Beginner's Android Development Tutorial

Georgia Tech Research Network Operations Center (RNOC)

cic.gatech.edu

Questions ? Get in touch

piazza.com/gatech/spring2015/cic

rnoc-lab-staff@lists.gatech.edu

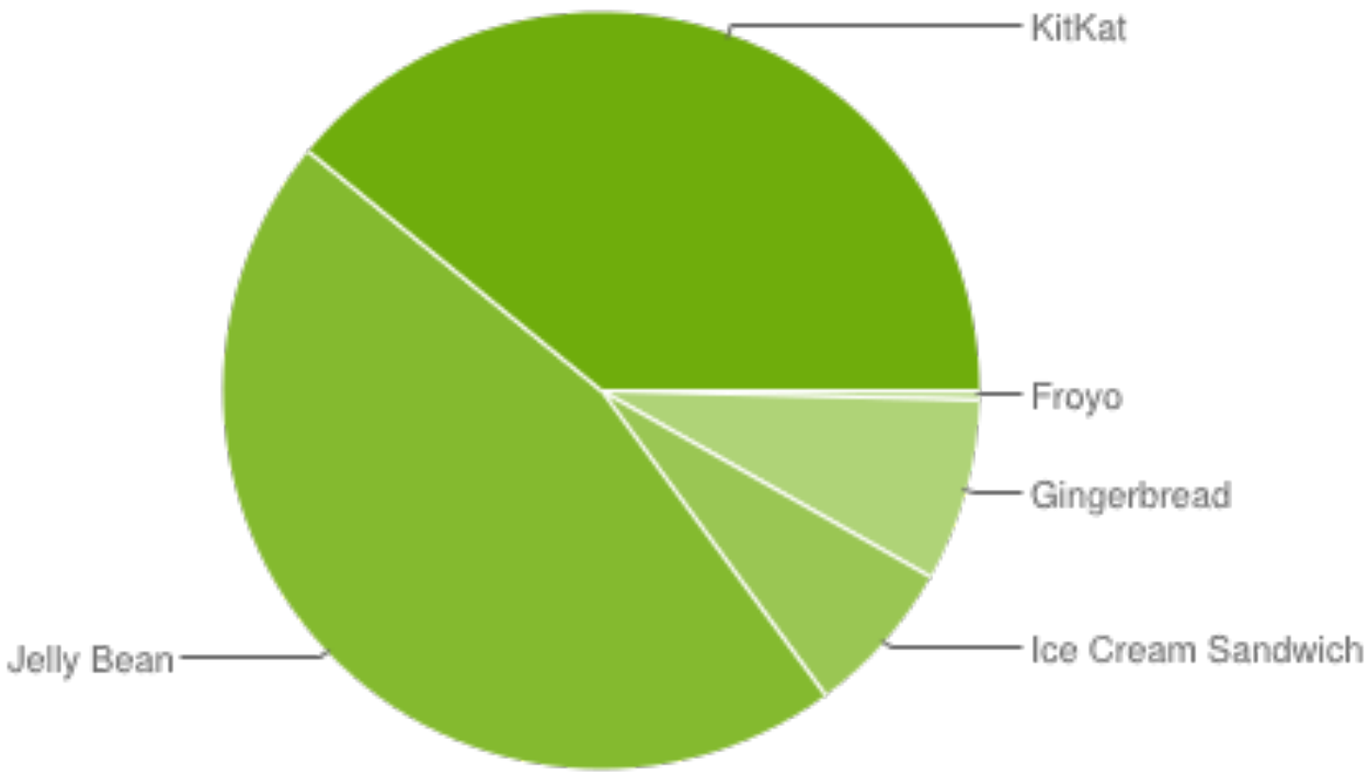
Advantages of Android Development

- Low barrier of entry
- Large user base
- Development OS not enforced
- Java is more familiar than Objective C to most developers

Disadvantages of Android Development

- Fragmentation of devices / OS
- Monetization-wise iOS apps still rule the charts
- Apple App Store's more stringent acceptance policies lead to better quality of apps.

Platform Distributions



Version	Codename	API	Distribution
2.2	Froyo	8	0.4%
2.3.3 - 2.3.7	Gingerbread	10	7.8%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	6.7%
4.1.x	Jelly Bean	16	19.2%
4.2.x		17	20.3%
4.3		18	6.5%
4.4	KitKat	19	39.1%

Data collected during a 7-day period ending on January 5, 2015.
Any versions with less than 0.1% distribution are not shown.

