

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Systémové inženýrství a informatika



Bakalářská práce

Vývoj hry Pairs (pexeso) pro Android tablety

Autor práce

Ketevan Kharashvili

2016 ČZU v Praze

Ráda bych tímto poděkovala vedoucímu mé bakalářské práce panu Ing. Josefu Pavlíčkovi, Ph.D. za jeho rady a čas, který mi věnoval při řešení různých úskalí, spojených s touto prací. V neposlední řadě také děkuji všem ostatním, kteří mi poskytli potřebné informace.

Prohlašuji, že jsem bakalářskou práci na téma „Vývoj hry Pairs (pexeso) pro Android tablety“ vypracovala pod vedením vedoucího bakalářské práce samostatně za použití v práci uvedených pramenů a literatury.

Prohlašuji, že odevzdaná elektronická verze bakalářské práce je identická s její tištěnou podobou.

Praha, 14.3 2016

.....

vlastnoruční podpis

Abstrakt

Tato bakalářská práce se zabývá problematikou „Vývoj hry Pairs (pexeso) pro Android tablety“ a je zaměřena na vývoj softwaru pro Android. V teoretické části popisuje operační systém Android, včetně jeho architektury a vydaných verzí, pokračuje popisem aplikačního rozhraní nezbytného pro vývoj hry pro tento systém a zakončuje nástroji důležitými pro vývoj. Mezi nástroji důležitými pro vývoj jsou uvedeny Android SDK Manager, který je potřebný pro správu nainstalovaných verzí aplikačního rozhraní na počítači a dále jednotlivá vývojová prostředí, včetně podmínek jejich užívání. Následuje přiblížení objektově orientovaného programovacího jazyka Java, ve kterém je hra v této bakalářské práci vytvořena. Důležité nástroje zakončující teoretickou část je Google Play, který slouží pro zveřejnění aplikace a její distribuci mezi uživatele.

V praktické části se popisuje samotná hra a vysvětluje programování her po samostatných krocích. Dále podrobně ukazuje detaily některých použitých prvků a poukazuje i na použité kódy. Konkrétně je to hra Pexeso.

Klíčová slova:

Android, Aplikace, Programování, Vývoj, Tablet, Android Studio, Java, Pexeso.

Abstract

This bachelor thesis deals with The pairs game development for Android tablets. In theoretical part it describes the operating system Android in its basics including its architecture and released versions, continuing with its application interface, which is necessary for the development of applications for this system, and concludes with the description of tools that are vital for the application creation process. Among the mentioned tools it explains the Android SDK Manager, which is needed for the management of installed versions of application interface on computers, as well as respective development environments including their terms of use. It follows with the introduction of object base programming language Java, which was used to create an application for this thesis. Other described tool is Google Play, which is used for application publishing and distribution among users.

Practical part illustrate the process of development of ealier mentioned game and gives step by step explanation of the game programming. It also shows details of some elements that were applied and points out on some used codes. Specificaly the game is Pairs.

Keywords:

Android, Application, Programming, Development, Tablet, Android Studio, Java, Pairs.

Obsah

1. Úvod	3
2. Cíl a metodika práce	3
2.1 Cíl práce.	3
2.2 Metodika práce.	3
3. Teoretická východiska	4
3.1 Operační systém Android	4
3.1.1 Architektura Android	5
3.1.2 Verze OS Android	6
3.1.3 Android přehled používaných verzí.....	14
3.2 Vývoj aplikace pro operační systém Android.	15
3.2.1 Android Studio.....	15
3.2.2 Programovací jazyk Java	15
3.2.3 Program Eclipse	16
3.2.4 Android SDK Manager	17
3.2.5 App Inventor	18
3.2.6 Xamarin	19
3.2.7 Microsoft Visual Studio.....	20
3.3 Nástroje důležité pro vývoj	21
3.3.1 Android Virtual Device Manager (AVD Manager).....	21
3.3.2 ADT Bundle.....	22
3.4 Pexeso	23
3.4.1 Historie.....	23
3.4.2 Pravidla	23
4. Praktická část práce	24
4.1 Založení projektu v Android Studio	24
4.2 Stručný návrh aplikace	24
4.3 Layouty	28
4.4.1 Linear Layout.....	30
4.4.2 Relative Layout.....	30
4.6 Tlačítka – Buttons	33
4.6.1 Tlačítko s popisem	33
4.6.2 Tlačítko s ikonou	34
4.6.3 Tlačítko s ikonou i popisem.....	34

4.7 Activity	35
4.7.1 Layout pro Activity y Activity v programovacím jazyce Java	35
4.8 Komentáře (comments)	36
4.9 Kontrola hry	36
5. Závěr	37
6. Seznam použitých zdrojů	38
7. Seznam obrázků	40
8. Seznam tabulek	40

1. Úvod

Téma bakalářské práce „**Vývoj hry Pairs (pexeso) pro Android tablety**“ bylo vybráno záměrně. Hlavním důvodem je zájem autora této práce o mobilní aplikace pro Android.

Toto téma je aktuální především z důvodu dnešního rozmachu světa mobilních her.

Vývoj hry Pairs zdá být pro Android tablety velmi zajímavým a v této době také přínosným.

Cílem této bakalářské práce je přiblížení vývoje hry pro Android tablety a také osvětlit potřebná teoretická východiska bez kterých by bylo nemožné hru vyvíjet.

Vysvětlen bude postup navržení a vytvoření mobilní hry včetně jejího otestování a nahrání na cílové Android tablety.

2. Cíl a metodika práce.

2.1 Cíl práce.

Bakalářská práce je tematicky zaměřena na problematiku vývoje her pro Android tablety. Práce se skládá z následujících dílčích cílů: popis architektury platformy Android a vývojových nástrojů, tvorba aplikace PEXESO v programovacím jazyce Java, distribuce výsledné aplikace, připojení aplikace na síť Athena.pef.czu.cz.

2.2 Metodika práce.

Rešeršní část bakalářské práce bude založena na analýze odborných a vědeckých dokumentů (časopisů a monografií). Následně budou získané poznatky synteticky využity k návrhové části tvorby aplikace pexeso. Pro tu bude navržena vhodná technologická varianta pro vývoj a testování. Aplikace bude možné provozovat na přenosných (mobilních) zařízeních. Tato aplikace též poslouží jako vzorový příklad vývoje mobilních aplikací. Řešení bude otestováno a na základě výsledků analýzy provedeného testu autor zformuluje závěry. Výsledná doporučení pak bude možné využít pro vývoj podobných aplikací.

3. Teoretická východiska

3.1 Operační systém Android

Platforma Android byla ohlášena Googlem 5. listopadu 2007, když se během tiskové konference odtajnil projekt otevřeného operačního systému postaveném na Linuxovém jádře, který by byl určený především pro chytré telefony, navigace a jiná mobilní zařízení. **Google** předal celou platformu i zdrojové kódy nově založené firmě Open Handset Alliance (sdružení mobilních operátorů, obchodníků, softwarových vývojářů a výrobců mobilních zařízení), která nedlouho poté uvolnila Android jako open source, kompletní zdrojový kód nově narozeného OS se tak dostal do rukou široké veřejnosti.

*„Cíl platformy Android je být maximálně otevřenou vývojářům, průmyslu i uživatelům. Uživatelé nepotřebují svolení k tomu, aby si nainstalovali libovolnou aplikaci. Výrobci telefonů ani mobilní operátoři nebudou potřebovat k nasazení Androida na svá zařízení žádné svolení od Googlu či kohokoliv z **Open Handset Alliance**. Mohou si upravit Androida jak sami uznají za vhodné a výhodné pro svůj business.“ – Robert Love*

Právě filozofie „open source“, kterou se Android ostře vymezil proti konkurenčním prostředím od firem Apple, Microsoft nebo Nokia, je jedním z mnoha důvodů jeho raketového rozšíření mezi uživateli i vývojáři.

I když logo a symbol platformy Android – roztomilý zelený robůtek – zůstává stále stejné, bylo by omylem myslet si, že pod povrchem neprobíhají závratné změny. To ostatně dokazují i různé verze OS Android, které se mimo číslo označují také jmény oblíbených amerických sladkostí. První zásadní aktualizací androidího novorozence byla verze **1.5 Cupcake. Android 1.6 Donut**, který v té době přinesl revoluční podporu WVGA rozlišení displeje, pozměněné prostředí fotoaparátu a galerie, takže se telefony s Androidem staly velmi zajímavou alternativou také pro příznivce mobilního fotografování a vůbec práce s multimédií. Zásadní proměnu prodělal Android při svém přečíslování na druhou řadu, verze **2.2 Froyo** je stále rozšířená a populární mezi běžnými uživateli. Další verze byla **2.3 Gingerbread**, který dnes najdeme takřka v polovině všech zařízeních běžících na Androidu. Hudbou budoucnosti je **Android 4.0**, který na konci roku 2011 představil Google s jihokorejským gigantem Samsung. Přídomek první verze čtvrté řady

