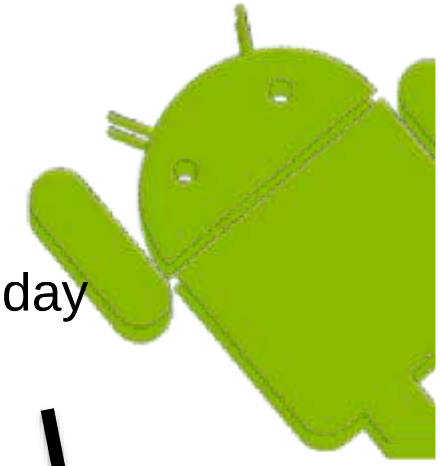
Mobile app development – week 2



Wednesday
Feb 3

Wednesday
Feb 10

Questions and comments to WorkplanW2

Java background

Background for next week

Exercises

Questions

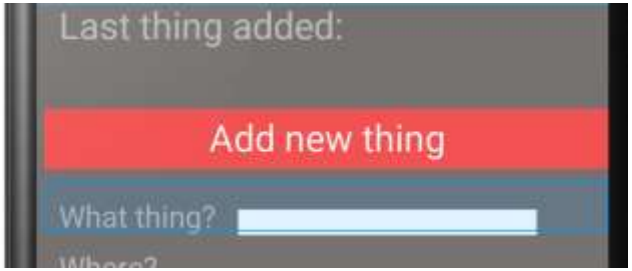
Alignment in linear layout

Package names

Libraries, gradles and SDK

2nd edition of book compared to 1st

Right alignment in a linear layout?



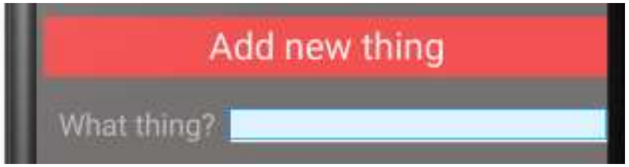
Last thing added:

What thing?

Where?

```
<EditText
    android:id="@+id/what_text"
    android:layout_width="200sp"
    android:layout_height="wrap_content"
     android:background="#ffffff"
    />
```

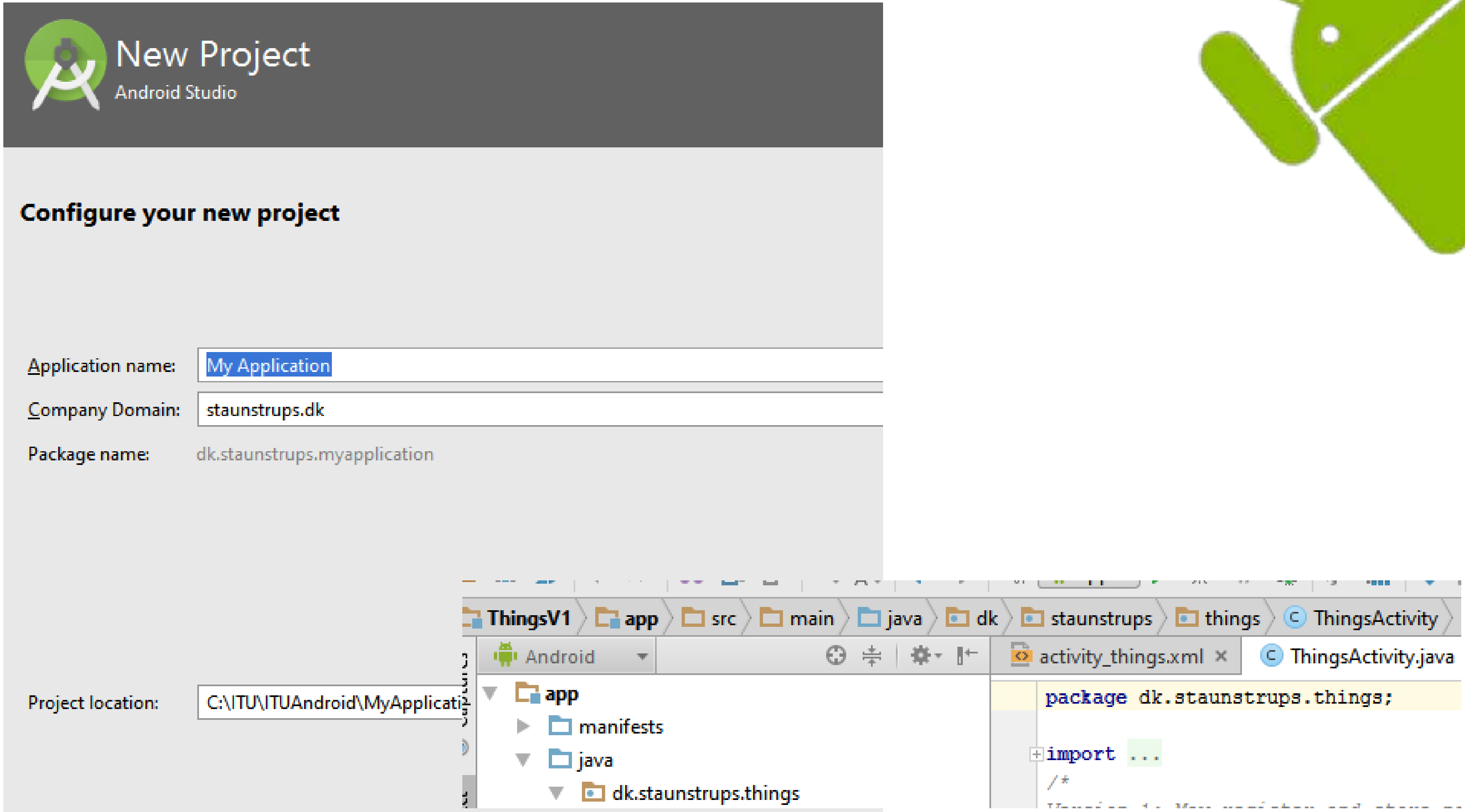
Right alignment in linear layout



```
<EditText
    android:id="@+id/what_text"
    android:layout_width=”match_parent"
    android:layout_height="wrap_content"
    android:background="#ffffff”

    android:gravity=”right”
    />
```

