CS 528 Mobile and Ubiquitous Computing

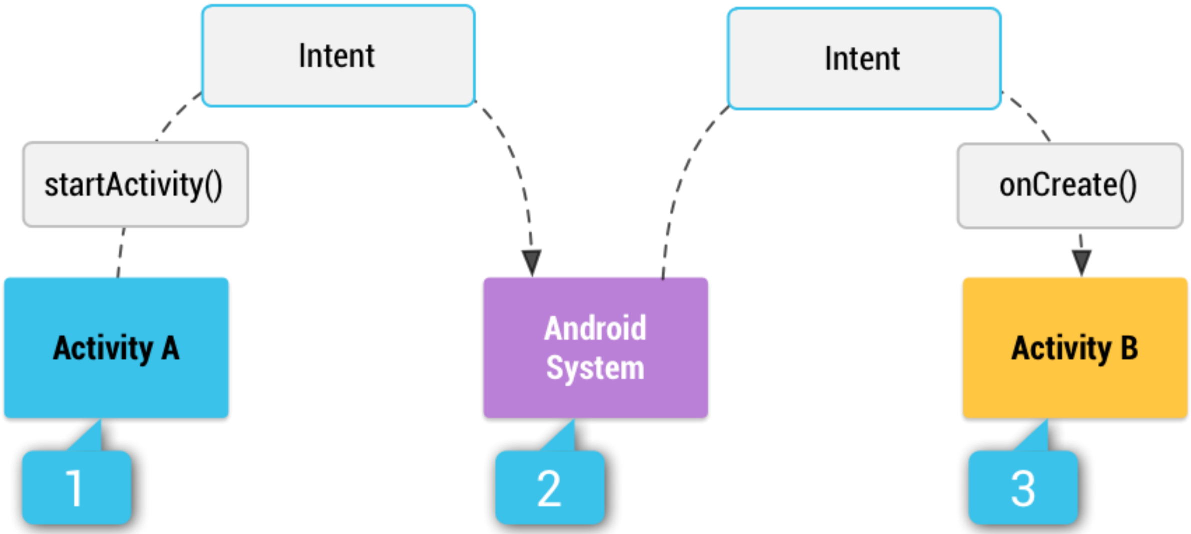
Lecture 3b: Intents, Fragments, Database and Camera

Emmanuel Agu

Intents

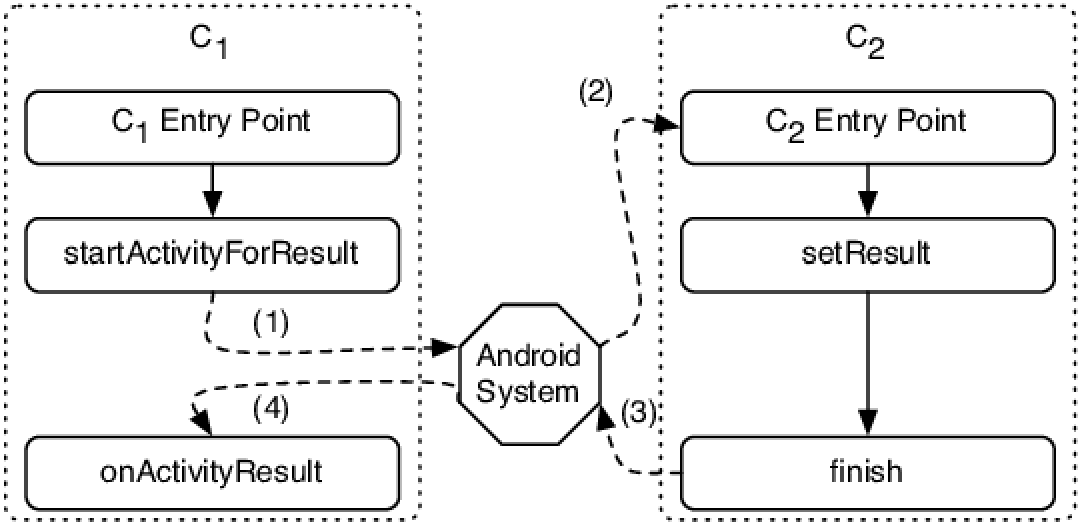
Intent

- **Intent:** a messaging object used by a component to request action from another app or component
- 3 main use cases for Intents
- **Case 1 (Activity A starts Activity B, no result back):**
  - Call **startActivity( )**, pass an Intent
  - Intent has information about Activity to start, plus any necessary data



Intent: Result Received Back

- **Case 2 (Activity A starts Activity B, gets result back):**
  - Call **startActivityForResult( )**, pass an Intent
  - Separate Intent received in Activity A's **onActivityResult( )** callback



Intent: Result Received Back

- **Case 3 (Activity A starts a Service):**
  - E.g. Activity A starts service to download big file in the background
  - Activity A calls **StartService( )**, passes an Intent
  - Intent contains information about Service to start, plus any necessary data

Implicit Vs Explicit Intents

- **Explicit Intent:** If components sending and receiving Intent are in same app
  - E.g. Activity A starts Activity B in same app
  - Activity A explicitly says what Activity (B) should be started

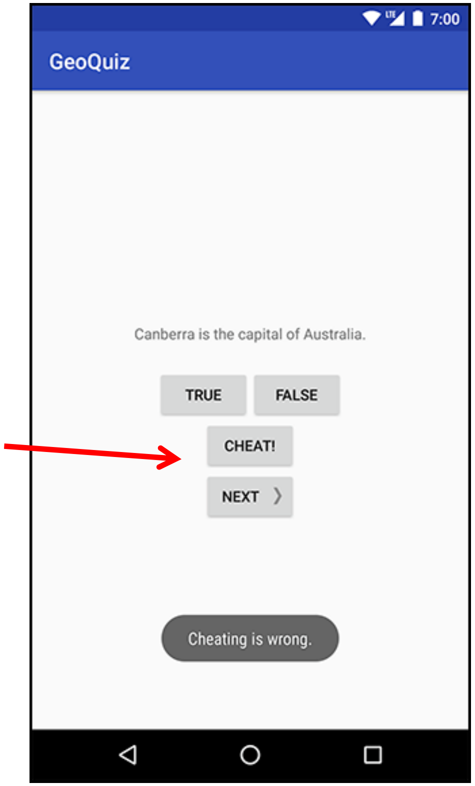
- **Implicit Intent:** If components sending and receiving Intent are in **different apps**
  - Activity B specifies what ACTION it needs done, doesn’t specify Activity to do it
  - Example of Action: take a picture, any camera app can handle this

Intent Example: Starting Activity 2 from Activity 1

Allowing User to Cheat
Ref: Android Nerd Ranch (3rd edition) pg 91

- **Goal:** Allow user to cheat by getting answer to quiz
- Screen 2 pops up to show Answer

Activity 1



Canberra is the capital of Australia.

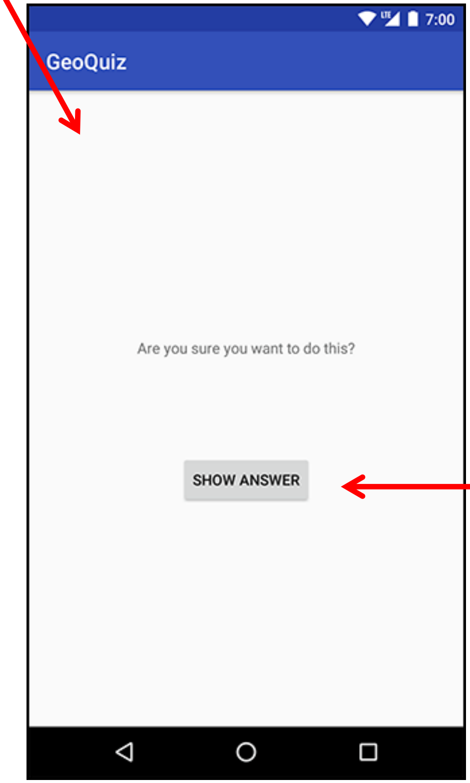
Cheating is wrong.

User clicks here to cheat

Correct Answer

If user cheated

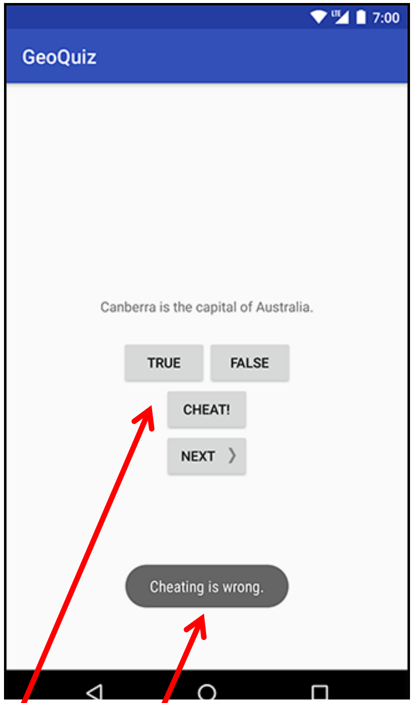
Activity 2



Are you sure you want to do this?

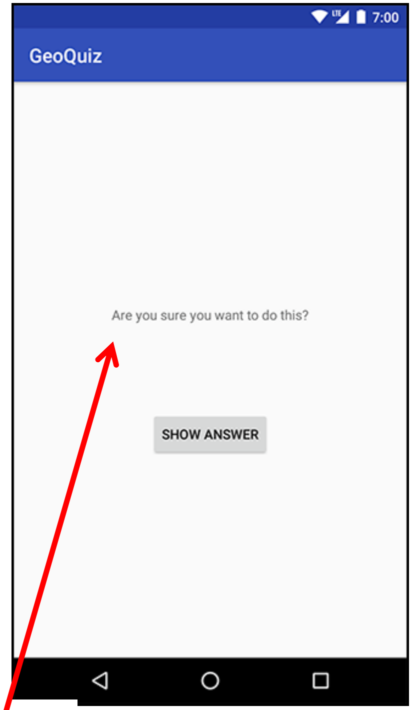
Ask again. Click here to cheat

Add Strings for Activity 1 and Activity 2 to strings.xml



Canberra is the capital of Australia.

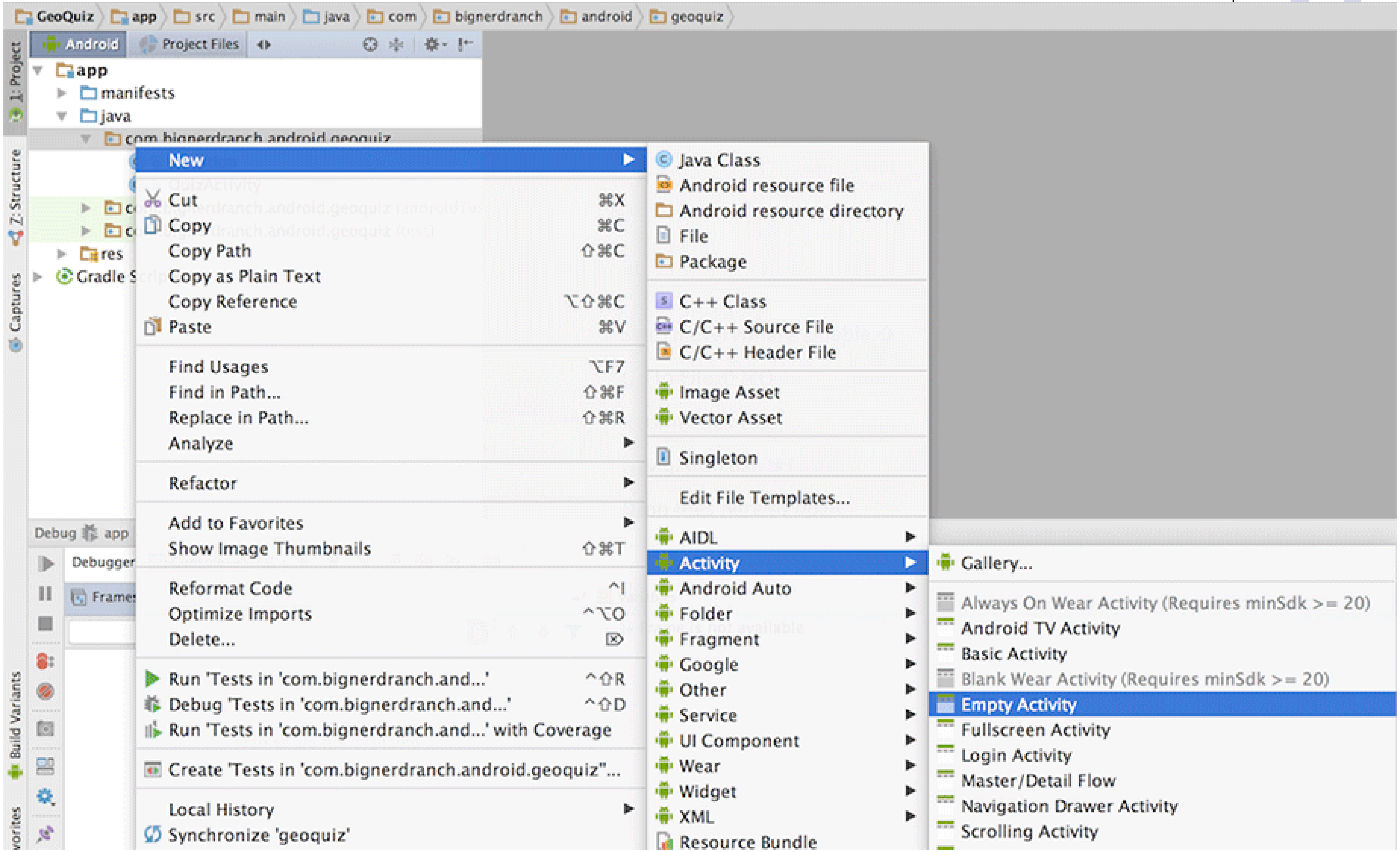
Cheating is wrong.



