



## Développement Mobile

# WorkShop 2 : Activity & Intents


Année universitaire 2019/2020



# Ce que vous devriez déjà savoir




- Vous devriez être familier avec:
  - Créez et exécutez des applications dans Android Studio.
  - Utilisez l'éditeur de Layout pour créer une Layout avec un ConstraintLayout, LinearLayout et RelativeLayout.
  - Editez le code XML de la mise en page.
  - Ajouter la fonction onClick à des Buttons.



# Ce que vous allez apprendre

- ✓ Comment créer un nouvelle activité dans Android Studio.
- ✓ Comment définir les activités parent et enfant pour la navigation vers le haut.
- ✓ Comment commencer un activité avec un `INTENT` explicite.
- ✓ Comment passer des données entre chacun activité avec un `INTENT` explicite.




# Introduction

- Une activité représente un seul écran dans votre application avec lequel votre utilisateur peut effectuer une tâche unique, telle que prendre une photo, envoyer un courrier électronique ou afficher une carte.
- Elle est généralement présentée à l'utilisateur sous la forme d'une fenêtre plein écran.



# Introduction

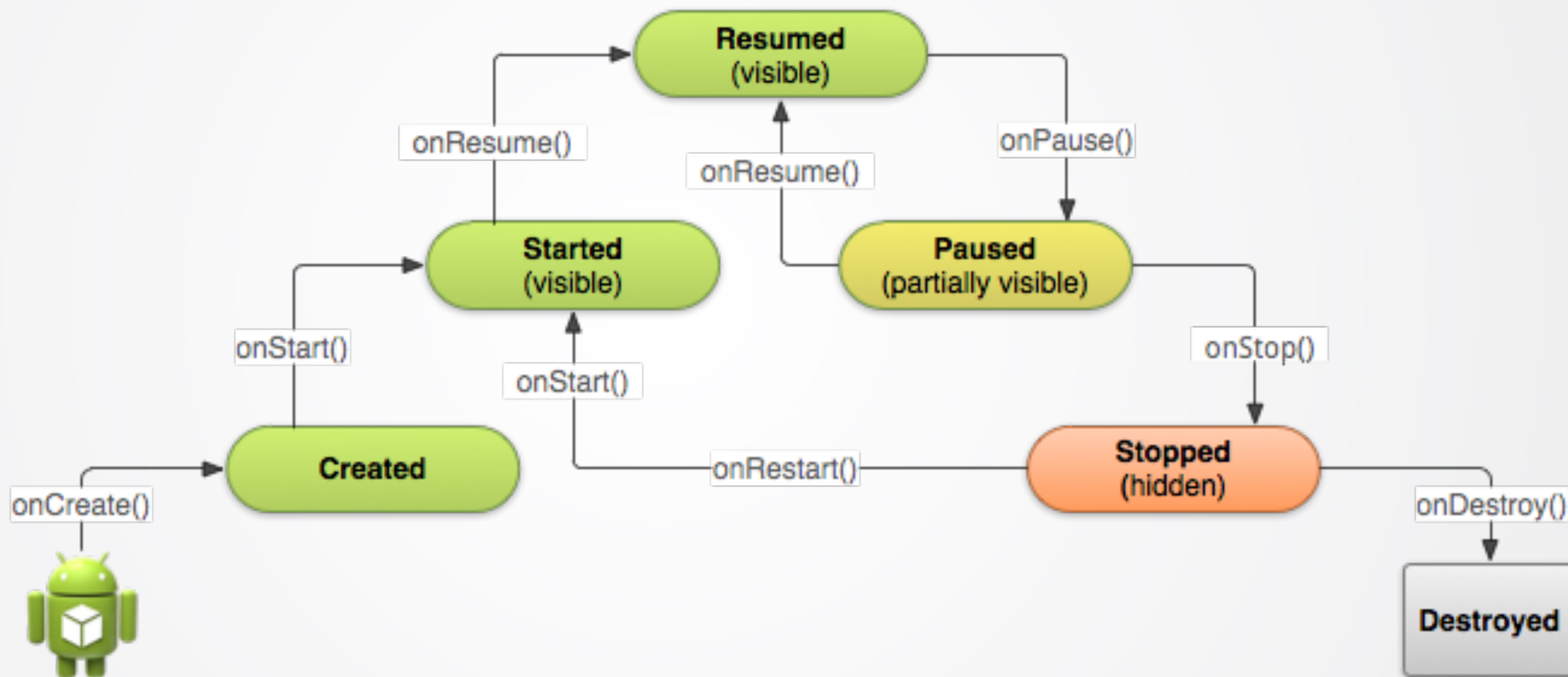
- Une application est généralement composée de plusieurs écrans liés de manière approximative. Chaque écran est une activité.
- En règle générale, ( `MainActivity.java`) dans une application est spécifiée comme activité "principale", qui est présentée à l'utilisateur lors du lancement de l'application. L'activité principale peut alors démarrer d'autres activités pour effectuer différentes actions.