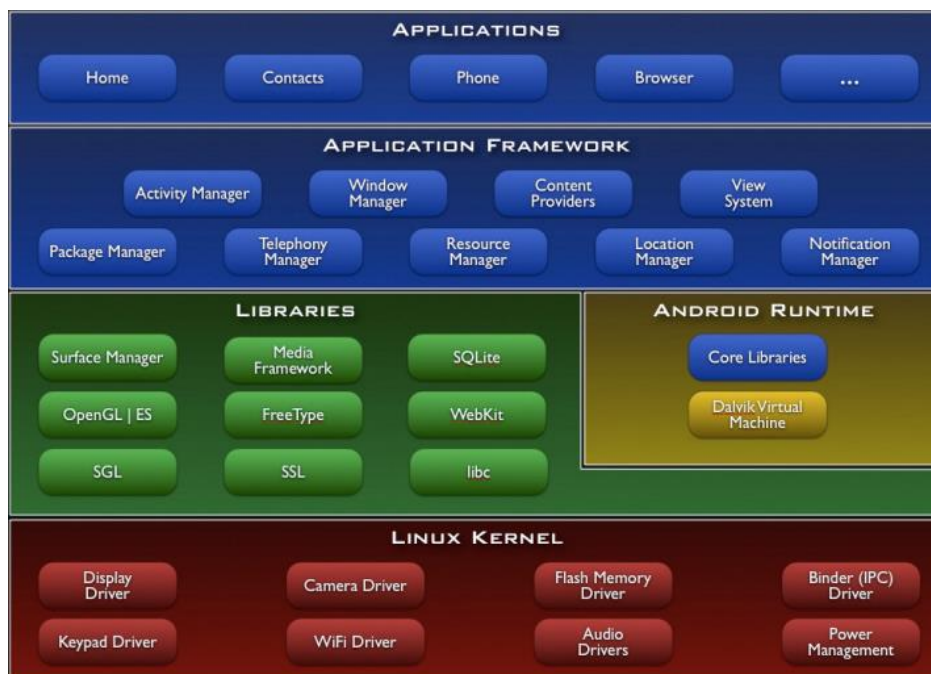
Androidu je **Ice Cream Sandwich** a jako jediná verze Androidu je nativně určený pro mobilní telefony i tablety. ¹

3.1.1 Architektura Android

Architektura operačního systému Android je rozdělen do pěti vrstev: jádro, knihovny, aplikační framework, běhové prostředí a aplikace. Každá vrstva má svůj účel a nemusí být přímo oddělena od statních vrstev.

Nejnižší vrstvou architektury je jádro operačního systému, které tvoří abstraktní vrstvu mezi používaným hardwarem a zbytkem softwaru ve vyšších vrstvách. Jádro mobilního operačního systému Androidu je postaveno na Linuxu ve verzi 2.6. Využívá celé řady jeho vlastností, např. podpory správy paměti, správy sítí, zabudované ovladače nebo správy procesů, např. souběžného běhu aplikací, které běží jako samostatné procesy s oprávněním definovaným systémem, což přispívá ke stabilitě a také zabezpečení systému. Naopak systém Android nepodporuje grafické uživatelské rozhraní X Window System a ani úplnou sadu GNU knihoven.

Důvodem použití jádra systému Linux byla také vlastnost poměrně snadného sestavení na různých zařízeních a zaručená přenositelnost.²



Obrázek 1- Architektura Android

¹ <http://androidaplikace.cz/index.php/co-je-operacni-system-android/>

² <http://www.elitecsoftware.cz/vyvoj-pro-android/>

3.1.2 Verze OS Android

Od první verze bylo vydáno několik aktualizací, které opravují chyby a přidávají novou funkcionalitu. Jednotlivé verze systému jsou označovány podle zákusků (Cupcake, Donut, Eclair, Froyo, Gingerbread).

3.1.2.1 Android 1.0.

API level – 1.

Linuxové jádro 2.6.25.

Táto verze byla představena 23. září 2008. Historie operačního systému Android začala první instalací, která běžela na telefonu HTC Dream a byla závislá na hardwaru tohoto zařízení, vybaveného celou řadou hardwarových tlačítek.

Funkce:

- ❖ Podpora notifikací
- ❖ Synchronizace s poštovní službou Gmail
- ❖ Widgety na domovské obrazovce
- ❖ Repozitář aplikací Android Market
- ❖ Plnohodnotný webový prohlížeč
- ❖ Podpora fotoaparátu
- ❖ Seskupování aplikací do složek
- ❖ Přístup k webovým e-mailovým serverům
- ❖ Synchronizace kalendáře a kontaktů s Googlem
- ❖ Mapy Google s pohledem z ulice StreetView
- ❖ Vyhledávání Google
- ❖ Komunikační aplikace Google Talk
- ❖ Aplikace pro komunikaci skrze SMS a MMS
- ❖ Multimediální přehrávač
- ❖ Hlasové vytáčení
- ❖ Možnost nastavení tapety na domovskou obrazovku
- ❖ Přehrávač videí z portálu YouTube
- ❖ Další aplikace jako budík, kalkulačka, Telefon, launcher, galerie a nastavení
- ❖ Podpora Wi-Fi a Bluetooth

3.1.2.3 Android 1.1.

API level – 2.

Linuxové jádro 2.6.25.

Verze byla představena 9. února 2009. Historie pokračovala a jen několik měsíců po uvedení první verze byl vypuštěn update, který řešil především nalezené chyby. Již tak dlouhý seznam možností a schopností se také rozrostl o další položky.

Funkce:

- ❖ Informace o firmách a podnicích v mapách
- ❖ Další interval vypnutí obrazovky při příchodím hovoru
- ❖ Možnost ukládání příloh ze zpráv
- ❖ Zobrazení a skrytí číselníku³

3.1.2.4 Android 1.5 (Cupcake)

API level – 3.

Linuxové jádro 2.6.27

30.04.2009 byla uvolněna aktualizace na Android 1.5 (Cupcake). Tento update přidává několik nových funkcí“

- ❖ Možnost nahrávat a sledovat videa z kamery
- ❖ Nahrávání videí na YouTube a fotografií na Picasu přímo z telefonu
- ❖ Nová softwarová klávesnice s automatickým dokončováním slov
- ❖ Bluetooth - podpora A2DP
- ❖ Možnost automaticky připojit Bluetooth headset
- ❖ Nové widgety a složky
- ❖ Animace při přechodu mezi obrazovkami
- ❖ Rozšířena funkce kopírovat a vložit⁴

³ <http://www.svetandroida.cz/historie-androidu-201506>

⁴ <http://developer.android.com/about/versions/android-1.5-highlights.html>

3.1.2.5 Android 1.6 (Donut)

API level – 4.

Verze uvolněna 15. září 2009 1.6 (Donut) SDK.Změny:

- ❖ Vylepšený Android Market
- ❖ Nové prostředí fotoaparátu, kamery a galerie
- ❖ Galerie umožňuje označit více fotografií k vymazání
- ❖ Aktualizované vyhledávání hlasem
- ❖ Quick Search Box - umožňuje vyhledávat záložky, historii a kontakty na webu z domovské obrazovky
- ❖ Podpora pro technologie CDMA/EV-DO, 802.1x, VPN, Gesta a syntéza řeči
- ❖ Podpora pro WVGA rozlišení displeje
- ❖ Vylepšení rychlosti vyhledávání a kamery⁵

3.1.2.6 Android 2.0/2.1 (Eclair).

API level – 5.

26. října 2009 bylo uvolněno 2.0 (Eclair) SDK, mezi změny patří:

- ❖ Optimalizována rychlost hardwaru
- ❖ Podpora pro více velikostí a rozlišení displeje
- ❖ Zdokonalené uživatelské prostředí
- ❖ Nové prostředí prohlížeče a podpora HTML5
- ❖ Nový seznam kontaktů
- ❖ Mapy Google aktualizovány na 3.1.2
- ❖ Podpora pro Microsoft Exchange
- ❖ Podpora přisvětlování diody
- ❖ Digitální zoom (fotoaparát)
- ❖ Vylepšená softwarová klávesnice
- ❖ Podpora pro Bluetooth 2.1
- ❖ Animované tepety na domovské stránce⁶

⁵ <http://developer.android.com/sdk/android-1.6-highlights.html>

⁶ <http://developer.android.com/sdk/android-2.0-highlights.html>

3.1.2.7 Android 2.2 (Froyo)

Linuxové jádro 2.6.32. API level – 6.

20. května 2010 na konferenci Google I/O byl představen Android 2.2. Přidává nové technologie a funkce uživatelského prostředí. Jedná se o zásadní upgrade, i když z čísla verze to není patrné:

- ❖ Možnost instalovat aplikace na paměťovou kartu
- ❖ Adobe vydalo plugin Adobe Flash 10.1. Není integrován do systému, distribuce je řešena přes Android Market nebo přes stránek Adobe
- ❖ Díky JIT (Just-in-time) kompilátoru se podařilo zvýšit rychlost systému na různých benchmarcích 2x až 5x. Dále je vylepšena správa paměti RAM
- ❖ Možnost vytvořit z telefonu WIFI hotspot nebo sdílet internetové připojení přes USB kabel
- ❖ Dva nové režimy telefonu – „car mode“ a „night mode“ (režim v autě a noční režim)
- ❖ Více nastavení fotoaparátu a kamery
- ❖ Přidává podporu pro OpenGL ES 2.0, vícebarevný trackball, vylepšena podpora pro Exchange, Bluetooth a přidána další vrstva vývojářského API⁷

3.1.2.8 Android 2.3/2.4 (Gingerbread)

Linuxové Jádro 2.6.35. API level – 9-10.

9.02.2011 byla vypuštěna verze Android 2.3 Gingerbread.