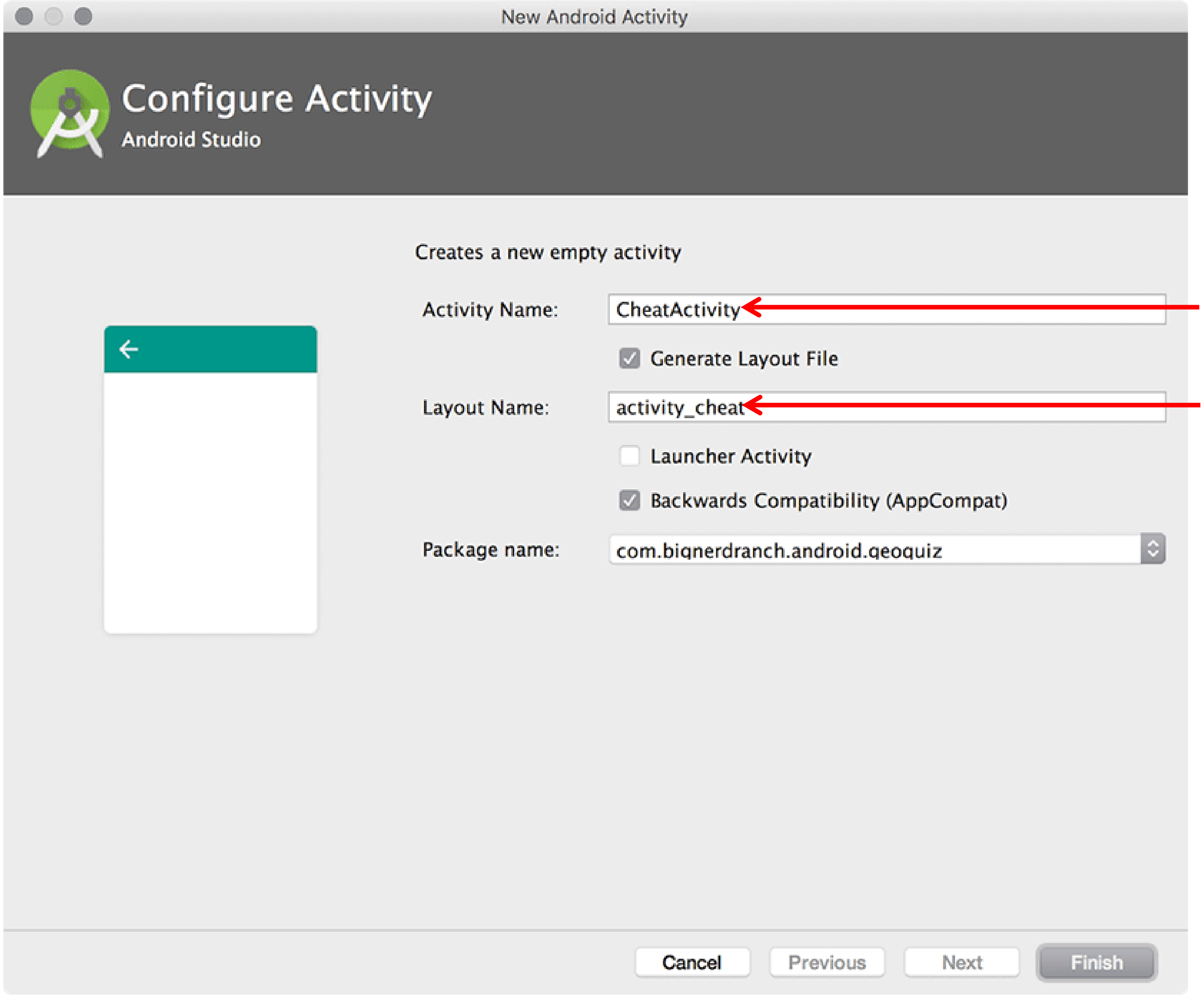
Are you sure you want to do this?

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    ...
    <string name="question_asia">Lake Baikal is the world\'s oldest and
deepest
      freshwater lake.</string>
    <string name="warning_text">Are you sure you want to do this?</string>
    <string name="show_answer_button">Show Answer</string>
    <string name="cheat_button">Cheat!</string>
    <string name="judgment_toast">Cheating is wrong.</string>

</resources>
```

Create Empty Activity (for Activity 2) in Android Studio



Specify Name and XML file for Activity 2



Creates a new empty activity

Activity Name: CheatActivity

☑ Generate Layout File

Layout Name: activity_cheat

□ Launcher Activity

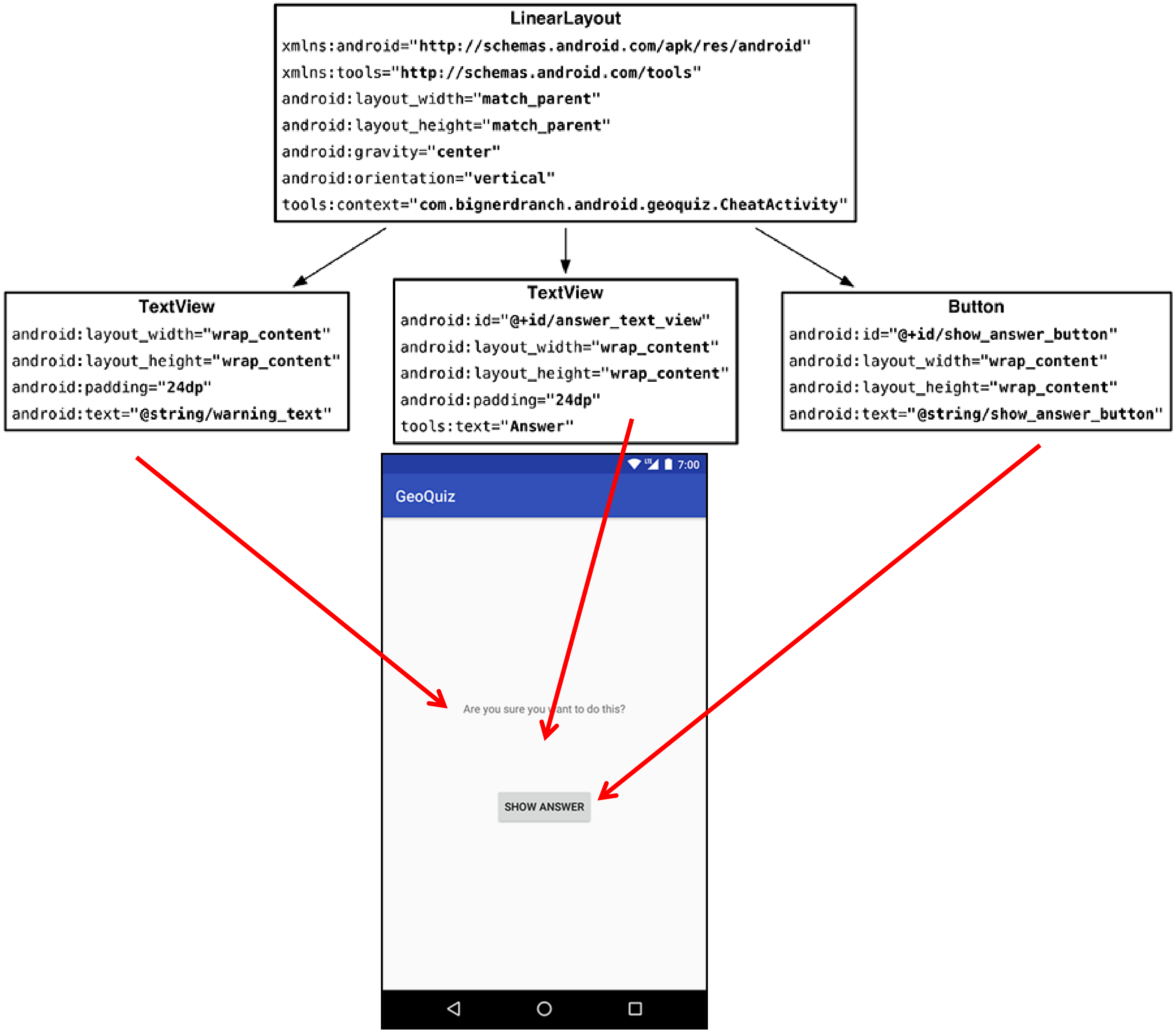
☑ Backwards Compatibility (AppCompat)

Package name: com.bignerdranch.android.geoquiz

Screen 2 Java code in CheatActivity.java

Layout uses activity_cheat.xml

Design Layout for Screen 2



LinearLayout

```
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:gravity="center"
android:orientation="vertical"
tools:context="com.bignerdranch.android.geoquiz.CheatActivity"
```

TextView

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:padding="24dp"
android:text="@string/warning_text"
```

TextView

```
android:id="@+id/answer_text_view"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:padding="24dp"
tools:text="Answer"
```

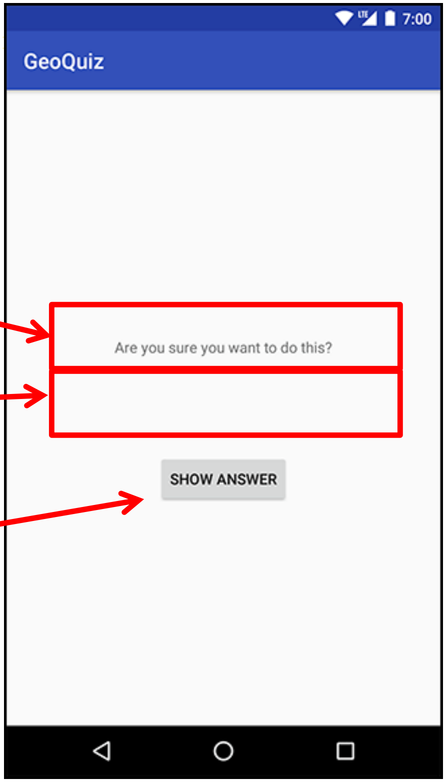
Button

```
android:id="@+id/show_answer_button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/show_answer_button"
```

Write XML Layout Code for Screen 2

```
<LinearLayout 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:orientation="vertical"
              android:gravity="center"
              tools:context="com.bignerdranch.android.geoquiz.CheatActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="24dp"
        android:text="@string/warning_text"/>

    <TextView
        android:id="@+id/answer_text_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="24dp"
        tools:text="Answer"/>

    <Button
        android:id="@+id/show_answer_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/show_answer_button"/>

</LinearLayout>
```

Activity 2



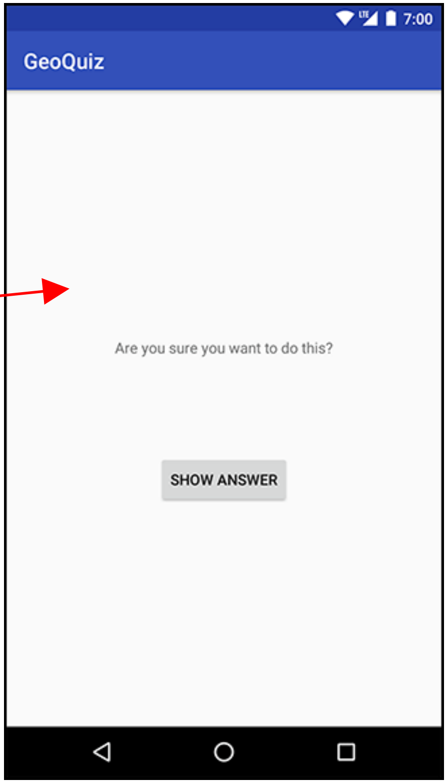
Declare New Activity (CheatActivity) in AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.bignerdranch.android.geoquiz" >

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <activity android:name=".QuizActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>