Android Studio

The Official IDE for Android

[Download Link]

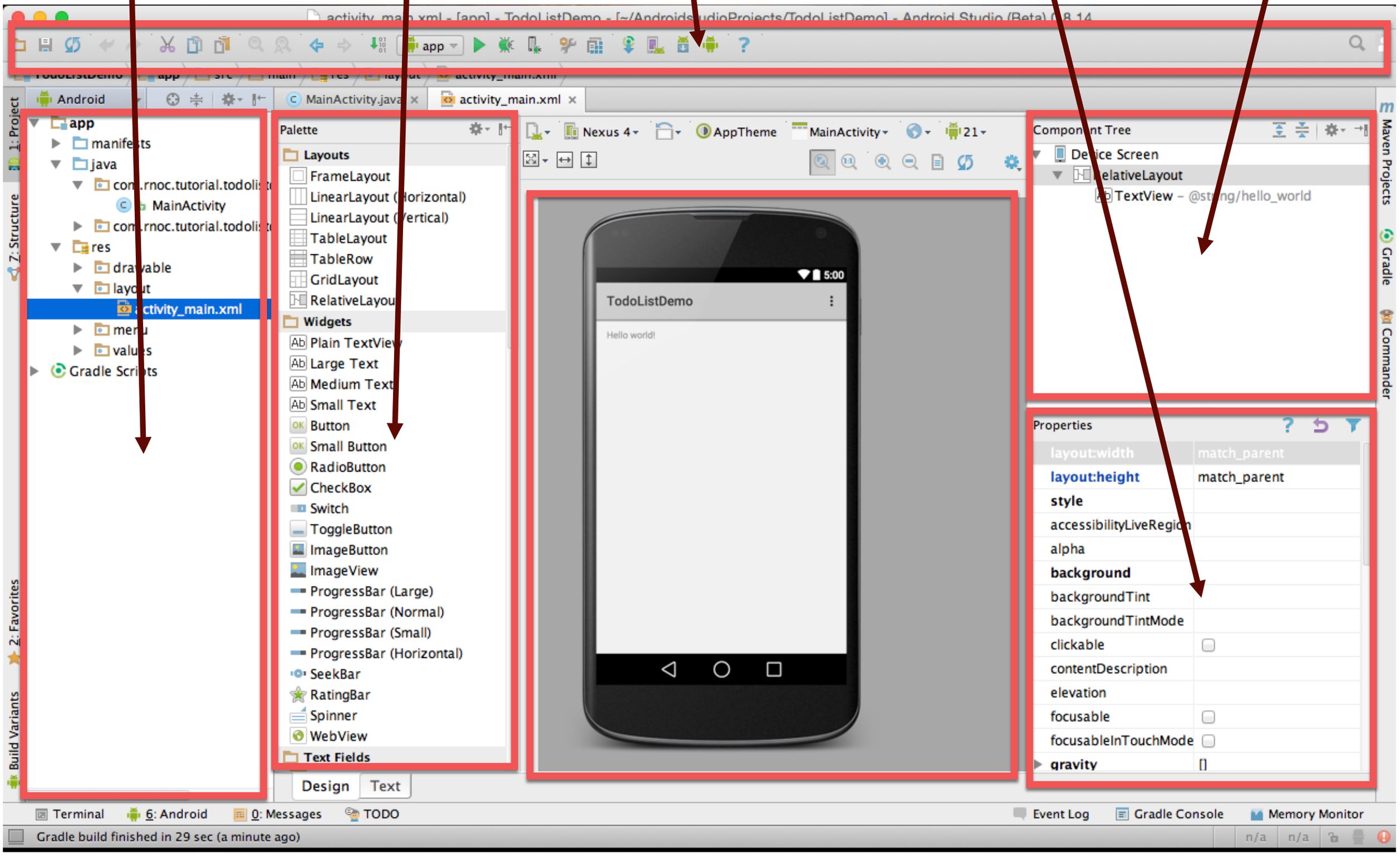
Project View

Palette

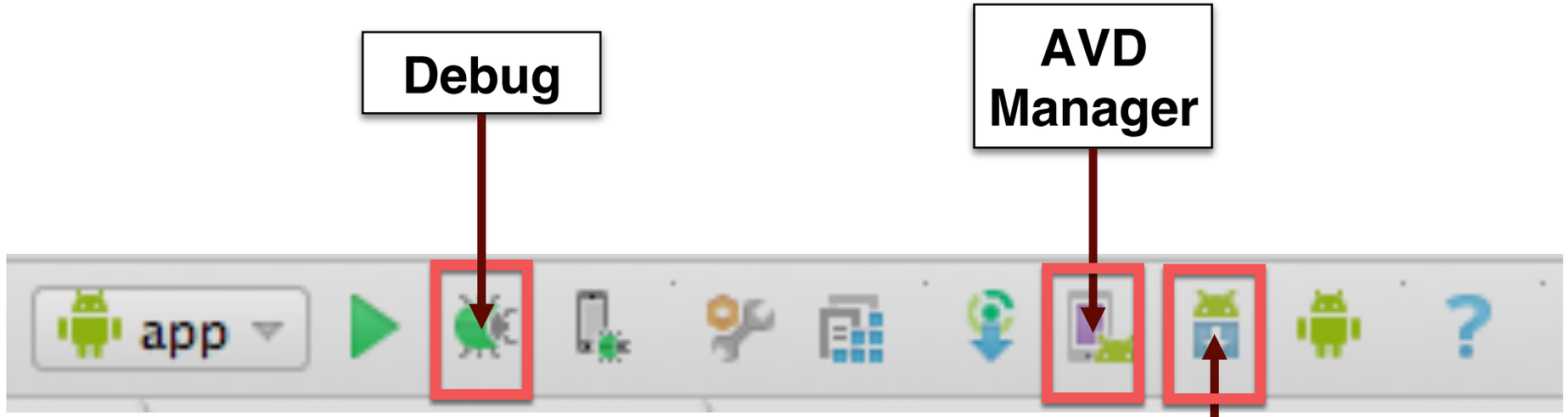
Toolbar

Properties Editor

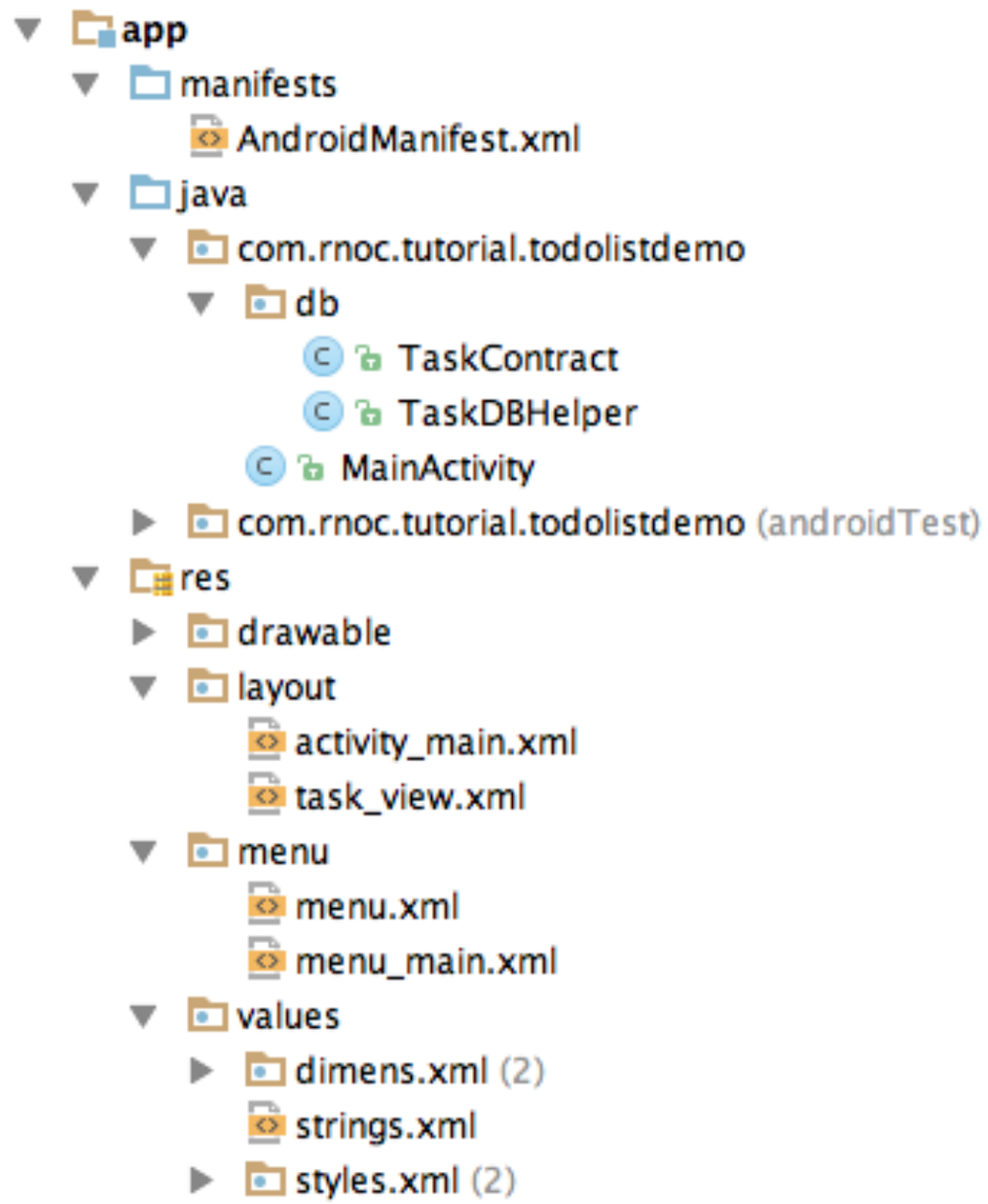
Component Tree



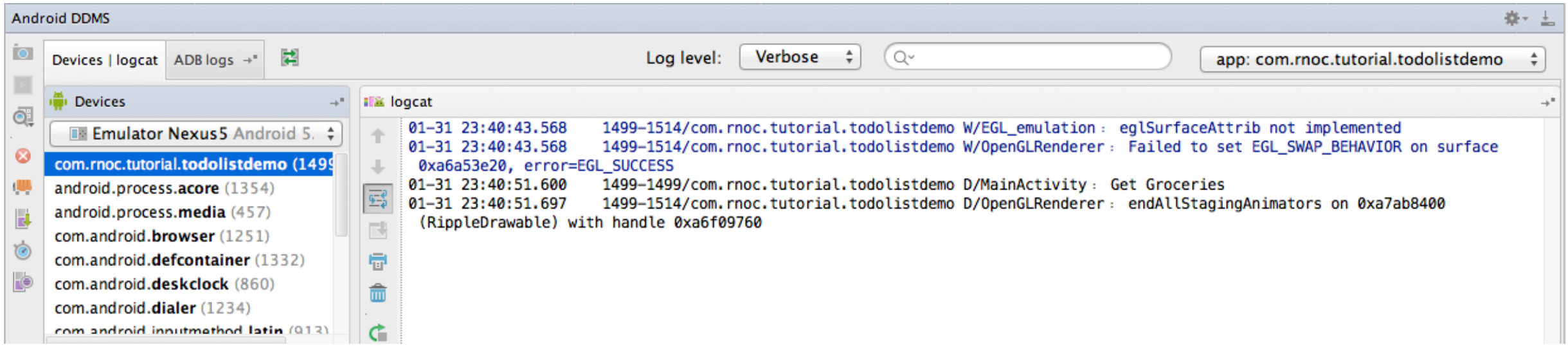
Toolbar



Project View



Logcat



Log.v(“Title”,”Message”); // verbose
Log.d(“Title”,”Message”); // debug
Log.i(“Title”,”Message”); // information
Log.w(“Title”,”Message”); // warning
Log.e(“Title”,”Message”); // error

“But isn’t Android just Java?”

Android MVC Architecture

Model

- Persistent Storage

SQLite

View

- Layout Files
- Graphical Layout Tool

