

# CRÉATION D'INTERFACE UTILISATEUR

## USER INTERFACE CREATION

CRÉATION D'APPLICATIONS ET DÉCOUVERTE DES ACTIVITÉS  
CREATION OF APPLICATIONS AND DISCOVERY OF ACTIVITIES

**E.I. Djebbar**

**Département de Génie des systèmes informatiques**

Department of Computer Systems Engineering

DÉVELOPPEMENT DES APPLICATIONS MOBILES (DAM)  
MOBILE APPLICATION DEVELOPMENT

**Ecole Nationale Polytechniques d'Oran-Maurice Audin-**  
National Polytechnic School of Oran-Maurice Audin-

# LES POINTS CLÉS D'ANDROID



- Innovante car toutes les dernières technologies de téléphonie y sont intégrées : écran tactile, accéléromètre, GPS, appareil photo numérique etc.
- Accessible car en tant que développeur vous n'avez pas à acheter de matériel spécifique.
- Ouverte parce la plate-forme Android est fournie sous licence open source.

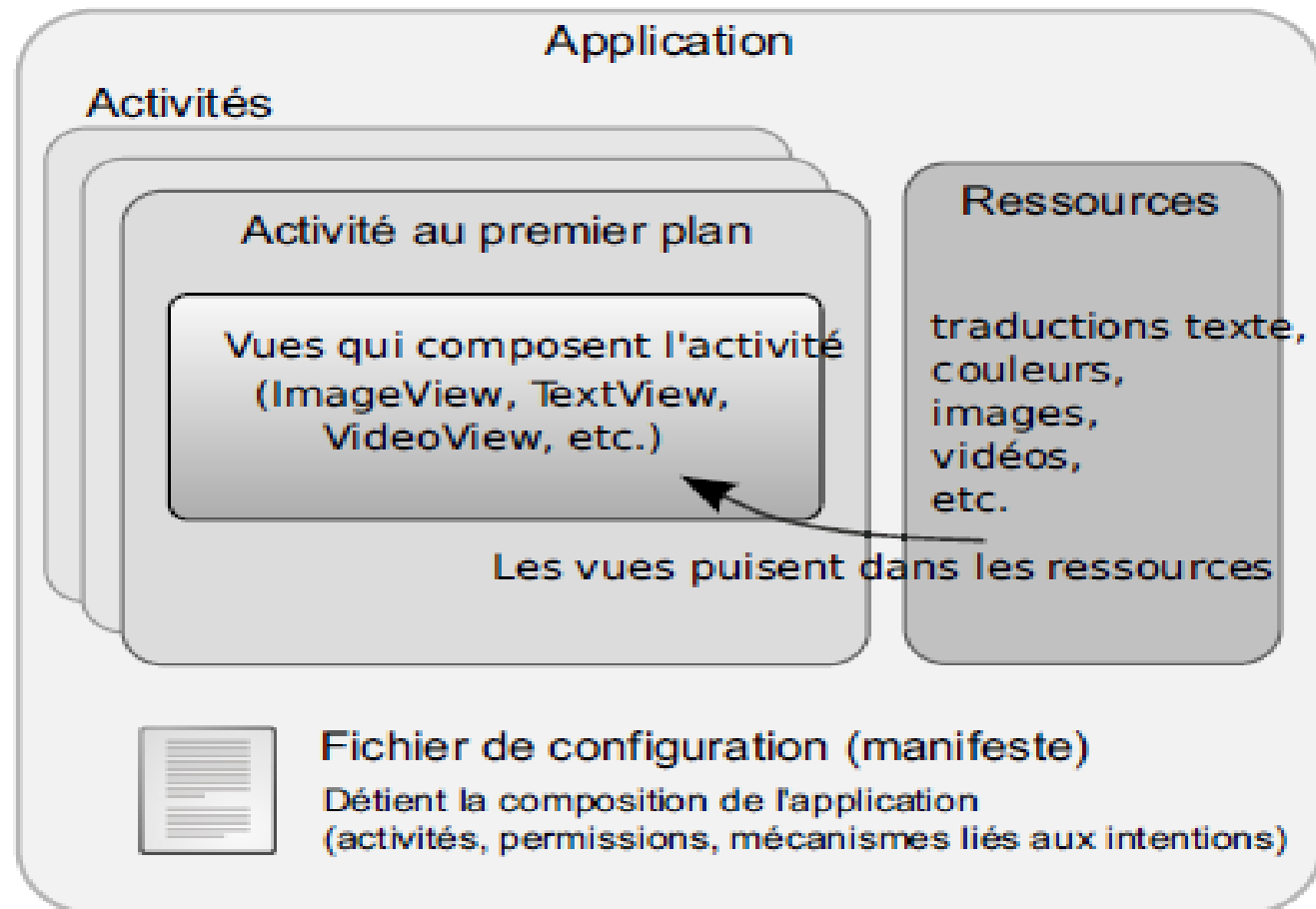
# ANDROID ET SES CONCURRENTS

- iPhone OS
- Windows Mobile
- Symbian
- BlackBerry OS
- Palm webOS, LiMo, MeeGo

# CONCEPTS FONDAMENTAUX

- Une application Android est un assemblage de composants liés grâce à un fichier de configuration.
- Concepts fondamentaux
  - Les activités ;
  - Les vues et contrôles (et leur mise en page) ;
  - Les ressources ;
  - Le fichier de configuration appelé également manifeste

# COMPOSITION D'UNE APPLICATION



# ÉLÉMENTS D'UNE APPLICATION ANDROID

- Activités ;
- Services ;
- Fournisseurs de contenu ;
- Gadgets ;
- Objets Intent ;
- Récepteurs d'Intents ;
- Notifications.

# ÉLÉMENTS D'UNE APPLICATION ANDROID

- L'activité représente le bloc de base d'une application. Elle correspond à la partie présentation de l'application et fonctionne par le biais de vues qui affichent des interfaces graphiques et répondent aux actions utilisateur.