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

</manifest>
```

Activity 1

Activity 2 (CheatActivity)

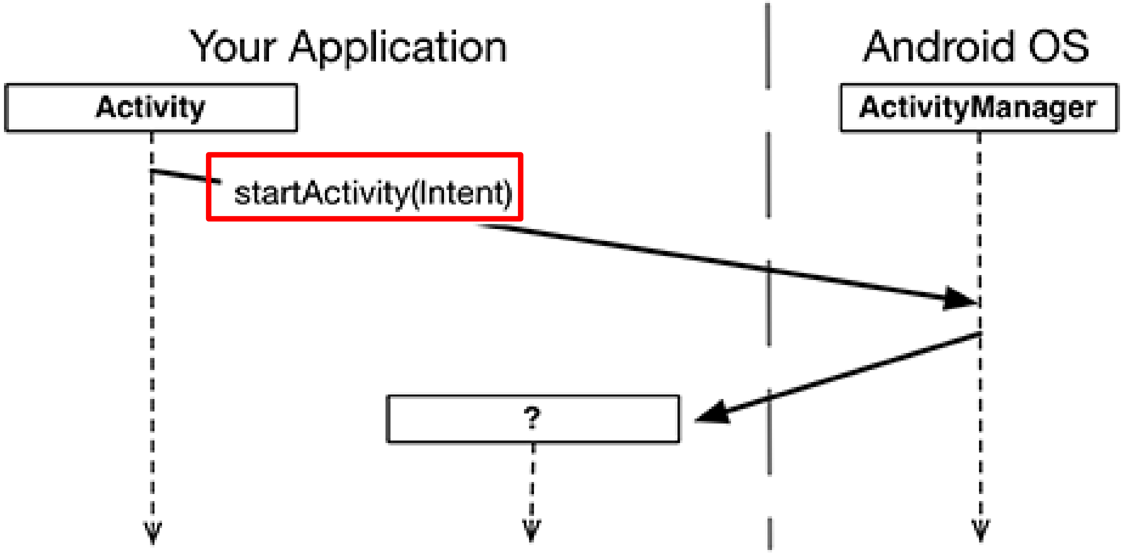
Activity 2 (CheatActivity)



Are you sure you want to do this?

Starting Activity 2 from Activity 1

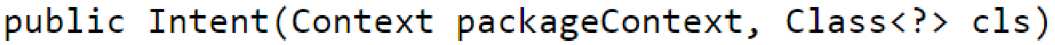
- Activity 1 starts activity 2
  - **through** the Android OS
  - by calling **startActivity(Intent)**
- Passes Intent (object for communicating with Android OS)



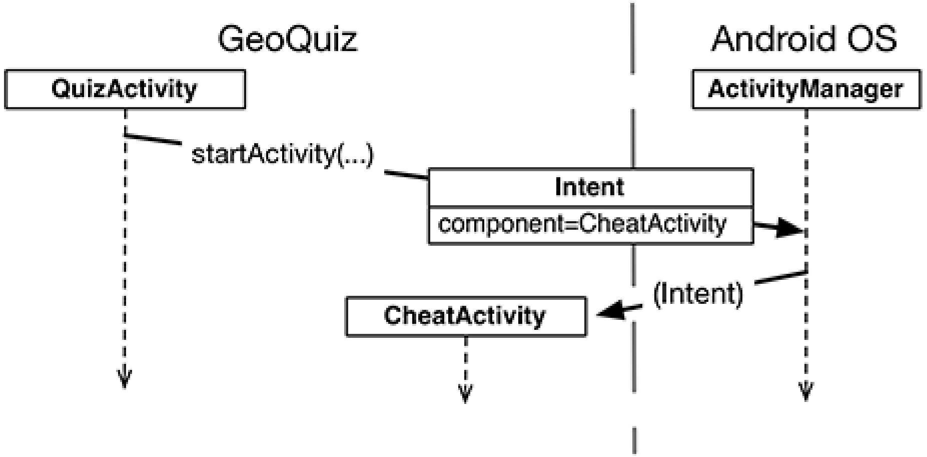
- Intent specifies which (target) Activity Android ActivityManager should start

Starting Activity 2 from Activity 1

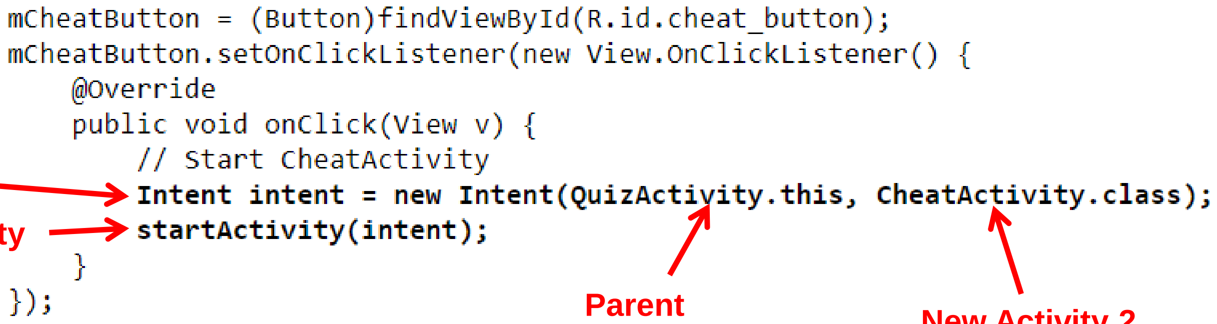
- Intents have many different constructors. We will use form:



```
public Intent(Context packageContext, Class<?> cls)
```



- Actual code looks like this



```
mCheatButton = (Button)findViewById(R.id.cheat_button);
mCheatButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Start CheatActivity
        Intent intent = new Intent(QuizActivity.this, CheatActivity.class);
        startActivity(intent);
    }
});
```

Build Intent

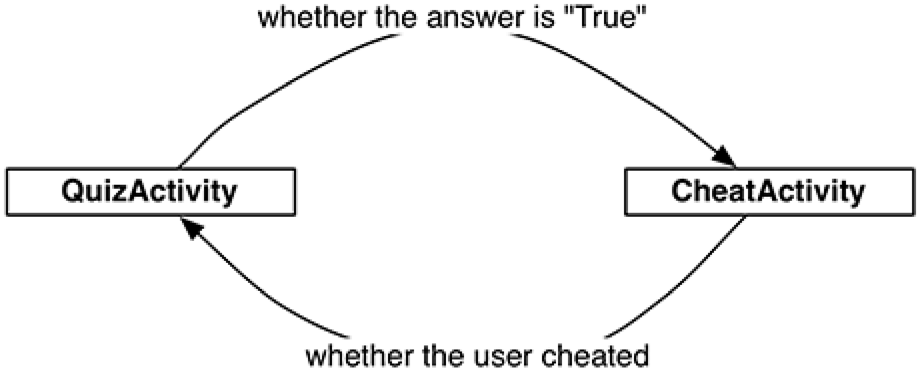
Use Intent to Start new Activity

Parent Activity

New Activity 2

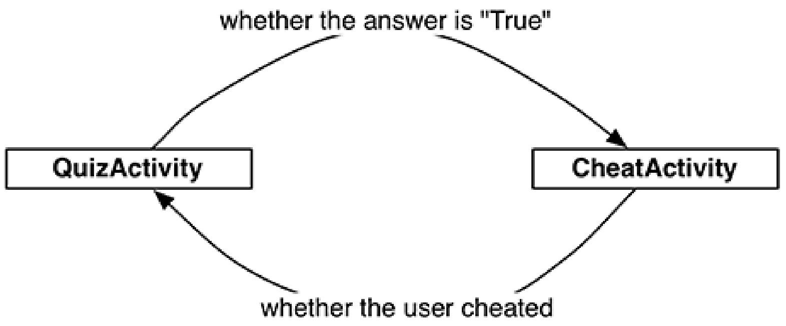
Implicit vs Explicit Intents

- Previous example is called an **explicit intent**
  - Activity 1 and activity 2 are in same app
- If Activity 2 were in another app, an **implicit intent** would have to be created instead
- Can also pass data between Activities 1 and 2
  - E.g. Activity 1 can tell Activity 2 correct answer (True/False)

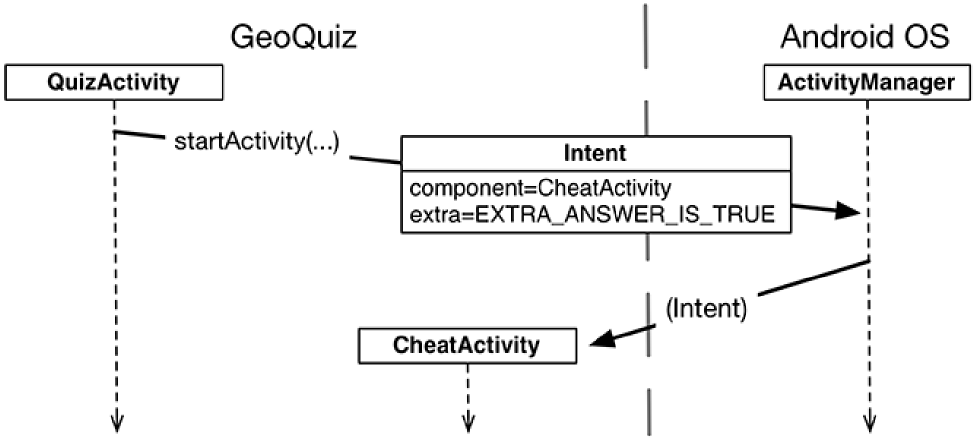


Passing Data Between Activities

- Need to pass answer (True/False from QuizActivity to CheatActivity)



- Pass answer as **extra** on the Intent passed into **StartActivity**
- **Extras** are arbitrary data calling activity can include with intent



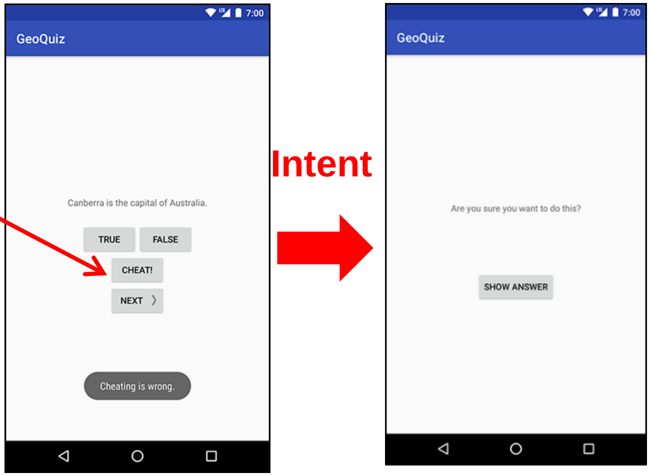
Passing Answer (True/False) as Intent Extra

- To add **extra** to Intent, use **putExtra( )** command
- Encapsulate Intent creation into a method **newIntent( )**

```
public class CheatActivity extends AppCompatActivity {

    private static final String EXTRA_ANSWER_IS_TRUE =
            "com.bignerdranch.android.geoquiz.answer_is_true";

    public static Intent newIntent(Context packageContext, boolean answerIsTrue) {
        Intent intent = new Intent(packageContext, CheatActivity.class);
        intent.putExtra(EXTRA_ANSWER_IS_TRUE, answerIsTrue);
        return intent;
    }
    ...
```

- When user clicks cheat button, build Intent, start new Activity

```
mCheatButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Start CheatActivity
        ~~Intent intent = new Intent(QuizActivity.this, CheatActivity.class);~~
        boolean answerIsTrue = mQuestionBank[mCurrentIndex].isAnswerTrue();
        Intent intent = CheatActivity.newIntent(QuizActivity.this, answerIsTrue);
        startActivity(intent);
    }
});
```



Canberra is the capital of Australia.

Cheating is wrong.

Are you sure you want to do this?

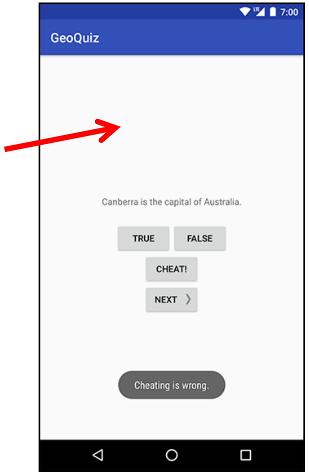
Passing Answer (True/False) as Intent Extra

- Activity receiving the Intent retrieves it using **getBooleanExtra( )**

```
public class CheatActivity extends AppCompatActivity {

    private static final String EXTRA_ANSWER_IS_TRUE =
            "com.bignerdranch.android.geoquiz.answer_is_true";

    private boolean mAnswerIsTrue;
    ...
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_cheat);

        mAnswerIsTrue = getIntent().getBooleanExtra(EXTRA_ANSWER_IS_TRUE, false)
    }
    ...
}
```

Calls getIntent( )

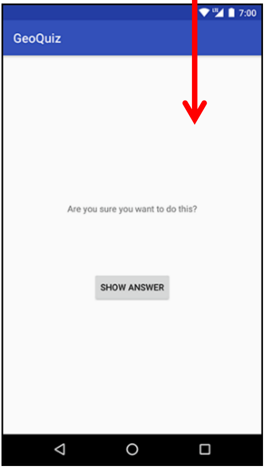
Calls startActivity(Intent)



Canberra is the capital of Australia.

Cheating is wrong.

Intent (Answer = Extra)

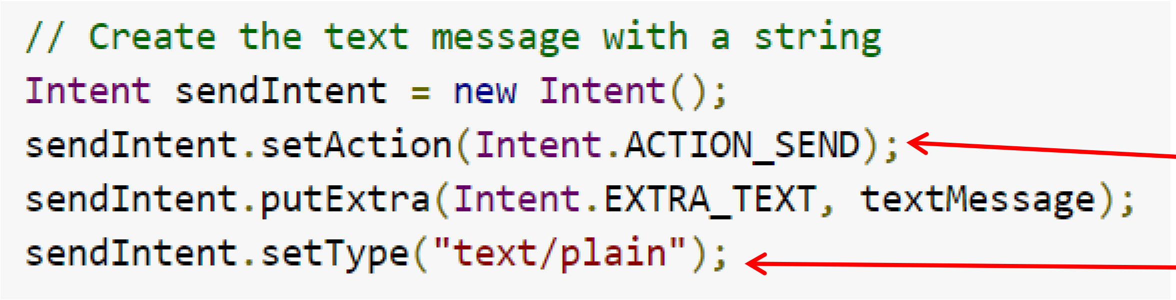


Are you sure you want to do this?

Important: Read Android Nerd Ranch (3rd edition) pg 91

Implicit Intents

- **Implicit Intent:** Does not name component to start.
- Specifies
  - **Action** (what to do, example visit a web page)
  - **Data** (to perform operation on, e.g. web page url)
- Typically, many components (apps) can take a given action
  - E.g. Many phones have installed multiple apps that can view images
- System decides component to receive intent based on **action, data, category**
- Example Implicit Intent to share data



