

Notions essentielles d'une application Android

Séquence 3 : Initiation au développement mobile

Université Virtuelle du Sénégal (UVS)
Licence 1 Informatique
2018

Plan du Cours

1. La plate-forme Android

- Les points clés d'Android en tant que plate-forme
- Les versions de la plate-forme
- Une architecture autour du noyau Linux

2. Les composants d'une application Android

- Les composants applicatifs
- Éléments d'interaction : intents, récepteurs, notifications

3. Les interfaces utilisateur

- Le concept d'interface
- Gérer les événements
- Intégrer des éléments graphiques dans votre interface

La Plate-forme Android Points Clés

- **Innovante** : Intègre les dernières technologies (écran tactile, accéléromètre, GPS, appareil photo).
- **Accessible** : Développement en **Java**, pas besoin de matériel spécifique (hormis un téléphone Android pour les tests physiques).
- **Ouverte** : Plate-forme **open source**, permettant la consultation et la modification des sources par les développeurs et constructeurs.

Versions de la Plate-forme

- Les versions Android évoluent rapidement, apportant de nouvelles fonctionnalités, mais pouvant affecter la compatibilité des applications.

| Code Name | Plate-forme | Niveau API |
|--------------------|-------------|------------|
| Cupcake | Android 1.5 | 3 |
| Donut | Android 1.6 | 4 |
| Eclair | Android 2.1 | 7 |
| Froyo | Android 2.2 | 8 |
| Gingerbread | Android 2.3 | 9 |
| Honeycomb | Android 3.0 | 11 |
| Ice Cream Sandwich | Android 4.0 | 14 |
| Jelly Bean | Android 4.1 | 16 |
| KitKat | Android 4.4 | 19 |
| Lollipop | Android 5.0 | 20 |
| Marshmallow | Android 6.0 | 23 |
| Nougat | Android 7.0 | 24 |

TABLE 1 – Versions de la plate-forme Android.

— Tableau des versions :

Architecture autour du Noyau Linux

- **Linux Kernel** (version 3.4 pour Lollipop 5.1) : Fournit les services fondamentaux (pilotes matériels, gestion des processus, mémoire, sécurité, réseau, gestion de l'énergie).
- **Android Runtime (ART)** : Remplace **Dalvik** depuis Lollipop, exécute les applications et sert de base au framework applicatif.
- **Libraries** : Incluent **Surface Manager**, **OpenGL ES**, **SQL**, **SSL**, **Libc**.

Les Composants d'une Application Android Composants

Applicatifs

- **Activité** (`android.app.Activity`) : Interface graphique de base, répondant aux actions utilisateur.
- **Service** (`android.app.Service`) : Tâche de fond invisible pour mises à jour ou notifications.
- **Fournisseur de contenu** (`android.content.ContentProvider`) : Gère et partage des données au sein ou entre applications.
- **Gadget** (`android.appwidget`) : Composant graphique sur l'écran d'accueil (ex. : calendrier, lecteur audio).

Éléments d'Interaction

- **Intent** (`android.content.Intent`) : Diffuse des messages pour demander des actions ou services.
- **Récepteur d'Intents** (`android.content.BroadcastReceiver`) : Écoute les intents envoyés par d'autres composants.

| tableheader Composant | Classe ou Paquetage |
|---------------------------------|---------------------------------|
| tablerow Activité | android.app.Activity |
| Service | android.app.Service |
| tablerow Fournisseur de contenu | android.content.ContentProvider |
| Gadget | android.appwidget |

TABLE 2 – Composants applicatifs Android.

| tableheader Élément | Classe |
|-----------------------|-----------------------------------|
| tablerow Intent | android.content.Intent |
| Récepteur d'Intents | android.content.BroadcastReceiver |
| tablerow Notification | android.app.Notification |

TABLE 3 – Éléments d'interaction Android.

- **Notification** (android.app.Notification): Signale des informations sans interrompre l'utilisateur.

Cycle de Vie d'une Activité

- Pas de méthode **main**; Android utilise des **callbacks** pour gérer les états d'une activité.
- Principales méthodes :

| tableheader Méthode | Signification |
|----------------------|--|
| tablerow onCreate() | Appelée lors de la création initiale de l'activité. |
| onStart() | Appelée après la création ou après onStop(); l'activité devient visible. |
| tablerow onResume() | Appelée lorsque l'activité est visible et interactive. |
| onPause() | Appelée lorsque l'activité n'est plus au premier plan. |
| tablerow onStop() | Appelée lorsque l'activité n'est plus visible. |
| onRestart() | Appelée lorsqu'une activité arrêtée est reprise. |
| tablerow onDestroy() | Appelée lorsque l'activité est terminée ou en arrière-plan. |

TABLE 4 – Cycle de vie d'une activité.

Les Interfaces Utilisateur Concept d'Interface

- Une interface est un assemblage hiérarchique de composants graphiques (**vues**) avec des attributs (taille, couleur, position).

- Organisée en arbre via **ViewGroup**, permettant des structures complexes.

Gabarits (Layouts)

- **LinearLayout** : Aligne les éléments horizontalement ou verticalement.
- **RelativeLayout** : Positionne les éléments par rapport aux autres.
- **FrameLayout** : Superpose les éléments, utilisé pour un seul composant.
- **TableLayout** : Organise les vues en lignes et colonnes.

Listing 1 – Exemple de LinearLayout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/
  android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical">
</LinearLayout>
```

Gérer les Événements

- Les actions utilisateur (clics, effleurements) sont des **événements**.
- Utilisation d'**écouteurs** (listeners) pour associer des actions à des événements.

Listing 2 – Ajout d'un bouton

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

Listing 3 – Écouteur de clic

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

public class Main extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

```

        ((Button) findViewById(R.id.monBouton)).setOnClickListener(
            new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    Toast.makeText(Main.this, "Bouton_cliqué!", Toast.
                        LENGTH_SHORT).show();
                }
            });
    }
}

```

Intégrer des Éléments Graphiques

- **ImageView** : Affiche des images via l'attribut src ou setImageResource.
- **EditText** : Permet la saisie de texte.
- Autres composants : **CheckBox**, **ImageButton**, **RadioGroup**, **RadioButton**, **RatingBar**, **DigitalClock**, **AnalogClock**.