

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA STROJNÍHO INŽENÝRSTVÍ**  
**ÚSTAV AUTOMATIZACE A INFORMATIKY**

**FACULTY OF MECHANICAL ENGINEERING**  
**INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE**

# TVORBA APLIKACÍ PRO ANDROID VE VÝVOJOVÉM PROSTŘEDÍ ECLIPSE

ANDROID APPLICATION PROGRAMMING USING ECLIPSE DEVELOPMENT KIT

**BAKALÁŘSKÁ PRÁCE**  
BACHELOR'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**ROBERT KOVÁČ**

**VEDOUCÍ PRÁCE**  
SUPERVISOR

**Ing. JAN ROUPEC, Ph.D.**

BRNO 2013



Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky

Akademický rok: 2012/13

## **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

student(ka): Robert Kováč

který/která studuje v **bakalářském studijním programu**

obor: **Aplikovaná informatika a řízení (3902R001)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

### **Tvorba aplikací pro Android ve vývojovém prostředí Eclipse**

v anglickém jazyce:

#### **Android Application Programming Using Eclipse Development Kit**

Stručná charakteristika problematiky úkolu:

Autor se seznámí s vývojovým prostředím Eclipse a vyvine ukázkové aplikace pro platformu Android.

Cíle bakalářské práce:

Cílem práce je vývoj ukázkových aplikací pro Android využívajících prostředí Eclipse. Vyvinuté aplikace budou tématicky zaměřené k propagaci studia na ÚAI.

Seznam odborné literatury:

Pluta J.: Eclipse: Step by Step. MC Press, 2003.

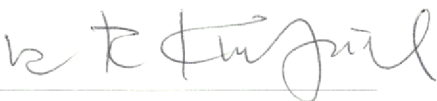
Vedoucí bakalářské práce: Ing. Jan Roupec, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2012/13.

V Brně, dne 15. 02. 2013



  
\_\_\_\_\_  
Ing. Jan Roupec, Ph.D.  
Ředitel ústavu

  
\_\_\_\_\_  
prof. RNDr. Miroslav Doupovec, CSc., dr. h. c.  
Děkan

## **ABSTRAKT**

Obsahem této práce je představení a popis platformy pro mobilní zařízení Android. V první části je seznámení s operačním systémem Android, jeho historie, architektura, verze a konkurenční platformy. Další část popisuje nástroje, které se nejčastěji používají při vyvíjení aplikací pro tento systém a poslední část se zabývá praktickým návrhem a sestavením ukázkové aplikace.

## **ABSTRACT**

This Thesis deals with the presentation and description of the platform for mobile device Android. In the first part, introduction of the OS Android, its history, architecture, versions and competing platforms can be found. The next part deals with tools that are mostly used when developing applications for this OS. The last part describes the practical proposal and the sample application.

## **KLÍČOVÁ SLOVA**

Android, Java, Eclipse, Google Play

## **KEY WORDS**

Android, Java, Eclipse, Google Play



## **PROHLÁŠENÍ O ORIGINALITĚ**

Prohlašuji, že jsem tuto práci vypracoval samostatně, pod vedením pana Ing. Jana Roupce, Ph.D.

.....

Robert Kováč  
22. května 2013

## **BIBLIOGRAFICKÁ CITACE**

KOVÁČ, R. Tvorba aplikací pro Android ve vývojovém prostředí Eclipse. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2013. 41 s. Vedoucí bakalářské práce Ing. Jan Roupec, Ph.D.





## **Poděkování**

Děkuji vedoucímu bakalářské práce panu Ing. Janu Roupčovi, Ph.D. za odbornou pomoc a cenné rady při zpracování mé bakalářské práce.



## Obsah

1	Úvod.....	13
2	Android OS .....	15
2.1	Historie Android .....	15
2.2	Architektura .....	16
2.2.1	Linux Kernel .....	16
2.2.2	Libraries (knihovny) .....	16
2.2.3	Android Runtime.....	17
2.2.4	Application Framework .....	17
2.2.5	Applications .....	17
2.3	Hlavní verze androidu.....	17
2.4	Podíl verzí .....	19
2.5	Licencování.....	19
2.5.1	GNU General Public Licence .....	19
2.5.2	Apache Licence .....	19
2.5.3	Android porušuje copyleft? .....	19
2.6	Jiné platformy .....	20
2.6.1	iOS .....	20
2.6.2	Windows Phone .....	20
2.6.3	BlackBerry OS .....	21
2.6.4	Symbian OS .....	21
2.6.5	Firefox OS.....	21
3	Nástroje pro vývoj aplikací .....	23
3.1	Nástroje a programy.....	24
3.1.1	Java Development Kit.....	24
3.1.2	Eclipse IDE .....	24
3.1.3	Netbeans.....	25
3.1.4	Android Development Tools.....	26
3.1.5	Android SDK .....	28
3.1.6	Android NDK.....	31
4	Vývoj aplikace.....	33
4.1	Návrh .....	33
4.2	Výběr vývojových nástrojů.....	33
4.3	Instalace .....	33
4.4	Volba API levelu .....	34
4.5	Nastavení projektu v Eclipse .....	34
4.6	Activity .....	35
4.7	Intenty .....	37
4.8	Vlastní aplikace.....	37
5	Závěr.....	41



## 1 Úvod

První mobilní zařízení se objevila již v 50. letech 20. století, ale největšího rozmachu dosahují až nyní. Zejména v posledních letech se trh zásadně rozrostl, a to především díky velkému množství nových typů těchto mobilních zařízení. K telefonům se přidaly mobilní počítače, PDA, GPS, tiskárny, tablety a mnoho dalších. V závislosti na tom dochází i k rozmachu trhu s platformami pro tato zařízení, a jednou z nejvýraznějších v poslední době je Android.

Pod označením Android se nachází rozsáhlá platforma, zahrnující operační systém, nástroje a pomůcky pro vývoj aplikací. Jedná se o projekt, který si svoji kvalitou, použitelností a dostupností rychle získal dominanci na trhu. Podařilo se vytvořit ekonomický systém, který přináší užitek nejen svým tvůrcům, výrobcům zařízení a vývojářům, ale i koncovým uživatelům.

Velká pozornost je věnována vývoji a distribuování aplikací. Vývojáři mají zdarma k dispozici veškeré nástroje pro programování a simulování, které mohou implementovat do svých vývojových prostředí a po zaplacení poplatku mají přístup k distribučnímu kanálu, kde mohou vydělávat na svých programech.

Obsahem mé práce je seznámení s operačním systémem, jeho srovnání s konkurencí a analýza nástrojů pro vývoj při vytváření vlastní aplikace. Systém Android jsem si vybral především proto, že je nejrozšířenější a je obklopen velkou komunitou vývojářů.



## 2 Android OS

### 2.1 Historie Android

Společnost Google roku 2005 koupila tehdy málo známou firmu Android Inc. A udělala z ní svoji dceřinou společností. Společně pod vedením Andyho Rubina, jednoho ze zakladatelů Android Inc., začali vyvíjet platformu pod názvem Android.

V roce 2007 bylo založeno sdružení Open Handset Alliance (OHA), neboli uskupení 84 společností [1] (např. Google, HTC, Samsung, Intel, NVIDIA) zabývajících se různými oblastmi mobilních technologií s cílem rozvíjet otevřené standardy pro mobilní zařízení. Prvním jejich produktem byl open-source systém Android. Jádro systému vyvíjí a spravuje převážně Google, ostatní členové se na vývoji podílí a někteří navíc vytváří pro svá zařízení upravené verze s různými nadstavbami a modifikacemi uživatelského prostředí (např. HTC Sens).

V říjnu 2008 firma HTC uvedla na trh první komerční mobilní zařízení se systémem Android s označením HTC Dream. Telefon obsahoval první verzi systému Android 1.0. Následovala řada aktualizací, které postupně zlepšovaly operační systém, přidávaly nové funkce a opravovaly chyby starších verzí. Každá hlavní verze byla v abecedním pořadí pojmenována po dezertu, nebo sladkosti např: verze 1.5 Cupcake, 1.6 Donut nebo nejnovější 4.2 Jelly Bean.

S HTC Dream přišla i první plnohodnotná verze Android software development kitu (SDK), tedy nástroje pro vývoj, testování a opravování chyb Android aplikací. S každou novou verzí Androidu vydává OHA i nové verze Android SDK (nejnovější: Android 4.2 SDK).

Platforma nabrala raketový vzestup. Ve druhém kvartálu roku 2009 zabírala 2,8 procentní podíl světového trhu se smartphones a již v čtvrtém kvartálu roku 2010 s 33% se stala nejrozšířenější mobilní platformou. Na konci roku 2012 70% smartphones využívalo operační systém Android a z celkového počtu prodaných mobilních zařízení zabíral 31% s velkým náskokem na druhý iOS (9%). Podíl na trhu se liší podle územních oblastí. Například v červenci roku 2012 v USA, kde je velmi populární americký Apple, měl Android podíl 52%, ale například v Číně ve stejném období 90%.

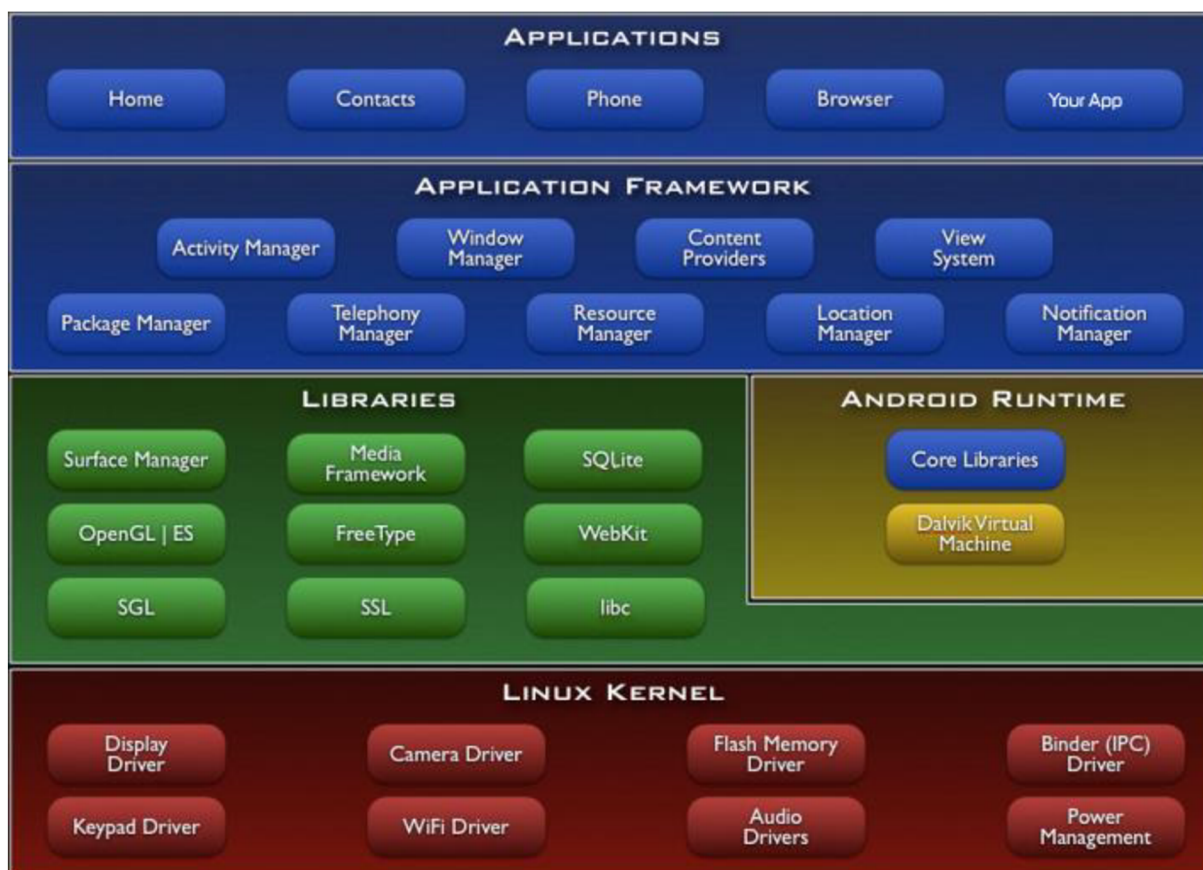
V červenci 2011 Google oznámil 100 miliónů aktivovaných zařízení při 550 000 aktivací denně. V září 2012 to bylo již 550 miliónů zařízení s Androidem a 1,3 miliónů aktivací denně [2]. Tabulka Tab.1 je ukazuje prodejní změny mezi roky 2011 a 2012 [2].

