# Annexe 7 – Intents implicites

Le téléphone / AVD contient plusieurs applications de base qu'on peut utiliser à partir de nos propres applications ( internet, géolocalisations, liste de contacts, le téléphone lui-même )

Afin de performer ces opérations dans nos projets, il est préférable de recourir à ces applications plutôt que de les recoder nous-mêmes. Étant donné l’Action demandée, Android répondra avec la ou les applications permettant de répondre à cette Action. On utilise des Intents implicites pour ce faire par opposition à des intents explicites utilisés lorsqu’on dit nous-mêmes vers quelle autre activité on veut se rendre.

Les objets Intent implicites qu'on doit créer dans ces situations sont constitués de deux paramètres la plupart du temps :

* Une Action : une constante représentant ce qu'on veut faire
* Des données ( data ) : les données qu'on a besoin de fournir afin de réaliser l'action ci-haut

Voici quelques couples action / données utiles; il en existe un très grand nombre dans les fichiers de l'API:

|  |  |  |
| --- | --- | --- |
| Type de l'action | Constante action | Données |
| Ouvrir une page web | Intent.ACTION\_VIEW | Uri.parse("http://www.cvm.qc.ca") |
| Faire un appel téléphonique | Intent.ACTION\_DIAL | Uri.parse( "tel:+7897895555") |
| Faire afficher un endroit sur maps.google ( **l'AVD doit supporter Google APIS** ) | Intent.ACTION\_VIEW | Uri.parse("geo:0,0?q=ville, +province, +pays ") |
|  |  |  |
| Prendre une photo | MediaStore.ACTION\_IMAGE\_CAPTURE | Pas de données à passer, l’appareil photo est démarré et attend notre prise |
| Envoyer du texte via messagerie | Intent.ACTION\_SEND | Pas de données à passer; on passe le texte en extra dans le l’intent avec la clé EXTRA\_TEXT |
| Choisir dans sa liste de contacts | Intent.ACTION\_PICK | Pas de données à passer, on doit choisir un contact |
|  |  |  |

Si nécessaire, on peut également passer les données nécessaires à la réussite de la demande à l'aide de la méthode setData .

On peut avoir à déterminer le type de données qu'on s'attend à recevoir suite à une intention. On peut le faire à l'aide de la méthode setType.

Ex. : intention.setType ( ContactsContract.Contacts.CONTENT\_TYPE) retournera les résultats ( data ) sous un type MIME permettant de regrouper toutes les infos d'un contact.

Une image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect.

**Exercice : Quoi faire aujourd'hui ? ( utiliser l’émulateur API 35 )**

1) Utilisez le projet IntentsImplicites / Annexe 7 sur LÉA comprenant une activité composée de Buttons et d’un ImageView en bas ( pour la photo ).

2) Dans le fichier .java, gérez les événements onClick.

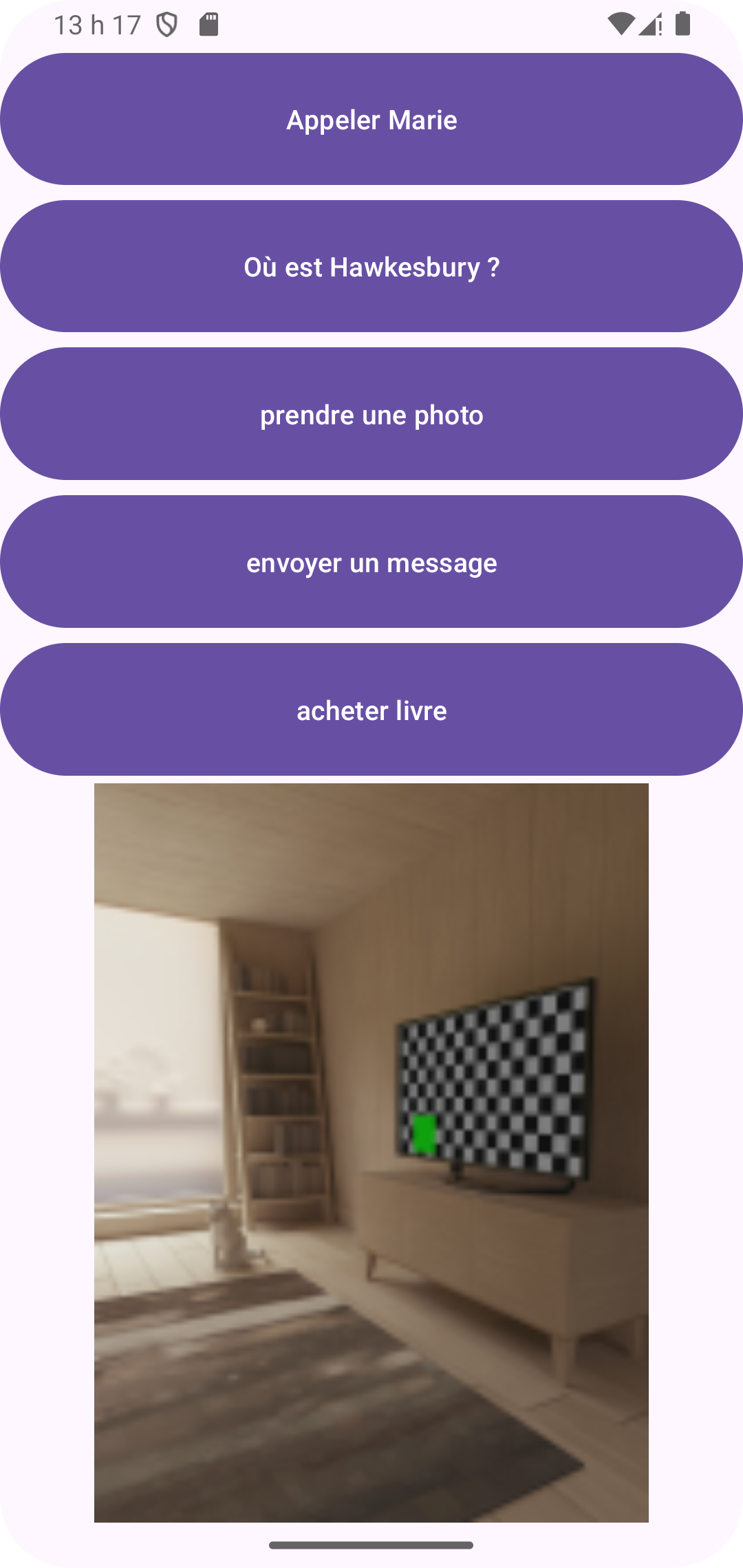
3) Faites en sorte qu’un clic sur le bouton « acheter livre » démarre le navigateur web à la page leslibraires.ca

