

CRÉATION D'INTERFACE UTILISATEUR (3)

USER INTERFACE CREATION (3)

CRÉATION D'APPLICATIONS ET DÉCOUVERTE DES ACTIVITÉS ET DES INTENTS

CREATION OF APPLICATIONS AND DISCOVERY OF ACTIVITIES AND INTENTS

E.I. Djebbar

Département de Génie des systèmes informatiques

Department of Computer Systems Engineering

PRINCIPE DE FONCTIONNEMENT DES INTENTS

- Les objets Intent ont essentiellement trois utilisations : ils permettent de démarrer une activité au sein de l'application courante ou de solliciter d'autres applications et d'envoyer des informations.

NAVIGUER ENTRE ÉCRANS AU SEIN D'UNE APPLICATION

- Une application est souvent composée de plusieurs écrans qui s'enchaînent les uns à la suite des autres en fonction de l'utilisateur et chaque écran est représenté par une activité définissant son interface utilisateur et sa logique. La principale utilisation d'un Intent est le démarrage de ces activités (une à la fois) permettant cet enchaînement.
- De façon plus générale, chaque composant d'une application nécessitera l'emploi d'un Intent pour être démarré

DÉMARRER UNE ACTIVITÉ

- Pour démarrer une activité sans attendre de retour, utilisez la méthode `startActivity` avec comme paramètre une instance de la classe `Intent` spécifiant le type de classe de l'activité à exécuter :

```
Intent intent = new Intent(this,ActiviteADemarrer.class);
```

```
startActivity(intent);
```

DÉMARRER UNE ACTIVITÉ ET OBTENIR UN RETOUR

- Lancer une activité enfant et en connaître sa valeur de retour avec la méthode `startActivityResult`

...

```
private static final int CODE_MON_ACTIVITE = 1;
```

...

```
Intent intent = new Intent(this,lambda.class);
```

```
startActivityResult(intent, CODE_MON_ACTIVITE);
```

RENNVOYER UNE VALEUR

```
@Override  
public void onClick(View v) {  
    switch(v.getId()){  
        case R.id.button1:  
            setResult(RESULT_OK);  
            finish();  
            break;  
        case R.id.button2:  
            setResult(RESULT_CANCELED);  
            finish();  
            break;  
    }  
}
```

