

CS 402: Mobile Development

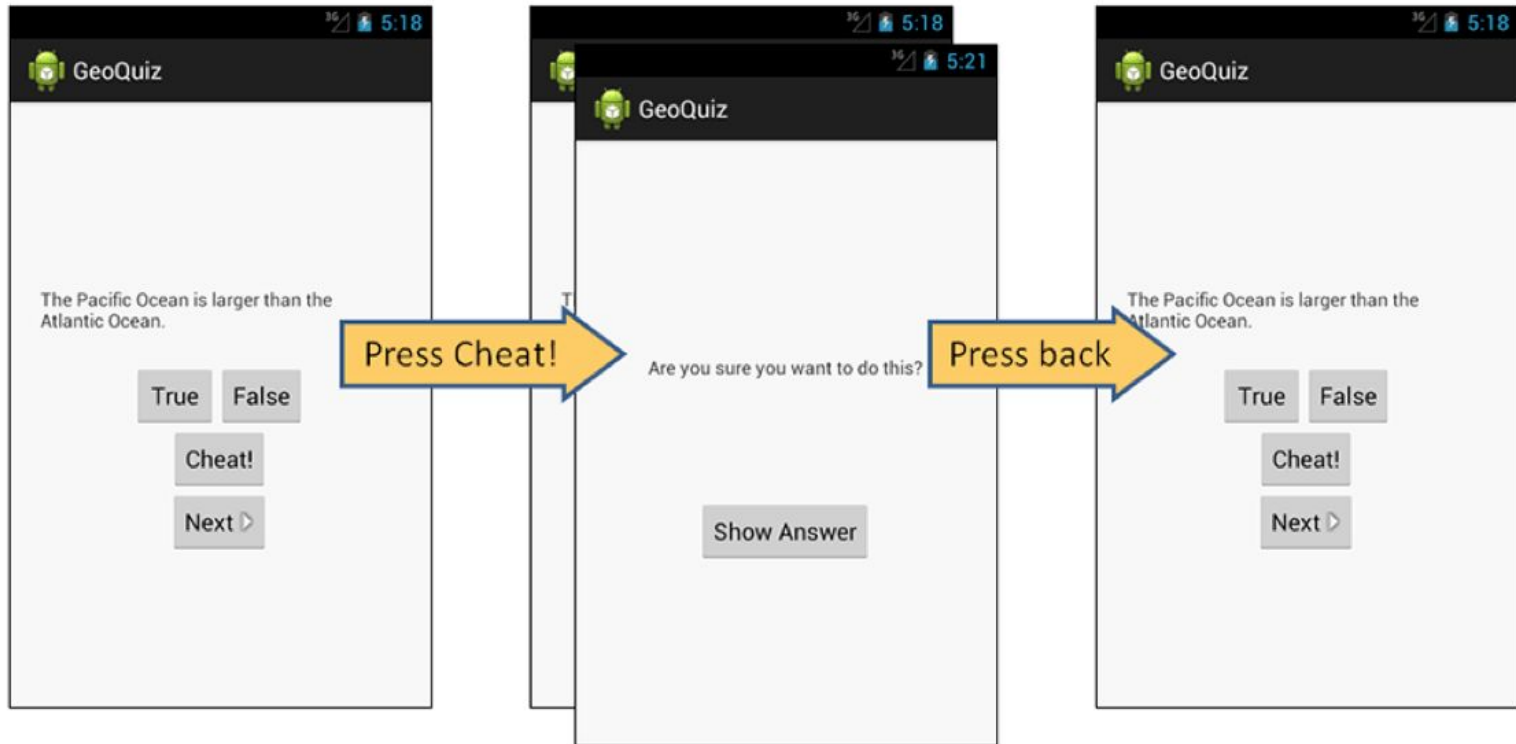
Michael Ziray - michaelziray@boisestate.edu

Fragments

Review

Intents

Activity Stack



Intents

Explicit

Defining of a new Intent and passing that to ActivityManager to start the new Activity.

*// MainActivity wants to launch SecondActivity, so we create an intent
// that contains the calling Activity and the second Activity to be
// created*

```
val intent = Intent(this, Main2Activity::class.java)
```

```
intent.putExtra("thekey", "the value")
```

```
intent.putExtra("person", person)
```

// Tell the activity manager to handle the intent

```
startActivity(intent)
```

Retrieving Data from Parent

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_main2)  
  
    val stringFromActivity:String = intent.getStringExtra("thekey")  
    val person:Person = intent.getSerializableExtra("person") as Person  
    setResult(Activity.RESULT_OK)  
}
```

1 - Retrieving Data from Child

```
// For parents awaiting responses from their children, instead of
// startActivity we use:
Intent newIntentObject =
    new Intent( MainActivity.this, SecondActivity.class );

startActivityForResult( newIntentObject, 0); // Intent, ID
```

2 - Retrieving Data from Child

