Android – II. előadás

Miről lesz ma szó?

- Egy Android alkalmazás felépítése
- XML 101
- A manifest fájl
- Erőforrások
- Egy tipikus alkalmazás komponensei
- Egy Activity élete és halála

Miből állhat egy Androidos alkalmazás?

- Mire van szükség?
  - Facebook, Gmail, Naptár app
  - Soroljatok még alkalmazásokat!
  - Milyen funkcióik vannak?

Az Android alkalmazás komponensei

- Egy Android alkalmazás környezete
- Activity
- Service
- ContentProvider
- BroadcastReceiver

Activity

- Különálló nézet, saját UI-al
- Például:
  - Emlékeztető alkalmazás
  - 3 Activity: ToDo lista, új ToDo felvitele, ToDo részletek
- Független Activity-k, de együtt alkotják az alkalmazást
- Más alkalmazásból is indítható az Activity, például:
  - Kamera alkalmazás el tudja indítani az új Facebook Post Activity-t és a képet hozzá rendeli a poszthoz
- Az android.app.Activity osztályból származik le

Service

- A Service komponens egy hosszabb ideig háttérben futó feladatot jelképez
- Nincs felhasználói felülete
- Például egy letöltő alkalmazás (torrent) fut a háttérben, míg előtérben egy másik programmal játszunk
- Más komponens (pl. Activity) elindíthatja, vagy csatlakozhat (bind) hozzá vezérlés céljából
- Az android.app.Service osztályból kell öröklődnie

ContentProvider

- A Content provider (tartalom szolgáltató) komponens feladata egy megosztott adatforrás kezelése
- Az adat tárolódhat fájlrendszerben, SQLite adatbázisban, web-en, vagy egyéb perzisztens adattárban, amihez az alkalmazás hozzáfér
- A Content provider-en keresztül más alkalmazások hozzáférhetnek az adatokhoz, vagy akár módosíthatják is azokat
- Például: CallLog alkalmazás, ami egy Content provider-tbiztosít, és így elérhető a tartalom
- Az android.content.ContentProviderosztályból származik le és kötelezően felül kell definiálni a szükséges API hívásokat

BroadcastReceiver

- A Broadcast receiver komponens a rendszer szintű eseményekre (broadcast) reagál
- Például: kikapcsolt a képernyő, alacsony az akkumulátor töltöttsége, elkészült egy fotó, bejövő hívás, stb.
- Alkalmazás is indíthat saját „broadcast”-ot, például ha jelezni akarja, hogy valamilyen művelettel végzett (letöltődött a torrent)
- Nem rendelkeznek saját felülettel, inkább valamilyen figyelmeztetést írnak ki például a status bar-ra, vagy elindítanak egy másik komponenst (jeleznek például egy service-nek)
- A android.content.BroadcastReceiver osztályból származik le; az esemény egy Intent (lásd. Később) formájában érhető el

XML 101

- Mindenki látott már HTML-t?
  - Milyen szabályai vannak?

```
1  <html>
2  <body>
3
4  <a href="http://www.w3schools.com">This is a link</a>
5
6  </body>
7  </html>
```

Manifest.xml

- Eddig is volt mindenhol, csak automatikus generált
- Alkalmazás leíró fájl
- Komponensek listája
- Engedélyek
- Min sdk verzió
- App neve, ikonja, sémája

Példa egy manifestre

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.adam.helloworld" >

    <uses-sdk android:minSdkVersion="15" />
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MyActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Erőforrások

- (inkább resources)
- UI elemek
- Nézetek
- Konstansok
- változók

- src
  - androidTest
  - main
    - java
      - com.example.adam.helloworld
        - MyActivity
    - res
      - drawable-hdpi
      - drawable-mdpi
      - drawable-xhdpi
        - ic_launcher.png
      - drawable-xxhdpi
      - layout
        - activity_my.xml
        - hello_world.xml
      - menu
        - my.xml
      - values
        - dimens.xml
        - strings.xml
        - styles.xml
      - values-w820dp
    - AndroidManifest.xml
  - .gitignore
  - app.iml
  - build.gradle
  - proguard-rules.pro

