Android "Master" Class 2013

Lecture 2, October 24, 2013

Prologue

- https://gist.github.com/ajk1311/7145620
- Contains some tedius code to use for examples

Agenda

- Fragments
- LIstViews
- ListAdapters
- Dialogs

Agenda

- **Fragments**
- LIstViews
- ListAdapters
- Dialogs

Fragments

- "A Fragment represents a behavior or a portion of user interface in an Activity"[1]

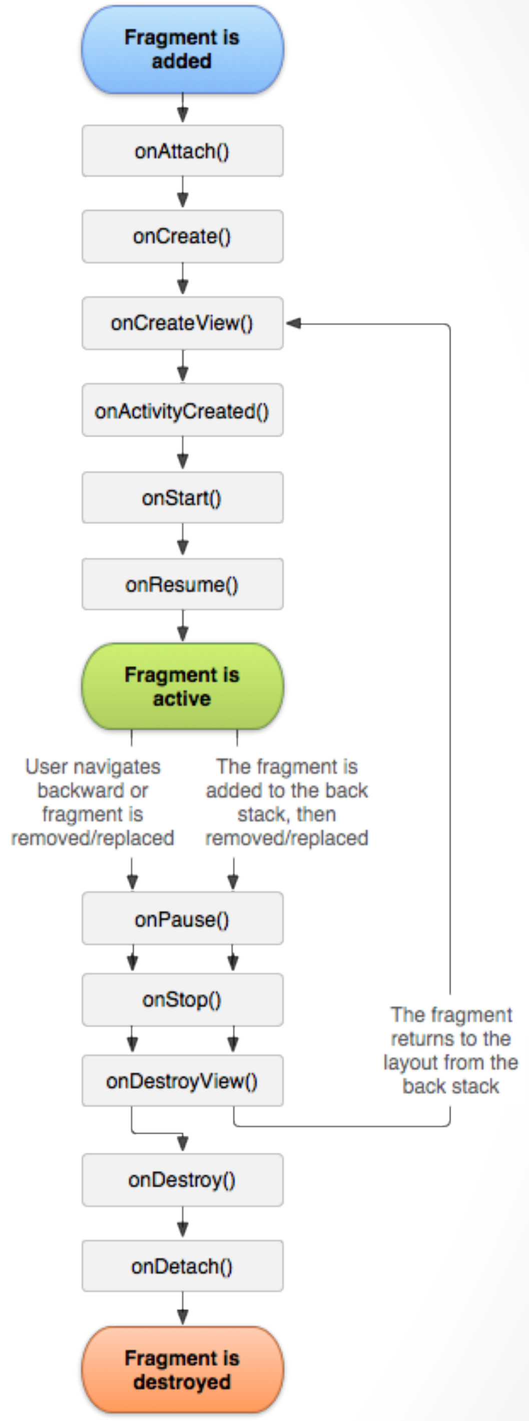
1. http://developer.android.com/guide/components/fragments.html

Fragments

- Activity can have multiple Fragments
- Fragments belong to one Activity

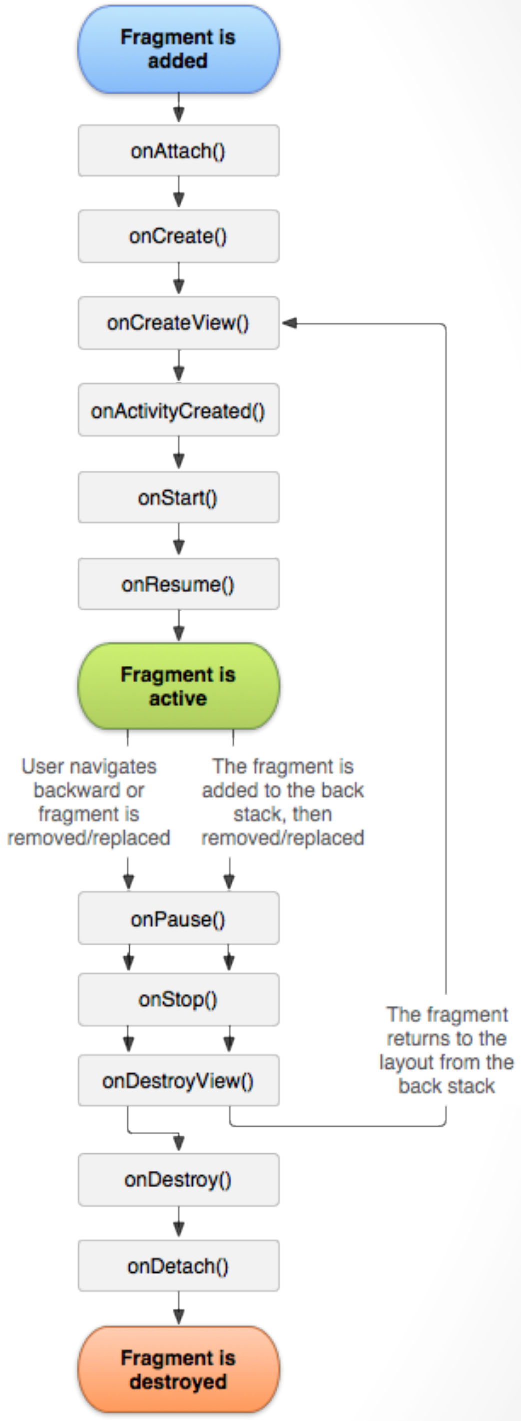
Fragments

- Similar lifecycle to Activiy:



