

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra informatiky a výpočetní techniky

Bakalářská práce

**Hra na platformě Android
propagující FAV**

Plzeň, 2015

Miroslav Gryč

Prohlášení

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

V Plzni dne 7. května 2015

Miroslav Gryč

Poděkování

Tímto bych chtěl poděkovat svému vedoucímu bakalářské práce Ing. Ladislavu Pešíčkovi za cenné rady a nápady, které mi během tvorby bakalářské práce poskytl.

Abstract

The main goal of this work is to create game for mobile devices based on Android platform, which will simulate and describe real process of a study at The Faculty of Applied Sciences in Pilsen. The game simulates the course of study at this faculty and helps applicants with better orientation in university campus. The game also showing basic information about the study at the university.

Bachelor's thesis describes the process of application development which includes crating game scenario and architecture design, choosing appropriate technologies, implementation and testing. The thesis also describes possible extension of the game. Programmer and user documentation is included.

Abstrakt

Cílem této práce je navrhnout, vytvořit a následně otestovat aplikaci pro mobilní zařízení na platformě Android, která propaguje Fakultu aplikovaných věd v Plzni. Hra simuluje průběh studia na této fakultě a pomáhá uchazečům o studium lépe se orientovat v univerzitním kampusu a zároveň sděluje základní informace o průběhu studia na vysoké škole.

Bakalářská práce popisuje průběh vývoje aplikace, který začíná výběrem vhodného scénáře pro hru. Poté jsou zmíněny použité technologie a je nastíněna implementace, ve které jsou popsány třídy a jejich metody. Na závěr je popsáno testování aplikace a návržení jejího možného rozšíření. Obsahuje programátorskou i uživatelskou dokumentaci k aplikaci.

Obsah

1. Úvod	1
2. Návrh aplikace.....	2
2.1 Cíle hry.....	2
2.2 Možnosti interakce s uživatelem.....	3
2.3 Výběr scénáře hry	4
2.3.1 Porovnání s aplikací FAVagotchi.....	4
2.4 Návrh hry	5
2.5 Možné konce hry.....	8
2.6 Další využití aplikace	8
3. Použité technologie	9
3.1 Android	9
3.2 Volba úrovně API	10
3.3 Android Studio	12
3.4 Perzistentní ukládání dat	13
3.5.1 SQLite databáze.....	13
3.5.2 Shared Preferences	13
3.5.3 Ukládání do souboru.....	14
3.6 Aktivity (Activity).....	14
3.7 View	16
3.8 Rozvržení (Layout)	16
3.8.1 Linear Layout (lineární rozvržení)	17
3.8.2 Relative Layout (relativní rozvržení)	17
3.9 Pop-up zprávy, upozornění uživatele.....	18
3.9.1 Bublinový dialog (Toast).....	19

3.9.2	Varovné/informativní dialogy	19
3.9.3	Oznámení (Notifications)	20
3.10	Záměry (Intents).....	21
4.	Struktura aplikace	22
4.1	Struktura projektu.....	23
4.2	Použité třídy	23
4.2.1	WelcomeScreenActivity.java	23
4.2.2	TimeActivity.java	25
4.2.3	TaskActivity.java.....	26
4.2.3	NtisActivity.java.....	27
4.2.4	GeneralActivity.java.....	27
4.2.5	GameOverActivity.java.....	27
4.2.6	ExamActivity.java	27
4.2.7	CampusActivity.java	28
4.2.8	Utils.java.....	28
4.2.9	Další pomocné třídy	28
5.	Testování a distribuce aplikace	29
5.1	Fyzická zařízení	29
5.2	Emulátor	31
5.3	Distribuce aplikace.....	33
6.	Možná rozšíření	34
6.1	Více lokací a stavových proměnných	34
6.2	Rozvrh předmětů.....	34
6.3	Rozšíření textů ve hře.....	35
6.3.1	Internacionalizace.....	35
6.3.2	Přidání zkuškových otázek	35
6.4	Grafické úpravy.....	36

6.5 Platforma iOS.....	36
7. Závěr.....	37
Přehled zkratk	38
Literatura.....	39
Přílohy.....	41
A Instalační příručka	42
B Uživatelská příručka.....	43
B.1 Ovládání aplikace	43
B.2 Náhledy aplikace	45
C Obsah CD	49

1. Úvod

V současné době patří již mobilní telefony k zařízením, které se nevyužívají pouze na telefonování. Nabízí se velmi široké spektrum aplikací, o které můžeme svůj chytrý telefon obohatit. Tato bakalářská práce popisuje průběh vývoje herní aplikace pro platformu Android, která má za cíl propagovat Fakultu aplikovaných věd v Plzni a simulovat průběh studia na této fakultě.

Cílem této aplikace je tedy seznámit studenty středních škol s vysokoškolským systémem a v neposlední řadě jim pomoci se lépe orientovat v univerzitním kampusu. Zároveň tak oslovit co nejvíce potenciálních uchazečů o studium prostřednictvím mobilní aplikace, kterou lze spustit na momentálně nejrozšířenější platformě mobilních zařízeních.

Čtenář je nejprve seznámen s návrhem hry a výběrem vhodného scénáře, poté jsou zmíněny použité technologie, které jsou využity pro tvorbu aplikace. Další kapitola popisuje strukturu aplikace, použité třídy a její metody. Tato část by se dala označit jako programátorská dokumentace. Poté následuje kapitola, která se věnuje testování aplikace a její možné distribuci mezi veřejnost. Poslední kapitola obsahuje možná rozšíření, kterými by se dala aplikace do budoucna vylepšit.

2. Návrh aplikace

V první řadě bylo nutné zvolit typ vhodné hry, kterým se bude děj aplikace odvíjet. Dále bylo třeba rozhodnout o možnostech interakce s uživatelem a poté vybrat vhodný výběr herního scénáře, který je v této kapitole popsán.

2.1 Cíle hry

Hlavním cílem hry je předat vybrané informace z reálií FAV prostřednictvím aplikace pro chytré telefony, která bude snadno rozšiřitelná a bude kompatibilní s co největším počtem telefonů na trhu. Aplikace, ve které se hráč dozví určité reálie o univerzitním areálu, uvidí skutečné univerzitní prostory a budovy, ve kterých probíhá výuka a bude si tak moci lépe představit průběh studia na vysoké škole. Dále představí výzkumnou činnost, která probíhá na katedře KIV (například v neuroinformatické laboratoři). Aplikace je primárně určena pro potencionální uchazeče o studium na vysoké škole, tedy pro studenty středních škol. Dává si proto za cíl také přiblížení vysokoškolského života jako takového. Seznamuje je tedy s novými termíny, které doposud neznají. Hráč zjistí, že existují přednášky a cvičení, poté je potřeba absolvovat semestrální práce, zápočty a na závěr zkoušku. Ta je ve hře reprezentována zobrazením jednoduchých otázek z technické problematiky. Aplikace slouží také jako prostředek pro získávání aktuálních informací o fakultě, které stahuje z webu. Tyto informace mohou být tedy kdykoliv editovány, což udržuje aplikaci stále aktuální. Má tedy i po dohrání herního příběhu stále praktické využití a s odstupem času neztrácí na aktuálnosti. Případně se zde nabízí možnost zjištění rozšířenosti aplikace podle počtů stažení ze serveru.

2.2 Možnosti interakce s uživatelem

Většina dnešních mobilních zařízení je vybavena operačním systémem, nejznámějšími jsou Android, iOS a Windows Phone. Také jsou vybaveny celou řadou senzorů, které se dají využít k interakci s uživatelem. Při tvorbě herní aplikace se nejčastěji využívají tyto senzory [5]:

- **Akcelometr**: také nazývaný jako senzor zrychlení slouží k měření rychlosti pohybu a k měření vibrací. Princip spočívá v zaznamenání změny z konstantní nebo nulové rychlosti. K tomu slouží mikroskopické krystaly, na kterých vzniká vibracemi napětí odpovídající určitému zrychlení. Jedná se o tzv. piezoelektrický jev [5]. Určuje směr pouze ve dvou osách. Přesnosti akcelerometru zvyšují aplikace, které umožňují provádět kalibraci tohoto senzoru.

- **Gyroskop**: detekuje naklonění či natočení přístroje. Oproti akcelerometru neměří zrychlení, ale úhlovou rychlost a rozpoznává pohyb i ve třetí ose. Proto je jeho použití výhodné v kombinaci s akcelerometrem.

S využitím těchto senzorů připadá v úvahu možnost ovládat herní postavičku nakloněním zařízení, která by se takto pohybovala po obrazovce. Tento způsob jsem však nepovažoval pro zadaný typ hry vhodný. Využívá se spíše u her, ve kterých se například udržuje kulička na dráze, řídí auto, apod. Vhodnější mi přišlo zvolit nejpřirozenější způsob interakce, a to dotykové ovládání pomocí kapacitního displeje, díky kterému musí například na leteckém snímku areálu ZČU rozpoznat určité budovy a po kliknutí na jejich lokaci se dostává na další obrazovku s již konkrétní budovou. V té se již nachází skupina tlačítek, které reprezentují jednotlivé místnosti v dané budově a kliknutím na ně se spouští další události.

Další způsob interakce mezi fakultou a uživatelem skrze aplikaci je realizován pomocí připojení k internetu, díky kterému získáme možnost uživatele informovat o určité aktuální informaci o fakultě v průběhu hry.

2.3 Výběr scénáře hry

Při výběru scénáře byl kladen důraz hlavně na propagaci fakulty. Aplikace částečně vychází z myšlenky již existující aplikace s názvem FAVagotchi, která je pro platformu Windows Phone a byla tématem bakalářské práce vydané v roce 2014 [15]. Rozhodně však nebylo cílem vytvořit identickou aplikaci pro platformu Android.

V mém návrhu se jedná o 2D hru, ve které se hráč přesouvá pomocí dotykové obrazovky mezi jednotlivými lokacemi v areálu ZČU, kde následně plní různé úkoly. Tyto úkoly se zobrazují postupně. Vždy po splnění přechozího se objeví další, dokud není hra u konce. Nejsou omezeny časově, hráč tedy nemusí aplikaci zapínat přesně v konkrétní čas, ale může úkoly plnit kdykoliv chce. V průběhu hry ubývá stavová proměnná *energie*, která se snižuje při přesunu mezi lokacemi. Dále ubývá také při absolvování přednášky, cvičení, nebo zkoušky. Tuto proměnou je potřeba průběžně doplňovat v budově menzy formou oběda, který lze koupit za další stavovou proměnnou *peníze*.

2.3.1 Porovnání s aplikací FAVagotchi

Jak již bylo zmíněno, aplikace FAVagotchi pro Windows Phone byla představena jako bakalářská práce v roce 2014. Její velkou nevýhodou však byla zvolená platforma, která není příliš rozšířena. Podle celosvětových statistik [16] byl koncem roku 2014 Windows Phone zastoupen pouze 2,8%, oproti tomu Android ve stejném období 76,6%. Spoustu uživatelů si ji tedy nemohlo vyzkoušet. Aplikace byla také velmi komplexní, odehrávala se totiž v reálném čase. Hra vyžadovala dodržování časového rozvrhu, uživatel si musel hlídat čas přednášek a navštěvovat je. Energie ubývala herní postavičce také v závislosti na čase, pokud tedy uživatel delší dobu hru nezapnul, postavička zemřela. Mělo to tedy i své nevýhody, neboť hra byla z tohoto důvodu velmi obtížná a stalo se téměř nemožné ji dohrát do konce, což mohlo vést k odrazení uživatelů od jejího opětovného hraní.

Oproti tomu navrhovaná aplikace je pro platformu Android, která je v současné době nejrozšířenější platformou pro mobilní zařízení. Při návrhu byl kladen důraz na uživatelsky příjemné grafické prostředí a hratelnost, která nebude pro uživatele příliš

náročná a bude tedy jednoduché dohrát hru až do jejího zdárného konce. Plnění úkolů tedy nezávisí na čase, úkoly se zobrazují uživateli postupně v průběhu hry.

Další přidanou funkcionalitou je také možnost stažení aktuálních informací o FAV z webu, která je detailněji popsána v kapitole 2.6.

