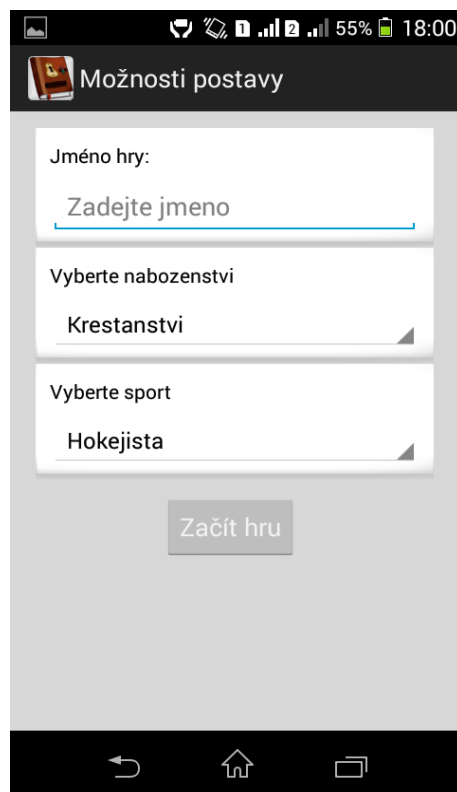
racích obrazovky se nabídka zobrazí kliknutím na šipku u pravého okraje obrazovky nebo táhnutím prstu od pravého okraje obrazovky. Pro pokračování je třeba kliknout na jednu z možností. Po kliknutí dojde k vyhodnocení a přesunu do nového stavu. V případě boje je zobrazen dialog s výsledkem boje (grafický boj zatím není možné procházet). U některých akcí je uživatel vyzván, aby hodil kostkou, viz obr. A.9. Podle výsledku se odvíjí další postup ve hře, poprvé může uživatel hodit v součtu např. 5 a bude nasměrován do stavu A. Pokud by stejný gamebook hrál podruhé a hodil vyšší číslo než 5 může být nasměrován do jiného stavu příběhu. Stav se mohou prolínat a touto vlastností je zajištěno, že pokud bude daný gamebook hrán dvakrát, nemusí se hra vždy vyvíjet stejně.

A.3.2 Knihovna

V knihovně jsou zobrazeny dvě záložky *Stažené* a *Ke stažení*, viz obr. A.11. Po kliknutí na vybraný gamebook je zobrazen detail gamebooku. V zařízeních s velkou obrazovkou a pokud je obrazovka zobrazena na šířku, je detail zobrazen vedle seznamu.



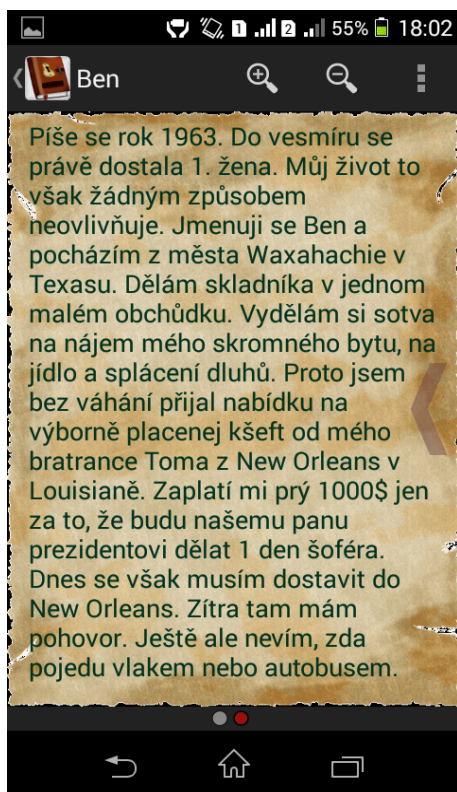
Obrázek A.6: Ukázka menu na zařízení s verzí 2.3