- Le service est un composant qui fonctionne en tâche de fond, de manière invisible. Ses principales utilisations sont la mise à jour de sources de données ainsi que d'activités visibles et le déclenchement de notifications.
- Le fournisseur de contenu permet de gérer et de partager des informations. Un même fournisseur permet d'accéder à des données au sein d'une application et entre applications.

# ÉLÉMENTS D'UNE APPLICATION ANDROID

- Le gadget est un composant graphique qui s'installe sur le bureau Android. Le calendrier qui affiche de l'information ou le lecteur audio qui permet de contrôler la lecture de fichiers sont deux exemples de gadgets que l'on trouve souvent sur un écran d'accueil.

Nom	Classe ou paquetage concerné(e)
Activité	<code>android.app.Activity</code>
Service	<code>android.app.Service</code>
Fournisseurs de contenu ( <i>content provider</i> )	<code>android.content.ContentProvider</code>
Gadgets ( <i>app widget</i> )	<code>android.appwidget.*</code>



# ÉLÉMENTS D'UNE APPLICATION ANDROID

- **Intent** (Intention) : ce sont des messages asynchrones qui permettent aux composants d'application de demander des fonctionnalités à partir d'autres composants **Android**.
- **Récepteur d'Intents** : il permet à une application d'être à l'écoute des autres afin de répondre aux objets Intent qui lui sont destinés et qui sont envoyés par d'autres composants applicatifs.
- **Notification** : une notification signale une information à l'utilisateur sans interrompre ses actions en cours.