Controller

- Activities

**While Android doesn't enforce MVC, this could be considered a "best practice" for development.

View

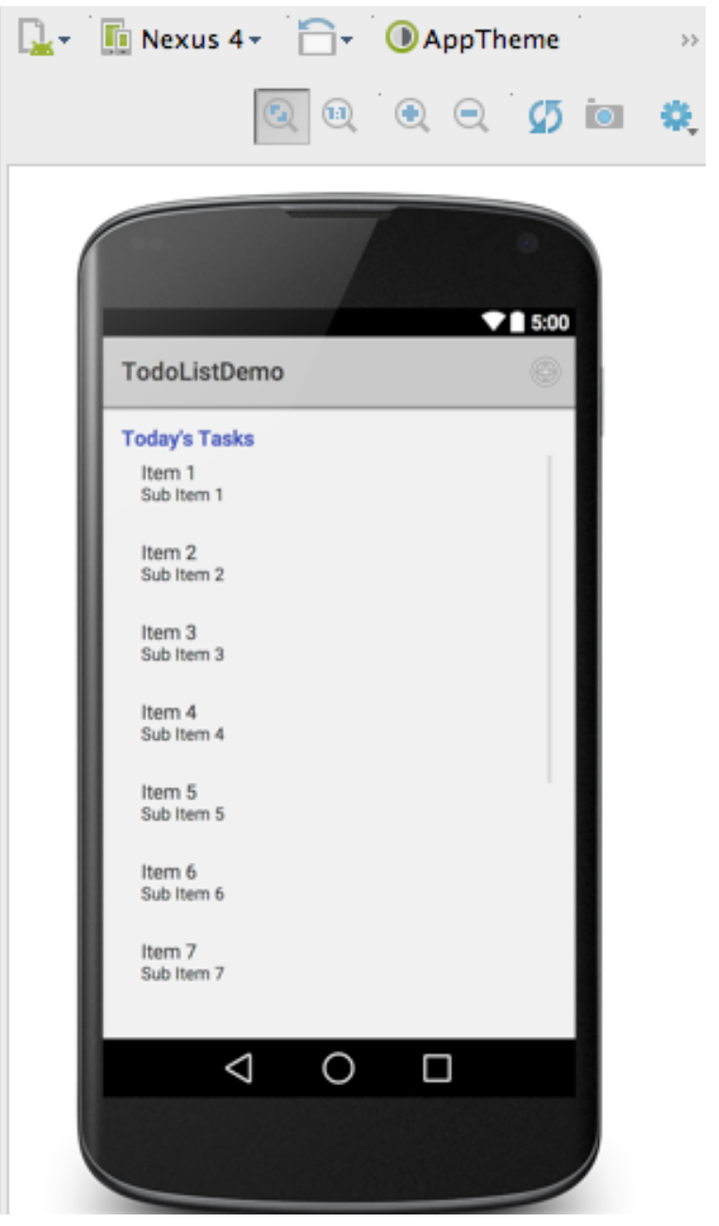
```xml
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity"
    android:id="@+id/mainAct">

    <!--@string/title_text :
        This links to the values/strings.xml file and pulls the value of the string from it.
        If the string value resource doesn't exist, or is not linked, it shows up as an error in red -->

    <TextView android:text="@string/title_text" android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/title_text"
        android:textColor="#ff2f3bac"
        android:textSize="17dp"
        android:textStyle="bold" />

    <ListView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@android:id/list"
        android:layout_weight="1"
        android:layout_below="@+id/title_text" />

</RelativeLayout>
```



Layout Types

Linear Layout

A layout that organizes its children into a single horizontal or vertical row. It creates a scrollbar if the length of the window exceeds the length of the screen.

Relative Layout

Enables you to specify the location of child objects relative to each other (child A to the left of child B) or to the parent (aligned to the top of the parent).