Platforma	4Q '11(mil. Kusů)	4Q '12(mil. Kusů)	4Q '11 (%)	4Q '12 (%)
Android	77,1	144,7	51	70
iOS	35,5	43,5	24	21
Blackberry	13,2	7,3	9	4
Microsoft	2,8	6,2	2	3
Bada	3,1	2,7	2	1
Symbian	17,5	2,6	12	1
Jiné	1,2	0,7	1	0
Celkem	150,2	207,7	100	100

Tab.1 Srovnání prodeje platform ve 4 kvartálu roku 2011 a 2012

## 2.2 Architektura

Architekturu Android operačního systému můžeme popsat jako software 5-ti vrstev. Tyto vrstvy jsou skupinou několika dalších dílčích programů, které společně tvoří operační systém, middleware a důležité aplikace. Každá vrstva poskytuje služby vrstvě nad ní [3].



Obr.1 Grafické znázornění architektury Android OS

### 2.2.1 Linux Kernel

První a základní vrstvou systému je jádro založené na Googlem upravené verzi Linuxu 2.6. Tato vrstva zařizuje komunikaci s hardwarem prostřednictvím ovladačů a také základní systémové služby jako bezpečnost, správa paměti a procesů, síťová připojení a správu napájení. Na rozdíl od Linuxu Android nepodporuje nativní X Window System (systém pro vytváření grafické uživatelské prostředí) a ani úplnou sadu GNU knihoven (linuxových knihoven). Je proto složité předělávat Linuxové aplikace a knihovny pro systém Android.

### 2.2.2 Libraries (knihovny)

Druhá vrstva obsahuje knihovny napsané v jazycích C a C++. Umožňují mobilním zařízením pracovat s různými typy dat.

#### Příklady knihoven:

System C library – standartní systémové knihovny

Media Libraries – podpora obrazových souborů a přehrávání audia a videa

SQLite – databázová knihovna

WebKit – knihovna pro zobrazení HTML



### 2.2.3 Android Runtime

V této vrstvě jsou obsaženy základní knihovny programovacího jazyka Java a virtuální stroj Dalvik. Dalvik Virtual Machine (DVM) je Googlem vytvořená náhrada za Java Virtual Machine (JVM). Důvodem vytvoření nového virtuálního stroje byly licenční podmínky (JVM není volně šiřitelný) a také snaha o optimalizaci pro mobilní zařízení (např. potřeba vyvážit výkon vzhledem k úspoře energie a výdrž baterie)

Android používá virtuální stroj pro spuštění každé aplikace s vlastním procesem [3]. Díky tomu nejsou spuštěné aplikace na sobě závislé, a pokud dojde k chybě, neovlivní jiné procesy ani běh zařízení. Také zjednodušuje správu paměti. Použitím virtuálního stroje se Android liší od jiných mobilních operačních systémů, jako Symbian, nebo iOS, které používají přímou kompilaci zdrojového kódu. Dalvik oproti tomu převádí Java byte kód pomocí kompilátoru a výsledný Dalvik byte kód je následně spuštěn na DVM.

### 2.2.4 Application Framework

Poskytuje jednoduchý způsob ovládání funkcí telefonu a jeho příslušenství. Pomocí frameworku může vývojář svým aplikacím zpřístupnit data, uživatelské rozhraní, hardware a spoustu jiných služeb.

#### Příklad služeb:

Activity Manager: řídí životní cyklus aplikací

Content Providers: spravuje sdílení data mezi aplikacemi

Telephony Manager: dává aplikaci přístup k hlasovým hovorům

Location Manager: určuje polohu pomocí GPS, nebo mobilní sítě

Resource Manager: dává aplikaci přístup k externím souborům

### 2.2.5 Applications

V nejvyšší vrstvě jsou uloženy samotné aplikace. Některé standardní jsou předinstalované (např. SMS klient, Email klient, webový prohlížeč, kalendář...). Další mohou být nainstalované od výrobce telefonu, mobilního operátora, anebo uživatelem stáhnuté z internetu.

## 2.3 Hlavní verze androidu

Verze a datum vydání	Základní informace a změny
Android 1.0 23. září 2008	<ul style="list-style-type: none"> <li>• první komerční Android</li> <li>• distribuovaný pouze na telefonu HTC Dream</li> <li>• aktualizován verzí 1.1 (stále pouze pro HTC Dream)</li> </ul>
Android 1.5 (Cupcake) 30. dubna 2009	<ul style="list-style-type: none"> <li>• Možnost nahrávat a sledovat videa z kamery</li> <li>• Nová softwarová klávesnice s automatickým dokončováním slov</li> <li>• Bluetooth – podpora A2DP</li> <li>• Možnost automaticky připojit Bluetooth headset</li> <li>• Rozšířena funkce kopírovat a vložit.</li> </ul>

<p>Android 1.6 (Donut)</p> <p>15. září 2009</p>	<ul style="list-style-type: none"> <li>• Nové prostředí fotoaparátu, kamery a galerie</li> <li>• Aktualizované vyhledávání hlasem</li> <li>• Quick Search Box (vyhledávání souborů, kontaktů...)</li> <li>• Podpora pro rozlišení displeje WVGA</li> <li>• Vylepšení rychlosti vyhledávání a kamery</li> </ul>
<p>Android 2.0/2.1 (Eclair)</p> <p>26. Října 2009</p>	<ul style="list-style-type: none"> <li>• Podpora pro více velikostí a rozlišení displeje</li> <li>• Nové prostředí prohlížeče a podpora HTML5</li> <li>• Podpora přisvětlovací diody</li> <li>• Digitální zoom (fotoaparát)</li> <li>• Bluetooth 2.1</li> </ul>
<p>Android 2.2 (Froyo)</p> <p>20. května 2010</p>	<ul style="list-style-type: none"> <li>• Možnost instalovat aplikace na paměťovou kartu</li> <li>• kompilátor JIT (zrychlení systému 2x až 5x)</li> <li>• Možnost vytvořit z telefonu WiFi hotspot</li> <li>• Podpora pro OpenGL ES 2.0</li> </ul>
<p>Android 2.3/2.4 (Gingerbread)</p> <p>6. prosince 2010</p>	<ul style="list-style-type: none"> <li>• Podpora více kamer a nových senzorů</li> <li>• Google Maps 5 s 3D přístupem</li> <li>• Rozšíření podpory nativního kódu</li> </ul>
<p>Android 3.0/3.1/3.2 (Honeycomb)</p> <p>22. února 2011</p>	<ul style="list-style-type: none"> <li>• Optimalizace pro velké obrazovky tabletů</li> <li>• Upravený multitasking</li> </ul>
<p>Android 4.0-4.0.4 (Ice Cream Sandwich)</p> <p>19. října 2011</p>	<ul style="list-style-type: none"> <li>• Odemčení telefonu obličejem</li> <li>• Vylepšené rozpoznání hlasu</li> <li>• Ukazatel přenesených dat</li> </ul>
<p>Android 4.1-4.2 (Jelly Bean)</p> <p>9. Července 2012</p>	<ul style="list-style-type: none"> <li>• Project Butter (zrychlené vykreslování obrazu)</li> <li>• Rozpoznávání hlasu offline</li> <li>• Více uživatelských účtů</li> </ul>

*Tab.2 Přehled hlavních verzí Android OS a jejich vylepšení [4]*

## 2.4 Podíl verzí

Android je označován jako roztržštěný operační systém. Velké množství verzí s různými funkcemi na různorodém hardwaru působí komplikace vývojářům aplikací. Tato skutečnost je jednou z hlavních nevýhod celé platformy

Verze	Kódové jméno	Datum vydání	API level	Podíl
2.3.3–2.3.7	Gingerbread	9. únor 2011	10	44%
4.0.x	Ice Cream Sandwich	16. prosince 2011	15	28.6%
4.1.x	Jelly Bean	9. červenec 2012	16	14.9%
2.2	Froyo	20. květen 2010	8	7.6%
2.0–2.1	Eclair	26. říjen 2009	7	1.9%
4.2.x	Jelly Bean	13. listopad 2012	17	1.6%
3.2	Honeycomb	15. červenec 2011	13	0.9%
3.1	Honeycomb	10. květen 2010	12	0.3%
2.3–2.3.2	Gingerbread	6. prosinec 2010	9	0.2%
1.6	Donut	15. září 2009	4	0.2%

Tab.3 Rozdělení verzí Android OS podle podílu na trhu ke dni 4. Březen 2013 [5]

## 2.5 Licencování

Zdrojový kód Androidu je vydáván v rámci open-source licencí. Je tedy k dispozici komukoliv pro jakékoliv účely použití a modifikace. Google spolupracuje s výrobcí zařízení a pro každou novou verzi systému zvolí „vlajkovou loď“. Při jejím vydání pustí do světa i daný zdrojový kód nového OS. Výjimka byla verze 3.0 Honeycomb, neboli verze pro tablety, u které se vydání zdrojového kódu na čas pozastavilo. To se stalo chvíli potom, když nějaký nadšený vývojář nainstaloval verzi 3.0 ne na tablet, ale na smartphone a objevil v té době ještě nevydanou aplikaci od Googlu. K zpřístupnění kódu znovu došlo s vydáním Android 4.0.

