Ecole d'ingénieurs et d'architectes de Fribourg
Hochschule für Technik und Architektur Freiburg

Mobile development 3

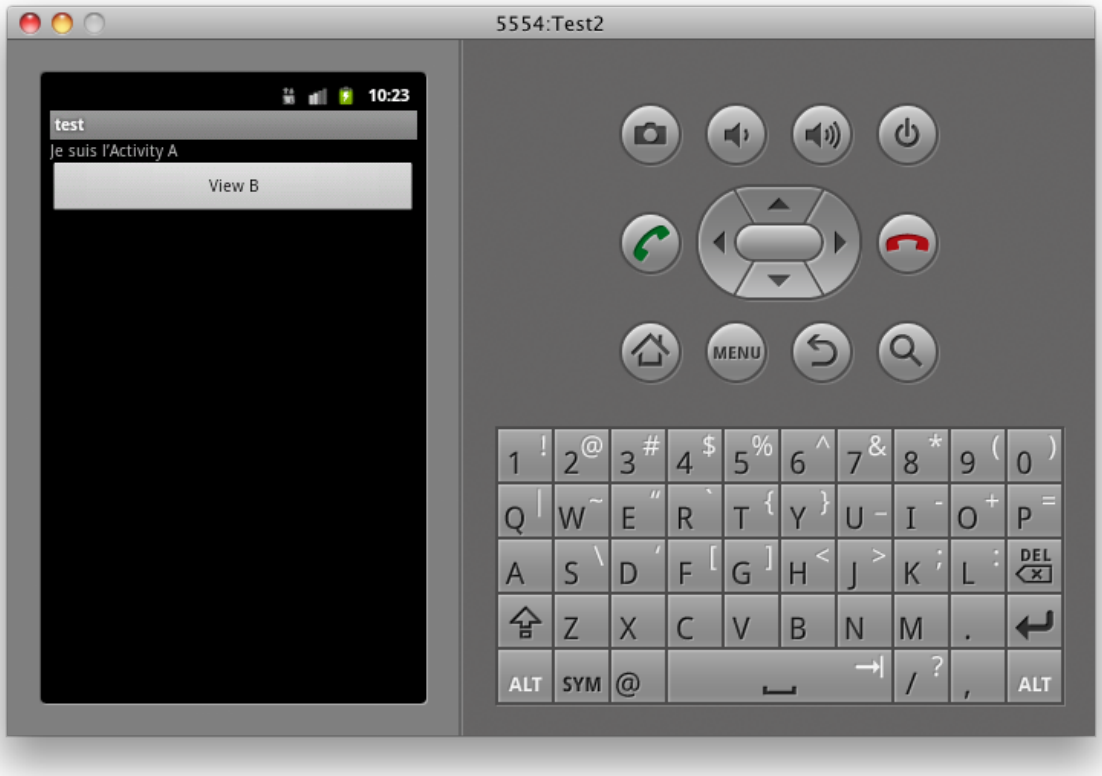
Report TD 5 :

