Programmation Mobile

Examen - 1h30 - Notes de cours autorisées

20 Février, 2019

1 Questions

1.1 Layouts

1. L'application "Search by Image" permet d'effectuer des recherches visuelles en utilisant le moteur de recherche "Google Images" : à partir d'une image, on souhaite retrouver les images visuellement les plus proches disponibles, sur Internet. Pour cela, l'application (Figure 1) propose plusieurs possibilités :
   - chercher des images en donnant une description textuelle, *p.e.* l'utilisateur peut écrire le texte dans l'EditText ("*image to search (e.g. Cat memes*") de la première ligne de l'UI) ;
   - sélectionner une image parmi les recherches récentes (deuxième ligne) ;
   - sélectionner une image dans le "Gallery" ou prendre une photo avec la caméra du dispositif (troisième ligne).

   L'image sélectionnée est visualisée au centre de l'interface et, avant de démarrer la recherche, il est également possible d'appliquer des modifications à l'image avec les 4 boutons.

   Donnez une organisation possible de cette interface graphique en utilisant les layouts et les widgets Android. Spécifiez les **types de layout** à utiliser, **leur structure**, leurs **attributs d'alignement**, et les widgets contenus.

   (Il n'est pas nécessaire d'écrire un code XML complet, vous pouvez vous contenter d'écrire les éléments principaux et les attributs nécessaires)

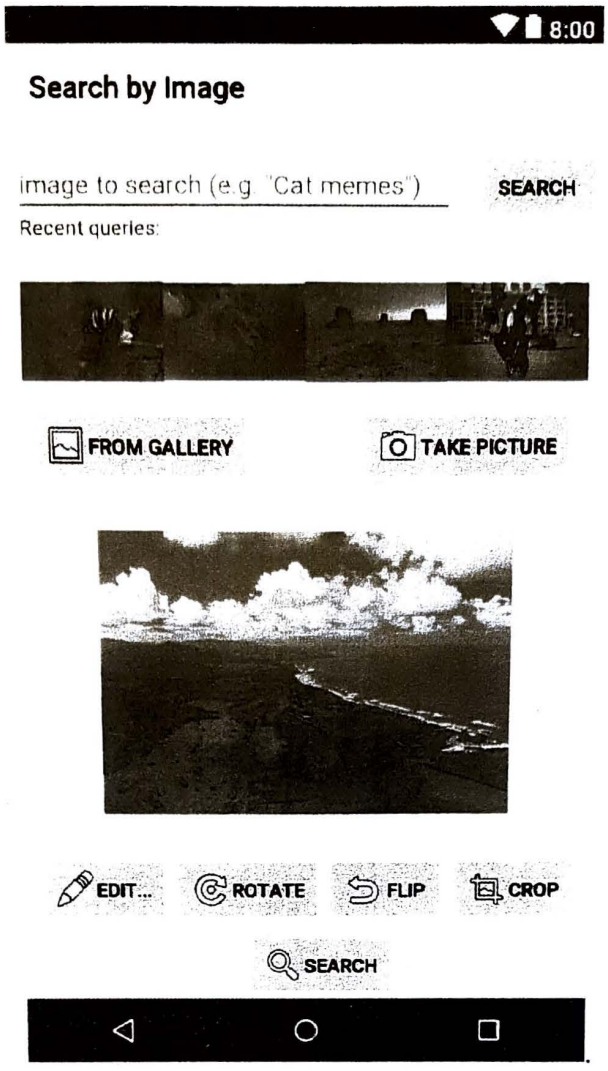

FIGURE 1 – La GUI de l'application "Search by Image" qui permet d'effectuer des recherches visuelles en utilisant Google Images.

values/strings.xml

<?xml version="1.0" encoding="utf-8"?>

<resources>

<string name="image_to_search">Image Search</string>

<string name="Recent queries">Recent Queries</string>

</resources>

2. Sur la Figure 1, les vignettes des images des recherches récentes ne sont pas bien espacées et le rendu n'est pas agréable visuellement. Proposez une solution pour mieux visualiser les 4 images, comme, *p.e.*, sur la Figure 2

FIGURE 2 – Layout de l'application avec un meilleur espacement des images.

