

ATELIER11

Classes : DSI3

Matière : Atelier Développement Mobile Natif

Nb pages : 3

Enseignants : S. Hadhri & M. Hadji & H. Souissi

L'objectif de ce TP est de programmer des applications android qui utilisent ML Kit.

Exercice1 (Scanner et ajouter)

Compléter le code de l'exercice1 pour obtenir une application qui fait le scan du code à barre d'un produit et l'ajoute au ListView.

- 1- Créer un nouveau projet "ScannerAjouter".
- 2- Importer le code et le layout du projet à partir de squelette.
- 3- Mettre le curseur sur l'erreur GmsBarcodeScanner, cliquer sur "More actions" puis sur "add dependency on com.google.android.gms:play-services-code-scanner and import".
- 4- Compléter le code de initScanner()

```
GmsBarcodeScannerOptions options = new GmsBarcodeScannerOptions.Builder()
```

```
.setBarcodeFormats(  
    Barcode.FORMAT_QR_CODE,  
    Barcode.FORMAT_EAN_13,  
    Barcode.FORMAT_ALL_FORMATS)  
.enableAutoZoom()  
.build();
```

```
scanner = GmsBarcodeScanning.getClient(this);
```

- 5- compléter le code de scan()

```
scanner.startScan()  
.addOnSuccessListener(  
    barcode -> {  
        String rawValue = barcode.getRawValue();  
        adpBc.add(rawValue);  
    })  
.addOnCanceledListener(  
    () -> {  
        Toast.makeText(this, "Canceled!", Toast.LENGTH_LONG).show();  
    })  
.addOnFailureListener(  
    e -> {  
        Toast.makeText(this, "Failure: "+e.getMessage(), Toast.LENGTH_SHORT).show();  
        e.printStackTrace();  
    });
```

- 6- Tester l'application avec différents code à barre

Exercice2 (Prix produit)

Compléter le code de l'exercice2 pour obtenir une application qui analyse les code à barre de l'image et affiche le prix du produit détecté.

- 1- Créer un nouveau projet "PrixProduit".
- 2- Importer le code, le layout et les images du projet à partir de squelette.
- 3- Ajouter le code de analyser()

```
Bitmap bitmap = ((BitmapDrawable) imgProduit.getDrawable()).getBitmap();
InputImage image = InputImage.fromBitmap(bitmap, 0);
BarcodeScanner scanner = BarcodeScanning.getClient();
Task<List<Barcode>> result = scanner.process(image)
    .addOnSuccessListener(new OnSuccessListener<List<Barcode>>() {
        @Override
        public void onSuccess(List<Barcode> barcodes) {
            // Task completed successfully
            // ...
            afficherPrix(barcodes);
        }
    })
    .addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            // Task failed with an exception
            // ...
            Toast.makeText(MainActivity.this, "Failure:" + e.getMessage(), Toast.LENGTH_SHORT).show();
        }
    });
```

- 4- Mettre le curseur sur les erreurs et ajouter les dépendances "ML kit" manquantes.

- 5- Ajouter le code de afficherPrix(...)

```
if(barcodes.size()>0){
    if(barcodes.size()==1){
        Barcode bar = barcodes.get(0);
        tvPrix.setText(getPrix(bar.getRawValue()));
    }else
        Toast.makeText(this, "Plusieurs code à barre dans l'image!", Toast.LENGTH_SHORT).show();
    }else
        Toast.makeText(this, "Aucun code à bar dans l'image!", Toast.LENGTH_SHORT).show();
```

- 6- Ajouter le code de getPrix(...)

```
String prix="Le prix ";
if(rawValue.equals("6191577600015"))
    prix+="de Eau Safia est 0.900 DT";
else if(rawValue.equals("123456789012"))
    prix+="de Eau M1 est 6.700 DT";
else if(rawValue.equals("3401312345624"))
    prix+="de Eau M2 est 4.300 DT";
else    prix+="Inconnu";
return prix;
```

Exercice3 (Traduction En Fr)

- 1- Créer un nouveau projet "TraductionEnFr".
- 2- Importer le code et le layout du projet à partir de squelette.
- 3- Mettre le curseur sur les erreurs et ajouter les dépendances "ML kit" manquantes.
- 4- Dans les imports de MainActivity, vérifier que Translator est importé de MLKit et non pas d'un autre package, si c'est le cas supprimer l'import et importer celui de MLKit.
- 5- Ajouter le code de initTraducteur()

```
TranslatorOptions options =  
    new TranslatorOptions.Builder()  
        .setSourceLanguage(TranslateLanguage.ENGLISH)  
        .setTargetLanguage(TranslateLanguage.FRENCH)  
        .build();  
enFrTranslator =  
    Translation.getClient(options);
```

```
DownloadConditions conditions = new DownloadConditions.Builder()  
    .requireWifi()  
    .build();  
enFrTranslator.downloadModelIfNeeded(conditions)  
    .addOnSuccessListener(  
        new OnSuccessListener() {  
            @Override  
            public void onSuccess(Object o) {  
                Toast.makeText(MainActivity.this, "Succès de téléchargement du modèle En-Fr",  
                    Toast.LENGTH_LONG).show();  
            }  
        })  
    .addOnFailureListener(  
        new OnFailureListener() {  
            @Override  
            public void onFailure(@NonNull Exception e) {  
                Toast.makeText(MainActivity.this, "Failure: "+e.getMessage(), Toast.LENGTH_LONG).show();  
            }  
        })  
    );
```

- 6- Ajouter le code de traduire()

```
if (!edPhraseEn.getText().toString().isEmpty()) {  
    enFrTranslator.translate(edPhraseEn.getText().toString())  
        .addOnSuccessListener(  
            new OnSuccessListener() {  
                @Override  
                public void onSuccess(Object o) {  
                    if(o!=null)  
                        edPhraseFr.setText(o.toString());  
                }  
            })  
        );
```

```

    }

    })

    .addOnFailureListener(
        new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                Toast.makeText(MainActivity.this, "Failure: "+e.getMessage(), Toast.LENGTH_LONG).show();
            }
        });
    }else
        edPhraseFr.setText("");

```

Exercice4 (Détecter les visages et dessiner)

- 1- Créer un nouveau projet "DetectionVisages".
- 2- Importer le code, le layout et le dossier raw (placer le dans res à coté de drawable) du projet à partir de squelette.
- 3- Dans les imports de MainActivity, vérifier que Face et FaceDetector sont importés de MLKit et non pas d'un autre package, si c'est le cas supprimer les imports et importer ceux de MLKit.
- 4- Ouvrir le code de activity_main.xml et remplacer le package "com.visage" par votre package.
- 5- Mettre le curseur sur les erreurs et ajouter les dépendances "ML kit" manquantes.
- 6- Tester l'application.