

## VÝVOJ NAVIGAČNÍ APLIKACE V PROSTŘEDÍ ANDROID

Jan VANDROL

Institut geoinformatiky, Hornicko-geologická fakulta, VŠB-TUO, 17. listopadu 15, 708 33, Ostrava – Poruba,  
Česká republika  
*jan.vandrol@vsb.cz*

### Abstrakt

Článek je pojat jako úvod do tvorby mapových aplikací na mobilních zařízeních. Na počátku je uvedeno několik možností vývojových prostředí a nástrojů, nejen pro systém Android. Také jsou rozebrány některé rozdíly od tvorby programů pro stolní počítače. Dále se článek specializuje na prostředí Android, jeho možné zdroje prostorových dat a jejich dostupnost. Hluběji je pak popsána knihovna Google Maps a její možnosti. Na závěr jsou připojeny tipy a doporučení pro tvorbu mobilních aplikací.

### Abstract

This contribution is created as an introduction into a development of applications for mobile devices. It starts with a list of several possible development environments, not only for Android system. Some differences from classic desktop programming are also mentioned. Next, we take more specialized look at Android system, its possible sources of spatial data, along with their availability. The Android Google Maps library is described in more detail. The conclusion discusses mainly tips and recommendation for mobile application development.

**Klíčová slova:** Google Maps; Android; mapová aplikace

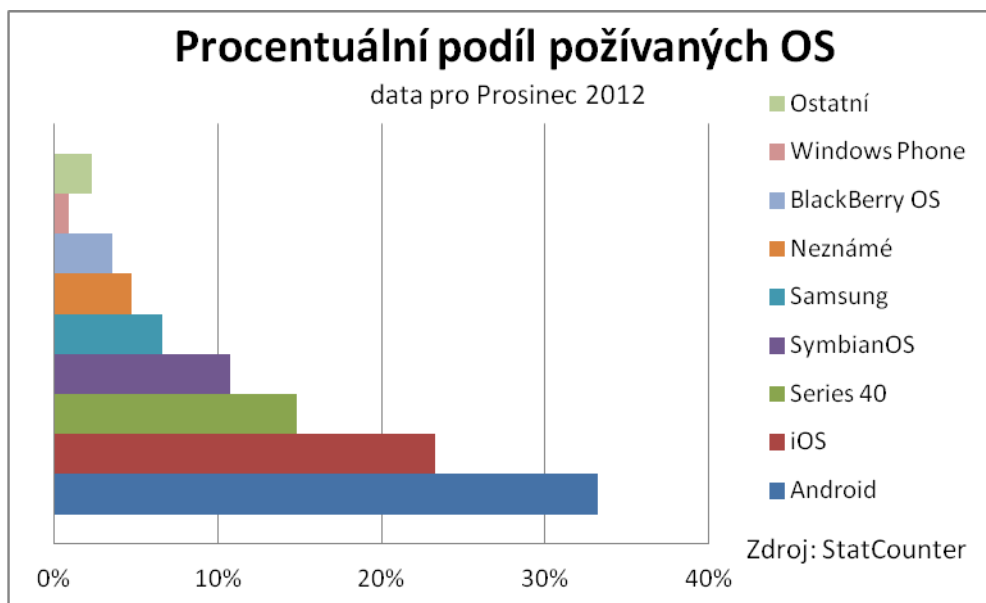
**Keywords:** Google Maps; Android; map application

### 1. ÚVOD

Mobilní zařízení se již vyvinula do stádia, kdy mohou často zastoupit desktopové systémy. Pokud tento fakt spojíme s možností využití v terénu a dnes již téměř globálním přístupem k internetu, stává se z mobilních zařízení ideální platforma pro vývoj geoinformačních aplikací. Nicméně se v této chvíli velmi málo využívá. Tento příspěvek se pokusí popsat úvod do vývoje aplikací pro mobilní zařízení se zaměřením na operační systém Android a knihovnu Google Maps.

### 2. OPERAČNÍ SYSTÉMY

V dnešní době existuje velké množství různých operačních systémů pro mobilní zařízení. Největší úspěch v této chvíli má systém Android, který postupně předbíhá svou konkurenci, i když je jedním z nejnovějších. Při tvorbě softwaru je vždycky podstatné zajistit maximální množství uživatelů čímž se Android stává velmi vhodnou platformou pro vývoj. Společně se svým konkurentem iOS od firmy Apple zabírá více než polovinu současného trhu. Díky tomu se budeme zaměřovat zejména na tyto systémy. Pokud je požadováno pokrytí maxima možných klientů, doporučuji místo aplikace vytvořit webovou službu s použitím optimalizovaných nástrojů pro mobilní zařízení, které poskytují potřebné funkce ale zároveň zabírají minimum místa. Tímto se obejde svázání programu s jedinou platformou a také neschopnost některých starších operačních systémů podporovat komplexnější mapové aplikace.