Fragments

- Similar lifecycle to Activiy:
- New callbacks
  - onAttach/onDetach
  - onCreateView/onDestroyView
  - onActivityCreated



Fragments

- onCreateView is to Fragment as setContentView is to Activity

Fragments

- onCreateView is to Fragment as setContentView is to Activity
- Must return the View rather than providing an ID

Fragments

- onCreateView is to Fragment as setContentView is to Activity
- Must return the View rather than providing an ID
- onViewCreated callback provided for when View is guarunteed to be ready

Fragments

- onCreateView is to Fragment as setContentView is to Activity
- Must return the View rather than providing an ID
- onViewCreated callback provided for when View is guarunteed to be ready

- Let's start coding!

Fragments

- Each Activity has a FragmentManager

Fragments

- Each Activity has a FragmentManager
  - Responsible for keeping track of Fragment states
  - Can add, find, remove, replace, and attach/detach Fragments

Fragments

- As a part of XML layout

Fragments

- As a part of XML layout
- As a part of Java code

Fragments

- As a part of XML layout
- As a part of Java code

- Try it out!

Fragments

- How do Fragments talk to their Activities?

Fragments

- How do Fragments talk to their Activities?
  - Just like Views: listener pattern

Fragments

- How do Fragments talk to their Activities?
  - Just like Views: listener pattern
  - We need to define the interface that Activity must implement

Fragments

- How do Fragments talk to their Activities?
  - Just like Views: listener pattern
  - We need to define the interface that Activity must implement
- Example

Agenda

- Fragments
- **LIstViews**
- ListAdapters
- Dialogs

ListView

- Most common element of any mobile application

ListView

- Most common element of any mobile application
- Simply displays data in a scrollable list

ListView

- Each row (or cell) in the list contains a View or ViewGroup

ListView

- Each row (or cell) in the list contains a View or ViewGroup
- Creates a View for each row…

ListView

- Each row (or cell) in the list contains a View or ViewGroup
- Creates a View for each row…
  - Problem?

ListView

- ListView is actually a very complicated piece of code

ListView

- ListView is actually a very complicated piece of code
- “Recycles” Views
  - Allows efficient scrolling with many, many rows

ListView

- ListView is actually a very complicated piece of code
- “Recycles” Views
  - Allows efficient scrolling with many, many rows
- Code may be complicated, but easy to use!

- Lets see how!

ListView

- How do ListViews get the Views associated with backing data?

ListView

- How do ListViews get the Views associated with backing data?
  - Adapters

Agenda

- Fragments
- LIstViews
- **ListAdapters**
- Dialogs

ListAdapters

- Adapt the backing data into a View

ListAdapters

- Adapt the backing data into a View
- ListViews are useless without Adapters

ListAdapters

- Adapters implement the Observer design pattern

ListAdapters

- Adapters implement the Observer design pattern
- They “watch” a set of data and take action when the data set is changed or erased

ListAdapters

- Adapters implement the Observer design pattern
- They "watch" a set of data and take action when the data set is changed or erased
  - notifyDataSetChanged(), notifyDataSetInvalidated()

ListAdapters

- The SDK offers some convenient classes for using Adapters
    - SimpleAdapter, ArrayAdapter, CursorAdapter, etc.

ListAdapters

- The SDK offers some convenient classes for using Adapters
  - SimpleAdapter, ArrayAdapter, CursorAdapter, etc.
- Reduces code to some one-liners, but hides implementation

ListAdapters

- The SDK offers some convenient classes for using Adapters
  - SimpleAdapter, ArrayAdapter, CursorAdapter, etc.
- Reduces code to some one-liners, but hides implementation
- Each inherits from BaseAdapter
  - Seems like a good place to start

ListAdapter

- What happens when a row is clicked?

ListAdapter

- What happens when a row is clicked?
  - Use onItemClickListener
  - Similar to using onClickListener

Agenda

- Fragments
- LIstViews
- ListAdapters
- **Dialogs**

Dialogs

- Use DialogFragment

Dialogs

- Use DialogFragment
  - Maintains state and has Fragment lifecycle

Dialogs

- Additional lifecycle callback: onCreateDialog

Dialogs

- Additional lifecycle callback: onCreateDialog
  - Override this when using the Dialog interface

Dialogs

- Additional lifecycle callback: onCreateDialog
  - Override this when using the Dialog interface
- Could also use onCreateView
  - Override this when providing some custom layout

Dialogs

- onCreateDialog
- Pros:
  - Use AlertDialog.Builder interface for adding all relevant pieces to dialog before showing it
  - Methods return the Builder instance to allow method chaining
  - Simple, self-explanatory methods
- Cons:
  - Not enough customization

Dialogs

- onCreateView
- Pros:
  - As custom as you want to make it
  - Familiar when working with other Fragments
  - Can use DialogFragment as regular Fragment
- Cons:
  - Clunky findViewById calls

Dialogs

- Show DialogFragment using FragmentManager differently
- DialogFragment.show(FragmentManager manager, String tag)

Thank You

Q&A until 9pm