Chapitre 3: Création des interfaces utilisateur

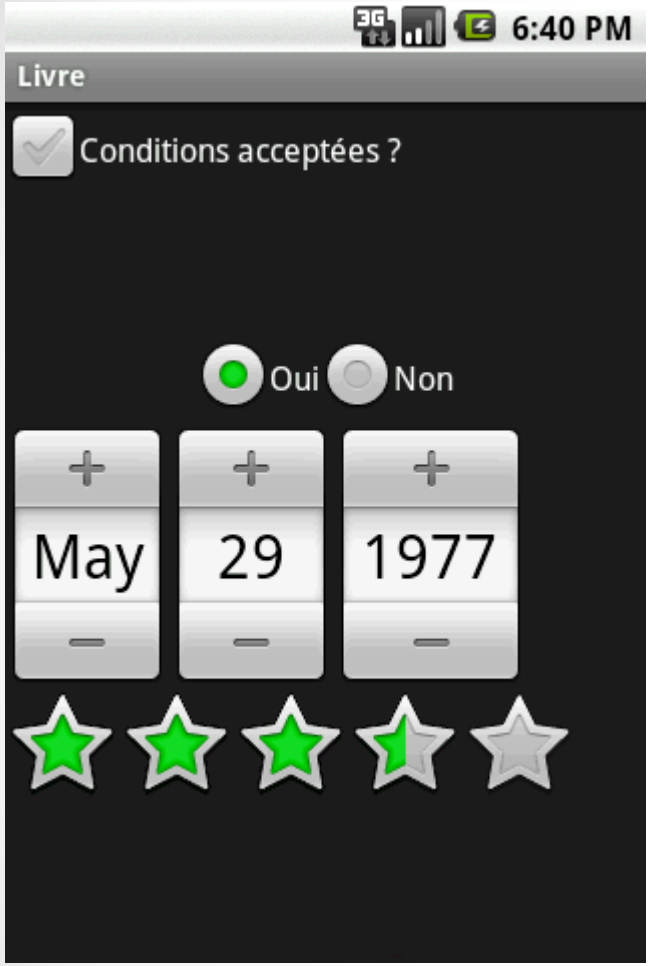
Développement mobile sous Android

Le concept d'interface

- Une interface n'est pas une image statique…
- C'est Un ensemble de composants graphiques dont leurs attributs peuvent être commun.
- Sous Android: 2 façons de déclarer une interface:

1. Avec une description XML (res/layout)
2. Avec le code Java (dans l'activity)

Le concept d'interface



Vue et layout

- Les **vue**: les composantes graphiques (boutton, image,cases à cocher …)
- Les vues heritent de la classe **View**
- Android offre la possibilité de regrouper ces vues dans une structure arborescente (class ViewGroup)
- Un **ViewGroup** peut contenir d'autre ViewGroup et View
- **Layout** (ou gabarit ou mise en page) une extension de la classe ViewGroup
- Layout est un conteneur qui aide à positionner des objets (gabarit ou vue)
- Layout parent peut contenir plusieurs layout enfants

Les types de layout

- **LinearLayout** : permet d'aligner de gauche à droite ou de haut en bas (suivant l'orientation).
- **RelativeLayout** : ses enfants sont positionnés les uns par rapport aux autres, le premier enfant servant de référence aux autres.
- **FrameLayout** : c'est le plus basique des gabarits. Chaque enfant est positionné dans le coin en haut à gauche de l'écran et affiché par-dessus les enfants précédents, les cachant en partie ou complètement
- **TableLayout** : permet de positionner vos vues en lignes et colonnes à l'instar d'un tableau.

Exemple de layout

```
<!-- Mon premier gabarit -->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="fill_parent">

</LinearLayout>
```

- layout_width: largeur du layout
- layout_height: hauteur du layout
- La valeur Fill_parent: le layout prend tout l'espace (largeur/hauteur) du layout parent
- La valeur Wrap_content: le layout l'espace qu'il lui faut

Les unités de mesure

- **Pixel (px):** correspond a un pixel à l'écran
- **Pouce (in):** basé sur la taille physique de l'écran (2,54 cm)
- **Millimètre (mm):** basé sur la taille physique de l'écran
- **Point (pt):** 1/72 d'un pouce
- **Pixel à densité indépendante (dp ou dip)** une unité relative se basant sur une taille physique de l'écran de 160 dpi (2 dp sur un écran de 160dpi= 2px et 3px sur un écran 240 dpi (2*240/160))
- **Pixel à taille indépendant (sp):** équivalent au dp et fonction de la taille de police spécifié par l'utilisateur. C'est la plus recommandé…

Création d'interface utilisateur

La création d'une interface se traduit par la création de deux éléments :

- une définition de l'interface utilisateur (gabarits, etc.) de façon déclarative dans un fichier XML
- une définition de la logique utilisateur (comportement de l'interface) dans une classe d'activité.

