



UNIVERSITÉ MOHAMMED V
FACULTÉ DES SCIENCES
RABAT

DÉVELOPPEMENT POUR MOBILES (ANDROID)



Chapitre 3 : La gestion des événements

Dr. KHALIFI Hamid
h.khalifi@um5r.ac.ma

Les événements

Implantation du comportement

- Les fichiers XML ne permettent que de :
 - positionner les composants ;
 - définir leurs caractéristiques.
- Nécessité de :
 - définir leur comportement
 - type d'interaction (clic court, clic long, etc.)
 - code de prise en compte (Java)
 - lier composant et code (XML ou Java)
 - XML : attribut android:onClick
 - Java : instancier un event listener

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements

Gestion de clic

- Il existe plusieurs façons d'interagir avec une interface graphique. Par exemple cliquer sur un bouton ou une touche du clavier, etc ... **Ces interactions s'appellent des événements.**
- **Méthode 1:** Créer un auditeur d'événements et l'enregistrer auprès du composant (View).
- **Méthode 2 :** les vues (Views) sont elles mêmes auditrices de certains événements : (cliquer sur un bouton par exemple). Il suffit donc de spécialiser la méthode adaptée et lancée lorsque l'événement survient.

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements

Méthode 1 : Solution Java

Création d'un auditeur d'événements

- Pour pouvoir réagir à l'apparition d'un événements, il faut utiliser un **objet qui va détecter l'événement** afin de vous permettre le traiter.
 - Ce type d'objet s'appelle un **listener**.
- Un **listener** est une interface qui vous oblige à redéfinir des méthodes qui vont être appelée un moment où se produira l'événement associé.

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements

Méthode 1 : Solution Java

Création d'un auditeur d'événements

L'implémentation d'un **listener** à une classe :

- Ca veut dire que la classe sera à **l'écoute des actions** qu'on effectuera sur les vues.
- A chaque fois qu'on effectuera une action, une méthode qui correspond à cette action sera appelée, et on définira dans cette méthode ce qu'il faut faire une fois l'action effectuée.

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements

Méthode 1 : Solution Java

Création d'un auditeur d'événements

- Il existe plusieurs interfaces, une pour chaque type d'actions.

Exemple de gestion d'un **clic normal** :

- L'interface utilisée s'appelle **View.OnClickListener** et nous exige à définir la méthode **void onClick (View v)**.
 - C'est dans cette méthode qu'on définira la conduite à adopter en cas de clic. L'argument de type **View** est la vue sur laquelle le clic a été effectué.

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Méthode 1 : Solution Java

Création d'un auditeur d'événements

- Avec cette méthode, l'événement est réalisé dans la méthode **onCreate()** de l'activité suivant les étapes ci-dessous :

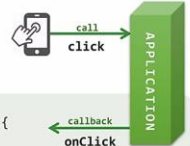
- On récupère le composant depuis le layout XML avec : **findViewById(R.layout.id_composant)**.
- On appelle le setter pour l'installation du listener.
- On insère le corps d'événement à l'intérieur du listener.

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Méthode 1 : Solution Java

Création d'un auditeur d'événements

Abonner l'activité à des événements spécifiques



```
/java/MainActivity.java
@Override
public void onCreate(Bundle savedInstanceState) {
    ...
    Button b = (Button) findViewById(R.id.btn);
    b.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v){
            ... // bouton cliqué!!!
        }
    });
}
```

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Méthode 1 : Solution Java

Création d'un auditeur d'événements

Exemple :

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        final EditText txt1;
        final EditText txt2;
        final EditText txt3;
        Button btn;
        Button btnReset;

        txt1 = (EditText) findViewById(R.id.edit1);
        txt2 = (EditText) findViewById(R.id.edit2);
        txt3 = (EditText) findViewById(R.id.edit3);
        btn = (Button) findViewById(R.id.buttonSum);
        btnReset = (Button) findViewById(R.id.buttonReset);

        btnReset.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                txt1.setText("");
                txt2.setText("");
                txt3.setText("");
            }
        });
    }
}
```



DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Méthode 2 : Solution XML

Vue auditrice d'événement

- Cette méthode permet d'écrire directement la gestion d'événements qui peuvent se produire dans la vue (View).
- L'attribut **android:onClick** d'un composant graphique indique le nom de la méthode qui sera lancée si on clique sur cette vue (View) c. à. d. Il doit être suivi du nom de la méthode à appeler en cas de déclenchement
- Prototype :
 - `public void nomDeLaMethode(View Vue)`

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Méthode 2 : Solution XML

Vue auditrice d'événement