http://sandipchitale.blogspot.co.uk/2010/05/linearlayout-gravity-and-layoutgravity.html

Package names



New Project
Android Studio

Configure your new project

Application name: My Application
Company Domain: staunstrups.dk
Package name: dk.staunstrups.myapplication

Project location: C:\ITU\ITUAndroid\MyApplicati

```
package dk.staunstrups.things;

import ...
/*
```

Libraries, gradles and SDK

```
android {
    compileSdkVersion 23
    buildToolsVersion "23.0.1"
    defaultConfig {
        applicationId "com.bignerdranch.android.geoquiz"
        minSdkVersion 16
        targetSdkVersion 19
    }
    buildTypes {
        release {
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:23.1.1'
}
```

Java background

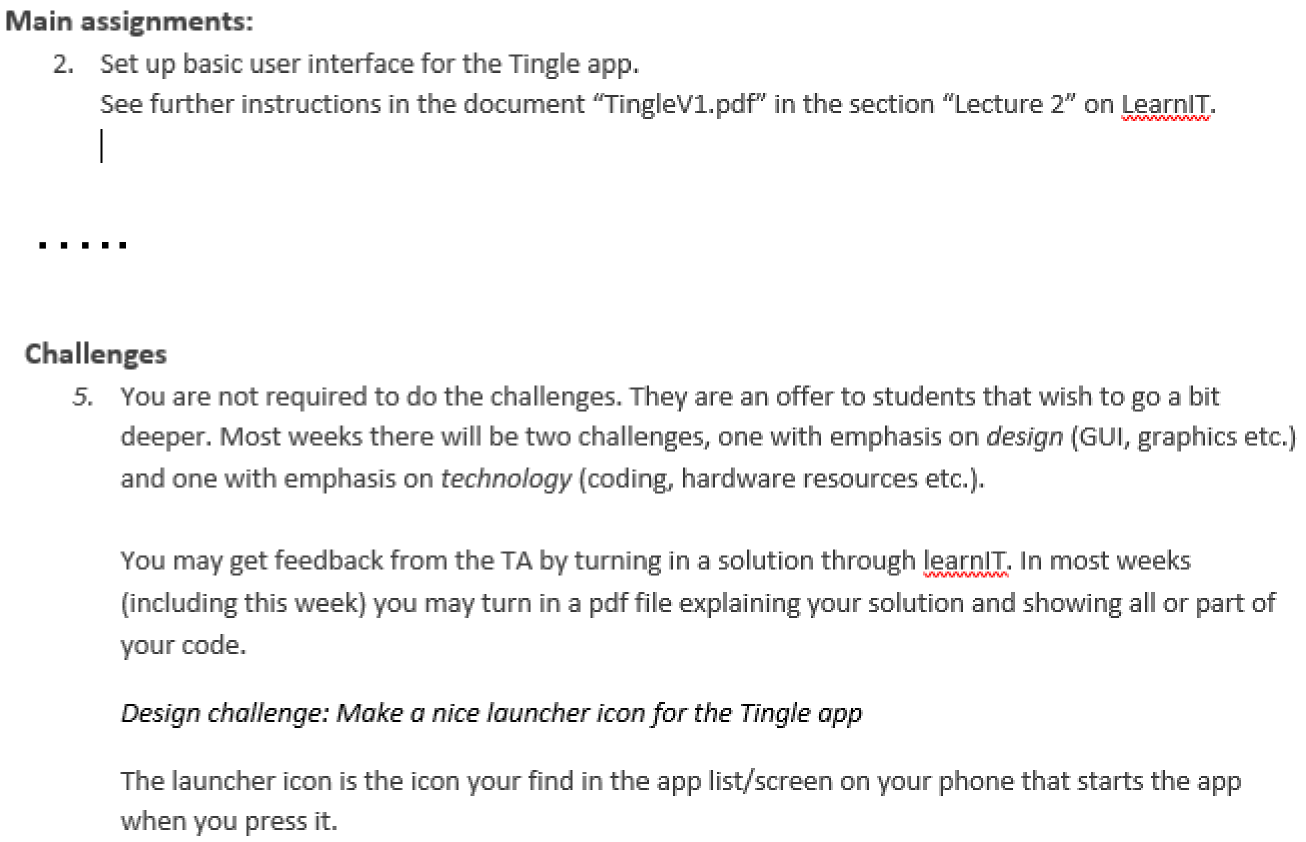
abstract class

interface

```
interface Bicycle {

  void changeCadence(int newValue);

  void changeGear(int newValue);

  void speedUp(int increment);

  void applyBrakes(int decrement);

 }
```

http://www.onsandroid.com/2011/12/difference-between-interface-and.html

Assignments and challenges



Main assignments:

2. Set up basic user interface for the Tingle app.
   See further instructions in the document "TingleV1.pdf" in the section "Lecture 2" on LearnIT.

. . . . .

Challenges

5. You are not required to do the challenges. They are an offer to students that wish to go a bit deeper. Most weeks there will be two challenges, one with emphasis on *design* (GUI, graphics etc.) and one with emphasis on *technology* (coding, hardware resources etc.).

   You may get feedback from the TA by turning in a solution through learnIT. In most weeks (including this week) you may turn in a pdf file explaining your solution and showing all or part of your code.

   Design challenge: Make a nice launcher icon for the Tingle app

   The launcher icon is the icon your find in the app list/screen on your phone that starts the app when you press it.

Kenneys solution to the design challenge

More questions?

Questions are welcome during the week

Abdulrashid Mas'ab Mohammed - abmm@itu.dk

Radoslav Todorov -Skype ID: radoslav.h.todorov
e-mail: rahr@itu.dk

Jørgen Staunstrup – jst@itu.dk

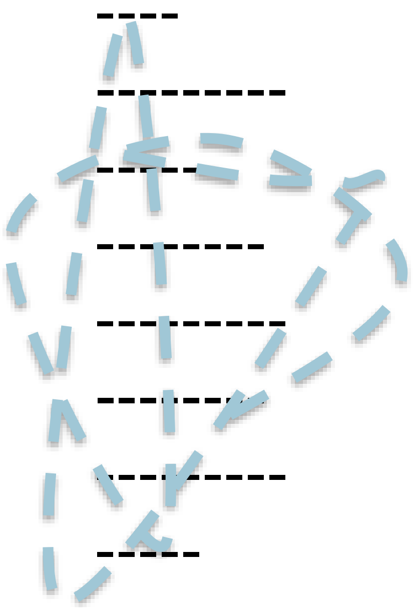
After break: Background for next week

Why is programming hard?

static

-----

dynamic



From computing to interaction

Transistormaskinen GIER, 1961

Single stream

```
------------
----
--------
-----
---
-------
```

Multistream

```
------------    ----           ------------
----            ---------      ----
--------        -----          --------
-----           --------       -----
---             ---------      ---
-------         --------       -------
                ---------
                -----
```