2.4 Návrh hry

Po spuštění hry se hráči jako první zobrazí úvodní obrazovka (viz Obrázek 2.1), která obsahuje tlačítka a zaškrťovací pole pro obsluhu hry. První z tlačítek spouští hru, při které pomocí proměnné v Shared Preferences zjišťuje, jestli se jedná o prvotní spuštění hry, nebo již opětovné. Pokud se jedná o první spuštění aplikace, nastaví se stavové proměnné na výchozí hodnoty a zobrazí se úvodní nápověda, ve které je stručně popsána podstata hry. Pokud by hráč chtěl kdykoliv začít hrát hru od začátku, má k dispozici druhé tlačítko, kterým nastaví všem proměnným výchozí hodnotu a vymaže tím všechen dosavadní postup ve hře.

Dalším tlačítkem je nápověda ke hře, ve které si hráč může i v průběhu hry přečíst informace o tom, jak hrát a na co si během hry dávat pozor. Dále je zde nápověda v podobě popsaného leteckého snímku, ve kterém jsou jednotlivé budovy popsány. Při samotné hře již tyto popisky nejsou, díky tomu je hráč po určité době nucen naučit se orientovat v univerzitním kampusu. Posledním tlačítkem je tlačítko Konec, které slouží k ukončení aplikace zavoláním metody *finish()*. Na této úvodní obrazovce se nachází ještě dvě zaškrťovací pole, umožňující povolení/zakázání zvukových efektů a hudby ve hře. Jejich hodnota je opět uložena v Shared Preferences, které uchovávají informaci o tom, v jakém stavu je zaškrťovací pole po opětovném načtení úvodní obrazovky.

Při spuštění herního příběhu se hráč ocitne na univerzitním kampusu, který je znázorněn leteckým snímkem (viz Obrázek 2.1). Přesuny mezi lokacemi provádí hráč dotykem na jednotlivé budovy. Bohužel nebyl k dispozici aktuální letecký snímek, který by zachycoval i novou budovu Fakulty aplikovaných věd a nově přistavenou knihovnu. Jelikož vývoj hry začal již v době, kdy nová budova FAV byla teprve ve výstavbě, bylo následně potřeba doplnit tento letecký snímek dvěma šipkami, které odkazují na další snímky s již zmiňovanými novými budovami. K reprezentaci ostatních univerzitních

prostor jsem použil fotografie, které jsem během tvorby aplikace fotil. Jedná se tedy o aktuální fotografie místností. Celkem bylo v aplikaci použito 15 vlastních fotografií.

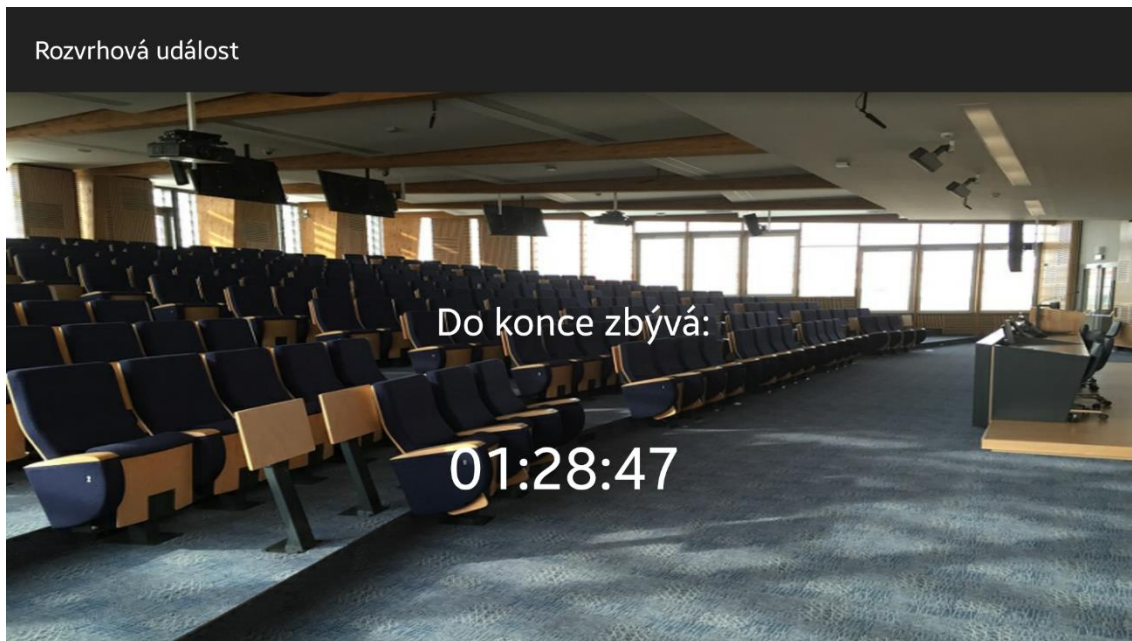


Obrázek 2.1: Obrazovka s herními lokacemi

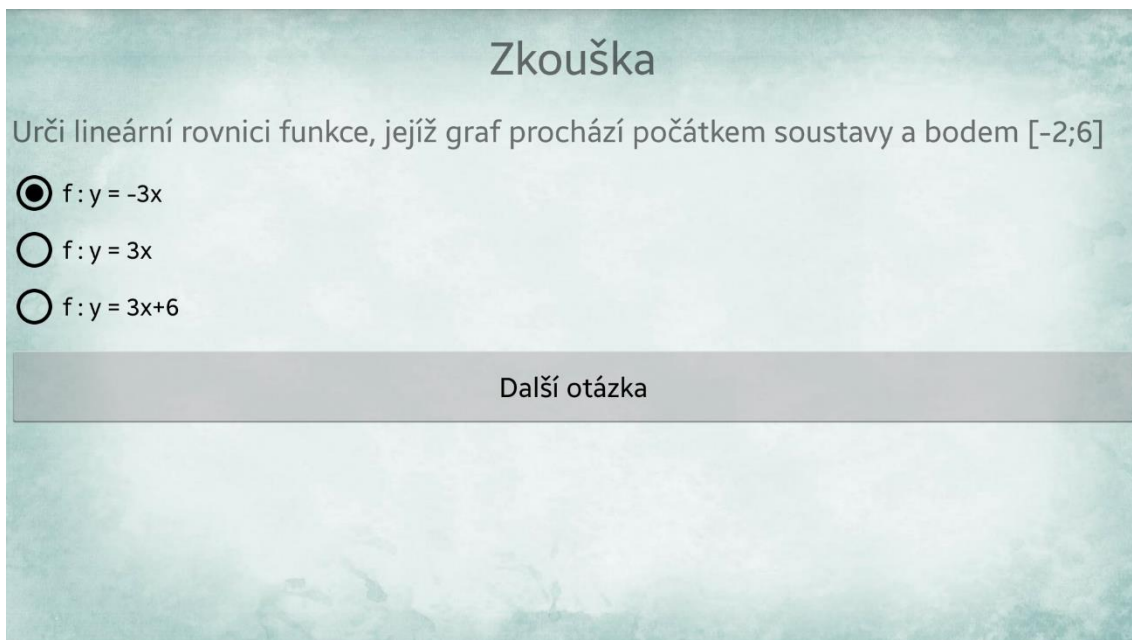
Jedním typem úkolů je rozvrhová událost v podobě absolvování cvičení nebo přednášky. Po jejím zobrazení je spuštěn odpočet času do konce události (viz Obrázek 2.3). Logický čas ve hře odpovídá 1:1 reálnému času. V průběhu odpočítávání je tato obrazovka uzamčena a není možné navštěvovat jiné lokace ve hře. Pokud tedy hráč aplikaci opustí a vrátí se k běžnému používání telefonu, je následně upozorněn v podobě oznámení (Notification), že událost skončila a po kliknutí na oznámení je hra opět spuštěna. Pokud by byla aplikace znovu spuštěna v době, kdy stále probíhá rozvrhová událost, zobrazí se opět pouze obrazovka s odpočtem času. To je zajištěno vymazáním zásobníku aktivit při ukončení aplikace a nastavením aktivity s odpočtem času do popředí, pokud stále probíhá odpočet. V dalším typu úkolů hráč navštěvuje zajímavé lokace univerzity, kde se o nich dozví určité informace a uvidí, jak prostory vypadají.

Ke konci hry uživatelé čekají zkoušky (viz Obrázek 2.4) z předmětů Matematika 1 (M1) a z předmětu Počítače a programování 1 (PPA1), otázky v těchto zkouškách jsem se snažil vytvořit tak, aby je dokázal absolvovat i běžný středoškolák, který se o danou problematiku alespoň trochu zajímá. Tyto otázky jsou vybírány náhodně,

uživateli se tedy nebudou opakovat ve stejném pořadí. A pokud zkoušku na první pokus úspěšně neabsolvuje, může mít při dalším pokusu zkoušky otázky jiné.



Obrázek 2.2: Odpočet času do konce rozvrhové události



Obrázek 2.3: Ukázka zkuškové otázky

2.5 Možné konce hry

Řádný konec hry nastává po splnění všech zadaných úkolů. V tento moment se zobrazí obrazovka, která oznamuje úspěšný konec hry. V průběhu hry však může nastat náhlý konec hry. A to v těchto případech:

- **nedostatek energie:** pokud hráč příliš „bloudí“ po areálu ZČU, mění tedy lokace a tím mu ubývá energie. Nebo si neohlídá stav energie před absolvováním cvičení, přednášky nebo zkoušky. Dochází následně ke konci hry z důvodu zkolabování herní postavičky FAVáka. Hru je poté na obrazovce oznamující konec hry možné spustit od začátku.

- **neúspěch u zkoušky:** na zkoušku má hráč klasicky tři pokusy, pokud však zkoušku úspěšně neabsolvuje ani na třetí pokus, hra pro něj končí a jedinou možností jak pokračovat ve hře, je stejná jako v reálném vysokoškolském životě a to začít celou hru (studium) od začátku.

2.6 Další využití aplikace

Jelikož aplikace primárně slouží k propagaci FAV, obsahuje mimo hraní herního příběhu i možnost stažení užitečné informace o studiu z webu. Na obrazovce se zadanými úkoly je k dispozici tlačítko *STÁHNI AKTUÁLNÍ INFO O FAV*. Při kliknutí na tlačítko se stáhne textový řetězec z webu, kde je uložen v souboru a je možno tento text kdykoliv editovat a udržovat tak aplikaci stále aktuální. V tomto souboru může být například informace o tom, kdy a kde se koná den otevřených dveří, seznam aktuálních studijních oborů, nebo jakákoliv jiná aktuální informace, prostřednictvím které lze sdělit to nejaktuálnější pro cílovou skupinu hráčů. Délka této informace může být maximálně 2048 znaků. Krátké zprávy se zobrazí přímo na pozadí obrazovky, pokud zpráva přesáhne délku 110 znaků, zobrazí se v informačním dialogovém okně, z důvodu lepšího zobrazení na telefonech s nízkým rozlišením displeje. Další výhodou síťové komunikace je i skutečnost, že může být monitorováno, kolik uživatelů tuto informaci stáhlo.

3. Použité technologie

Aplikace je vyvíjena pro platformu Android v jazyce Java. Jako IDE pro vývoj hry byl použit nejprve Eclipse s nainstalovaným ADT pluginem, což byl pro Android standard v době začátku vývoje aplikace. Avšak koncem roku 2014 společnost Google dokončila první finální verzi vývojového prostředí Android Studio ve verzi 1.0, které se poté stalo oficiálním IDE pro Android. Vývoj aplikace dále pokračoval v tomto IDE.

V této kapitole jsou dále popsány technologie, které jsou v práci použity. Velká část této kapitoly se poté věnuje seznámení s komponentami použitými ve hře a vysvětlení základních pojmů.

3.1 Android

Android je rozsáhlý operační systém společnosti Google, který je založen na Linuxovém jádře, které zajišťuje zabezpečení systému, správu procesů a paměti, přístup k síti a ovladače pro komponenty. První oficiální verze (Android 1.0) byla veřejnosti představena 23. září 2008 na zařízení HTC Dream (T-Mobile G1) [2].