⇒ Une séparation stricte entre la présentation et la logique fonctionnelle de votre application

Création d'interface utilisateur avec XML

```
<?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" >

    <TextView
            android:id="@+id/monText"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Bonjour  tout le monde…."
            />
</LinearLayout>
```

Associer interface au code java

```
import android.app.Activity;
import android.os.Bundle;
public class MainActivity extends Activity {
        @Override
        public void onCreate(Bundle
        savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
//R.layout.activity_main est l'identifiant du fichier
activity_main.xml
```

Récupération d'un view

```
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
public class MainActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    TextView  monTx =
    (TextView)findViewById(R.id.monText);
    monTx.setText("Bonjour les amis !");
    }
}
```

Créer interface **sans XML**

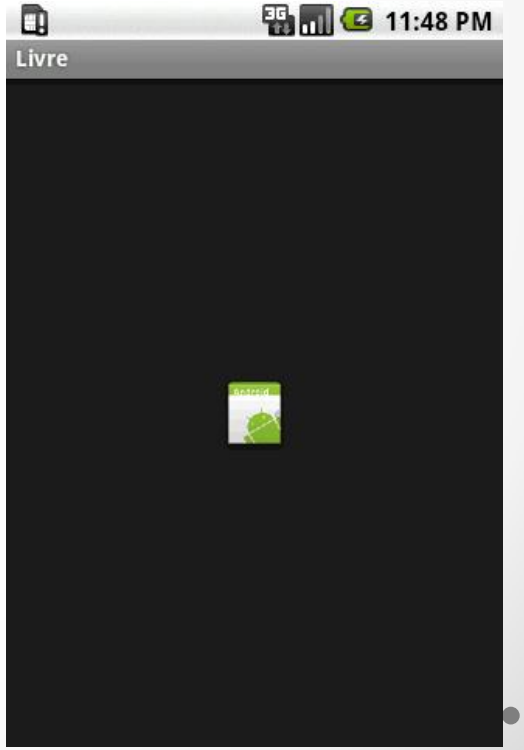
```
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
public class MainAcitivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    TextView monTextView = new TextView(this);
    monTextView.setText("Bonjour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

    }
}
```

Intégrer des vues dans layout

```
public class MainActivity extends Activity {
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // Nous instancions un LinearLayout dans lequel nous intégrerons nos différents TextView
    LinearLayout monLinearLayout = new LinearLayout(this);
    // Nous paramétrons monLinearLayout afin qu'il affiche les vues les unes au-dessus des autres
    monLinearLayout.setOrientation(LinearLayout.VERTICAL);
    // Nous instancions nos deux TextViews à afficher
    TextView monTextView1 = new TextView(this);
    TextView monTextView2 = new TextView(this);
    // Nous ajoutons les deux TextViews dans notre monLinearLayout
    monLinearLayout.addView(monTextView1);
    monLinearLayout.addView(monTextView2);
    // Nous appliquons monLinearLayout sur notre activité
    setContentView(monLinearLayout);
    // Nous paramétrons un texte à afficher sur nos 2 TextViews
    monTextView1.setText("Bonjour tout le monde !");
    monTextView2.setText("Ceci est mon 2eme texte");
    }
}
```

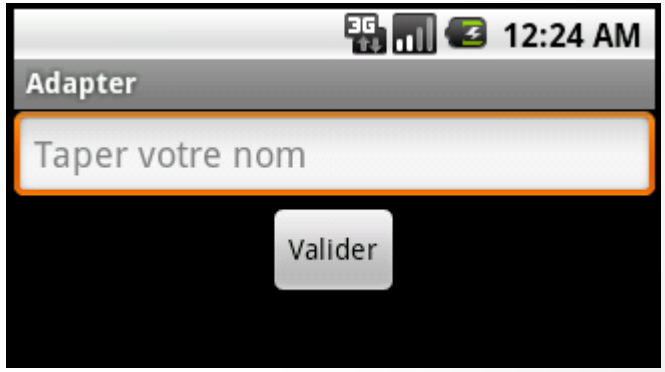
Intégrer une image dans une interface

```
<?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"
>
    <ImageView
    android:id="@+id/monImage"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/icon"
    >
   </ImageView>
</LinearLayout>
```



Intégrer une boite de saisie

```
<?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">
    <EditText
    android:id="@+id/monEditText"
    android:layout_height="wrap_content"
    android:hint="Taper votre nom"
    android:layout_width="fill_parent">
    </EditText>
    <Button
    android:id="@+id/monBouton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Valider"
    android:layout_gravity="center_horizontal">
    </Button>
</LinearLayout>
```



Integrer un checkBox , un RadioButton, imageButton

```
<!-- Case a cocher -->
<CheckBox
android:id="@+id/CheckBox01"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Conditions acceptées ?">
</CheckBox>

<!-- Bouton avec une Image -->
<ImageButton
android:id="@+id/ImageButton01"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:src="@drawable/icon">
<!-- @drawable/icon est une Image qui se trouve dans le dossier /res/drawable
de notre projet -->
</ImageButton>
```