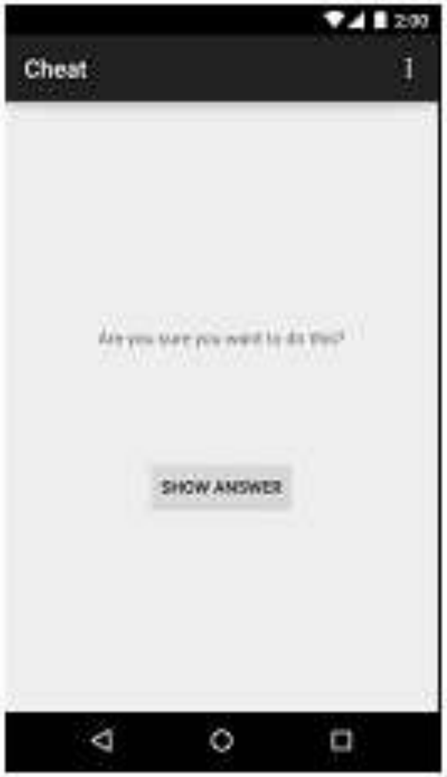
Handling a button in Android

```
mTrueButton = (Button) findViewById(R.id.true_button);
mTrueButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(QuizActivity.this,
                R.string.correct_toast,
                Toast.LENGTH_SHORT).show();
    }
});
```

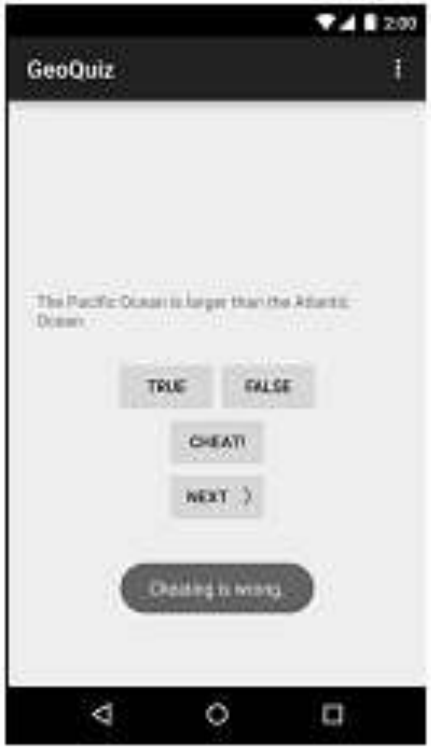
```
mTrueButton = (Button) findViewById(R.id.true_button);
mTrueButton.setOnClickListener((v) -> {
Toast.makeText(QuizActivity.this,
                R.string.correct_toast,
                Toast.LENGTH_SHORT).show();
    }
});
```

