


**Android 2.2, Froyo**

Voice Typing lets you input text, and Voice Actions let you control your phone, just by speaking.



**CHAPITRE 5**  
**LA CLASSE « FRAGMENT »** - VERSION 3.0, TABLETS • Mr. MEGHAZI

2019-2020

**Cours pour les Master II - GL**

1

## Tablets UIs

- Les tablets possèdent des écrans plus larges que ceux des téléphones.
- Peuvent prendre en charge plusieurs volets de l'UI / comportement de l'user à la fois
  - ▣ 1 activité = 1 chose que l'user peut faire – une heuristique qui peut ne pas marcher pour des écrans plus larges.

2

2

## QuoteViewer

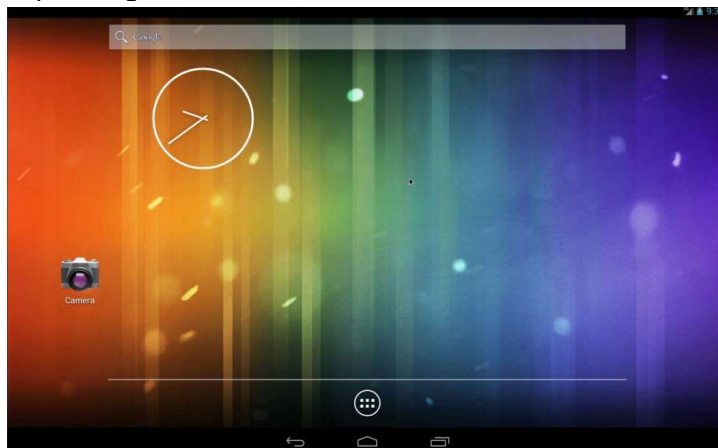
- Une Application qui utilise deux activités :
  - Une qui montre des titres de pièces de théâtre pour «Shakespeare» et permet à l'utilisateur de choisir un Titre.
  - L'autre affiche une citation d'une pièce de théâtre.

3

3

## QuoteViewer UI (1)

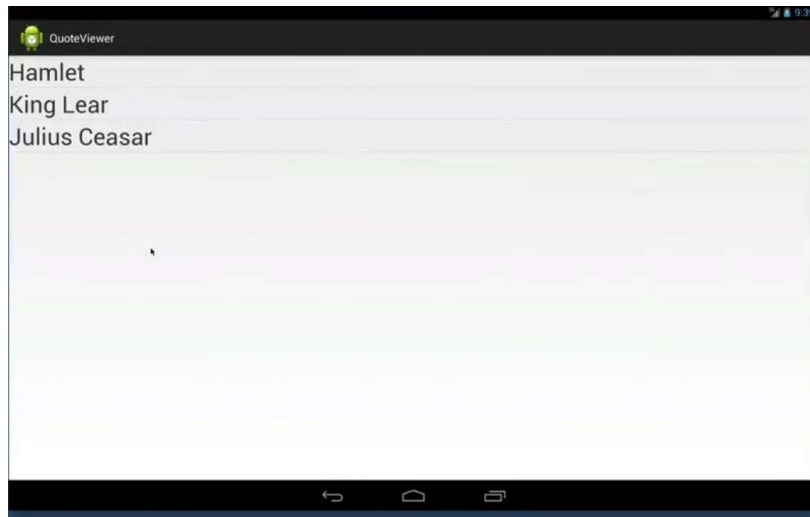
- Raisonnable pour téléphone mais inefficace pour un appareil plus large.



4

4

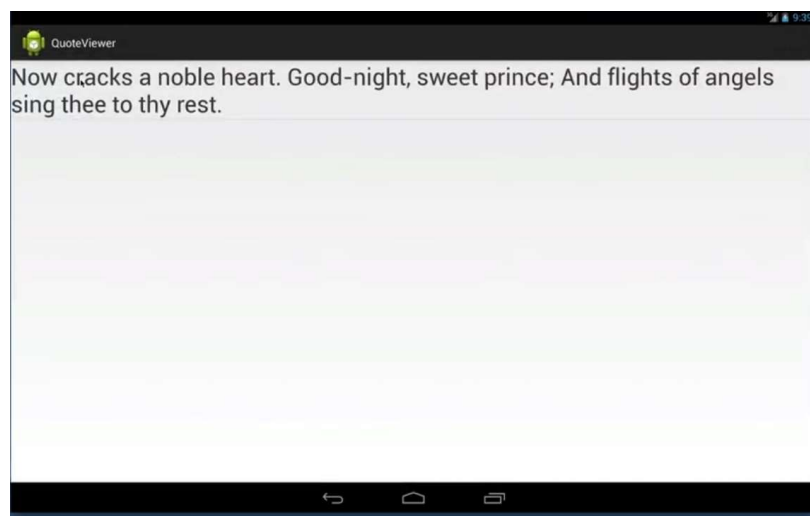
## QuoteViewer UI (2)



