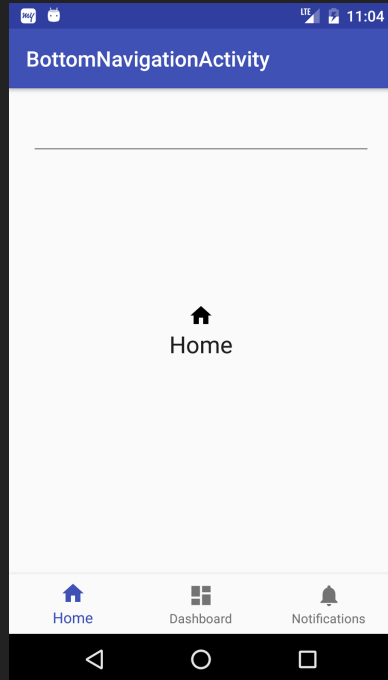


Activity



- Représente en gros une "page"
- Généralement prends tout l'écran
- Gère l'interaction entre la vue et l'utilisateur
- Peut démarrer d'autres Activity dans la même app ou d'autres
- Obéit à un "Lifecycle"
- Les Activity peuvent être hiérarchisée dans le manifest (pour la navigation)
- Une Activity "inflate" un layout XML dans `onCreate`
- 🍏 `UIViewController` (ex: `viewDidLoad`)

Inflating Layout in Activity

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
        // access resources^ ^layouts    ^layout file name  
    }  
}
```

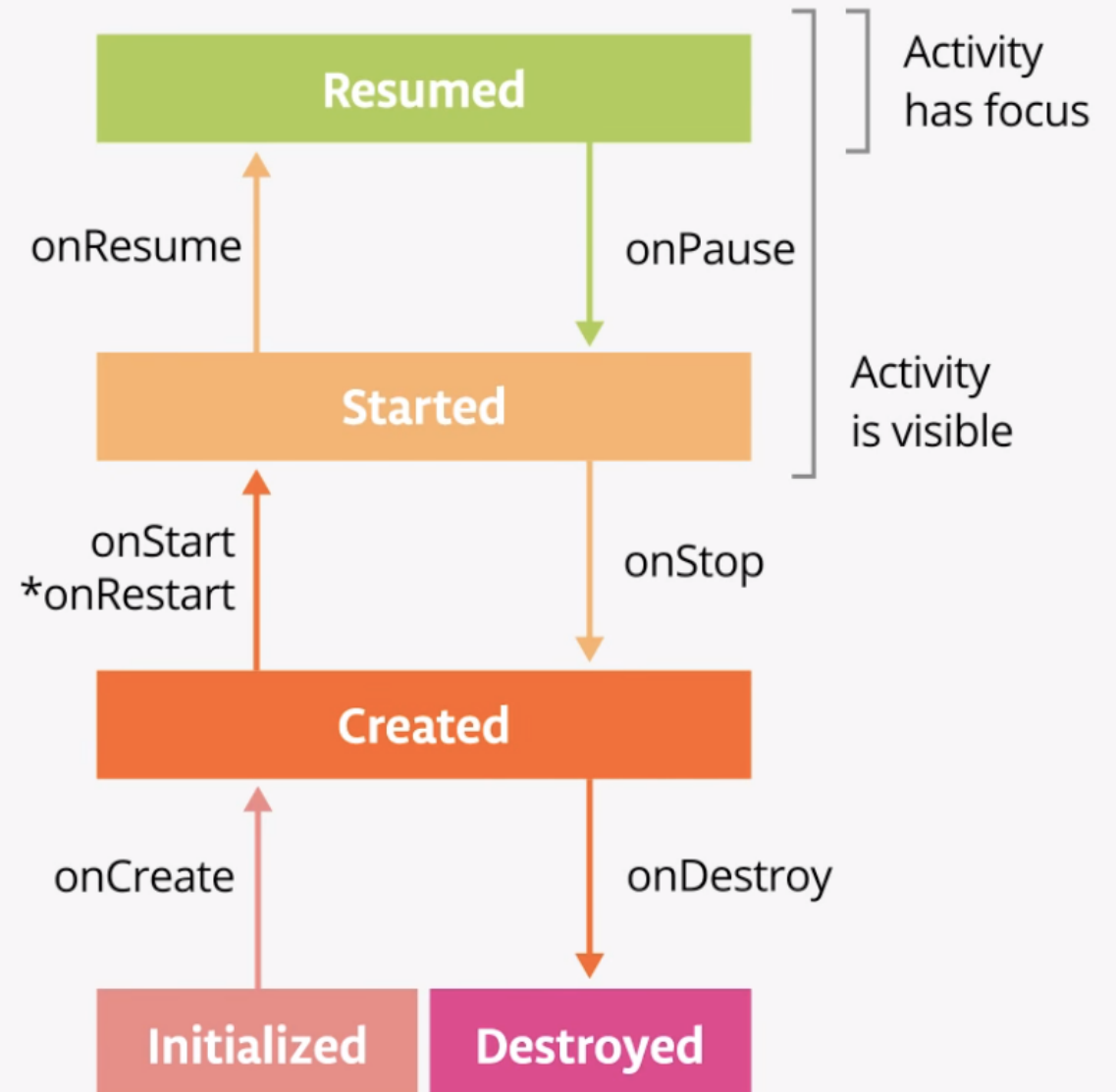
Declare main activity in manifest