Jonathan Stoppani et Elias Medawar

Version : 30 mars 2011

1 Exercice : 1

1.1 Activité avec changements d'états

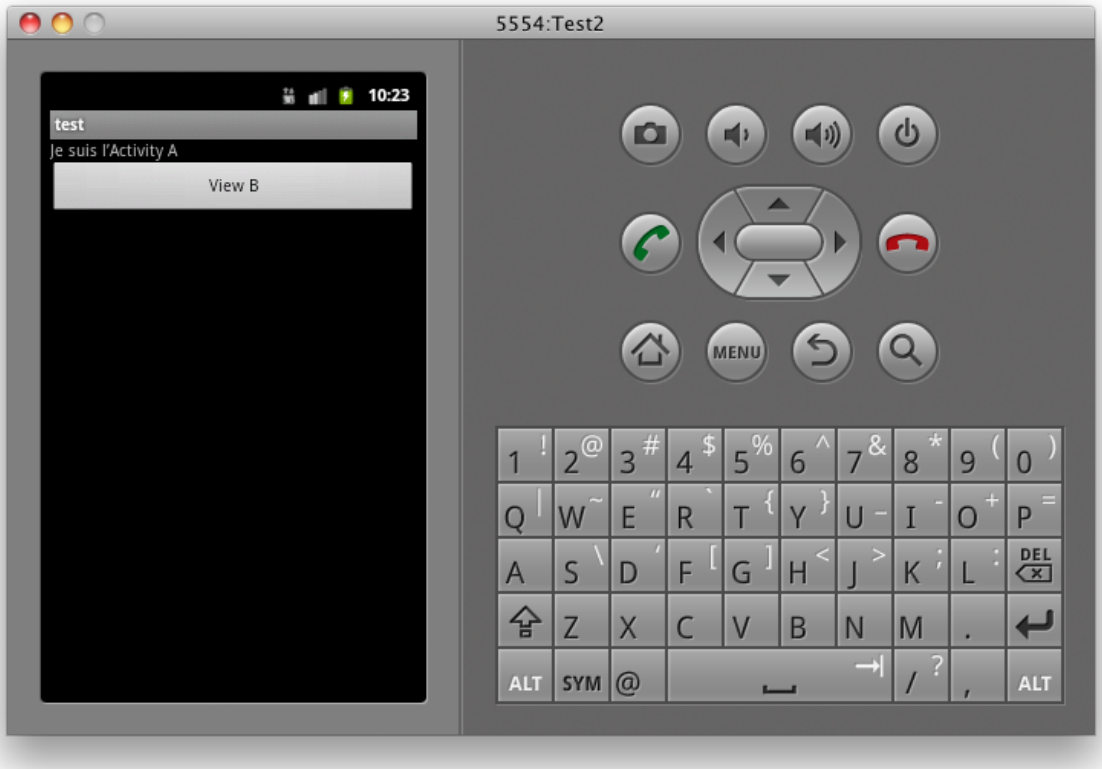


Une activité basique avec un texte et un bouton

Time	pid	tag	Message
03-24 15:55:22. I	652	Ex1	onStart()
03-24 15:55:22. I	652	Ex1	onResume()
03-24 15:55:28. I	652	Ex1	Click on button
03-24 15:55:33. I	652	Ex1	onPause()
03-24 15:55:34. I	652	Ex1	onStop()
03-24 15:55:34. I	652	Ex1	onDestroy()
03-24 15:55:46. I	652	Ex1	onStart()
03-24 15:55:46. I	652	Ex1	onResume()
03-24 15:55:53. I	652	Ex1	onPause()
03-24 15:55:54. I	652	Ex1	onStop()
03-24 15:56:01. I	652	Ex1	onRestart()
03-24 15:56:01. I	652	Ex1	onStart()
03-24 15:56:01. I	652	Ex1	onResume()
03-24 15:56:04. I	652	Ex1	Click on button

Le résultat des différents passages d'états.

1.2 ActivityA avec changements d'états



Une activité basique avec un texte et un bouton

Time	pid	tag	Message
03-24 15:55:22. I	652	Ex1	onStart()
03-24 15:55:22. I	652	Ex1	onResume()
03-24 15:55:28. I	652	Ex1	Click on button
03-24 15:55:33. I	652	Ex1	onPause()
03-24 15:55:34. I	652	Ex1	onStop()
03-24 15:55:34. I	652	Ex1	onDestroy()
03-24 15:55:46. I	652	Ex1	onStart()
03-24 15:55:46. I	652	Ex1	onResume()
03-24 15:55:53. I	652	Ex1	onPause()
03-24 15:55:54. I	652	Ex1	onStop()
03-24 15:56:01. I	652	Ex1	onRestart()
03-24 15:56:01. I	652	Ex1	onStart()
03-24 15:56:01. I	652	Ex1	onResume()
03-24 15:56:04. I	652	Ex1	Click on button

Le résultat des différents passages d'états.

1.3 ActivityB depuis activityA

```
protected void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
  setContentView(R.layout.main);
  Button b = (Button)findViewById(R.id.btnCreateView);
  b.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
```

```
      Log.i("Ex1","Click␣on␣button");
      Intent intent = new Intent(getApplicationContext(),ActivityB.class);
      startActivity(intent);
    }
  });
}
```

Listing 1 – Code pour capturer l'envement du click sur le bouton

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

Listing 2 – Code de l'activityB

2 Exercice : 2

Le code source des interfaces graphique ainsi que des actions étant déjà fournit dans l'exercice, il ne sera pas insérer dans le rapport. Néanmoins il est disponible sur gitHub à l'adresse : http ://gitHub.com

2.1 Standard

1. Dès le clic sur le bouton, un erreur se produit et l'application se termine. Ce résultat n'est pas étonnant vu qu'on a pas encore déployer(installer) l'application 2. En dehors de cela les bons écrans sont affiché.