- ❖ Podpora video formátu WebM pro HTML5 video
- ❖ Podpora pro Near Field Communication standard, který dnes podporují některé mobilní telefony
- ❖ Podpora SIP protokolu pro internetovou telefonii
- ❖ Lepší správa prostředků
- ❖ Upravená virtuální klávesnice
- ❖ Zlepšená funkce kopírovat a vložit
- ❖ Podpora více kamer a nových sensorů

⁷ <http://developer.android.com/sdk/android-2.2-highlights.html>

- ❖ Nové Google Maps 5 s 3D přístupem (Dostupné přes market pro všechny verze androida)
- ❖ Rozšíření podpory nativního kodu

3.1.2.9 Android 3.0/3.1/3.2 (Honeycomb)

API level – 12-13.

22.02.2011

Verze pro tablety. Změny zahrnují:

- ❖ Optimalizaci pro velké obrázkové tablety
- ❖ Mnoho nových prvků uživatelského rozhraní
- ❖ Podpora více jádrových procesorů
- ❖ Hardwarová akcelerace pro grafiku
- ❖ Přístup ke Google eBook⁸

3.1.2.10 Android 4.0/4.0.1/4.0.2 (Ice Cream Sandwich).

API level – 15.

16.12 2011 byla představena verze 4.0, která přináší funkcionalitu verze Honeycomb do chytrých telefonů. Mezi nejzajímavější novinky této verze patří:

- ❖ Nové bezpečnostní funkce
- ❖ Rozpoznávání obličeje
- ❖ Sjednocené kontakty sociálních sítí
- ❖ Offline vyhledávání v emailu
- ❖ Sdílení informací pomocí NFC⁹

⁸ <http://developer.android.com/sdk/android-3.1-highlights.html>

⁹ <http://developer.android.com/sdk/android-4.0-highlights.html>

3.1.2.11 Android 4.2/4.2.2/4.3 (Jelly Bean).

API level – 16-18.

Představen 9 července 2012 na Google I/O.

- ❖ Project Butter
- ❖ Rozpoznávání hlasu offline
- ❖ Vylepšena aplikace fotoaparátu
- ❖ Vylepšena informační lišta
- ❖ Google Now¹⁰

3.1.2.12 Android 4.4 (Kitkat)

API level – 19.

Byl představen 24. 10. 2013. Společnost Google před nedávnem všechny šokovala, když změnila název nové chystané verze operačního systému Android z Key Lime Pie (citronový dort) na KitKat. John Langerling z Google to jednoduše obhájil slovy, že “Jen málo lidí zná chuť citronového dortu. Nyní je systém Android 4.4 na světě a nám nezbyvá nic jiného, než potvrdit, že se jedná o nejvýznamnější update za několik posledních let.”

- ❖ Vylepšená technologie NFC s podporou Host Card Emulation
- ❖ Nové uživatelské rozhraní, průhledné notifikační lišty, větší ikony
- ❖ Zamykací obrazovka s přidanými funkcemi, režim celé obrazovky
- ❖ Hlasové vyhledávání a vládání OK Google
- ❖ Tisk na dálku přes Wi-Fi
- ❖ Podpora titulků při přehrávání videa v aplikacích
- ❖ Podpora příslušenství (krokoměr)
- ❖ Nativní snímání senzorů – šetření baterie
- ❖ Handouts obsahuje SMS, MMS, IM zprávy a videohovory
- ❖ Záznam obrazovky v podobě videa do formátu MP4¹¹

¹⁰ <http://home.zcu.cz/~hodlova/html/vyvoj.html>

¹¹ <http://cdr.cz/clanek/os-android-44-kitkat>

3.1.2.13 Android 5.0/5.1 (Lollipop)

Linuxové jádro 3.4.0. API level – 22.

Byl představen 21.04.2014. Zatím posledním bodem historie je Android Lollipop, který přinesl další změnu designu – alfou a omegou všeho se stává vzhled ve stylu „Material.“

Funkce:

- ❖ Design Material
- ❖ Vylepšená spotřeba baterie – projekt Volta
- ❖ Inovovaná oznamovací oblast
- ❖ Odemykáč obrazovka nabízí vstup k nastavení oznámení aplikací
- ❖ Android 5.0 Lollipop ale upustil od widgetů na odemykáč obrazovce
- ❖ Uživatelsky nastavitelné priority pro oznámení aplikací
- ❖ Runtime ART nahrazuje ve výchozím stavu dosavadní Dalvik
- ❖ Podpora 64bitových procesorů
- ❖ OpenGL ES 3.1
- ❖ Podpora náhledu tisku
- ❖ Aplikace třetích stran získaly zpět možnost číst a upravovat data
- ❖ Možnost připnutí jedné aplikace na obrazovku jako omezení přístupu k dalším programům
- ❖ Seznam posledních spuštěných aplikací si systém „pamatuje“ i po restartu
- ❖ Možnost rychlého přenosu nastavení do nového zařízení díky funkci Tap & Go s použitím NFC a Bluetooth
- ❖ Integrována aplikace pro rozsvícení přisvětlovací LED fotoaparátu
- ❖ Funkce chytrého zamykání zařízení¹²

6.0	Marshmallow	Developer preview (Q3 2015)	M
5.1.1	Lollipop	April 21.2015	22

¹² <http://www.svetandroida.cz/historie-androidu-201506>

5.0	Lollipop	October 17 , 2014	21
4.4W-4.4W.2	KitKat+wearbles	October 24 , 2013	20
4.4-4.4.4	KitKat	October 24 , 2013	19
4.3	Jelly Bean	July 24 , 2013	18
4.2.x	Jelly Bean	November 13 , 2012	17
4.1.x	Jelly Bean	July 9 , 2012	16
4.0-4.0.4	Ice Cream Sandwich	December 16, 2011	15
3.2.6	Honeycomb	February 22 , 2011	13
3.0	Honeycomb	February 22 , 2011	12
2.3.3-2.3.7	Gingerbread	February 9 , 2011	10
2.3-2.3.2	Gingerbread	February 9 , 2011	9
2.2-2.2.3	Froyo	May 20 , 2010	6
2.0-2.1	Eclair	Oct 26 , 2009	5
1.6	Donut	Sept 15, 2009	4
1.5	Cupcake	April 30, 2009	3
1.1	Base_1_1	February 9, 2009	2

1.0	Base	2008	1
-----	------	------	---

Tabulka 1



Obrázek 2 - Verze Androidu

3.1.3 Android přehled používaných verzí.

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
2.2 Froyo	8	99,7%
2.3 Gingerbread	10	94,0%
4.0 Ice Cream Sandwich	15	88,7%
4.1 Jelly Bean	16	73,1%
4.2 Jelly Bean	17	55,0%
4.3 Jelly Bean	18	49,5%
4.4 KitKat	19	9,7%
5.0 Lollipop	21	

Obrázek 3 - Android přehled používaných verzí

3.2 Vývoj aplikace pro operační systém Android.

Platforma Android umožňuje vývojářům psát kód v jazyce Java. I když Android nepoužívá pevně stanovené javovské standardy (např. Java SE a ME), vývoj na bázi javy znamená, že platforma je otevřená pro ohromné množství vývojářů. Operační systém Android používá Dalvik Virtual Machine se souborem instrukcí na základě registrů a svým vlastním byte kódem.

Pro vývoj Android aplikací používají vývojáři Googlem vyvinuté javovské knihovny zvané Application Framework API.

Vytvářet aplikace pro operační systém Android je možné ve více programovacích jazycích a ještě více vývojových prostředích. Pro praktickou část práce bylo vybráno vývojové prostředí Android studio, ve kterém se programuje v objektivě orientovaném jazyce Java. Je ale také možné vytvářet kód pro Android v jazyce C#, který je dostupný například v prostředích Microsoft Visual Studio nebo Xamarin.¹³

3.2.1 Android Studio

Android studio je nové vývojové prostředí založené na IntelliJ IDEA. Android Studio bylo firmou Google oficiálně představeno 16. května 2013 na konferenci Google I/O. IntelliJ IDEA je speciálně vytvořena pro rozvoj Androidu a je k dispozici ke stažení pro Windows, Mac OS X a Linux.

3.2.2 Programovací jazyk Java

Java je objektivě orientovaný programovací jazyk, který vyvinula firma Sun Microsystems a představila ho 23. května 1995. Java je programovací jazyk výslovně disponovaný pro distribuované prostředí internetu. Byl navrhnut se "vzhledem a citem" jako jazyk C++, to jest prosazení objektivého modelu programování, avšak lehčí používáním. S použitím Javy lze vytvořit aplikace pro jednotlivé počítače, distribuované servery i klienty. Nadále lze využít pro vývoj menších aplikačních modulů či apletů pro webové stránky, které umožňují uživateli webových stránek interakci se stránkou. Při

¹³ <http://www.mautilus.com/cs/knowhow/vyvoj-mobilnich-aplikaci/#android>

spuštění Java programu je kód velmi rychle převeden na strojový kód daného procesoru (s ohledem na použitý operační systém) – to provádí tzv. Java Virtual Machine (JVM).

- ❖ Platforma JSE (Java Standart Edition)
- ❖ Platforma JEE (Java Enterprise Edition)
- ❖ Platforma JME (Java Micro Edition)
- ❖ Platforma JavaCard¹⁴

3.2.3 Program Eclipse

Eclipse je open source vývojová platforma, která je známa jako vývojové prostředí (IDE) určen pro programování v jazyce Java.