```
// Create the text message with a string
Intent sendIntent = new Intent();
sendIntent.setAction(Intent.ACTION_SEND);
sendIntent.putExtra(Intent.EXTRA_TEXT, textMessage);
sendIntent.setType("text/plain");
```

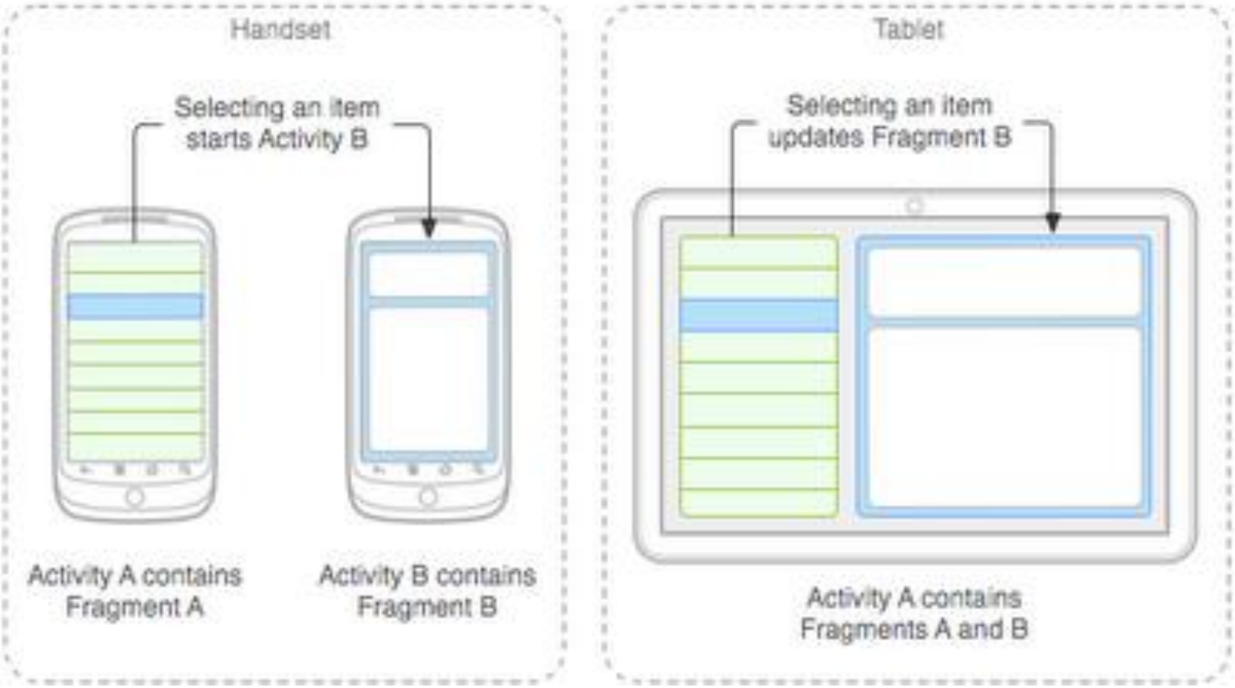
ACTION (No receiving Activity specified)

Data type

Fragments

Recall: Fragments

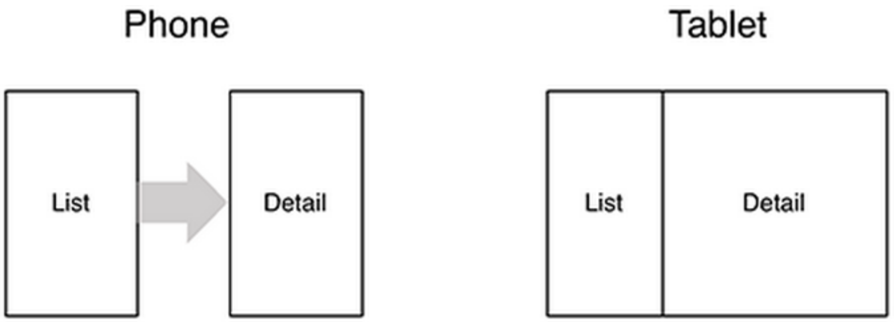
- Sub-components of an Activity (screen)
- An activity can contain multiple fragments, organized differently on different devices (e.g. phone vs tablet)
- Fragments need to be attached to Activities.

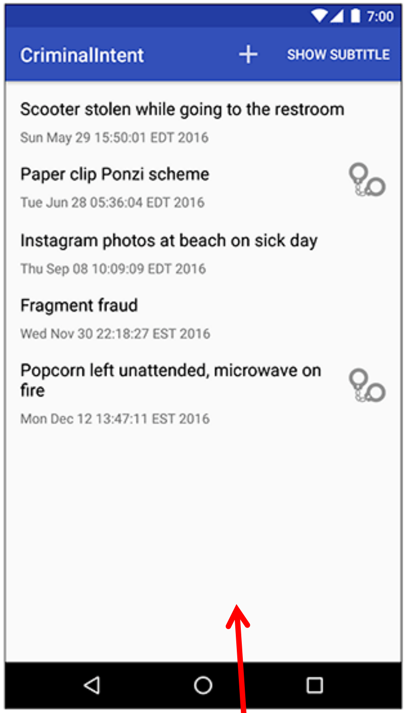

Handset
Selecting an item starts Activity B
Activity A contains Fragment A
Activity B contains Fragment B
Tablet
Selecting an item updates Fragment B
Activity A contains Fragments A and B

Fragments

Ref: Android Nerd Ranch (3rd ed), Ch 7, pg 123

- To illustrate fragments, we create new app **CriminalIntent**
- Used to record "office crimes" e.g. leaving plates in sink, etc
- Crime record includes:
  - Title, date, photo
- List-detail app using fragments





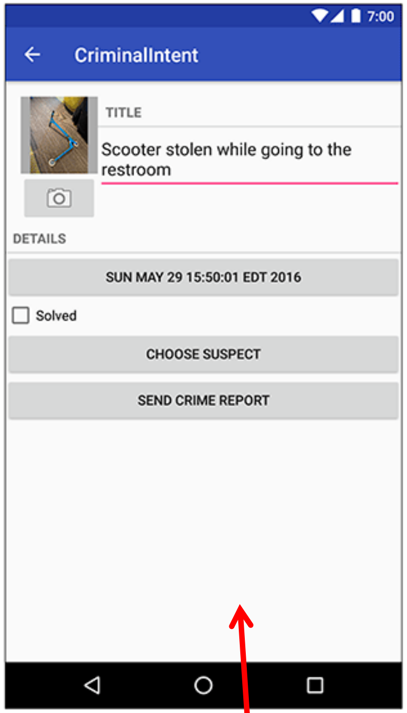
Scooter stolen while going to the restroom
Sun May 29 15:50:01 EDT 2016

Paper clip Ponzi scheme
Tue Jun 28 05:36:04 EDT 2016

Instagram photos at beach on sick day
Thu Sep 08 10:09:09 EDT 2016

Fragment fraud
Wed Nov 30 22:18:27 EST 2016

Popcorn left unattended, microwave on fire
Mon Dec 12 13:47:11 EST 2016



Scooter stolen while going to the restroom

□ Solved

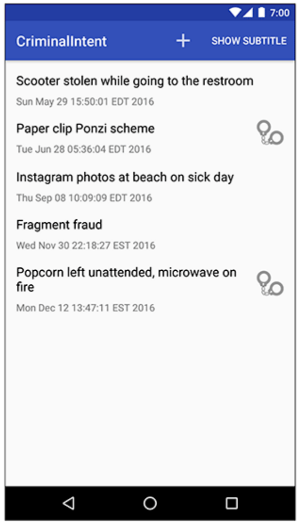
- **On tablet:** show list + detail
- **On phone:** swipe to show next crime

Fragment 1
(list of Crimes)

Fragment 2
(Details of selected Crime)

Fragments

- Activities can contain multiple fragments
- Fragment's views are inflated from a layout file
- Can rearrange fragments as desired on an activity
    - i.e. different arrangement on phone vs tablet



7:00

CriminalIntent + SHOW SUBTITLE

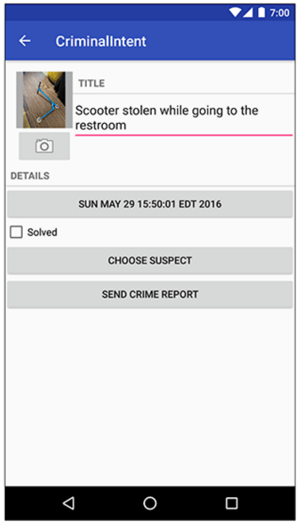
Scooter stolen while going to the restroom
Sun May 29 15:50:01 EDT 2016

Paper clip Ponzi scheme
Tue Jun 28 05:36:04 EDT 2016

Instagram photos at beach on sick day
Thu Sep 08 10:09:09 EDT 2016

Fragment fraud
Wed Nov 30 22:18:27 EST 2016

Popcorn left unattended, microwave on fire
Mon Dec 12 13:47:11 EST 2016



7:00

CriminalIntent

TITLE

Scooter stolen while going to the restroom

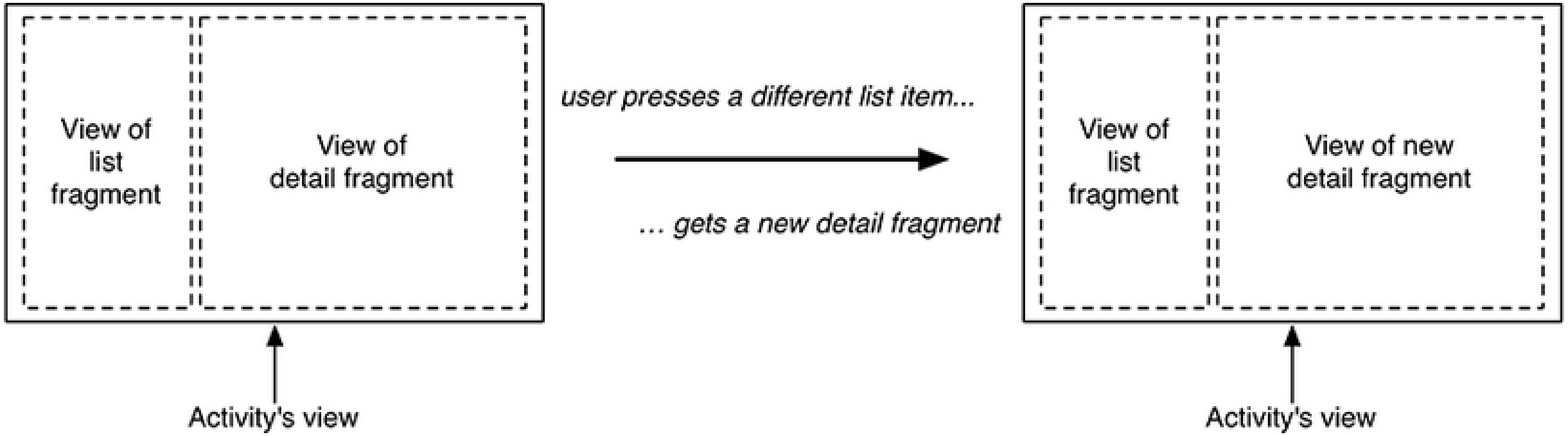
DETAILS

SUN MAY 29 15:50:01 EDT 2016

□ Solved

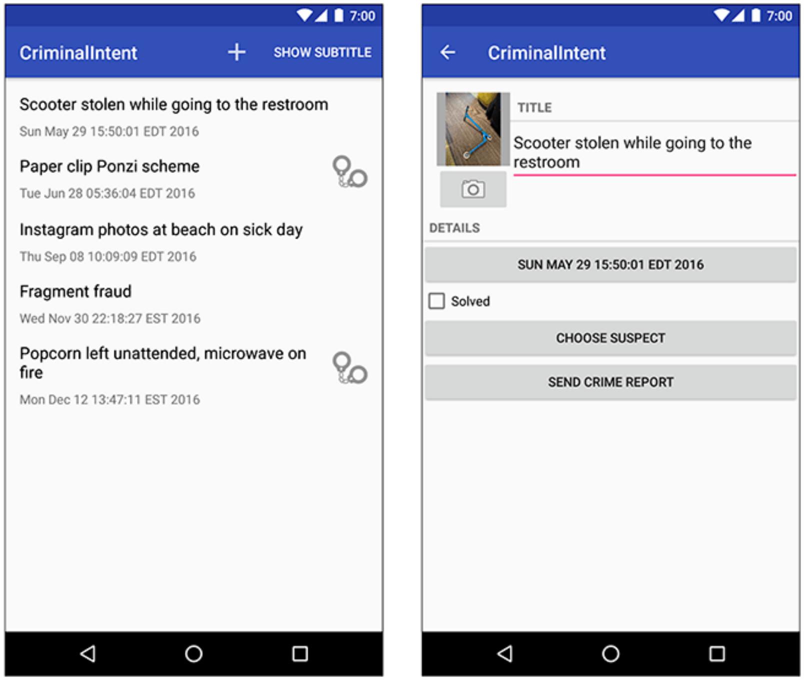
CHOOSE SUSPECT

SEND CRIME REPORT



Starting Criminal Intent

- Initially, develop detail view of **CriminalIntent** using Fragments



7:00

CriminalIntent + SHOW SUBTITLE

Scooter stolen while going to the restroom
Sun May 29 15:50:01 EDT 2016

Paper clip Ponzi scheme
Tue Jun 28 05:36:04 EDT 2016

Instagram photos at beach on sick day
Thu Sep 08 10:09:09 EDT 2016

Fragment fraud
Wed Nov 30 22:18:27 EST 2016

Popcorn left unattended, microwave on fire
Mon Dec 12 13:47:11 EST 2016

7:00

CriminalIntent

TITLE

Scooter stolen while going to the restroom

DETAILS