```
<Button
    android:id="@+id/button_send"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="sendMessage"
    android:text="@string/button_send" />
```

- Ajout de l'attribut **android:onClick** à l'élément Button
- Valeur : méthode définie dans l'activité contenant la vue (View)
- Signature standardisée : **public void** et un paramètre de type **View**

```
/** Called when the user touches the button */
public void sendMessage(View view) {
    // Do something in response to button click
}
```

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Méthode 2 : Solution XML

Vue auditrice d'événement

Exemple :

```
/res/layout/activity_main.xml
<Button
    android:onClick="func"
    android:id="@+id/btn" />
...
```

```
/java/MainActivity.java
@Override
public void onCreate(Bundle savedInstanceState) {
    ...
    public void func(View v) {
        ... // bouton cliqué!!!
    }
}
```

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Méthode 2 : Solution XML

Vue auditrice d'événement

Exemple:

```
<Button
    android:id="@+id/buttonReset"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginRight="20dp"
    android:layout_weight="1"
    android:onClick="buResetClick"
    android:text="Reset"/>
```

```
public void buResetClick(View view) {
    EditText txt1 = (EditText) findViewById(R.id.editText1);
    EditText txt2 = (EditText) findViewById(R.id.editText2);
    EditText txt3 = (EditText) findViewById(R.id.editText3);
    txt1.setText("");
    txt2.setText("");
    txt3.setText("");
}
```

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Méthode 2 : Solution XML

Plusieurs vues pour le même événement

- Il est possible de récupérer des informations sur le composant graphique qui a généré l'événement.

```
<Button
    android:id="@+id/monBouton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/monTexte"
    android:onClick="onBoutonClique"
/>
```

```
public void onBoutonClique(View maVue) {
    System.out.println("le bouton a été cliqué");
}
```

Récupération :

`maVue.getId()`

→ `R.id.monBouton`

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Événement Toast

La notification rapide (Toast)

- Apparaître un message de notification, tout simple :

Visualisation :



Code :

```
Toast.makeText(getApplicationContext(), "msg msg", Toast.LENGTH_SHORT).show();
Toast.makeText(getApplicationContext(), "msg msg", Toast.LENGTH_LONG).show();
```

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Événement Toast

- La classe **Toast** permet de créer un texte qui apparaît en premier plan puis disparaît au bout d'un temps donnée. A considérer comme un message d'information ou d'avertissement.
- La classe Toast permet de créer un message avec la méthode **makeText()**, cette dernière prend 3 paramètres :
 - Le contexte de l'application.
 - Le message à afficher.
 - La durée d'affichage (Peut être : `Toast.LENGTH_SHORT` ou `Toast.LENGTH_LONG`).

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Événement Toast

Exemple:

```
Toast.makeText(Context, "Hello!", Toast.LENGTH_SHORT).show();
```

- La méthode **makeText** construit une instance de **Toast** avec ces paramètres que vous n'aurez plus qu'à afficher via la méthode **show()**.

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Événement Toast

Remarque 1 :

- Il est possible d'utiliser **makeText** avec un identifiant sur le texte à afficher en passant ainsi par une ressource texte :

```
Toast.makeText (Context context, int resId, duration)
```

Avec:

```
resId = R.string.Nom_resource_string
```

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Événement Toast

Remarque 2 :

- Il est possible de déclarer et afficher le message Toast avec deux instructions différentes :

Exemple