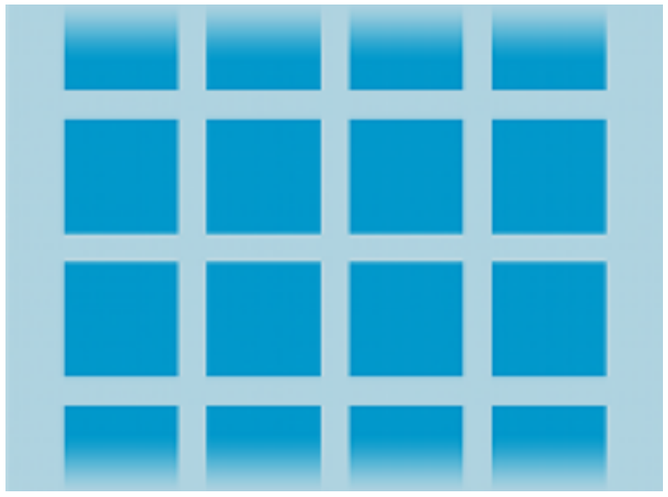
Web View

Displays web pages.

Layouts built with an Adapter

List View

Displays a scrolling single column list.



Grid View

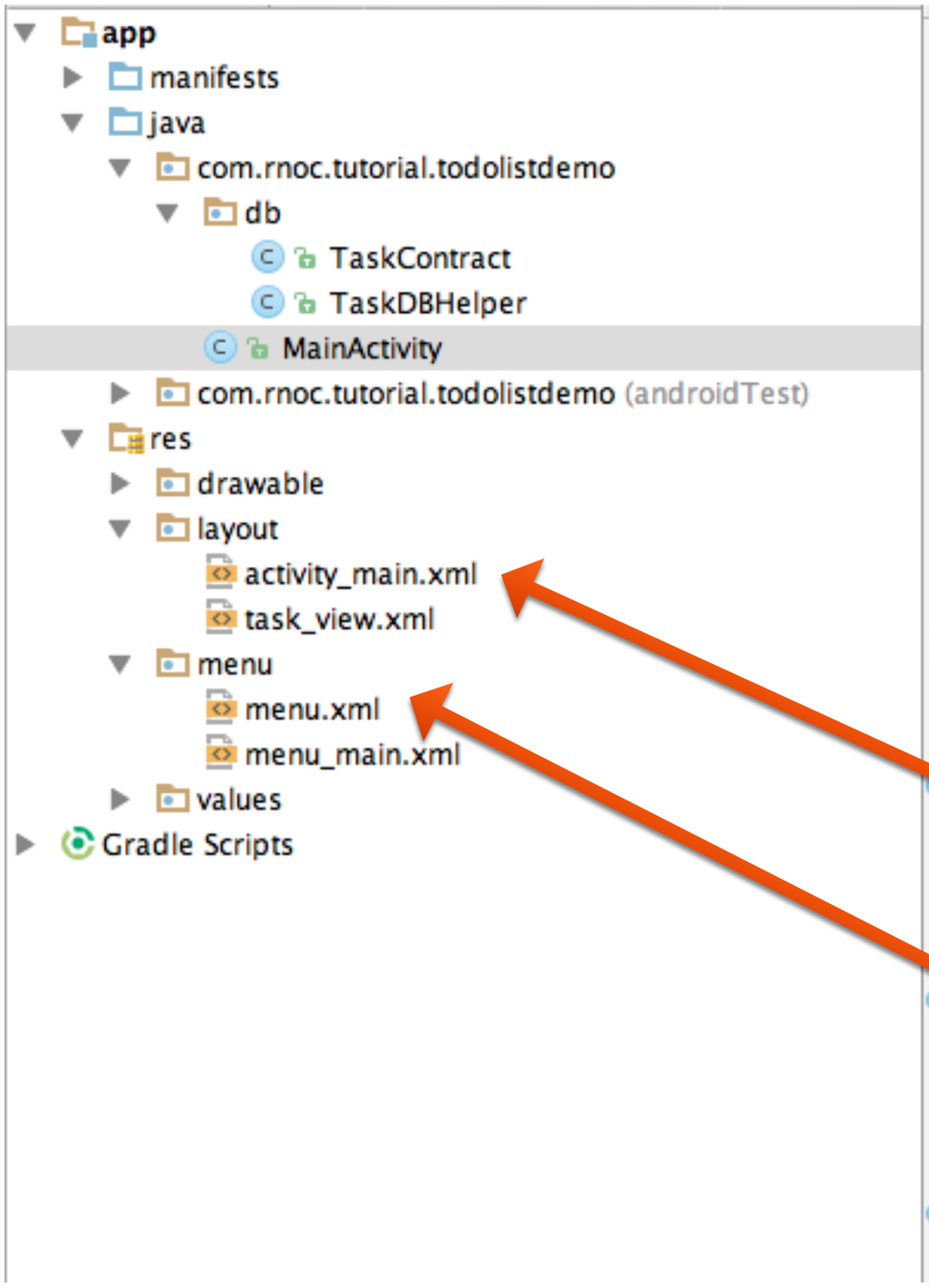
Displays a scrolling grid of columns and rows.

You can populate an AdapterView by binding the AdapterView instance to an Adapter, which retrieves data from an external source and creates a View that represents each data entry.