5

5

## QuoteViewer UI (3)



6

6

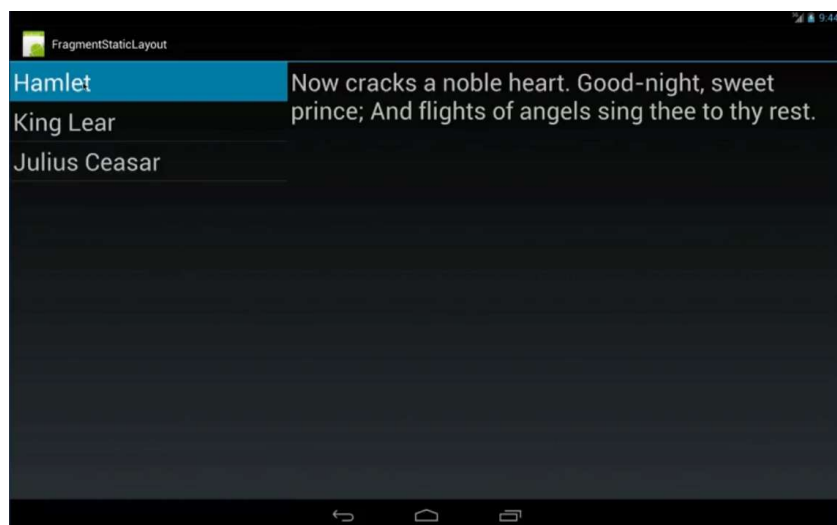
## QuoteViewer (Tablet) – (1)

- Une meilleure disposition (Layout)
  - ▣ Utilise deux unités de « Layout » qui coopèrent sur un seul écran

7

7

## QuoteViewer (Tablet) – (2)



8

8

## Fragment

- Représente un comportement / portion d'un UI au sein d'une activité
- Plusieurs fragments peuvent être intégrés dans une activité afin de créer une interface multi-volet.
- Un seul fragment peut être réutilisé à travers plusieurs activités

9

9

## Cycle de vie d'un fragment

- Le cycle de vie d'un fragment évolue en coordination avec le cycle de vie de l'activité qui le contient.
- Les fragments ont leurs propres cycle de vie et reçoivent leurs propre appels.

10

10

## Etats du cycle de vie d'un fragment

- **Resumed**
  - ▣ Le fragment est visible dans l'activité qui est en cours d'exécution
- **Paused**
  - ▣ Une autre activité est en premier plan et à le contrôle, l'activité conteneur est visible
- **Stopped**
  - ▣ Le fragment n'est pas visible

11

11

## Méthodes du cycle de vie

12

12

## Création d'une activité: OnAttach()

- Le fragment est d'abord attaché à son activité

**onAttach()**

13

13

## OnCreate()

- Initialisation du fragment
  - onCreate() - du fragment
- Différence majeure avec onCreate() - Activity : on ne configure pas l'UI dans cette méthode.

**onAttach()**



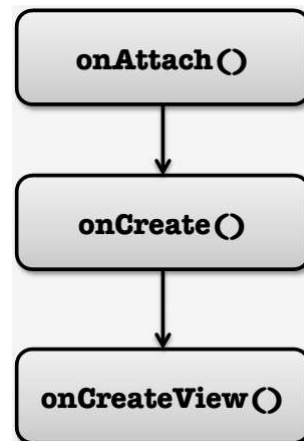
**onCreate()**

14

14

## onCreateView()

- Configure le fragment et retourne son Interface d'utilisateur.
- Installer cette vue dans la hiérarchie des vues de l'activité.