RÉCUPÉRER LA VALEUR DE RETOUR

AGIR EN FONCTION DE LA VALEUR DE RETOUR

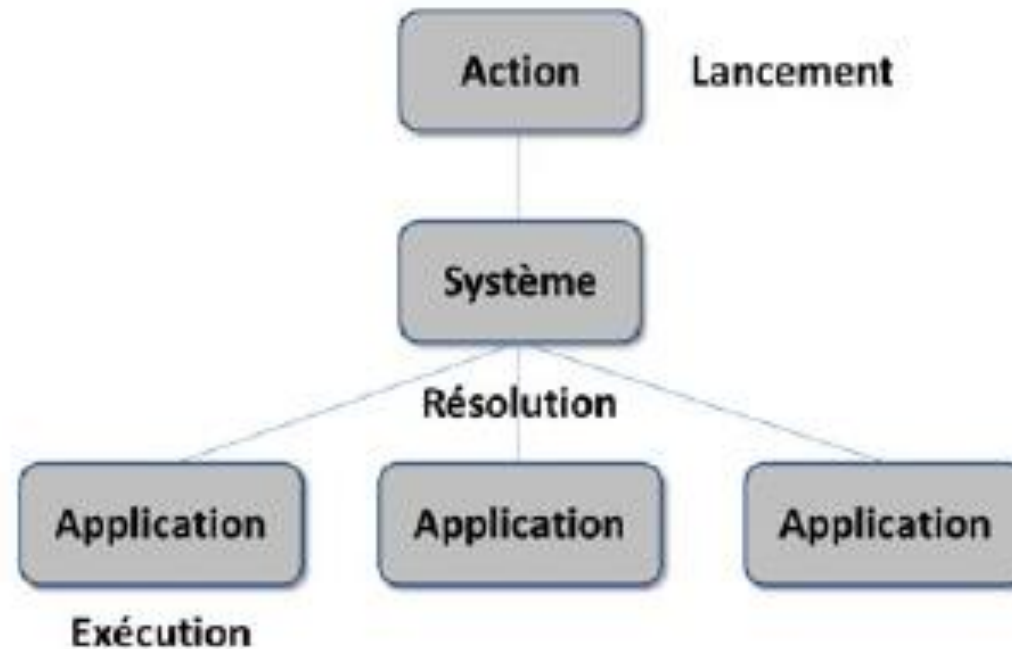
@Override

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    switch(requestCode){  
        case CODE_MON_ACTIVITE:  
            switch(resultCode){  
                case RESULT_OK:  
                    Toast.makeText(this, "Action validée", Toast.LENGTH_LONG).show();return;  
                case RESULT_CANCELED:  
                    Toast.makeText(this, "Action annulée", Toast.LENGTH_LONG).show();return;  
                default: return;  
            }  
        default: return;  
    }  
}
```

SOLLICITER D'AUTRES APPLICATIONS

- L'envoi d'un Intent permet également de demander à un composant d'une autre application que la vôtre de traiter l'action que vous souhaiteriez réaliser. C'est le système qui décide alors de l'application à utiliser pour accomplir votre souhait. Pour décider du composant le plus approprié, le système se base sur les informations que vous spécifiez dans votre objet Intent : action, données, catégorie, etc.

SOLLICITER D'AUTRES APPLICATIONS



DÉLÉGUER AU SYSTÈME LE CHOIX DE L'APPLICATION

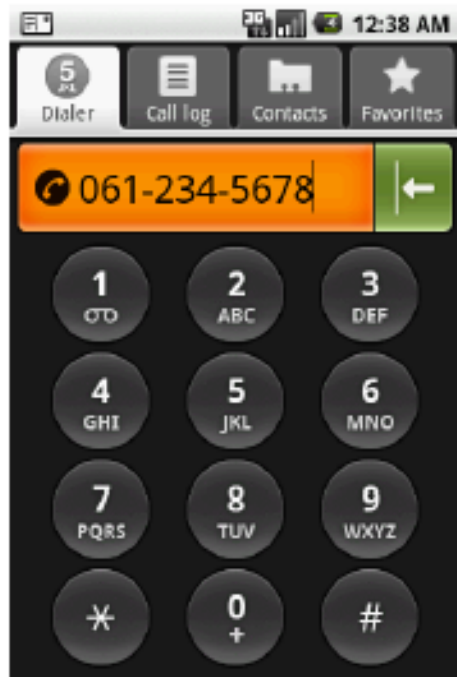
- Nous souhaitons composer un numéro de téléphone. Nous allons utiliser le type d'action `ACTION_DIAL`
- Composer un numéro de téléphone

```
Uri uri = Uri.parse("tel:0661553322");
```

```
Intent intent = new Intent(Intent.ACTION_DIAL, uri);
```

```
startActivity(intent)
```

DÉLÉGUER AU SYSTÈME LE CHOIX DE L'APPLICATION



ACTIONS NATIVES

Action	Définition
ACTION_ANSWER	Prendre en charge un appel entrant.
ACTION_CALL	Appeler un numéro de téléphone. Cette action lance une activité affichant l'interface pour composer un numéro puis appelle le numéro contenu dans l'URI spécifiée en paramètre.
ACTION_DELETE	Démarrer une activité permettant de supprimer une donnée identifiée par l'URI spécifiée en paramètre.

ACTIONS NATIVES

Action	Définition
ACTION_DIAL	Afficher l'interface de composition des numéros. Celle-ci peut être pré-remplie par les données contenues dans l'URI spécifiée en paramètre.
ACTION_EDIT	Éditer une donnée.
ACTION_SEARCH	Démarrer une activité de recherche. L'expression de recherche de la pourra être spécifier dans la valeur du paramètre SearchManager.QUERY envoyé en extra de l'action.
ACTION_SEND	Envoyer des données texte ou binaire par courriel ou SMS. Les paramètres dépendront du type d'envoi.
ACTION_SENDTO	Lancer une activité capable d'envoyer un message au contact défini par l'URI spécifiée en paramètre.
ACTION_VIEW	Démarrer une action permettant de visualiser l'élément identifié par l'URI spécifiée en paramètre. C'est l'action la plus commune. Par défaut les adresses commençant par <i>http:</i> lanceront un navigateur web, celles commençant par <i>tel:</i> lanceront l'interface de composition de numéro et celles débutant par <i>geo:</i> lanceront Google Map.
ACTION_WEB_SEARCH	Effectuer une recherche sur Internet avec l'URI spécifiée en paramètre comme requête.

DÉMARRER UN NAVIGATEUR INTERNET

- Démarrer un navigateur internet