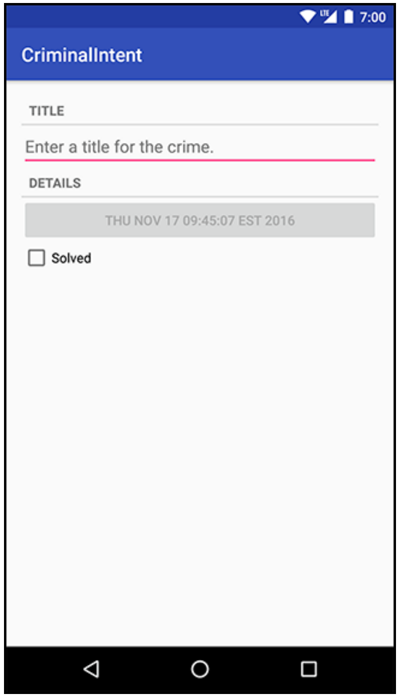
SUN MAY 29 15:50:01 EDT 2016

□ Solved

CHOOSE SUSPECT

SEND CRIME REPORT

Final Look of CriminalIntent



7:00

CriminalIntent

TITLE

Enter a title for the crime.

DETAILS

THU NOV 17 09:45:07 EST 2016

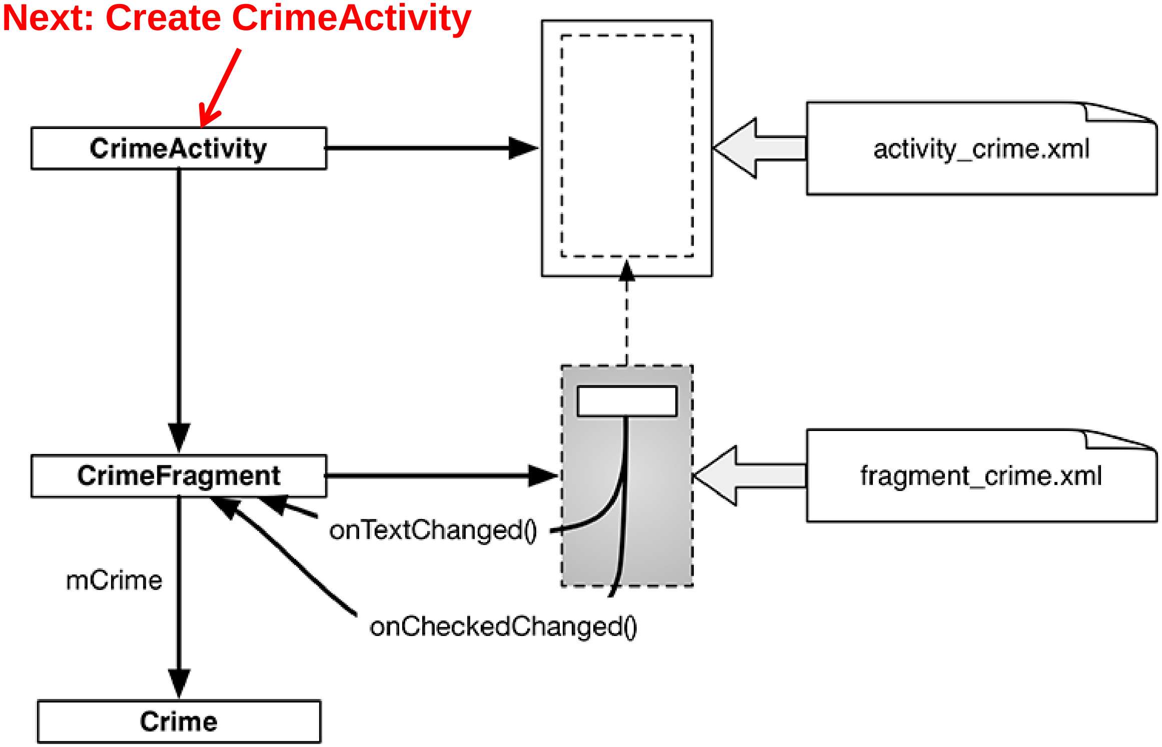
□ Solved

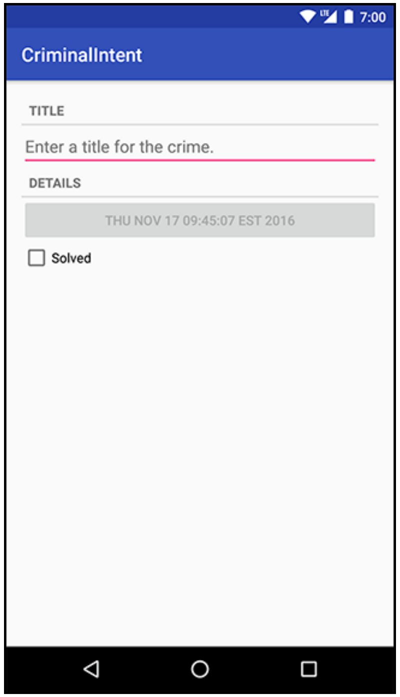
Start small
Develop detail view using Fragments

Starting Criminal Intent

- **Crime:** holds record of 1 office crime. Has
  - **Title** e.g. “Someone stole my yogurt!”
  - **ID:** unique identifier of crime
- **CrimeFragment:** UI fragment to display Crime Details
- **CrimeActivity:** Activity that contains **CrimeFragment**

Next: Create CrimeActivity





CriminalIntent

TITLE

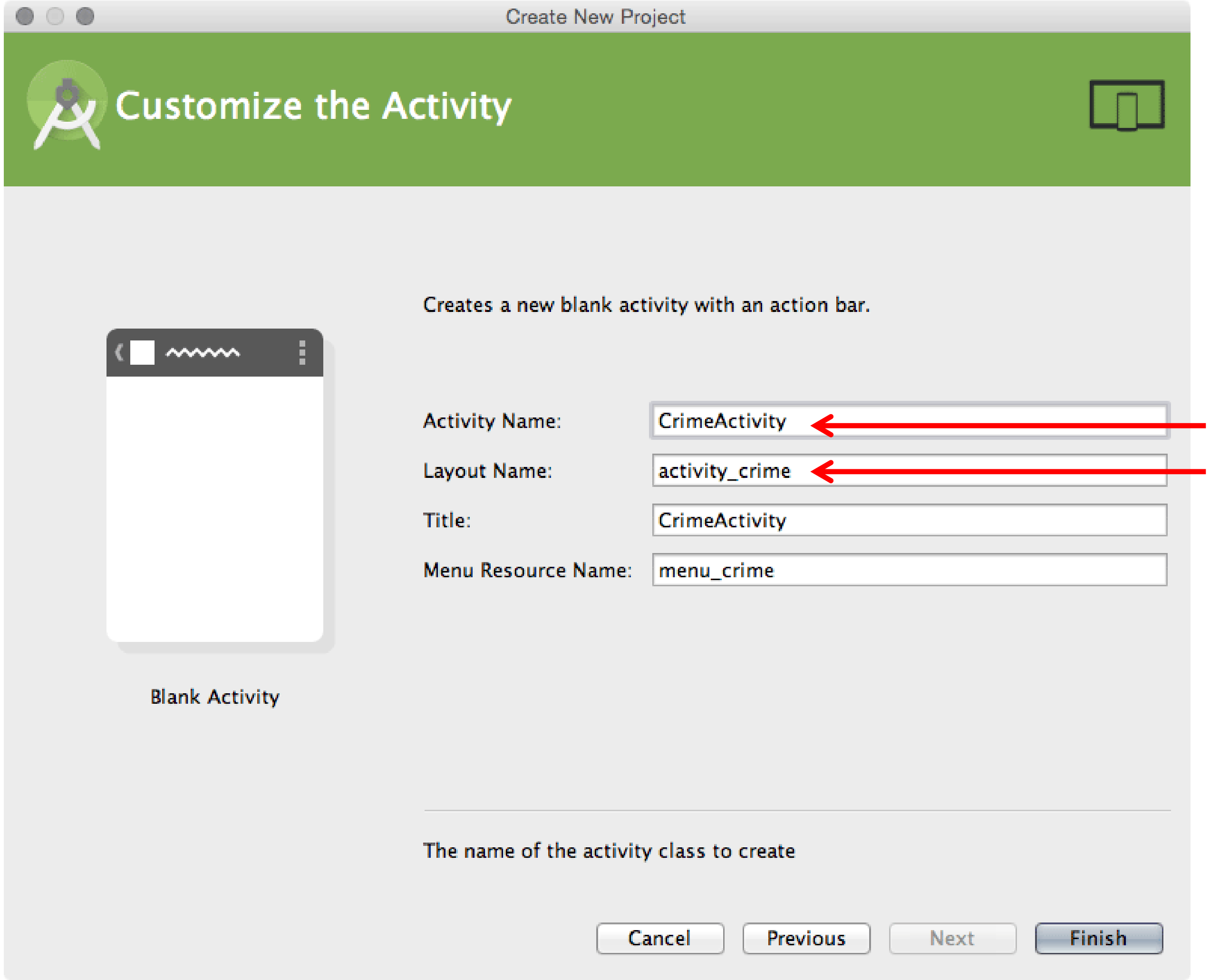
Enter a title for the crime.

DETAILS

THU NOV 17 09:45:07 EST 2016

□ Solved

Create CrimeActivity in Android Studio



Customize the Activity

Creates a new blank activity with an action bar.

Activity Name: CrimeActivity

Layout Name: activity_crime

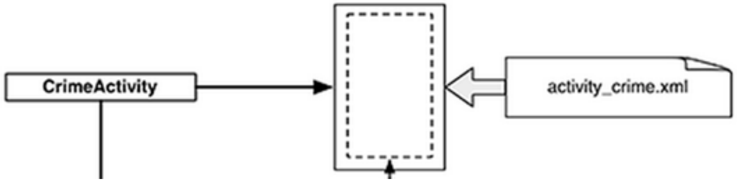
Title: CrimeActivity

Menu Resource Name: menu_crime

The name of the activity class to create

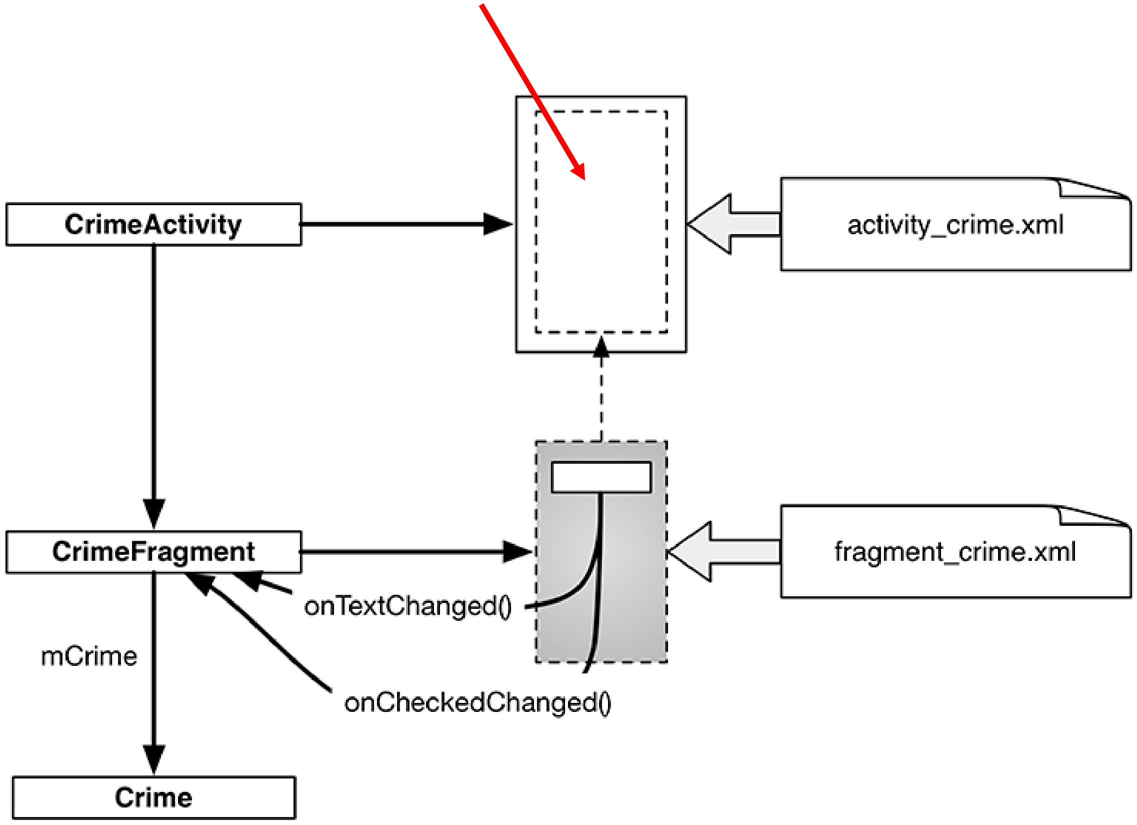
Creates CrimeActivity.java

Formatted using activity_crime.xml



Fragment Hosted by an Activity

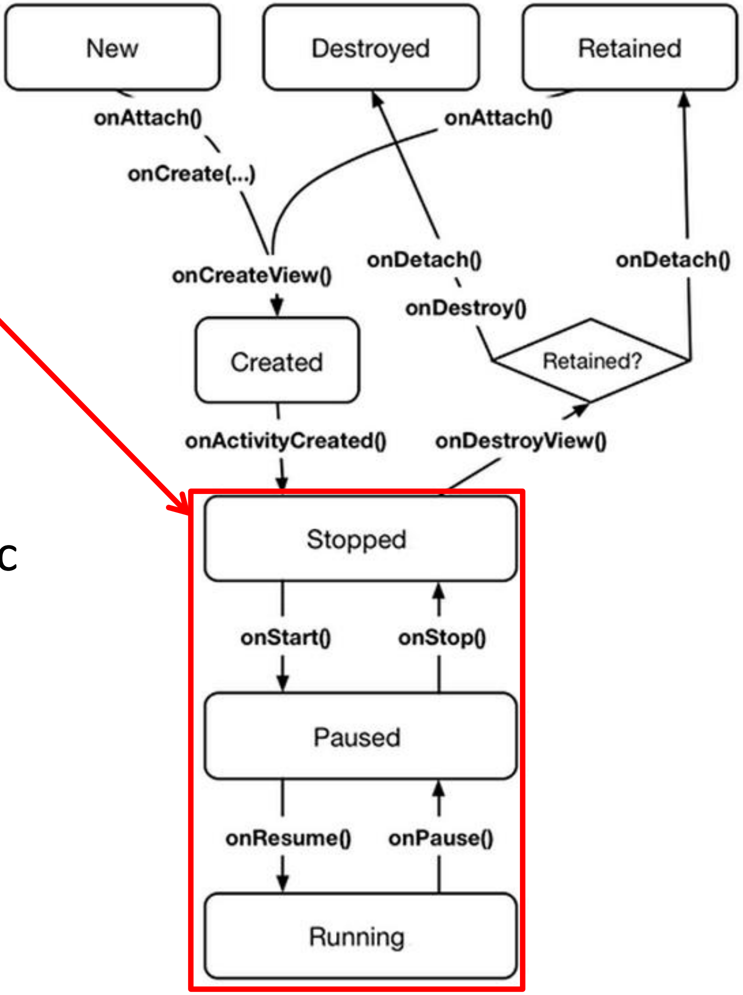
- Each fragment must be hosted by an Activity
- To host a UI fragment, an activity must
  - Define a spot in its layout for the fragment
  - Manage the lifecycle of the fragment instance (next)
- E.g.: **CrimeActivity** defines "spot" for **CrimeFragment**



Fragment's Life Cycle

- Fragment's lifecycle similar to activity lifecycle
  - Has states **running**, **paused** and **stopped**
  - Also has some similar activity lifecycle methods (e.g. **onPause()**, **onStop( )**, etc)

- **Key difference:**
  - Android OS calls Activity's onCreate, onPause( ), etc
  - Fragment's **onCreateView( )**, onPause( ), etc **called by hosting activity NOT Android OS!**
  - E.g. Fragment has **onCreateView**

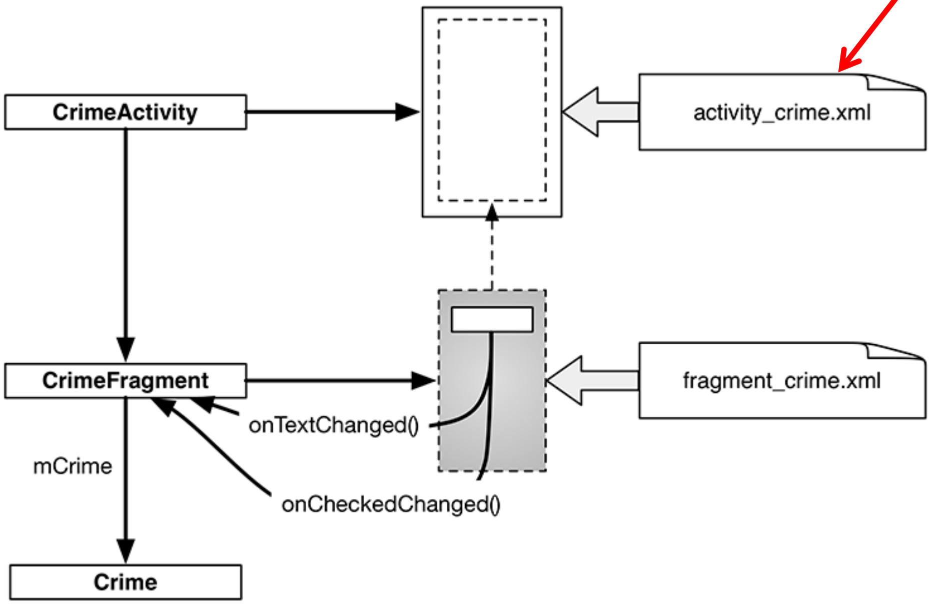


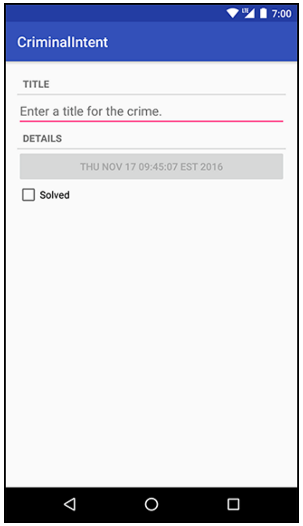
Hosting UI Fragment in an Activity

- 2 options. Can add fragment to either
  - **Activity's XML file (layout fragment),** or
  - **Activity's .java file** (more complex but more flexible)
- We will add fragment to activity's XML file now
- First, create a spot for the fragment's view in **CrimeActivity's** XML layout

FrameLayout

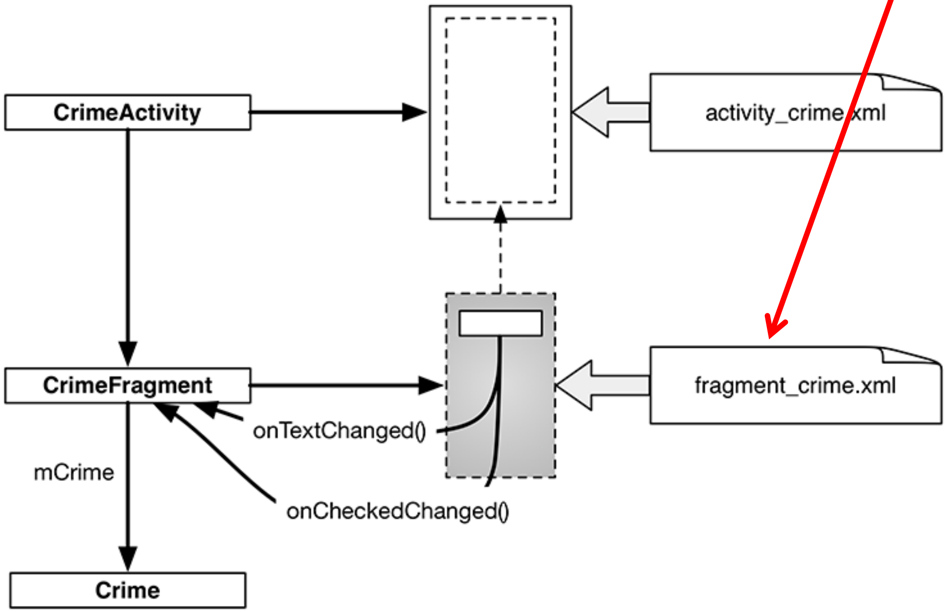
```
xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/fragment_container"
android:layout_width="match_parent"
android:layout_height="match_parent"
```

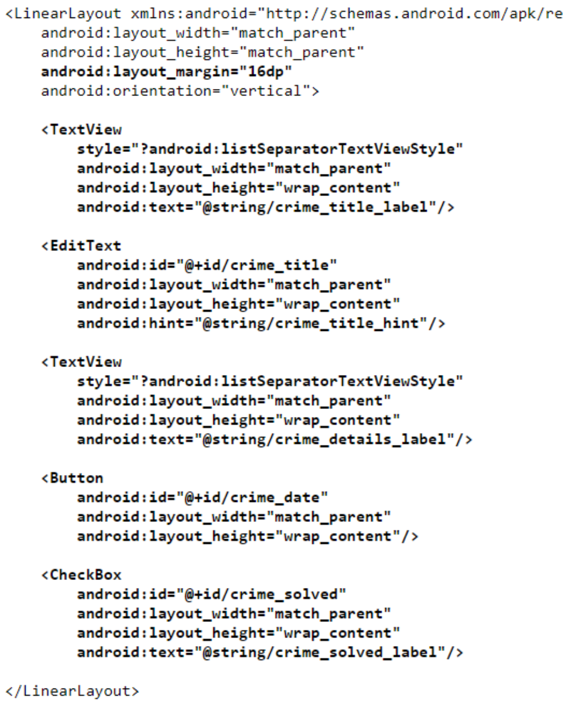




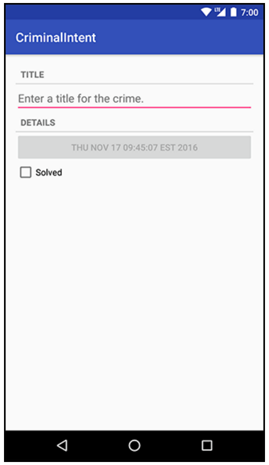
Creating a UI Fragment

- Creating Fragment is similar to creating activity
  1. Define widgets in a layout (XML) file
  2. Create java class and specify layout file as XML file above
  3. Get references of inflated widgets in java file (findviewbyId), etc
- XML layout file for **CrimeFragment (fragment_crime.xml)**



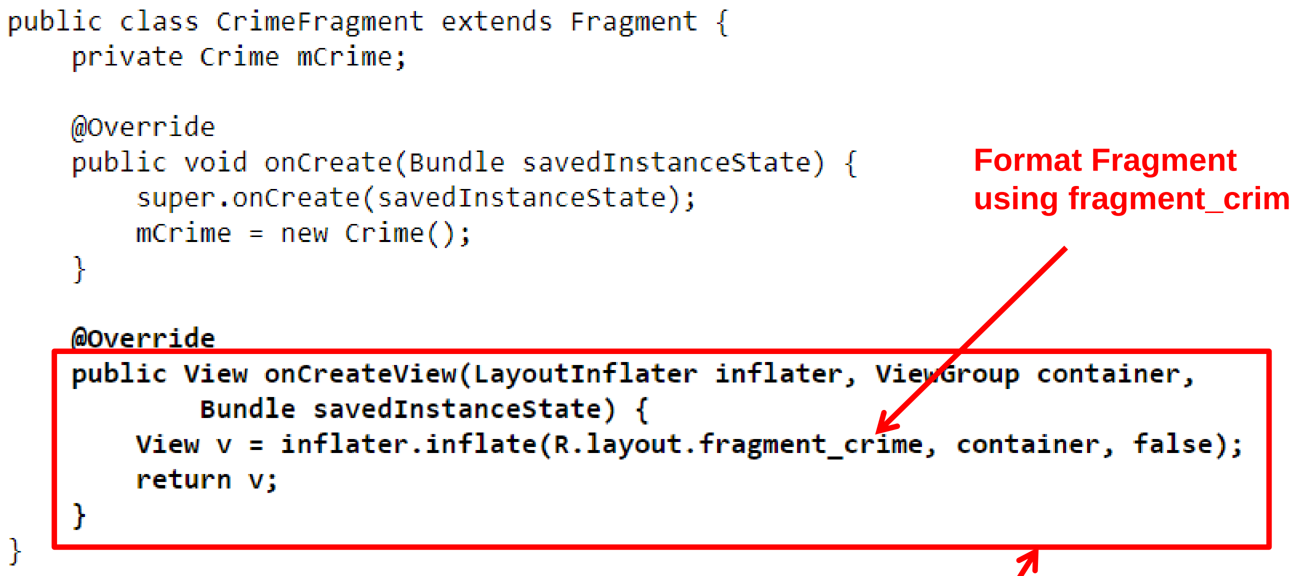