Nom	Classe concernée
Intent ( <i>Intent</i> )	<code>android.content.Intent</code>
Récepteur d'Intents ( <i>Broadcast Receiver</i> )	<code>android.content.BroadcastReceiver</code>
Notification ( <i>Notification</i> )	<code>android.app.Notification</code>

# ACTIVITÉ ANDROID

- Une activité peut être assimilée à un écran qu'une application propose à son utilisateur. Pour chaque écran de votre application, vous devrez donc créer une activité. La transition entre deux écrans correspond au lancement d'une activité ou au retour sur une activité placée en arrière-plan.
- Une activité est composée de deux volets :
  - La logique de l'activité et la gestion du cycle de vie de l'activité qui sont implémentés en Java dans une classe héritant de `Activity`
  - L'interface utilisateur, qui pourra être définie soit dans le code de l'activité soit de façon plus générale dans un fichier XML placé dans les ressources de l'application.

# SQUELETTE MINIMAL POUR CRÉER UNE PREMIÈRE ACTIVITÉ

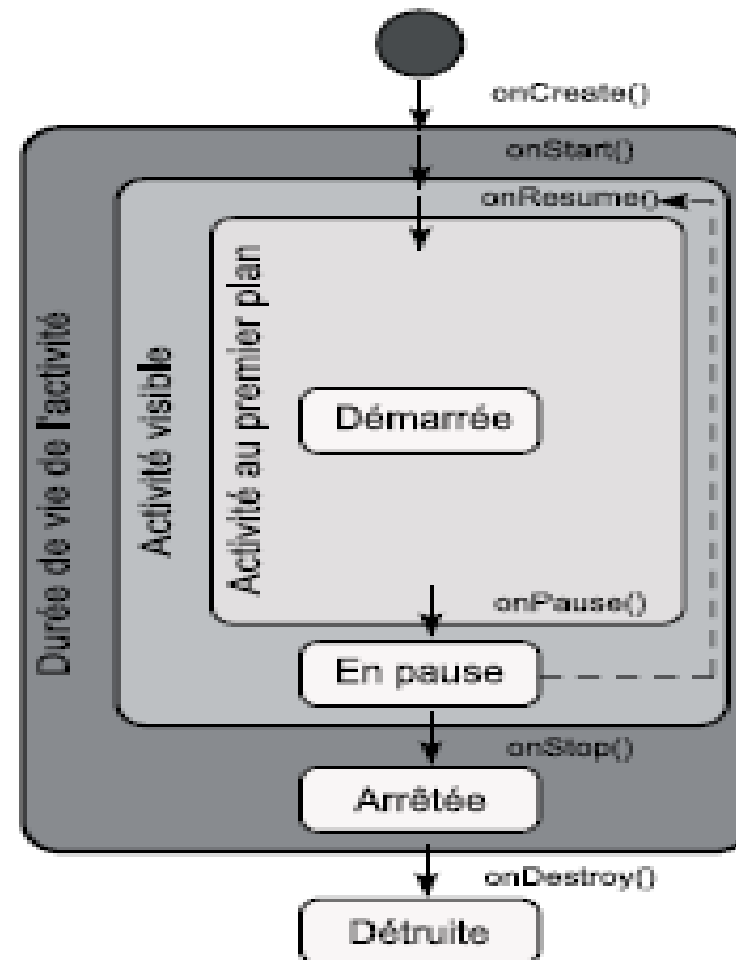
```
import android.app.Activity;
import android.os.Bundle;

public class ActiviteSimple extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}
```

# CYCLE DEVIE D'UNE ACTIVITÉ

- Active (active) : activité visible qui détient le focus utilisateur et attend les entrées utilisateur. C'est l'appel à la méthode `onResume`, à la création ou à la reprise après pause qui permet à l'activité d'être dans cet état. Elle est ensuite mise en pause quand une autre activité devient active grâce à la méthode `onPause` ;
- Suspendue (paused) : activité au moins en partie visible à l'écran mais qui ne détient pas le focus. La méthode `onPause` est invoquée pour entrer dans cet état et les méthodes `onResume` ou `onStop` permettent d'en sortir ;
- Arrêtée (stopped) : activité non visible. C'est la méthode `onStop` qui conduit à cet état.