Eclipse je komunita pro jednotlivce i organizace, které chtějí spolupracovat na obchodně-přátelské rovině „open source software“. Projekty jsou zaměřeny na budování otevřené vývojové platformě složené z výsuvných rámců, nástroje a runtime pro budování, implementaci a správu softwaru v celém životním cyklu. Eclipse Foundation je ne-profitní, člen podporované společnosti, která je hostitelem projektu Eclipse a pomáhá kultivovat i open source komunitu a ekosystém doplňkových produktů a služeb. Eclipse Projekt byl původně vytvořen IBM v listopadu 2001 a podporovaný konsorciem dodavatelů softwaru. Eclipse Foundation byla založena v lednu 2004 jako nezávislá ne-profitní korporace zisku a působí jako správce komunity Eclipse. Dnes se komunita Eclipse skládá z jednotlivců a organizací z průřezu softwarového průmyslu.

¹⁴ <https://www.interval.cz/clanky/naucte-se-javu-uvod/>

3.2.4 Android SDK Manager

S Android SDK můžete vyvíjet, ladit a testovat aplikace přes vestavěný emulátor, jenž simuluje Operační Systém Android přímo v systému LinuxMint.

Android SDK Manažer odděluje SDK nástroje, platformy a další komponenty do balíčků pro snadný přístup a správu. Můžete také upravit weby Kontroly SDK manažer pro nové nebo aktualizované SDK balíčky a add-on nástroje. Například můžete nakonfigurovat SDK Manager tak, aby automaticky zjišťoval aktualizace a upozornil vás, když je aktualizován nainstalovaný SDK Tools balíček. Obdržíte-li toto oznámení, můžete se pak rychle rozhodnout, zda změny stáhnout.

Výchozí Android Studio nekontroluje aktualizace Android SDK. Chcete-li povolit automatické kontroly Android SDK:

- ❖ Choose **File > Settings > Appearance & Behavior > System Settings > Updates**.
- ❖ Check the **Automaticly check updates for Android SDK** checkbox and select an update channel.
- ❖ Click **OK** or **Apply** to enable the update checking.

You can launch the SDK Manager in one of the following ways:

- ❖ From the Android Studio **File** menu: **File > Settings > Appearance & Behavior > System Settings > Android SDK**.
- ❖ From the Android Studio **Tools** menu: **Tools > Android > SDK Manager**.
- ❖ From the SDK Manager icon in the menu bar.¹⁵

¹⁵ <http://developer.android.com/tools/help/sdk-manager.html>

Packages			
Name	API	Rev.	Status
Android 4.4.2 (API 19)			
<input checked="" type="checkbox"/> SDK Platform	19	4	<input checked="" type="checkbox"/> Installed
<input checked="" type="checkbox"/> Samples for SDK	19	6	<input checked="" type="checkbox"/> Installed
<input checked="" type="checkbox"/> ARM EABI v7a System Image	19	2	<input checked="" type="checkbox"/> Installed
<input checked="" type="checkbox"/> Intel x86 Atom System Image	19	2	<input checked="" type="checkbox"/> Installed
<input checked="" type="checkbox"/> Google APIs (x86 System Image)	19	10	<input checked="" type="checkbox"/> Installed
<input checked="" type="checkbox"/> Google APIs (ARM System Image)	19	10	<input checked="" type="checkbox"/> Installed
<input checked="" type="checkbox"/> Glass Development Kit Preview	19	11	<input type="checkbox"/> Not installed
<input checked="" type="checkbox"/> Sources for Android SDK	19	2	<input checked="" type="checkbox"/> Installed
▶ Android 4.3.1 (API 18)			
▶ Android 4.2.2 (API 17)			
▶ Android 4.1.2 (API 16)			
<input type="checkbox"/> SDK Platform	16	5	<input type="checkbox"/> Not installed
<input type="checkbox"/> Samples for SDK	16	1	<input type="checkbox"/> Not installed
<input type="checkbox"/> ARM EABI v7a System Image	16	3	<input type="checkbox"/> Not installed
<input type="checkbox"/> Intel x86 Atom System Image	16	1	<input type="checkbox"/> Not installed
<input type="checkbox"/> MIPS System Image	16	4	<input type="checkbox"/> Not installed
<input type="checkbox"/> Google APIs	16	3	<input type="checkbox"/> Not installed
<input type="checkbox"/> Sources for Android SDK	16	2	<input type="checkbox"/> Not installed
▶ Android 4.0.3 (API 15)			
▶ Android 4.0 (API 14)			
<input checked="" type="checkbox"/> SDK Platform	14	4	<input checked="" type="checkbox"/> Installed
<input checked="" type="checkbox"/> Samples for SDK	14	2	<input checked="" type="checkbox"/> Installed
<input type="checkbox"/> ARM EABI v7a System Image	14	2	<input type="checkbox"/> Not installed
<input type="checkbox"/> Google APIs	14	2	<input type="checkbox"/> Not installed
<input checked="" type="checkbox"/> Sources for Android SDK	14	1	<input checked="" type="checkbox"/> Installed
▶ Android 3.2 (API 13)			

Obrázek 4 - Android SDK Manager

3.2.5 App Inventor

V současnosti je stále ve fázi beta a jeho možnosti jsou zatím omezené, přesto už teď představuje velmi zajímavý způsob vývoje jednoduchých aplikací pro platformu Android a to i pro běžné uživatele, kteří s programováním mají jen malé nebo vůbec žádné zkušenosti.

V tomto článku si stručně popíšeme výhody a nevýhody současné verze App Inventoru a na závěr si v praktické ukázce na videu předvedeme, jak snadno lze nějakou aplikaci vytvořit a dostat na zařízení s Androidem. Ukázkovou aplikaci si také můžete hned stáhnout a vyzkoušet na svém telefonu.¹⁶

¹⁶ <http://www.svetandroida.cz/google-app-inventor-snadny-vyvoj-aplikaci-pro-android-201010>

3.2.6 Xamarin

Xamarin je společnost, která se snaží a řeší implementaci .NET na různých platformách jako je Mac, iOS či Android. Umožňuje vytvořit mobilní aplikace pro různé platformy cílené na Android, iOS a Windows pomocí jazyka C# .NET a Visual Studio. Podrobné informace o Xamarin jsou dohledatelné na: Xamarin aplikace v sadě Visual Studio (VisualStudio.com).

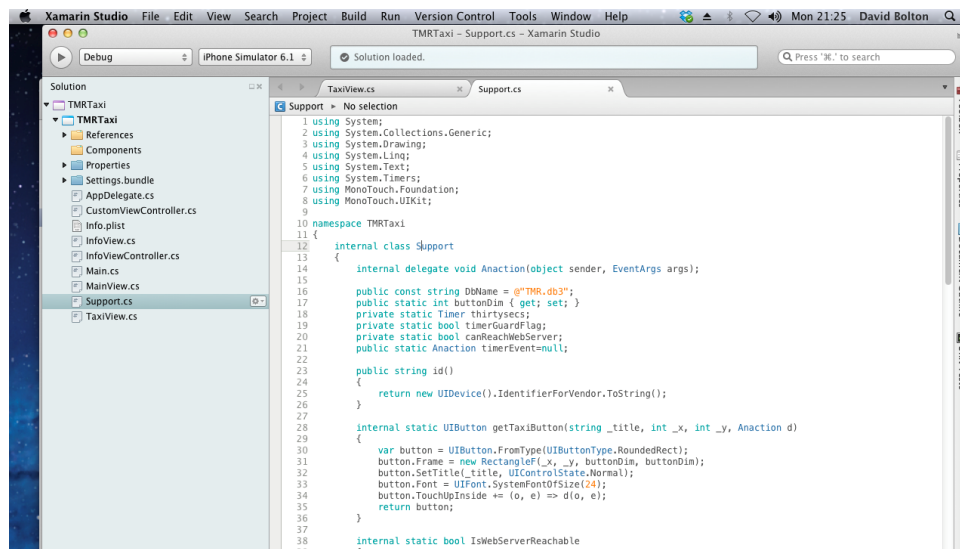
Vývoj aplikací pro moderní platformy zahrnuje řadu aktivit a řeší více, než jen psaní kódu. Tyto činnosti uvedené jako vývoj a provoz (vývoj + operace) zahrnují plánování a sledování práce, navrhování a provádění kódu, správu úložiště zdrojového kódu, spuštění sestavení, správu průběžné integrace a nasazení, testování (včetně testování částí a testy uživatelského rozhraní), monitorování aplikace v reálném čase prostřednictvím telemetrické analýzy výkonu a uživatelského chování.

Visual Studio společně s Visual Studio Online a Team Foundation Server poskytují různé možnosti pro vývoj a provoz, označované jako správa životního cyklu aplikací. Mnoho z nich jsou použitelné pro různé platformy projektů.

To platí zejména s aplikacemi Xamarin vzhledem k tomu, že tyto šablony jsou sestaveny s C# a .NET, na které jsou některé správy životního cyklu aplikací nástroje závislé. Další nástroje však vyžadují integraci se sestavení a modul runtime prostředí. Vzhledem k tomu, že Xamarin aplikace funguje na jiné platformě než Windows a používá Mono implementaci rozhraní .NET.