Activity in Android

An activity is a single, focused thing that the user can do



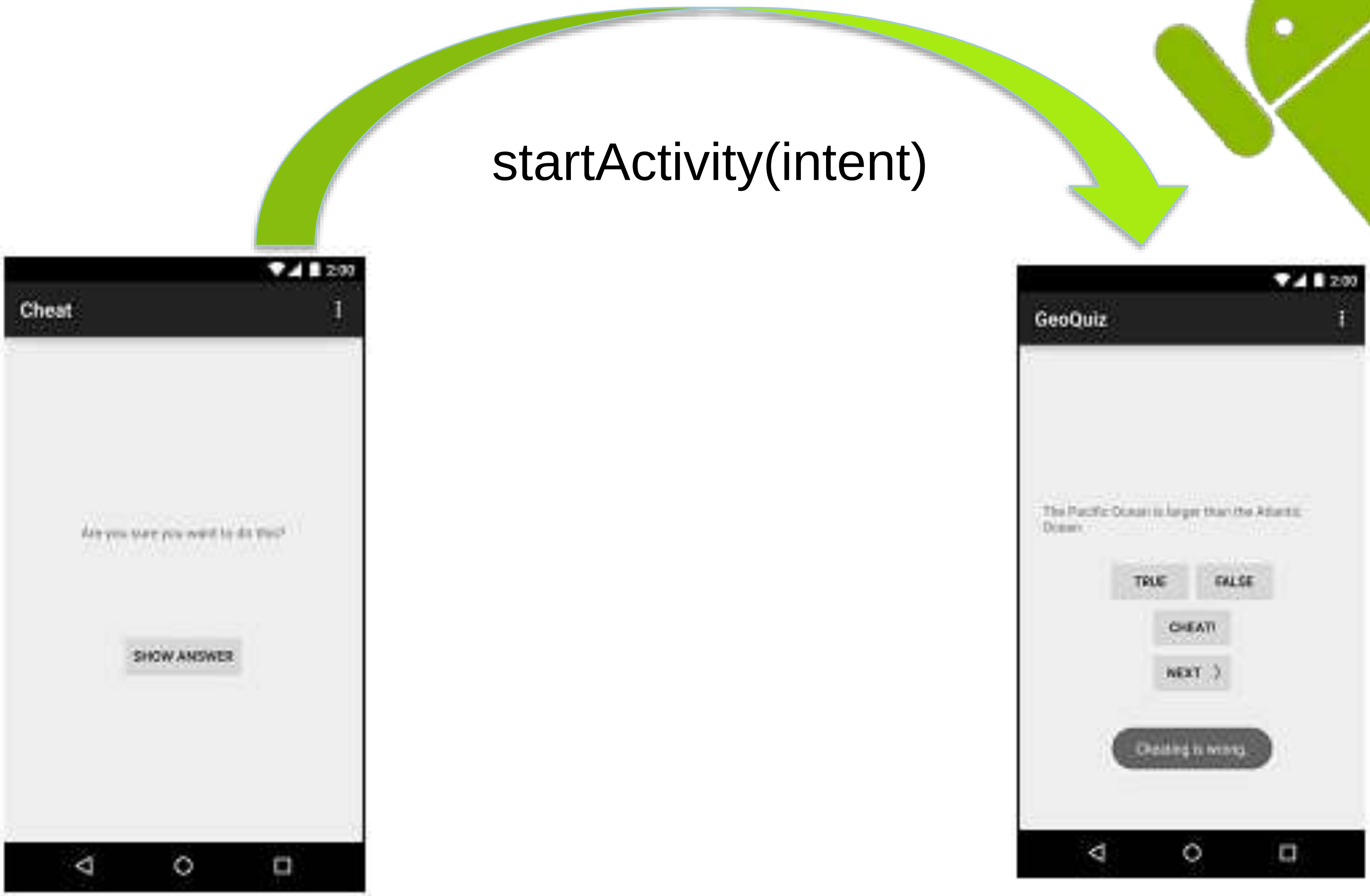
Are you sure you want to do this?



The Pacific Ocean is larger than the Atlantic Ocean.