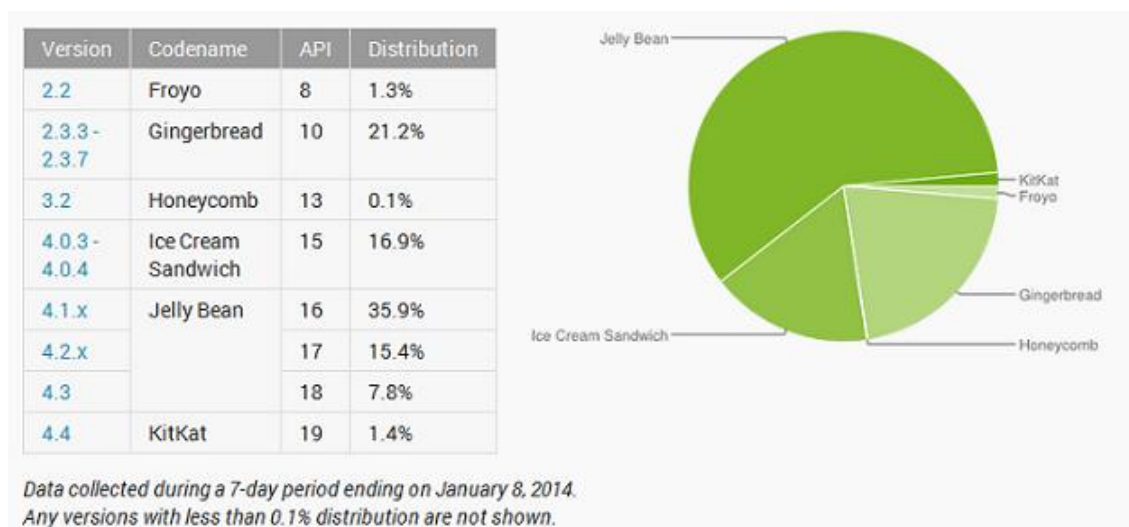
Jedná se o software s otevřeným zdrojovým kódem (open-source). Proto většina předních výrobců mobilních zařízení vyvinula vlastní verzi systému s různými nadstavbami a upraveným uživatelským rozhraním. Například TouchWiz od firmy Samsung, HTC Sense od firmy HTC nebo MOTORBLUR od firmy Motorola.

Google pro pojmenování zásadních updatů Androidu používá názvy dezertů. V současné době již byla představena doposud nejnovější verze Androidu 5.1 s označením Lollipop a verzí API 21, která přináší například zobrazení oznámení (notifikací) i na uzamčené obrazovce [14].

3.2 Volba úrovně API

Velkou výhodou aplikací pro platformu Android, je její velmi široké zastoupení na trhu s mobilními zařízeními. Jelikož je cílem aplikace zaujmout co nejvíce uživatelů, bylo důležité i vhodné zvolení **minimální verze API (Min version)**, pro kterou bude aplikace vyvíjena. Zvolíme-li příliš starou verzi, okrádáme se o prvky novějších verzí. Pokud naopak zvolíme verzi příliš novou, spoustu uživatelů nebude moci aplikaci nainstalovat na svá starší zařízení. Zvolení **cílové verze API (Target version)** určuje, na kterém zařízení se očekává, že bude aplikace nejčastěji využívána.

Naštěstí společnost Google, Inc. pravidelně zveřejňuje zastoupení jednotlivých verzí systému. Tuto statistiku získává na základě stahování aplikací z Google Play [1]. Díky tomu se vývojáři mohou snadněji rozhodnout, kterou verzi API mají zvolit. Na následujícím grafu (viz Obrázek 3.1) je znázorněno zastoupení jednotlivých verzí API v době začátku vývoje navrhované aplikace.



Obrázek 3.1: Zastoupení verzí Androidu – leden 2014

Z grafu jde jasně vidět, že API verze 8 s názvem Froyo se téměř přestávala používat, naopak nejvíce v té době vzrůstal počet zařízení s Jelly Bean. Avšak začátkem roku 2014 byla zařízení s API 10 Gingerbread stále zastoupena 21,2% [1]. Proto byla jako minimální verze API nastavena právě tato. Jako cílové API bylo zvoleno API 16 Jelly Bean, které bylo nejrozšířenější.

Nové oficiální vývojové prostředí Android Studio IDE ve verzi 1.0, již nabízí zobrazení jednotlivých API úrovní a procentuální pokrytí dané úrovně.

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
2.2 Froyo	8	99,5%
2.3 Gingerbread	10	90,4%
4.0 Ice Cream Sandwich	15	82,6%
4.1 Jelly Bean	16	61,3%
4.2 Jelly Bean	17	40,9%
4.3 Jelly Bean	18	33,9%
4.4 KitKat	19	< 0,1%
5.0 Lollipop	21	

Obrázek 3.2: Zastoupení verzí Androidu zobrazené v Android Studiu (březen 2015)

V současné době pokrývá již 99,5% trhu mobilních zařízení s operačním systémem Android verze API 10 (Gingerbread) a vyšší (viz Obrázek 3.2). Kvůli použití některých prvků, které obsahuje Android až od API 11, byla v průběhu hry změněna minimální požadovaná verze z původních API 10 na API 11 (Android 3.0). Jako cílové API zůstalo API 16 (Jelly Bean).

3.3 Android Studio

Android Studio je nové oficiální vývojové prostředí pro vývoj aplikací na zařízení s operačním systémem Android. Nejedná se pouze o mobilní telefony a tablety, ale i například chytré hodinky a televizory. Vývojové prostředí je založené na IntelliJ IDEA, díky kterému získává všechny jeho možnosti práce s kódem (navigace v kódu, našeptávání, refaktoring, analýza kódu, atd.), ve kterých je IDEA mezi nejlepšími v oboru. Součástí instalace je kromě samotného Android Studia IDE i Android SDK Tools, kompilátor Android a základní emulátory s plnohodnotným systémem Android [3].

Oproti Eclipse nabízí usnadnění lokalizace aplikací, které jsou uložené v *resources* a díky tomu je překlad zjednodušen, stačí zvolit *Add translation*, vybrat jazyk a v zobrazené tabulce editovat překlad jednotlivých textů.

Velkým přínosem pro tvorbu vyvíjené aplikace byla další nová funkce Android Studia, kterou je navrhování vzhledu aktivit. Stejně jako v Eclipse zde jde navrhovat design v XML, nebo v Design módu, ve kterém lze jednotlivé komponenty přetahovat myší do Layoutu. Zde je nově náhled ve vybraném rozlišení i při psaní přímo v XML módu. Velkou výhodou je potom funkce *Preview All Screen Sizes*, které zobrazí najednou náhledy ve všech možných rozlišeních. Jelikož jsem měl neustále k dispozici jen jeden testovací telefon, bylo tím velmi ulehčeno testování na telefonech s menším, či naopak mnohem větším rozlišením displeje. Finální testování probíhalo i na dalších fyzických zařízeních.

Pro překlad zde neslouží Ant nebo Maven, obojí nahrazuje Gradle. Jedná se o automatizační nástroj, neboli jazyk pro automatizaci Domain Specific Language (DSL) [3]. Dalo by se říci, že se jedná o kombinaci Antu, ze kterého bere jeho sílu a flexibilitu a Mavenu, ze kterého využívá konvence, dependency management a pluginy [4].

3.4 Perzistentní ukládání dat

Systém Android podporuje různé způsoby pro ukládání perzistentních dat. Každý ze způsobů se hodí na něco jiného. Avšak všechna tato úložiště jsou privátní a slouží pouze k ukládání dat z jedné aplikace. Prvním ze způsobů je ukládání do běžného souboru, další možností je vestavěná SQLite databáze a poté tu jsou Preferences, ke kterým patří i tzv. Shared Preferences, které oproti klasickým Preferences nejsou pouze pro jednu aktivitu, ale jsou globální, tedy pro celou aplikaci. A právě tento způsob ukládání perzistentních dat byl použit ve vyvíjené aplikaci.

3.5.1 SQLite databáze

SQLite sice není databázový systém určený pouze pro Android, ale je velmi šetrný k paměti zařízení a poměrně rychlý, díky této výhodě je často používán v mobilních zařízeních. První verze vznikla v roce 2000 [2]. Tato databáze reprezentuje relační databázový systém, který je obsažen v jedné knihovně.

3.5.2 Shared Preferences

Ukládání perzistentních dat pomocí Shared Preferences funguje na jednoduchém principu key-value (klíč-hodnota). Jedná se o úložiště, pracující jen s primitivními datovými typy. Ačkoliv podle názvu to tak nevypadá, Shared Preferences neslouží pouze pro uživatelská nastavení, ale dají se použít i pro jakékoliv key-value data.

Ukázka použití Shared Preferences:

- vytvoření objektu SharedPreferences:

```
SharedPreferences pref =  
getApplicationContext().getSharedPreferences("MyPref", MODE_PRIVATE);
```

- vytvoření objektu editoru, který slouží k ukládání dat:

```
Editor editor = pref.edit();
```

- uložení dat:

```
editor.putInt("key_name", "int value");
```

- načtení dat (pokud hodnota pro daný klíč neexistuje, vrátí hodnotu druhého parametru):

```
pref.getInt("key_name", 0);
```

- uložení změn do SharedPreferences:

```
editor.commit();
```

3.5.3 Ukládání do souboru

Nejjednodušším způsobem je pak zápis do běžného souboru. K tomu je potřeba získat objekt typu `InputStream`, popřípadě `OutputStream`. Metoda `openFileInput()`, respektive `openFileOutput()` poté obsahuje jako parametr název souboru, ke kterému chceme přistupovat.

3.6 Aktivity (Activity)

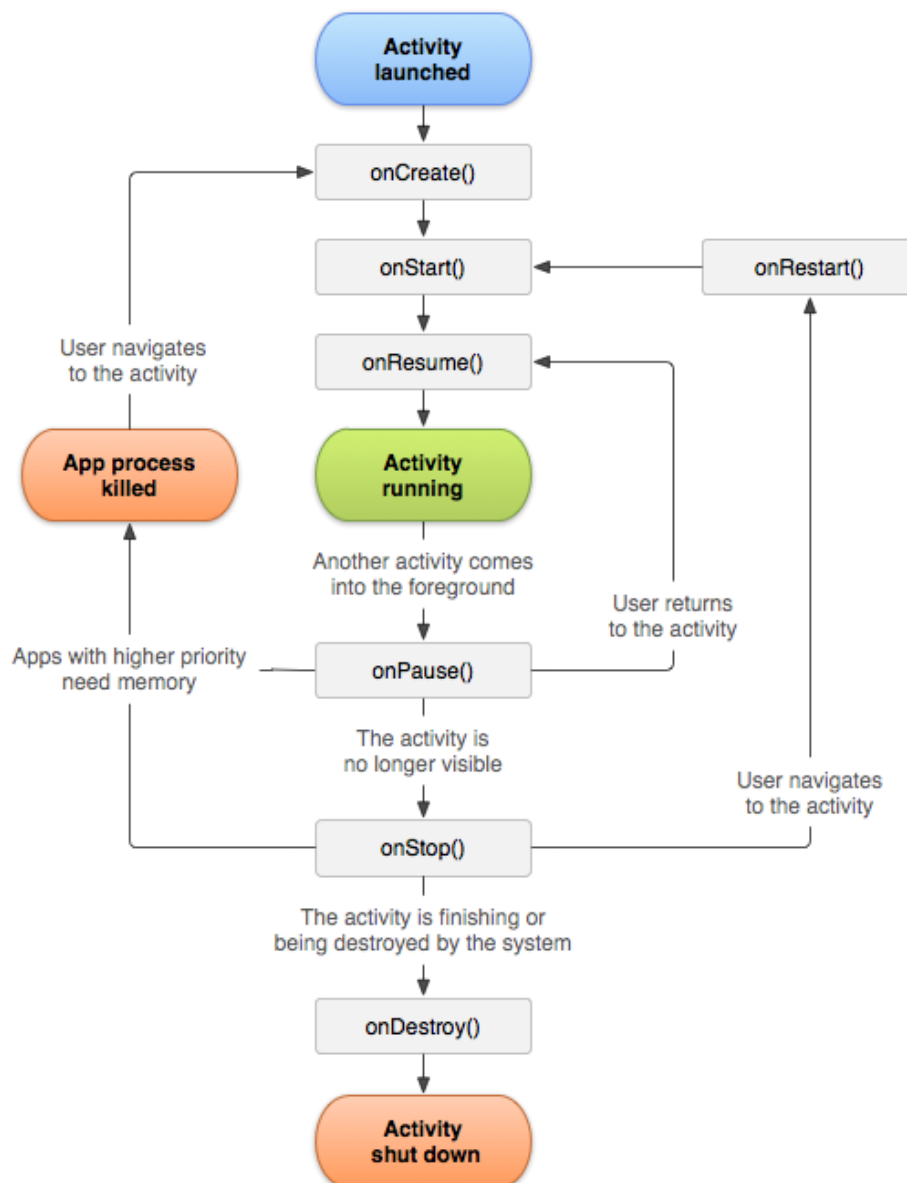
Aktivity představuje prezentační vrstvu aplikace. Jedná se o jednu obrazovku aplikace, která obvykle vyplňuje celou obrazovku zařízení. Tyto aktivity jsou následně vázány mezi sebou, pokaždé když se spustí nová aktivita, předchozí je pozastavena a uložena do zásobníku, který funguje na principu LIFO (last in, first out). Stiskne-li uživatel tlačítko Zpět, vybere se ze zásobníku poslední uložená aktivita a právě vykonávaná aktivita se zruší. Jedna aktivita je v aplikaci nastavená jako hlavní a ta se zobrazuje při spuštění aplikace.