**Obr. 1** Využívané operační systémy mobilních zařízení v Prosinci 2012

Vývoj aplikací je také ovlivněn otevřeností operačního systému. V případě chybějící funkcionality OS si může schopný programátor přidat vše sám, zatímco u uzavřených platforem je odkázán pouze na funkce jemu dodané. Zde opět září open source systém Android, společně se Series 40 a několika dalšími. Naopak uzavřený iOS je obecně označován jako obtížný pro vývoj jak z hlediska funkcí, tak i programovacího jazyka.

Zejména pro začínající vývojáře je velmi podstatné množství podpory a návodů pro jednotlivé operační systémy. Dovolím si tvrdit že zde opět vede Android, ke kterému je možno najít velké množství obsáhlých a volně dostupných tutoriálů. Systém iOS má dnes již také velmi slušnou podporu nicméně se nemůže rovnat objemu informací, rad, návrhů a otevřených zdrojových kódů, jenž má Android.

### 3. GIS VÝVOJOVÉ BALÍČKY

Existuje celá řada vývojových nástrojů, které značně zrychlí tvorbu mapových aplikací. Většinou jsou vázány na určitou mapovou službu nebo technologii a jejich kód nemusí být otevřený. Samozřejmě je možné vybudovat aplikaci od začátku, nicméně pro začátečníky i pokročilé uživatele je doporučeno vyzkoušet již hotová řešení. Pro příklad si uvedeme jedny z nejčastěji užívaných API.

#### Esri Mobile GIS

Toto SDK je dostupné pro Android, iOS a Windows Phone. Umožňuje aplikaci nahrávat mapy z ArcGIS serveru, využívat jeho geoprocenční funkce a přidávat nebo upravovat data v databázích. Pokud jsou v projektu využívány technologie Esri, je tento balíček efektivní cestou k publikování dat, tvorbu aplikací pro mobilní mapování a další úlohy.

#### Bing Maps

Mapová služba firmy Microsoft poskytuje balíčky pro Android, iOS a Windows Phone. Jak lze očekávat, existuje zde vysoká provázanost a optimalizace s platformou Windows Phone, v jehož nových verzích je toto SDK nativně zahrnuto. Naopak verze pro systém Android obsahuje několik chyb které značně zpomalují rychlost běhu aplikace.

Jsou podporovány všechny základní funkce Bing Maps, jako je geokódování, hledání cest a dotazy na databázi, vykreslení vektorových symbolů a zobrazení pop-up zpráv.

### Google Maps

Jedna z nejvyužívanějších mapových služeb na internetu. Poskytuje SDK pouze pro Android a iOS.

Funkcionalita je prakticky totožná se službou Bing Maps, nicméně díky délce provozu služby již Google Maps nasbíralo mnohem větší objemy dat a je často přesnější při dotazech. Dokumentace je také obsáhlejší než v předchozím případě.

### Open Street Map

Mapový projekt, tvořen dobrovolníky z celého světa sice oficiálně neposkytuje vlastní SDK, nicméně je možné vybrat nástroje z jednoho z mnoha open source projektů, které jsou pro něj vytvořeny. Namátkou je možné například zmínit projekt *Osmdroid* pro Android platformu, jenž se snaží duplikovat funkcionalitu Google Maps balíčku. *RoutMe* je naopak projekt, soustředící se na iOS. Projekt *Leaflet* je pak knihovna pro webové aplikace, optimalizovaná pro mobilní zařízení. Existuje celá další řada dalších balíčků, které je možno použít.

Všechny jsou vytvářené komunitou, což znamená menší záruky funkčnosti, než u oficiálních SDK. Nicméně díky otevřenosti jejich kódu se stávají mnohem flexibilnější, než dříve zmíněné alternativy.

## 4. VÝVOJ V PROSTŘEDÍ ANDROID

Před popisem samotné mapové knihovny by bylo vhodné si projít aplikační komponenty systému Android a další položky se kterými se při tvorbě aplikací setkáte.

