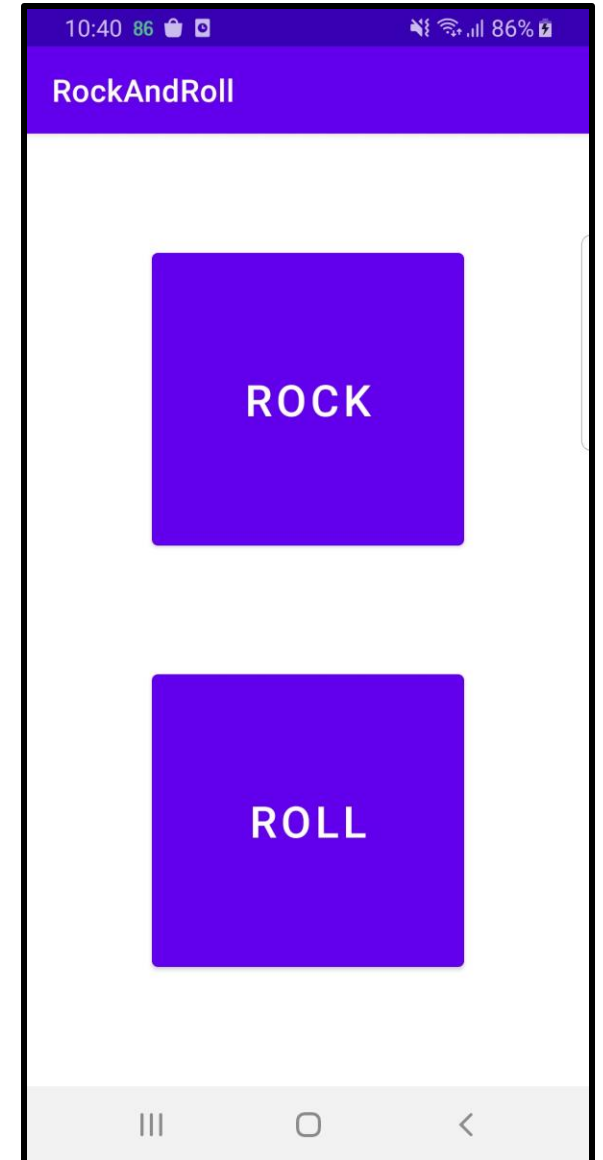


Rock and Roll

Prvá aplikácia, úvod do programovania
Android aplikácie v Java.



O čom by boli úvodné slajdy

- História Androidu
- Podiel na trhu
- Štatistika: Koľko telefónov je na svete
- Dalvik (optimalizovaný virtuálny stroj)
- Rôzne verzie SDK
- Pohľad do útrob až k linux kernelu a možnosti písať kód aj v C++
- Kotlin vs. Java
- Publikovanie na Google Play
- APK, podpis aplikácie

Verzie SDK

- Snaha Androidu o **kompatibilitu** (aj forward nie len spätnú)
- **min <= target <= compile**
- **CompileSdkVersion** – použité pri kompilácii, odporúča sa najnovšia verzia, zmena neovplyvní správanie aplikácie za behu
- **MinSdkVersion** – minimálne požiadavka overovaná Google Play
- **TargetSdkVersion** - voči akej verzii je appka testovaná. Ak je API na zariadení väčšie, Android môže spustiť *compatibility behavior*
- Ideálne: min (lowest possible) <= target == compile (latest SDK)

Štruktúra projektu

- **manifest**
 - xml súbor, základné nastavenia projektu
 - definuje, ktorá aktivita je pri spustení
- **res**
 - resources – xml alebo iné (png a pod.)
 - layout – rozloženie widgetov v aktivite
 - values – stringy, štýly, farby ...
- **java**
 - zdrojové kódy

#1 Layout - pridávanie tlačidiel

- Preskúmať existujúci **TextView** *Hello world* a vymazať ho
- Pridať nový **Button** a nastaviť vhodné atribúty (rozmery, constraints, text)
- Druhý **Button** a spoločné uchytenie – potrebné ID
- Vložiť text do **strings.xml**

Button widget

- **ConstraintLayout**

- vyžaduje constraints – uchytenie k niečomu (okraj obrazovky/rodiča)

- veľkosť tlačidla – 200dp, veľkosť textu – 28sp

- **px** – pixel (rôzna veľkosť na rôznych zariadeniach)
- **dp** – density-independent pixel
- **sp** – scale-independent pixel

- **onClick** – obslužná metóda

- text – string vyextrahovaný do **/res/strings.xml**

#2 Prepojenie XML a Java kódu

- button **ROLL**
- pridať **onClick** metódu v XML
- v Jave vytiahnúť referenciu na **Button** cez **findViewById**
- implementovať metódu v Jave – nastaviť príslušný text s náhodným číslom