```
private void doWebBrowser(){  
    Uri uri = Uri.parse("http://www.google.fr/");  
    Intent intent = new Intent(Intent.ACTION_VIEW, uri);  
    startActivity(intent);  
}
```

ACCORDER LES PERMISSIONS LIÉES AUX ACTIONS

- Certaines actions requièrent des privilèges spéciaux qui vous amèneront à ajouter des permissions dans le fichier de configuration de votre application.
- Par exemple, pour émettre un appel depuis votre application, il ne suffira pas uniquement de créer un objet Intent avec l'action ACTION_CALL.
- Il vous faudra également ajouter la permission android.permission.CALL_PHONE

ACCORDER LES PERMISSIONS LIÉES AUX ACTIONS

```
Uri uri = Uri.parse("tel:0612345678");  
Intent intent = new Intent(Intent.ACTION_CALL, uri);  
startActivity(intent);
```

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


LES INTENTS: EMBARQUER DES DONNÉES SUPPLÉMENTAIRES

- La classe Intent dispose de méthodes putExtra et getExtras grâce auxquelles un objet de type Bundle véhiculera vos données d'une activité à une autre.
- L'échange des données se fait par l'association d'une clé (représentée par une chaîne de caractères) à une donnée à échanger. Pour ajouter chaque donnée à échanger, il faut appeler la méthode putExtra avec la clé et la donnée à échanger. La méthode putExtra dispose d'une surcharge pour pratiquement tous les types de données primitifs : int, float, String, etc.

AJOUTER UNE DONNÉE À UN INTENT

```
Intent intent = new Intent(this, MonActivite.class);  
intent.putExtra("maclé", "hello !");  
startActivity(intent);
```

RÉCUPÉRER UNE DONNÉE D'UN BUNDLE

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    Bundle extra = this.getIntent().getExtras();  
    if(extra != null)  
        String data = extra.getString("maclé");  
}
```

DIFFUSER ET RECEVOIR DES INTENTS

- Un Intent peut être diffusé à l'ensemble des applications du système pour transmettre des informations à destination des autres applications, que ce soit pour demander la réalisation d'actions spécifiques, ou pour fournir des informations sur l'environnement (appel entrant, réseau Wi-Fi connecté, etc.).
- Ce mécanisme de diffusion permet à chaque processus de communiquer un état spécifique, une information ou une action de manière homogène à l'ensemble de ses pairs. Dans ce contexte, ces Intents sont alors appelés des *Broadcast Intents* alors que ses récepteurs sont appelés les *Broadcast Receiver* ou « récepteurs d'Intents ».

DIFFUSER UN INTENT

```
private void doBroadcast(){  
    Intent intent = new Intent(MyBroadcastReceiver.VIEW);  
    intent.putExtra( "extra", "hello !");  
    sendBroadcast(intent);  
}
```

LES MESSAGES D'INFORMATIONS NATIFS

- Android, à l'instar des actions, propose plusieurs actions prédéfinies et utilisées pour communiquer sur l'état d'un service ou le changement d'un composant du système.

LES MESSAGES D'INFORMATIONS NATIFS

Action	Définition
ACTION_BOOT_COMPLETED	Est diffusée lorsque le système a fini de démarrer. La permission RECEIVE_BOOT_COMPLETED est requise.
ACTION_SHUTDOWN	Indique que le système est en train de s'éteindre. Seul le système peut envoyer ce type de message.
ACTION_CAMERA_BUTTON	Survient lorsque le bouton de l'appareil photo est appuyé.
ACTION_DATE_CHANGED	Survient lorsque la date a été changée.
ACTION_TIME_CHANGED	Survient lorsque l'heure a été changée.
ACTION_SCREEN_ON / OFF	Permet d'être informé lorsque que l'écran s'éteint et s'allume.
ACTION_POWER_CONNECTED / DISCONNECTED	Diffusée lorsque l'alimentation électrique a été branchée / débranchée.
ACTION_PACKAGE_ADDED / REMOVED	Permet de prendre en charge un appel entrant.
ACTION_MEDIA_MOUNTED / UNMOUNTED	Survient lorsque qu'un support de stockage, une carte SD par exemple, a été correctement monté ou démonté du système de fichier de l'appareil.