

Trimestre Été, 2014

Mohamed Lokbani

IFT1155 – Examen Intra –

Inscrivez tout de suite : votre nom et le code permanent.

Nom : _____ | Prénom(s) : _____ |

Signature : _____ | Code perm : _____ |

Date : jeudi 05 juin 2014

Durée : 2 heures (de 17h30 à 19h30)

Local : 3195; Pavillon André-Aisenstadt

Directives :

- Toute documentation est permise.
- Calculatrice **non** permise.
- Répondre directement sur le questionnaire.
- Les réponses **doivent être brèves, précises, claires et nettement présentées.**

1. _____ /20 (1.1 à 1.10)
2. _____ /20 (2.1, 2.2, 2.3, 2.4, 2.5)
3. _____ /20 (3.1)
4. _____ /40 (4.1 4.2)

Total : _____ /100

Directives officielles

* Interdiction de toute communication verbale pendant l'examen.

* Interdiction de quitter la salle pendant la première heure.

* L'étudiant qui doit s'absenter après la première heure remettra sa carte d'étudiant au surveillant, l'absence ne devant pas dépasser 5 minutes. Un seul étudiant à la fois peut quitter la salle.

* Toute infraction relative à une fraude, un plagiat ou un copiage est signalée par le surveillant au directeur de département ou au professeur qui suspend l'évaluation.

F.A.S

Exercice 1 (20 points) Répondre par « vrai » ou « faux » en y incluant une très courte explication.

- 1.1** [VRAI | FAUX] La plate-forme Android « KitKat » est représentée par l'API numéro 10.
- 1.2** [VRAI | FAUX] Le plug-in « ADT » est composé d'un ensemble d'outils permettant d'intervenir sur le processus de développement d'une application Android.
- 1.3** [VRAI | FAUX] « Eclipse » apporte trois perspectives propres au développement d'Android.
- 1.4** [VRAI | FAUX] Le développeur a un contrôle total sur les activités (activity).
- 1.5** [VRAI | FAUX] L'activité « A » lance l'activité « B » : A est empilé sur B.
- 1.6** [VRAI | FAUX] L'activité « A » lance l'activité « B », on clique sur la touche « retour » : « B » est détruite et dépilée.
- 1.7** [VRAI | FAUX] L'activité « A » lance l'activité « B », on clique sur la touche « home » : la tâche est mise en arrière-plan.
- 1.8** [VRAI | FAUX] L'activité « A » lance l'activité « B » : le système « Android » peut détruire ces deux activités s'il a besoin de ressources.
- 1.9** [VRAI | FAUX] Chaque application indique dans son manifeste le type d'intention qui peut l'appeler.
- 1.10** [VRAI | FAUX] Un service n'a pas de durée définie.

Exercice 2 (20 points) Répondre aux questions en y incluant une courte explication.

2.1 Quelle est l'importance de fixer les permissions lors du développement de votre application?

2.2 Quelles sont les étapes à suivre afin de vous permettre d'utiliser un appareil physique (tablette ou téléphone) pour le développement d'applications Android sous un système d'exploitation dont vous êtes familier (« Windows » ou « Mac » ou « Linux »).

2.3 Comment obtenir les messages d'un « handler »?

2.4 Quelle est la différence entre un « intent » associé à une activité et un « intent » associé à un « broadcast receiver »?

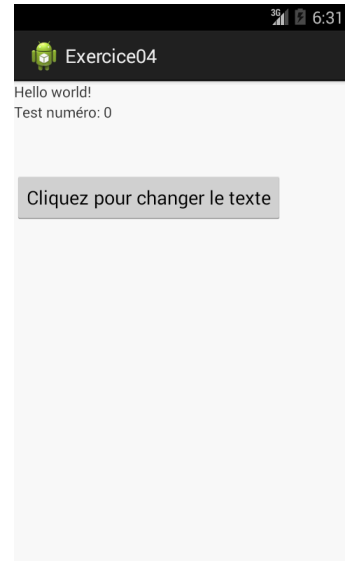
2.5 À travers un exemple simple comme le téléchargement d'une application Android quelconque du « Google Play store », expliquez le fonctionnement, et l'interaction entre elles, des composantes suivantes : « Activity », « Service » et « Broadcast Receiver ».

Exercice 3 (20 points) Dessiner l'interface graphique à partir du fichier XML ci-dessous, en prenant soin d'inclure un résumé de votre démarche.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" >
    <TextView
        android:id="@+id/Label"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/Courriel" />
    <EditText
        android:id="@+id/inputEmail"
        android:inputType="textEmailAddress"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/Label" />
    <Button
        android:id="@+id/btnLogin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@id/inputEmail"
        android:layout_marginRight="10dp"
        android:text="@string/Login" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@id/btnLogin"
        android:layout_toRightOf="@id/btnLogin"
        android:text="@string/Annuler" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:text="@string/Enregistrer" />
</RelativeLayout>
```


Exercice 4 (40 points) Nous allons reprendre l'activité de l'exemple « BasicThraed », mentionné dans le cours, afin d'améliorer son fonctionnement. Pour rappel, en cliquant sur le bouton « Cliquez pour changer le texte », on met en veille le thread un certain temps. Dès que le temps est écoulé, on change le contenu du texte. Le même processus se répète si on clique une 2^e fois, une 3^e fois, une énième fois. Malheureusement, l'interface n'est pas assez explicite pour montrer que nous sommes en train de tester son fonctionnement pour la énième fois. Pour y remédier, nous allons ajouter un compteur qui s'incrémente à chaque fois après le changement de texte. Ce compteur est représenté par l'étiquette « Test numéro ». Au départ, la valeur du compteur est à « 0 », après un clic, elle passe à « 1 », etc.

4.1 (20 points) Écrire le contenu des fichiers « activity_exercice04.xml » et « strings.xml » associés à l'activité représentée par l'interface graphique ci-dessous, en prenant soin d'inclure un résumé de votre démarche.



4.2 (20 points) Compléter le programme Java ci-dessous pour inclure le compteur décrit précédemment.

```
1 package ca.umontreal.iro.ift1155.exercice04;
2
3 import android.os.Bundle;
4 import android.app.Activity;
5 import android.view.View;
6 import android.widget.TextView;
7
8 public class Exercice04 extends Activity {
9
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_main);
14     }
15     public void buttonClick(View view){
16         long endTime = System.currentTimeMillis() + 20*100;
17
18         while (System.currentTimeMillis() < endTime) {
19             synchronized (this) {
20                 try {
21                     wait(endTime - System.currentTimeMillis());
22                 } catch (Exception e) {
23                 }
24             }
25         }
26         TextView myTextView = (TextView)findViewById(R.id.myTextView);
27         myTextView.setText("Bouton a été cliqué");
28     }
29 }
```