Licencování Androidu je rozděleno na 2 části. Modifikovaná verze jádra Linux je vydávána, stejně jako samotný Linux, pod licencí GNU General Public Licence verze 2 a zbytek softwaru je šířeno pod ochranou Apache Licence.

### 2.5.1 GNU General Public Licence

GNU GPL je licencí pro svobodný software původně napsaná Richardem Stallmanem. Základem licence je prosazování copyleftu, tedy důraz na to, aby díla odvozená, nebo využívající dílo chráněné touto licencí, byla také šířena pod GPL [6].

### 2.5.2 Apache Licence

Licence volného softwaru Apache je výtvořem organizace Apache Software Foundation. Podobně jako jiné volné licence dovoluje uživateli používat software pro jakýkoliv účel, distribuovat, modifikovat a šířit modifikované verze softwaru v souladu s podmínkami licence bez autorských poplatků. Na rozdíl od GNU GPL není nutné, aby odvozeniny softwaru měly stejnou licenci [7].

### 2.5.3 Android porušuje copyleft?

Objevují se názory, že Android porušuje GNU GPL licenci, která chrání Linuxové jádro. Google vzal některé hlavičkové soubory jádra, modifikoval je a použil v softwaru Bionic, který tak umožňuje aplikacím využívat rozhraní Linuxu pod licencí, která neprosazuje copyleft.

## 2.6 Jiné platformy

### 2.6.1 iOS

iOS je operační systém vyvíjen společností Apple, která ho využívá na svých zařízeních. Původně byl určen pouze pro mobilní telefony iPhone, ale později byl distribuován i s dalšími zařízeními od Applu, jako iPod Touch, iPad a nejnověji Apple TV. Právě díky oblíbenosti těchto zařízení zabírá iOS v současnosti 21% trhu se smartphones a je po Androidu druhým nejpoužívanějším operačním systémem.

iOS je odvozená mobilní verze operačního systému Mac OS X používaného na počítačích firmy Appel. K jeho naprogramování byly použity jazyky C, C++ a Objective-C [8].

První verze systému řady 1.x vyšla na prvním telefonu iPhone 29.června 2007. Nejnovější verze je iOS 6.1.4.

Na rozdíl od Androidu není iOS volně dostupný a podléhá uzavřené politice Applu. Uživatel nemá přístup k systémovým souborům, ani možnost instalování aplikací, které neprošli schvalovacím procesem. To vedlo uživatele k vytvoření tzv. „Jailbreaku“, neboli procesu, který modifikuje systém tak, aby uživatel tyto možnosti měl, nebo si mohl na Apple zařízení nainstalovat jiný operační systém.

Vývoj aplikací pro iOS je pro programátory lákavější než vývoj pro Android a to například z důvodů:

- Vyšší průměrné výdělky z prodeje aplikací na App Store, než na Google Play.
- Jednodušší navrhování, programování a testování aplikací.
  - iOS nepodléhá fragmentaci stejně jako Android. Počet zařízení od Applu není tak velký a aplikace nemusí být optimalizována pro několik různých velikostí displejů a použitého hardwaru. Většina Apple zařízení používá také nejnovější verzi operačního systému a s tím i nejnovější funkce a knihovny.

Programátoři ovšem musí každý rok platit 99 dolarů za používání plné verze nástrojů pro vývoj a každá aplikace musí procházet schvalovacím procesem.

### 2.6.2 Windows Phone

Jedná se o mobilní operační systém vyvíjený společností Microsoft, jako nástupce za starší Windows Mobile, se kterým již není kompatibilní [9]. První verze Windows Phone 7 vyšla 21. Října 2010. Následovala verze označená 7.8, a poté nejnovější Windows Phone 8. Systém řady 7 je postaven na platformě Windows CE, kdežto novější řada 8 má přizpůsobené jádro z verze Windows NT. To způsobuje opět zpětnou nekompatibilitu, a proto aplikace vytvořené pro Windows Phone 7 nefungují na jejím nástupci Windows Phone 8. Microsoft vytvořil pro Windows Phone nové grafické rozhraní Metro.

Mezi největší distributory tohoto OS patří výrobce mobilních zařízení Nokia, dále pak HTC, Samsung, LG či Acer.

Výhoda WP je v podpoře služeb a nástrojů od Microsoftu jako Microsoft Office, Exchange, Lync, vyhledávání pomocí Bing, propojení s Xbox Live (ve vývoji) a další. Nevýhodou je malá rozšířenost a podpora, počet dostupných aplikací je 4x až 5x nižší než u Androidu a iOS.

### 2.6.3 BlackBerry OS

Je operační systém vyvíjen firmou Research In Motion (RIM). Jedná se převážně o systém optimalizovaný pro firemní účely.

První verze 1.0 určená pro pager BlackBerry 850 vyšla v lednu roku 1999 a nejnovější verze 10 je z roku 2013. Po příchodu Androidu a iOS tento operační systém výrazně ztratil na podílu trhu, ale počet uživatelů stále roste. V prosinci 2012 ho používalo téměř 80 miliónů uživatelů [10].

I BlackBerry podporuje aplikace třetích stran, jejich vývoj a distribuci přes službu BlackBerry World.

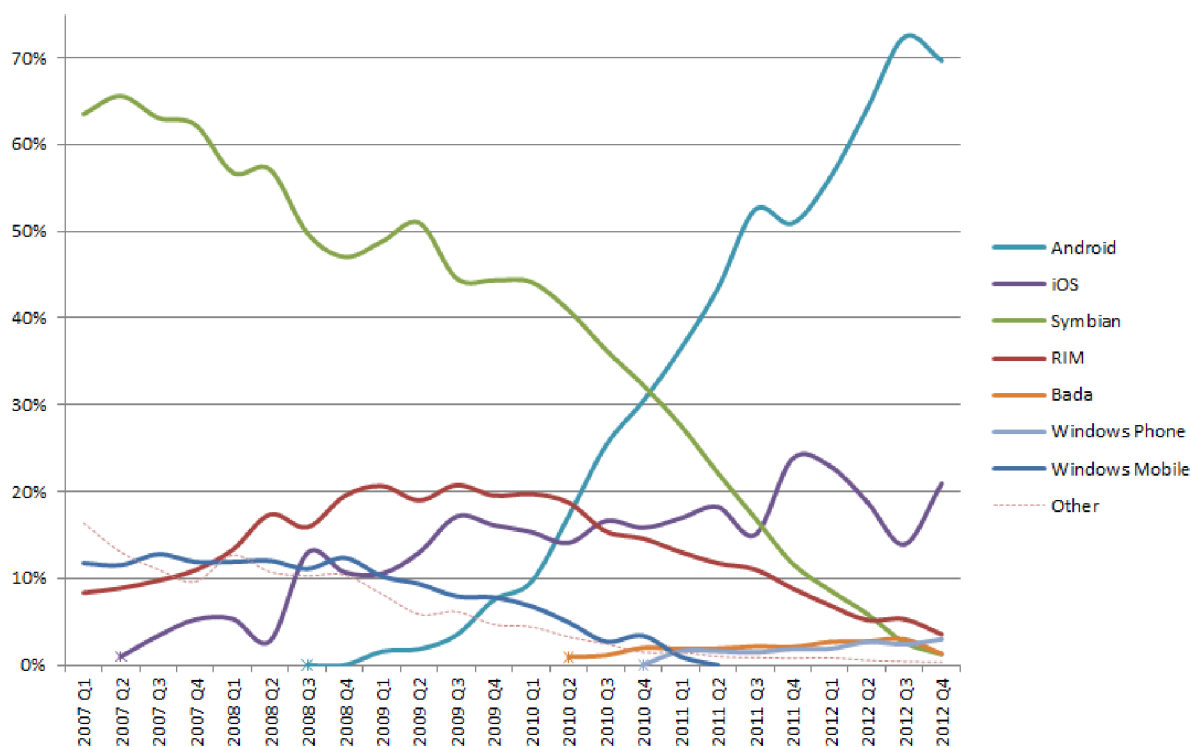
### 2.6.4 Symbian OS

V dnešní době ustupující OS, který ještě na začátku roku 2010 měl největší podíl na trhu chytrých telefonů. Jedná se o volně dostupný OS vyvíjený organizací Symbian Foundation (Nokia, Motorola, Ericsson a další), který byl používán převážně na zařízeních firmy Nokia. Na začátku roku 2011 Nokia oznámila ústup od dalšího vývoje Symbianu a přechod k platformě Windows Phone 7. V současnosti je Symbian stále vyvíjen, ale pouze pomocí malých upgradů a aktualizací [11].

### 2.6.5 Firefox OS

Nový otevřený operační systém pro smartphones a tablety vyvíjený firmou Mozilla. První verze vydána 21. Února 2013.

Systém je postavený na linuxovém jádře a navržen tak, aby aplikace naprogramované v HTML 5 jazyce mohli přímo komunikovat s hardwarem pomocí Javascriptu a knihoven API [12].



Obr.2 Grafické znázornění vývoje prodejů smartphones podle platformy [13]



### 3 Nástroje pro vývoj aplikací

Google se v rámci platformy Android snažil zajistit pro vývojáře kvalitní nástroje a distribuční kanál. Aby podpořil rozjezd, vyhlásil 10 měsíců před uvedením prvního zařízení soutěž Android Developers Challenge o celkem 10 milionů [14] dolarů za nejlepší aplikace v různých oblastech. Na pomoc soutěžícím byla uvolněna beta verze Android SDK.

Platforma má úspěchy i v oblasti vývoje aplikací. Android obklopuje velká komunita vývojářů, nadšenců i firem zabývajících se komerčním i nekomerčním vývojem. Velké oblíbenosti se těší hned z několika důvodů:

- Android je open source, to znamená, že zdrojový kód je volně dostupný a je možné si celý systém přetvořit ke svým potřebám. To samozřejmě ulehčuje výrobcům mobilních zařízení výběr operačního systému, který použijí na svých produktech, protože velice levně získají kvalitní software a mohou snížit cenu. Tato vlastnost měla za následek i výskyt mnoha neoficiálně modifikovaných verzí systému, které získaly oblibu u uživatelů, protože poskytovaly nadstandardní funkce a byly často rychleji aktualizované a vylepšované než verze oficiální, ovšem bez rozsáhlého testování a zjištění kvality. Výsledkem byla reakce ze strany výrobců zařízení v podobě hardwarového omezení, anebo zrušení záruky přístroje.
- Nezávislost na použitém hardwaru. Systém je vyvíjen takovým způsobem, že pomocí ovladačů může běžet na většině zařízení nezávisle na hardwaru a použitém chipsetu. Jde tedy použít jak na mobilních telefonech, tak i tabletech, GPS, netboocích, fotoaparátech a mnoha dalších.
- Android SDK, tedy nástroje pro vývoj aplikace. Je dostupný pro všechny hlavní platformy operačních systémů (Windows OS, Linux OS, Mac OS). Obsahuje vše potřebné pro vývoj, testování, správu a analýzu aplikací.
- Google Play, neboli distribuční služba, vzniklá 6. března 2012 spojením služeb Google Music a Android Market se primárně zabývá distribucí aplikací pro smartphony a tablety s Androidem. Zahrnuje také online distribuci hudby, filmů a videí a také elektronických knih. Pro vývojáře je služba vhodným nástrojem pro distribuci aplikací, aktualizací a získávání zpětné vazby od uživatelů. Google požaduje jednorázový poplatek za službu ve výši 25 dolarů [15]. V současné době není služba k dispozici ve všech zemích v plném rozsahu. Například služba pro distribuci hudby funguje pouze v USA, distribuce elektronických knih jenom v zemích USA, Spojeném království, Kanadě, Francii, Německu, Itálii a Austrálii. Google plánuje postupně rozšířit službu celosvětově.

