|  |  |  |
| --- | --- | --- |
| Logo__SSPU_2016_Barva | | |
| **Závěrečná studijní práce**  **dokumentace** | | |
| **Mobilní informační aplikace** | | |
| Michael Hluchník | | |
|  | | |
|  | |  |
| **Obor:** | 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  se zaměřením na počítačové sítě a programování | |
| **Třída:**  **Školní rok:** | IT4  2018/2019 | |

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité   
informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední průmyslové   
a umělecké škole v Opavě, Praskova 399/8.

V Opavě 31. 12. 2018

*podpis autora práce*

**ANOTACE**

Cílem projektu bylo vytvořit mobilní aplikaci vypisující informace o daných předmětech ze hry Warframe. Zajišťuje jednoduché a přehledné vypisování dat, aby uživatel neměl problém najít jakýkoliv předmět ve hře. Aplikace byla vyvinuta pomocí jazyka Java ve vývojovém prostředí Android studio. Data jsou uložena v podobě JSON souborů. Tyto soubory jsou uloženy na internetových stránkách GitHub. Aplikaci tvoří čistě informační vzhled. Aplikace obsahuje tří hlavní stránky, mezi kterými se uživatel přemisťuje dle potřeby. Prioritou při tvorbě aplikace byl výsledný vzhled aplikace, aby byla přehledná a jednoduše ovladatelná. Aplikaci lze využívat ve všech mobilních zařízeních, které mají operační systém Android 5 a vyšší.

**Klíčová slova:** mobilní aplikace, Android Studio, JSON soubory, Java

**ANNOTATION**

The goal of the project was to create a mobile application listing information items from the game Warframe. It provides a simple and clear listing of the data so that the user does not have any problems finding any object in the game. The application was developed using Java using the Android Studio IDE. Data is stored as JSON files. JSON files are stored on GitHub. The application is purely informative. The application contains three main pages, among which the user moves as needed. The main priority of the application’s development was its resulting design and that it be easy to use. The application can be used on all mobile devices that run Android 5 or higher.

**Keywords:** mobile app, Android Studio, JSON files, Java

OBSAH

[Úvod 5](#_Toc535232737)

[1 Teoretická a metodická východiska 6](#_Toc535232738)

[1.1 Začátky v Android Studiu 6](#_Toc535232739)

[1.1.1 Emulátor nebo vlastní zařízení? 6](#_Toc535232740)

[1.2 „Activity“ 6](#_Toc535232741)

[1.3 Soubory JSON 6](#_Toc535232742)

[1.4 AsyncTask 7](#_Toc535232743)

[2 Využité technologie 8](#_Toc535232744)

[2.1 Android studio 8](#_Toc535232745)

[2.2 Java 8 8](#_Toc535232746)

[2.3 Android OS 8](#_Toc535232747)

[2.4 SDK 26.1.1 9](#_Toc535232748)

[3 Způsoby řešení a použité postupy 10](#_Toc535232749)

[3.1 Začátky v Android Studiu 10](#_Toc535232750)

[3.1.1 Nastavení pro vývojáře v mobilu 10](#_Toc535232751)

[3.2 Vytvoření projektu 10](#_Toc535232752)

[3.3 Struktura 11](#_Toc535232753)

[3.4 Activity main 12](#_Toc535232754)

[3.4.1 ListView 12](#_Toc535232755)

[3.4.2 Vytvoření nové activity 12](#_Toc535232756)

[3.4.3 Přepínaní mezi aktivitami 13](#_Toc535232757)

[3.4.4 Posílání proměnné mezi aktivitami 13](#_Toc535232758)

[3.5 JSON parsing 13](#_Toc535232759)

[3.5.1 Struktura JSON souboru 14](#_Toc535232760)

[3.5.2 Přístup k internetu 14](#_Toc535232761)

[3.5.3 Parsování 15](#_Toc535232762)

[3.6 Vypisování 15](#_Toc535232763)

[4 Výsledky řešení, výstupy, uživatelský manuál 16](#_Toc535232764)

[Závěr 17](#_Toc535232765)

[Seznam použitýCH INFORMAČNÍCH ZDROJů 18](#_Toc535232766)

Úvod

Warframe Codex je mobilní aplikace naprogramována pomocí jazyka Java. Tato aplikace vypisuje informace o jednotlivých předmětech ze hry Warframe a byla vytvořena ve vývojovém prostředí Android Studio.