```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="16dp"
    android:orientation="vertical">

    <TextView
        style="?android:listSeparatorTextViewStyle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/crime_title_label"/>

    <EditText
        android:id="@+id/crime_title"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/crime_title_hint"/>

    <TextView
        style="?android:listSeparatorTextViewStyle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/crime_details_label"/>

    <Button
        android:id="@+id/crime_date"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

    <CheckBox
        android:id="@+id/crime_solved"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/crime_solved_label"/>

</LinearLayout>
```



Java File for CrimeFragment

- In **CrimeFragment** Override CrimeFragment's **onCreateView( )** function



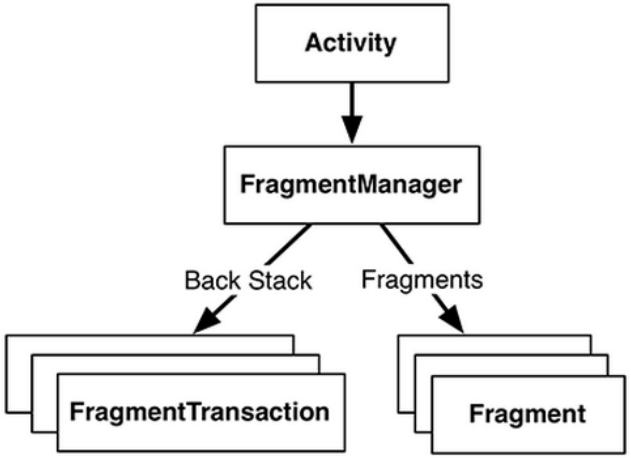
```
public class CrimeFragment extends Fragment {
    private Crime mCrime;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        mCrime = new Crime();
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
            Bundle savedInstanceState) {
        View v = inflater.inflate(R.layout.fragment_crime, container, false);
        return v;
    }
}
```

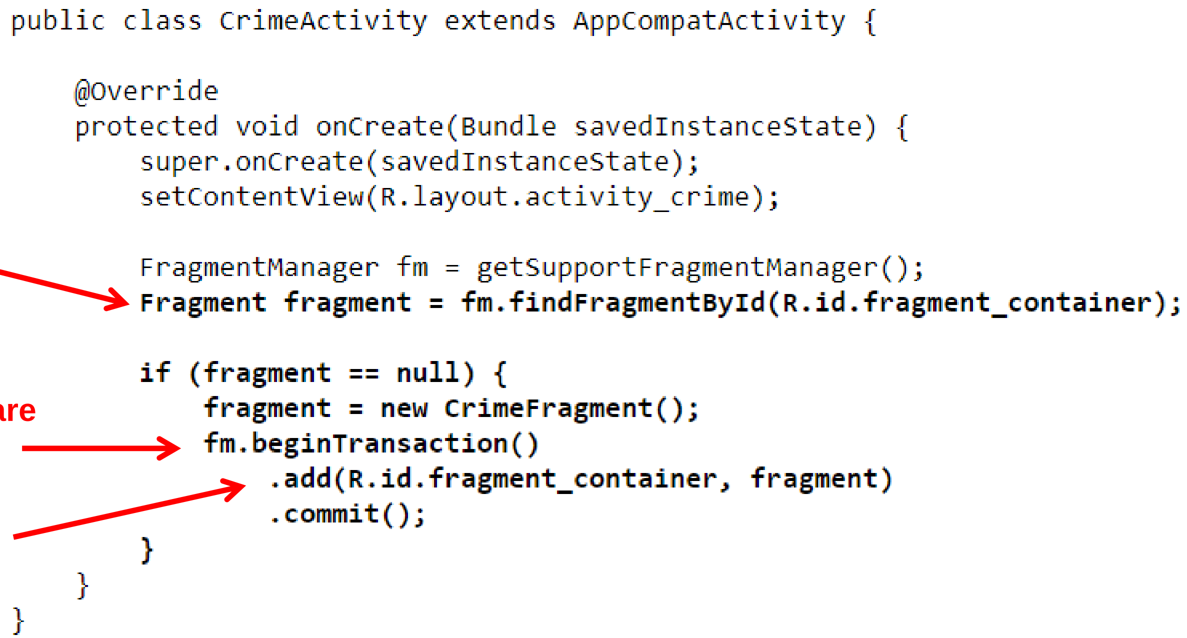
Format Fragment
using fragment_crime.xml

- **Note:** Fragment's view inflated in **Fragment.onCreateView()**, NOT **onCreate**

Adding UI Fragment to FragmentManager

- An activity adds new fragment to activity using **FragmentManager**
- **FragmentManager**
  - Manages fragments
  - Adds fragment’s views to activity’s view
  - Handles
    - List of fragments
    - Back stack of fragment transactions





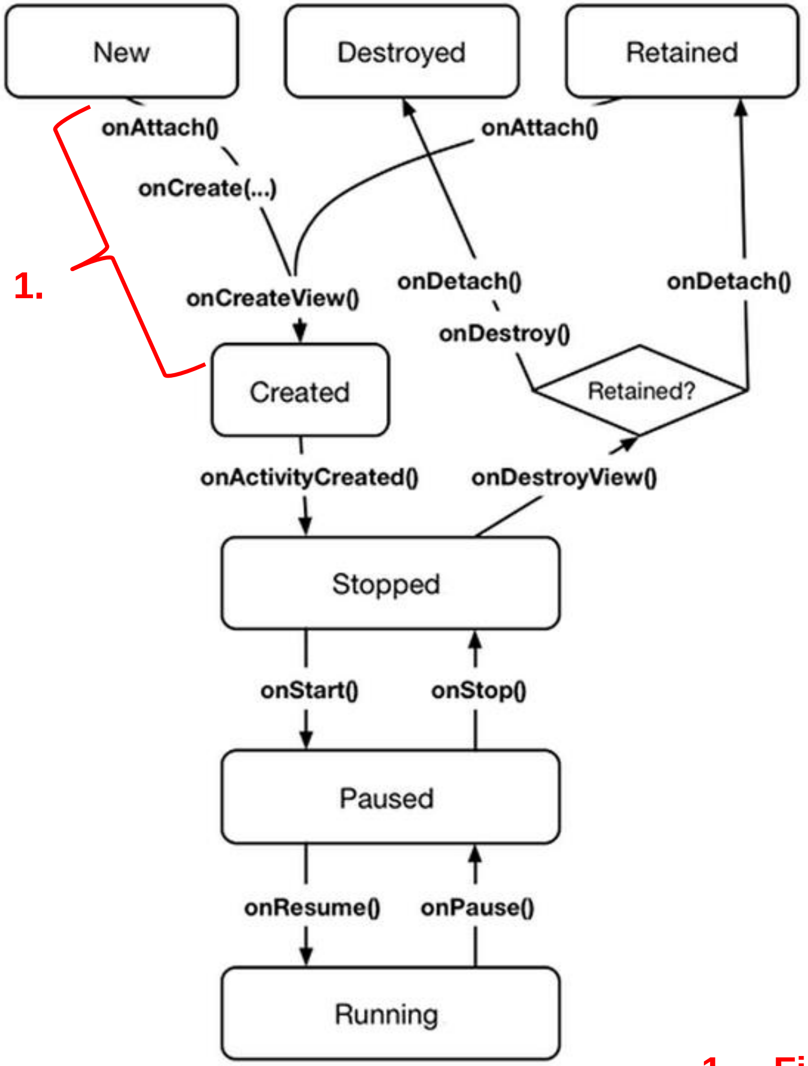
```
public class CrimeActivity extends AppCompatActivity {