Radio Bouton et Radio Groupe

```
<!-- Groupe de boutons radio -->
<RadioGroup
 android:id="@+id/RadioGroup01"
 android:layout_width="wrap_content"

android:layout_height="wrap_content"
 android:orientation="horizontal"

android:layout_gravity="center_horizo
ntal">
 <!-- Radio Bouton 1 -->
 <RadioButton
  android:id="@+id/RadioButton01"
  android:layout_width="wrap_content"

android:layout_height="wrap_content"
 android:text="Oui"
  android:checked="true">
```

```
<!-- Nous avons mis checked=true par
défaut sur notre 1er bouton radio afin
qu'il soit coché -->

</RadioButton>
 <!-- Bouton radio 2 -->
 <RadioButton
 android:id="@+id/RadioButton02"
 android:layout_width="wrap_content"
android:layout_height="wrap_content"
 android:text="Non">
 </RadioButton>
</RadioGroup>
```

Horloge

```
<DigitalClock
android:text="Horloge"
android:id="@+id/DigitalClock01"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center_vertical">
</DigitalClock>
<!-- Horloge Analogique -->
<AnalogClock
android:id="@+id/AnalogClock01"
android:layout_width="wrap_content"
android:layout_height="wrap_content">
</AnalogClock>
```

Date

```
<!-- Sélectionneur de date -->
<DatePicker
android:id="@+id/DatePicker01"
android:layout_width="wrap_content"
android:layout_height="wrap_content">
</DatePicker>
<!-- Barre de vote -->
<RatingBar
android:id="@+id/RatingBar01"
android:layout_width="wrap_content"
android:layout_height="wrap_content">
</RatingBar>
```

Ecouteur sur un bouton (1)

```
((Button)findViewById(R.id.monBouton)).setOnClickListener(
    new OnClickListener() {
        @Override
        public void onClick(View v) {
        // On récupère notre EditText
        EditText  texte = ((EditText)findViewById(R.id.monEditText));
        // On garde la chaîne de caractères
        String nom = texte.getText().toString();
        // On affiche ce qui a été tapé
        Toast.makeText(Main.this, nom,
        Toast.LENGTH_SHORT).show();
        }
});
```

Ecouteur sur Boutton (2)

```
public class Principale extends Activity implements View.OnClickListener {
/** Called when the activity is first created. */
    Button  monbouton;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.principale_layout);
        monBouton = (Button) findViewById(R.id.monBouton);
        monbouton.setOnClickListener(this);
    }
    @Override
    public void onClick(View view){
    Toast.makeText(Principale.this, "Bravo", Toast.LENGTH_SHORT).show();
    }
}
```

Ecouteur sur un CheckBox

```
// On récupère notre case à cocher pour intercepter
//l'événement d'état (cochée ou pas)

((CheckBox)findViewById(R.id.CheckBox01))
                              .setOnCheckedChangeListener(
        new CheckBox.OnCheckedChangeListener() {
        public void onCheckedChanged(CompoundButton
                                    buttonView, boolean isChecked)
        {
                afficheToast("Case cochée ? : " + ((isChecked)?
                                                    "Oui" :"Non"));
        }
 });
```

Ecouteur sur un DatePicker

```
// On récupère notre sélectionneur de date (DatePicker) pour attraper l'événement
//du changement de date
// Attention, le numéro de mois commence a 0 dans Android, mais //pas les jours.
//Donc si vous voulez mettre le mois de Mai, vous //devrez fournir 4 et non 5

((DatePicker)findViewById(R.id.DatePicker01)).init(1977, 4, 29,
  new DatePicker.OnDateChangedListener() {
@Override
        public void onDateChanged(DatePicker view, int year,
                        int monthOfYear, int dayOfMonth)
        {
                afficheToast("La date a changé\nAnnée : "
                                + year + " | Mois : " + monthOfYear
                                + " | Jour : " + dayOfMonth);
        }
});
```

Ecouteur sur un RadioButton

```
// On récupère notre groupe de bouton radio pour
//attraper le choix de l'utilisateur
((RadioGroup)findViewById(R.id.RadioGroup01)).
        setOnCheckedChangeListener(
                new RadioGroup.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(RadioGroup group,
                                                int checkedId)
        {

                afficheToast("Vous avez répondu : "
         + ((RadioButton)findViewById(checkedId)).getText());
        }
});
```

Ecouteur sur RatingBar

```
// On récupère notre barre de vote pour attraper la
// nouvelle note que sélectionnera l'utilisateur
((RatingBar)findViewById(R.id.RatingBar01))
        .setOnRatingBarChangeListener(
         new RatingBar.OnRatingBarChangeListener() {
        @Override
        public void onRatingChanged(RatingBar ratingBar,
                                    float rating, boolean fromUser)
        {
// On affiche la nouvelle note sélectionnée par l'utilisateur
                afficheToast("Nouvelle note : " + rating);
        }
});
```

Ecouteur sur une image

```
// On récupère notre Bouton Image pour attraper le clic
//effectué par l'utilisateur
((ImageButton)findViewById(R.id.ImageButton01))
                    .setOnClickListener( new OnClickListener() {
        @Override
        public void onClick(View v) {
        // On affiche un message pour signalé que le bouton
        //image a été pressé
        afficheToast("Bouton Image pressé");
        }
});
}
```