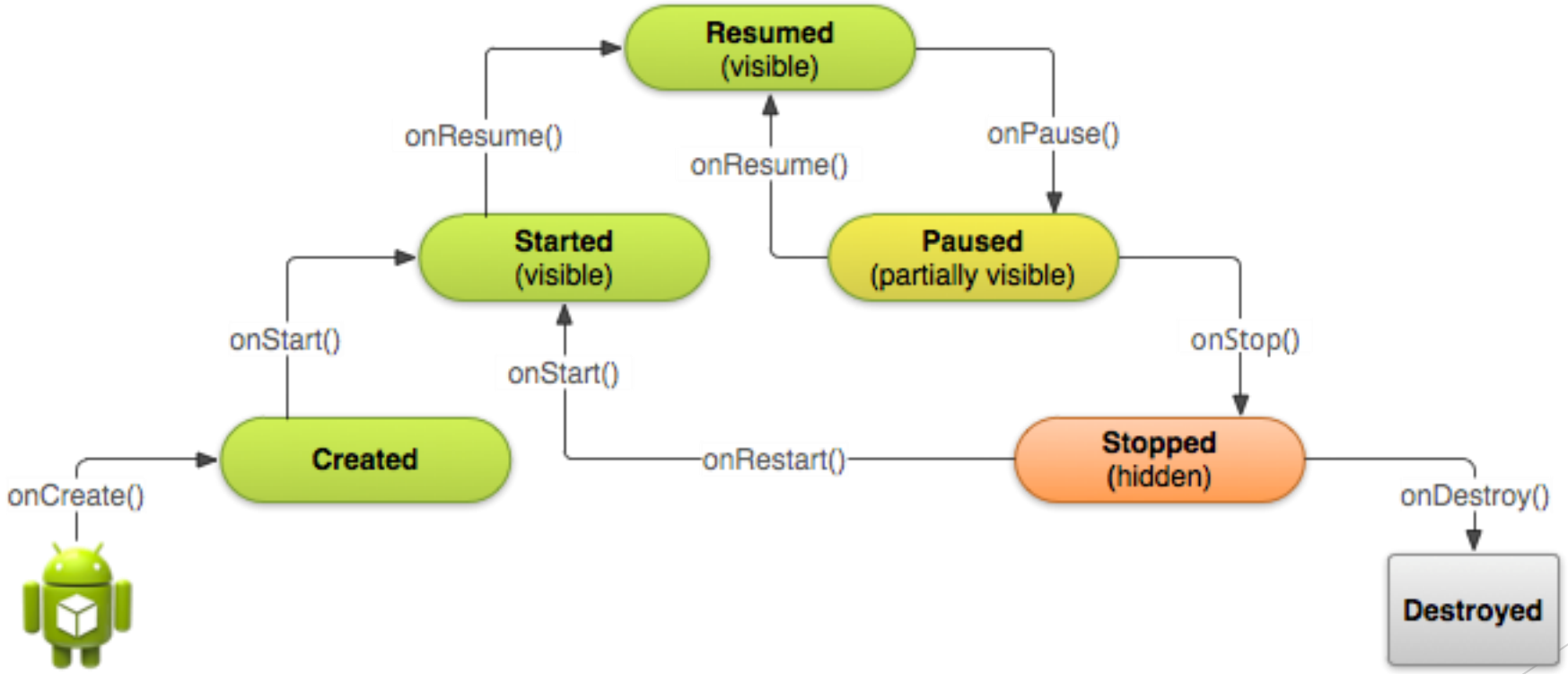
Egy tipikus alkalmazás komponensei

- Néhány Activity
  - Minden külön nézetű képernyőhöz egy-egy
  - Egy indító, fő Activity

Egy alkalmazás indítása

- El lehet indítani másik alkalmazás egy komponensét
- Ez nagy rugalmasságot ad a rendszernek
- Például egy fénykép készítéséhez nem kell új
- Activity-t készíteni, hanem elegendő a kamera alkalmazás kép készítő Activity-jét meghívni
- Hívás után a képet megkapja a hívó alkalmazás
- A felhasználónak úgy tűnik, mintha a kamera funkció az alkalmazás része lenne
- Újrafelhasználhatóság magas fokú támogatása