OS Android se stará o vytvoření instancí aktivit a poté i o jejich zrušení. Na následujícím diagramu (viz Obrázek 3.3) [7] je znázorněn životní cyklus aktivity. Tento cyklus by se dal rozdělit na tři stavy:

Aktivní – aktivita je aktuálně zobrazena uživateli

Pozastavená – částečně překryta jinou aktivitou, nepřijímá uživatelský vstup

Zastavená – aktivita není vidět, ale nebyl ještě zničen její objekt



Obrázek 3.3: Životní cyklus aktivity [7]

V případě navrhované aplikace však byla potřeba v některých situacích nespouštět hlavní aktivitu, například při probíhající rozvrhové události, při které se odpočítává čas do konce přednášky nebo cvičení. V tento moment by nebylo vhodné, aby uživateli bylo umožněno procházet mezi ostatními aktivitami. Proto bylo potřeba zajistit, aby se při stisknutí tlačítka Zpět aplikace nevrátila v zásobníku o jednu aktivitu zpět, ale aby se aplikace ukončila a při opětovném spuštění v době, kdy stále probíhá odpočet času, se vrátila zpět do této aktivity.

3.7 View

Tato třída představuje základní stavební kámen pro komponenty uživatelského rozhraní. View je specifikováno staticky v XML souborech, které jsou uloženy ve složce `/res/layout`. V případě vyvíjené aplikace, bylo nutné vytvořit tyto layouty dva. A to konkrétně pro aktivitu zobrazující kampus ZČU, ve které se nachází spousta klikacích oblastí a ty musí korespondovat se zobrazenými budovami na pozadí aktivity. První se nachází klasicky ve složce `/res/layout` a druhý ve složce `/res/layout-xlarge`, při spuštění aplikace se systém postará o adaptaci aplikace podle velikosti a rozlišení obrazovky [9]. V případě tabletů by tedy byl použit layout ze složky `/res/layout-xlarge`.

3.8 Rozvržení (Layout)

Při založení aktivity se k ní vytvoří příslušný `.xml` soubor, ve kterém je definován vzhled dané aktivity. Můžeme vycházet z předpřipravených šablon, nebo vytvořit vlastní [10]. Android Studio umožňuje editaci Layoutu v XML nebo v Design módu. V XML módu zobrazuje náhled obrazovky, ve kterém jde po editaci XML kódu vidět, jak bude rozložení vypadat na vybraném zařízení.

3.8.1 Linear Layout (lineární rozvržení)

V tomto rozvržení jsou prvky uspořádány vertikálně pod sebe, nebo horizontálně vedle sebe. To je určeno atributem `android:orientation`. Dalším atributem je `android:gravity`, který určuje, jak bude prvek zarovnán (`left`, `right`, `center_horizontal`). Lineární rozvržení umožňuje nastavit atribut `weight`, pomocí kterého je umožněno nastavit jeho důležitost (váhu). Atribut s větší hodnotou potom dostane víc prostoru k vykreslení. Dojde-li k zaplnění celé obrazovky, umožní se automaticky posun.

3.8.2 Relative Layout (relativní rozvržení)

Při použití relativního rozvržení definujeme pozici relativně vůči ostatním objektům, nebo k obrazovce. Pozicování by se dalo rozdělit do dvou základních druhů:

- **vůči `RelativeLayout`** – určují, která strana view bude ležet na stejné přímce se stranou `RelativeLayout`. Používají se tyto čtyři atributy:

- `android:layout_alignParentLeft`
- `android:layout_alignParentTop`
- `android:layout_alignParentRight`
- `android:layout_alignParentBottom`

- **vůči ostatním view** – zarovnávají se vzhledem k ostatním view, které se nachází ve stejném `RelativeLayout`. Těchto atributů již je více:

- `android:layout_alignLeft`
- `android:layout_alignTop`
- `android:layout_alignRight`
- `android:layout_alignBottom`
- `android:layout_toLeftOf`
- `android:layout_above`
- `android:layout_toRightOf`
- `android:layout_below`

V následující ukázce (viz Obrázek 3.4) je znázorněno relativní rozložení pro aktivitu `CampusActivity`:


```

<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/areal"
    tools:context=".MainActivity" >

    <Button

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:id="@+id/buttonFEL"
        android:layout_above="@+id/buttonStred"
        android:layout_toRightOf="@+id/textViewEnergie"
        android:layout_below="@+id/textViewEnergie"
        android:layout_alignRight="@+id/buttonMenza"
        android:layout_alignEnd="@+id/buttonMenza"
        android:background="@android:color/transparent"
        android:onClick="buttonFELClicked" />

</RelativeLayout>

```

Obrázek 3.4: Ukázka relativního rozložení pro aktivitu CampusActivity

3.9 Pop-up zprávy, upozornění uživatele

V OS Android je k dispozici mnoho způsobů, kterými lze informovat uživatele o nějaké události. Ta může být pouze informativní, k té slouží bublinový dialog (Toast), který uživatele informuje krátkým textem a ten po chvíli sám zmizí. Není tedy vhodně tento způsob používat na důležité informace. Je vhodný například na zprávy typu: „připojení proběhlo úspěšně“, atd. Pro důležitější zprávy existují varovné/informativní dialogy, které jsou zobrazené tak dlouho, dokud uživatel zprávu nepotvrdí stiskem tlačítka OK. Varovné zprávy zpravidla mohou obsahovat místo jediného tlačítka pro potvrzení i možnosti ANO, NE. Tento dialog se používá při důležitých akcích, které vyžadují konečné potvrzení uživatelem, aby bylo zamezeno nechtěnému kliknutí.

3.9.1 Bublinový dialog (Toast)

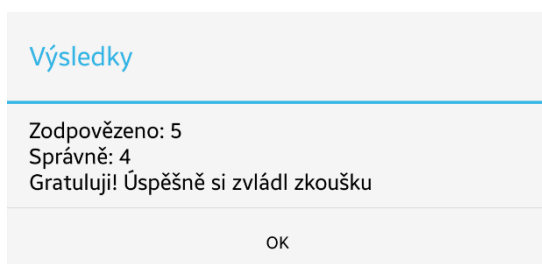
Jedná se o nejjednodušší informační dialog v Androidu (viz Obrázek 3.4). Vytváří se pomocí třídy `Toast`. Jde tedy o informativní upozornění bez dalších komponent (tlačítek). Objeví se v tzv. bublině a po několika sekundách sám zmizí, tento čas lze nastavit metodou `setDuration()` pouze na dvě hodnoty: `LENGTH_SHORT`, nebo `LENGTH_LONG`.



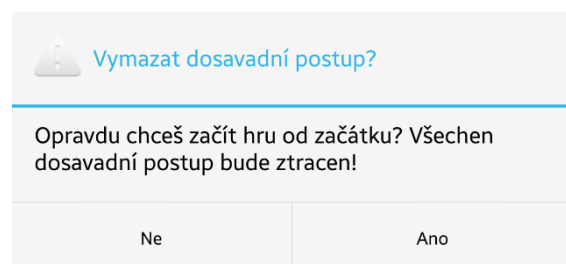
Obrázek 3.5: Ukázka objektu `Toast`

3.9.2 Varovné/informativní dialogy

Tyto dialogy se vytváří prostřednictvím třídy `AlertDialog`. Dialogy vyžadují asistenci uživatele, nezmizí tedy automaticky, jako to bylo v případě předchozích Toastů. Ale jsou zobrazeny, dokud uživatel neklikne na tlačítko. Používají se tedy pro důležitější zprávy, u kterých chceme mít jistotu, že si je uživatel opravdu přečte. Prostřednictvím třídy `Builder` nastavíme veškeré součásti dialogového okna (nadpis, text zprávy, text pozitivního tlačítka, text negativního tlačítka, apod.). Na obrázku 3.6 je ukázka dialogu s pouze kladným potvrzením a na obrázku 3.7 je již dialog s kladným i záporným potvrzením.



Obrázek 3.6 Dialog

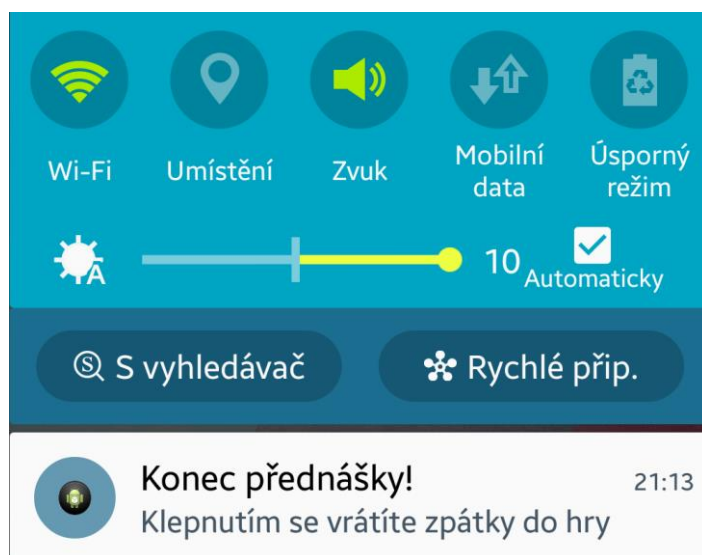


Obrázek 3.7 Dialog s potvrzením

3.9.3 Oznámení (Notifications)

Oznámení je zpráva, která se uživateli zobrazuje mimo aplikaci v tzv. oznamovací liště. Používá se pro upozornění uživatele v případech, kdy aplikace vyžaduje jeho pozornost. Nepřerušuje uživatelskou aktuální činnost a může si notifikaci přečíst kdykoliv. Klepnutím na notifikaci vede ve většině případů k otevření dané aplikace. Notifikace se s novějšími verzemi Androidu měnily, největší změna přišla s verzí Jelly Bean (API 16), změnil se jejich vzhled i způsob vytváření. [11].

Na obrázku 3.7 je znázorněna notifikace z vyvíjené aplikace, která oznamuje konec přednášky. Ukázka je znázorněna na nejnovějším Androidu 5.0.1 v telefonu Samsung Galaxy S4.



Obrázek 3.7: Ukázka notifikace

3.10 Záměry (Intents)

Záměry jsou asynchronní zprávy, které umožňují komponentám aplikace vyžádat funkcionalitu od jiných komponent Androidu, například spuštění jiné aktivity. Umožňují komunikovat s komponentami stejné aplikace, ale i s komponentami jiných aplikací. Například mohou spustit externí aktivitu fotoaparátu, pro pořízení snímku. Záměry jsou součástí `android.content.Intent`. [2, 17]

Záměry rozdělujeme do dvou skupin:

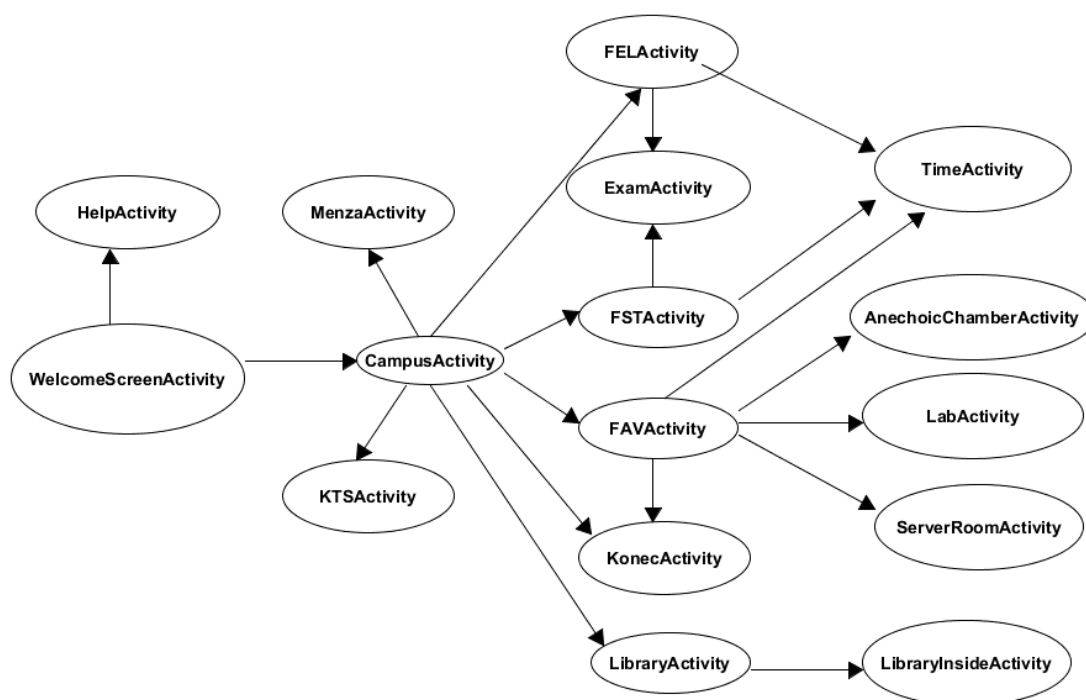
- **Explicitní** – spouští aktivity, které jsou součástí aplikace. Určují přímo třídu, která se má spustit.
- **Implicitní** – nespecifikují určitou komponentu, popisují pouze záměr (intent), ale již neurčují přesný způsob, jak ho provést. Systém poté zvolí, která komponenta bude pro tento záměr vhodná.

