

# Devoir N° 3

## Android Camera

### 1- Android Camera

**Camera** est un périphérique qui vous permet de prendre des photos ou d'enregistrer une vidéo. Dans Android, il existe 2 façons de travailler avec la Caméra.

#### Méthode 1 :

Votre application peut appeler l'application par une **implicit Intent** pour exiger une certaine action avec la **camera**, demander à la caméra de prendre des photos ou d'enregistrer une vidéo, et puis les résultats sont retournés.

#### Méthode 2 :

Vous pouvez utiliser une **API** pour faire fonctionner directement la **caméra**, je vous recommande donc l'utilisation de l'API camera2.

### 2- Vue d'ensemble

On utilise **Intent implicit** (implicit intent) pour appeler l'application **caméra** ; vous pouvez demander à la **caméra** de prendre une photo ou d'enregistrer une vidéo.

```
// Create an implicit intent, for image capture.
Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);

int REQUEST_ID_IMAGE_CAPTURE = 100;

// Start camera and wait for the results.
this.startActivityForResult(intent, REQUEST_ID_IMAGE_CAPTURE);
```

On a différents types d'Intent pour **Camera** :

Des types d'Intent	Description
<b>ACTION_IMAGE_CAPTURE_SECURE</b>	Il renvoie l' <b>Image</b> capturée à partir de la <b>Camera</b> , lorsque l'appareil photo est sécurisé
<b>ACTION_VIDEO_CAPTURE</b>	Il appelle l'application vidéo existante dans <b>Android</b> pour capturer une vidéo à partir de la <b>Camera</b> .
<b>EXTRA_SCREEN_ORIENTATION</b>	Il sert à régler l'orientation de l'écran en " <b>vertical</b> " ou en " <b>landscape</b> " (paysage)
<b>EXTRA_FULL_SCREEN</b>	Il sert à contrôler l'interface utilisateur du <b>ViewImage</b>
<b>INTENT_ACTION_VIDEO_CAMERA</b>	Cette intention permet de lancer la <b>Camera</b> en mode vidéo
<b>EXTRA_SIZE_LIMIT</b>	Il est utilisé pour spécifier la limite de taille de la taille de capture vidéo ou image.

Dans le cas où vous souhaitez enregistrer des photos ou des vidéos enregistrées sur l'appareil, vous devez configurer les autorisations pour lire et écrire des données sur le périphérique. Configurez sur **AndroidManifest.xml** :

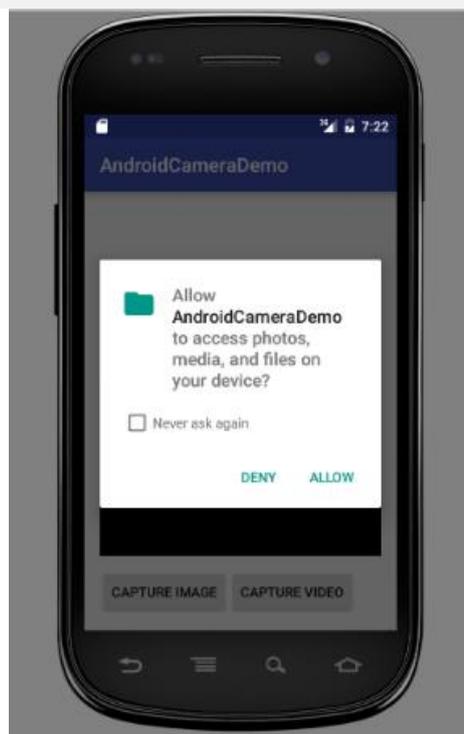
```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

Avec **Android Level >= 23**, vous devez utiliser le code pour demander à l'utilisateur l'autorisation de lire et d'écrire des données sur le périphérique

```
// With Android Level >= 23, you have to ask the user
// for permission to read/write data on the device.
if (android.os.Build.VERSION.SDK_INT >= 23) {

    // Check if we have read/write permission
    // Kiểm tra quyền đọc/ghi dữ liệu vào thiết bị lưu trữ ngoài.
    int readPermission = ActivityCompat.checkSelfPermission(this,
        Manifest.permission.READ_EXTERNAL_STORAGE);
    int writePermission = ActivityCompat.checkSelfPermission(this,
        Manifest.permission.WRITE_EXTERNAL_STORAGE);

    if (writePermission != PackageManager.PERMISSION_GRANTED ||
        readPermission != PackageManager.PERMISSION_GRANTED) {
        // If don't have permission so prompt the user.
        this.requestPermissions(
            new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE,
                Manifest.permission.READ_EXTERNAL_STORAGE},
            REQUEST_ID_READ_WRITE_PERMISSION
        );
    }
}
```



```

// When you have the request results
@Override
public void onRequestPermissionsResult(int requestCode,
    String permissions[], int[] grantResults) {

    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    //
    switch (requestCode) {
        case REQUEST_ID_READ_WRITE_PERMISSION: {

            // Note: If request is cancelled, the result arrays are empty.
            // Permissions granted (read/write).
            if (grantResults.length > 1
                && grantResults[0] == PackageManager.PERMISSION_GRANTED
                && grantResults[1] == PackageManager.PERMISSION_GRANTED) {

                Toast.makeText(this, "Permission granted!", Toast.LENGTH_LONG).show();

                this.captureVideo();

            }
            // Cancelled or denied.
            else {
                Toast.makeText(this, "Permission denied!", Toast.LENGTH_LONG).show();
            }
            break;
        }
    }
}

```

### 3- Travail à faire

Créez un projet nommé **AndroidCameraDemo** qui affiche deux fragments, une **imageView** et une **videoView** avec deux boutons : Capturer Image et Capturer Vidéo.

En suivant ce qui a été introduit plus haut, concevoir le projet AndroidCameraDemo et lancer-le sur l'émulateur.

### 3- Ce que vous devez rendre

1. Les fichiers XML,
2. Les fichiers Java,
3. Un enregistrement vidéo qui montre les différentes étapes de création de cette application et son exécution.