Cheating is wrong.

Intent



startActivity(intent)

Are you sure you want to do this?

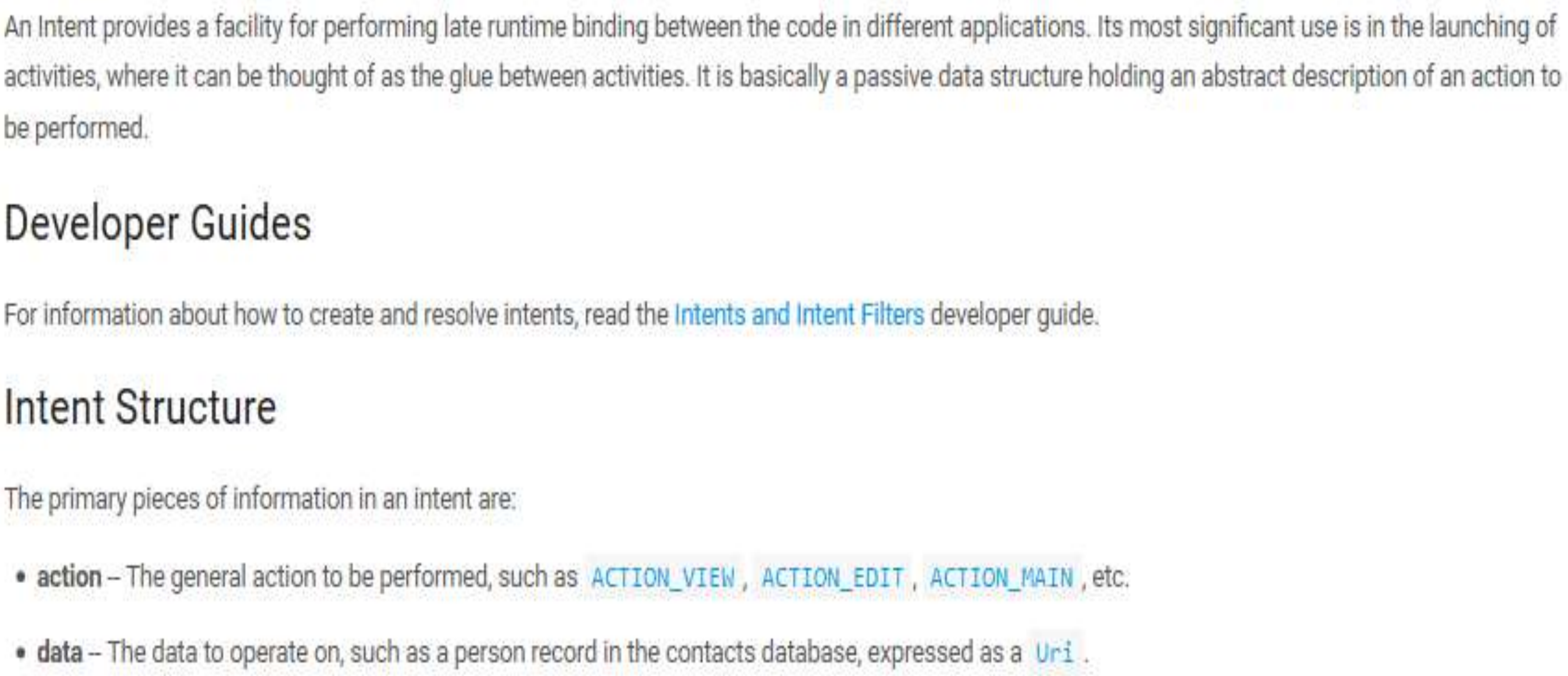
The Pacific Ocean is larger than the Atlantic Ocean.