Rok	Měsíc	Dostupných aplikací	Počet stáhnutí
2009	Březen	2 300	
	Prosinec	16 000	
2010	Březen	30 000	
	Duben	38 000	
	Srpen	80 000	1 miliarda
2011	Říjen	100 000	
	Květen	200 000	4,5 miliardy
	Červenec	250 000	6 miliard
	Říjen	319 000	
2012	Prosinec	380 297	10 miliard
	Leden	400 000	
	Únor	450 000	
	Květen	500 000	
	Červen	650 000	20 miliard
	Září	675 000	25 miliard

Tab.4 Časový přehled počtu dostupných aplikací a stáhnutí v Google Play [16]

### 3.1 Nástroje a programy

Vyvíjet aplikace pro Android jde několika způsoby. V drtivé většině případů je používán jazyk Java v kombinaci s XML a pomocné nástroje ze sady Android Software Development Kit. Existují ovšem i další možnosti, jako například použití nativního programování s jazyky C, nebo C++ a knihoven napsaných v těchto 24stém24y.

#### 3.1.1 Java Development Kit

Pro vývoj v Javě je v první řadě zapotřebí Java Development Kit (JDK). Jsou to freeware softwarové nástroje od firmy Oracle Corporation zahrnující Java Runtime Environment (JRE), který obsahuje virtuální stroj a sadu základních knihoven, a další programy pro vývoj, spuštění, monitorování a ladění aplikací napsaných v programovacím jazyce Java. Nástroje jsou k dispozici pro systémy Solaris, Linux, Mac OS X a Windows.

Některé z nástrojů JDK:

- extcheck – funkce pro detekování konfliktů v JAR souborech
- javac – Java compilátor, který umožňuje převod zdrojového kódu do Java byte kódu
- javadoc – generátor dokumentace a nápověd, které se automaticky generují podle zdrojového kódu
- jConsole – konzole pro řízení a monitorování
- jdb – debugger

#### 3.1.2 Eclipse IDE

Dalším důležitým nástrojem je vývojové prostředí. Oficiálně podporované je Eclipse IDE, které je v současnosti nejpopulárnější vývojové prostředí pro Javu. Eclipse začal jako projekt americké společnosti IBM. Roku 2001 se zformovala skupina firem (IBM, Broland, Merlant a další), která měla v úmyslu Eclipse dále vyvíjet jako open source software. V lednu roku 2004



bylo založeno společenství Eclipse Foundation a jeho členové se společně podílí na vytváření otevřené platformy Eclipse.

Základní verze Eclipse je převážně napsaná v Javě a obsahuje nástroje pro vytváření Java aplikací jako například kompilátor, debugger a další. Díky flexibilnímu návrhu platformy je možné rozšířit množství podporovaných jazyků pomocí přídavných pluginů o jazyky Ada, C, C++, COBOL, Fortran, Haskell, JavaScript, Perl, PHP, Python, R, Ruby, Scala, Clojure, Groovy, Scheme a Erlang. Možné je také vytvářet skripty pro software Mathematica [17].

### **SWT systém**

Pro účely tohoto projektu byl jako součást Eclipse vyvinut grafický systém SWT (Standard Widget Toolkit), tedy nástroj pro vytváření grafického rozhraní aplikace. Je to alternativa k AWT a Swingu od Sun Microsystems, které jsou součástí Java SE. Knihovna SWT je také napsána v Javě a slouží k vykreslování grafických prvků z nativních knihoven operačních systémů prostřednictvím Java Native Interface [17]. To znamená, že grafické prostředí aplikace využívající SWT bude mít stejný vzhledový styl, jako daný operační systém na kterém aplikace běží. Výhodou tedy oproti ostatním grafickým frameworkům je, že pokud dojde ke změnám v knihovnách grafického rozhraní operačního systému, tyto změny se ihned projeví i v aplikaci. Ale například Swing, který využívá vlastních knihoven, se musí při každé změně operačního systému aktualizovat. Používání nativních knihoven má i svoji nevýhodu. Každý operační systém používá jiné knihovny a tedy i SWT musí být vytvořena pro každou platformu zvlášť.

### **Architektura**

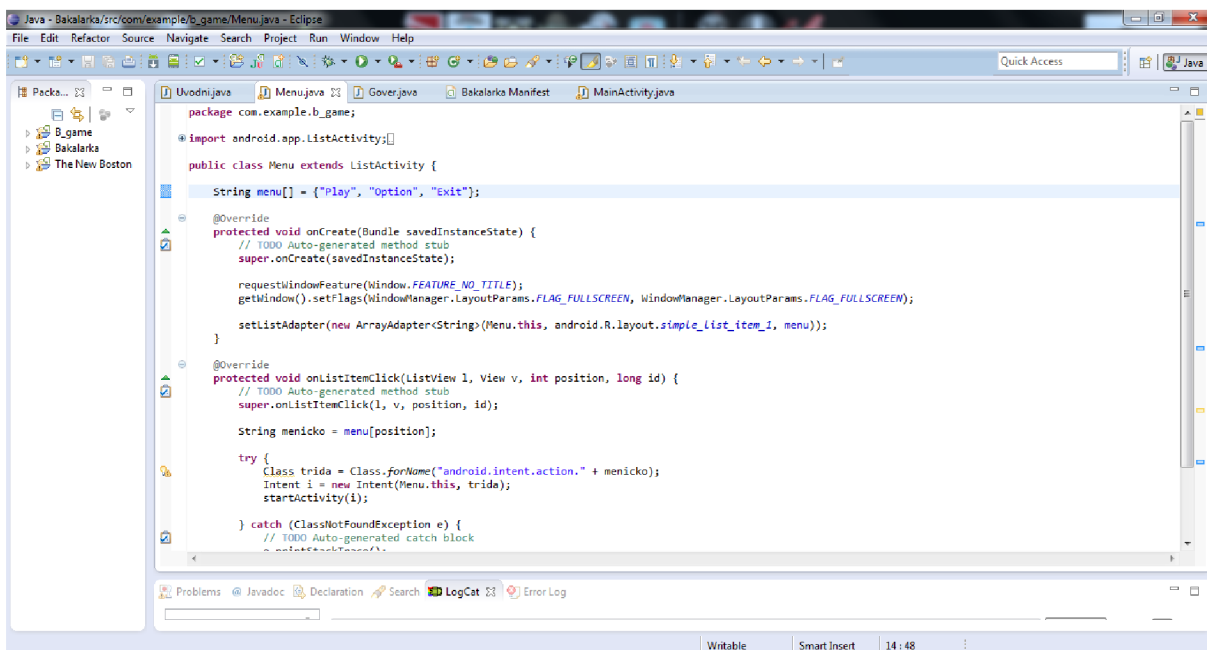
Architektura Eclipse je založená na plug-in mechanismu. Všechno, kromě jádra, je plugin a Eclipse jich poskytuje velké množství pro širokou škálu funkcí. Umožňují nejen rozšířit IDE o nové programovací jazyky, ale také o možnosti pracovat například se sítovými, nebo webovými aplikacemi, databázovými systémy a řadou dalších platform. Některé z pluginů jsou výtvořeny třetími stranami, tedy ne přímo pod oficiálním projektem Eclipse. Mezi tyto neoficiální projekty patří například Adobe Flash Builder, který umožní v Eclipse vytvářet aplikace pro platformu Flash, dále třeba Google Plugin, díky kterému jde navrhovat a optimalizovat cloud aplikace a další. Pro vyvíjení aplikací pro Android je potřebný Android Development Tools Plugin (ADT).

### **Licence**

Eclipse Foundation distribuuje platformu pod vlastní licenci Eclipse Public License. Jedná se o volně dostupný software, který je možné využívat, upravovat, kopírovat a následně i distribuovat. Základem je licence Common Public License (CPL) s odstraněnými podmínkami týkajícími se patentů.

#### **3.1.3 Netbeans**

Druhým nejoblíbenějším vývojovým prostředím pro programování v Javě je Netbeans. Pluginů pro tuto platformu je výrazně méně než pro Eclipse. Mnoho nástrojů se píše výhradně pro Eclipse a dále se už pro jiná IDE neportují, tím poté ztrácejí na atraktivnosti. Netbeans má oproti Eclipse lépe vyvážené pluginy s ohledem na snadnost použití. Vytvářet Android aplikace lze i na této platformě, je k ní dostupný ADT. Dále si s Android aplikacemi poradí i například IntelliJ IDEA IDE, nebo jakýkoli textový editor s pomocí JDK a Apach Ant.



Obr.3 Vývojové prostředí Eclipse IDE

### 3.1.4 Android Development Tools

Oficiální verze ADT je k dispozici pro vývojové prostředí Eclipse. Poskytuje sadu nástrojů pro vývoj a také přidává do vývojového prostředí přístup k řadě funkcí pro efektivnější a pohodlnější programování.

Nejstarší v dnešní době použitelnou verzí je 0.9.4, která vyšla v říjnu 2009 společně s první verzí OS z dvojkové série Androidem 2.0 Eclair. Následovalo dalších 26 verzí, z nichž nejnovější ADT 21.1.0 vyšla v únoru 2013 [18].