# Introduction

- Chaque fois qu'une nouvelle activité démarre, l'activité précédente est arrêtée, mais le système conserve l'activité dans une pile ("back stack"). Quand une nouvelle activité commence, cette nouvelle activité est poussée dans la pile arrière et prend le focus de l'utilisateur.
- La pile arrière suit la logique de pile de base "last in, first out". Lorsque l'utilisateur a terminé l'activité en cours et appuie sur le bouton précédent, cette activité est détruite et l'activité précédente reprend.

# ► Cycle de vie





# Introduction

- Une activité est démarrée ou activée avec un **INTENT**.
- Un **INTENT** est un message asynchrone que vous pouvez utiliser dans votre activité pour demander une action à une autre activité ou à un autre composant d'application. Vous utilisez **INTENT** pour démarrer une activité à partir d'une autre activité et de transmettre des données entre les activités.





# Introduction



- Un **INTENT** peut être explicite ou implicite :
  - Une intention explicite est une intention dans laquelle vous connaissez la cible de cette intention. C'est-à-dire que vous connaissez déjà le nom de classe complet de cette activité spécifique.
  - Une intention implicite est une intention dans laquelle vous n'avez pas le nom du composant cible, mais une action générale à effectuer.



# ▶ Ce que vous allez faire

- Créer une nouvelle application Android avec une activité principale et une activité seconde.
- Passez des données de l'activité main à la seconde en utilisant un **INTENT** , et affichez ces données dans la seconde activité .
- Envoyez des données de la seconde activité vers la principale, en utilisant également un **INTENT**.

# ▶ Aperçu de l'application à faire



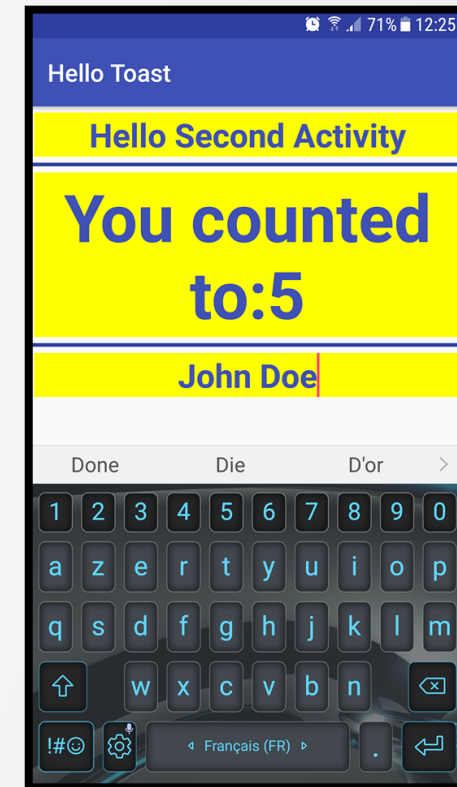
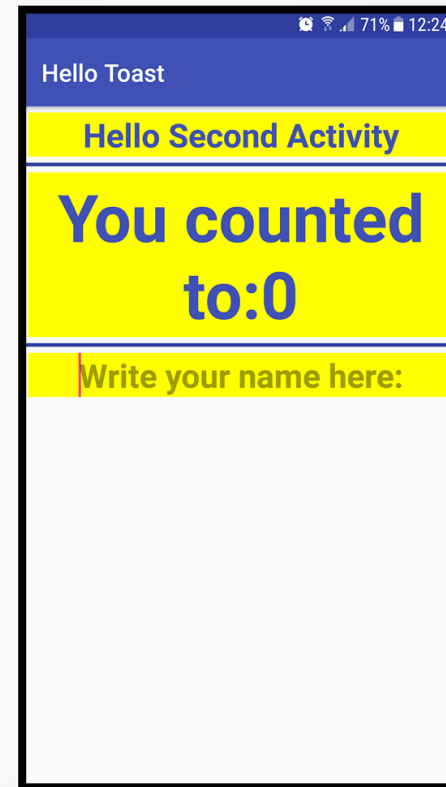
- L'application **Hello Toast** est composée de deux activités. Notre activité main doit contenir deux **TextView** et quatre **Boutons**.
- Les boutons **COUNT** et **TOAST** seront les mêmes que dans le workshop précédent.
- Le bouton **SEND** permet d'envoyer à la deuxième activité le nombre du compteur.
- Le bouton **S&R** fait la même action que le bouton **SEND** mais on doit attendre en retour un texte qui va être remplacé dans le 1<sup>er</sup> **TextView**.