Cheating is wrong.

Intent

```
Intent startIntent;
```

http://developer.android.com/reference/android/content/Intent.html



An Intent provides a facility for performing late runtime binding between the code in different applications. Its most significant use is in the launching of activities, where it can be thought of as the glue between activities. It is basically a passive data structure holding an abstract description of an action to be performed.

Developer Guides

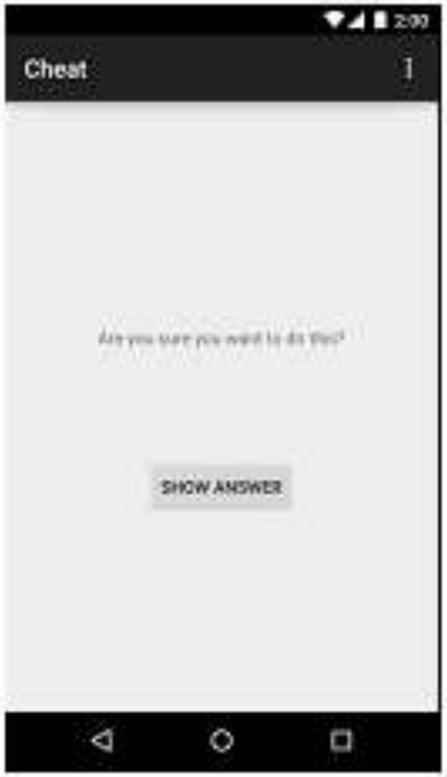
For information about how to create and resolve intents, read the Intents and Intent Filters developer guide.

Intent Structure

The primary pieces of information in an intent are:

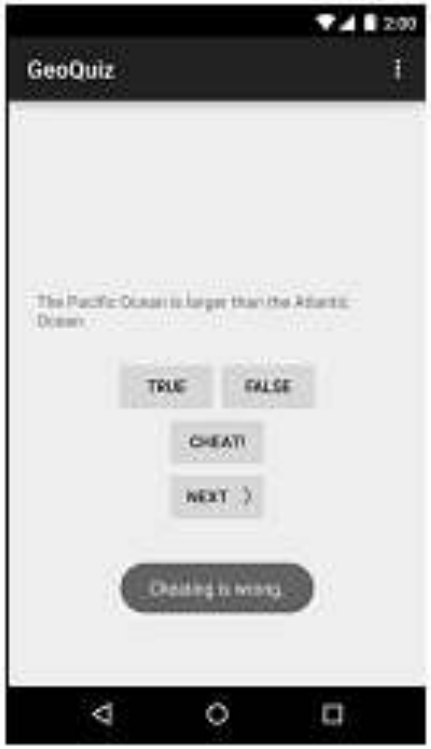
- **action** -- The general action to be performed, such as `ACTION_VIEW` , `ACTION_EDIT` , `ACTION_MAIN` , etc.
- **data** -- The data to operate on, such as a person record in the contacts database, expressed as a `Uri` .