```
Toast t = Toast.makeText(v.getContext(), "msg", Toast.LENGTH_SHORT);
t.show();
```

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Événement Toast

Remarque 3 :

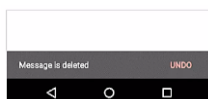
- Il est possible de personnaliser un Toast grâce à la bibliothèque externe Toasty :



DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Snackbars

- Afficher des messages et interagit avec l'utilisateur :



Méthodes statiques de la classe Snackbar

`make(Context context, CharSequence text, int duration)`

Exemple

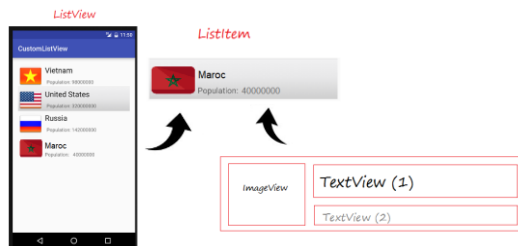
```
import android.support.design.widget.Snackbar;
...
Snackbar s = Snackbar.make(this, "msg", Snackbar.LENGTH_INDEFINITE);
s.setAction("UNDO", new View.OnClickListener() {
    @Override
    public void onClick(View v) { ... }
});
s.show();
```

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Widgets & Event

Les listes (ListView) - Rappel

- Une ListView est fait à partir d'un groupe de ListItem.
- Une ListItem est une pièce de l'interface qui peut être créée par un nombre de View.



DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Widgets & Event

Les listes (ListView) - Rappel

- Le code Java:

```
import android.app.Activity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;

public class TutolistesActivity extends Activity {
    ListView liste = null;

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

        liste = (ListView) findViewById(R.id.listView);
        List<String> exemple = new ArrayList<String>();
        exemple.add("Item 1");
        exemple.add("Item 2");
        exemple.add("Item 3");

        ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_1, exemple);
        liste.setAdapter(adapter);
    }
}
```

Item 1
Item 2
Item 3

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Widgets & Event

Les listes (ListView)

- L'évènement :

```
liste.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view,
        int position, long id) {
        showdetail(Publication.bonocas.get(position).numpub);
    }
});
```

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

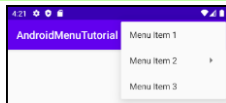
Les événements Widgets & Event

Les menus d'option (Menu/MenuItem) - Rappel

- En XML:


```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:title="Menu Item 1" />
  <item android:title="Menu Item 2" >
    <menu>
      <item android:title="Menu Item 2.1" />
      <item android:title="Menu Item 2.2" />
    </menu>
  </item>
  <item android:title="Menu Item 3" />
</menu>
```

- Visualisation :



DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Widgets & Event

Les menus d'option (Menu/MenuItem) - Rappel

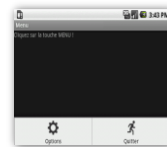
- En Java :

```
public boolean onCreateOptionsMenu(Menu menu) {
    MenuItem mn;
    mn=menu.add(0,1,0,"Menu1");
    mn.setIcon(R.drawable.icon_ref);

    mn=menu.add(0,2,0,"Menu2");
    mn.setIcon(R.drawable.icon_param);

    mn = menu.add(0,3,0,"Quitter");
    mn.setIcon(R.drawable.icon_out);

    return true;
}
```



DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Widgets & Event

Les menus d'option (Menu/MenuItem)

- L'évènement :

```
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case 1:
            //instructions...
            break;
        case 2:
            //instructions...
            break;
        case 3: //Quitter
            System.exit(0);
            break;
    }
    return true;
}
```

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Résumé

- Il est possible d'associer un évènement à un attribut XML :

res/layout/main.xml

```
<Button ... android:onClick="sayHello" .../>
```

HelloWorldActivity.java

```
public class HelloWorldActivity extends Activity {
    private TextView helloLabel;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        helloLabel = (TextView) findViewById(R.id.hello_label);
    }
    public void sayHello(View v) {
        helloLabel.setText(R.string.hello);
    }
}
```

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Résumé

- Ou bien :