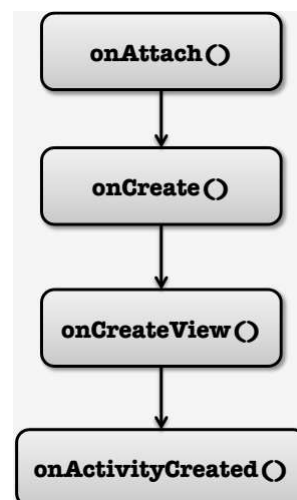


15

15

## onActivityCreated()

- Quand l'activité contenant le fragment a complété l'exécution de onCreate() et le fragment a été installé



16

16



## En suite ...

Une fois le fragment est attaché à l'activité qui le contient, son cycle de vie dépendra du cycle de vie de l'activité

17

17

## L'activité est démarrée: onStart()

- L'activité contenant le fragment est sur le point d'être visible, un appel de `onStart()` est reçu par le fragment.

A grey rectangular button with rounded corners and a thin black border. It contains the text "onStart()" in a bold, black, sans-serif font, followed by a small circle containing a right-pointing arrow.

18

18

## Reprise de l'activité: onResume()

- L'activité contenant le fragment est sur le point d'être visible et prête pour l'interaction avec l'utilisateur

**onResume()**

19

19

## L'activité en pause: onPause()

- L'activité contenant le fragment est visible mais n'a pas le contrôle.

**onPause()**

20

20

## L'activité stoppée: onStop()

- L'activité contenant le fragment n'est plus visible.

**onStop()**

21

21

## L'activité détruite: onDestroyView()

- La vue précédemment créée par `onCreateView()` est maintenant détaché de l'activité.
- Actions typiques:
  - ▣ Libérer les ressources de la vue.

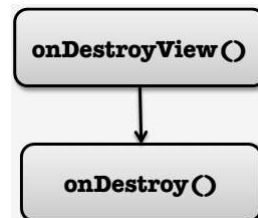
**onDestroyView()**

22

22

## onDestroy()

- Le fragment n'est plus utilisé
- Actions typiques:
  - ▣ Libérer les ressources du fragment

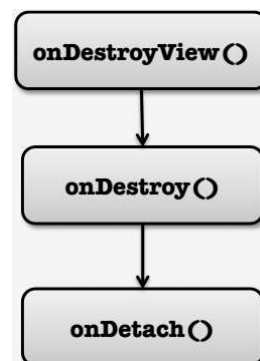


23

23

## onDetach()

- Quand le fragment n'est plus attaché à son activité
- Actions typiques:
  - ▣ Affecter « Null » aux références de l'activité qui contient le fragment.



24

24

## Ajouter des fragments aux activités

En général, il existe deux (02) manières d'ajouter des fragments à un « Layout » d'une activité:

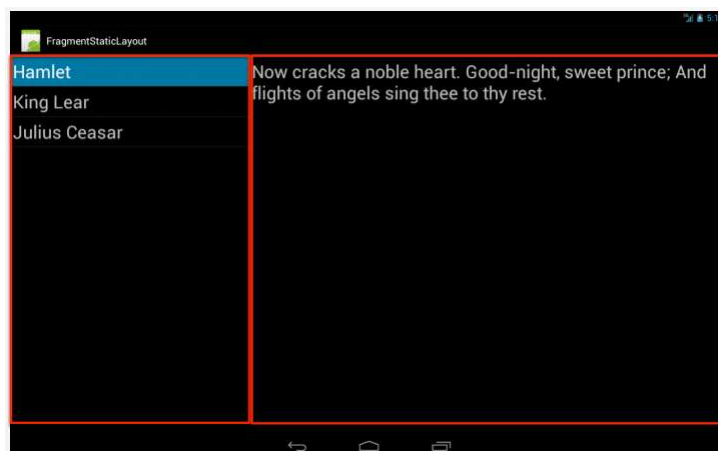
- En faisant une **déclaration statique** dans le fichier « Layout » de l'activité.
- En les ajoutant **par voix de programmation** en utilisant le « **FragmentManager** »

25

25

## Exemple (1)

### FragmentStaticLayout



26

26

## Ajouter des fragments Dynamiquement

Quand une activité est en cours d'exécution, on peut ajouter un fragment à son « Layout »:

1. Se procurer une référence au « FragmentManager »
2. Commencer la transaction du fragment
3. Ajouter le fragment à l'activité
4. Valider la transaction du fragment

27

27

## Exemple (2)

**FragmentProgramaticLayout**

28

28

## Un « Layout » dynamique

- Les transitions des fragments permettent le changement dynamique de l'UI de l'application.
- Ce qui peut rendre l'interface plus fluide et mieux exploiter l'espace disponible de l'écran

29

29

## Exemple (3)

**FragmentManager**

30

30