Obrázek A.7: Ukázka obrazovky s počátečním nastavením gamebooku

V detailu pro stažené gamebooky je možné vidět obecné informace o gamebooku. Dále je zde položka s hodnocením gamebooku, na kterou je možné kliknout a gamebook ohodnotit. Pod touto položkou jsou tlačítka pro smazání gamebooku a výběr gamebooku. Pokud uživatel klikne na tlačítko *Vybrat*, bude zobrazeno menu aplikace a vybraný gamebook bude možné hrát. Hry jsou automaticky ukládány. Pokud je rozehrán např. gamebook *A*, a je vybrán gamebook *B*, postup z gamebooku *A* je uložen. Po opětovném vybrání gamebooku *A* je možné pokračovat v původní hře, ale je možné mít rozehranou pouze jednu hru od každého gamebooku. Při odstranění gamebooku tlačítkem *Smazat* jsou odstraněny i uložené hry.

V detailu pro gamebooky, které je možné stáhnout, viz obr. A.12, jsou také zobrazeny obecné informace o gamebooku a položka s hodnocením. Zde je hodnocení daného gamebooku pouze zobrazeno a není možné ho hodnotit, to lze až po jeho stažení. V tomto detailu je také umístěno tlačítko pro stažení gamebooku *Stáhnout*, po kliknutí dojde k zahájení stahování. Pokud by během stahování uživatel opustil obrazovku s detailem gamebooku, pak je potřeba po dokončení stahování se na obrazovku vrátit, aby mohl být stažený gamebook rozbalen. Následně bude gamebook zobrazen v záložce stažených



