

Programmation des Applications Mobiles avec Android
Licence Génie Informatique
TP/Td n°3: Intents – Multimedia

Exercice 1 – Convertisseur

L'application contient 05 fenêtres. La première est l'écran d'accueil proposant le menu de l'application avec trois boutons dont les deux permettront de charger les deux autres activités de l'application ainsi qu'un bouton quitter. Elle devra également contenir deux ImageButtons alignés sur la même ligne :

- About : affiche une nouvelle Activité en expliquant l'application et les formules utilisées...
- Support : affiche une Activité avec votre nom, email et votre numéro de téléphone (En cliquant dessus, on pourra vous appeler directement).

La seconde activité permet une conversion monétaire du CFA à l'Euro et respectivement de l'Euro au CFA (à noter 01 Euro=655,55 F CFA). La troisième activité pour une conversion Km ↔ Miles (à noter 01 KM=0,62 Miles). Chacune de ces deux activités contient les éléments suivants :

- Un titre sous la forme d'un TextView avec une couleur de fond rouge et une couleur du texte blanche écrit en gras ;
- 02 boutons radio (Conversion CFA en EURO) et (Conversion EURO en CFA) pour la première fenêtre et (Conversion Km en Miles) et (Conversion Miles en Km) pour la deuxième ;
- 01 zone de texte pour saisir la valeur à convertir ;
- 01 bouton pour calculer la valeur ;
- 01 TextView pour afficher le résultat ;
- 01 bouton retour permettant de revenir à la fenêtre principale de l'application ;
- Un bouton proposant d'envoyer le résumé de la conversion par SMS.

On souhaite également appliquer les éléments de design suivant :

1. Au moins deux familles de fonts ;
2. Les contenus des TextViews doivent être centrés ;
3. Des images de fonds différentes aux fenêtres ;
4. Une icône de votre choix à chaque bouton.

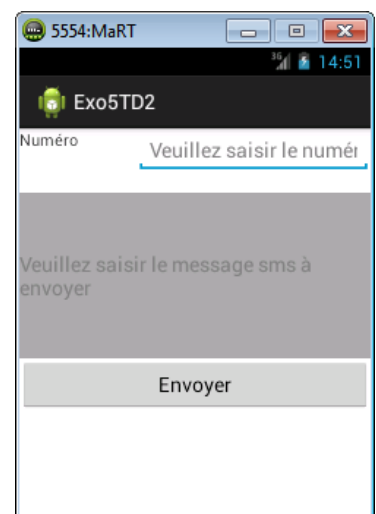
Exercice 2 – Envoi de SMS

Nous allons développer une petite application d'envoi de SMS qui a l'apparence de la figure

1. Créer un nouveau Projet puis modifier le fichier activity_main.xml et string.xml pour créer une activité ayant l'apparence de la figure ci-contre.
2. Pour envoyer un SMS contenant le message message à un numéro numero, il suffit d'utiliser la classe SMSManager de la façon suivante : `SmsManager.getDefault().sendTextMessage(numero, null, message, null, null);`

Ajouter une méthode sendSMS qui réagit à un clic sur le bouton Envoyer en envoyant le message par SMS au destinataire. La méthode vérifiera que le numéro contient au moins 4 chiffres et que le message n'est pas vide.

3. Pour tester votre application avec un émulateur, il vous définir un deuxième émulateur et démarrer l'application sur le 1er émulateur. Pour



envoyer un message d'un émulateur à un autre, il vous suffit de composer comme numéro les 4 chiffres qui se trouvent sur la fenêtre de l'émulateur comme dans la figure ci-dessous.

4. Modifier votre application pour que l'utilisateur puisse entrer plusieurs numéros séparés par un point virgule (;). Le message devra être envoyé à tous les destinataires. Tester votre application en utilisant 3 émulateurs et un utilisateur qui envoie un message aux 2 autres.

ATTENTION : votre clavier doit être en français sinon vous risquez des problèmes d'encodage de certains caractères. Pour changer la langue de saisie, presser longuement le champ texte. Dans le menu qui apparaît, choisissez Input Method puis Android Keyboard.



5. En utilisant la classe de synthèse vocale *TextToSpeech* de rajouter une fonction permettant de lire le message sms reçu en langue française.

Pour que tout ceci fonctionne il est essentiel d'autoriser les sms dans le fichier manifest.

Exercice 3 – Envoi d'un appel téléphonique d'émulateur à émulateur

Nous allons développer une petite application d'envoyer un appel téléphonique.

1. Créer un nouveau Projet puis modifier le fichier `activity_main.xml` et `string.xml` pour ajouter :
 - a. une zone de saisie de texte avec comme texte par défaut "Entrez un numéro de téléphone" permettant de saisir le numéro du téléphone cible de l'appel.
 - b. Un bouton ayant pour label "Appel" permettant de lancer l'appel téléphonique. Pour envoyer lancer un appel d'un émulateur à un autre, il vous suffit de composer comme numéro les 4 chiffres qui se trouvent sur la fenêtre de l'émulateur.
2. Pour lancer un appel d'un émulateur vers un autre, il suffit d'utiliser un Intent standard dans Android : **Intent.ACTION_CALL**, et utiliser un écouteur : **PhoneStateListener**, pour recevoir les événements permettant de suivre l'état de l'appel.