# ▶ Aperçu de l'application à faire



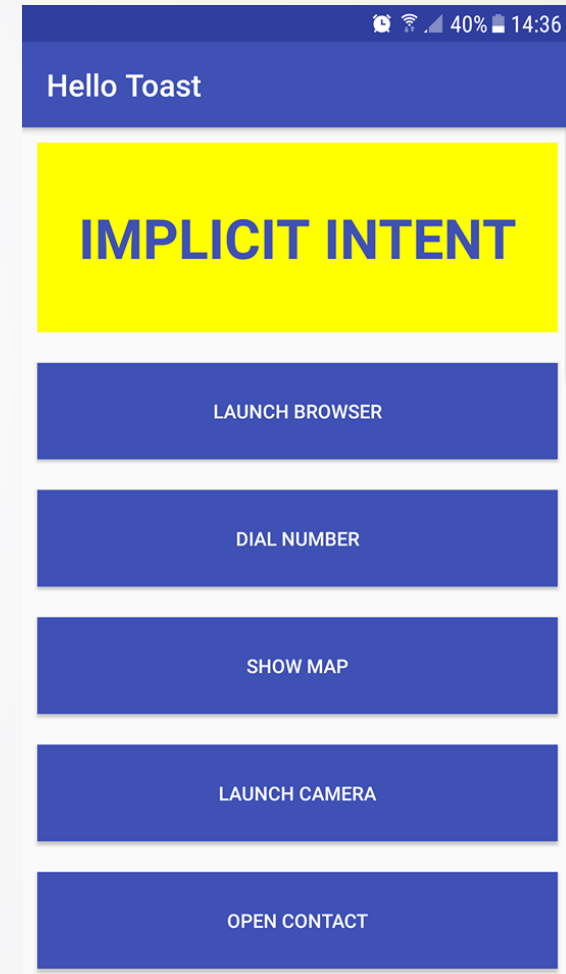
- La deuxième activité contient deux **TextView**, deux **Séparateur** et un **EditText**
- Le 1<sup>er</sup> texte contient « Hello Second Activity »
- Le 2<sup>eme</sup> texte contient « You counted to: » avec une concaténation du compteur de la 1<sup>ere</sup> activité
- Le **EditText** est utilisable si l'utilisateur a cliquer sur le bouton **S&R**, il doit donc entrer un nom qui sera afficher dans la **mainActivity** quand on clique sur le bouton retour du Smart Phone.




# ▶ Aperçu de l'application à faire



- Intention Implicite
  - Créer une troisième activité
  - Changer ensuite le code du bouton TOAST pour afficher cette activité
  - Créer les Intent implicite pour chaque bouton





# ▶ Coding challenge

- Surcharger les méthodes et ajouter un Log.d()

- onStart()
- onResume()
- onPause()
- onStop()
- onDestroy()

Exemple :

```
@Override  
public void onStart(){  
    super.onStart();  
    Log.d(LOG_TAG, "onStart");  
}
```

- Ensuite observez le journal pendant l'exécution