Two common adapters :

- ArrayAdapter
- SimpleCursorAdapter

View - Controller




```
package com.rnoc.tutorial.todolistdemo;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.ContentValues;
import android.content.DialogInterface;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.util.Log;
import android.widget.EditText;

import com.rnoc.tutorial.todolistdemo.db.TaskContract;
import com.rnoc.tutorial.todolistdemo.db.TaskDBHelper;


public class MainActivity extends Activity {
    private TaskDBHelper helper;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu,menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {...}
}
```

Model Options

Shared Preferences

- Store private primitive data in key-value pairs.

Internal Storage

- Store private data on the device memory.

External Storage

- Store public data on the shared external storage.

SQLite Databases

- Store structured data in a private database.

Network Connection

- Store data on the web with your own network server.

Data Backup Service

- Let users store and recover application and system data.

Syncing to the Cloud

- Different strategies for cloud enabled applications including backing up data

Model (SQLite)

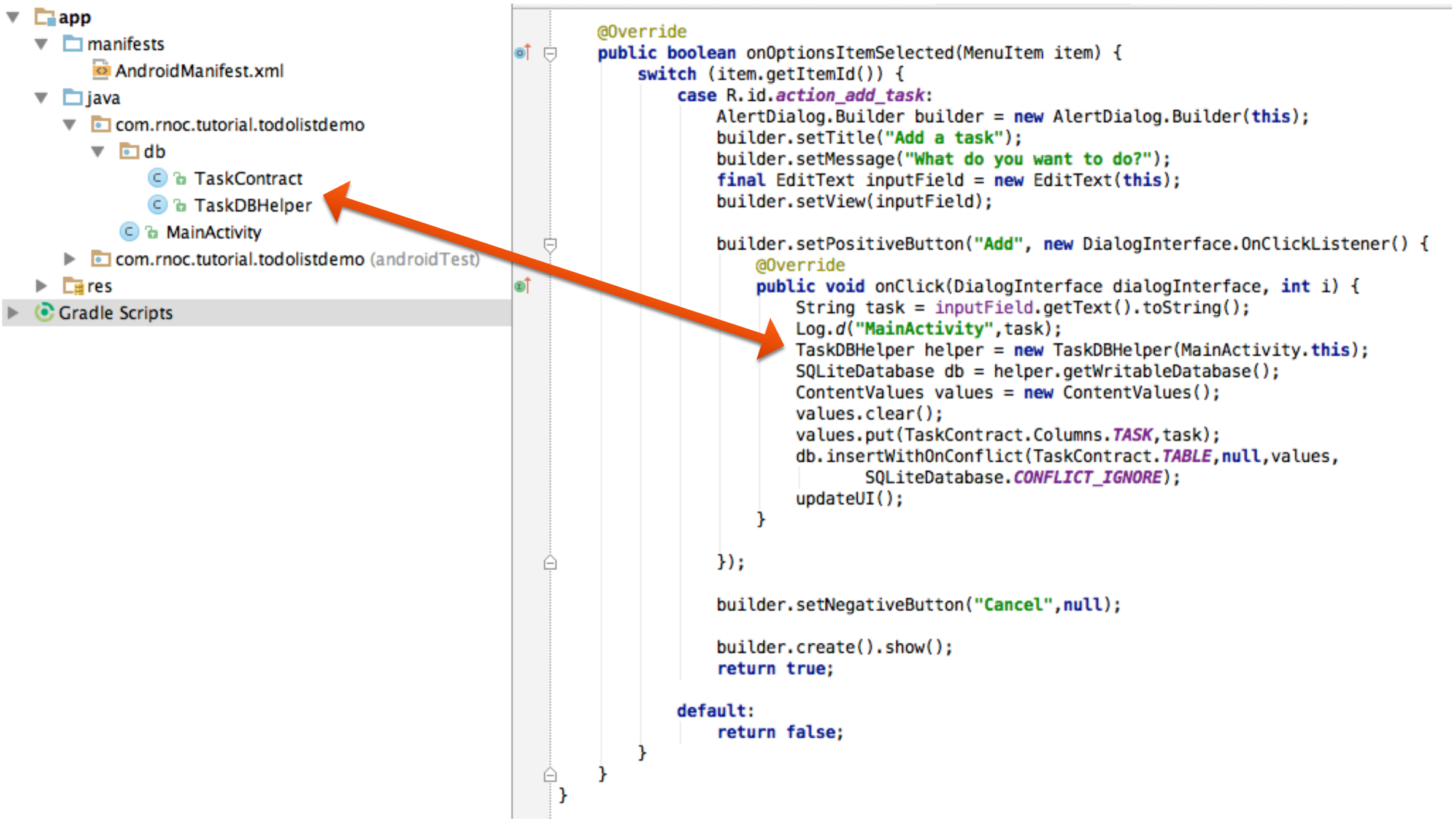
The Todo List Demo Database

```
public class TaskContract {
    public static final String DB_NAME = "com.example.TodoList.db.tasks";
    public static final int DB_VERSION = 1;
    public static final String TABLE = "tasks";

    public class Columns {
        public static final String TASK = "task";
        public static final String _ID = BaseColumns._ID;
    }
}
```

Find more Android SQLite Tutorials Here

Model - Controller



Android Manifest XML

- This file contains a lot of fundamental settings of your Android app like the name, the target android version, filters, libraries or permissions/ rights.