FIGURE 1 – Erreur application non existante

2. Les écrans sont affichés dans le bon ordre(Activity 1 - 2 -3 - A) et sans problème. Dans la données il n'était pas indiqué que dans le ficher *AndroidManifest.xml* 3 de l'application 1 l'activityA devait répondre au filtre depuis une autre application(comme pour l'activity2). Nous avons conclus que c'était un oubli et avons tout de même insérer le code3 qui nous de faire permet cet opération. Nous ne sommes pas encore capable d'expliquer exactement ce code mais ceci sera traité dans la prochaine leçon.

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3        package="ch.eiafr.tic.application1"
4        android:versionCode="1"
5        android:versionName="1.0">
6
7      <application android:icon="@drawable/icon" android:label="Application1">
8          <activity android:name=".ActivityA"
9                    android:label="@string/app_name">
10             <intent-filter>
11                 <action android:name="android.intent.action.MAIN" />
12                 <category android:name="android.intent.category.LAUNCHER" />
13             </intent-filter>
```

```
14              <intent-filter>
15                  <action android:name="ch.eiafr.tic.application1.VIEW_ACTIVITYA" />
16                  <category android:name="android.intent.category.DEFAULT" />
17              </intent-filter>
18          </activity>
19          <activity android:name=".ActivityB"
20                    android:label="@string/app_name">
21          </activity>
22      </application>
23  </manifest>
24  }
```

Listing 3 – Fichier AndroidManifest de l'application 1 avec un filtre(ligne 14 à 16) permetant l'accés à l'activityA depuis une autre application

3. Refaire le point 1.
   Les fenêtre s'affiche dans le bon ordre et sans aucune erreur. Ceci semble logique, vu que l'on a désormais les 2 applications qui sont désormais installé.

2.2 SingleTask

1. Démarrer l'Activity1 (via un clique sur son icône), affiche l'activity1
2. Cliquer sur le bouton de l'Activity1, affiche l'activity2
3. Cliquer sur le bouton de l'Activity2, affiche l'activity3
4. Cliquer sur le bouton HOME, affiche la page principale du système.
5. Démarrer l'ActivityA (via un clique sur son icône), affiche l'activityA
6. Cliquer sur le bouton de l'ActivityA, affiche l'activity B.
7. Cliquer sur le bouton de l'ActivityB, affiche l'activity2
8. Cliquer sur le bouton BACK,affiche l'activityB
9. Cliquer sur le bouton BACK,affiche l'activityA
10. Cliquer sur le bouton BACK, affiche la fenêtre de sélection(lancement) d'application.
11. Démarrer l'Activity1 (via un clique sur son icône),affiche l'actvity 3.
12. Cliquer sur le bouton BACK, affiche l'activity 2
13. Cliquer sur le bouton BACK , affiche l'activity 1
14. Cliquer sur le bouton BACK , affiche la fenêtre de sélection(lancement) d'application.

Modification Activity2 en Single Task