Následující tabulka identifikuje funkci správy životního cyklu aplikací Visual Studio, které lze očekávat, při práci s Xamarin projektem. Ty, které mají omezení naleznete v dokumentaci propojené s podrobnými informacemi o funkci samotné.¹⁷

¹⁷ <https://msdn.microsoft.com/cs-cz/library/mt162217.aspx>



Obrázek 5 – Xamarin

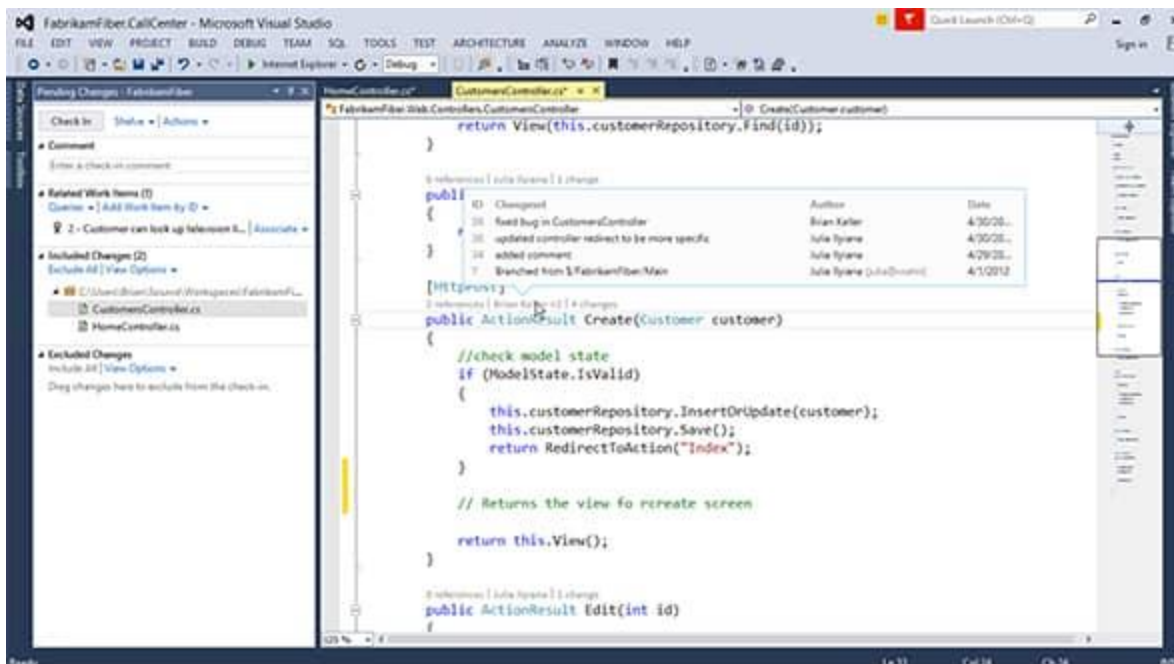
3.2.7 Microsoft Visual Studio

Microsoft Visual Studio je vývojové prostředí (IDE) od Microsoftu. Může být použito pro vývoj konzolových aplikací a aplikací s grafickým rozhraním spolu s aplikacemi Windows Forms, webovými stránkami, webovými aplikacemi a webovými službami jak ve strojovém kódu, tak v řízeném kódu na platformách Microsoft Windows, Windows Mobile, Windows CE, .NET, .NET Compact Framework a Microsoft Silverlight.

Visual Studio obsahuje editor kódu podporující IntelliSense a refaktorování. Integrovaný debugger pracuje jak na úrovni kódu, tak na úrovni stroje. Další vestavěné nástroje zahrnují designer formuláře pro tvorbu aplikací s GUI, designer web, třídu databázových schémát. Je možné přidávat rozšíření, což vylepšuje funkčnost na téměř každé úrovni – od doplnění podpory pro verzovací systémy až po nové nástroje jako editory a vizuální designery pro doménově specifické jazyky nebo nástroje pro další aspekty návrhu programu.

Visual Studio podporuje jazyky prostřednictvím jazykových služeb, což umožňuje, aby editor kódu a debugger podporoval jakýkoliv programovací jazyk. Mezi vestavěné jazyky patří C/C++ ,Visual Basic .NET, a C# .

Je podporováno XML/XSLT, HTML/XHTML, JavaScript a CSS. Existují i verze Visual Studia pro určitý jazyk, které uživatelům poskytují omezenější jazykové služby. Tyto individuální balíčky jsou Microsoft Visual Basic, Visual J#, Visual C# a Visual C++.



Obrázek 6 - Microsoft Visual Studio

3.3 Nástroje důležité pro vývoj

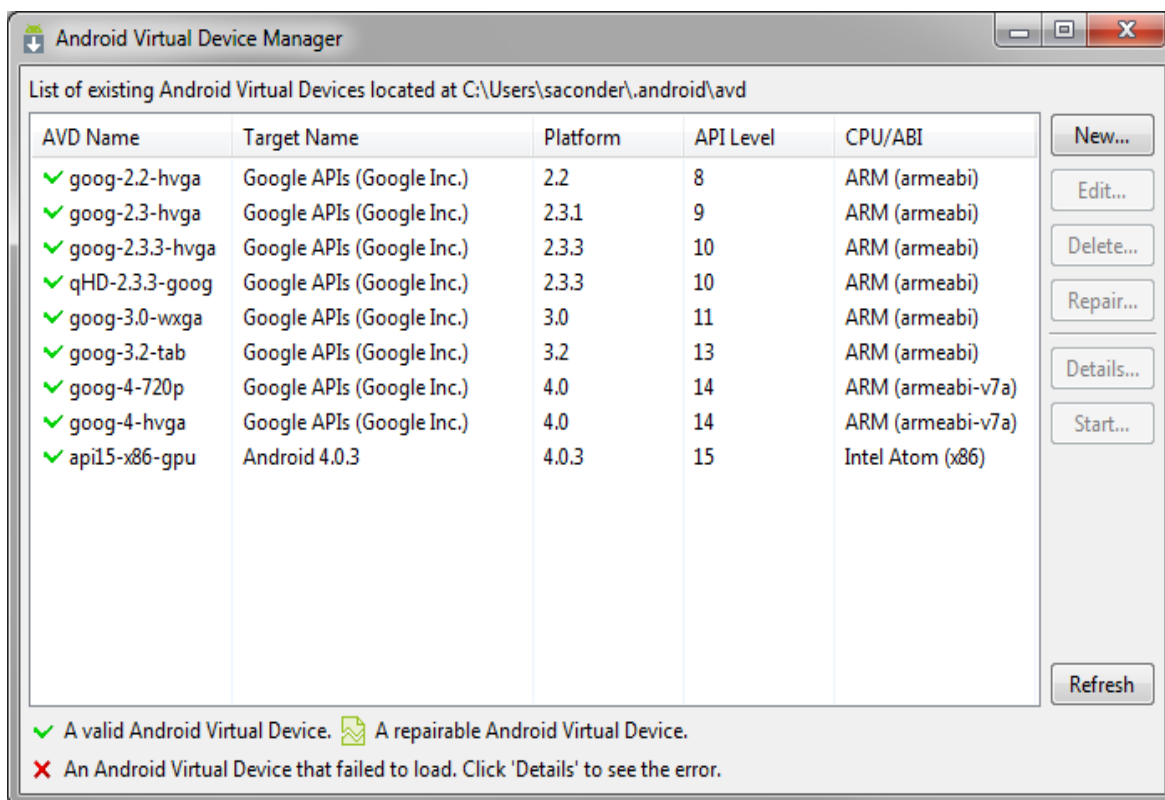
3.3.1 Android Virtual Device Manager (AVD Manager)

AVD Manager poskytuje grafické uživatelské rozhraní, ve kterém můžete vytvářet a spravovat Android virtuálních zařízení (AVDs), které jsou požadovány v Android emulátoru.

Můžete spustit Správce AVD v jednom z následujících způsobů:

- ❖ V Eclipse: Vyberte Okno> Android Virtual Device Manager nebo klikněte na ikonu AVD Manager, v panelu nástrojů
- ❖ V Android Studio: Vyberte položku Nástroje> Android> AVD Manager nebo klikněte na ikonu AVD Manager v panelu nástrojů
- ❖ V ostatních IDE: navigovat do SDK tools / adresáře a spustit android AVD¹⁸

¹⁸ <http://developer.android.com/tools/help/avd-manager.html>



Obrázek 7 - AVD Manager

3.3.2 ADT Bundle

ADT Bundle neboli balík Android Developer Tools obsahuje všechny potřebné nástroje a knihovny pro vývoj aplikací. Jedná o Android SDK, integrované vývojové prostředí (IDE) Eclipse s ADT pluginem a nejnovější verzi platformy Android. Dříve bylo nutné všechny tyto části stahovat a instalovat jednotlivě, rozbalit a spouštět soubor eclipse.exe v adresáře eclipse. Jednotlivé části balíku jsou popsány v následujících kapitolách.¹⁹

¹⁹ <http://developer.android.com/tools/help/adt.html>

3.4 Pexeso

Pexeso je karetní hra pro nejméně dva hráče zaměřená na paměť a soustředění. Její původní název je Memory. Název PEXESO je český akronym vzniklý ze sousloví „PEKELNĚ SE SOUSTŘEĎ“. Hraje se speciální sadou karet. Balíček se skládá ze sudého počtu, nejčastěji 64 karet, které tvoří dvojice. Cílem hry je nalézt co nejvíce stejných dvojic.

3.4.1 Historie

Předchůdce Pexesa bychom mohli najít v tradiční japonské hře se škeblemi zvané Kai awase.