Intent



Are you sure you want to do this?

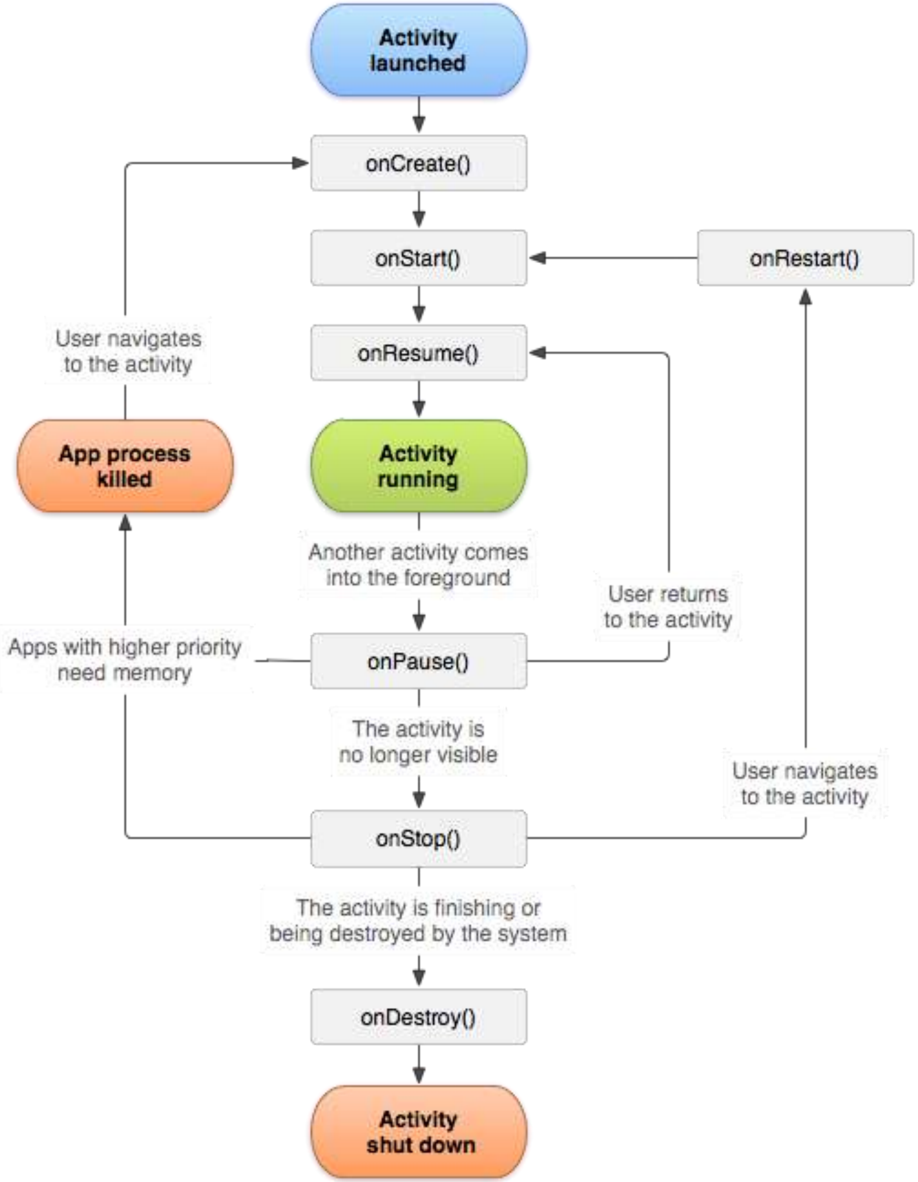
startActivity(intent)



The Pacific Ocean is larger than the Atlantic Ocean.

Cheating is wrong.