Ukázka použití záměru pro spuštění jiné aktivity, v tomto případě zajišťuje přechod z aktivity `PocatecniAktivita.class` do `CilovaAktivita.class`:

```
Intent i = new Intent(PocatecniAktivita.this, CilovaAktivita.class);
startActivity(i);
```

4. Struktura aplikace

Na obrázku 4.1 je znázorněna návaznost jednotlivých aktivit. Z každé aktivity se lze tlačítkem zpět vrátit do přechozí. Při běžném spuštění aplikace se zobrazí WelcomeScreenActivity, probíhá-li rozvrhová událost, spustí se po spuštění aplikace rovnou TimeActivity a není umožněno přejít do jiné aktivity během odpočítávání časovače. Spustí-li se aplikace klepnutím na notifikaci, hráč se ocitne v aktivitě, ze které byla spuštěna TimeActivity.



Obrázek 4.1 Návaznost všech aktivit v aplikaci

4.1 Struktura projektu

Projekt pro Android Studio má poměrně rozsáhlou strukturu s adresáři, zde jsou uvedeny nejpoužívanější z nich:

/java – obsahuje veškeré zdrojové soubory *.java*

/res/drawable-hdpi – obsahuje obrázky a ikony ve vysokém rozlišení

/res/layout – obsahuje *.xml* soubory, které specifikují View

/res/layout/xlarge – slouží pro zařízení s velkým displejem (tablety)

/res/values – obsahuje soubor *strings.xml*, který obsahuje veškeré texty

/res/raw – soubory zvukových efektů a hudebního pozadí aplikace

AndroidManifest.xml – soubor, který slouží pro deklaraci a propojení komponent

4.2 Použité třídy

V této kapitole jsou popsány důležité třídy v aplikaci a některé jejich metody. Nejdříve jsou uvedeny třídy, které představují jednotlivou obrazovku aplikace – aktivity. Poté třídy pomocné.

4.2.1 WelcomeScreenActivity.java

Tato třída (aktivita) reprezentuje úvodní obrazovku aplikace. Obsluhuje jednotlivé události po kliknutí na tlačítka a zaškrtačací pole. Obsahuje tyto metody:

void onCreate(Bundle savedInstanceState) – metoda, která je volána při vytvoření aktivity. Nastaví požadovaný Layout pro aktivitu. Získá přístup k jednotlivým View. Pokud se jedná o první spuštění aplikace, nastaví stavové proměnné na defaultní hodnoty.

void loadSavedPreferences () – slouží k načtení dat ze SharedPreferences, ve kterých jsou uloženy hodnoty jednotlivých zaškrtačacích polí, které určují, jestli jsou zapnuté/vypnuté zvukové efekty, popřípadě hudba. Podle těchto uložených hodnot při opětovném spuštění aplikace následně nastaví v jakém stavu je zaškrtačacího pole.

void savePreferences(String key, boolean value) – uloží hodnotu z parametru value do SharedPreferences pod klíčem z parametru key.

void buttonStartGameClicked(View Button) – pomocí Intentu spustí aktivitu CampusActivity.

void buttonClearGameClicked(View Button) – nejprve zobrazí varovný dialog a pokud je kladně potvrzen, nastaví veškeré stavové a pomocné proměnné na výchozí hodnoty.

void buttonQuitGameClicked(View Button) – ukončí hru zavoláním metody finish().

void sfcOffClicked(View v) – nastavení proměnné soundEffect podle stavu zaškrtačacího pole, tato proměnná se poté předává do ostatních aktivit.

void musicOffClicked(View v) – nastavení proměnné music podle stavu zaškrtačacího pole.

void buttonAboutClicked(View Button) – spouští aktivitu HelpActivity, která obsahuje nápovědu ke hře.

static void stopMusic() – zastavuje hudbu na pozadí aplikace.

void playMusic() – spouští hudbu na pozadí aplikace.

void onDestroy() – metoda, která je volána před ukončením aktivity. Zastavuje hudbu při ukončení aplikace.

void onResume () – metoda, která je volána bezprostředně před vytvořením aktivity. Volá metodu `loadSavedPreferences ()`.

4.2.2 TimeActivity.java

Třída obsluhující veškeré časové události ve hře (přednášky, cvičení). Zobrazuje zbývajícím čas do konce časové události a provádí veškerou práci s časem po ukončení aktivity, aby bylo následně možné po opětovném spuštění aplikace pokračovat v čase, ve kterém byla aktivita ukončena. Obsahuje jednu vnořenou třídu a metody:

class MyCountDownTimer extends CountdownTimer – třída pro časovač, která obsahuje dvě metody:

- **void onTick(long millisUntilFinished)** – která zajišťuje průběh odpočítávání. Zobrazuje zbývajícím počet milisekund, které převádí do časového formátu HH:MM:SS.

- **void onFinish ()** – která je volána při uplynutí odpočítávaného času. Zobrazuje text, oznamující konec přednášky/cvičení. Volá metodu pro vytvoření notifikace.

void onCreate(Bundle savedInstanceState) - metoda, která je volána při vytvoření aktivity. Nastaví požadovaný Layout pro aktivitu. Vytváří objekt `SharedPreferences` a Editoru.

void showNotification () – vytváří a zobrazuje notifikaci. Nastavuje texty notifikace a zařizuje, aby se při probíhající události uživatel vrátil zpět do stejné aktivity.

void onStop () – metoda, která je volána při opuštění aktivity. Uloží čas vypnutí, aby bylo možné při opětovném zapnutí nastavit nový čas pro odpočet.

void setBackground () – detekuje verzi API v zařízení a podle toho vybírá správnou metodu pro nastavení pozadí Layoutu. Pokud je verze API nižší než 16, použije se

"deprecated" metoda. V opačném případě se použije novější metoda, která je ale dostupná až od API 16.

void onResume() – nastavuje pozadí místnosti podle aktuálního úkolu a provádí operace s časem.

void onBackPressed() – obsluhuje událost kliknutí na tlačítko ZPĚT. Zobrazí dialogové okno a zařizuje, aby nebyla aplikace klasicky vrácena o jednu aktivitu v zásobníku, ale aby se vrátila na hlavní obrazovku telefonu (opustila aplikaci). To je zajištěno zavoláním Intentu pro přesun na hlavní obrazovku. Také dojde k vymazání zásobníku aktivit a nastavení této aktivity do popředí pomocí příkazu `intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP)`.

4.2.3 TaskActivity.java

Zobrazuje obrazovku s aktuálním úkolem, stavem peněz a tlačítkem pro stažení aktuálních informací o FAV z textového souboru na webu.

void onCreate() – získává přístup k SharedPreferences a zobrazuje aktuální úkol a stav peněz ve hře.

void onResume() – přidává energii o 1, tato aktivita se nebere jako lokace, tudíž by neměla ubírat energie za přesun.

void createTaskList() – vytváří ArrayList, do kterého přidává herní úkoly.

void fillQuestion(Ukol mTask) – zobrazuje zadání jednoho úkolu v TextView.

void buttonBonusClicked(View Button) – metoda obsahující vlákno, které stahuje z webu aktuální informace o FAV. Tento soubor se nachází na adrese http://students.kiv.zcu.cz/~pesicka/favgame/aktualni_informace.txt. Může obsahovat textový řetězec s maximální délkou 2048 znaků, neboť na tuto hodnotu je inicializován buffer. Pokud má zpráva méně než 110 znaků, zobrazí se na pozadí aktivity, pro delší zprávu se vytvoří dialogové okno. V případě potřeby lze tento limit upravit hodnotou proměnné `delkaKratkeZpravy`. Velikost bufferu lze upravit hodnotou proměnné `maxDelkaZpravy`.

4.2.3 NtisActivity.java

Reprezentuje novou budovu FAV a NTIS, obsluhuje události po kliknutí na tlačítka v této lokaci.

void runTimeActivity(int task) – spouští TimeActivity po kliknutí na tlačítka UC311 nebo US217.

void changeLocation(Class destination, int task) – provádí přesun do lokací, které neobsahují odpočet času (tlačítka UN323, UN340, UN343).

4.2.4 GeneralActivity.java

Jako jediná z aktivit nepředstavuje obrazovku ve hře. Jedná se pouze o pomocnou aktivitu, která obsahuje část kódu, který se opakuje ve většině ostatních aktivit. Tato aktivita dědí od ActionBarActivity a ostatní aktivity v aplikaci poté dědí od GeneralActivity. Obsahují tedy prvky z obou aktivit.

4.2.5 GameOverActivity.java

Aktivita, která se zobrazí, pokud nastane konec hry. Spouští jí CampusActivity, nebo NtisActivity. Obsahuje tlačítko pro nové spuštění hry od začátku.

void buttonRestartGameClicked(View Button) – nastaví veškeré stavové a pomocné proměnné na výchozí hodnoty.

4.2.6 ExamActivity.java

Třída (aktivita) pro zkuškové úkoly ve hře. Přidává otázky do ArrayListu, ze kterého náhodně vybírá a zobrazuje otázky ve hře. Stará se o jejich správně zobrazení v aktivitě, na závěr zkoušku vyhodnotí a zobrazí výsledky v dialogovém okně. Přidání otázek do zkoušky je podrobně zmíněno v kapitole 6.3.

void onCreate(Bundle savedInstanceState) – nastavení Layoutu a tlačítek, které obsahuje. Podle aktuálního úkolu vybere a zobrazí zkouškové otázky.

void fillQuestion(Otazka mQuestion) – vyplňuje otázku do TextView a do skupiny RadioButton.

void createQuestionListMath() – vytváří seznam otázek z matematiky.

void createQuestionListPPA() – vytváří seznam otázek z předmětu PPA1.

int selectQuestion() – slouží k náhodnému výběru otázky.

void showResult() – zobrazuje výsledky zkoušky a vybírá vhodný text podle zbývajících počtu pokusů.

4.2.7 CampusActivity.java

Představuje obrazovku s univerzitním areálem. Spouští ostatní aktivity po klepnutí na jejich lokace na mapě.

4.2.8 Utils.java

Pomocí této pomocné třídy se volá z ostatních aktivit metoda **loseEnergy(SharedPreferences pref, SharedPreferences.Editor editor)**, která ubírá hodnotu energie a ukládá jí do SharedPreferences.

4.2.9 Další pomocné třídy

Question.java - reprezentuje jednu otázku ke zkoušce.

Task.java - reprezentuje jeden herní úkol.

5. Testování a distribuce aplikace

Důležitou součástí práce bylo její testování a ověření veškeré funkcionality. Testování probíhalo na několika reálných zařízeních a také v emulátoru, který je součástí Android Studia. Tato část byla velice časově náročná, bylo potřeba ošetřit všechny možnosti, které mohou v průběhu hry nastat. Velký důraz byl také kladen na správné umístění komponent, aby bylo zajištěno správné zobrazení na telefonech s nejrůznějšími úhlopříčkami displejů a rozlišeními.

Dalším problémem bylo vhodné zvolení metod, které podporují i telefony se starší verzí úrovně API. Některé užitečné metody, které byly nejdříve zvoleny, podporovaly telefony až s verzí API 16 a vyšší. Což by ve velké míře omezilo počet telefonů, na kterých by bylo možné hru spustit. Z tohoto důvodu jsem tedy použil i některé „zastaralé“ metody (označovány `deprecated`).

