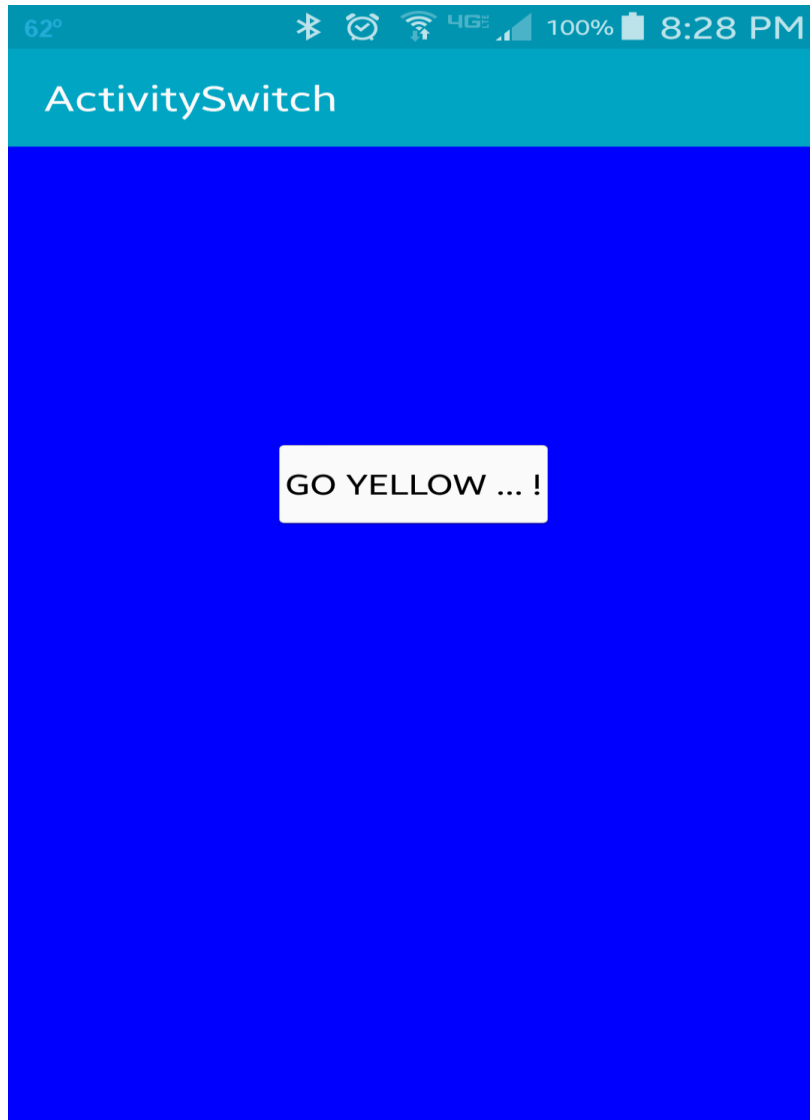


Activity – Activities (Explicit)

