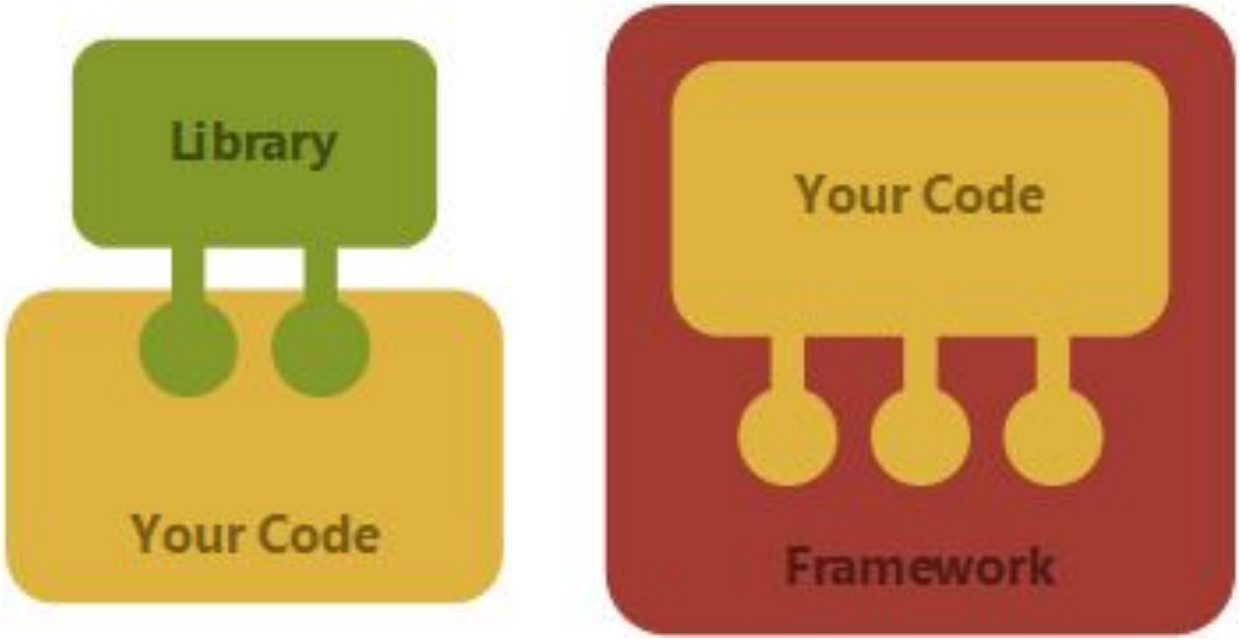
Android


Cupcake
Donut
Eclair
Froyo
Gingerbread
Honeycomb
ICE Cream-Sandwitch
Jelly Bean
Kitkat
lollipop
Lollipop
Marshmallow
Nougat

Framework vs Library



Coordination avec le cycle de vie d'une activité

onAttach() : appelée quand le fragment est associé à une Activité

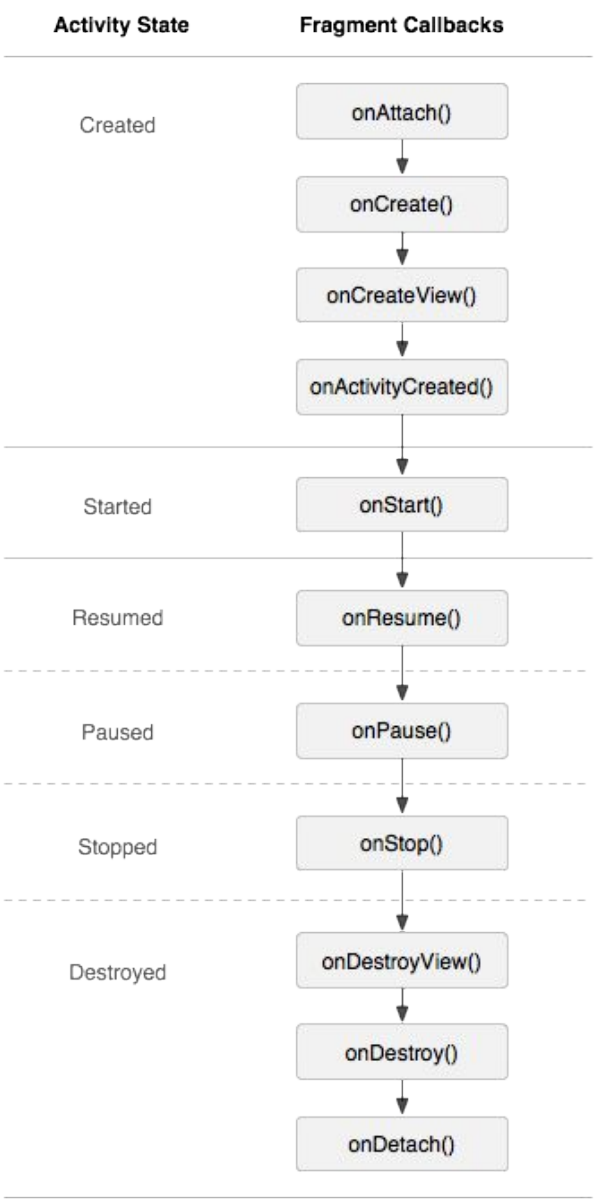
onCreateView() : appelée pour créer la vue associée au fragment