5.1 Fyzická zařízení

Aplikaci jsem primárně testoval na svém telefonu Samsung Galaxy S4. Při konzultacích se zadavatelem práce bylo možné aplikaci testovat na telefonech Google Nexus S a HTC One V. Začátkem vývoje aplikace byl k testování také používán Samsung Galaxy SII s verzí Androidu 4.0.4 a Samsung SIII s verzí Androidu 4.3.

V následující tabulce (viz Tabulka 5.1) jsou porovnány tři hlavní chytré telefony, na kterých probíhalo testování. Každý telefon obsahuje rozdílnou verzi Androidu. To bylo velkou výhodou pro testování. Dalším důležitým faktorem je rozdílná velikost úhlopříčky displeje a hlavně velmi rozdílné rozlišení. To zásadně měnilo umístění lokací na pozadí a také zobrazení veškerých textů. Pomocí relativního rozložení se povedlo umístit veškeré lokace tak, aby odpovídaly budově na pozadí a to pro všechny velikosti displejů a rozlišení.

Zařízení	Samsung Galaxy S4	HTC One V	Google Nexus S
Úhlopříčka displeje	5" (111 × 62 mm)	3,7" (81 × 48 mm)	4" (87 × 52 mm)
Rozlišení displeje	1 080 × 1 920 bodů	480 × 800 bodů	480 × 800 bodů
DPI (jemnost disp.)	30 096 bodů/cm ²	9 846 bodů/cm ²	8 440 bodů/cm ²
Procesor	1 900 MHz (4 jádra)	1 000 MHz (1 jádro)	1 000 MHz (1 jádro)
RAM	2 048 MB	512 MB	512 MB
Verze Androidu	Android 4.4.2 (KitKat)*	Android 4.0.3 (Ice Cream Sandwich)	Android 4.1.2 (Jelly Bean)

Tabulka 5.1: Technické parametry testovaných telefonů

* v půlce dubna 2015, byl telefon aktualizován na současně nejnovější verzi Androidu 5.0.1 s kódovým jménem Lollipop.

Samsung Galaxy S4 (viz Obrázek 5.2) nabízí velký výpočetní výkon, běh aplikace byl tedy naprosto plynulý. Po nejdélší část testování obsahoval verzi Androidu 4.4.2 (API 19). Nebyl tedy žádný problém s nově přidanými metodami, neboť se jednalo v té době o druhou nejnovější verzi Androidu. Později proběhla aktualizace na Android 5.0.1 (API 21), v době psaní práce se jednalo o nejnovější verzi. Aplikace byla poté znovu testována, hlavní důraz byl kladen na funkčnost metod, které jsou již označovány `deprecated`. Toto značení znamená, že jsou již nahrazeny novějšími metodami, ale stále je zachována zpětná kompatibilita. Avšak i na současně nejnovější verzi stále všechny tyto metody fungovaly. Nebylo tedy potřeba zvyšovat minimální verzi API a tím zamezit používání aplikace uživatelům, kteří vlastní starší telefon.



Obrázek 5.2: Samsung Galaxy S4



Obrázek 5.3: HTC One V

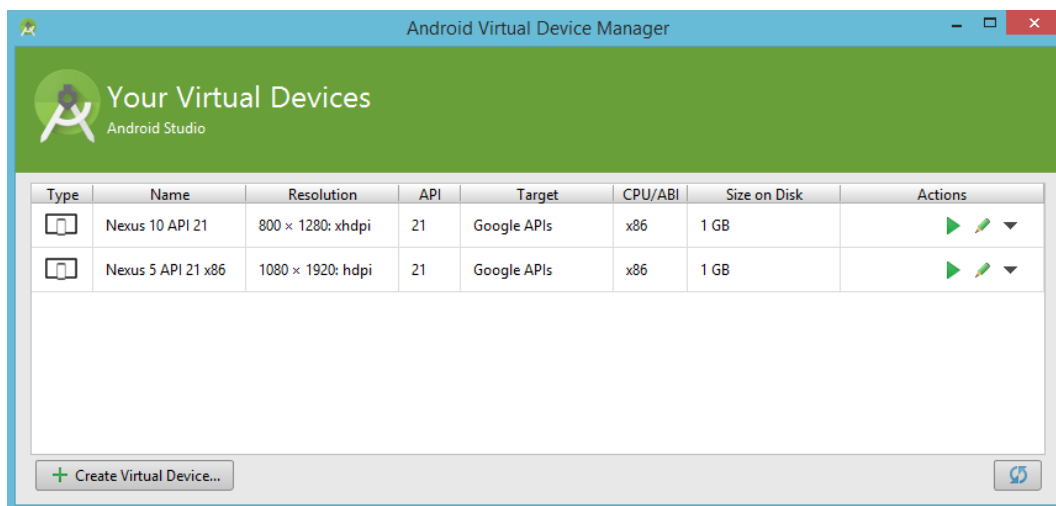
Na telefonu Google Nexus S (viz Obrázek 5.4) nenastaly žádné komplikace při testování. To se ovšem říct nedá o posledním z testovaných telefonů HTC One V (viz. Obrázek 5.3). Ze všech testovaných telefonů obsahuje nejstarší verzi Androidu 4.0.3 (API 15). Tvorba notifikací (Notifications) byla nejdříve implementována pomocí metod dostupných až od API 16. To mělo za následek pád aplikace v době, kdy měla být notifikace vytvořena. Tento problém byl vyřešen použitím starší metody ke tvorbě notifikací, která je dostupná již od API 11.



Obrázek 5.4: Google Nexus S

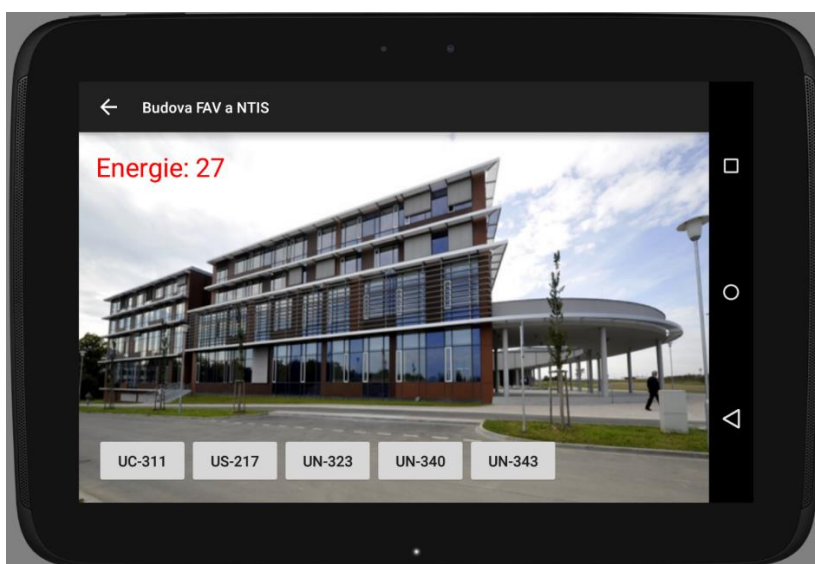
5.2 Emulátor

Součástí Android SDK je i mobilní emulátor zařízení. Který dokáže vytvořit plnohodnotné virtuální mobilní zařízení. Virtual Device Manager (viz. Obrázek 5.5) slouží k vytváření a editaci virtuálních zařízení. To nemusí být pouze mobilní telefony, ale také tablety, chytré hodinky a televize.



Obrázek 5.5: Android Virtual Device Manager

Jelikož jsem neměl možnost testování na fyzickém tabletu, použil jsem k tomuto účelu emulátor (viz Obrázek 5.6), ve kterém jsem si vytvořil tablet Nexus 10 s úhlopříčkou 10.06". I na takto velkém zařízení fungovala aplikace bez problémů a všechny lokace se nacházely na správném místě. Kvalita fotografií byla také stále velmi kvalitní, přestože jsou veškeré fotky zmenšeny na velikost do 200 kB z důvodu úspory datové velikosti celé aplikace.



Obrázek 5.6: Ukázka emulátoru – virtuální tablet

5.3 Distribuce aplikace

Na rozdíl od iOS Android umožňuje instalaci aplikací i z jiných zdrojů, než z výchozího obchodu s aplikacemi. Avšak tato možnost musí být povolena v nastavení telefonu. U telefonu Samsung Galaxy S4 je tato možnost v Nastavení → Další → Zabezpečení → Neznámé zdroje. Pokud je tato možnost povolena, je následně možné instalační soubor s příponou *.apk* stáhnout z libovolných webových stránek.

Další možnosti jsou neoficiální obchody s aplikacemi, které však mnoho uživatelů nevyužívá. Proto považují za nejlepší způsob rozšíření aplikace mezi uživatele její nahrání na primární úložiště pro Android – obchod Google Play Store. Ten je nastaven jako výchozí ve většině zařízeních s OS Android, kde má i svou klientskou aplikaci a je považován za nejbezpečnější. Umožňuje také filtrování aplikací podle každého uživatele tak, aby byla zajištěna kompatibilita s jeho zařízením. Nenabídne mu tedy aplikaci, která např. využívá NFC, pokud telefon technologii NFC neobsahuje. Nezobrazí také aplikace, které mají nastavenou vyšší minimální verzi API v souboru `AndroidManifest.xml`, než dané zařízení podporuje.

Je tedy zřejmé, že umístění aplikace na Google Play umožňuje největší rozšíření aplikace mezi veřejnost. K jejímu nahrání by bylo za potřeby zaregistrovat účet vývojáře, který je při registraci zpoplatněn částkou 25\$.

U vyvíjené aplikace lze předpokládat její budoucí umístění na Google Play.

6. Možná rozšíření

Ve vyvíjené hře by se dala vymyslet celá řada rozšíření, která by hru vždy o něco obohatila a přidala na realističnosti. V této kapitole uvedu některé z nich.

6.1 Více lokací a stavových proměnných

V současné podobě aplikace obsahuje 16 lokací z areálu ZČU. Jedná se o nejčastěji navštěvované místnosti v rozvrhu studenta Katedry informatiky. Obsahuje také zajímavá místa z nové budovy FAV, jako například neuroinfomatická laboratoř, anechoická komora, serverovna. V těchto lokacích se hráč dozví, k čemu se využívají. Takových místností by ve hře mohlo být více. Stejně tak více místností pro přednášky a cvičení.

Dalšími lokacemi ve hře by mohli být i prostory mimo areál ZČU na Borech. Například kolej, studentské hospody, fotbalové a hokejové hřiště. V těchto lokacích by hráč mohl doplňovat vedle proměnné „energie“ i další stavovou proměnnou „zábava“.

6.2 Rozvrh předmětů

Přidáním více předmětů, by aplikace mohla uchazeči o studium zobrazit reálný počet a skladbu předmětů v konkrétním studijním programu. Potencionální uchazeč by tedy díky aplikaci přesněji věděl, které předměty ho při studiu čekají. Předměty by mohly být zobrazeny v rozvrhu, podle kterého by je postupně navštěvoval.

6.3 Rozšíření textů ve hře

Ve hře se vyskytuje mnoho textů, které se uživatelům v průběhu hry zobrazují. V následující kapitole jsou popsány možnosti jejich rozšíření a editace.

6.3.1 Internacionalizace

Veškeré texty se nachází v adresáři `/res/values/strings`. Zde by i při případné jiné jazykové verzi stačilo vytvořit například soubor `strings-en.xml`, ve kterém by v elementu byl stejný atribut "name", ale text by byl anglicky.

Takto by vypadal řádek v souboru `strings-cs.xml`:

```
<string name="restart_game">Zkusit to znovu</string>
```

A takto v souboru `strings-en.xml`:

```
<string name="restart_game">Try the game again</string>
```

Operační systém Android už se postará o použití správného souboru, který vybere na základě zvoleného jazyka v nastavení telefonu.