Obrázek A.8: Ukázka herní obrazovky

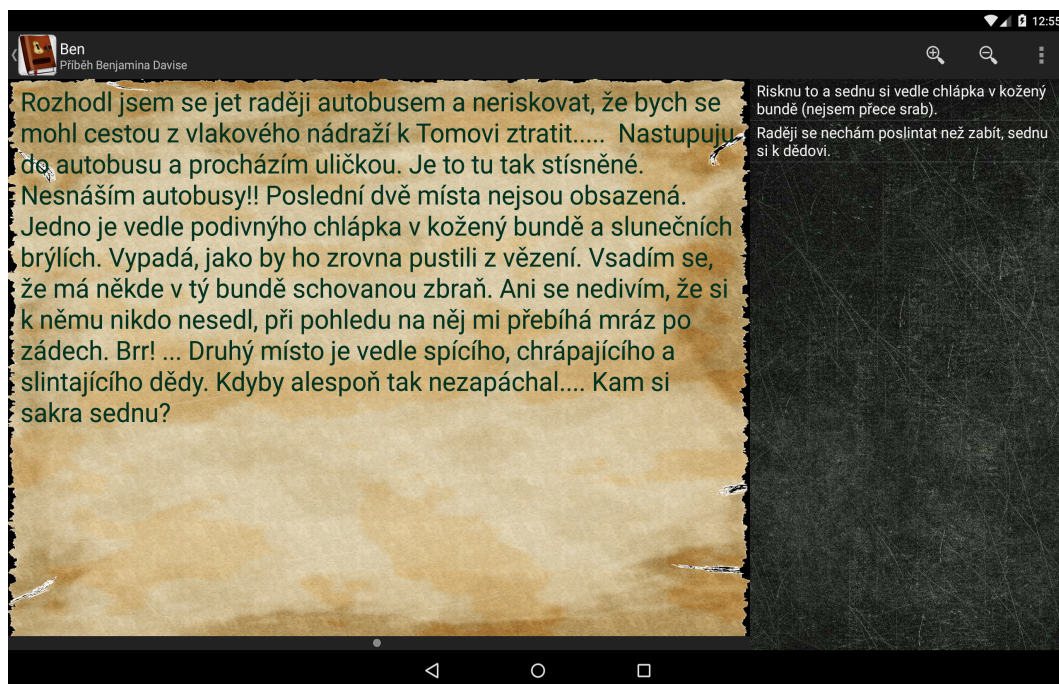


Obrázek A.9: Ukázka dialogu s kostkami

A.3.3 Nastavení aplikace

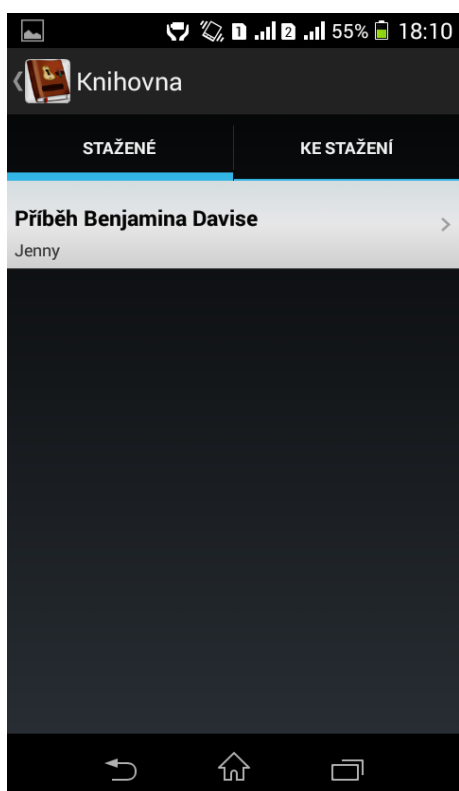
Do nastavení aplikace, viz obr. A.13, je možné dostat se z menu aplikace kliknutím na akci *Nastavení* v *Action Baru* (ikona ozubeného kolečka). V nastavení jsou dvě oddělené sekce, a to *Herní Nastavení* a *Aplikace a Vývojář*.

V sekci *Herní Nastavení* je možné nastavit některá nastavení pro procházení hry. Položka *Používat kostky* označuje, zda mají být ve hře zobrazovány dialogy s kostkou, viz obr. A.9 (je možné hodit kostkami u určitých akcí). Pokud je položka vypnuta, není zobrazen dialog s kostkami a daná akce se chová jako ostatní rozhodnutí (uživatel je poslán do příslušného stavu bez možnosti hodu kostkou). Druhá položka zobrazuje nabídku pro výběr typu písma, který bude používán pro text při procházení gamebooku. Další položka zobrazí dialog, viz obr. A.14, ve kterém je možné vybrat velikost písma, které bude použito při spuštění nové hry. Minimální velikost je 10 a maximální je 50. Poslední položka umožňuje nastavit, zda má být zobrazováno tlačítko se šipkou v herní obrazovce pro vysouvání nabídky s možnými rozhodnutími.

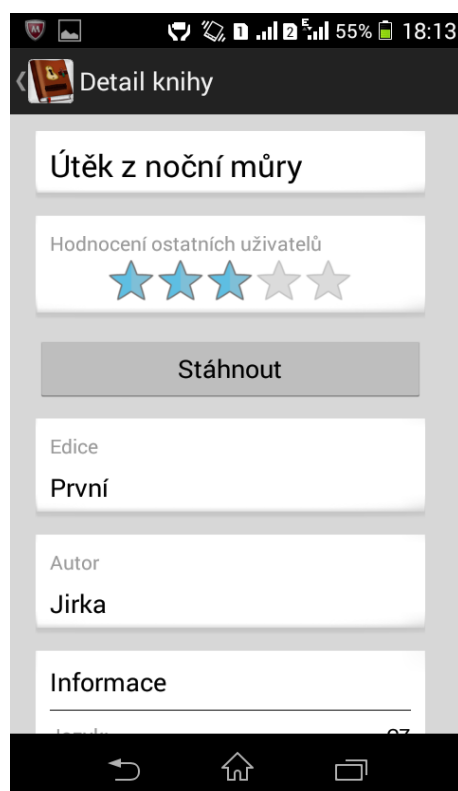


Obrázek A.10: Ukázka rozložení komponent ve hře při orientaci na šířku u velké obrazovky

