COMP30510 Mobile Application Development

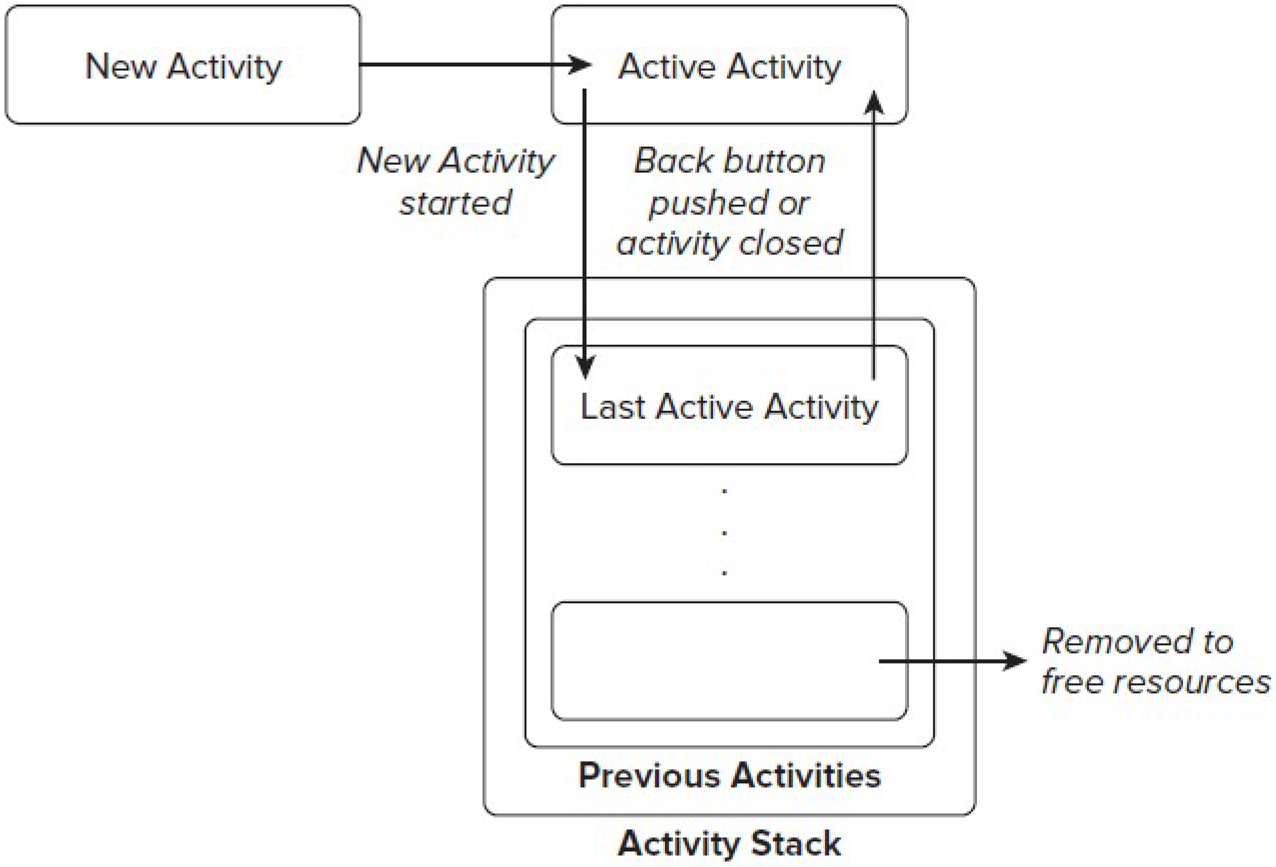
Rotation Handling

Dr. Abraham Campbell

University College Dublin

Abey.campbell@ucd.ie

Activity Stack Recap



©Reto Meier

Rotation Basics

- Android devices provide multiple ways to indicate a need for rotation
  - Accelerometer, opening keyboard...
- Rotation is done by destroying and recreating active/paused activity

Such 'an Appetite for Destruction?'

- Activity destruction/recreation can occur for many different reasons, and screen rotation is just one of them
  - Locale change, low memory...
- Rotation is no different since it can trigger change in layouts as well...

What About Activity's State?

- We save it via *onSaveInstanceState()* when it gets destroyed, passing all the required info into its Bundle argument
- An view linked to an object in code should maintain its state.(*not recommend*)
- Other views will need to be restored afterwards via *onRestoreInstanceState()*

Example:

```
protected void onSaveInstanceState(Bundle
outState)
{
        super.onSaveInstanceState(outState);
        if (myvar!=null) {
                                outState.putString("myVar",
                                myvar.toString());
                           }
}
```

Example:

```
@Override
public void onCreate(Bundle
savedInstanceState)
{
            super.onCreate(savedInstanceState);
            ...
            if (savedInstanceState!=null)
         {
            String myvar = state.getString("myvar");
            }
            ...
}
```

Problems with savedInstanceState()

- Limited to Bundle, *i.e.* it does not help with running threads, opened sockets, etc...
- Significant performance penalty if onCreate() does lots of initialization

Alternatives

- Passing context object via:
  - Object onRetainNonConfigurationInstance()
  - getLastNonConfigurationInstance()!=null
  - It can preserve threads, opened sockets, other information
- Using onCreate() of Application singleton to hold this information instead
- However, Activity destruction/recreation still takes place

Manual Rotation

- It is possible to do manual rotation configuration per activity
- Adding android:configChanges parameter to <activity ...> declaration

  ```
  android:configChanges="keyboardHidden|
  orientation">
  ```

- Implementing onConfigurationChanged() in code

Forcing Screen Orientation

- Add android:screenOrientation = "portrait" to keep screen oriented in portrait
- BUT Activity will still be recreated! To prevent activity destruction/recreation AND keeping screen orientation, use both
  - android:configChanges
  - android:screenOrientation

Questions

- Please ask in the Discussion Forum