6.3.2 Přidání zkouškových otázek

V souboru `ExamActivity.java` jsou v `ArrayListu` obsaženy veškeré otázky ke zkoušce. V případě potřeby rozšíření otázek, by přidání vypadalo následovně:

```
questionList.add(new Otazky("Text otázky",  
"odpověď1", odpověď2", "odpověď3", cisloSpravneOdpovedi));
```

Texty otázek i jednotlivých odpovědí jsou uloženy v `.xml` souboru `/res/values/strings`. Lepším řešením vkládání do `ArrayListu` je tedy pomocí odkazu na text do tohoto souboru `strings.xml` pomocí metody `getString()`:

```
getString(R.string.question_Math1_answer0)
```

Otázky jsou rozděleny podle předmětů do metod, které naplňují ArrayList:

- createQuestionListMath – otázky z Matematiky

- createQuestionListPPA – otázky z PPA1

6.4 Grafické úpravy

Veškeré fotky, které jsou použity pro pozadí k jednotlivým lokacím, se nachází v adresáři /res/drawable-hdpi. Pokud by došlo ke změně v dané lokaci, nebo by bylo z jiného důvodu potřeba některý obrázek nahradit, stačí pouze nahrát nový obrázek pod stejným názvem a nahradit ho. Z důvodu redukce velikosti instalačního souboru jsem všechny fotky zmenšil a dosáhl tak velikosti u každé fotky pod 200 kB. Všechny použité fotky (21) zabírají tedy méně než 3 MB.

6.5 Platforma iOS

Podobná hra, zmíněná v kapitole 2.3.1, již existuje pro platformu Windows Phone. Pro úplné pokrytí všech tří nejpoužívanějších mobilních platforem zbývá ještě implementace pro operační systém iOS mobilních telefonů společnosti Apple Inc.

7. Závěr

Cílem bakalářské práce bylo navrhnout a vytvořit aplikaci pro platformu Android v jazyce Java, která bude vhodným způsobem propagovat Fakultu aplikovaných věd v Plzni. Nejprve byla zvažena možná interakce s uživatelem a dále následoval návrh scénáře pro hru. Poté bylo potřeba se seznámit s vývojem aplikací pro mobilní zařízení s operačním systémem Android a vybrat vhodné prvky, které tento operační systém nabízí.

Výsledkem je herní aplikace, která seznamuje potencionální uchazeče o studium s vybranými prostory univerzity a simuluje průběh studia na vysoké škole. Dále jim přibližuje některé vysokoškolské pojmy a aktivity, se kterými se na střední škole nesetkali. Velký důraz byl také kladen na testování aplikace, ve kterém byly úspěšně odstraněny všechny problémy, které v průběhu testování nastaly. V poslední kapitole jsou zmíněny vhodná rozšíření, kterými by se dala aplikace obohatit.

Mimo herní vlastnosti aplikace slouží i jako informační zdroj, díky kterému lze zobrazit aktuální informace o fakultě, které se stahují z webu. Získává tím aktuálnost a stává se užitečná i po dohrání herního příběhu. Vzhledem k rozšířenosti mobilních zařízení s operačním systémem Android má aplikace velkou šanci se rozšířit mezi studenty středních škol při propagaci fakulty na dnu otevřených dveří, kteří mohou být následně prostřednictvím aplikace informováni o dalších aktivitách FAV.

Přehled zkratk

ADT – *Android Development Tools*

API – *Application Programming Interface* - rozhraní pro vývoj aplikací

DPI – *Dots per inch* – počet obrazových bodů na jeden palec

IDE – *Integrated Development Environment* – prostředí pro vývoj aplikací

iOS – mobilní operační systém vytvořený společností Apple Inc.

LIFO – *Last In – First Out* – způsob ukládání dat v zásobníku

NFC – *Near Field Communication* – technologie bezdrátové komunikace

OS – *Operating System*

RAM – *Random-Access Memory* – paměť s přímým přístupem

SDK – *Software Development Kit* – soubor nástrojů pro vývoj software

XML – *Extensible Markup Language* – obecný značkovací jazyk

Literatura

- [1] Dashboards. Android Developers [online]. 2013 [cit. 2013-01-17]. Dostupné z: <http://developer.android.com/about/dashboards/index.html>
- [2] UJBÁNYAI, Miroslav. *Programuje pro Android* Grada, Publishing, a.s., 2012. ISBN 978-80-247-3995-3
- [3] Gradle [online] 2015 [cit. 2015-03-24]. Dostupné z: <https://gradle.org/getting-started-android/>
- [4] Gradle, moderní nástroj na automatizaci [online] 2013 [cit. 2015-03-06]. Dostupné z: <http://www.zdrojak.cz/clanky/gradle-moderni-nastroj-na-automatizaci/>
- [5] Senzory mobilních zařízení [online] 2013 [cit. 2015-04-02]. Dostupné z: <http://mobilenet.cz/clanky/techbox-vas-telefon-je-prospikovany-senzory-12496>
- [6] Android Studio Overview [online] 2015 [cit. 2015-04-05]. Dostupné z: <http://developer.android.com/tools/studio/index.html>
- [7] DiMarzio, J.F., *Programujeme hry pro Android 4*, Computer Press, 2012. ISBN 978-80-251-3754-3
- [8] Activity [online] 2015 [cit. 2015-27-3]. Dostupné z: <http://developer.android.com/reference/android/app/Activity.html>
- [9] Supporting Multiple Screens [online] 2015 [cit. 2015-03-29]. Dostupné z: http://developer.android.com/guide/practices/screens_support.html
- [10] Layouts [online] 2015 [cit. 2015-03-29]. Dostupné z: <http://developer.android.com/guide/topics/ui/declaring-layout.html>

- [11] Notifications [online] 2015 [cit. 2015-03-29]. Dostupné z:
<http://developer.android.com/guide/topics/ui/notifiers/notifications.html>

- [12] MEIER, Reto. *Professional Android application development*. Wiley Publishing, Inc., 2009. ISBN 978-0-470-34471-2

- [13] GRANT, Allen. *Android 4 – Průvodce programováním mobilních aplikací*. Computer Press, 2013. ISBN 978-80-251-3782-6

- [14] Android 5.0 APIs [online] 2015 [cit. 2015-04-28]. Dostupné z:
<http://developer.android.com/about/versions/android-5.0.html>

- [15] KOHOUT, František, *Bakalářská práce: FAVagotchi Real-timeová hra pro Windows Phone*, 2014

- [16] Smartphone OS Market Share [online] 2015 [cit. 2015-05-01] Dostupné z:
<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>

- [17] Intent [online] 2015 [cit. 2015-05-05] Dostupné z:
<http://developer.android.com/reference/android/content/Intent.html>

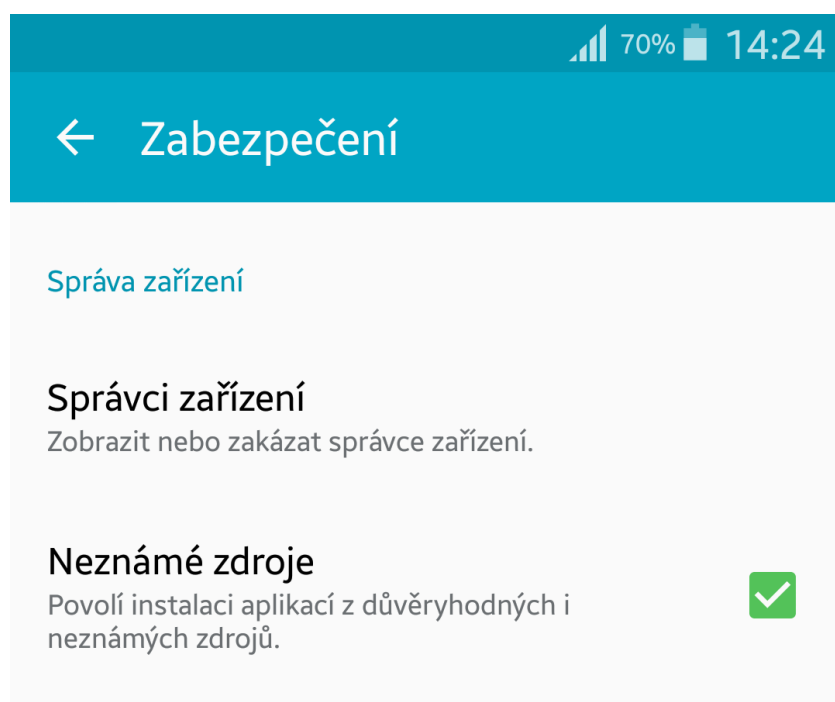
Přílohy

- Příloha A: Instalační příručka
- Příloha B: Uživatelská příručka
- Příloha C: Obsah CD

A Instalační příručka

Jelikož se nejedná o instalaci z oficiálního obchodu s aplikacemi Google Play, je nejprve potřeba povolit v zařízení instalaci z neznámých zdrojů. Cesta k této možnosti je následující:

NASTAVENÍ > ZABEZPEČENÍ > NEZNÁMÉ ZDROJE (viz Obrázek A.1).



Obrázek A.1: Povolení instalace z neznámých zdrojů

Poté nahrajte do zařízení instalační balíček s příponou .apk. Poklepání na tento soubor se spustí instalace, která si nejprve vyžádá potvrzení práv, které aplikace vyžaduje. Jsou-li tato práva potvrzena, aplikace se nainstaluje a je připravena ke spuštění. Její ikona se zobrazí v seznamu aplikací.

Minimální verze systému Android pro tuto aplikaci je Android 3.0 Honeycomb (API 11).

B Uživatelská příručka

V této kapitole jsou popsány základní prvky hry a vysvětleno jak hru ovládat.

B.1 Ovládání aplikace

Po spuštění hry se nejprve zobrazí úvodní obrazovka (viz Obrázek B.1), která obsahuje tlačítka a zaškrťovací pole pro obsluhu hry. Po stisknutí tlačítka *Spustit hru* dojde k spuštění hry. Jedná-li se o historicky první spuštění, zobrazí se nápověda, ve které je stručně popsána podstata hry. Pokud hráč chce kdykoliv začít hru od začátku, má k dispozici druhé tlačítko, kterým vymaže všechn dosavadní postup ve hře. Dalším tlačítkem je nápověda ke hře, ve které si hráč může i v průběhu hry přečíst informace o tom, jak hrát a na co si během hry dávat pozor. Dále je zde letecký snímek areálu ZČU, ve kterém jsou jednotlivé budovy popsány. Při samotné hře již tyto popisky nejsou. Posledním tlačítkem je tlačítko *Konec*, které slouží k ukončení aplikace. Na této úvodní obrazovce se nachází ještě dvě zaškrťovací pole, umožňující povolení/zakázání zvukových efektů a hudby ve hře.



Obrázek B.1: Úvodní obrazovka

Po spuštění hry se zobrazí obrazovka s univerzitním kampusem (viz Obrázek B.2). Dotykem na jednotlivé budovy lze navštěvovat lokace ve hře, které reprezentují danou budovu. Na této obrazovce se v levém dolním rohu nachází tlačítko pro návrat na úvodní obrazovku a v pravém dolním rohu nachází tlačítko *ÚKOLY A PENĚŽENKA*. Po jeho stisknutí se zobrazí obrazovka (viz Obrázek B.3), ve které je vždy zobrazen aktuální úkol a stav peněz, za které lze kupovat oběd v budově menzy. Poté se zde nachází tlačítko *STÁHNI AKTUÁLNÍ INFO O FAV*, které načítá z webu aktuální informaci o fakultě a zobrazuje ji na této obrazovce. K této možnosti je potřeba být připojen k internetu.

Úkoly ve hře by bylo možné rozdělit do následujících skupin:

Rozvrhové události: (přednášky a cvičení) při kterých je potřeba najít správnou učebnu a poté vyčkat do konce dané události. Odpočet času pokračuje i po opuštění aplikace a po konci odpočtu uživatele oznámí v oznamovacím panelu. (Viz Obrázek B.11)

Návštěva zajímavých prostor: opět je potřeba vyhledat správnou místnost, ve které je následně zobrazen text s informací, k čemu daná místnost slouží a co se v ní odehrává. (Viz Obrázky B.12 a B.13)

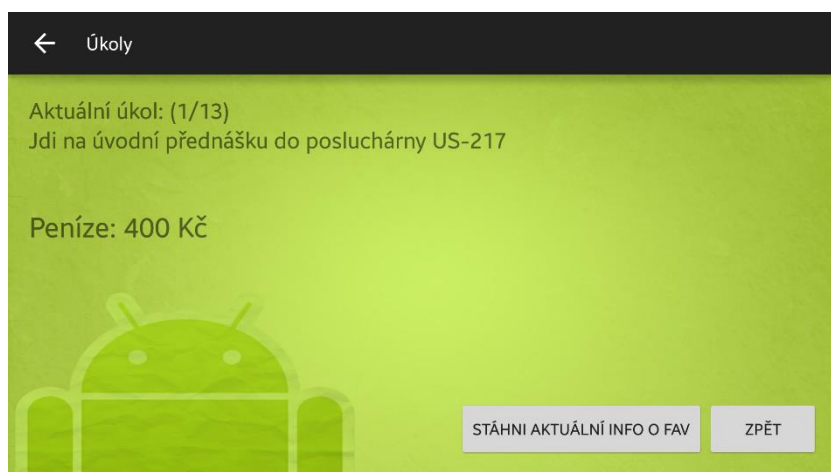
Zkouška: obsahuje sadu otázek, ke kterým jsou vždy na výběr tři možnosti, z nichž vždy pouze jedna je správná. K úspěšnému složení zkoušky je potřeba správně odpovědět minimálně tři otázky z pěti. Otázka obsahuje vždy jen jednu správnou odpověď a na zkoušku jsou tři pokusy. Pokud ani po třetím pokusu není zkouška úspěšně absolvována, nastává konec hry. (Viz obrázek B.10)

Další lokací je budova menzy (viz Obrázek B.5), která již není součástí úkolů, ale je potřeba ji v průběhu hry navštěvovat a doplňovat v ní formou oběda energii. Klesne-li hodnota energie pod nulu, postava FAVáka umírá vyčerpáním.