```
import ...
public class myActivity extends Activity implements OnClickListener {
    public void onCreate(Bundle savedInstanceState) {
        ...
        Button b = (Button) findViewById(R.id.mButton);
        b.setOnClickListener(this);
    }
    public void onClick(View v) {
        Toast.makeText(this, "click", Toast.LENGTH_LONG).show();
    }
    ...
}
```

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événements Différents types d'événements

- Le fonctionnement des événements dépendant du type de composant et du dispositif d'entrée.
- Les principaux événements supportés par l'ensemble des vues :
 - **Click** : clic sur un composant
 - **longClick** : clic long sur un composant
 - **key** : appui sur une touche d'un clavier physique
 - **drag** : événement de glissé (lors d'un glissé)

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événementsDifférents types d'événements

Evénements généraux :

Les Views :

Evénement	Association Classe	Méthode • Paramètres
Clic	setOnClickListener View.OnClickListener	onClick(View) • Élément concerné
Clic long	setOnLongClickListener View.OnLongClickListener	onLongClick(View) • Élément concerné
Clavier	setOnKeyListener View.OnKeyListener	onKey(View, int, KeyEvent) • Élément concerné • Code clavier • Événement clavier
Touché	setOnTouchListener View.OnTouchListener	onTouch(View, MotionEvent) • Élément concerné • Événement de touché

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événementsDifférents types d'événements

Evénements spécifiques:

TextView et EditText :

Evénement	Association Classe	Méthode • Paramètres
Fin de saisie	setOnEditorActionListener TextWatcher	onEditorAction(TextView, int, KeyEvent) • Élément concerné • Code de l'action (EditorInfo.IME_ACTION_DONE) • Événement clavier (peut être null)
Modification	addTextChangedListener TextChangeListener	beforeTextChanged(CharSequence, int, int, int) afterTextChanged(CharSequence, int, int, int) • Texte • Point de départ de la modification • Nombre de cars remplacés • Nombre de cars de remplacement
Saisie	setKeyListener KeyListener	onKeyDown(View, Editable, int, KeyEvent) onKeyUp(View, Editable, int, KeyEvent) • Élément concerné • Texte • Code de la touche • Événement clavier

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événementsDifférents types d'événements

Evénements spécifiques:

ListView , GridView , Spinner et Gallery :

Evénement sur un élément	Association Classe	Méthode • Paramètres
Clic	setOnClickListener AdapterView.OnItemClickListener	onItemClick(AdapterView<View>, int, long) • Vue concernée • Élément cliqué • Rang de cet élément • ID de cet élément
Sélection	setOnItemSelectedListener AdapterView.OnItemSelectedListener	onItemSelected(AdapterView<View>, View, int, long) onNothingSelected(AdapterView<View>) • Idem

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

Les événementsDifférents types d'événements

Autres événements :

onClick	OnClickListener() : Ceci est appelé lorsque l'utilisateur touche l'élément (en mode tactile), ou se concentre sur l'élément avec les touches de navigation ou la boule de commande et appuie sur la touche "Entrée" appropriée ou appuie sur la boule de commande.
onLongClick	OnLongClickListener() : Ceci est appelé lorsque l'utilisateur touche et tient l'objet (en mode tactile), ou se concentre sur l'élément avec les touches de navigation ou la boule de commande et appuie sur la touche "Entrée" appropriée ou appuie sur la boule de commande (pour une seconde).
onFocusChange	OnFocusChangeListener() : Ceci est appelé lorsque l'utilisateur exécute une action qualifiée d'événement tactile, y compris une pression, une libération ou un geste de mouvement sur l'écran (dans les limites de l'élément).
onKey	OnKeyListener() : Ceci est appelé lorsque l'utilisateur est concentré sur l'élément et appuie ou libère une clé matérielle sur l'appareil.
onTouch	OnTouchListener() : Ceci est appelé lorsque l'utilisateur appuie sur la touche, relâche la touche, ou tout geste de mouvement à l'écran.
onMenuItemClick	OnMenuItemClickListener() : Ceci est appelé lorsque l'utilisateur sélectionne un élément de menu.
onCreateContextMenu Menu	onCreateContextMenuListener() : Ceci est appelé lors de la construction d'un menu contextuel (résultat d'un long clic prolongé)

DÉVELOPPEMENT POUR MOBILES (ANDROID) - DR. KHALIFI HAMID