Cílem tohoto projektu bylo získání dat v podobě JSON souborů z internetu a jejich následné vypisování v přehledné a jednoduché formě. To vše v mobilní aplikaci.

Tato dokumentace se bude zabývat postupy, které byly při tvorbě tohoto dokumentu použity a technologiemi, které byly využity. Vytvářením projektu v Android Studiu. Řešením problematiky získávání a parsování JSON souborů z internetu a jejich následné přehledné vypisování.

# Teoretická a metodická východiska

## Začátky v Android Studiu

Android Studio je vývojové prostředí zaměřené přímo na vývoj mobilních aplikací, z tohoto důvodu bylo vybráno pro tento projekt.

Je zde také Visual Studio, které má také možnost vytváření mobilních aplikací, ale z důvodu náročnějšího ovládání a velkého využití paměti bylo od něj upuštěno.

### Emulátor nebo vlastní zařízení?

Samotný emulátor je jednoduchý na spuštění a ovládání, ale bohužel není vhodný pro všechny počítače, kvůli jeho náročnosti na hardware.

Vlastní mobilní zařízení si musíte nastavit do režimu pro vývojáře a propojit s počítačem. Poté si ho musíte přepnout na přenos souborů. Nadále by si Android Studio mělo zjistit přístupné mobilní zařízení a s vaším dovolením program na něm spustit.

## „Activity“

„Activity“, je to, co uživatel vidí. Je tvořena dvěma soubory – „class“ a „layout“. „Class“ vykonává úkony a provádí operace. „Layout“ se zabývá grafickou stránkou. „Activity“ jako celek komunikuje s uživatelem a vytváří okna. Většinou je „Activity“ prezentována jako celá obrazovka, ale může mít i jiné použití, jako třeba plovoucí obrazovka či přímo vložená do jiné „Activity“.

## Soubory JSON

JSON (JavaScript Object Notation) je způsob zápisu dat nezávislý na počítačové platformě. Je určený pro přenos dat, která mohou být organizována v polích nebo agregována v objektech. Používají se pro přenos dat. Je člověkem čitelný a jednoduše editovatelný.

Pro práci byla zvolena tato metoda díky jednoduché manipulaci s daty a jednoduchosti čtení v jazyce Java.

## AsyncTask

Generická abstraktní třída „AsyncTask“ umí asynchronně spustit jiný kód. Užitečná je zejména v tom, že se nemusí ručně vytvářet nová vlákna a starat se o ně. Využívá se většinou při náročnějších operacích. Využity jsou tyto funkce.

Funkce „onPreExecute()“ se volá před započetím práce asynchronního úkolu, zde se vytváří progres dialogu, který informuje uživatele o probíhající činnosti. Ve funkci „doInBackground“ probíhá výpočet. A ve funkci „onPostExecute()“ se zpracovává vrácený výsledek. „AsyncTask“ se spouští zavoláním funkce „execute()“ s parametry.

# Využité technologie

## Android Studio

Android studio je vývojové prostředí založené na IntelliJ IDEA, které podporuje programovací jazyky jako je Java a Groovy. Také podporuje programovací jazyky C++ a Kotlin. Pro tento projekt byl vybrán jazyk Java. Byl vytvořen firmou Google pro platformy Windows, Mac OS X a Linux. Android studio obsahuje mnoho různých funkcí, které usnadňují programování. Mezi užitečné nástroje patří například „debugger“, který testuje vytvořený program před tím, než je nahrán do procesoru.

Toto vývojové prostředí bylo vybráno kvůli jednoduchosti a na doporučení jiných uživatelů.