```
// In Child Activity, in a method that handles the response
val returnIntent:Intent = Intent()
returnIntent.putExtra( NAME_CONSTANT, variable )
setResult( RESULT_OK /* or RESULT_CANCEL */, returnIntent );
finish()
```

3 - Retrieving Data from Child

// Back in Parent Activity

```
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {  
    super.onActivityResult(requestCode, resultCode, data)  
}
```


Fragments

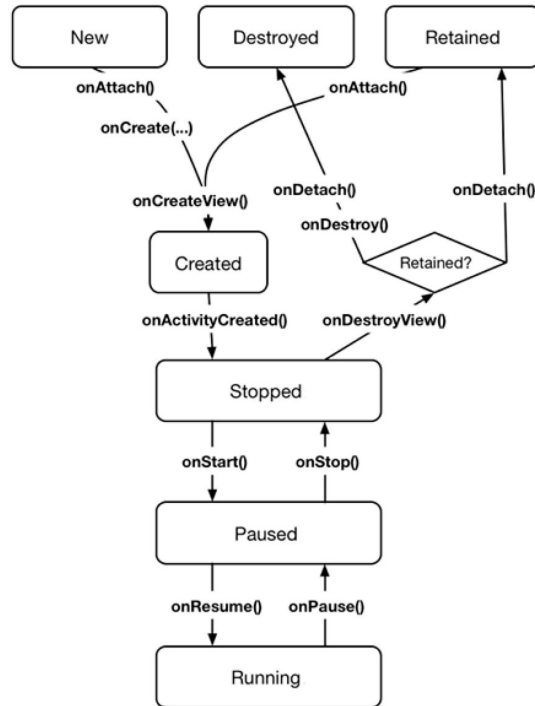
Implemented in Honeycomb (3.0)

All Activity class implementations before that don't know how to handle Fragments.

Fragments allow different layouts to be used in an Activity

Fragment Lifecycle

Figure 7.18 The fragment lifecycle, again



Declare Your Fragment Class

```
class SomeFragment: Fragment
```

Fragment's onCreate

override fun onCreate(savedInstanceState: Bundle?)

Activity's onCreate method is protected
Fragment's is public

onCreateView

```
override fun onCreateView(inflater: LayoutInflater?, container: ViewGroup?,
    savedInstanceState: Bundle?): View?
```

Inflate the Layout

```
val viewToReturn = inflater!!.inflate(R.layout.fragment_gallery, container, false)
```

```
return viewToReturn
```

Get a UI Element Reference

```
val someTextView: TextView = viewToReturn.findViewById(R.id.something)
```

Get Fragment Manager

```
// In parent Activity
supportFragmentManager
    .beginTransaction()
    .replace(R.id.frameContainer, galleryFragment)
    .addToBackStack(galleryFragment.toString())
    .setTransition(FragmentTransaction.TRANSIT_FRAGMENT_OPEN)
    .commit()
```


Load the Fragment

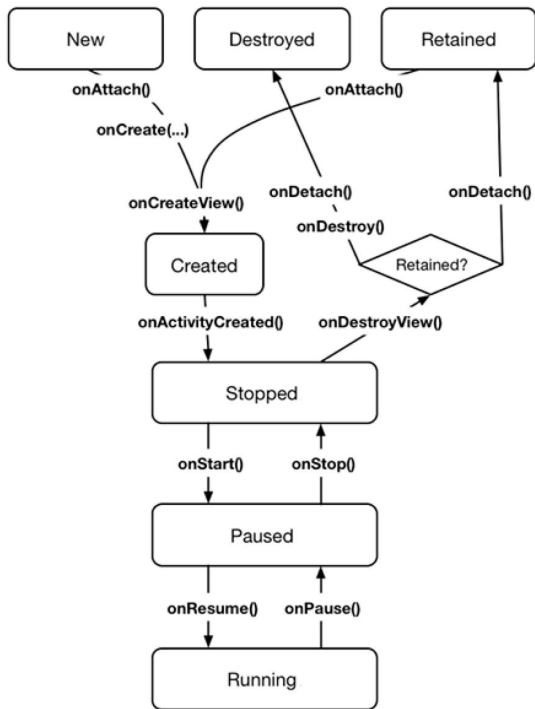
```
val someFragment =  
    fragmentManager.findFragmentById(R.id.fragmentContainer);
```

Create Fragment

```
val newFragment = AFragmentClass.newInstance()
```

Fragment Lifecycle

Figure 7.18 The fragment lifecycle, again



Debugging

Log.d - Debug messages

Log.v - Verbose

Log.i - Info

Log.e - Errors

Log.w - Warning

Log.wtf - WTF

Demo

Debugging in Android Studio

Toasts

```
Toast.makeText( Context, Message, Duration);
```

```
// Example
```

```
Toast.makeText(this, "Toast", Toast.LENGTH_LONG).show();
```

Demo

Toasts