


TP1

Les objectifs de ce TP sont :

1. Se familiariser avec l'environnement de développement
2. Comprendre le cycle de vie d'une application
3. S'initier à l'usage des intent pour communiquer avec les sous-activités ou d'autres activités.

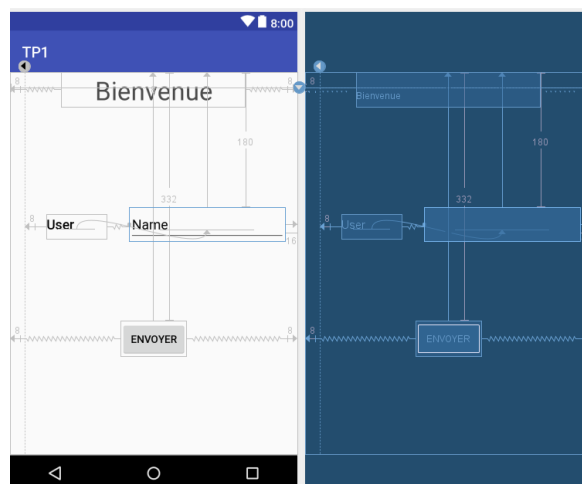
Préparation de l'environnement de développement :

- Installez la plate-forme (Version 3.5) via ce lien :
<http://developer.android.com/sdk/index.html>
- Allez vers le SDK Manager et ajoutez les différentes API nécessaires.
- Configurer un AVD pour tester votre projet ou utilisez votre téléphone :
 <http://developer.android.com/tools/devices/managing-avds.html>
- <http://developer.android.com/tools/devices/emulator.html>

EX1 : Cycle de vie d'une activité

- Créer un projet « TP1 » : File->New Project puis choisir :
 - o l'API target que vous voulez puis Empty Activity
- Visualiser le cycle de vie d'une activité en ajoutant successivement des `Log.i("tag", "message")` au niveau des fonctionnalités suivantes : (onCreate, onDestroy, onStart, onStop, onPause, onResume). Allez vers Logcat et filtrez les output pour visualiser vos messages

EX2 : Modifier l'écran d'accueil de l'application comme suit :



- Allez dans le layout « `activity_main.xml` » et ajoutez les Views nécessaires (`TextView`, `EditView`, `Button`) via Editeur (Design)



<https://developer.android.com/studio/write/layout-editor.html>

EX3 : Interaction avec les sous-activités

Les sous activités sont des tâches qui vont être appelées par le programme principal ou d'autres sous-activités.

- Pour accéder aux vues ajouter dans l'activité (avec les imports nécessaires):

```
button= (Button) this.findViewById(R.id.Boutton);  
edit= (EditText) this.findViewById(R.id.edit);
```

- Pour interagir avec la commande bouton ajouter :

```
button.setOnClickListener(new View.OnClickListener() {  
  
    public void onClick(View v) {  
  
        myClick(v);  
  
    }  
  
});
```

- interagir avec une sous activité par envoi de donnée :
 - o Dans la méthode myClick ajouter les lignes suivantes (⚠on ne peut pas accéder directement à la class main_activity c'est au système android de le faire) :
 - Ajouter dans myClick les lignes suivantes

```
//création de l'intention  
Intent intent= new Intent(this, MainActivity.class );  
//envoyer une donnée à la sous activité  
intent.putExtra("madonne", "Hi");  
// Démarrer la sous-activité  
this.startActivity(intent) ;
```

- o Dans onCreate ajoutez le code suivant :

```
//récupérer la donnée par la sous-activité  
if (this.getIntent().getExtras()!=null) {  
  
    String s = this.getIntent().getExtras().getString("madonne");  
    edit.setText(s);  
  
}
```



- L'*intent* permet de transporter des informations (Extra) à destination de l'activité cible (sous-activité)
 - putExtra : insérer des extras

Développement des applications mobiles (Android)

- `getStringExtra` : récupérer les extraits d'un intent
 - o Lancer le projet pour exécution
 - **interagir** avec une sous activité avec un retour d'information :
 - o dans `onMyClick` remplacer `this.startActivity` par
- ```
this.startActivityForResult(intent, 100)
```
- o dans la méthode `onCreate` ajouter :

```
//la donnée à envoyer par la sous-activité
Intent data = new Intent();
data.putExtra("Val", "Bonjour");
// Activité ayant le code 2000 retourne data
this.setResult(2000, data);
```

- o Ajoutez dans l'activité la méthode:  
`onActivityResult(int requestCode, int resultCode, Intent data)`

```
if (requestCode==100){
 if (resultCode==2000){
 edit.setText("" + data.getExtras().getString("Val"));
 }
}
```

- o Lancez le projet et cliquez sur le bouton puis mettre « Echap (retour) » pour voir l'interaction entre l'activité et la sous-activité

### Ex4 : (à vous)

- **Gérer l'interaction avec une autre activité**
- **Personnaliser votre interface**