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

        FragmentManager fm = getSupportFragmentManager();
        Fragment fragment = fm.findFragmentById(R.id.fragment_container);

        if (fragment == null) {
            fragment = new CrimeFragment();
            fm.beginTransaction()
                .add(R.id.fragment_container, fragment)
                .commit();
        }
    }
}
```

Find Fragment using its ID

Interactions with FragmentManager are done using transactions

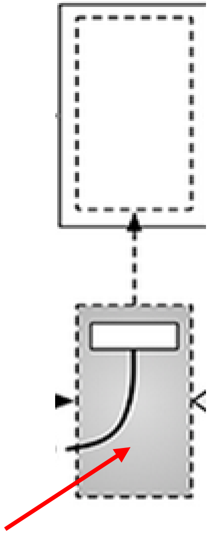
Add Fragment to activity’s view

Examining Fragment’s Lifecycle



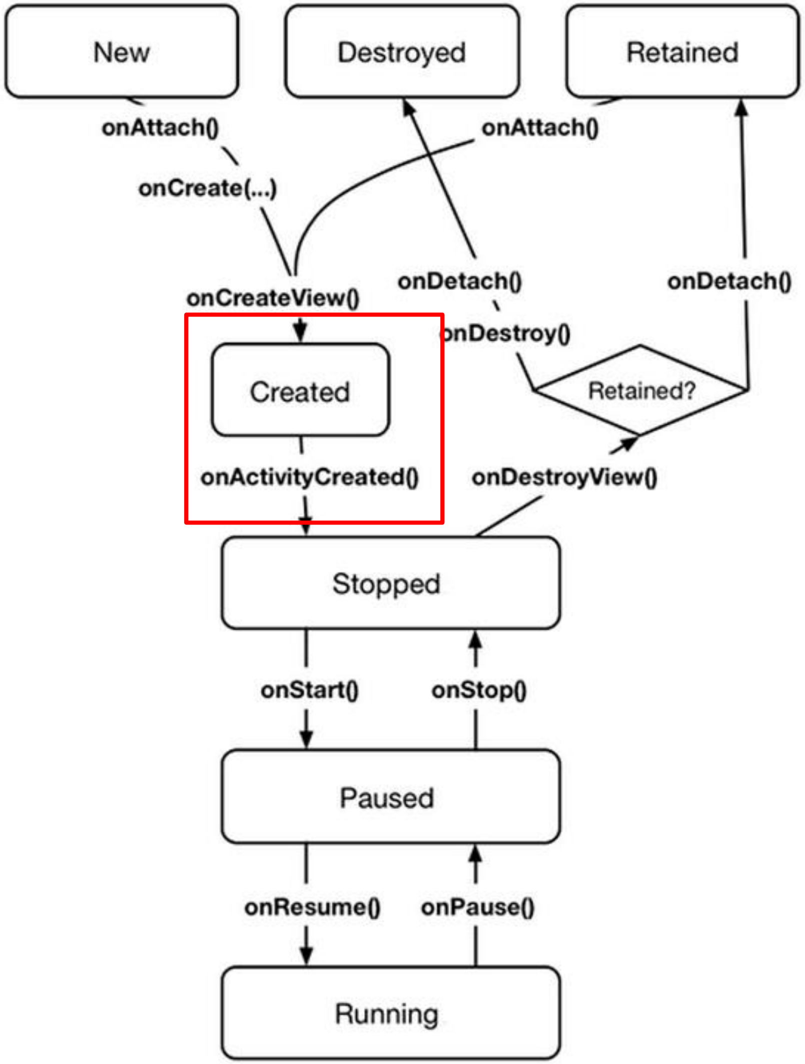
- **FragmentManager** calls fragment lifecycle methods

- **onAttach( ), onCreate( )** and **onCreateView()** called when a fragment is added to **FragmentManager**



1. **First create fragment**
   ….. then wait for Activity to add fragment

Examining Fragment’s Lifecycle

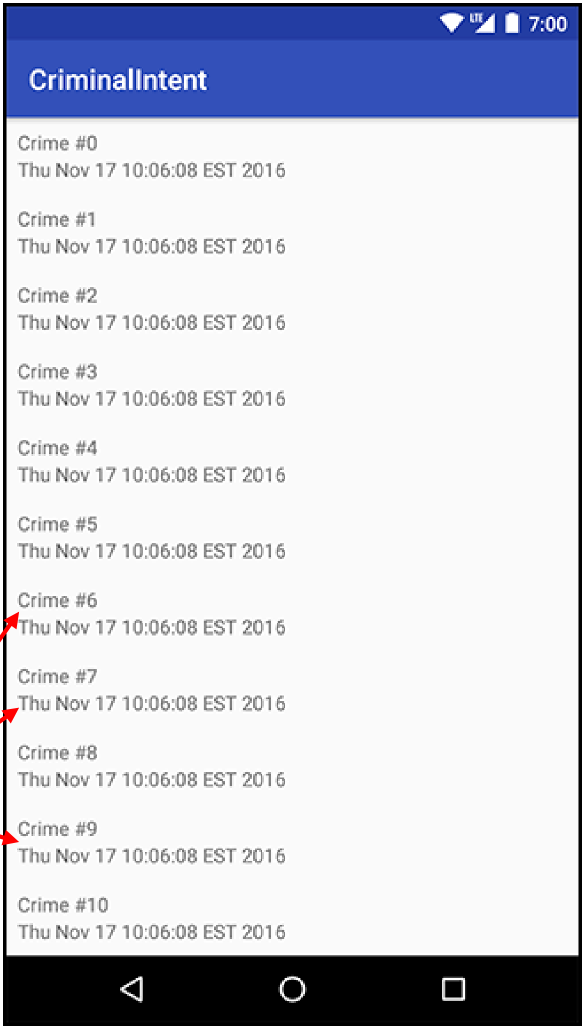


- **FragmentManager** calls fragment lifecycle methods
- **onAttach( ), onCreate( )** and **onCreateView()** called when a fragment is added to **FragmentManager**
- **onActivityCreated( )** called after hosting activity’s **onCreate( )** method is executed
- If fragment is added to already running Activity then **onAttach( ), onCreate( ), onCreateView()**, **onActivityCreated( )**, **onStart( )** and then **onResume( )** called

Android Nerd Ranch CriminalIntent Chapters Skipped

Chapter 8: Displaying Lists with RecyclerView

- Skipped several **UI chapters**
- These features are programmed into the **CriminalIntent** code you will be given for project 2
- RecyclerView facilitates view of large dataset
- E.g Allows crimes (title, date) in **CriminalIntent** to be listed



Crime #0
Thu Nov 17 10:06:08 EST 2016

Crime #1
Thu Nov 17 10:06:08 EST 2016

Crime #2
Thu Nov 17 10:06:08 EST 2016

Crime #3
Thu Nov 17 10:06:08 EST 2016

Crime #4
Thu Nov 17 10:06:08 EST 2016

Crime #5
Thu Nov 17 10:06:08 EST 2016

Crime #6
Thu Nov 17 10:06:08 EST 2016

Crime #7
Thu Nov 17 10:06:08 EST 2016

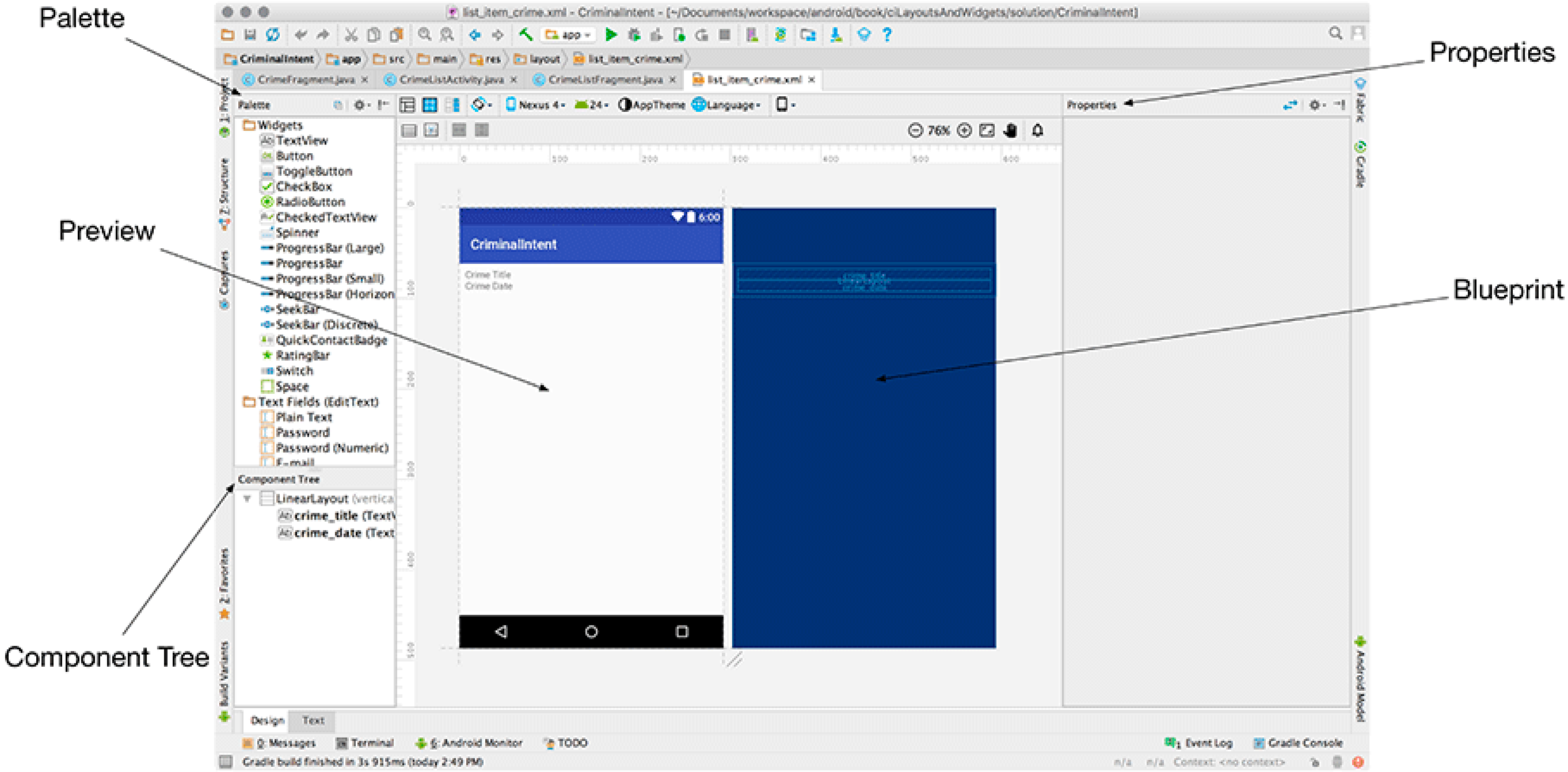