La solution est d'utiliser le parametre marginLeft (ou marginRight) pour chaque bouton excepté le premier (ou le dernier) afin d'ajouter un espace entre les widgets pour les séparer les uns des autres

3. Expliquez ce qu'il faut mettre en place pour bien gérer l'internationalisation de l'application et, plus généralement, adapter une application Android à différentes langues. Comment faut-il organiser, *p.e.*, le texte associé à chaque widget de la Figure 1 ?

Il faut utiliser values/strings.xml afin de pouvoir changer facilement le texte selon la langue.
Ainsi chaque texte des widgets devra être appelé de cette façon @strings/....

1.2 Activities et Intents

1. Le bouton "Edit..." de la Figure 1 permet de lancer une autre application (*p.e.* Photoshop, Google Photos *etc.*) pour apporter des modifications à l'image sélectionnée. À l'aide de la Table 1, expliquez quel type de Intent est nécessaire ("Implicit", "Explicit" *etc.*) et quel type de "Action" est le plus adapté. En supposant quel'on ait un attribut `Uri imageUri` qui contient déjà le URI de l'image sélectionnée, essayez d'écrire le code minimal pour réaliser la fonctionnalité.

L'intent nécessaire est un intent implicite puisque seule l'action souhaitée est précisée et non la classe exacte dont on aura besoin.
Le type de Action le plus adapté est : ACTION_EDIT

```
public void onClick(View v)
{
    Intent newInt = new Intent(android.provider.MediaStore.ACTION_EDIT);
    newInt.putExtra(android.provider.MediaStore.EXTRA_OUTPUT, imageUri);
    startActivityForResult(newInt, CAMERA_RESULT);
}
```

2. L'"Activity" de la Figure 1 peut répondre à des intents pour les actions `ACTION_SEARCH` et `ACTION_WEB_SEARCH` (*c.f.* Table 1), alors que l'utilisateur est en train d'utiliser **une autre application** et décide, par exemple, de sélectionner une image et de chercher une image similaire. Indiquez ce qu'il faut mettre en place pour que l'activité puisse être connue et choisie par Android lors d'une demande de ces Intents.

Il faut que dans les activités pour sélectionner une image et chercher une image les intents ACTION_SEARCH et ACTION_WEB_SEARCH soient définis dans le manifest file.

3. Expliquez de quels choix vous disposez pour réaliser les deux fonctionnalités "From Gallery" et "Take Picture" en Android et ce qu'il faut mettre en place pour les réaliser. En particulier, donnez au moins deux possibilités pour réaliser la fonctionnalité "Take Picture" **Il n'est pas nécessaire d'écrire du code**.

Pour réaliser la fonctionnalité "Take Picture" nous pouvons soit utiliser une application externe pour prendre des photos en utilisant des intents ou soit créer une application customisée de caméra. Pour la fonctionnalité "From Gallery", l'utilisation d'intents est nécessaire avec un ACTION_PICK.

1.3 Permissions

1. Pour simplifier le développement des applications, Android permet de déclarer les "permissions" en utilisant des groupes de "permissions" (*c.f.* Figure 3) : les différentes "permissions" sont organisées par groupes, selon les fonctionnalités et le matériel à utiliser. Par exemple, le groupe "CALENDAR" regroupe l'ensemble des permissions permettant d'interagir avec le calendrier de l'utilisateur, comme lire les événements, créer un événement, *etc.* Autoriser un groupe de permission revient donc à autoriser toutes les permissions de ce groupe.
   À l'aide de la Figure 3, indiquez quels groupes de permissions sont nécessaires, à votre avis, pour l'application "Search by Image" **en motivant** chaque choix et en faisant des hypothèses sur le fonctionnement de l'application si nécessaire.

Constants

String	CALENDAR Used for runtime permissions related to user's calendar.
String	CAMERA Used for permissions that are associated with accessing camera or capturing images/video from the device.
String	CONTACTS Used for runtime permissions related to contacts and profiles on this device.
String	LOCATION Used for permissions that allow accessing the device location.
String	MICROPHONE Used for permissions that are associated with accessing microphone audio from the device.
String	PHONE Used for permissions that are associated telephony features.
String	SENSORS Used for permissions that are associated with accessing camera or capturing images/video from the device.
String	SMS Used for runtime permissions related to user's SMS messages.
String	STORAGE Used for runtime permissions related to the shared external storage.

FIGURE 3 – Les groupes de permissions.