Activity lifecycle


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Activity skeleton

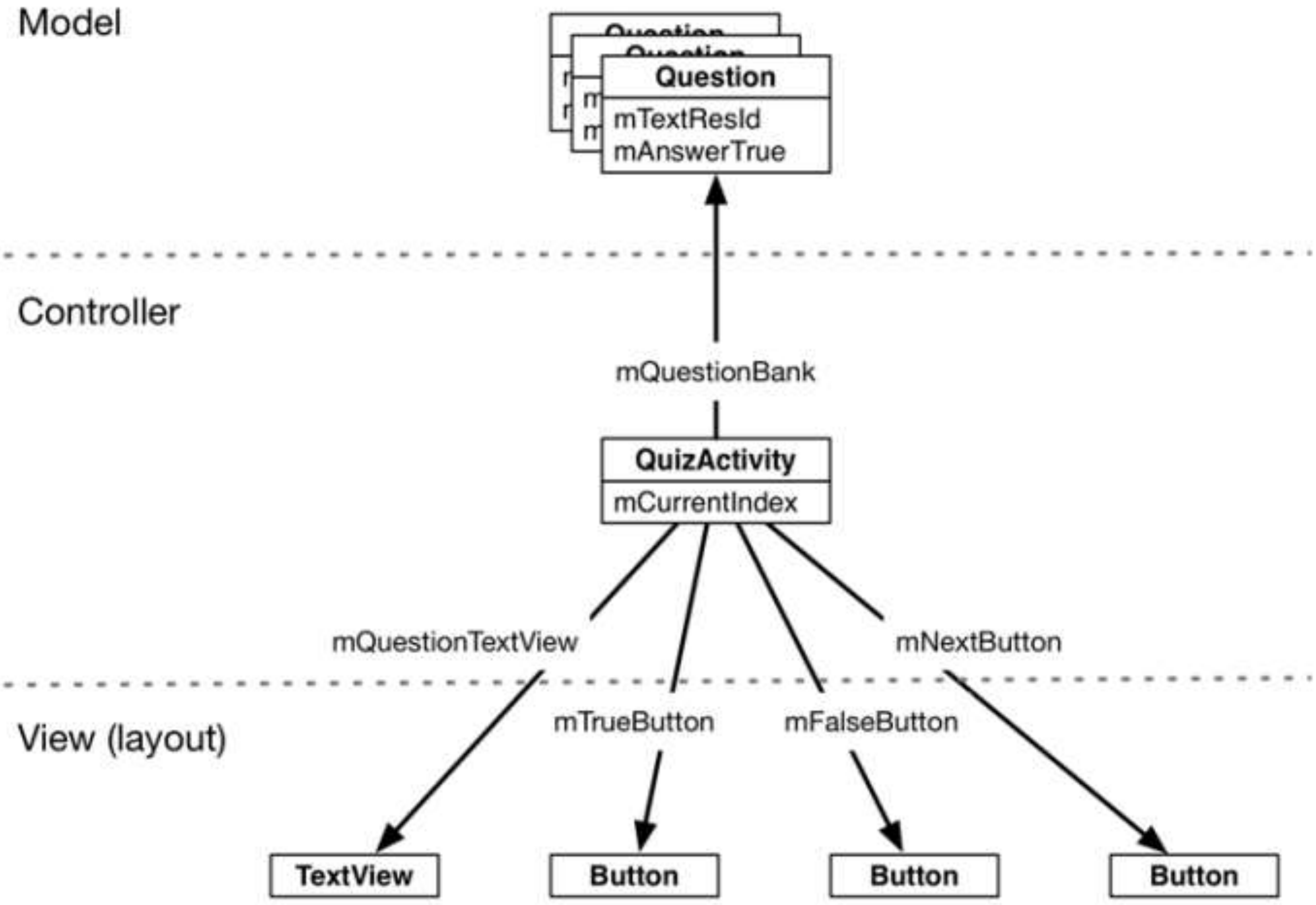
```
public class MainActivity extends Activity {


        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.main);        ...   }

        protected void onDestroy() {
                super.onDestroy();
                ...    }

        protected void onPause() {
                super.onPause();
                ...    }

        protected void onRestart() {
                super.onRestart();
                ...    }

        protected void onResume() {
                super.onResume();
                ...    }

        protected void onStart() {
                super.onStart();
                ...    }

        protected void onStop() {
                super.onStop();
                ...    }
}
```

Structuring Android code

Multistream

```
------------
----                    ----         ------------
--------                ---------    ----
-----                   -----        --------
---                     --------     -----
-------                 ---------    ---
                        --------     -------
                        ---------
                        -----
```

Model-View-Controller pattern



Android architecture

https://www.youtube.com/watch?v=deq8mkt_cxQ