## Java 8

Je objektově orientovaný [programovací jazyk](https://cs.wikipedia.org/wiki/Programovac%C3%AD_jazyk), který vyvinula firma Sun Microsystems. Java je nejpopulárnější programovací jazyk. Díky své [přenositelnosti](https://cs.wikipedia.org/wiki/P%C5%99enositelnost) je používán pro [programy](https://cs.wikipedia.org/wiki/Po%C4%8D%C3%ADta%C4%8Dov%C3%BD_program), které mají pracovat na různých systémech počínaje čipovými kartami přes [mobilní telefony](https://cs.wikipedia.org/wiki/Mobiln%C3%AD_telefon) a různá zabudovaná zařízení. Nebo také pro aplikace pro desktopové počítače až po rozsáhlé distribuované systémy pracující na řadě spolupracujících počítačů rozprostřené po celém světě. Je také multiplatformní.

## Android OS

Je [mobilní operační systém](https://cs.wikipedia.org/wiki/Mobiln%C3%AD_opera%C4%8Dn%C3%AD_syst%C3%A9m) založený na [jádře Linuxu](https://cs.wikipedia.org/wiki/Linux_(j%C3%A1dro)), který je dostupný jako otevřený software. Je používán na smartphonech, tabletech, [chytrých televizích](https://cs.wikipedia.org/wiki/Smart_TV) a dalších zařízeních. Jeho vývoj vede firma [Google](https://cs.wikipedia.org/wiki/Google) pod hlavičkou konsorcia firem Open Handset Alliance. Má mnoho verzí, podle kterých se odvádí název. Systém Android má největší zastoupení na světě mezi všemi operačními systémy.

Vytváření aplikace pro tento operační systém byla vybrána z důvodu vlastnictví mobilu, který užívá tento operační systém. Projekt je kompatibilní s operačním systémem android 5.0 a vyšší.

## SDK 26.1.1

Software development kit (SDK)je typická sada vývojových nástrojů umožňující vytváření aplikací pro jisté softwarové balíčky, pro určitý počítačový framework, platformy, počítačové systémy, herní konzole, operační systémy nebo podobnou platformy. Vlastní aplikace se bez SDK neobejde. Většinou jsou přiděleny k přímo k vývojovému prostředí.

# Způsoby řešení a použité postupy

## Začátky v Android Studiu

První zkušenost s Android Studiem, bylo potřeba se s ním nejprve seznámit. Jsou zde tři možnosti zkontrolování kódu:

* **Make Project** – tlačítko ve tvaru kladívka, které kontroluje kód, jestli je bezchybný. Ale samotný program to nespustí.
* **Debug** – zubaté kolečko s trojúhelníčkem, který vypisuje všechny atributy a pomáhá najít chybu.
* **Run** – velký zelený trojúhelník spouští program a zároveň kontroluje kód, který běží buď na emulátoru, nebo na mobilním zařízení.

### Nastavení pro vývojáře v mobilu

Pro vývoj aplikace pro vlastní mobilní zařízení, je potřeba mobilní telefon takzvaně přepnout do režimu „Pro vývojáře“. Toto umožňuje instalovat a zkoušet aplikaci v mobilním zařízení. V nastavení telefonu je potřeba najít informace o zařízení. Tam je pak zapotřebí 6krát rychle kliknout na číslo sestavení daného telefonu. To by mělo umožnit přepnutí režimu do režimu pro vývojáře.