Pour passer un appel :

```
String n = "tel:5556";
```

```
Intent intent = new Intent( Intent.ACTION_CALL, Uri.parse( n ) );  
startActivity( intent );
```

3. Voici le code de l'activité en entier :

```

public class MakeACall_ltmActivity extends Activity {
    private Button      btn_1;
    private EditText    edit_1;

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

        TelephonyManager manager = (TelephonyManager) this.getSystemService( this.TELEPHONY_SERVICE );
        manager.listen( new TelListener(), PhoneStateListener.LISTEN_CALL_STATE );

        btn_1 = (Button)findViewById( R.id.button1 );
        edit_1 = (EditText)findViewById( R.id.editText1 );

        btn_1.setOnClickListener( new OnClickListener() {
            @Override
            public void onClick(View v) {
                Editable n = edit_1.getText();
                if( n.toString().equals("") ){
                    Toast.makeText( MakeACall_ltmActivity.this, "Mauvais numéro",
                                   Toast.LENGTH_SHORT ).show();
                    return;
                }
                Intent intent = new Intent(Intent.ACTION_CALL, Uri.parse( "tel:" + n.toString() ));
                startActivity(intent);
            }
        });
    }

    class TelListener extends PhoneStateListener {
        @Override
        public void onCallStateChanged(int state, String numero ) {
            if(TelephonyManager.CALL_STATE_RINGING == state)
                Log.i("LTM", "RINGING, numéro: " + numero);
            if(TelephonyManager.CALL_STATE_OFFHOOK == state)
                Log.i("LTM", "OFFHOOK");
            if(TelephonyManager.CALL_STATE_IDLE == state)
                Log.i("LTM", "IDLE");

            super.onCallStateChanged(state, numero);
        }
    }
}

```

Ci-dessus on utilise l'intent avec en paramètre le numéro à appeler précédé de "tel:". Pour suivre l'état de la communication on utilise un listener comme ci-dessous. La partie du code suivant est celle qui permet d'enregistrer l'écouteur décrit ci-avant.

TelephonyManager manager =(TelephonyManager) this.getSystemService(this.TELEPHONY_SERVICE);
manager.listen(new TelListener(), PhoneStateListener.LISTEN_CALL_STATE);

4. Si vous lancez le programme tel quel, cela ne fonctionnera pas ! Pourquoi ? Par souci de sécurité, par défaut, l'OS interdit certaines actions aux applications, comme par exemple les appels téléphoniques depuis une application tierce. Pour "autoriser" une application "à passer un coup de fil", il faut modifier le fichier "AndroidManifest.xml". Ajouter au manifest l'autorisation de lancer un appel téléphonique de la manière suivante :

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

Exercice 4: Intent entres classes

1. Créer un projet dont le premier écran a une TextView qui affiche: "Activité 1" et dispose d'un bouton avec comme label "Aller à l'activité 2". Le second écran de l'Activité 2 montrent une TextView qui affiche "Activité 2" et ont un bouton qui ayant pour label "Allez à l'activité 1".
6. Créer une autre activité qui prend deux nombres et leur somme. La façon la plus simple de le faire est de créer une activité SommeActivity contenant un TextView simple avec le texte "La somme de xx et yy zz

- est" qui sera lue en utilisant la classe de synthèse vocale *TextToSpeech* en langue française. Ajouter un bouton à votre activité initiale permettant d'invoquer la classe *SumActivity* avec des valeurs arbitraires.
2. Ajouter un autre bouton à l'activité initiale. Faites-le aussi invoquer la classe *SommeActivity*, mais en envoyant deux nombres aléatoires. (Rappelez-vous que *Math.random()* renvoie nombre un réel aléatoire compris entre 0,0 et 1,0) que *SommeActivity* va utiliser pour calculer la somme.
 3. Créer une activité avec deux zones de texte ainsi qu'un bouton permettant lorsque l'on appuie sur le bouton d'invoquer *SommeActivity* avec les deux valeurs saisies qui vont être récupérées pour calculer leur somme.

Exercice 5: Intent avec onglets

Modifier le projet précédent qui contient *SommeActivity*. Ajouter un bouton à l'écran initial qui permet de faire passer l'utilisateur à un écran qui a des fenêtres à onglets.

1. Dans le premier onglet invoquer *SommeActivity* par son nom de classe, sans données.
2. Dans le deuxième second onglet invoquer *SommeActivity* par son nom de classe, avec envoi de valeurs aléatoires.
3. Dans le troisième second onglet deux zones de texte ainsi qu'un bouton permettant lorsque l'on appuie sur le bouton d'invoquer *SommeActivity* avec les deux valeurs saisies qui vont être récupérées pour calculer leur somme.

Exercice 6: Lecteur multimédia

1. Créer une application permettant la lecture d'un fichier vidéo (Attention : les formats reconnus sont 3gp, ogg et wmv) avec le code permettant la gestion appropriée des événements. Votre application doit contenir quatre boutons radio avec quatre titres différents :
 - Un bouton Play pour jouer la piste sélectionnée
 - Un bouton Stop
 - Un bouton Aide qui affiche un *Toast* avec votre nom et des informations sur l'application
 - Un bouton About qui affiche dans une nouvelle fenêtre des informations sur le chanteur (la piste...)
2. Créer une application avec les mêmes caractéristiques pour la lecture d'un fichier audio.