Plugin přizpůsobuje vývojové prostředí speciálně pro tvorbu Android aplikací. Umožňuje vytvářet Android projekty, používat nástroje a balíčky Android SDK přímo z nabídky Eclipse, nebo například exportovat aplikace v .apk souboru. Obsahuje editor programovacího jazyka java, který zvládá běžné funkce, jako automatickou kompilaci, nebo kontrolu kompilace syntaxe a má integrované dokumentace ke knihovnam Android API. Dále obsahuje editory, pro vytváření a editaci XML souborů. Jsou to:

- **Grafický layout editor**

Umožňuje graficky navrhovat uživatelské rozhraní aplikace. Je rozdělen do 4 základních částí:

Canvas – zobrazuje náhled na layout aplikace. Podporuje „dragging and dropping“, objektů z palety, tedy uživatel může jednoduše myší vybrat objekt a přetáhnout do náhledu layoutu. Každá verze platformy má více, či méně odlišný grafický styl od ostatních verzí. Canvas použije styl, podle nastavené cílové verze platformy. Pravým klikem myši na objekt je možné jednoduše nastavovat velikost objektů, jejich odsazení od krajů, orientaci a další. Pro verze 3.0 a vyšší je možnost v Canvas přehrávat animace

Outline – plní podobnou funkci jako Canvas, s tím rozdílem, že náhled aplikace nevykresluje, ale zobrazuje ho v hierarchickém seznamu. Podporuje též „dragging and dropping“, objektů z palety. Outline poskytuje lepší možnosti organizace, rychlé vybírání a uspořádání objektů a upravování jejich parametrů.

Palette – poskytují sadu widgetů. Jsou zobrazené ve vykresleném náhledu a kategorizovány do několika skupin, pro rychlé vyhledání a přehlednost.

Configuration Chooser – nabízí možnosti nastavení vykreslování layoutu v Canvasu. Umožňuje:

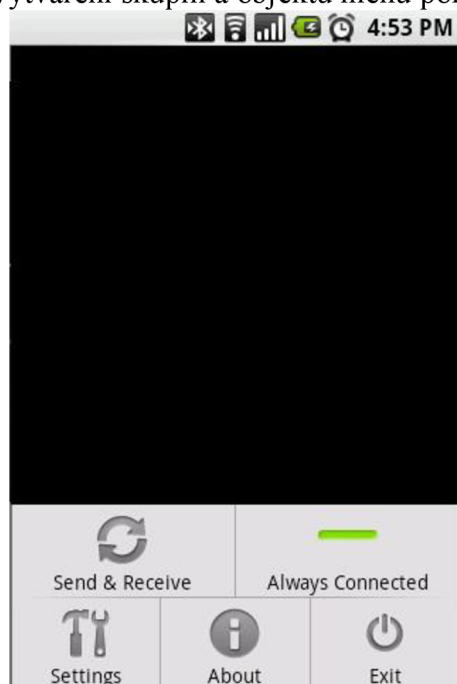
- vybrat předdefinované nastavení konfigurací běžných typů obrazovek, anebo si vytvořit konfiguraci vlastní
- nastavení Portrait (vertikální), nebo Landscape (horizontální) orientace obrazovky
- výběr předdefinovaných, nebo vlastních témat
- určit cílovou verzi platformy, podle které se vykreslí Canvas, určí widgety a témata

- **Android Manifest editor**

Obsahuje jednoduché grafické rozhraní pro editaci AndroidManifest.xml souboru.

- **Menu editor**

K vytváření skupin a objektů menu pomocí jednoduchého grafického rozhraní.



*Obr.4 Příklad menu vytvořeného v Menu editoru*

Upravování XML souboru umožňuje Eclipse nejen v grafických prostředích, ale i pomocí editoru kódu.

Jednou z dalších funkcí, kterou do Eclipse přináší ADT je rychlý přístup k různým typům zdrojů a proměnných pomocí odkazového linku. Některé typy odkazových linků:

- Zdrojový identifikátor typu *R.id.tlacitko* odkazuje na odpovídající XML objekt podle identifikátoru id.
- V souboru R.java deklarace typu `public static final int button_zmacknuto=0x7f020003` odkáže na příslušnou XML definici.
- Definice aktivity v manifestu typu `<activity android:name=".NovaActivity">` odkáže na příslušnou Java třídu pojmenovanou NovaActivity.
- Odkaz na proměnné typu `@string/app_name`, odkáže na hodnotu proměnné `app_name` bez ohledu na to, ve kterém XML souboru je definována.
- Pro otevření souborů odkazy typu `@drawable/icon`, otevře soubor `icon` v adresáři `drawable`.

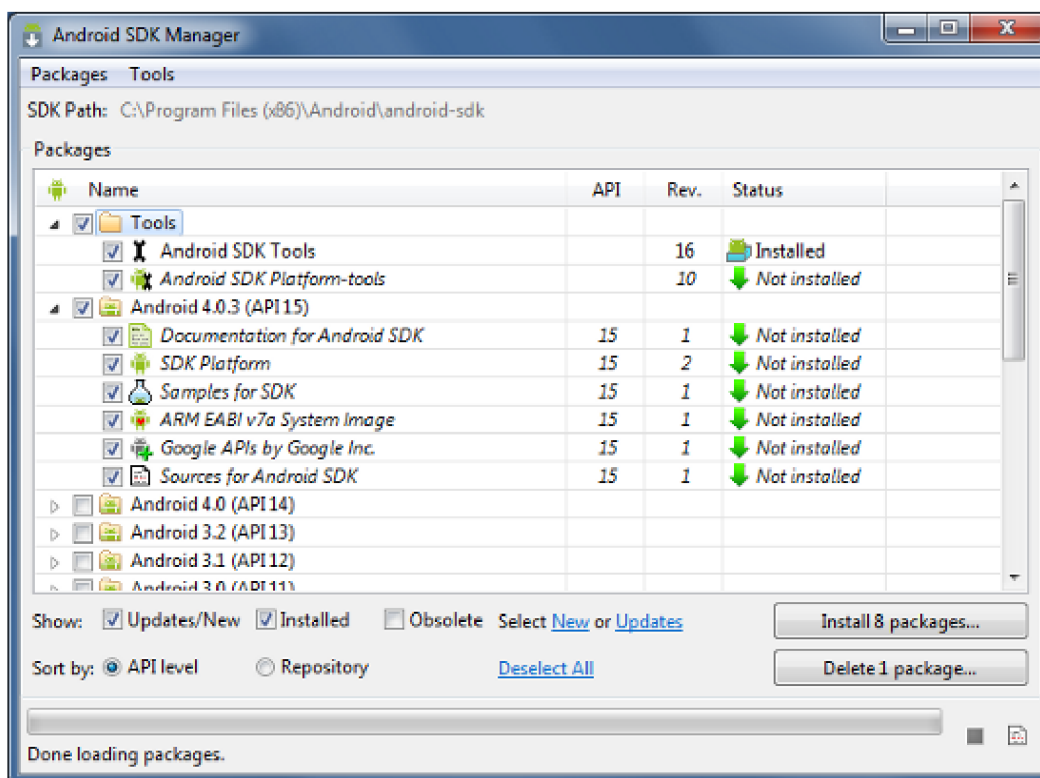
S Android SDK se pracuje přes příkazový řádek. Pro rychlý přístup k důležitým funkcím SDK z vývojového prostředí přidává ADT odkazy přímo do menu Eclipse. Jedná se o funkce:

- Traceview: umožňuje nastavovat profil spuštění vytvořeného programu. (Window > Open Perspective > Traceview)
- android odkazové tlačítka pro přístup k Android SDK Manageru a AVD Manageru, pro nastavení a aktualizace
- Hierarchy Viewer: vizualizuje hierarchické zobrazení aplikace pro celkovou přehlednost a odstranění nedostatků (Window > Open Perspective > Hierarchy Viewer)
- Pixel Perfect: slouží pro podrobné zkoumání grafického rozhraní aplikace (Window > Open Perspective > Pixel Perfect)
- DDMS: nabízí ladění funkcí a vyhledávání chyb, včetně snímání obrazu, informací o vláknech a další... (Window > Open Perspective > DDMS)
- adb: poskytuje přístup k dalším funkcím potřebným pro vývoj aplikací přes příkazový řádek
- ProGuard: umožňuje znesrozumitelnit, optimalizovat a smrštít zdrojový kód. Pokud je ProGuard zapnutý, stane se součástí programu při jeho stavění.

### 3.1.5 Android SDK

Android Software Development Kit zahrnuje rozsáhlý výčet nástrojů, funkcí a knihoven, které pomáhají a zjednodušují vývojářům práci při programování. Minimálně s každou novou verzí OS přijde i nová verze SDK. Nejnovější verze je 21.1.0 vypuštěná v únoru 2013 [19].

Nejjednodušší cestou ke stažení SDK nástrojů je využití SDK Managera, který je součástí ADT pluginu a má přímý odkaz z prostředí Eclipse (Window > Android SDK Manager). Manager poskytuje informace o již instalovaných nástrojích a dalších, které jsou k dispozici.



Obr.5 SDK Manager ukazující nainstalované a dostupné nástroje

SDK zahrnuje balíčky:

- SDK Tools – obsahuje nástroje pro testování, ladění aplikací a další, které jsou nutné pro vývoj aplikace. Ve startovací balíčce SDK je přítomna nejnovější verze
- SDK Platform-tools – zahrnuje také nástroje pro vývoj a ladění aplikací, ale závislé na platformě. Tyto nástroje podporují nejnovější funkce v Androidu a aktualizují se pouze, když je uvolněna nová verze systému. Vždy jsou kompatibilní se staršími verzemi platformy
- Documentation – dokumentace pro Android API
- SDK Platform – existuje jedna SDK Platforma pro každou verzi Androidu. Obsahuje android.jar soubor a knihovny potřebné pro postavení a vyexportování programu
- System Images – každá verze Androidu má jeden, nebo více System Images pro více různých systémů jako například ARM, x86 a další. Tyto soubory jsou potřebné pro běh emulátoru.
- Sources for Android SDK – kopie zdrojového kódu platformy
- Samples for SDK – kolekce ukázkových aplikací, které ukazují možnosti platformy a API
- Google APIs – poskytuje jinou možnost vývoje s pomocí jiných API
- Android Support – statické knihovny, které nejsou součástí standardní platformy

## Debugging

SDK poskytuje většinu nástrojů potřebné pro odstranění chyb a ladění kódu. Potřebným nástrojem je také kompatibilní JDWP ladící program, který poskytuje funkce typu krokování kódu, zobrazování hodnot proměnných, pozastavení běžící aplikace a další. V Eclipse je JDWP přímo implementovaný a funguje automaticky bez potřeby nastavování. Pro jiná vývojová prostředí je možné použít jejich debugger a propojit je se speciálním portem, aby mohli komunikovat s virtuálním strojem.

Nástroje z Android ladícího systému:

- adb

Android Debug Bridge (adb) funguje jako prostředník mezi zařízením a vývojovým prostředím. Umožňuje komunikovat s emulátorem nebo připojeným Android zařízením.

- DDMS

Dalvik Debug Monitor Server (DDMS) je grafický program, který umožňuje měnit přesměrování portů, snímání obrazovky, poskytuje data o vláknech, procesech a mnoha dalších důležitých informacích, které slouží k optimalizaci aplikace a vyhledávání chyb. Umí také testovat chování aplikace při příchozím hovoru nebo SMS.

Každá aplikace v Androidu běží s vlastním procesem a každý tento proces funguje na svém virtuálním stroji (VM). Tento virtuální stroj je pak připojen k unikátnímu portu, ke kterému se také může připojit debugger a monitorovat aplikaci.

