Anthony C Martini
User Interface Assignment
Mobile Application Development
Fall 2014

Table of Contents

Questions and Methods

Cite 5 methods of the Activity class *(Text from source developer.android.com by Google (inc), last updated Oct 1, 2014)* *http://developer.android.com/reference/android/app/Activity.html*	
onCreate()	Called when the activity is first created. This is where you should do all of your normal static set up: create views, bind data to lists, etc. This method also provides you with a Bundle containing the activity's previously frozen state, if there was one. Always followed by onStart().
onRestart()	- Called after your activity has been stopped, prior to it being started again. Always followed by onStart()
onStart()	Called when the activity is becoming visible to the user. Followed by onResume() if the activity comes to the foreground, or onStop() if it becomes hidden
onResume()	- Called when the system is about to start resuming a previous activity. This is typically used to commit unsaved changes to persistent data, stop animations and other things that may be consuming CPU, etc. Implementations of this method must be very quick because the next activity will not be resumed until this method returns. Followed by either onResume() if the activity returns back to the front, or onStop() if it becomes invisible to the user.
onStop()	Called when the activity is no longer visible to the user, because another activity has been resumed and is covering this one. This may happen either because a new activity is being started, an existing one is being brought in front of this one, or this one is being destroyed. Followed by either onRestart() if this activity is coming

	back to interact with the user, or onDestroy() if this activity is going away.
onDestroy()	- The final call you receive before your activity is destroyed. This can happen either because the activity is finishing (someone called finish() on it, or because the system is temporarily destroying this instance of the activity to save space. You can distinguish between these two scenarios with the isFinishing() method.

What methods in the Activity class interact with the menu?

onPreparePanel(int featureId, View view, Menu menu)	Prepare a panel to be displayed.
onPanelClosed(int featureId, Menu menu)	Called when a panel is being closed.
onMenuOpened(int featureId, Menu menu)	Called when a panel's menu is opened by the user.
onMenuItemSelected(int featureId, MenuItem item)	Called when a panel's menu item has been selected by the user.
onCreatePanelMenu(int featureId, Menu menu)	Initialize the contents of the menu for panel 'featureId'.
onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo)	Called when the context menu for this view is being built.
unregisterForContextMenu(View view)	Prevents a context menu to be shown for the given view.
registerForContextMenu(View view)	Registers a context menu to be shown for the given view (multiple views can show the context menu).
openOptionsMenu()	Programmatically opens the options menu.
onPreparePanel(int featureId, View view, Menu menu)	Default implementation of onPreparePanel(int, View, Menu) for activities.
onPrepareOptionsMenu(Menu menu)	Prepare the Screen's standard options menu to be displayed.
onPanelClosed(int featureId, Menu menu)	Default implementation of onPanelClosed(int, Menu) for activities.
onOptionsMenuClosed(Menu menu)	This hook is called whenever the options menu is being closed (either by the user canceling the menu with the back/menu button, or when an item is selected).
onOptionsItemSelected(MenuItem item)	This hook is called whenever an item in your options menu is selected
onMenuOpened(int featureId, Menu menu)	Called when a panel's menu is opened by the user.
onMenuItemSelected(int featureId, MenuItem item)	Default implementation of onMenuItemSelected(int, MenuItem) for activities.
onCreatePanelMenu(int featureId, Menu menu)	Default implementation of onCreatePanelMenu(int, Menu) for activities.
onCreateOptionsMenu(Menu menu)	Initialize the contents of the Activity's standard options menu.
onCreateContextMenu(ContextMenu menu, View v,	Called when a context menu for the view is about to

ContextMenu.ContextMenuInfo menuInfo)	be shown.
onContextMenuClosed(Menu menu)	This hook is called whenever the context menu is being closed (either by the user canceling the menu with the back/menu button, or when an item is selected).
onContextItemSelected(MenuItem item)	This hook is called whenever an item in a context menu is selected.
invalidateOptionsMenu()	Declare that the options menu has changed, so should be recreated.
getMenuInflater()	Returns a MenuInflater with this context.
closeOptionsMenu()	Progammatically closes the options menu.
closeContextMenu()	Programmatically closes the most recently opened context menu, if showing.

How do the classes Activity and ActionBarActivity relate to each others?https://developer.android.com/reference/android/support/v7/app/ActionBarActivity.html

First, this class has inherited fields from the Activity superclass. The action bar allows for users to have a familiar experience to navigate through applications downloaded or created. The ability to relate to the Activity class is most useful in adding "Action Items" using easy to identify "Action Buttons".

(Picture taken from developer.android.com by Google (inc), last updated Oct 1, 2014)

When the app's activity starts the system lines the menu bar with action icons using the onCreateOptionsMenu() which inflates the resource that defines the items within the menu.

** Read chapter 3 of the book - Check*

** Following the book do lab2 of the lecture notes (See the slides: mobile2-android-ui-2014.pptx)*
In reference to

Beginning Android Programming(Paperback ed.2014*) Kevin Grant, Chris Hasemen|Develop and Design: PeachPit Press [pg 76-77].*

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:layout_alignBottom="@+id/next"
   android:layout_alignParentTop="true">

   <View
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:background="#333333"
      android:layout_alignBottom="@+id/next"
      android:layout_alignParentTop="true"/>

   <Button
      android:id="@+id/prev"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentLeft="true"
      android:layout_alignParentTop="true"
      android:text="@string/prev_str"
      android:textColor="@android:color/white"/>

   <Button
      android:id="@id/next"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentTop="true"
      android:layout_alignParentRight="true"
      android:text="@string/next_str"
      android:textColor="@android:color/white"/>

   <TextView
      android:id="@+id/text_view"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
```

```
    android:gravity="center"
    android:layout_toLeftOf="@id/next"
    android:layout_toRightOf="@id/prev"
    android:layout_alignTop="@id/prev"
    android:layout_alignBottom="@id/prev"
    android:text="Cats, Dogs and Birds"
    android:textColor="@android:color/white"/>


  <ImageView

    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/prev"
  android:scaleType="centerCrop"
    android:src="@drawable/one"/>
</RelativeLayout>
```

Taken from Nexus 4.7” on Android Version 2.3.3


Lab2
Prev
Cats, Dogs and Birds
Next
ANGRY BIRDS

Taken on Nexus 4.7” Android Version 3.2

Taken on Nexus 4.7” Android version 4.3