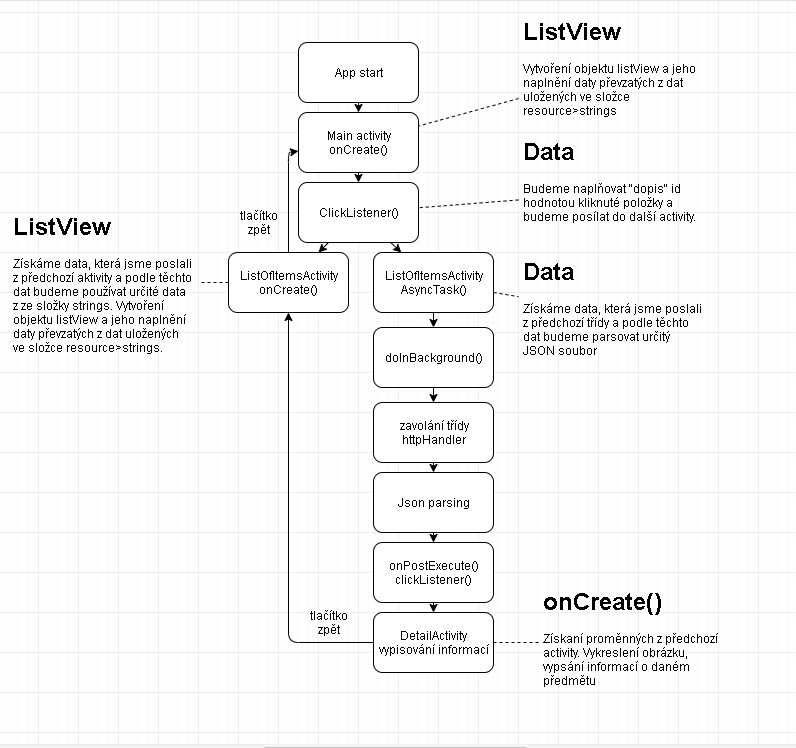
## Vytvoření projektu

Při vytváření nového projektu je důležité si vybrat správně, pro jaký typ operačního systému Android ho vytváříme. Samotné Android Studio nám dá na výběr z mnoha možností a samo nám doporučí ty, které se nejvíce používají. Ale pozor, pokud chcete mít aplikaci pro vlastní použití, tak váš mobilní telefon nemusí mít stejnou verzi, kterou vám studio doporučí. Z tohoto důvodu je potřeba si toto pohlídat a zkontrolovat v nastavení telefonu.

Poté je potřeba vybrat, jak by měla úvodní obrazovka vypadat. Zobrazí se okno s graficky ukázanou nabídkou úvodních obrazovek. Pro tento projekt byla vybrána prázdná úvodní obrazovka.

Jakmile je vytvořen projekt, je k dispozici „class mainActivity“ a její „layout activity\_main“.

## Struktura



Obr. 1 Struktura JSON

## Activity main

### ListView

„ListView“ je skupina zobrazení, která zobrazuje seznam rolovacích položek. Položky seznamu se automaticky vkládají do seznamu pomocí adaptéru, který přenáší obsah ze zdroje. Pracuje podobně jako pole nebo databázový dotaz a konvertuje každý výsledek položky do zobrazení, které je umístěno do seznamu.

Pro tuto práci byl zvolen „ListView“, který byl graficky vložen do „layout main\_activity“. Pomůže jednoduše rolovat mezi položkami, které jsou v něm vložené. Poté byla vytvořena „class Adapter“, která pomáhá v zobrazení položek v „ListView“. V „activity main“ si vytvoříme funkci, která bude reagovat na kliknutí na tyto položky a odkazuje na další „activity“. Tato funkce už je předdefinována v jazyce Java.

Je potřeba si tento objekt ve třídě inicializovat:

1. **private** ListView itemsListView;
2. itemsListView = findViewById(R.id.itemsListView);

### Vytvoření nové activity

Vytvoření nové „activity“ je jednoduchá záležitost. Stačí pravým kliknutím myši na danou záložku vybrat „new“ > „activity“ a potom si vybrat, zda chceme prázdnou či nějak předem upravenou „activity“. Pro tento projekt jsem si vybral prázdnou.

Do této „activity“ si taktéž graficky přidáme „ListView“. Samozřejmě mu dáme ID odlišné od minulého „ListView“.

### Přepínaní mezi aktivitami

Toto je jedna z nejdůležitějších akcí ve všech projektech, které se týkají mobilních aplikací. Její syntaxe je velice jednoduchá a rychlá. Nejprve je potřeba ji inicializovat.

1. Intent showDetailActivity = **new** Intent(getApplicationContext(), DetailActivity.**class**);

A poté spustit.

1. startActivity(showDetailActivity);

Tento kousek kódu byl vložen do funkce, která reaguje na kliknutí.

### Posílání proměnné mezi aktivitami

U této činnosti je potřeba dávat pozor na správné indexovaní proměnných. Tato činnost má dvě fáze. Nejprve si v „activity“ ze které chceme posílat, vložíme následující ukázku kódu, který umožňuje přeposílání proměnných. V tomto projektu to bude konkrétně ve funkci při kliknutí.

