

# Matériel

Chantal Keller

13 novembre 2015

# Plan

- 1 Matériel
- 2 Utilisation d'une application existante
- 3 À la main
- 4 À vous

## Présence du périphérique

L'utilisateur n'a pas toujours le périphérique requis :

- est-il vraiment nécessaire ?
- sinon, faire une application pouvant fonctionner lorsque le périphérique n'est pas présent (ce qu'on peut tester)

## Présence du périphérique

L'utilisateur n'a pas toujours le périphérique requis :

- est-il vraiment nécessaire ?
- sinon, faire une application pouvant fonctionner lorsque le périphérique n'est pas présent (ce qu'on peut tester)

Permissions :

- `<uses-permission android:name="android.permission.CAMERA" />`
- choisir les permissions minimales

Fonctionnalités :

- `<uses-feature android:name="android.hardware.camera.autofocus" />`
- pas toujours présentes

# Plan

- 1 Matériel
- 2 Utilisation d'une application existante
- 3 À la main
- 4 À vous

# Principe

Méthode usuelle pour lancer une activité :

- intention portant l'action que l'on souhaite effectuer
- utilisée pour démarrer une nouvelle activité
- Android va rechercher une activité (au sein d'une application) susceptible de faire cette action et la lancer
- on peut récupérer un résultat

# Avantages et inconvénients

## Avantage :

- très simple

## Inconvénients :

- il faut qu'il y ait déjà une application faisant ce qu'on veut
- l'appareil de l'utilisateur n'a pas forcément l'application à laquelle on pensait
- dépendance

# Plan

- 1 Matériel
- 2 Utilisation d'une application existante
- 3 À la main
- 4 À vous

# Principe

## Manipulation directe du périphérique :

- API Android
- fortement dépendante du périphérique (pas de code dans ces slides, voir TD9)
- encore et toujours de la programmation objet événementielle

# Une ressource partagée

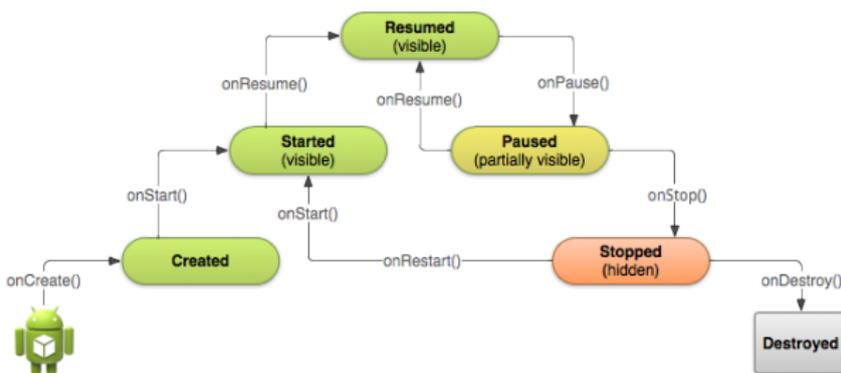
## Périphérique :

- partagé entre toutes les applications
- plusieurs applications ne peuvent utiliser le même périphérique en même temps
- début de l'utilisation : peut-on acquérir la ressource ?
- fin de l'utilisation : **relâcher la ressource**

# Durée d'accès à une ressource

La plus courte possible :

- acquisition le plus tard possible : dans `onResume`
- relâchement le plus tôt possible : dans `onPause`



© Android

# Conclusion

## Bonnes pratiques :

- bien choisir les périphériques dont on a besoin
- proposer une solution de remplacement si l'utilisateur ne les a pas
- choisir la méthodologie adaptée à notre application : utiliser l'existant ou recoder entièrement
- accéder à la ressource le moins longtemps possible

# Plan

- 1 Matériel
- 2 Utilisation d'une application existante
- 3 À la main
- 4 À vous

# TD9

## Application d'appareil photo :

- faisant appel à l'application existante
- entièrement à la main

↔ vous êtes encouragés à en faire le maximum possible et vous pouvez me le rendre si vous souhaitez un retour