### Aplikační komponenty

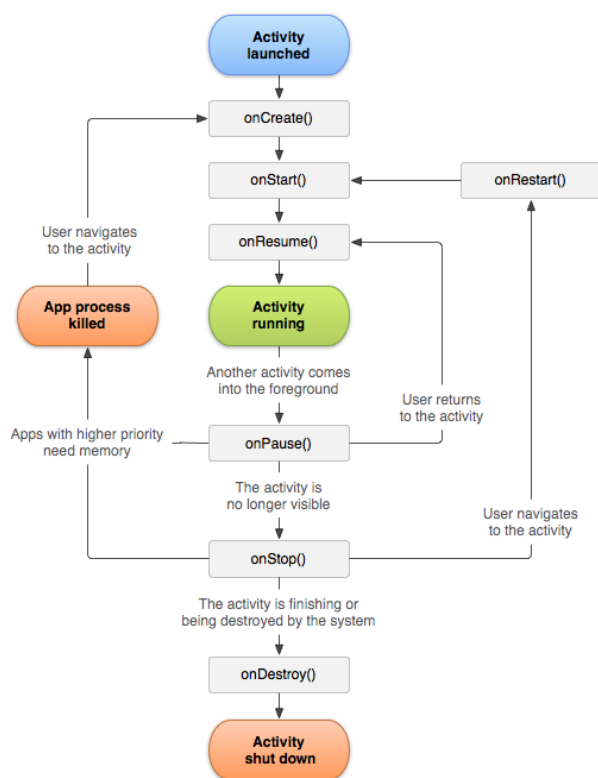
Základy jakékoli Android aplikace. Jsou rozděleny do čtyř kategorií, které článek [Android Fundamentals] popisuje takto:

- *Aktivita* reprezentuje jednu obrazovku s uživatelským rozhraním. Například mapová aplikace by mohla mít aktivitu se samotnou mapou a aktivitu pro plánování cesty. Každá má vlastní specifické UI (User Interface) a funkci. I když aktivity pracují společně pro vytvoření celistvé aplikace, každá z nich je samostatná. Mohou se navzájem zapínat podle potřeby.
- *Služba* běží na pozadí aplikace a vykonává dlouhodobé operace nebo pracuje se vzdálenými procesy. Na rozdíl od aktivity neposkytuje uživatelské rozhraní. Služba například může přehrávat hudbu, zatímco uživatel pracuje s jinou aplikací. Nebo v pozadí nahrává data z internetu bez blokování uživatele. Ostatní komponenty, jako třeba aktivita, můžou inicializovat službu, nechat ji běžet nebo ji svázat s vlastním průběhem pro docílení interaktivity.
- *Poskytovatel obsahu* spravuje sdílené části aplikačních dat. Data se mohou ukládat do souborového systému na disku, SQLite databáze, na web nebo do dalších perzistentních úložišť dat, které se na zařízení dají napojit. Pomocí něj mohou aplikace provádět dotazy nad daty a popřípadě je i modifikovat.
- *Přijímač vysílání* reaguje na systémová oznámení. Většina oznámení pochází přímo od systému. Například ohlášení vypnutí obrazovky, vybíjení baterie nebo pořízení snímků. Aplikace mohou také vysílat oznámení, většinou pro sdělení ostatním aplikacím, že byla stažena nebo vytvořena nová data, která jsou nyní dostupná k použití. Oznámení se implicitně nezobrazují uživatelům na obrazovce. Přijímač většinou jen v pozadí inicializuje různé služby podle druhu přijaté zprávy.

### Životní cyklus aktivity

V systému Android se aplikace nevypínají uživatelem. Pokaždé, když je spustěna nová aktivita, předchozí se zastaví a její poslední stav se zachovává v paměti. Pokud se k ní uživatel chce vrátit, aktivita je z paměti opět obnovena. V případě nedostatku paměti jsou postupně vymazávány nejstarší aktivity a jejich nahrané

prostředky. Při každé z těchto akcí se volá určitá funkce aktivity, jak je ukázáno na obrázku z článku [Activities].



Obr. 2 Životní cyklus aktivity

## Aplikační popis

Pomocí xml jsou definovány popisy aplikace, ať již z interního nebo vizuálního hlediska. Při tvorbě aplikací jsou podstatné dva základní návrhy:

- *Aplikační manifest* slouží k definici komponent programu, určení veškerých práv, minimální úrovně Android API a vypsání hardwaru mobilního zařízení který bude využívat. Slouží k určení kompatibility programu se zařízením, zajištění poskytnutí práv pro správný běh programu a alokaci paměti.
- *Návrh UI* je individuální pro každou aktivitu. V tomto xml se definuje pozice všech elementů uživatelského rozhraní na obrazovce.