onActivityCreated() : lorsque l'activité est créée

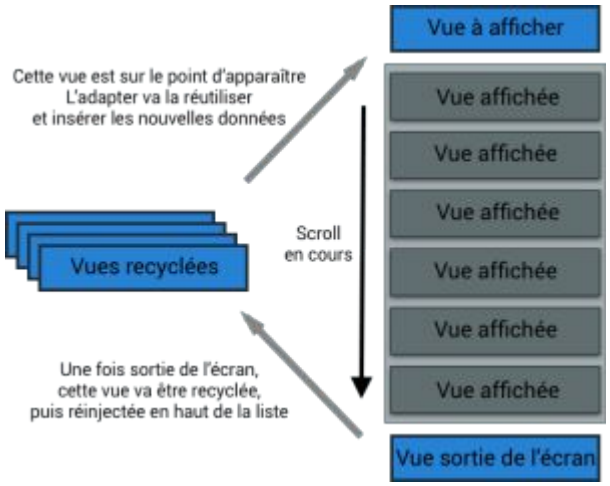
onDestroyView() : lorsque la vue associée au fragment est tuée

onDetach() : lorsque le fragment est dissocié de l'Activité

Ce n'est que lorsqu'une activité est dans l'état *resumed*, qu'il est possible d'ajouter et effacer des fragments à l'activité …