4) Faites en sorte qu'un clic sur le 2ème bouton démarre un appel téléphonique à Marie

5) Faites en sorte qu'un clic sur le 3ème bouton fasse apparaître une carte de la ville de Hawkesbury (**Google APIs doit être installé sur le téléphone / émulateur** )

6) Envoyer la requête pour prendre une photo ( 4e bouton ) à l’aide d’un nouvel Intent en utilisant un Intent implicite et avec la technique du « boomerang / startActivityForResult ». On démarre donc l’appareil photo et on attend le résultat ( la photo sous forme de miniature, le retour du boomerang ) dans l’activité présente.

\*\*\* aucune permission explicite n’est nécessaire car on utilise l’appareil photo sans avoir recours à son api particulier ni à ses ressources \*\*\*



*J’ai pris une photo de ma « cuisine virtuelle » à partir de l’émulateur*

Est-ce que l’interface mise en œuvre pour le retour du boomerang / Intent *ActivityResultCallback* est une interface fonctionnelle ? Cliquez ou appuyez ici pour entrer du texte.

Si oui, vous pouvez simplifier la syntaxe utilisée avec une expression lambda.

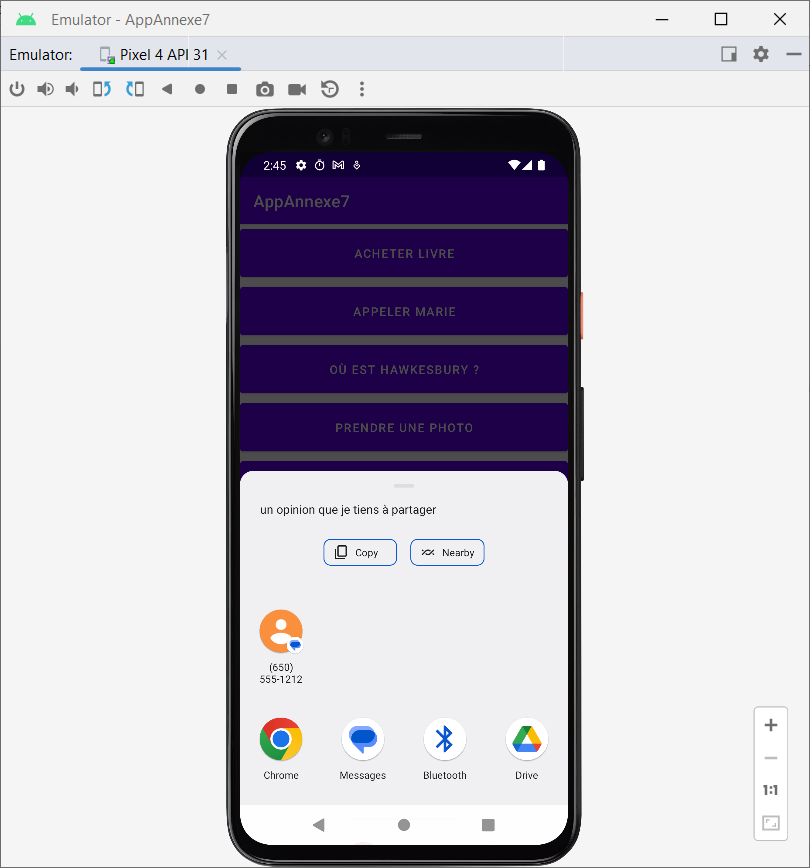
**EXTRA / BONUS**

**7. Envoyer un message vers la messagerie texte ou les réseaux sociaux**

Utiliser le dernier bouton sur l’interface qui permettra d’envoyer du texte à un contact

Lire le début du fichier d’aide d’Android (<https://developer.android.com/training/sharing/send> ) …on doit utiliser un Android Sharesheet afin de fournir les choix de courriel / messageries disponibles. On peut joindre le « sharesheet » à l’aide de la méthode statique createChooser

Jetez un coup d’œil dur les types de données MIME qu’on peut passer dans le tableau du document.



*Plus de choix sur mon téléphone que sur l’émulateur mais fonctionne ! Vous devrez ajouter des contacts sur votre émulateur*

Fonction apply : permet d’appeler des fonctions / propriétés directement sur l’objet sur lequel on appelle la fonction apply, sans avoir à répéter sans cesse la référence à l’objet

La fonction apply retourne l’objet sur lequel on l’a appelé ( l’intent dans ce cas-ci )