1. showDetailActivity.putExtra("com.example.michal.SHIELD", shields);

Ve druhé „activity“ tento dopis vybalíme. Lze posílat jakýkoliv typ proměnných, avšak pak je potřeba přizpůsobit i kód. A to změnou v názvu a atributech funkcí „getExtra()“.

1. Intent in = getIntent();
2. **int** index = in.getIntExtra("com.example.michal.ITEM",-1);

Byly získávány v rozdělovači „switch“, protože jsem si posílal hodnoty, které byly různé podle toho na jakou položku z „ListView“ klineme. Toto všechno bylo zkoušeno pomocí obyčejných proměnných, které byly vytvořeny jako testovací verze.

## JSON parsing

Jazyk JAVA má knihovny určené k parsování JSON souborů a proto bylo očekáváno, že to bude jednoduchá a rychlá záležitost. Bohužel tomu tak nebylo. V rámci práce se objevila problematika která, určovala, kde by se to mělo „parsovat“, aby se to už zobrazilo přesně tam, kde bylo potřeba a kdy bylo potřeba. Nakonec byla zvolena asynchronní funkce. JSON soubory byly uloženy na internetových stránkách, díky čemuž se objevila zcela jiná komplikace, která musela být vyřešena.

### Struktura JSON souboru

Data byla převzata z internetových stránek nebo jiných JSON souborů, které jsou zprostředkovány na internetových stránkách. Struktura JSON souborů je velmi primitivní a skládá se jenom z jednoduchých objektů, atributů a jejich hodnotami.

1. [{
2. "name": "Adarza Kavat",
3. "description": "Piercing eyes reflect deadly instincts.",
4. "health": 40,
5. "shield": 70,
6. "armor": 50,
7. "stamina": 8,
8. "power": 100,
9. "type": "Pets",
10. "imageName": "adarza-kavat.png"
11. },...]

### Přístup k internetu

Aby aplikace mohla přistupovat k internetu a využívat informace z něho získané musíme si toto povolit v souboru „AndroidManifest.xml“. Příkazem:

1. <uses-permission android:name="android.permission.INTERNET" />

Poté byla vytvořena třída, která zajistila ověření internetové adresy a zajistila přijetí dat z JSON souboru v datovém typu string.

### Parsování

Samotné „parsování“ je řešeno v druhé „activity“, kde pro ni byla vytvořena asynchronní funkce.

V asynchronní funkci „DoInBackground“ si inicializujeme třídu, která nám zajišťuje komunikování s internetovou adresou. Pokud máme více stránek s JSON soubory můžeme si je posílat do této třídy postupně pomocí switche či jiného přepínače. V této práci je více souborů JSON a proto je tato funkce využívána.

Poté si pomocí předdefinovaných funkcí v jazyce JAVA na parsování JSON souboru jako jsou „JSONArray();“ „JSONObject();“ a opakovače „for ()“, rozdělíme tento datový typ string podle toho, jak potřebujeme. Jako například:

1. JSONArray frames = **new** JSONArray(jsonStr1);
2. **for**(**int** i =0;i < frames.length();i++){
3. JSONObject c = frames.getJSONObject(i);
4. String name = c.getString("name");
5. }

Nakonec jsou tyto data přeposlána, kde potřebujeme ve funkci „onPostExecute()“. Kde je další funkce při kliknutí na položky v „listView“ a až v ní jsou posílány informace dál. Poté je volána třetí „activity“.