Egy Activity élete és halála



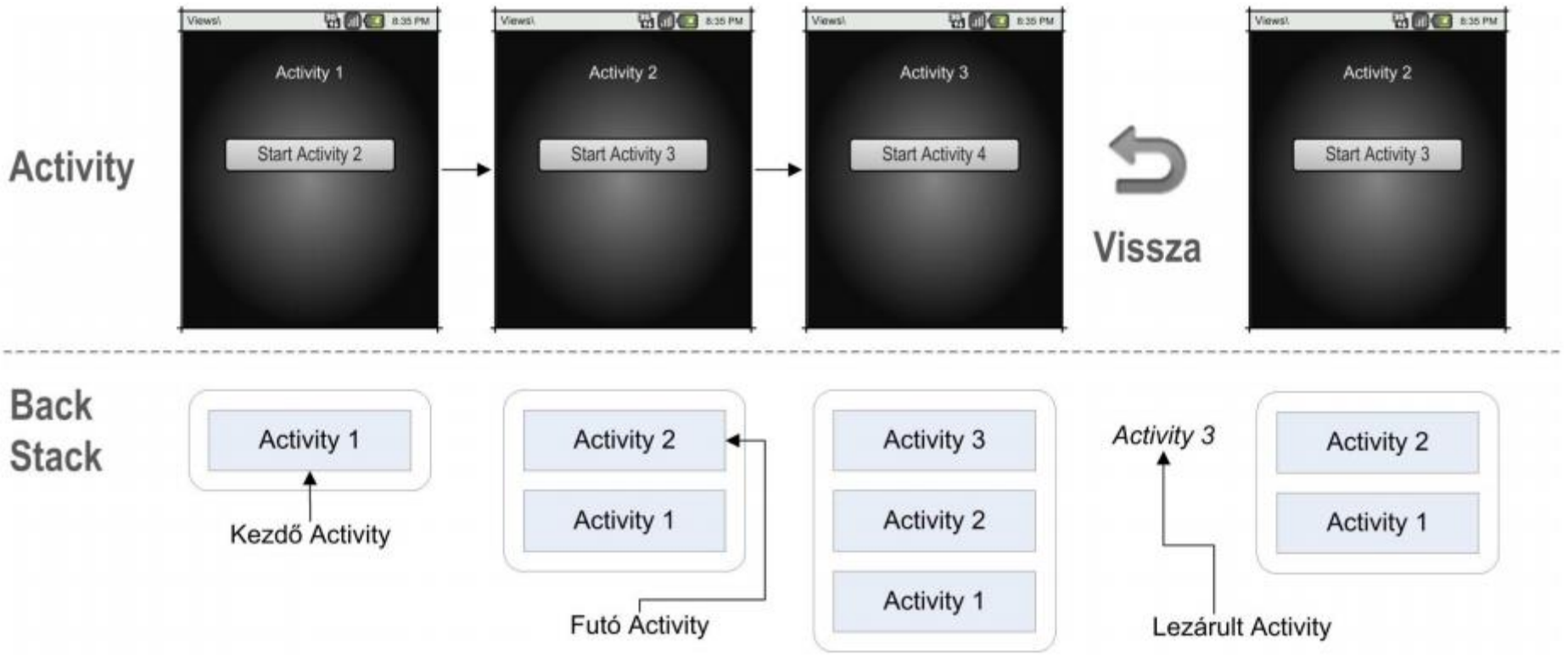
Állapotok közötti váltás

- Minden állapotváltásnál
- A megfelelő meghívódik
  - Ha nem definiáltuk felül, akkor az ősosztályé (Activity)
  - Ha felüldefiniáltuk, akkor először mindig meg kell hívni az ősosztályét!
  - super.onCreate()
- A rendszer ezeket automatikusan meghívja
- A fejlesztőnek feladata ezeket logikával feltölteni

Activity váz

```
public class MyActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //setContentView(R.layout.activity_my);

        //Létrejövünk...
    }
    @Override
    protected void onStart() {
        super.onStart();

        //Elindulunk...
    }

    @Override
    protected void onResume() {
        super.onResume();

        //Láthatóvá válunk...
    }

    @Override
    protected void onPause() {
        super.onPause();

        //Háttérbe szorulunk...
    }

    @Override
    protected void onStop() {
        super.onStop();

        //Leállunk...
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();

        //Rövid kis életünk itt véget ért.
    }
}
```

Back Stack


Activity
Activity 1
Start Activity 2
Activity 2
Start Activity 3
Activity 3
Start Activity 4
Vissza
Activity 2
Start Activity 3
Back Stack
Activity 1
Kezdő Activity
Activity 2
Activity 1
Futó Activity
Activity 3
Activity 2
Activity 1
Activity 3
Activity 2
Activity 1
Lezárult Activity

Böngésszük meg egy projektet!

Erőforrások (kód)testközelben

- Strings.xml-be:

```
<string name="welcome_message">Üdvözöllek, kedves %s!</string>
```

- Használat kódban:

```
String username = "Petőfi Sándor";
String customWelcomeMessage = getString(R.string.welcome_message, username);
```

- Ugyanígy lehet bármilyen konstanst használni
- Miért jó?
  - Layoutokból is elérhetőek (bár nem paraméterezhetőek)
  - Egy helyen tartható minden szöveg (internationalization)
  - Akár egyszerű szkriptekkel generálható is táblázatból

Logolás Androidon

- android.util.Log
  - Log.d → debug
  - Log.e → error
  - Log.i → info
  - Log.w → warning
  - Log.v → verbose

```
Log.d("TAG", "üzenet");
```

```
Logcat:
02-02 18:52:57.132: VERBOSE/ProtocolEngine(24): DownloadRate 104166 bytes per sec. Downloaded B
08-03 13:31:16.196: DEBUG/dalvikvm(2227): HeapWorker thread shutting down
08-03 13:31:16.756: INFO/dalvikvm(2234): Debugger is active
08-03 16:26:45.965: WARN/ActivityManager(564): Launch timeout has expired, giving up wake lock!
08-04 16:19:11.166: ERROR/AndroidRuntime(4687): Uncaught handler: thread main exiting due to un
08-04 16:24:11.166: ASSERT/Assertion(4687): Expected true but was false
```