# CYCLE DEVIE D'UNE ACTIVITÉ



# CYCLE DEVIE D'UNE ACTIVITÉ

```
import android.app.Activity;
import android.os.Bundle;

public final class TemplateActivity extends Activity {

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Override
    public void onDestroy(){
        super.onDestroy();
    }

    @Override
    public void onStart(){
        super.onStart();
    }
}
```

# CYCLE DEVIE D'UNE ACTIVITÉ

```
@Override  
public void onStop() {  
    super.onStop();  
}
```

```
@Override  
public void onRestart() {  
    super.onRestart();  
}
```

```
@Override  
public void onPause() {  
    super.onPause();  
}
```

```
@Override  
public void onResume() {  
    super.onResume();  
}
```

# UTILISATION DES RESSOURCES

## Exemple de classe R générée automatiquement

```
public final class R {  
    public static final class string {  
        public static final int bienvenue=0x7f040000;  
        public static final int texte_bouton_quitter=0x7f040001;  
        public static final int texte_titre_ecran=0x7f040002;  
    };  
    public static final class layout {  
        public static final int ecran_de_demarrage=0x7f030001;  
        public static final int ecran_principal=0x7f030000;  
    };  
    public static final class drawable {  
        public static final int image_android=0x7f020000;  
    };  
};
```



# ACTIVITÉ PRINCIPALE CRÉÉE PAR DÉFAUT

```
import android.app.Activity;
import android.os.Bundle;
public class ActivitePrincipale extends Activity{
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

# APERÇU DU FICHER MAIN.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/
android"
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
>
<TextView
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="@string/hello"
/>
</LinearLayout>
```

# CRÉATION D'UNE ACTIVITÉ

```
import android.app.Activity;
import android.os.Bundle;
import android.widget.LinearLayout;
import android.widget.TextView;
public class MainActivity extends
Activity {
    @Override
    public void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
        LinearLayout layout = new
        LinearLayout(this);
        TextView text = new TextView(this);
        text.setText(R.string.hello);
        layout.addView(text);
        setContentView(layout);
    }
}
```

## DÉFINITION D'INTERFACE (EXEMPLE 2)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+id/monText"
    />
</LinearLayout>
```

## EXAMPLE 2

```
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
public class Main extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        TextView monTexte = (TextView)findViewById(R.id.monText);
        monTexte.setText("Bonjour tout le monde !");
    }
}
```

# CRÉATION D'INTERFACE SANS DÉFINITION XML

```
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
public class Main extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView monTextView = new TextView(this);
        setContentView(monTextView);
        monTextView.setText("Bonjour tout le monde !");
    }
}
```

# CRÉATION D'UN GROUPE DEVUES DE FAÇON PROGRAMMATIQUE

```
import android.app.Activity;

import android.os.Bundle;

import android.widget.LinearLayout;
import android.widget.TextView;

public class Main extends Activity {

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        LinearLayout monLinearLayout = new LinearLayout(this);
        monLinearLayout.setOrientation(LinearLayout.VERTICAL);

        TextView monTextView1 = new TextView(this);
        TextView monTextView2 = new TextView(this);
        monLinearLayout.addView(monTextView1);
        monLinearLayout.addView(monTextView2);

        setContentView(monLinearLayout);

        monTextView1.setText("Bonjour tout le monde !");
        monTextView2.setText("Ceci est mon 2eme texte");

    }

}
```

# INSERTION D'UN BOUTON DANS L'INTERFACE

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_height="fill_parent"
android:layout_width="fill_parent"
android:gravity="center_vertical|center_horizontal"
>
<Button
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/monBouton"
android:text="Cliquez ici !"
>
</Button>
</LinearLayout>
```



# CRÉATION D'UN ÉCOUTEUR SUR UN BOUTON

```
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;

public class Main extends Activity {

    Button bt;

    @Override

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        bt= (Button) findViewById(R.id.monBouton)
        bt.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(Main.this, "Bouton cliqué !", Toast.LENGTH_LONG).show();
            }
        });
    }
}
```