## Vypisování

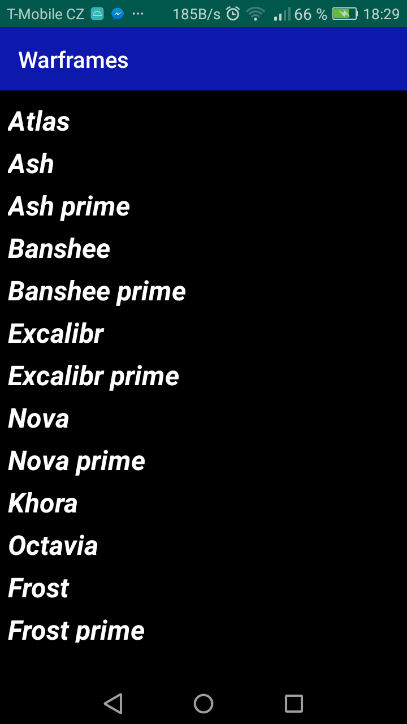
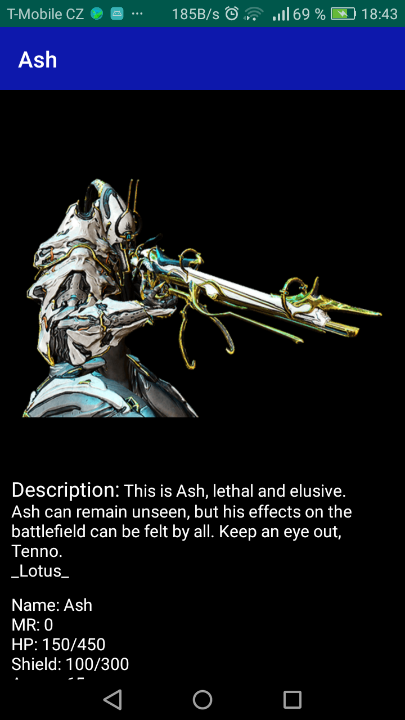
Ve třetí „aktivity“ je přidán obrázek, který zobrazuje danou položku a obyčejný „textView“, ve kterém se vypisuji informace o této položce.

# Výsledky řešení, výstupy, uživatelský manuál



Obr. 2 Úvodní stránka

Při zapnutí aplikace se načte tato stránka. Tady je potřeba vybrat, jaký typ položky chceme hledat pouhým kliknutím na název.

Obr. 3 List předmětů Obr. 4 Výpis dat

Následně se přemístíme na obrazovku s předměty, které máme v souboru JSON. Zároveň se JSON soubor začne stahovat. Pokud máme přístup k internetu, lze okamžitě kliknout na určitý předmět a přesunout se na obrazovku vypisování údajů ze souboru JSON.

# **Závěr**

Cílem tohoto projektu bylo vytvoření funkční mobilní informační aplikace. Tuto aplikaci jsem navrhl pro vlastní potřebu, abych měl rychlejší a jednoduší nalezení daných informací, když je budu potřebovat. Určené cíle jako zobrazení dat z JSON souborů, které byly převzaty z internetu a jejich přehledné vypisování bylo splněno.

Do budoucna by bylo možné přidat přihlašovací systém, který by byl propojen s přihlašovacím systémem hry Warframe.

Seznam použitýCH INFORMAČNÍCH ZDROJů

[1] BUTTERFIELD, Bill. Android Studio For Beginners Part 1. *Youtube.com* [online]. 13. 6. 2017 [cit. 2018-10-08]. Dostupné z: <https://www.youtube.com/watch?v=dFlPARW5IX8>

[2] BUTTERFIELD, Bill. Android Studio For Beginners Part 2. *Youtube.com* [online]. 17. 4. 2017 [cit. 2018-10-08]. Dostupné z: <https://www.youtube.com/watch?v=6ow3L39Wxmg&t>

[3] BUTTERFIELD, Bill. Android Studio For Beginners Part 3. *Youtube.com* [online]. 9. 8. 2017 [cit. 2018-10-08]. Dostupné z: https://www.youtube.com/watch?v=rdGpT1pIJlw&t

[4] PANWAR, Abhishek. JSON DATA FETCHING AND PARSING FROM URL ANDROID STUDIO TUTORIAL | Developing an ANDROID APP 2017. *Youtube.com* [online]. 14. 7. 2017 [cit. 2018-11-08]. Dostupné z: https://www.youtube.com/watch?v=Vcn4OuV4Ixg&t

[5] Android - JSON Parser. *Tutorialspoint.com* [online]. [cit. 2018-11-08]. Dostupné z: https://www.tutorialspoint.com/android/android\_json\_parser.htm

[6] *Cs.wikipedia.org* [online]. [cit. 2018-12-08]. Dostupné z: https://cs.wikipedia.org/wiki/Hlavn%C3%AD\_strana