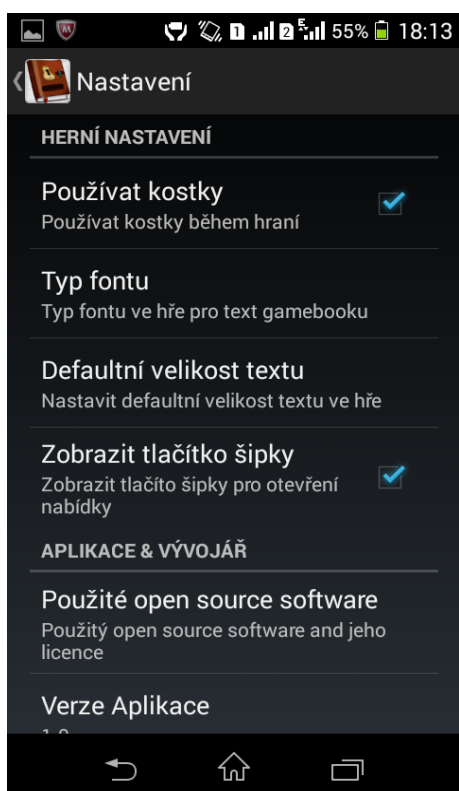
Ve druhé sekci nastavení *Aplikace a Vývojář* nejsou žádná funkční nastavení pouze informace o verzi aplikace, seznam použitých open source knihoven a softwaru (zobrazí se po kliknutí na položku) a položka, která umožňuje odeslat email se zpětnou vazbou nebo zprávou o problému.



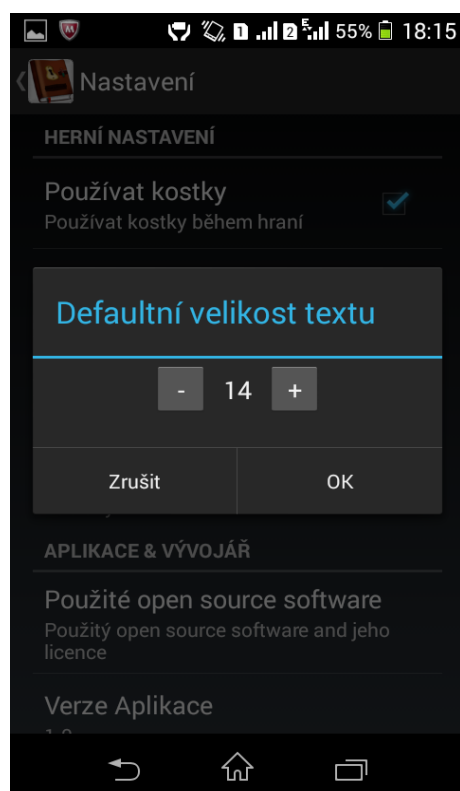
Obrázek A.11: Ukázka zobrazení knihovny (seznamu gamebooků)