## 5. GOOGLE MAPS KNIHOVNA

Než je vůbec možné začít využívat funkce této knihovny, je třeba si získat vývojový API klíč Google Map. Po instalaci SDK by se měl soubor *debug.keystore* implicitně nacházet v umístění *C:\Documents and Settings\<uživatel>\Local Settings\Application Data\Android*. Z něj se generuje digitální otisk pomocí *Keytool.exe* programu, který lze nalézt v lokální instalaci Javy. Příkaz pro spuštění je následující: *keytool.exe -list -alias androiddebugkey -keystore "<cesta>\debug.keystore" -storepass android -keypass android*

Tento MD5 klíč se následně registruje na stránkách Google, kde následně získáte API klíč pro aplikaci. Ten se musí zapsat k definici mapového náhledu v aplikaci, jinak se data nezobrazí.

Knihovna poskytuje speciální implementaci Android *Activity*, nazývaní se *MapActivity*. Kromě obvyklých funkcí se také stará o bufferování mapy, odstraňování starých mapových dlaždic z paměti a běh služeb na pozadí mapy.

Po založení nového Android projektu je tedy nutné místo regulérní použít *MapActivity* jakožto hlavní aktivitu programu. Do návrhu UI této aktivity se pak nadefinuje vložení mapového náhledu, což je třída *MapView*. Zde se také zapisuje již dříve získaný API klíč. Pokud je třeba zobrazit i lištu s tlačítky pro přiblížení a oddálení, je třeba se vrátit do kódu aktivity a povolit jej pomocí příkazu *setBuiltInZoomControls*.

V neposlední řadě je nutné do aplikačního manifestu přidat práva pro používání internetu a označit Google Maps jako využívanou knihovnu. Pro tyto účely se použijí tagy *uses-permission* a *uses-library*.

```
public class MainActivity extends MapActivity {

    private MapView mapView;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mapView = (MapView) findViewById(R.id.map);
        mapView.setBuiltInZoomControls(true);
    }

    @Override
    protected boolean isRouteDisplayed() {
        return false;
    }
}
```

Obr. 3 Kód Aktivity programu

```
<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" >

    <com.google.android.maps.MapView
        android:id="@+id/map"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:enabled="true"
        android:clickable="true"
        android:apiKey="0W2-8xPbFUPP9UsvDmIbcaf5Rm754P-S_eIu9Sw"
    />

</RelativeLayout>
```

Obr. 4 Návrh UI aktivity

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.test"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="15" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >

        <uses-library android:name="com.google.android.maps" />

        <activity
            android:name=".MainActivity"
            android:label="@string/title_activity_main" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

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

</manifest>
```

Obr. 5 Aplikační manifest



Obr. 6 Výsledný náhled

Úvodní verzi programu máme připravenou. Mapa je zobrazena, můžeme se v ní pohybovat pomocí gest, přiblížení a oddálení zajišťují opět gesta nebo tlačítka. Nyní je možné začít vytvářet nadstavby. Mapu

můžeme zobrazit i vektorově nebo pomocí satelitních snímků přidáním příkazů `mapView.setSatellite(true)` nebo `mapView.setStreetView(true)`.

Pro získávání polohy z GPS modulu zařízení je nutné do manifestu přidat další dvě oprávnění:

```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

V hlavní aktivitě je pak třeba implementovat třídu *LocationListener*, který při změně polohy volá funkci *onLocationChanged* kde je nadefinováno posunutí mapy na dané místo. Toho dosáhne pomocí třídy *MapController* což je ovladač mapového náhledu, který má každá instance *MapView*. Samotná registrace změn se povolí přes třídu *LocationManager*, která se stará o aktualizace polohy ať už z GPS modulu, nebo i internetu.

```
public class MainActivity extends MapActivity{

    private MapView mapView;
    private MapController mc;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mapView = (MapView) findViewById(R.id.map);
        mapView.setBuiltInZoomControls(true);
        mc = mapView.getController();

        LocationManager locMan = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
        LocationListener locLis = new MyLocationListener();
        locMan.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, locLis);
    }

    protected boolean isRouteDisplayed() {}

    private class MyLocationListener implements LocationListener
    {
        @Override
        public void onLocationChanged(Location location) {
            if (location != null) {
                GeoPoint point = new GeoPoint(
                    (int) (location.getLatitude() * 1E6),
                    (int) (location.getLongitude() * 1E6));

                mc.animateTo(point);
                mapView.invalidate();
            }
        }
    }

    public void onProviderDisabled(String provider) {}

    public void onProviderEnabled(String provider) {}

    public void onStatusChanged(String provider, int status, Bundle extras) {}
}
}
```