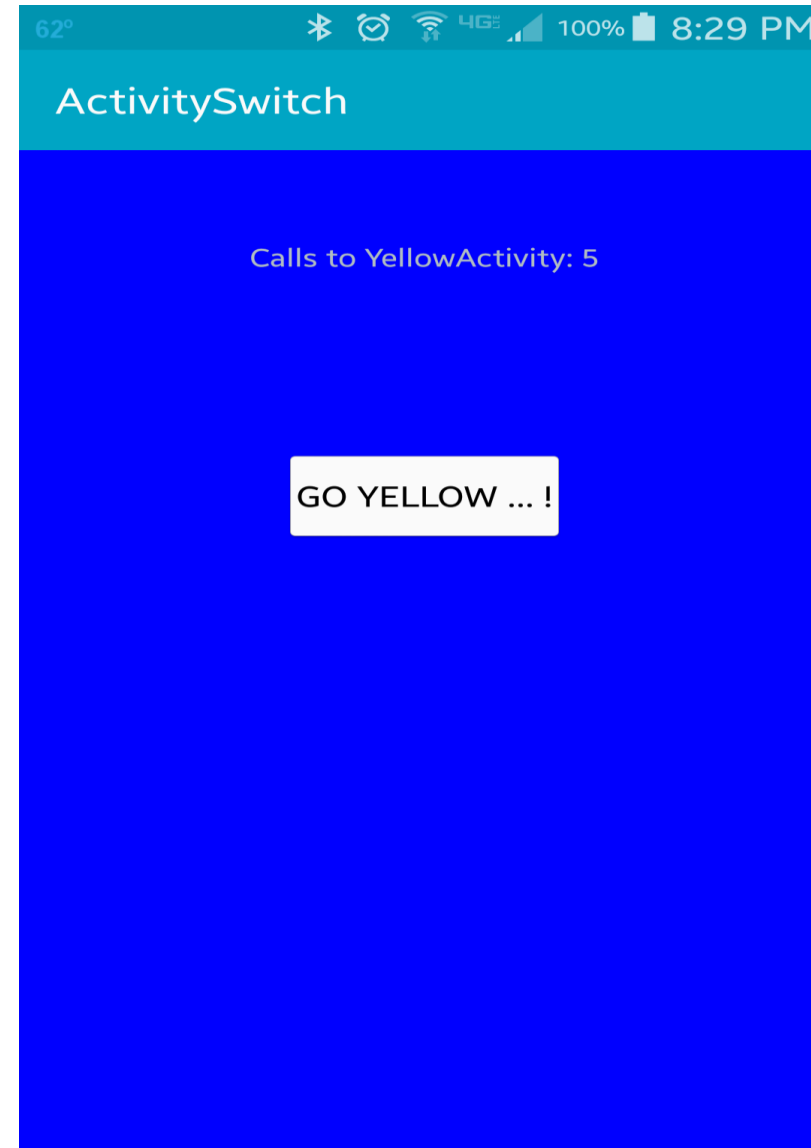
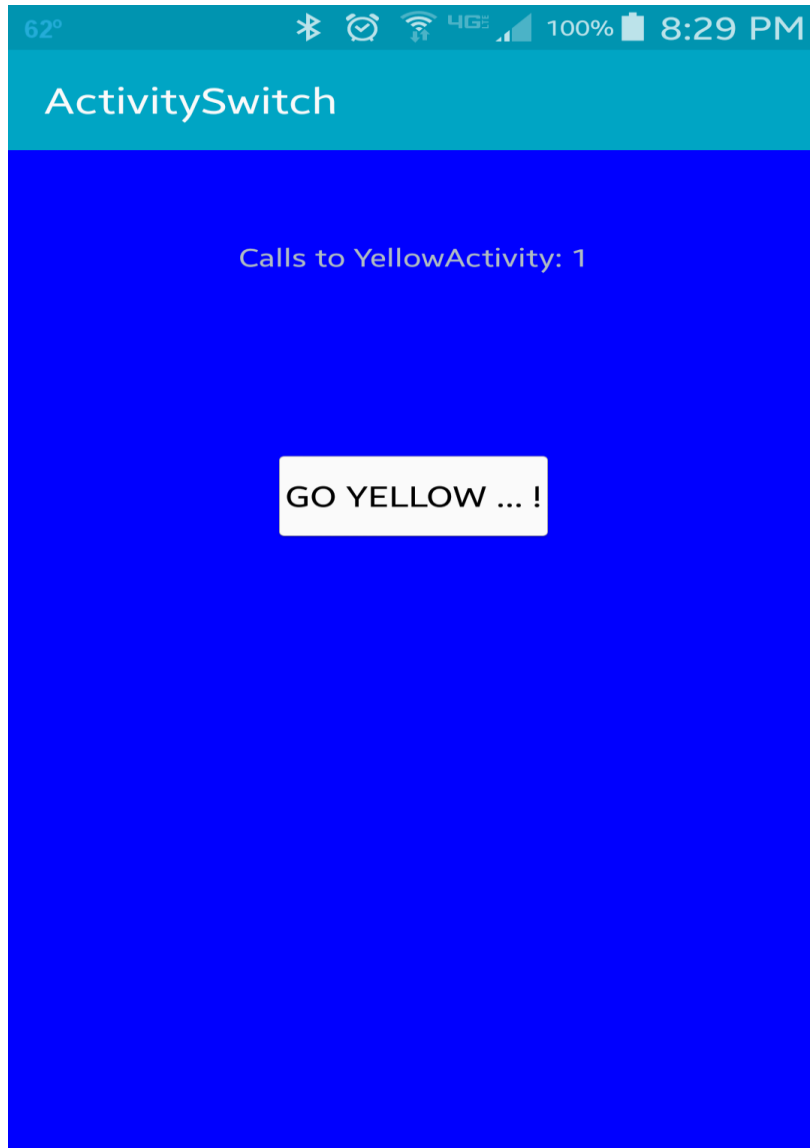
- An application to illustrate the activity life cycle
- This application consists of two activities that call each other
- Activities are referenced explicitly by naming their Java class objects
- If an activity is called, then it is pushed onto the stack to become the main activity
- Exiting main activity results in that activity being popped off the stack.
- Each activity has its own lifecycle and Android will call callbacks at the appropriate moments
- LoggingActivity: a helper class to demonstrate activity lifecycle:
 - Overrides the activity lifecycle methods and logs appropriate messages.