Když je DDMS spuštěno připojí se k adb. Pokud je monitorované zařízení připojeno, mezi adb a DDMS se vytvoří služba VM monitoring service, která oznamuje DDMS, jestli je virtuální stroj na zařízení spuštěn nebo ukončen. Jakmile je VM spuštěn, DDMS načte ID procesu virtuálního stroje přes adb a otevře připojení k debuggeru prostřednictvím adb daemon. Poté může DDMS komunikovat s VM pomocí síťového protokolu.

- AVD

Pro ladění je potřebné, aby aplikace byla spuštěna. K tomu může sloužit zařízení připojené pomocí USB nebo také Android Virtual Device (AVD). AVD je emulátor neboli program, který nahrazuje skutečné zařízení podle nastavené konfigurace hardwaru a softwaru. Je tedy schopný spustit aplikaci a simulovat její chování, jakoby byla spuštěná na skutečném zařízení.

Hardwarovou konfiguraci lze libovolně změnit. Například je možné určit, jak velkou paměť bude zařízení obsahovat, zda bude mít fotoaparát, fyzickou klávesnici, dále umožňuje nastavení rozměrů obrazovky, vzhledu, nebo simulaci přítomnosti SD karty a další možnosti. Ze softwarového hlediska se nastavuje verze platformy. Tato nastavitelnost umožňuje vývojářům testovat fungování aplikace v různých podmínkách a předcházet tak problémům, které jsou spojeny s rozříštěností platformy Android.

## API level

API level identifikuje verzi API (application programming interface), která je nabízena jako součást dané verze platformy Android. Obsahuje knihovny, metody a funkce, které aplikace mohou používat pro komunikaci se systémem.

API se skládá z:

- sada balíčků a tříd,
- sada XML prvků a funkcí pro deklarování manifest souboru,
- sada XML prvků a funkcí pro deklarování přístupu ke zdrojům,
- intent funkce,
- sada oprávnění, které mohou aplikace vyžadovat,

Nová verze platformy Android většinou přináší i aktualizace API. Tyto aktualizace jsou navrhovány tak, aby byla zajištěna zpětná kompatibilita se staršími verzemi platformy, tedy změny jsou většinou spíše aditivní a zavádějí se nové funkce. Dochází i k nahrazování starších funkcí za nové, ale tyto nahrazené funkce nejsou odstraňovány, aby je mohly využít starší verze systému. Jediné případy, kdy dochází k vymazání některých funkcí, jsou z důvodu zajištění bezpečnosti a odolnosti systému.

API level je tedy důležitým směrníkem kompatibility. OS má používanou hodnotu API levelu zapsanou přímo v systému. Poskytuje informaci o tom, jakou maximální verzi API platforma podporuje. Aplikace mohou k popisu využívat soubor manifest, ve kterém se může nastavit více atributů:

`android:minSdkVersion` – specifikuje minimální hodnotu API levelu pod kterou je aplikace schopná fungovat. Výchozí hodnota je 1.

`Android:targetSdkVersion` – definuje hodnotu API levelu, pro kterou byla aplikace navržena.

`Android:maxSdkVersion` – uvádí maximální API level, pod kterým je aplikace schopna fungovat.

V případě že aplikace vyžaduje vyšší API level, který není platformou podporován, by systém neměl povolit instalaci.

### 3.1.6 Android NDK

Nativ Development Kit (NDK) je nástroj, který umožní vytvořit aplikaci nebo její část pomocí nativních programovacích jazyků, kterými jsou například jazyky C a C++. Takto napsaný program se nekompile ve virtuálním stroji, ale přistupuje přímo k hardwaru zařízení, který ho provádí. Oficiální zdroje zabývající se vývojem pro Android nedoporučují použití NDK, pokud to není nezbytně nutné pro fungování aplikace. Nativní přístup má málokdy za následek znatelné zlepšení výkonu, ale vždy dochází k zvýšení složitosti programu.

Použití NDK je vhodné u operací náročných pro procesor, jako zpracování signálu, simulace fyziky a dalších. Vždy je ale dobré si nejdříve zjistit, jestli dostatečné funkce neposkytuje i Android SDK.





## 4 Vývoj aplikace

### 4.1 Návrh

Před začátkem vývoje je vhodné vytvořit si přibližný návrh a mít promyšlené k čemu bude aplikace sloužit a jak bude fungovat. Výhodné je dopředu si ujasnit nebo nejlépe graficky načrtnout části aplikace a přechody mezi nimi. V mém případě jsem chtěl, aby aplikace byla jednoduše ovladatelná hra, ve které se sestřelují padající objekty. Moje aplikace má přechod mezi hrou samotnou a výsledným zobrazením výhry či prohry a následné opětovné spuštění hry.

### 4.2 Výběr vývojových nástrojů

Dalším krokem je výběr nástrojů pro vývoj. Většina aplikací pro Android je napsaná v jazyce Java s pomocí Eclipse vývojového prostředí, ADT pluginu a Android SDK nástrojů. Tato varianta je nejvýhodnější hlavně vzhledem k množství návodů, tutoriálů, instruktážních videí a dalších zdrojů popisujících vývoj s těmito nástroji, dostupné v literatuře a na internetu. Velice dobrý zdroj těchto informací je oficiální web android vývojářů <http://developer.android.com/index.html>, na kterém se dá najít dokumentace ke každému nástroji a funkci z Android SDK.

### 4.3 Instalace

Pro programování v Javě je potřeba Java Development Kit (JDK), který je volně dostupný na stránkách společnosti Oracle:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

JDK je součástí všech balíčků Java (SE, EE, ME a dalších) a je dostupný pro většinu platform. Po stáhnutí stačí spustit a nainstalovat.

Dalším krokem je instalace IDE neboli vývojového prostředí. Eclipse IDE je distribuován ve více verzích s různými modifikacemi. Pro programování v Javě je vhodná verze s názvem **Eclipse IDE for Java Developers**, která je dostupnou například na oficiálního webu Eclipse <http://www.eclipse.org/downloads/>. Instalace Eclipse požaduje vybrat umístění a pojmenování workspace, tedy složky, ve které se ukládají projekty a veškeré soubory.

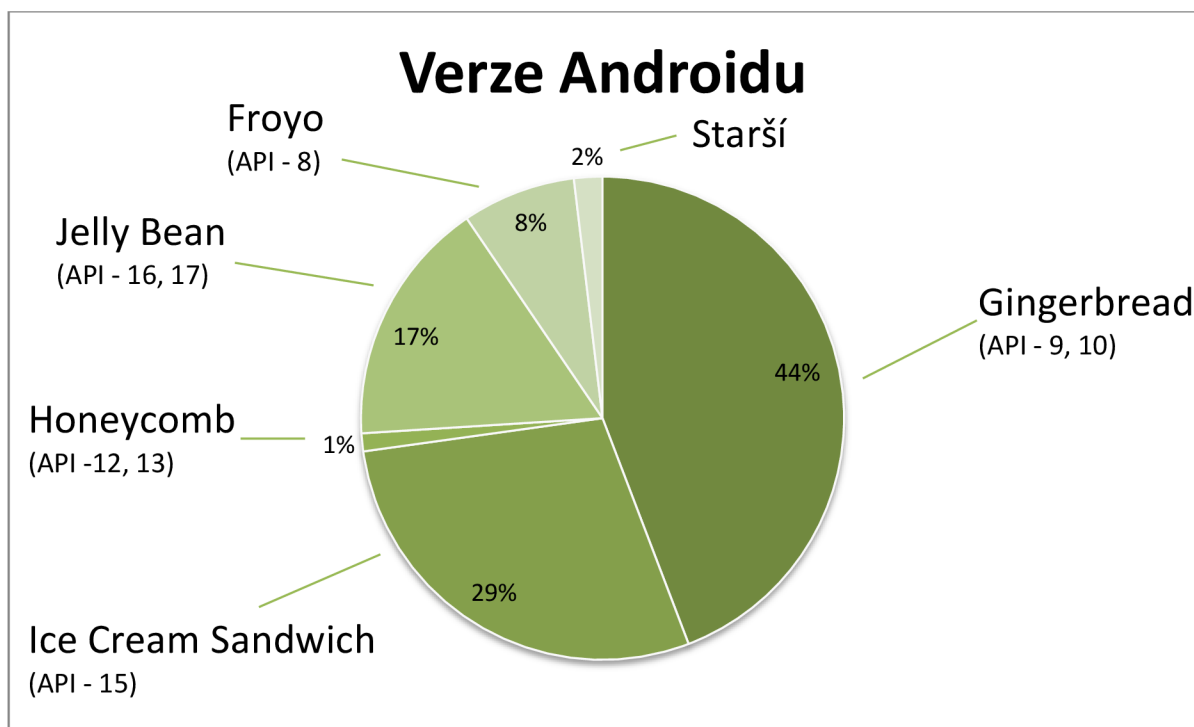
Jako další je na řadě Android ADT plugin. Tento nástroj si přes internet umí stáhnout přímo Eclipse. Postup implementace pluginu:

1. spustit Eclipse a vybrat Help > Install New Software
2. kliknout na Add
3. do kolonky Name napsat ADT plugin (nebo něco podobného) a kolonky Location vložit odkaz : <https://dl-ssl.google.com/android/eclipse/>
4. kliknout OK
5. zaškrtnout políčko Developers Tools a kliknout Next
6. po přečtení přijmout licenční dohodu a kliknout Finish
7. po dokončení instalace restartovat Eclipse

Jako poslední je potřeba stáhnout Android SDK ze stránek Android vývojářů <http://developer.android.com/sdk/index.html>. Stáhnutý soubor se extrahuje a poté je potřeba spustit SDK Manager, ve kterém se zvolí nástroje a jejich verze ke stáhnutí. V položce virtual devices se vytvoří emulátor, kterému se přiřadí jméno a verze Android platformy, pro kterou se bude vyvíjet. Následně v nastavení Eclipse (Preferences) se zvolí položka Android a zadá se cesta ke složce s SDK.

#### 4.4 Volba API levelu

Dalším důležitým krokem před vývojem aplikace je volba cílového API levelu. Pokud je potřeba využít nejnovější funkce, které přišli s poslední verzí systému, vybere se nejvyšší API level, ale tím se omezí počet zařízení, na kterých aplikace bude fungovat. V zájmu rozšíření trhu pro tuto aplikaci je nutné najít kompromis mezi dostatečným množstvím funkcí dané platformy a jejím zastoupením na trhu. K tomu je nápomocný graf na obrázku Obr.6.



Obr.6 Zastoupení verzí Android platformy na trhu ke dni 4. Březen 2013 [5]

Například, když se zvolí cílová verze Gingerbread, aplikace bude fungovat na všech platformách podporujících API level 9 a vyšší. Pro svoji aplikaci jsem zvolil verzi Froyo s API levelem 8 a to z důvodu, že funkce obsažené v knihovnách této verze jsou dostatečné pro moji jednoduchou aplikaci a také bude fungovat na 98 % zařízení s Androidem.