~~Metóda onClick~~

- **android:onClick="roll"**
 - definované v layoute
- **public void roll(View view) { }**
 - definované v aktivite
- musia mať zhodný názov
- metóda musí byť public a bez návratového typu
- metóda má iba 1 parameter – View na ktorý sa kliklo
- **DEPRECATED od API 31** - náhrada **OnClickListener**

Prepojenie layoutu a aktivity

- Layout
 - widget má ID – `@+id/buttonRoll`
 - `@` hovorí, že ide o referenciu
 - `+` určuje, že sa vytvára nová referencia
- Aktivita
 - `findViewById(R.id.buttonRoll)`
 - ID sa získa cez triedu R
 - button setText
 - v onClick metóde
 - parameter **String** nastaví text
 - parameter **int** – ID stringu z /res/strings.xml

R

- Triedy (`R.id`, `R.string`, `R.layout ...`) generované automaticky pri buildovaní projektu
- `public static final int` premenné
- Manažment resources
 - Jazykové verzie
 - Konfigurácie - šírka obrazovky, natočenie...

#3 Resources a životný cyklus aktivity

- vložíme nejaký mp3 súbor do **/res/raw** – ak to nie je vytvorené, tak klikneme na res: **New->Android Resource Directory** a vytvoríme raw folder
- **Media Player** – poctivá práca s tým je komplikovanejšia. Kde pôjdu jednotlivé metódy create, start a release?
- vyskúšajme logovanie a sledujme životný cyklus aktivity

Resources - zvuk

- umiestnime do `/res/raw`
- inicializácia v `onResume`
 - `MediaPlayer.create(this, R.raw.sound);`
- odpojenie v `onPause`
 - `mediaPlayer.release();`
- spustenie prehrávania v `onClick` metóde
 - `mediaPlayer.start();`

Logovanie

- Logcat (okno v Android Studio) – zobrazuje logy
- `verbose`, `debug`, `info`, `warn`, `error`, `assert`
- `Log.d(tag, message)`
- Tipy:
 - `if (BuildConfig.DEBUG)` – debug verzia aplikácie
 - `getClass().getName()` – názov triedy ako tag
 - filtrovanie správ v logcat
 - logcat screenshot nástroj

Životný cyklus aktivity

onCreate()

onStart()

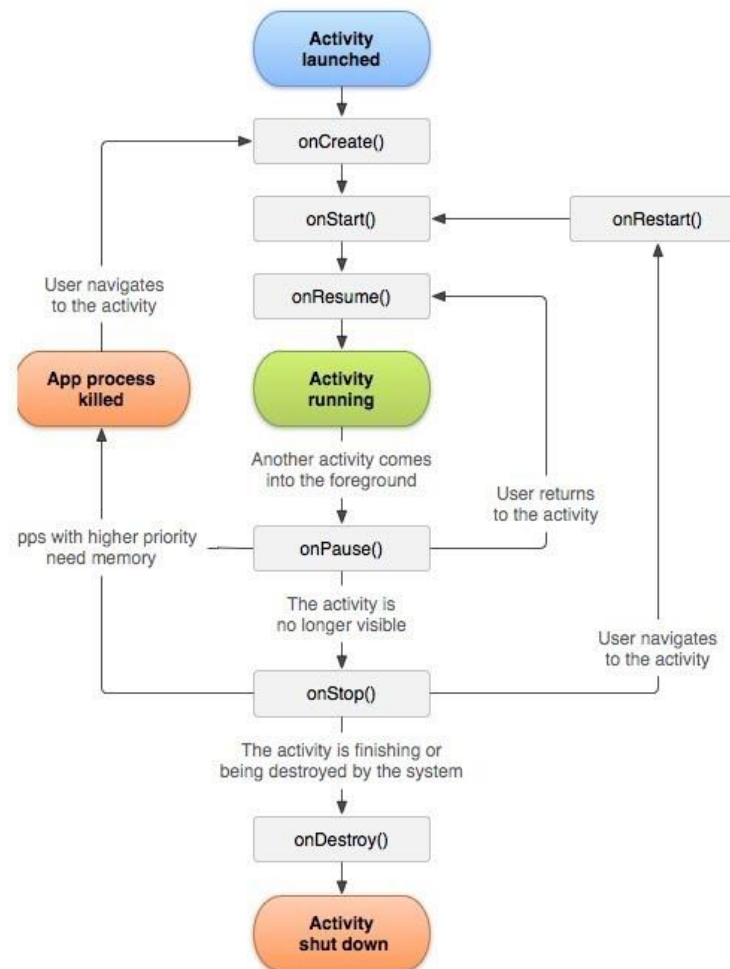
onRestart()

onResume()

onPause()

onStop()

onDestroy()



Gradle



- Beží nezávisle od Android studia - dá sa buildovať cez príkazový riadok
- Groovy alebo Kotlin
- Android projekt – aj cez nastavenia
- **Verzie SDK**
- **Dependencies**
- **Build nastavenia**