- The app has two activities:
 - `BlueActivity`
 - Main activity
 - An intent has to be constructed to call `YellowActivity`
 - On press of button the `YellowActivity` will be called
 - `YellowActivity`
 - `BlueActivity` is pushed into the background
 - Pressing a button returns to `BlueActivity`
 - A call to `finish()` ends the current activity and returns to the previous one on the activity stack
 - Leaving `YellowActivity` – returns to `BlueActivity`
 - `onActivityResult()` of `BlueActivity` will be called
 - A return value in the form of an Intent can be returned
 - Back button has to be handled by overriding `onBackPressed()`

- On click of button in BlueActivity, the YellowActivity will be pushed on the Stack
- On click of button in YellowActivity, YellowActivity will be popped out of the stack and returned back to BlueActivity.



The YellowActivity keeps track of the number of times the activity was called. This is returned to the blue activity in the form of an Intent which is displayed in the TextView of the BlueActivity.



- When running this application, the logging output can be observed (using DDMS) to understand the activity lifecycle methods that are invoked by Android.

Log	
tag	Message
D alvikvm	GC_EXPLICIT freed 33 objects / 1856 bytes in 55ms
ActivityManager	Starting activity: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] flg=0x10200000 cmp=o
org.xmlvm.tutorial.android.activity.BlueActivity	onCreate() called
org.xmlvm.tutorial.android.activity.BlueActivity	onStart() called
org.xmlvm.tutorial.android.activity.BlueActivity	onResume() called
ActivityManager	Displayed activity org.xmlvm.tutorial.android.activity/.BlueActivity: 530 ms (total 530 ms)
ActivityManager	Starting activity: Intent { cmp=org.xmlvm.tutorial.android.activity/.YellowActivity }
org.xmlvm.tutorial.android.activity.BlueActivity	onPause() called
org.xmlvm.tutorial.android.activity.YellowActivity	onCreate() called
org.xmlvm.tutorial.android.activity.YellowActivity	onStart() called
org.xmlvm.tutorial.android.activity.YellowActivity	onResume() called
ActivityManager	Displayed activity org.xmlvm.tutorial.android.activity/.YellowActivity: 303 ms (total 303 ms)
org.xmlvm.tutorial.android.activity.BlueActivity	onStop() called
org.xmlvm.tutorial.android.activity.YellowActivity	onPause() called
org.xmlvm.tutorial.android.activity.BlueActivity	onRestart() called
org.xmlvm.tutorial.android.activity.BlueActivity	onStart() called
org.xmlvm.tutorial.android.activity.BlueActivity	onResume() called
org.xmlvm.tutorial.android.activity.YellowActivity	onStop() called
org.xmlvm.tutorial.android.activity.YellowActivity	onDestroy() called
ActivityManager	Starting activity: Intent { act=android.intent.action.MAIN cat=[android.intent.category.HOME] flg=0x10200000 cmp=com.a
org.xmlvm.tutorial.android.activity.BlueActivity	onPause() called
InputManagerService	Ignoring hideSoftInput of: com.android.internal.view.InputMethodClient\$Stub\$Proxy@450c9cf8
SntpClient	request time failed: java.net.SocketException: Address family not supported by protocol
org.xmlvm.tutorial.android.activity.BlueActivity	onStop() called

References

- [Activity](#)
- [Activity lifecycle](#)
- [Intent](#)

Exercise

- Add a GreenActivity that can be reached via a button from the YellowActivity. No data needs to be passed between the YellowActivity and the GreenActivity.