Obrázek A.12: Ukázka detailu stažitelného gamebooku



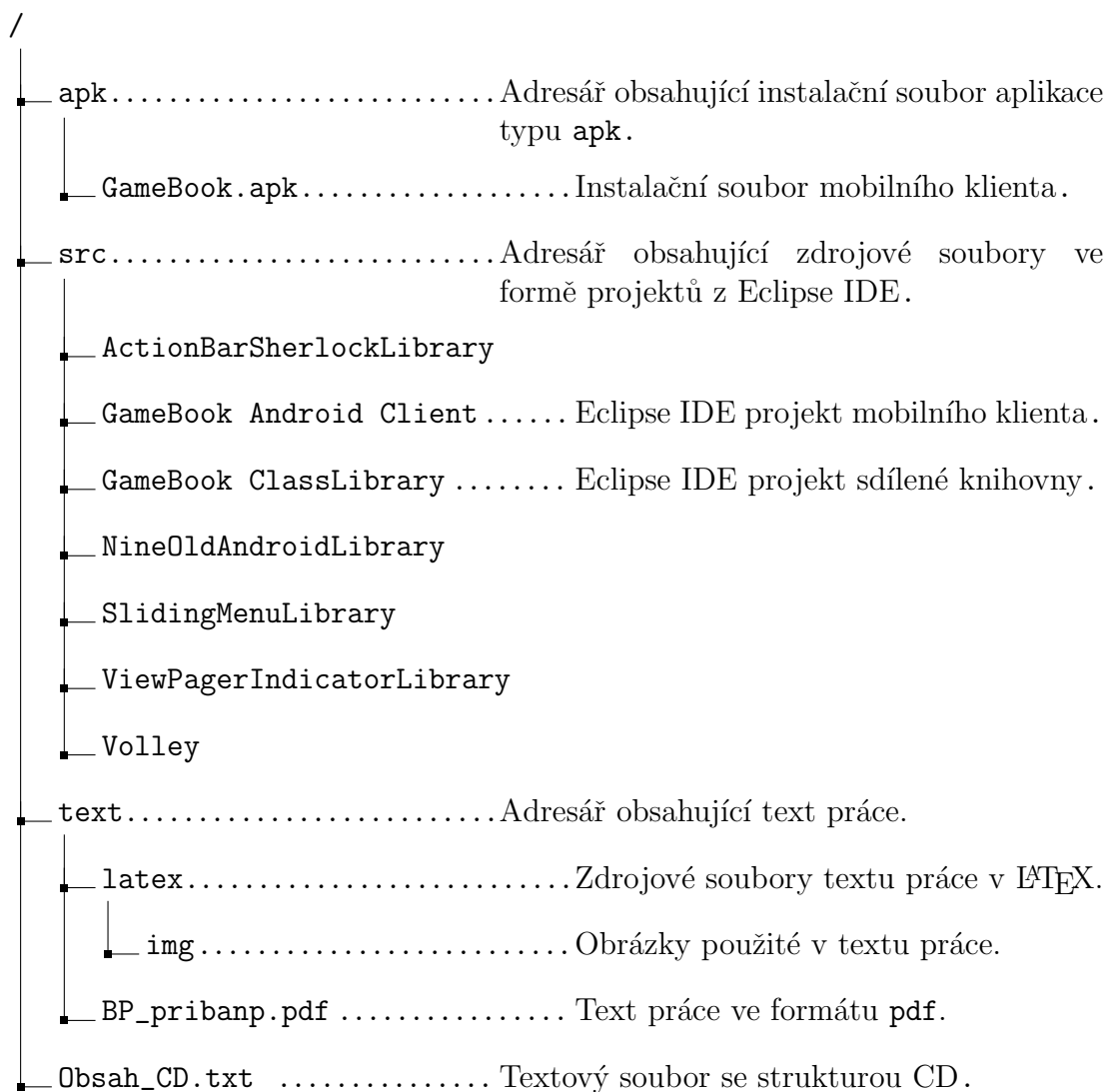
Obrázek A.13: Ukázka obrazovky nastavení



Obrázek A.14: Ukázka dialogu pro výběr základní velikosti písma

B Obsah CD

Příložené CD obsahuje text práce ve formátu pdf spolu s jeho zdrojovými L^AT_EX soubory (adresář text). Dále obsahuje výslednou aplikaci ve formě instalačního souboru GameBook.apk v adresáři apk a také ve formě zdrojových souborů. Zdrojové soubory jsou umístěny ve složce src. V této složce jsou obsaženy jednotlivé Eclipse IDE projekty knihoven a mobilního klienta. Adresářová struktura je vidět na obr. B.1.



Obrázek B.1: Adresářová struktura příloženého CD

C Použité knihovny

Knihovna	Dostupná z adresy
logback-android	https://github.com/tony19/logback-android
logback	http://logback.qos.ch/
ActionBarSherlock	http://actionbarsherlock.com/usage.html
Nine Old Androids	http://nineoldandroids.com/
ViewPagerIndicator	http://viewpagerindicator.com/
SlidingMenu	https://github.com/jfeinstein10/SlidingMenu

Tabulka C.1: Adresy, ze kterých jsou dostupné použité knihovny