- All the elements that can appear in the manifest file are listed in alphabetical order. These are the only legal elements; you cannot add your own elements or attributes.

Android Manifest XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<!--Presents essential information about your application to the Android system. It's needed before the code can be run.-->

    <!--Names the Java package for your application. Serves as a unique identifier for the app-->
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.rnoc.tutorial.todolistdemo" >

    <!--Describes the components (activities, services used, broadcast receivers, etc) of your application-->
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="TodoListDemo"
        android:theme="@style/AppTheme" >

        <!--List of activities within the application-->

        <!--Name of the classes that implement the components with their capabilities-->
        <activity
            android:name="com.rnoc.tutorial.todolistdemo.MainActivity"
            android:label="@string/app_name" >

            <!-- An intent is a bundle of info that describes a desired action,
                    category of components that can perform this action,
                    the data that it will act on, etc -->

            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Android Components

Activity

- An activity represents the graphical user interface.
- It consists of buttons or other UI objects.
- The layout can be created in XML or dynamically in Java code

Services

- Services don’t have a user interface
- They are used for working on tasks in the background.
- For instance, downloading some files, or playing music

ContentProvider

- This is used to exchange information between apps
- For instance, the Address Book is a ContentProvider
- Other applications may request information from the Address Book

Android Components

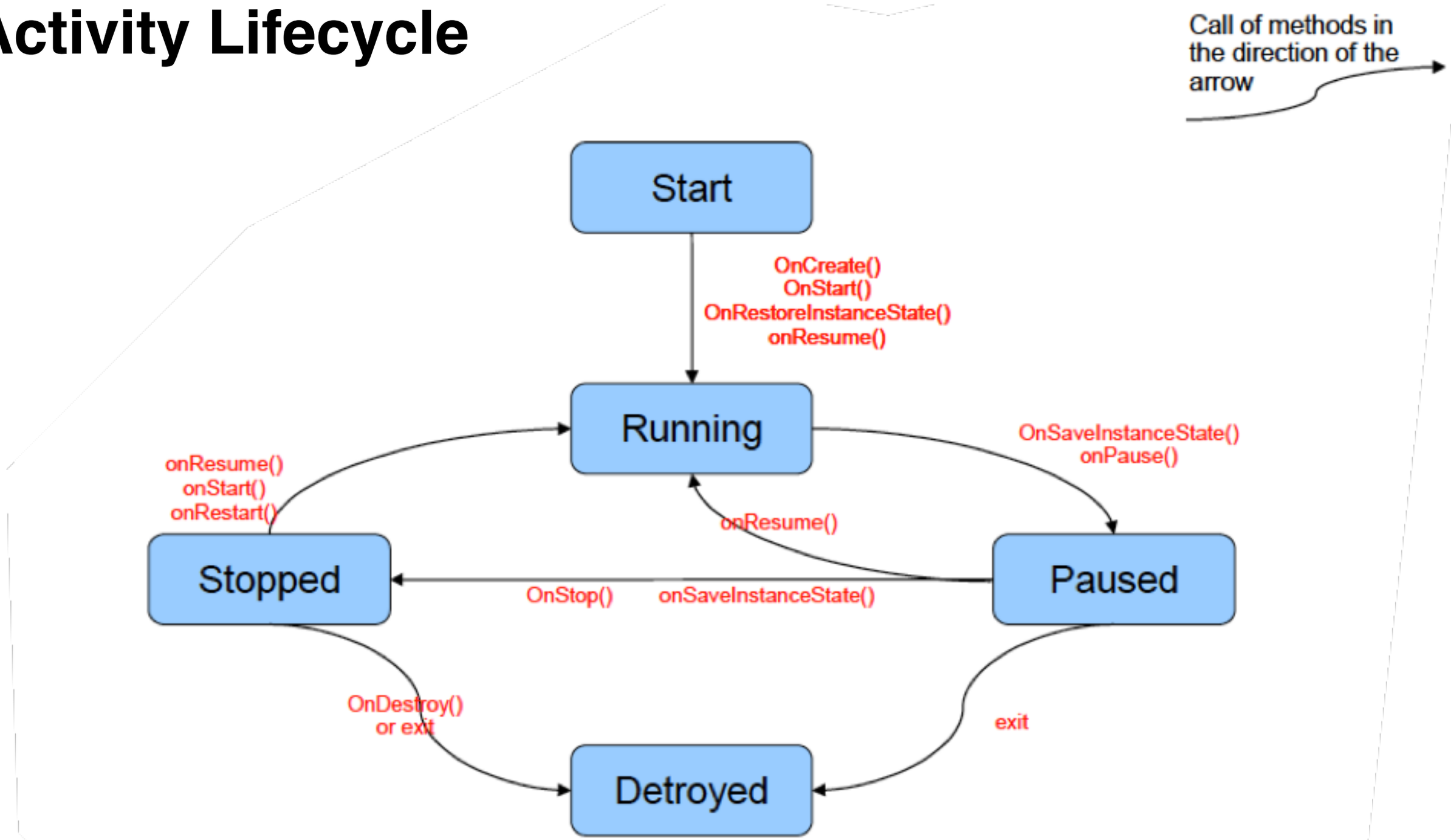
Broadcast Receiver

- Used to receive system events and to react to them
- There are two types : Static Broadcast Receivers (declared in the manifest file) and Dynamic Broadcast Receivers (created during runtime)
- Example events received are booting up of applications, or a headset being plugged in.