Fonctionnement de ListView : recyclage et fluidité

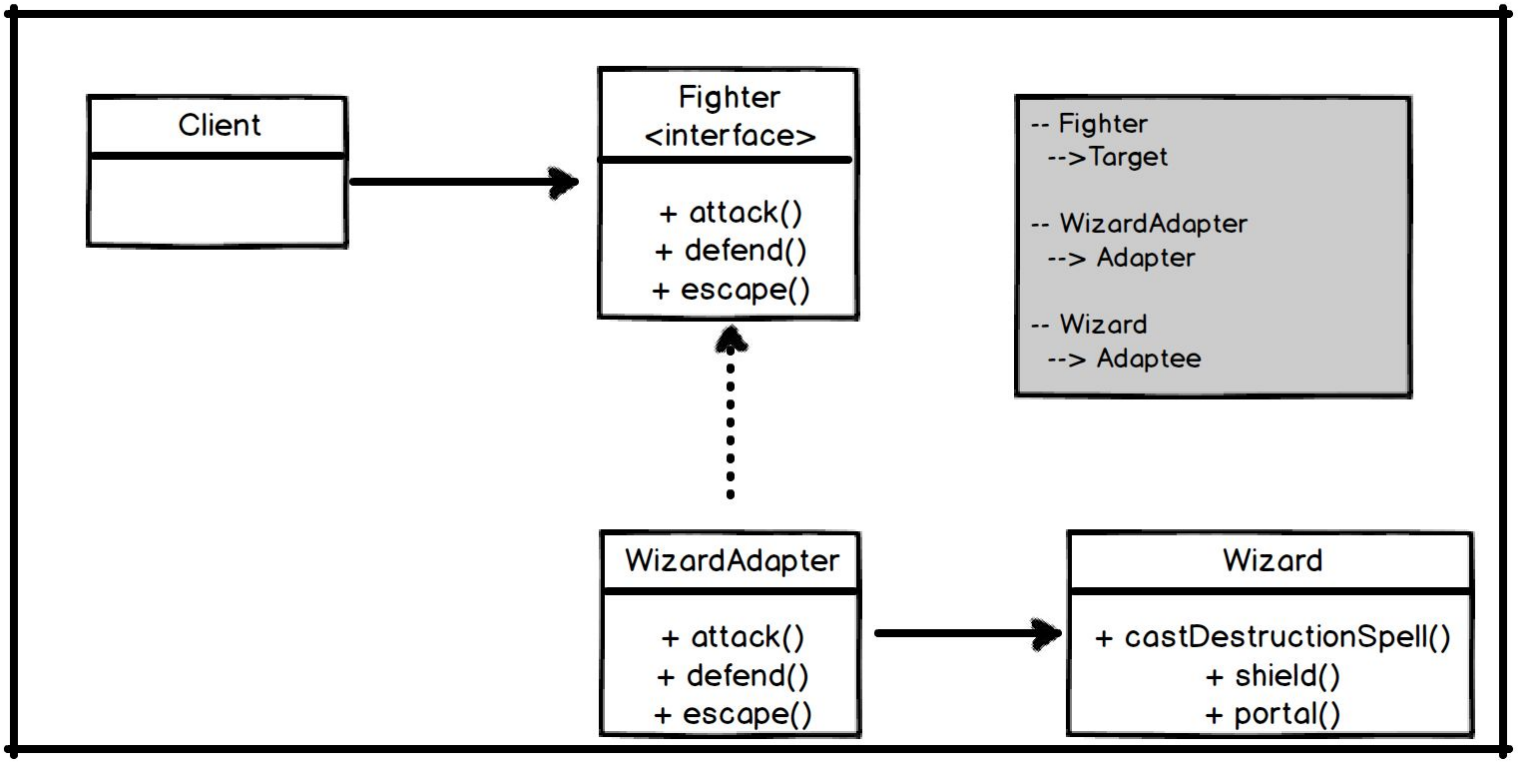


Pour réduire la consommation en mémoire la ListView stocke seulement les vues qu'elle a la capacité d'afficher,
Lorsqu'une vue sort de l'écran (scroll) elle est réutilisée pour la nouvelle vue à apparaître.

Afin d'éviter d'appeler les méthodes findViewById à chaque réutilisation des vues, Android a rajouté un concept, le ViewHolder (gardien/protecteur de vue) : mini controleur, associé a chaque cellule, et qui va stocker les références vers les sous vues.
C'est une propriété de la vue (dans l'attribut tag) : une vue n'a qu'un seul ViewHolder, et inversement.

http://tutos-android-france.com/listview-afficher-une-liste-delements/

Adapter


Client
Fighter
<interface>
+ attack()
+ defend()
+ escape()
-- Fighter
-->Target
-- WizardAdapter
--> Adapter
-- Wizard
--> Adaptee
WizardAdapter
+ attack()
+ defend()
+ escape()
Wizard
+ castDestructionSpell()
+ shield()
+ portal()