```
<activity android:name="Activity2"
                  android:label="@string/app_name"
                  android:launchMode="singleTask">
            <intent-filter>
              <action android:name="ch.eiafr.tic.application2.VIEW_ACTIVITY2"/>
              <category android:name="android.intent.category.DEFAULT"/>
           </intent-filter>
</activity>
```

Listing 4 – Modification dans le fichier manifest pour mettre l'activity2 accésible en mode singleTask

1. Démarrer l'Activity1 (via un clique sur son icône), affiche l'activity1
2. Cliquer sur le bouton de l'Activity1, affiche l'activity2
3. Cliquer sur le bouton de l'Activity2, affiche l'activity3
4. Cliquer sur le bouton HOME, affiche la page principale du système.
5. Démarrer l'ActivityA (via un clique sur son icône), affiche l'activityA
6. Cliquer sur le bouton de l'ActivityA, affiche l'activity B.
7. Cliquer sur le bouton de l'ActivityB, affiche l'activity2
8. Cliquer sur le bouton BACK,affiche l'activity1

9. Cliquer sur le bouton BACK,affiche l'activityB
10. Cliquer sur le bouton BACK,affiche l'activityA
11. Cliquer sur le bouton BACK , affiche la fenêtre de sélection(lancement) d'application.
12. Démarrer l'Activity1 (via un clique sur son icône),affiche l'actvity 1.

Le changement commence à l'étape 8 quand on clique BACK, l'activity1 sera affiché au lieu de l'activityB est une fois les activity de la tache courante terminé, on reprendra à au sommet de la pile de la tache précédente. Ceci est dû à la spécification du mode singleTask. Explication trouvé dans les slides du cours à la page 11.

2.3 SingleInstance

1. Démarrer l'Activity1 (via un clique sur son icône), affiche l'activity1
2. Cliquer sur le bouton de l'Activity1, affiche l'activity2
3. Cliquer sur le bouton de l'Activity2, affiche l'activity3
4. Cliquer sur le bouton BACK, affiche l'activity1
5. Cliquer sur le bouton BACK, affiche l'activity2
6. Cliquer sur le bouton BACK, affiche la fenêtre de sélection(lancement) d'application.
7. Cliquer sur le bouton BACK, affiche la page principale du système.

Deuxième séquence

1. Démarrer l'ActivityA (via un clique sur son icône), affiche l'activityA
2. Cliquer sur le bouton de l'ActivityA, affiche l'activity B.
3. Cliquer sur le bouton de l'ActivityB, affiche l'activity2
4. Cliquer sur le bouton de l'Activity affiché, affiche l'activity3
5. Cliquer sur le bouton HOME, affiche la page principale du système.
6. Démarrer l'Activity1 (via un clique sur son icône), affiche l'activity3
7. Cliquer sur le bouton BACK, affiche la fenêtre de sélection(lancement) d'application.
8. Démarrer l'ActivityA (via un clique sur son icône), affiche l'activityB
9. Cliquer sur le bouton BACK, affiche l'activityA
10. Cliquer sur le bouton BACK, affiche la fenêtre de sélection(lancement) d'application.

Il existe qu'une activity par tasks c'est pour cela qu'au point 7 on reviendra pas à l'activity 2 mais au menu de sélection d'application.(http ://developer.android.com/guide/topics/fundamentals/tasks-and-back-stack.html)

2.3.1 Questions

A quoi doit t'on faire attention si l'Activity est une activity singleInstance ? Quand on redémarre une application (exemple point 6 de la dernière séquence) l'historique de la navigation n'est pas garder. En effet on garde la task avec juste une activity et donc les autres activity ne seront plus mémorisé.

Pourquoi doit on s'assurer que les 2 applications ne soient pas déjà démarrer avant d'effectuer les séquences ? Si elles étaient déjà démarrer, on verrait au démarrage la dernière activity visité et non la première de l'application.

2.4 SingleTop

```
1 public class ActivityB extends Activity {
2  @Override
3  protected void onCreate(Bundle savedInstanceState) {
4   setContentView(R.layout.activityb);
5   super.onCreate(savedInstanceState);
6   Button b = (Button)findViewById(R.id.Bto2);
7   b.setOnClickListener(new OnClickListener() {
8    @Override
```

```
9       public void onClick(View v) {
10       Log.i("Ex2","Click on button Bto2");
11       Intent intent = new Intent(getApplicationContext(),ActivityB.class);
12       startActivity(intent);
13      }
14     });
15    }
16   }
```

Listing 5 – Modification faites dans la classe ActivityB pour ouvrir l'activityB quand on presse le bouton

1. Démarrer l'ActivityA (via un clique sur son icône), affiche l'activityA
2. Cliquer sur le bouton de l'ActivityA, affiche l'activity B.
3. Cliquer sur le bouton de l'ActivityB, affiche l'activityB
4. Cliquer sur le bouton BACK, affiche l'activityB.
5. Cliquer sur le bouton BACK, affiche l'activityA.
6. Cliquer sur le bouton BACK, affiche la fenêtre de sélection(lancement) d'application.

```
1  <activity android:name=".ActivityB"
2      android:label="@string/app_name"
3      android:launchMode="singleTop">
4  </activity>
```

Listing 6 – Modification dans le fichier manifest pour mettre l'activityB en mode singleTop

1. Démarrer l'ActivityA (via un clique sur son icône), affiche l'activityA
2. Cliquer sur le bouton de l'ActivityA, affiche l'activity B.
3. Cliquer sur le bouton de l'ActivityB, ne change pas d'activity
4. Cliquer sur le bouton BACK, affiche l'activityA.
5. Cliquer sur le bouton BACK, affiche la fenêtre de sélection(lancement) d'application.
6. Cliquer sur le bouton BACK, affiche la page principale du système.

Avec le mode singleTop, l'activityB ne sera pas crée plus d'une fois.
Toute les sources sont téléchargeable et accessible sur :