V druhé polovině šedesátých let uváděla Československá televize soutěžní hru Pekelně se soustřeď. Karetní hru na podobném principu následně (přibližně v roce 1965) vyprodukovalo nakladatelství Pressfoto, které ji v dalších letech vydávalo v mnoha tematických mutacích. Za tvůrce se považuje výtvarník Zdenek Princ. Aby nebylo nutné pořizovat licenci za původní název televizní soutěže, vznikl akronym PEXESO.²⁰

3.4.2 Pravidla

Karty se zamíchají a rozloží lícem dolů tak, aby žádný z hráčů neznal rozložení karet. Hráči postupně otáčejí dvojici karet lícem vzhůru, aby je viděli i ostatní hráči. Pokud karty patří k sobě (stejný obrázek), hráč je odebere a otáčí další dvojici (lze hrát i variantu, ve které i po nalezení shodné dvojice pokračuje další hráč v pořadí). Pokud karty k sobě nepatří, otočí je zpět lícem dolů a pokračuje další hráč v pořadí. Hraje se tak dlouho, dokud nejsou všechny karty rozebrány. Vítězem se stane hráč s největším počtem nalezených dvojic.²¹

²⁰ <https://cs.wikipedia.org/wiki/Pexeso>

²¹ <http://tojemi.cz/pravidla>

4. Praktická část práce

V této části práce je popsán vývoj dané hry. Praktická část je dále vedena po důležitých krocích, vysvětlených stylem. Výhledem k rozsahu práce je popisován každý krok a ještě doplněné ukázky kódu s vysvětlením problematiky a také zajímavosti tvoření hry pro operační systém Android, právě ve vývojovém prostředí Android Studio.

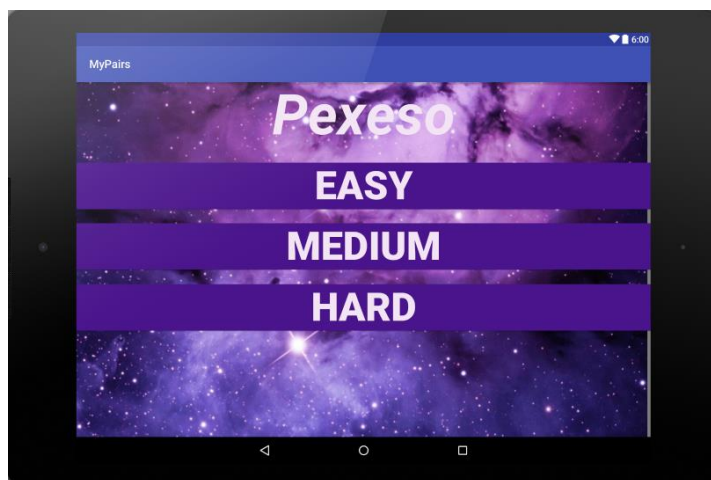
4.1 Založení projektu v Android Studio

Pro vytvoření nového projektu v Android studiu je třeba zapsat název projektu (*application name*) *com.pant Domain*, *package name* and *project location*. Poté vybrat pro co vytváříš tuto aplikace nebo hra, například pro telefon nebo *tablet*, *wear*, *tv*. *Android auto*, *glass*. Poté vybrat aktivity podle toho, jaký typ hry nebo aplikaci vytváříš. Poslední fáze před začátkem programování je název aktivity (*Activity Name*), pro vybrání názvu je třeba stisknout tlačítko *Finish* a pak je možné konkrétně vytvářet.

4.2 Stručný návrh aplikace

Sestavení hry začalo s vytvořením *main activity* a vytvořením jeho designu a k tomu jsem použila *TextView*, *ImageView* a *button* atributy.

Pomocí *ImageView* jsem vytvořila pozadí aktivity, pomocí *TextView* jsem nadepsala název (Pexeso), pomocí *Button* jsem vytvořila tři tlačítka, které pomůže uživateli vybrat si požadovanou obtížnost, *easy button*, *medium button*, *hard button*.



Obrázek 8 - Design MainActivity

Druhá část se skládá z *click button a open new activity*. Toto je třeba k tomu, že po stisknutí vybraného tlačítka, *easy button*, *medium button* nebo *hard button* se dostanete na nové aktivity, kde budete moci začít hru.

- Pro Easy Button Click

```
easy.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        System.out.println("Easy");  
        startActivity(new Intent(MainActivity.this, EasyActivity.class));  
    }  
});
```

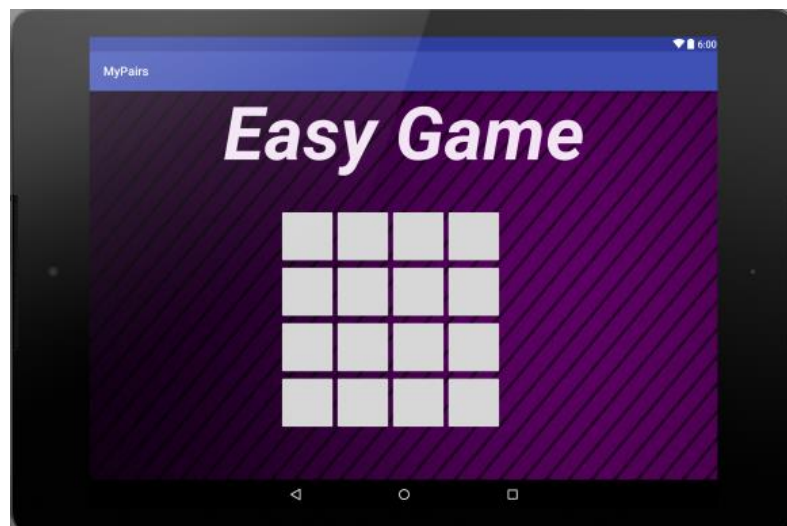
- Pro Medium Button Click

```
medium.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        System.out.println("Medium");  
        startActivity(new Intent(MainActivity.this,MediumActivity.class));  
    }  
});
```

- Pro Hard Button Click

```
hard.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        System.out.println("Hard");  
        startActivity(new Intent(MainActivity.this, HardActivity.class));  
    }  
});
```

Po dokončení již začínám pracovat na designu *easy activity* a rovněž pozadí a v čele nadpisem aktivity with *TextView*. Následuje rozložení šestnácti buttonu na čtvercové ploše se stejnou vzdáleností.



Obrázek 9 - Design EasyActivity

Další fází je, aby se po stisknutí tohoto tlačítka zobrazil obrázek, proto potřebujeme pro každé tlačítko obrázek.

```
View.OnClickListener btn1_handler = new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        setButton(btn1, R.drawable.pp1);
    }
};
```

Tento kód pokračuje až do fáze, dokud všechna tlačítka nebudou mít vlastní obrázek.

Začínáme programovat hlavní podstatu hry pexesa, když najdeme dvojici stejných obrázku tak tyto obrázky zmizí a pokud uděláte chybu automaticky se zavřou. Pote se může vyzkoušet znovu. Důležité je i to, že než fotky zmizí nebo se zavřou, program počká 2 vteřiny. Stejným postupem se vytváří medium a hard aktivity, rozdíl je pouze v počtech obrázku a tlačítek. Medium potřebuje 36 tlačítek s obrázky, hard 64 tlačítek s obrázky.

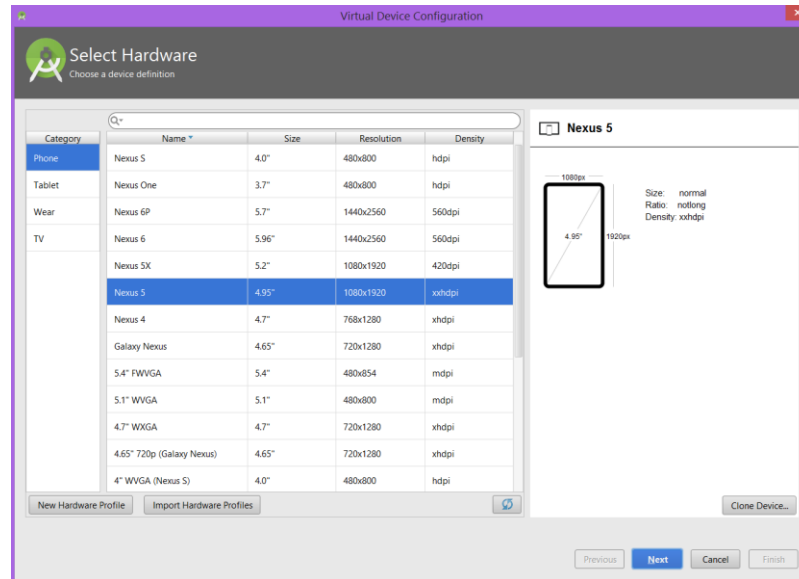
```
public void check() {
    Timer timer = new Timer();
    timer.schedule(new TimerTask() {
        @Override
        public void run() {
```

```

runOnUiThread(new Runnable() {
    @Override
    public void run() {
        if (button1 != null && button2 != null) {
            android.graphics.drawable.Drawable d1 = button1.getBackground();
            android.graphics.drawable.Drawable d2 = button1.getBackground();
            if (d1 != null && d2 != null) { if
(button1.getBackground().getConstantState().equals(button2.getBackground().getConstantState())
) {
                button1.setVisibility(View.INVISIBLE);
                button2.setVisibility(View.INVISIBLE);
                counter -= 2;
            } else {
                button1.setBackgroundResource(defaultBackground);
                button2.setBackgroundResource(defaultBackground);
            }
            button1 = null;
            button2 = null;
        }
    }
}
});
}
}, 2000);
}
}

```

Kontrola hry probíhá v Android Studio nebo ve vlastním device, který pomocí USB se připojí k počítači. Hlavní podmínkou, aby se hra do telefonu nebo do tabletu nahrála je to, že jeho Api level musí souhlasit s api levelem, který se zadává na začátku při vytvoření hry a nebo musí být vyšší než je uvedeno ve Android Studio. Co se týká emulatoru, pro jeho zapnutí si taktéž musíte vybrat model a api level. Vše se vybírá v AVD manager a api level se vybírá v SDK manager.