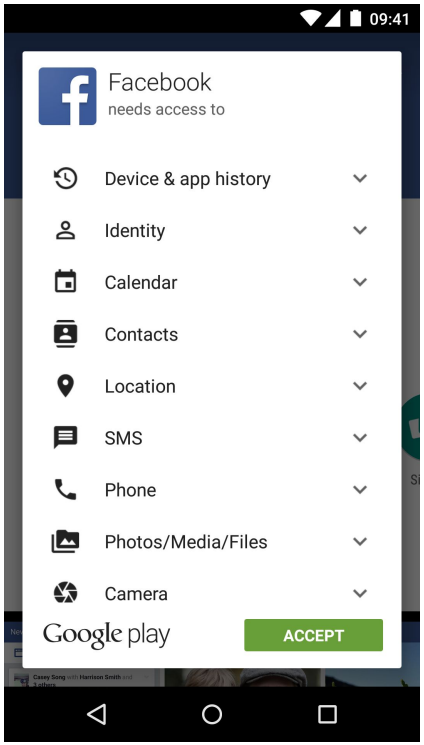
Fragment vs Vue - ATTENTION

La cellule d’une ListView, ce n’est pas un fragment

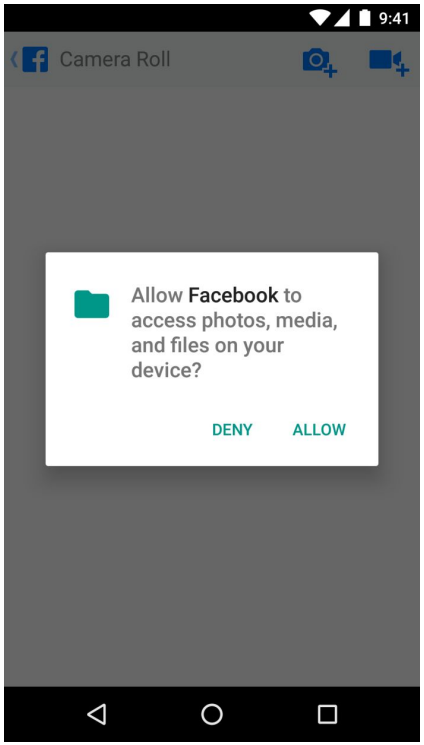
C’est pas parce que ça utilise tout deux des Adapters que c’est le même concept

Une cellule est une vue mais vous ne gérez pas son cycle de vie => ce n’est pas un fragment

Permissions



A l'installation



A runtime (Android 6.0)

https://theelfismike.wordpress.com/2015/10/21/android-6-0-runtime-permissions-facebook-case-study/

Type de permissions

Les basiques :

- ACCESS_NETWORK_STATE
- ACCESS_WIFI_STATE
- BLUETOOTH
- CHANGE_NETWORK_STATE
- CHANGE_WIFI_STATE
- EXPAND_STATUS_BAR
- INSTALL_SHORTCUT
- INTERNET
- KILL_BACKGROUND_PROCESSES
- MANAGE_OWN_CALLS
- MODIFY_AUDIO_SETTINGS
- NFC
- SET_ALARM
- USE_FINGERPRINT
- VIBRATE
- WAKE_LOCK

Groupes de permissions pour Calendrier, Appel, SMS, ...

https://developer.android.com/guide/topics/permissions/index.html

Demande et vérification de permissions

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
        package="com.example.snazzyapp">
    <uses-permission android:name="android.permission.SEND_SMS"/>
    <!-- other permissions go here -->
    <application ...>
        ...
    </application>
</manifest>
```

```
if (ContextCompat.checkSelfPermission(thisActivity, Manifest.permission.WRITE_CALENDAR)
        != PackageManager.PERMISSION_GRANTED) {
    // Permission is not granted
}
```

Location API - Usages

Tagguer un message avec la position géographique (Instagram, Twitter, …)

Guider pour un trajet (Maps, Waze, …)

Prédire l'heure des départs pour un rdv (Calendar, ...)

Donner des informations contextualisés (Actualités et météo, …)

Location API - Dernière position

ACCESS_COARSE_LOCATION

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    // ...
    mFusedLocationClient = LocationServices.getFusedLocationProviderClient(this);
    mFusedLocationClient.getLastLocation()
        .addOnSuccessListener(this, new OnSuccessListener<Location>() {
            @Override
            public void onSuccess(Location location) {
                // Got last known location. In some rare situations this can be
null.
                if (location != null) {
                    // Logic to handle location object
                }
            }
        });
}
```