B.2 Náhledy aplikace



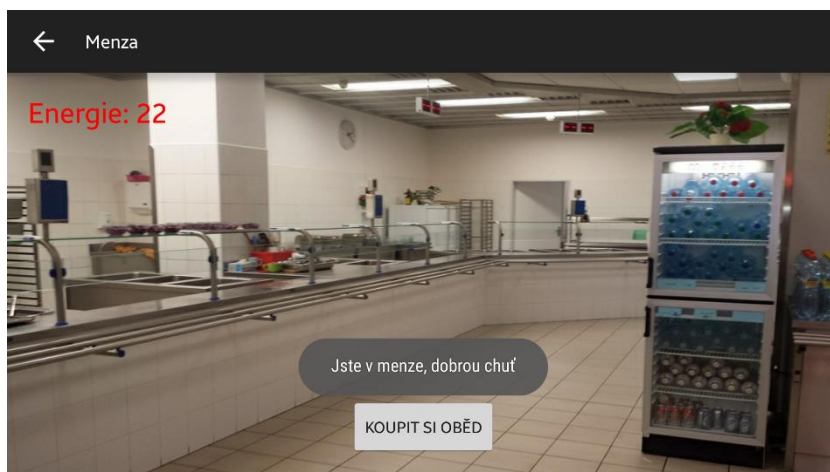
Obrázek B.2: Areál ZČU



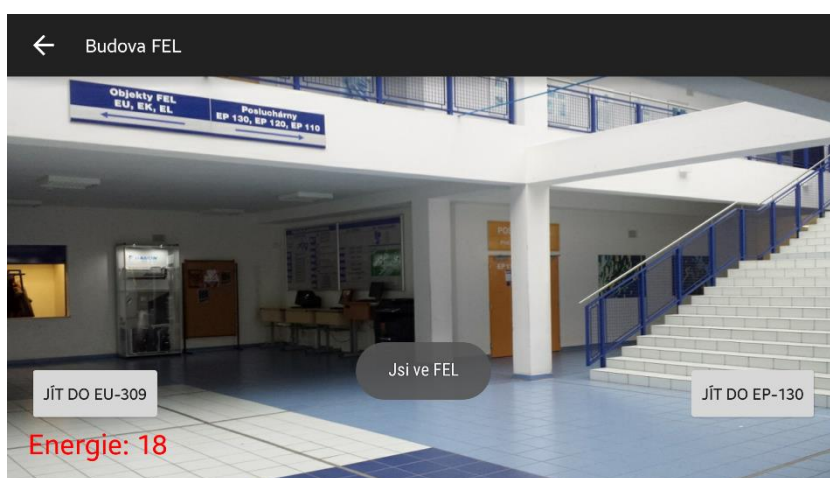
Obrázek B.3: Náhled obrazovky s úkoly



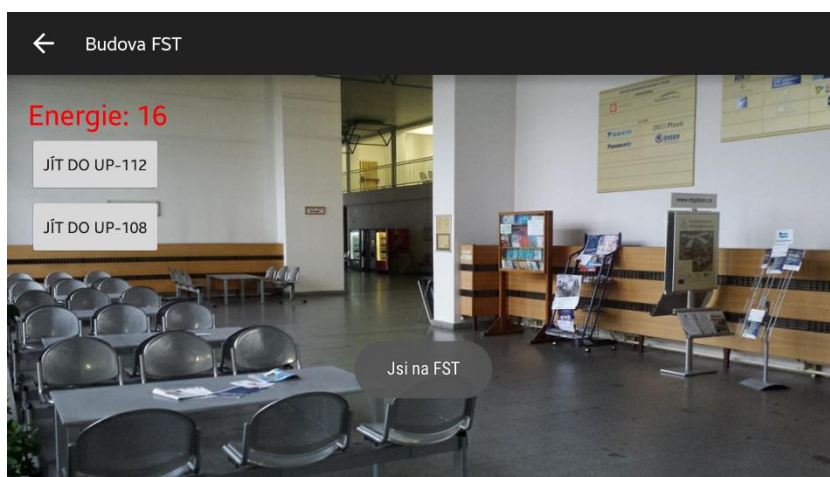
Obrázek B.4: Budova FAV a NTIS



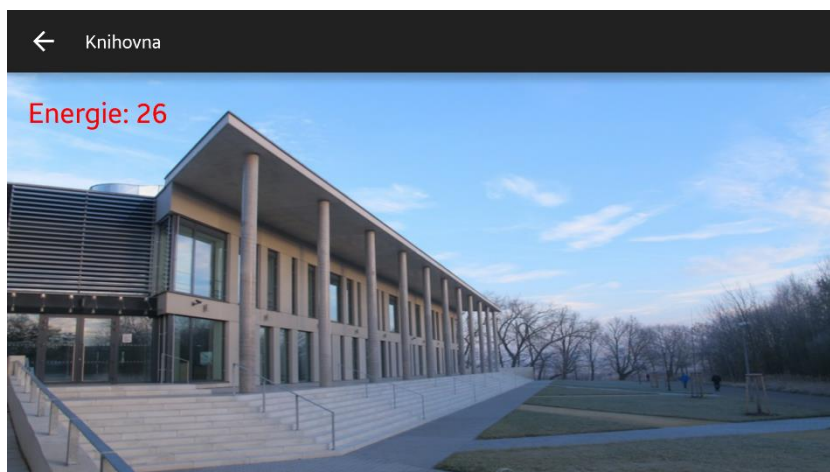
Obrázek B.5: Menza



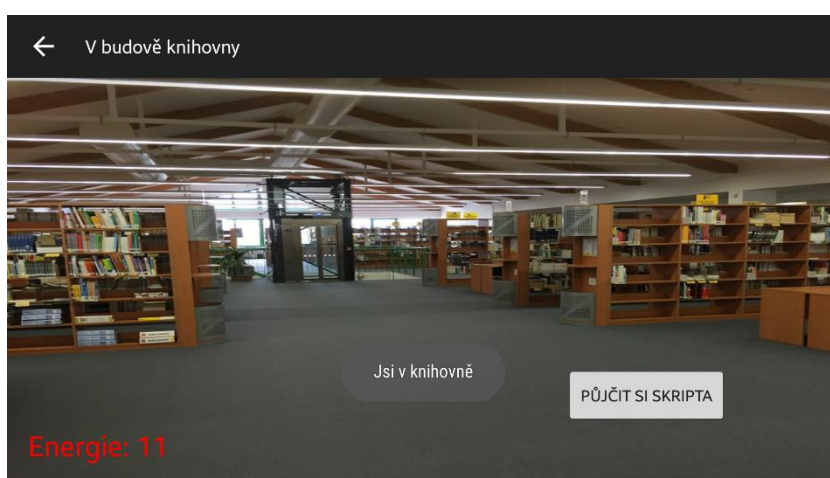
Obrázek B.6: Budova FEL



Obrázek B.7: Budova FST



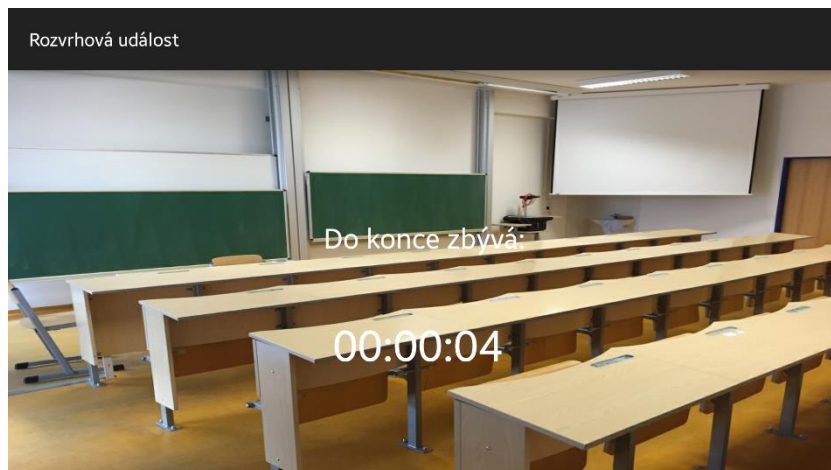
Obrázek B.8: Budova knihovny



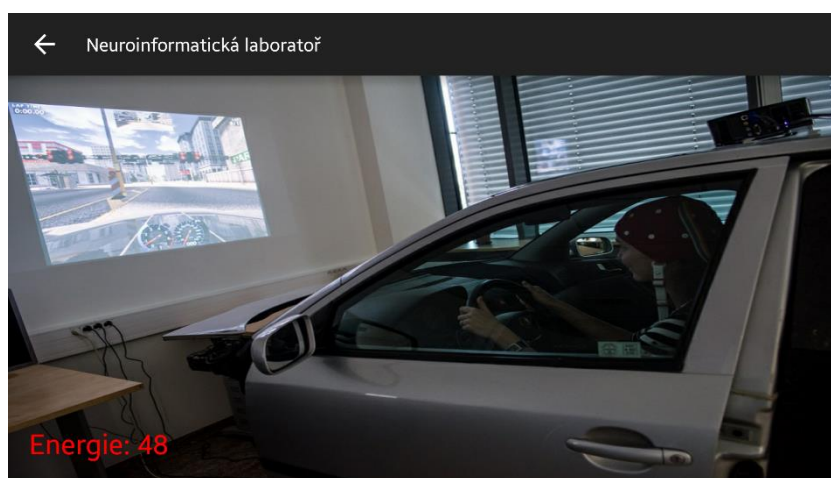
Obrázek B.9: Knihovna uvnitř



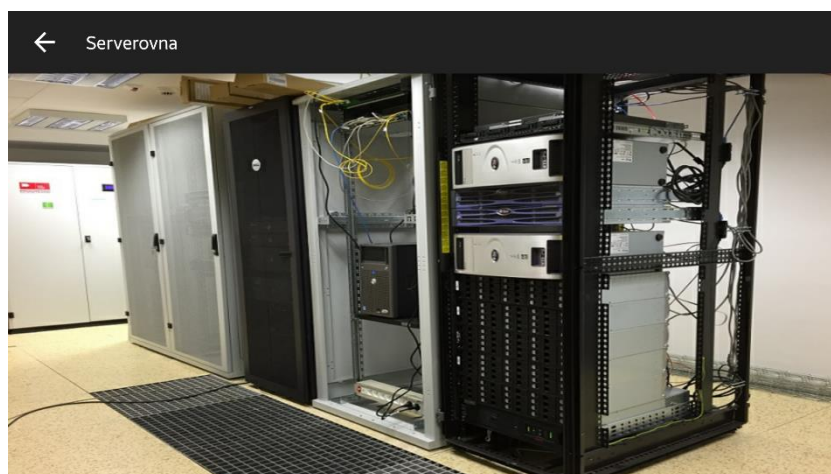
Obrázek B.10: Náhled zkuškové otázky



Obrázek B11: Cvičení v EU-309



Obrázek B.12: Neuroinformatická laboratoř



Obrázek B.13: Serverovna

C Obsah CD

K práci je přiloženo CD s následující strukturou:

- **adresář: *aplikace/*** - instalační balíček programu, zdrojové kódy aplikace

V této složce se také nachází celý projekt pro Android Studio. Instalační balíčky s příponou .apk se zde nachází ve dvou verzích. První (FAVagoči.apk) obsahuje plnohodnotnou aplikaci pro uživatele. Druhý (app-debug.apk) slouží k rychlejšímu testování aplikace, jsou v něm proto zkráceny časy pro rozvrhové události na 1 minutu.

- **adresář: *dokument/*** - text práce v elektronické podobě