Obrázek 10 - jak vybrat model emulátoru

4.3 Layouty

Jednotky používané v programování Android hry jsou:

- Pixely (px)
 - Běžně známá jednotka udává velikost přesně v pixelech
- Palce (in - inches)
 - Přesné určení polohy pomocí jednotky palce
- Milimetry (mm)
 - Další přesné určení, tentokrát pomocí bodů
- Body (pt - points)
 - Poslední přesné určení, tentokrát pomocí bodů
- Pixely dle stupňů (sp)
 - Tyto jednotky jsou podobné níže uvedeným dp, ovšem používají se spíše pro určování velikosti písma
- Pixely nezávislé na hustotě (dp)
 - Jedná se o abstraktní jednotku, která pracuje s hustotou pixelů na obrazovce. Jde vždy o poměrnou část. Zadána je v každém případě hodnota vzhledem k obrazovce, která má 160dpi. Kompilátor zvládne přeložit i jednotku *dip*, i když je dp korektnější

Jednotky přizpůsobitelné více obrazovkám jsou *sp* a *dp*, které jsou také doporučené pro použití při vývoji.

Ostatní jednotky používané v Android (px, in, mm, pt) určují přímo danou velikost a proto nejsou vhodné pro tzv. čistý kód. Není možné s nimi napsat aplikaci, která nebude mít problém se přizpůsobit jak menší, tak větší obrazovce smartphonu.

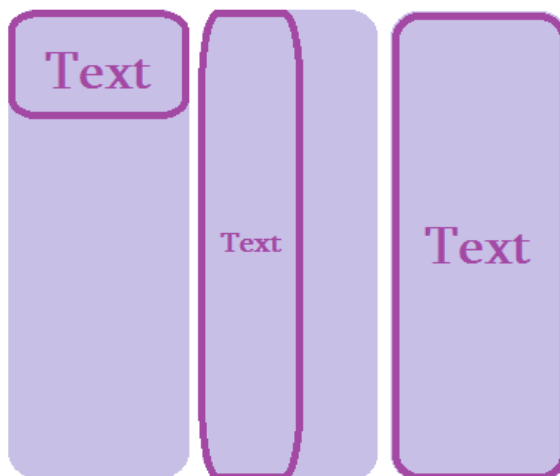
Pokud nejsou žádány relativní jednotky, atributy mohou nabývat dvou základních relativních hodnot:

- `Wrap_content`
- `Match_parent` (`fill_parent`)

Hodnota *wrap_content* znamená, že *View* bude tak veliké, aby obsáhlo svůj obsah. Platí to i pro nadřazené prvky. Jejich obsahu je nadřazená část také přizpůsobena.

Hodnota *match_parent* se snaží vyplnit prostor nadřazeného prvku. Respektive *fill-parent*, což jsou synonyma a obě značí, že se *View* roztáhne na rozměry rodiče, v tomto případě tedy na celou obrazovku. Dvě hodnoty jsou proto, že do API 8, tzn. Android 2.2.x, existovalo jen *fill-parent*, pak však bylo rozhodnuto, že *match-parent* zní lépe. Vzhledem k tomu, že kompilovat budeme vždy na nejnovější verzi API (obě hodnoty se zkompilují stejně, takže i *match-parent* funguje na starších Androidech), nemá smysl *fill-parent* používat, Eclipse by jen házel warningy.

Na následujícím obrázku je vidět, že text má v první části šířku nastavenou na *match_parent*, ale výšku na *wrap_content*. V druhé části je to opačně, šířku nastavenou na *wrap_content*, ale výšku na *match_parent*. V třetí části text obsahuje hodnotu *match_parent*, zabírající celé volné místo nadřazeného prvku.



Obrázek 11 - Vysvětlení "match_parent" a "wrap_content"

4.4.1 Linear Layout

Nezákladnějším typem rozložení je Linear Layout , i proto je použito ve vtváření aplikaci.

LinearLayout je skupina view, která seřadí všechny potomky do jednoho směru, vertikálně nebo horizontálně. Orientace se dá specifikovat atributem *android:orientation*. Všichni potomci *LinearLayoutu* jsou poskládáni jeden na druhém, takže vertikální seznam bude vždy mít pouze jednoho potomka na každém řádku, bez ohledu na to, jak jsou širocí a horizontální seznam bude vždy vysoký jednu řadu (výška nejvyššího potomka plus vycpávka). *LinearLayout* respektuje rozdíly mezi potomky a zarovnání každého potomka.

4.4.2 Relative Layout

Dalším typem rozmístění prvků je *Relative Layout*, který je opakem *Linear Layout*. *RelativeLayout* je skupina View, která zobrazuje dceřiné View v relativním rozložení. Pozice každého View může být specifikována relativně k jakémukoliv jeho sourozenci (jako např. 'vlevo od', nebo 'pod') nebo také ve kterékoliv pozici relativně k nadřazenému *RelativeLayout* (jako např. 'zarovnáno ke spodu', 'vlevo', nebo 've středu'). *RelativeLayout* je velice užitečný nástroj pro návrh a tvorbu uživatelského rozhraní, protože dokáže eliminovat nested view group a zároveň dokáže zachovat hierarchy layout flat, což dokáže značně zlepšit výkon. Pokud používáte několik nested *LinearLayout* skupin, měli byste být schopni je všechny nahradit jedním *RelativeLayout*em.

```
<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"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/title_text_view"
        android:text="Pexeso"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="15sp"
        android:textColor="#AB47BC"
        android:textStyle="bold"/>
```

<ImageView

```
    android:id="@+id/quantity_text_view"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:src="@drawable/pic1"  
    android:scaleType="centerCrop"  
    android:layout_alignParentLeft="true"  
    android:layout_alignParentStart="true"  
    android:layout_alignParentTop="true"/>
```

<TextView

```
    android:id="@+id/textView"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textAppearance="?android:attr/textAppearanceLarge"  
    android:text="Pexeso"  
    android:textColor="#F3E5F5"  
    android:textSize="100sp"  
    android:inputType="textMultiLine"  
    android:textStyle="bold|italic"  
    android:layout_alignParentTop="true"  
    android:layout_centerHorizontal="true" />
```

<Button

```
    android:id="@+id/easy_button"  
    android:text="easy"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:textSize="70sp"  
    android:background="#4A148C"  
    android:textColor="#F3E5F5"  
    android:textStyle="bold"  
    android:onClick="buttonClickFunction"  
    android:clickable="true"  
    android:layout_below="@+id/textView"  
    android:layout_centerHorizontal="true"  
    android:layout_marginTop="26dp" />
```

```
<Button
    android:id="@+id/medium_button"
    android:text="medium"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textSize="70sp"
    android:background="#4A148C"
    android:textColor="#F3E5F5"
    android:layout_marginTop="25dp"
    android:textStyle="bold"
    android:onClick="buttonClickFunction"
    android:clickable="true"
    android:layout_below="@+id/easy_button"
    android:layout_centerHorizontal="true" />
```

```
<Button
    android:id="@+id/hard_button"
    android:text="hard"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textSize="70sp"
    android:background="#4A148C"
    android:textColor="#F3E5F5"
    android:layout_below="@+id/medium_button"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="26dp"
    android:textStyle="bold"
    android:onClick="buttonClickFunction"
    android:clickable="true"/>
```

```
</RelativeLayout>
```

Takhle vypada *Relative Layout* pro *Main Activity*.

4.6 Tlačítka – Buttons

U programování pro operační systém Android existuje více možností jak ho zobrazit, tlačítko má tři části: běžné tlačítko s textem, tlačítko s ikonou a tlačítko s ikonou i popisem.

4.6.1 Tlačítko s popisem

Zápis každé varianty vypadá v kódu jinak. Možnost zobrazit na tlačítku jen text je brána jako základní typ tlačítka, proto se u něj nachází hlavně nutný atribut id, který je důležitý pro práci s daným prvkem.

```
<Button
    android:id="@+id/easy_button"
    android:text="easy"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textSize="70sp"
    android:background="#4A148C"
    android:textColor="#F3E5F5"
    android:textStyle="bold"
    android:onClick="buttonClickFunction"
    android:clickable="true"
    android:layout_below="@+id/textView"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="26dp" />
```

Stejně vypadá kód pro *Medium Button* i *Hard Button*.