Les groupes de permission nécessaires sont :
- CAMERA : en effet si on clique sur le bouton Take Picture, il faut avoir la permission d'utiliser l'outil caméra par appel d'intents
- STORAGE : en effet pour afficher les dernières recherches, il faut récupérer dans la mémoire ces dernières recherches, ou pour récupérer une image depuis la gallery il faut avoir accès au stockage
- SENSORS : Idem CAMERA
- LOCATION : pour pouvoir afficher l'image aussi récupérée

2. Quelles "permissions" sont nécessaires pour réaliser les fonctionnalités de la question 1.2.3 ("From Gallery" et "Take Picture"), selon les différentes réalisations que vous avez proposées ? **Motivez votre réponse**.

- CAMERA : Pour pouvoir utiliser l'outil caméra et prendre une photo
- STORAGE : Pour avoir accès aux éléments stockés dans le téléphone et pouvoir aussi les récupérer

3. Expliquez la différence entre `uses-features` et les "permissions". Donnez un exemple en utilisant l'application "Search by Image".

2 Question de TP

1. Imaginons de changer l'application "Web Browser" pour l'utiliser comme un "assistant personnel intelligent" qui lance des "intents" spécifiques selon le contenu introduit par l'utilisateur dans le champ `editTextInput`. Le code ci-dessous est une implantation possible pour le "listener" `onClick` du bouton "Go", qui permet d'analyser le contenu et de lancer l'intent correspondant. Complétez le code en ajoutant l'"Action" pour l'intent `tostart` pour chaque cas en choisissant parmi les "Actions" de la Table 1. On peut supposer que l'utilisateur introduit le texte avec le bon format `URI` nécessaire pour chaque action (*e.g.* "geo :47.6,-122.3" pour visualiser une position sur la carte géographique, *etc.*).

```
...
buttonGo.setOnClickListener( new Button.OnClickListener()
{

    public void onClick( View v )
    {
        stringPath = editTextInput.getText().toString();
        Intent tostart = new Intent();
        // check if the user has entered a valid web address
        if( URLUtil.isNetworkUrl( stringPath ) )
        {
            tostart.setAction( Intent.ACTION_VIEW );
            tostart.setDataAndType( Uri.parse( stringPath ), "text/*" );
        }
        // check if the user has entered a valid phone number
        else if( isValidPhoneNumber( stringPath ) )
        {
            // (no type required for the data)
```

tostart.setAction(Intent.ACTION_DIAL)
tostart.setData(Uri.parse(stringPath), "text/s");

```
        }
        // check if the user has entered a valid email address
        else if( isValidEmailAddress( stringPath ) )
        {
```

tostart.setAction(Intent.ACTION_SEND)
tostart.setDataAndType(Uri.parse("mailto:"), "text/plain")

```
        }
        // check if the user has entered a valid set of GPS coordinates
        else if( isValidGPSCoordinate( stringPath ) )
        {
            // (no type required for the data)
```

tostart.setAction(Intent.ACTION_PICK)

```
        }
        // search the Internet for the input text
        else
        {
            // (no type required for the data)
                tostart .setAction(Intent.ACTION_WEB_SEARCH)



        }
        startActivity( tostart );
    }
} );
```

Action	Description
ACTION_MAIN	Start as a main entry point, does not expect to receive data.
ACTION_VIEW	Display the data to the user.
ACTION_EDIT	Provide explicit editable access to the given data.
ACTION_PICK	Pick an item from the data, returning what was selected.
ACTION_CHOOSER	Display an activity chooser, allowing the user to pick what they want to before proceeding.
ACTION_GET_CONTENT	Allow the user to select a particular kind of data and return it.
ACTION_DIAL	Dial a number as specified by the data. This shows a UI with the number being dialed, allowing the user to explicitly initiate the call.
ACTION_CALL	Perform a call to someone specified by the data.
ACTION_CALL_BUTTON	The user pressed the "call" button to go to the dialer.
ACTION_SEND	Deliver some data to someone else.
ACTION_SENDTO	Send a message to someone specified by the data.
ACTION_ANSWER	Handle an incoming phone call.
ACTION_INSERT	Insert an empty item into the given container.
ACTION_DELETE	Delete the given data from its container.
ACTION_RUN	Run the data, whatever that means.
ACTION_SYNC	Perform a data synchronization.
ACTION_PICK_ACTIVITY	Pick an activity given an intent, returning the class selected.
ACTION_SEARCH	Perform a search.
ACTION_WEB_SEARCH	Perform a web search.

TABLE 1 – The most common actions for Android Intents