Obr. 7 Aktivita pro posun mapy podle GPS souřadnic zařízení

Toto jsou jen jedny z mála funkcí, které Google Maps knihovna nabízí. Její spojení s Android platformou představuje ideální prostředí pro vývoj jednoduchých i komplexních mapových řešení. Je zejména vhodná pro nové uživatele na vyzkoušení principů vývoje mapových aplikací na mobilních zařízeních. Nicméně po získání dostatečného množství zkušeností je vhodné přejít na open source řešení, které může každý vývojář maximálně přizpůsobit svým potřebám.

## 6. DOPORUČENÍ PŘI VÝVOJI

Aplikace pro mobilní zařízení musí oproti desktopovým dbát na jistá technická omezení. V této chvíli nelze počítat s velkým množstvím operační paměti. Další překážkou je i výpočetní síla, i když s novou generací

procesorů tento problém již není závažný. Je nutné také brát ohledy na spotřebu energie. Špatně navržený program může vyprázdnit baterii klidně během hodiny. Zde uvádím několik tipů na které by se mělo pamatovat při vývoji:

- Vypínat služby při pauze aplikace – Po minimalizaci programu již uživatel nemůže vidět změny své polohy v mapě. Takže je například nutné vypnout požadavky na GPS modul pro novou polohu. Toto nejen zabírá paměť zařízení, ale hlavně značně snižuje životnost baterie.
- Při pauze programu ukládat data – Pokud zařízení dojde opereční paměť, může se stát, že minimalizovanou aplikaci ukončí. Tím uživatel ztratí veškerá data se kterými pracoval. Při každé pauze je nutné uložit jak nastavení, tak i rozpracované úlohy
- Testovat aplikaci pro všechny verze Android – Všechny verze vyšší než je určená minimální musí být otestovány. Aplikace může fungovat kvalitně v Android verzi 2.3.3 ale 4.1 třeba bude chybná.
- Maximálně omezovat instance stejných tříd – Pokud to jde, používat jednu třídu na všechny funkce programu. Příkladem může být třída Color při obarvování elementů v mapě. Je zbytečné mít tři instance pro tři barvy. Stačí jen jedna instance, jejíž vlastnosti se budou měnit v průběhu programu. Uvolníte si tak místo v paměti pro podstatnější věci.
- Pravidelně uvolňovat aplikační zdroje – Nepoužívané zdroje by se měly okamžitě uvolnit. Zpomalují nejen mapovou aplikaci ale mohou způsobit zbytečné vypínání ostatních programů ve chvíli, kdy se systém snaží získat více operační paměti.

## 7. ZÁVĚR

Mobilní mapové aplikace budou hrát velkou roli v budoucím vývoji GIS. Jejich flexibilita se nedá srovnávat s žádným desktopovým řešením a výkonové rozdíly postupně přestávají hrát roli. S příchodem cloud řešení a globální internetového pokrytí se stávají ideálním způsobem jak doručit data přímo k uživateli, ať je kdekoliv.

Proto je nutné věnovat tomuto odvětví GIS velkou pozornost a dbát na vývoj korektních postupů a standardů. Knihovny jako Google Maps v této chvíli poskytují kvalitní nástroje pro práci, nicméně s dalším vývojem přichází nutnost pro kvalitní, spolehlivá a hlavně otevřená řešení, která budou sloužit jako základ budoucí generaci mapových aplikací.

## LITERATURA

Android (2012) Android Fundamentals <http://developer.android.com/guide/topics/fundamentals.html>

Android (2012) Activities <http://developer.android.com/guide/components/activities.html>

Leaflet (2012) <http://leafletjs.com/>

Osmdroid (2012) <http://code.google.com/p/osmdroid/>

Microsoft (2012) Bing Maps Developer Resources <http://www.microsoft.com/maps/developers/mobile.aspx>

Esri (2012) ArcGIS for smartphones and tablets <http://www.esri.com/software/arcgis/smartphones/develop>

Google (2012) Google Maps API <https://developers.google.com/maps/>

StatCounter (2012) <http://statcounter.com/>

Vandrol J. (2012) Standard NAVIS v prostředí systému Android. DP VŠB-TU Ostrava