https://developer.android.com/training/location/index.html

Location API - Notification de maj position

ACCESS_FINE_LOCATION

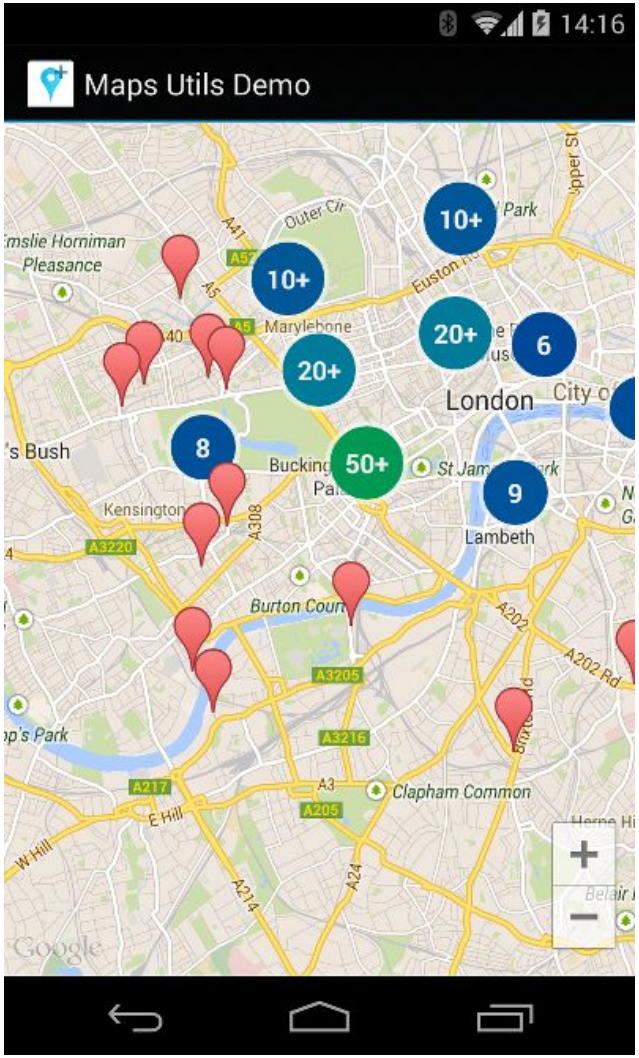
```
@Override
protected void onResume() {
    super.onResume();
    if (mRequestingLocationUpdates) {
        startLocationUpdates();
    }
}
private void startLocationUpdates() {
    mFusedLocationClient
    requestLocationUpdates(mLocationRequest,
            mLocationCallback,
            null /* Looper */);
}
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    mLocationCallback = new LocationCallback() {
        @Override
        public void onLocationResult(LocationResult
locationResult) {
            if (locationResult == null) {
                return;
            }
            for (Location location :
locationResult.getLocations()) {
                // Update UI with location data
                // ...
            }
        };
    };
}
```

https://developer.android.com/training/location/index.html

Location API - Surcouche

```
public class MapsActivity extends FragmentActivity
implements OnMapReadyCallback {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        SupportMapFragment mapFragment =
        (SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
                LatLng sydney = new LatLng(-34, 151);
        mMap.addMarker(new MarkerOptions().position(sydney)
          .title("Marker in Sydney"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
    }
}
```



https://developers.google.com/maps/documentation/android-api/start?hl=fr

Camera API - Usages

Scanner une information de type Qrcode (Fidme, Yuka, Envibus, …)

Envoyer une photo avec un message (Instagram, Twitter, …)

Immersion dans une réalité augmentée (Tango, Pokémon Go, …)

Camera API

```
<uses-permission android:name="android.permission.CAMERA" />
<uses-feature android:name="android.hardware.camera" />
```

```
public static Camera getCameraInstance(){
    Camera c = null;
    try {
        c = Camera.open(); // attempt to get a Camera instance
    }
    catch (Exception e){
        // Camera is not available (in use or does not exist)
    }
    return c; // returns null if camera is unavailable
}
```

https://developer.android.com/guide/topics/media/camera.html

Camera API - Camera Preview

```
public class CameraPreview extends SurfaceView implements SurfaceHolder.Callback {

    private SurfaceHolder mHolder;

    private Camera mCamera;

    public CameraPreview(Context context, Camera camera) {

        super(context);

        mCamera = camera;

        // Install a SurfaceHolder.Callback so we get notified when the underlying surface is created and destroyed.

        mHolder = getHolder(); mHolder.addCallback(this);

        // deprecated setting, but required on Android versions prior to 3.0

        mHolder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);

    }

    public void surfaceCreated(SurfaceHolder holder) {

        // The Surface has been created, now tell the camera where to draw the preview.

        mCamera.setPreviewDisplay(holder);

        mCamera.startPreview();

            }

}
```