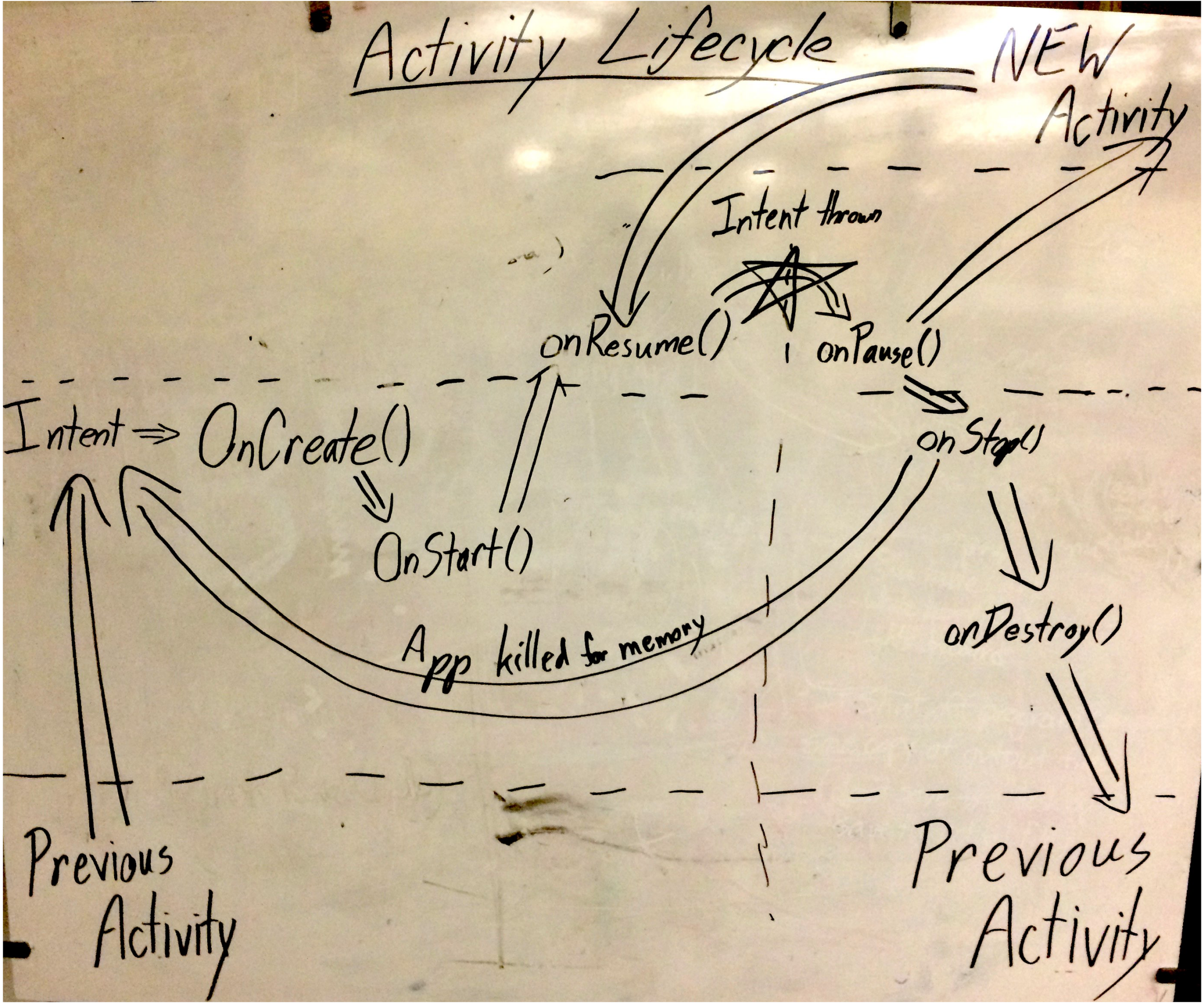
Intents

- Briefly describe what should be done
- There are two types of intents : Explicit Intents (the receiver is known), and Implicit Intents (receiver unknown)
- For Implicit Intents the system decides what action needs to be taken.
- For instance, you could transfer viewing to another app to, say, open a PDF file.

Activity Lifecycle



Activity Lifecycle



Sample Intents

View Webpage

Intent viewIntent = new Intent(“android.intent.action.View”, Uri.parse(“http://www.google.com”))

Start Activity (Explicit)

Intent i = new Intent(Intent.ACTION_DIAL(Uri.parse(“tel:(404)123-4567”));

Show Position on Map

Uri uri = Uri.parse(“geo:54.1234,52.1234?z=22”)
Intent i = new Intent(Intent.ACTION_VIEW);
i.setData(uri)
startActivity(i)

Other Tools : BlueStacks

- BlueStacks App Player lets you run Android apps fast and fullscreen on Windows and Mac.

Demo