```
// MainActivity needs to include intent-filter to start from launcher
<activity
    android:name=".MainActivity"
    android:label="@string/app_name"
    android:theme="@style/AppTheme.NoActionBar">
    <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
</activity>
```

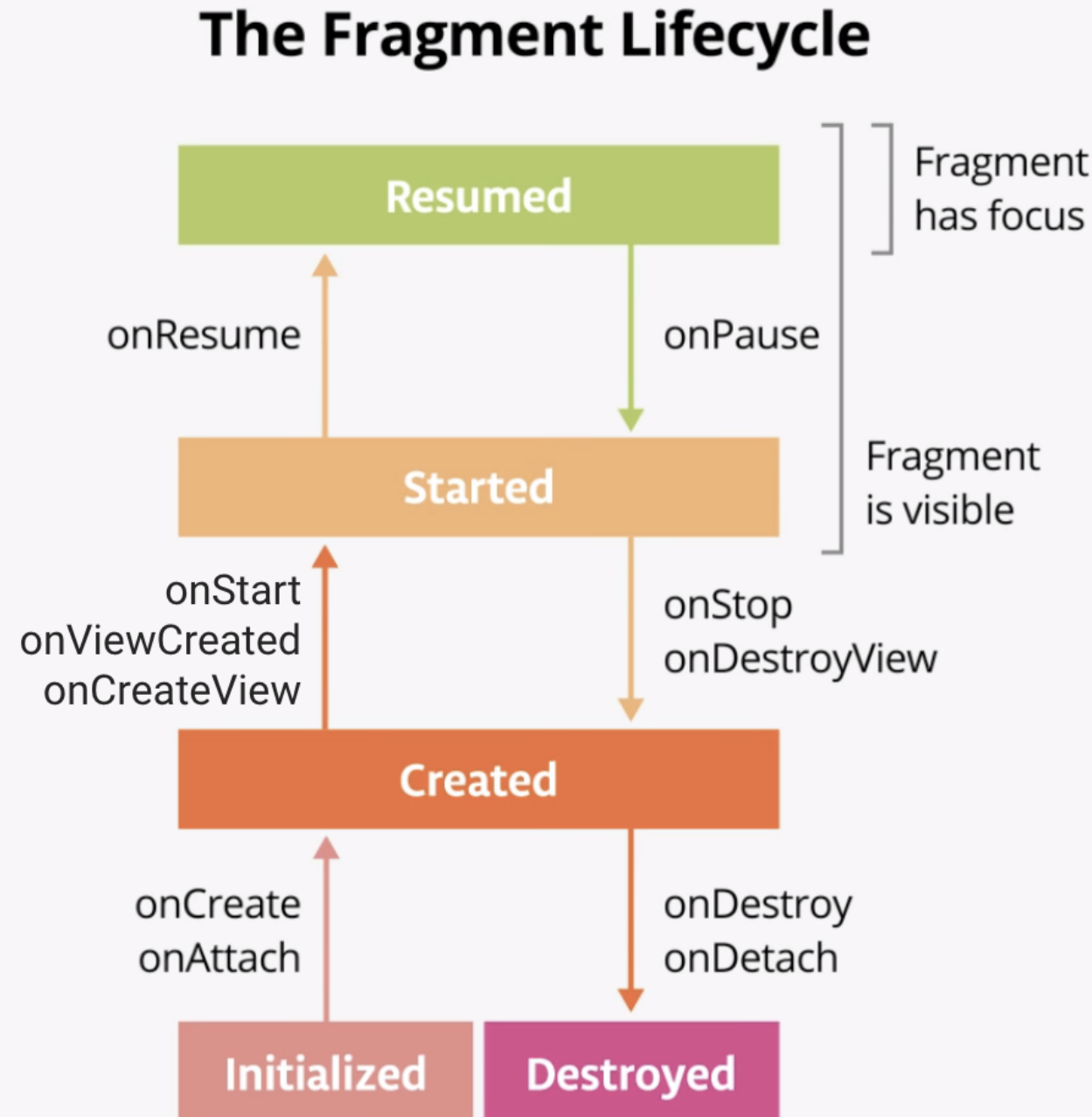
Lifecycle

Activity lifecycle

The Activity Lifecycle



Fragment lifecycle



Configuration Changes

- Rotation
- Changement de langage
- Mode multi-fenêtre

Android garde seulement :

- L'intent éventuellement utilisé
- L'état des vues ayant un ID (ex: RecyclerView scroll position)



InstanceState

```
override fun onSaveInstanceState(outState: Bundle) {  
    super.onSaveInstanceState(outState)  
    outState.putString("key", count_text_view.text.toString())  
}  
  
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_main)  
    savedInstanceState?.getString("key").let { count ->  
        count_text_view.setText(count)  
    }  
}
```

- Préserve les données de l'Activity
- Configuré manuellement
- Perdu si on quitte l'Activity
- Plus persistant: DB, Web, SharedPreferences, DataStore

iOS

- rotation: `viewWillTransition()`
- link: set VC in StoryBoard / Nib / `loadView()`
- state restoration: `restorationIdentifiers` on VC and Views