Obrázek 12 - Tlačítko pouze s textem (vlastní zpracování)

4.6.2 Tlačítko s ikonou

Druhá varianta zobrazení tlačítka je menší a úspornější. Tlačítko s ikonou je více používanější než *Tlačítko s popisem* nebo *Tlačítko s ikonou i popisem*. Potřebuje vytvořit *ImageButton* class:

```
<Button
  android:id="@+id/easy_button"
  android:text="easy"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:src="drawable/button_icon" .../>
```



Obrázek 13 - Tlačítko pouze s ikonou

4.6.3 Tlačítko s ikonou i popisem

Tlačítko s ikonou i popisem používá třídu *Button* s atributem *android:drawableLeft*.

```
<Button
  android:id="@+id/easy_button"
  android:text="easy"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:drawableLeft="drawable/button_icon" .../>
```



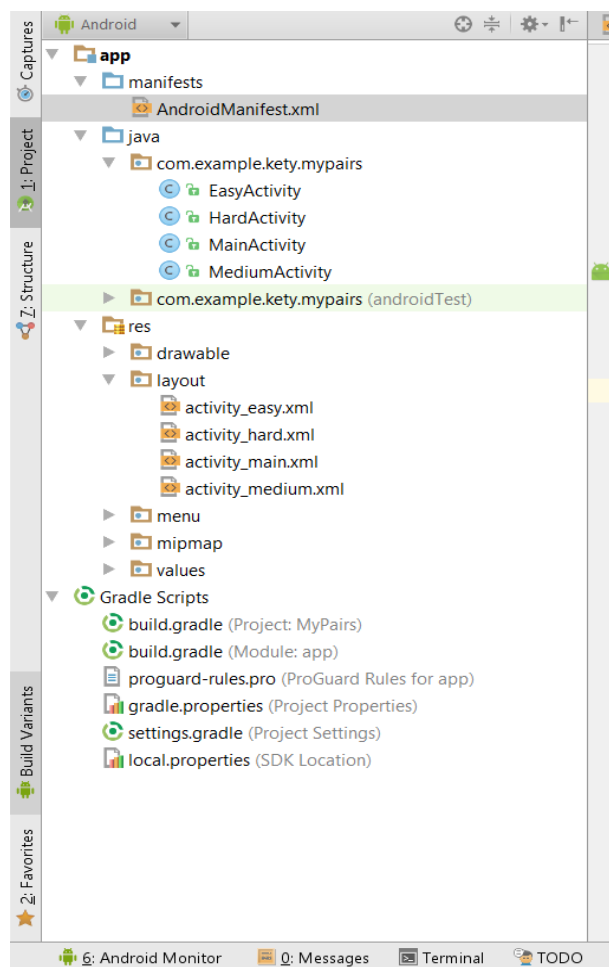
Obrázek 14 - Tlačítko s popisem a také ikonou

4.7 Activity

Activity jsou v Android projektu velmi důležité, pracují s nimi hlavně dva dokumenty v projektu. Activity napsaná v programovacím jazyce Java a layout pro danou Activity napsaný ve značkovacím jazyce XML. Jsou rozdílné a každý z nich má na starosti jiné záležitosti.

I přes to, že se tyto dva dokumenty starají o dvě rozdílné věci, úzce spolupracují a jeden bez druhé by neměly význam.

4.7.1 Layout pro Activity y Activity v programovacím jazyce Java



Obrázek 15 - app Build Variants

4.8 Komentáře (comments)

Komentáře jsou v programování pro Android určeny použitými formáty dokumentů. Komentáře mohou být v xml kódu i v jazyce Java, jednořádkové komentáře nebo víceřádkové komentáře.

- Jednořádkový komentář je dán pouze dvěma lomítky.

// Capture button clicks

- Další varianta v programovacím jazyce Java je víceřádkový komentář.

*/***

** called when the activity is first created.*

**/*

4.9 Kontrola hry

Kontrola hry probíhá buď v Android Studio nebo ve vlastním zařízení, které se pomocí USB připojí k počítači. Hlavní podmínkou, aby se hra do telefonu nebo do tabletu nahrala je to, že jeho Api level musí souhlasit s api levelem, který se zadává na začátku při vytvoření hry a nebo musí být vyšší než je uvedeno ve Android Studio. Co se týká emulátoru, pro jeho zapnutí je taktéž nutné vybrat model a api level. Vše se vybírá v AVD manager a api level se vybírá v SDK manager.

5. Závěr

Cílem této bakalářské práce je vysvětlení programování pro platformu Android. Představena byla architektura Androidu a verze operačního systému Android. Byly uvedeny nástroje pro vývoj aplikací v rozhraní Android, ze kterých je zásadním programovacím jazykem Java a vývojové prostředí Android Studio.

Dalším nástrojem popsáním v této práci je Android SDK Manager, který slouží pro správu verzí Android API. Správné vybírání je nutné pro následné testování aplikací pomocí emulátoru nebo za použití vlastního tabletu.

Na konci teoretické části se zabývám historií a pravidly hry, kterou jsem vytvořila pro mou bakalářskou práci.

Druhá část bakalářské práce se zabývá hrou pexeso.

Od založení projektu v Android Studio přes postupné a důsledné prozkoumání částí, které jsou pro vytvoření aplikace potřeba. Domnívám se, že nejdůležitější element hry je Design aplikace, který rozhodne, zda upoutá cílového uživatele. Taktéž je uveden kód, jak se tvořil design *MainAktivity*.

Další část práce poukazuje na to, jakým způsobem je možné otevřít *NewActivity* pomocí stisknutí na tlačítka. Ve vlastním projektu je pro ilustraci využit kód a obrázek, na kterém je ukázán *EasyAktivity*, který se otevře při stisknutí tlačítka *Easy*.

Hlavní částí práce je konkrétní programování pravidel hry, což je zajímavý proces a obsahuje rozložení tlačítek, přiřazení obrázků, při stisknutí tlačítek převrácení obrázků a na konci při vybírání správných obrázku, zmizení obrázku. Přesně to je hlavní podstatou hry pexesa- abychom nacházeli a přiřazovali dvojici obrázků.

V poslední části práce se zabývám otestováním hry, které je možné dvěma způsoby. Já osobně preferuji testování na vlastním tabletu, protože se jedná o rychlejší cestu a rovněž je rychle patrné, jak vypadá dokončená hra.

Jako autorka si myslím, že programování v Androidu je velmi zajímavý proces, který mne velmi zaujal, do budoucna bych se v dané problematice chtěla dále rozvíjet a do podstaty programování v tomto rozhraní více proniknout.

6. Seznam použitých zdrojů

About, Operační systém Android. Androidaplikace:

<http://androidaplikace.cz/index.php/co-je-operacni-system-android/>

About, Architektura Android. Elitecsoftware:

<http://www.elitecsoftware.cz/vyvoj-pro-android/>

About Android 1.1. Scetandroida:

<http://www.svetandroida.cz/historie-androidu-201506>

About Android 1.5. Developer.Android:

<http://developer.android.com/about/versions/android-1.5-highlights.html>

About Android 1.6. Developer.Android:

<http://developer.android.com/sdk/android-1.6-highlights.html>

About Android 2.0. Developer.Android:

<http://developer.android.com/sdk/android-2.0-highlights.html>

About Android 2.2. Developer.Android:

<http://developer.android.com/sdk/android-2.2-highlights.html>

About Android 3.0/3.1/3.2. Developer.Android:

<http://developer.android.com/sdk/android-3.1-highlights.html>

About Android 4.0/4.0.1/4.0.2. Developer.Android:

<http://developer.android.com/sdk/android-4.0-highlights.html>

About Android 4.2/4.2.2/4.3. Home:

<http://home.zcu.cz/~hodlova/html/vyvoj.html>

About Android 4.4. cdr:

<http://cdr.cz/clanek/os-android-44-kitkat>

About Android 5.0/5.1. svetandroida:

<http://www.svetandroida.cz/historie-androidu-201506>

About Vývoj aplikace pro operační systém Android. Mautilus:

<http://www.mautilus.com/cs/knowhow/vyvoj-mobilnich-aplikaci/#android>

About Programovací jazyk Java. Interval:

<https://www.interval.cz/clanky/naucte-se-javu-uvod/>

About Android SDK Manager. Developer.android:

<http://developer.android.com/tools/help/sdk-manager.html>

About App Inventor. Svetandroida:

<http://www.svetandroida.cz/google-app-inventor-snadny-vyvoj-aplikaci-pro-android-201010/>

About Xamarin. Msdn.microsoft:

<https://msdn.microsoft.com/cs-cz/library/mt162217.aspx>

About Android Virtual Device Manager. Developer.android:

<http://developer.android.com/tools/help/avd-manager.html>

About ADT Bundle. Developer.android:

<http://developer.android.com/tools/help/adt.html>

About Pexeso. Wikipedia: <https://cs.wikipedia.org/wiki/Pexeso>

About pravidla Pexeso. Tojemi: <http://tojemi.cz/pravidla>

7. Seznam obrázků

Obrázek 1- Architektura Android	5
Obrázek 2 - Verze Androidu.....	14
Obrázek 3 - Android přehled používaných verzí.....	14
Obrázek 4 - Android SDK Manager	18
Obrázek 5 – Xamarin.....	20
Obrázek 6 - Microsoft Visual Studio	21
Obrázek 7 - AVD Manager.....	22
Obrázek 8 - Design MainActivity.....	24
Obrázek 9 - Design EasyActivity	26
Obrázek 10 - jak vybrat model emulatoru	28
Obrázek 11 - Vysvětlení "match_parent" a "wrap_content"	29
Obrázek 12 - Tlačítko pouze s textem (vlastní zpracování)	33
Obrázek 13 - Tlačítko pouze s ikonou	34
Obrázek 14 - Tlačítko s popisem a také ikonou.....	34
Obrázek 15 - app Build Variants	35

8. Seznam tabulek

Tabulka 1	14
-----------------	----