#### 4.5 Nastavení projektu v Eclipse

Díky ADT pluginu se dají v Eclipse vytvářet Android projekty, které jsou již optimalizované pro vývoj Android aplikací. V Eclipse se zadá

**File> New Software> Android Application Project** (v případě, že není dispozici File> New Software> Other> Android Application Project).

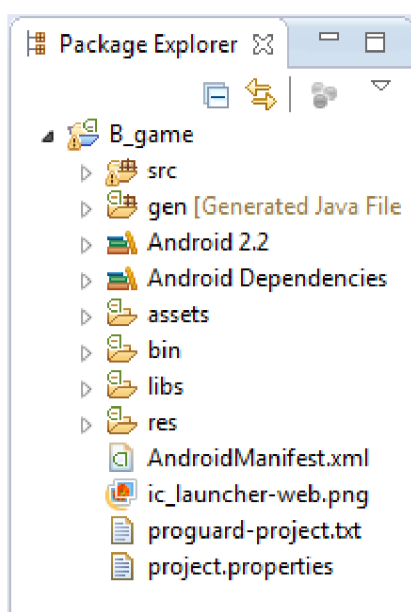
V nastavení se zvolí jméno projektu, jméno aplikace a Package name.

Package name musí být unikátní. Pokud by nějaká aplikace na Google Play měla stejný Package name, aplikace by se nedala na server nahrát.

Dále se nastavuje cílová verze platformy, název hlavní Java třídy a nakonec také obrázek loga aplikace.

Struktura projektu:

- src – do složky src (source) se umísťují soubory s Java zdrojovými kódy
- gen – je složka, ve které se automaticky generují odkazy na použité zdroje
- Android 2.2 – obsahuje veškeré balíčky, metody a funkce pro danou platformu (v mém projektu pro platformu Froyo 2.2)
- Android Dependencies – je virtuální složka, která ukazuje závislé .jar soubory
- asset – zde se ukládají určité typy zdrojů (např. fonty)
- bin – je složka generovaná kompilátorem, která obsahuje všechny .class soubory, aby mohl Eclipse vytvořit .apk soubor spustitelný v Androidu
- libs – obsahuje sdílené knihovny
- res – do res (resources) se vkládají obrázky (možné vložit ve třech různých rozlišení), XML layout soubory (vytvářejí vzhled aplikace) nebo zvuky, soubory s proměnnými a další
- AndroidManifest – je XML soubor, ve kterém jsou popsány části aplikace a všechny důležité atributy, které dávají aplikaci dohromady



Obr.7 Struktura Android Application Project

#### 4.6 Activity

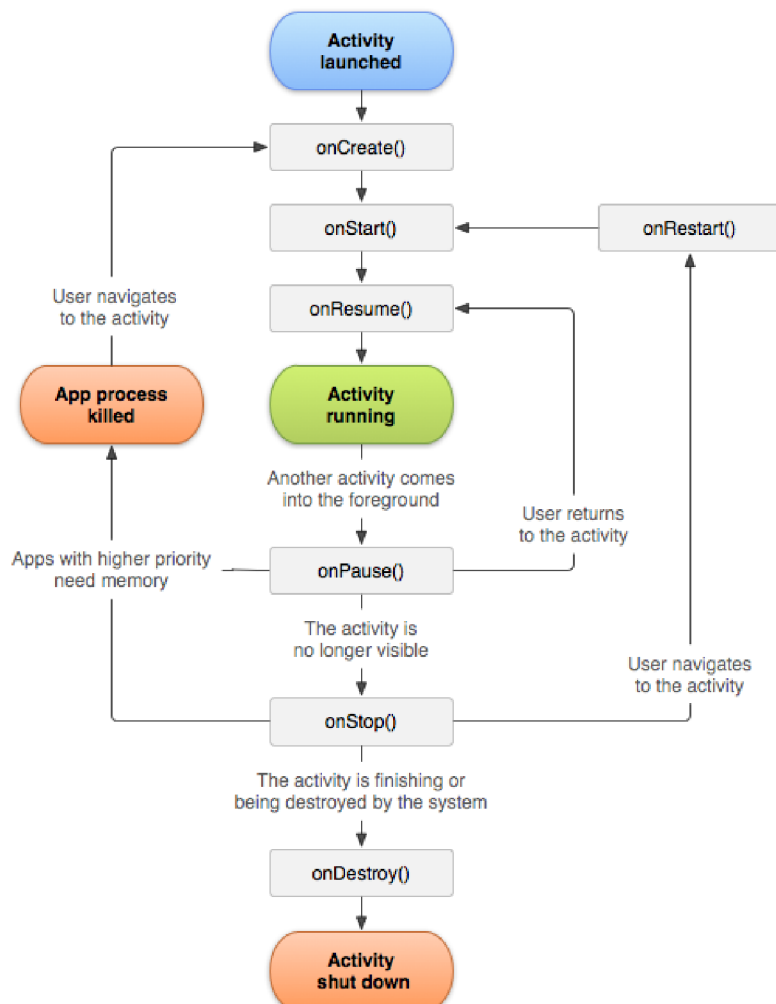
Základním prvkem aplikací jsou aktivity. Lze si je představit jako jednotlivé obrazovky aplikace, ale mohou být i bez uživatelského rozhraní. Tedy pokud má aplikace 4 obrazovky (úvod, menu, účelná část, závěr), je postavena minimálně ze 4 aktivit (pokud aktivity používá).

Aktivita je vytvořena ze třídy přidáním syntaxe podle obrázku Obr.8. Takto nově vytvořená aktivita může být aktivitou hlavní, která se spustí při startu aplikace, nebo rozšiřující a spustí se v návaznosti na jiné aktivity, nebo se nespustí vůbec a slouží jako poskytovatel obsahu nebo služeb. Přechody aplikací jsou nastavené v souboru Manifest.xml.

```
public class Uvodni extends Activity {
```

Obr.8 Příklad vytvoření Activity

Při vývoji aplikací pomocí aktivit je důležité znát jejich životní cyklus. Spuštěná aktivita postupně prochází cyklem a volá podle toho příslušné funkce.



Obr.9 Životní cyklus aktivity

- `onCreate()` - zavolá se po startu aplikace. Je vhodné zde provést veškeré statické nastavení (vzhled, proměnné, atd.),
- `onStart()` – zavolá se před zobrazením aplikace uživateli,
- `onResume()` – zavolá se před zahájením interakce s uživatelem,
- `onPause()` – zavolá se, když se systém chystá spustit jinou aktivitu, v této části je vhodné uložit data, zastavit animaci a další procesy, které by mohli následně zatěžovat procesor,
- `onStop()` – zavolá se, když aplikace již není pro uživatele viditelná,
- `onRestart()` – volá se tehdy, když se aktivita uživateli má znovu zobrazit,
- `onDestroy()` – volá se v případě, že je aktivita zrušena.

U své aplikace také používám aktivity. Hlavní aktivitou je samotná animace hry, která se provádí, dokud uživatel nevyhraje, nebo neprohraje. Podle výsledku hry se zavolá aktivita oznamující vítězství, nebo aktivita s oznámením o prohře. Následně se hra spustí od začátku.

## 4.7 Intenty

Intent je objekt, který umí pojmout jednoduchá data a posílat je mezi aktivitami. Rozdělují se na explicitní a implicitní.

V případě použití explicitního intentu systém ví přesně, co chceme spustit.

```
Intent intent = new Intent(context, Gover.class);
```

*Obr.10 Příklad explicitního intentu*

Implicitní intent umožňuje pouze popsat záměr, nikoliv způsob jak ho provést. To velice zjednodušuje práci a umožňuje interakci aplikací v rámci Androidu.

Pokud je třeba například zobrazit stránku [www.google.com](http://www.google.com), tak s pomocí explicitního intentu je potřeba nejprve vyzkoušet přítomnost všech možných prohlížečů a spustit první, který je dostupný. Při použití implicitního intentu podle obrázku Obr.11, Android zjistí, které aktivity mají nastavený takový intent filter, který je slučitelný s daty v intentu. Pokud vyhledá více možností, nabídne uživateli výběr.

```
Uri uri = Uri.parse("http://www.google.com");
Intent intent = new Intent(Intent.ACTION_VIEW, uri);
startActivity(intent);
```

*Obr.11 Příklad implicitního intentu*

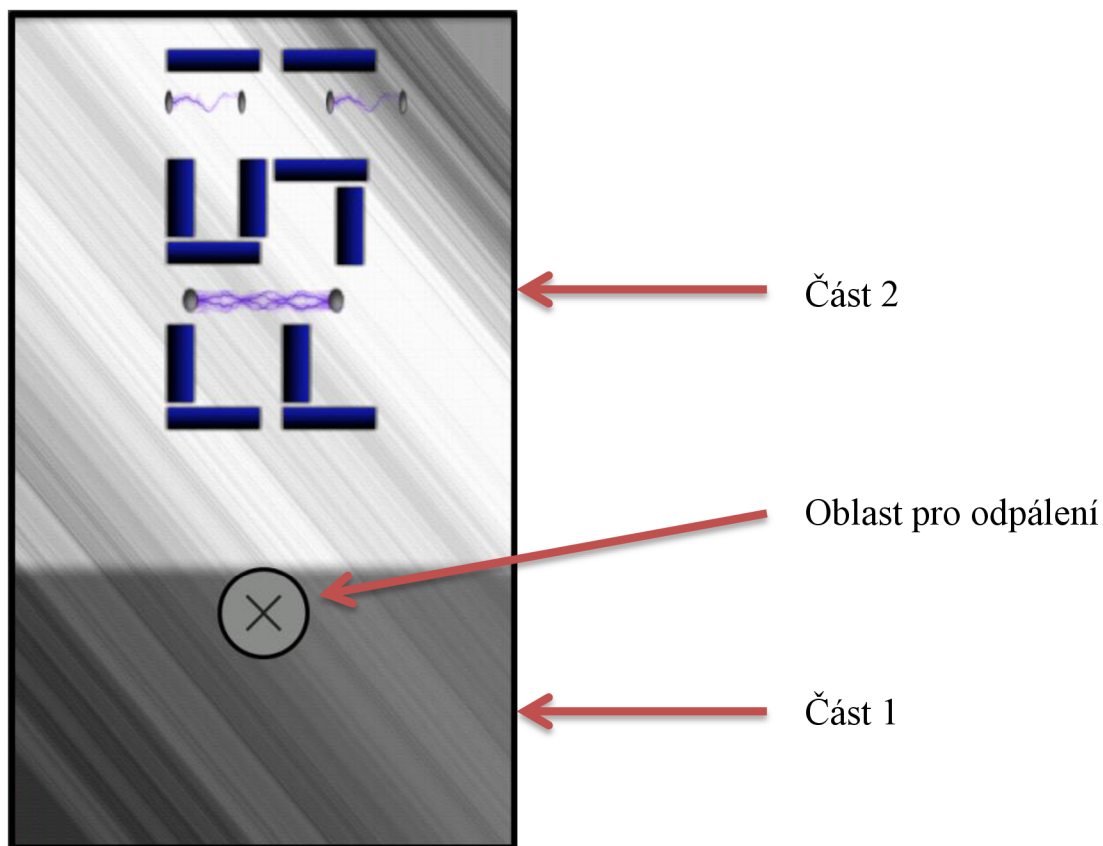
V mé aplikaci používám explicitní intenty, protože vyžadují spuštění určitých aktivit. Využívám intenty ke spuštění obrazovek o výsledku hry a také k následnému spuštění hry od začátku.