Camera API - Take Picture

```
Button captureButton = (Button) findViewById(id.button_capture);

captureButton.setOnClickListener(

    new View.OnClickListener() {

        @Override

        public void onClick(View v) {

            // get an image from the camera

            mCamera.takePicture(null, null, mPicture);

        }

    }

);
```

https://developer.android.com/guide/topics/media/camera.html

Camera API - Surcouche

```
Intent intent = new Intent(getApplicationContext(),CaptureActivity.class);
intent.setAction("com.google.zxing.client.android.SCAN");
intent.putExtra("SAVE_HISTORY", false); startActivityForResult(intent, 0);
```

Retrieving the results from the scan in onActivityResult ():

```
if (requestCode == 0) {
    if (resultCode == RESULT_OK) {
        String contents = data.getStringExtra("SCAN_RESULT");
        Log.d(TAG, "contents: " + contents);
    } else if (resultCode == RESULT_CANCELED) {
        // Handle cancel
        Log.d(TAG, "RESULT_CANCELED");
    }
}
```

https://stackoverflow.com/questions/29159104/how-to-integrate-zxing-barcode-scanner-without-installing-the-actual-zxing-app

Wearable App - Usages

Guidage piéton (Maps, GoHere, …)

Horloge spécialisée (Watchfaces, …)

eSanté (Google Fit, Samsung Health, …)

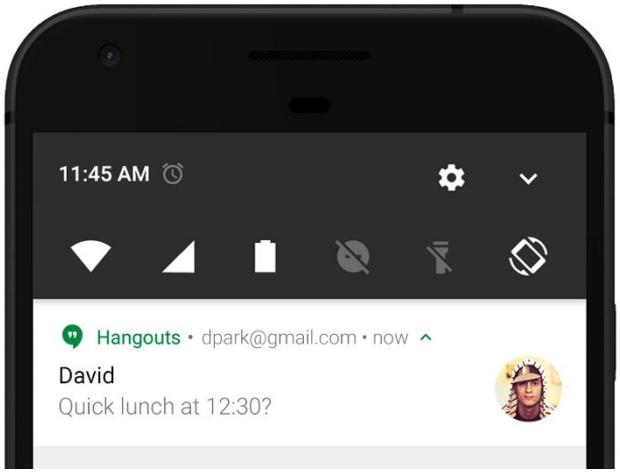
Coach sportif (Runtastic, Smart Caddie, …)

Wearable App - Montre

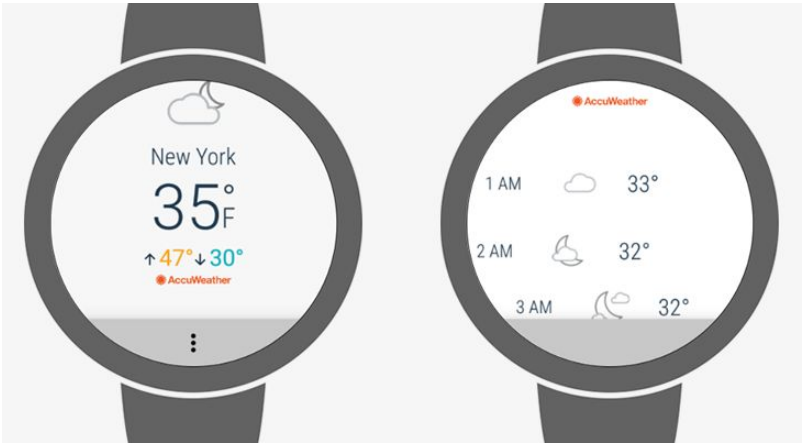
- Ecran rond ou carré
- Bluetooth
- Wifi, 4G
- GPS
- NFC
- Capteur cardiaque
- Haut-parleur/micro

Wearable App - Types

- Notifications



- Standalone



https://developer.android.com/training/wearables/apps/index.html

Autres API

- Capteurs : https://developer.android.com/guide/topics/sensors/index.html
  - Luminosité
  - Accelerомètre
  - Gravité
  - Température
  - …

- Connectivité : https://developer.android.com/guide/topics/connectivity/index.html
  - Bluetooth et BLE
  - Wifi
  - NFC
  - …