## 4.8 Vlastní aplikace

Hra nazvaná B\_game je jednoduchou aplikací skládající se z animace padajících objektů a vystřelovacího míčku, který objekty ničí. Pro ztížení sestřelování jsou mezi objekty zábrany, které míček eliminují. Úkolem hráče je zničit všechny objekty dříve, než zábrana dopadne na místo vystřelování. Hrací plocha zobrazena na obrázku Obr.12.

Při programování aplikace jsem využil některé funkce z Android SDK. Například:

- onTouchListener() – tato metoda je implementována přímo v hlavní třídě MainActivity a umožní volat funkci onTouch(), která reaguje na dotyk obrazovky,
- MotionEvent – s pomocí této metody se nastavují akce, při dotyku obrazovky
  - MotionEvent.ACTION\_DOWN – se spustí, pokud je obrazovka stlačena,
  - MotionEvent.ACTION\_UP – se spustí, při uvolnění
- SurfaceView – je součástí třídy funkcí android.view.View. Zajišťuje, aby se daný program zobrazil na vymezené obrazovce, kterou má k dispozici. Pomocí rozhraní SurfaceHolder se nastavují parametry pro SurfaceView.
- Canvas – obsahuje metody pro vykreslování a práci s bitmapovou grafikou
- BitmapFactory – umožní nahrát bitmapy
- SoundPool – spouští a nastavuje hudbu

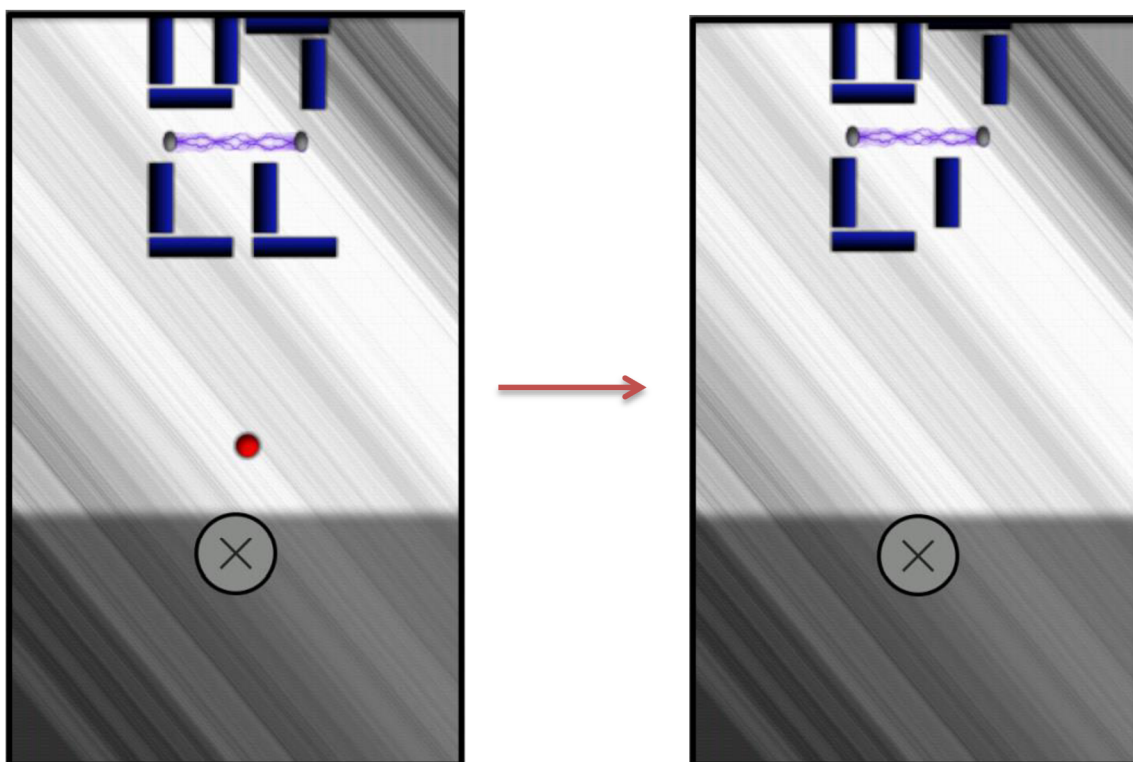


Obr.12 Hrací plocha hry

Hrací plocha je rozdělena vodorovně na 2 části. V části číslo 1 se odehrává vystřelování míčku a v části číslo 2 dochází k padání objektů a kontaktům míčku s objekty nebo zdmi.

K vystřelení dojde po dotyku obrazovky v oblasti pro odpálení a následném natažení míčku směrem dolů (v oblasti části 1) s trajektorií, po které míček následně vystřelí. Při opětovném dotyku v části 1 míček zmizí.

Padající objekty jsou poskládané tak, aby tvořili zkratku FSI pro propagaci fakulty strojního inženýrství.



*Obr.13 a 14 Sestřelení objektu*





## 5 Závěr

Android je velice zajímavý systém, který poskytuje mobilním zařízením spoustu možností a volnosti. Oblíbenost tohoto softwaru je následkem především jeho kvality, dostupnosti.

Cílem práce bylo zmapovat platformu Android a seznámit se s nástroji pro vývoj aplikací a jejich praktické vyzkoušení. První část se zabývá operačním systémem, jeho historií a architekturou. Popsány jsou také různé verze Androidu, jejich podíl na trhu, licencování a konkurence. V druhé části jsou popsány nástroje potřebné pro vývoj aplikací, jejich porovnání a možnosti využití. Poslední část práce obsahuje potřebné informace pro samotné programování a popis vývoje jednoduché aplikace ve formě hry. Tato hra je k dispozici na přiloženém DVD i se zdrojovými kódy a soubory.

Pro vývoj lepších a kvalitnějších aplikací je potřebná důkladná znalost systému Android a hlavně vývojových nástrojů. Důležité jsou pokročilejší schopnosti a zkušenosti v oblasti programování.



## Seznam použité literatury

- [1] What would it take to build a better mobile phone?. Open Handset Alliance [online]. 18.6 2011 [cit. 2013-05-22]. Dostupné z: <http://www.openhandsetalliance.com/>
- [2] Android (operating 43systém). In: Wikipedia: the free encyclopedia [online]. Aktualizováno 21.5.2013 v 08:06. San Francisco (CA): Wikimedia Foundation [cit. 2013-05-22]. Dostupné z: [http://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))
- [3] Android Architecture – The Key Concepts of Android OS. In: Android-App-Market [online]. 17.2.2012 [cit. 2013-05-22]. Dostupné z: <http://www.android-app-market.com/android-architecture.html>
- [4] Android (operační systém). In: Wikipedia: the free encyclopedia [online]. Aktualizováno 14. 5. 2013 v 08:07. San Francisco (CA): Wikimedia Foundation [cit. 2013-05-22]. Dostupné z: [http://cs.wikipedia.org/wiki/Android\\_\(operační\\_systém\)](http://cs.wikipedia.org/wiki/Android_(operační_systém))
- [5] Dashboards. In: Android Developers [online]. 1.2.2013 [cit. 2013-05-22]. Dostupné z: <http://developer.android.com/about/dashboards/index.html>
- [6] GNU General Public License (Všeobecná veřejná licence GNU). In: GNU Operating System [online]. © 1999 [cit. 2013-05-22]. Dostupné z: <http://www.gnu.org/licenses/gpl-2.0.html>
- [7] Apache License, Version 2.0. In: The Apache Software Foundation [online]. © 2012 [cit. 2013-05-22]. Dostupné z: <http://www.apache.org/licenses/LICENSE-2.0.html>
- [8] Develop Apps for iOS. In: Develop for iOS – Apple Developer [online]. © 2013 [cit. 2013-05-22]. Dostupné z: <https://developer.apple.com/technologies/ios/>
- [9] Windows Phone [online]. © 2013 [cit. 2013-05-22]. Dostupné z: <http://www.windowsphone.com/cs-cz>
- [10] BlackBerry [online]. © 2013 [cit. 2013-05-21]. Dostupné z: <http://us.blackberry.com/>
- [11] SymbianCentrum [online]. © 2009-2012 [cit. 2013-05-22]. Dostupné z: <http://www.symbian-nokia.cz/>
- [12] Firefox OS. Mozilla [online]. ©1998–2013 [cit. 2013-05-21]. Dostupné z: <http://www.mozilla.org/en-US/firefox/partners/>
- [13] Mobile operating 43systém. In: Wikipedia: the free encyclopedia [online]. 20.5 2013 v 12:30. San Francisco (CA): Wikimedia Foundation, [cit. 2013-05-21]. Dostupné z: [http://en.wikipedia.org/wiki/Mobile\\_operating\\_system](http://en.wikipedia.org/wiki/Mobile_operating_system)

- [14] Android Developer Challenge. Google Developers [online]. © 2008 [cit. 2013-05-21]. Dostupné z: <https://developers.google.com/android/adc/?hl=cs>
- [15] Get Started with Publishing. Android Developers [online]. © 2013 [cit. 2013-05-21]. Dostupné z: <http://developer.android.com/distribute/googleplay/publish/register.html>
- [16] Google Play. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation [cit. 2013-05-22]. Dostupné z: [http://cs.wikipedia.org/wiki/Google\\_Play](http://cs.wikipedia.org/wiki/Google_Play)
- [17] PLUTA, Joe. Eclipse step by step. Lewisville, TX: MC Press Online, 2003. ISBN 978-158-3470-442.
- [18] ADT Plugin. Android Developer [online]. © 2013 [cit. 2013-05-21]. Dostupné z: <http://developer.android.com/tools/sdk/eclipse-adt.html>
- [19] SDK Tools. In: Android Developers [online]. © 2013 [cit. 2013-05-22]. Dostupné z: <http://developer.android.com/tools/sdk/tools-notes.html>

**Obsah příloh (DVD)**

- Elektronická verze práce
- Eclipse projekt (složka zkomprimována v souboru Bakalarka.rar)
- Bakalarka.apk (soubor spustitelný v systému Android)
- Návod na spuštění aplikace