Crime #8
Thu Nov 17 10:06:08 EST 2016

Crime #9
Thu Nov 17 10:06:08 EST 2016

Crime #10
Thu Nov 17 10:06:08 EST 2016

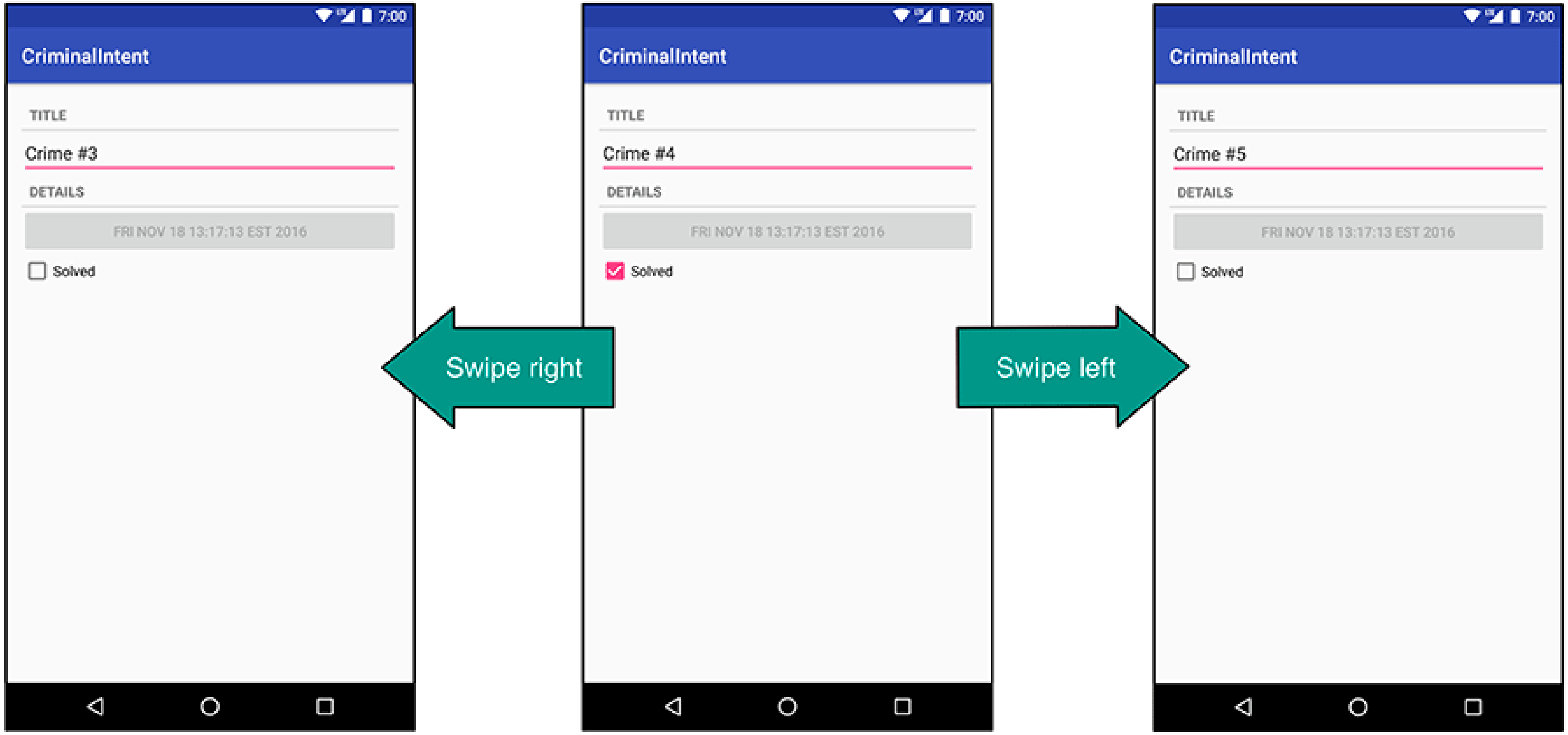
Chapter 9: Creating Android Layouts & Widgets

- Mostly already covered
- Does introduce Contraint Layout (specify widget positions using constraints)



Chapter 11: Using ViewPager

- ViewPager allows users swipe left-right between screens
  - Similar to Tinder
- E.g. Users can swipe left-right between Crimes in CriminalIntent



7:00

CriminalIntent

TITLE

Crime #3

DETAILS

FRI NOV 18 13:17:13 EST 2016

□ Solved

7:00

CriminalIntent

TITLE

Crime #4

DETAILS

FRI NOV 18 13:17:13 EST 2016

☑ Solved

7:00

CriminalIntent

TITLE

Crime #5

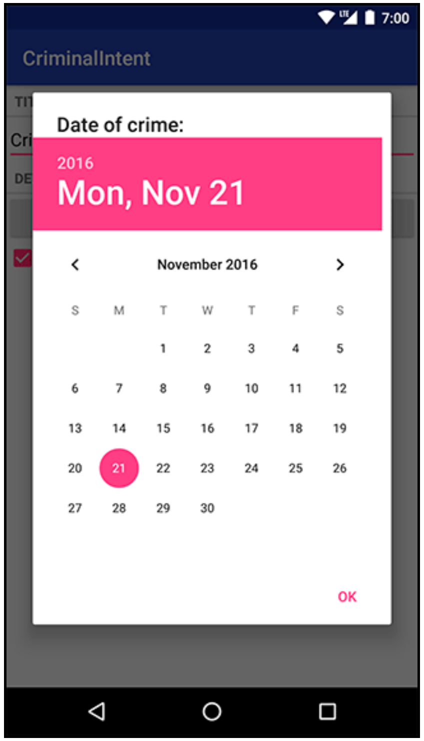
DETAILS

FRI NOV 18 13:17:13 EST 2016

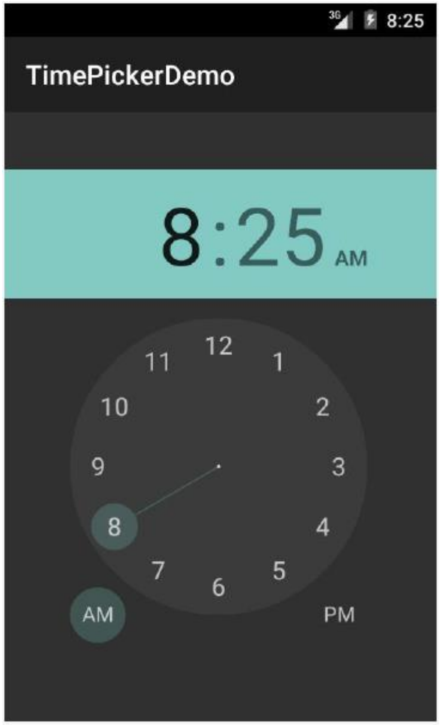
□ Solved

Chapter 12: Dialogs

- Dialogs present users with a choice or important information
- DatePicker allows users pick date
- Users can pick a date on which a crime occurred in **CriminalIntent**



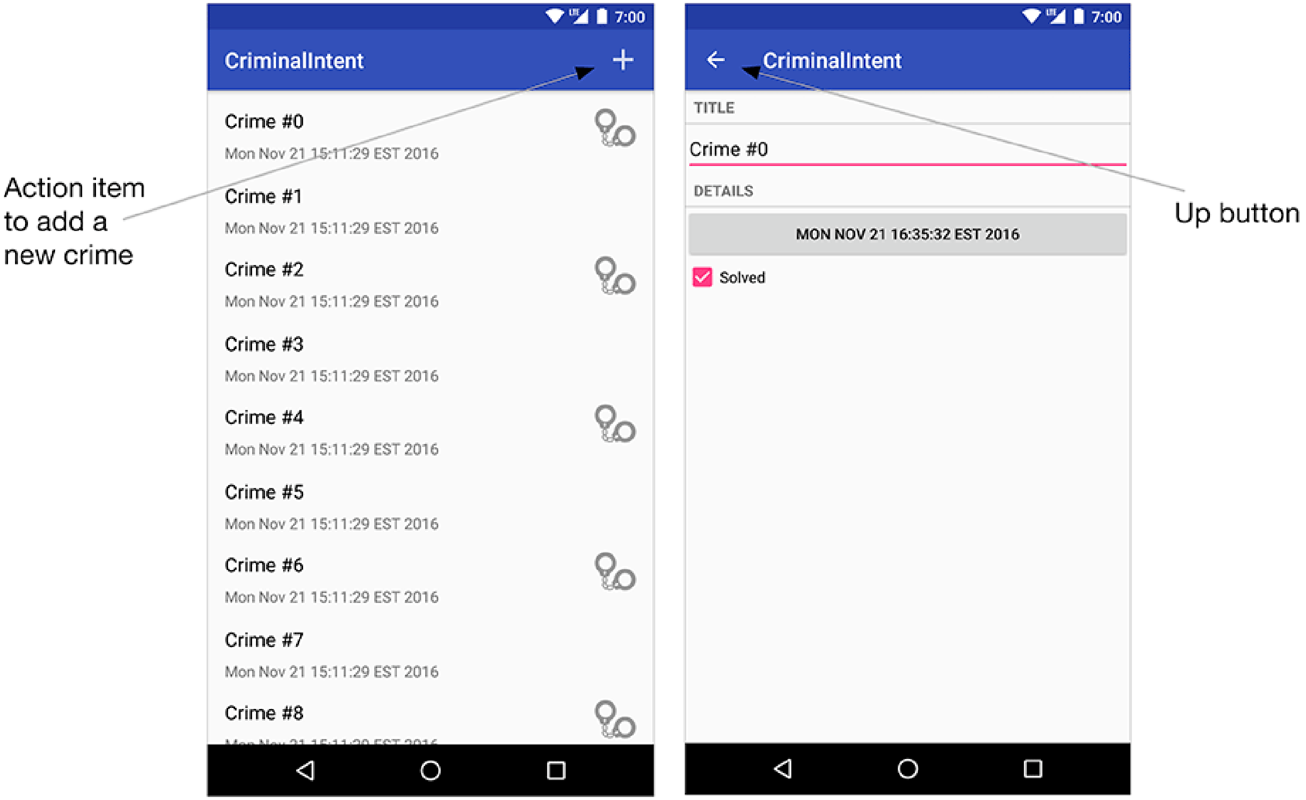
DatePicker



TimePicker also exists

Chapter 13: The Toolbar

- Toolbar includes actions user can take
- In CriminalIntent, menu items for adding crime, navigate up the screen hierarchy



7:00

CriminalIntent +

Crime #0
Mon Nov 21 15:11:29 EST 2016

Crime #1
Mon Nov 21 15:11:29 EST 2016

Crime #2
Mon Nov 21 15:11:29 EST 2016

Crime #3
Mon Nov 21 15:11:29 EST 2016

Crime #4
Mon Nov 21 15:11:29 EST 2016

Crime #5
Mon Nov 21 15:11:29 EST 2016

Crime #6
Mon Nov 21 15:11:29 EST 2016

Crime #7
Mon Nov 21 15:11:29 EST 2016

Crime #8

7:00

← CriminalIntent

TITLE

Crime #0

DETAILS

MON NOV 21 16:35:32 EST 2016

☑ Solved

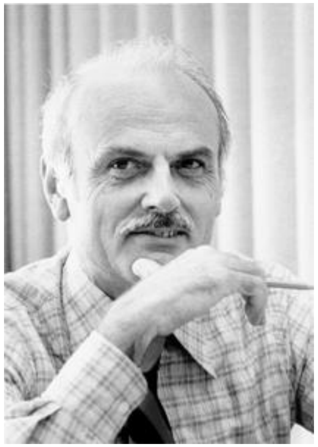
Action item to add a new crime

Up button

Android Nerd Ranch Ch 14
SQLite Databases

Background on Databases

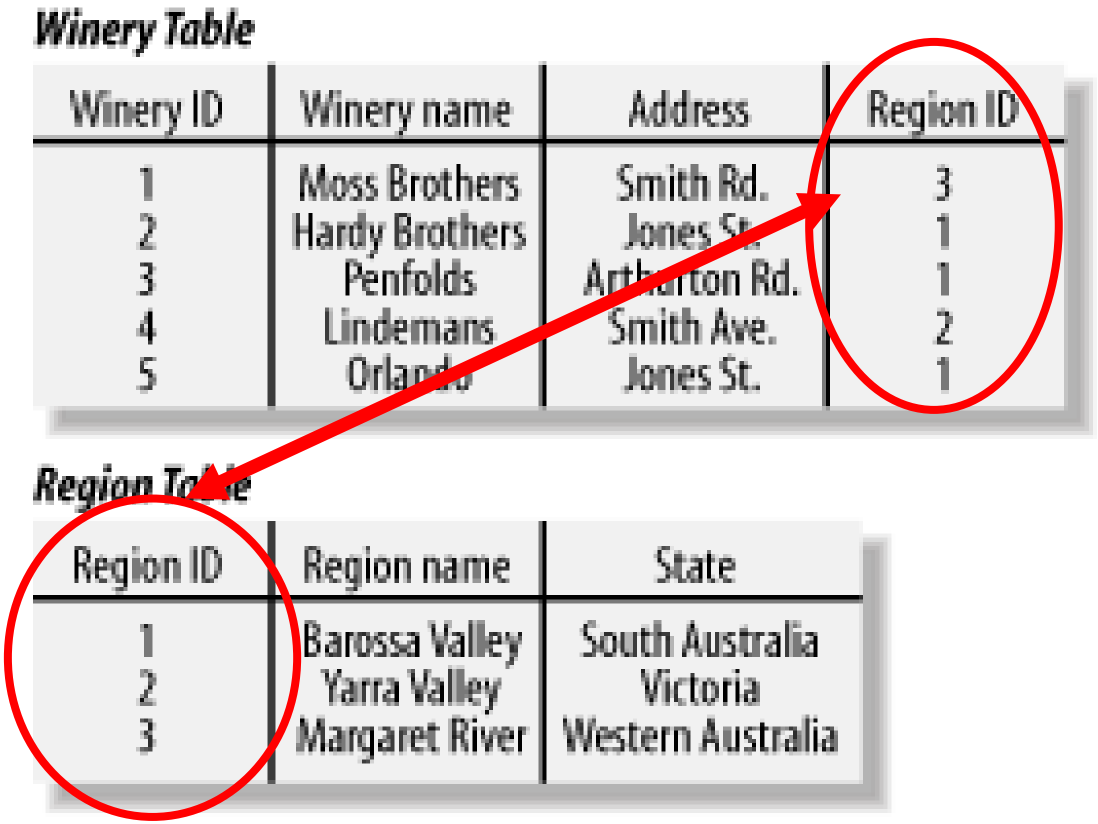
- Relational DataBase Management System (RDBMS)
  - Introduced by E. F. Codd (Turing Award Winner)



- Relational Database
  - data stored in tables
  - relationships among data stored in tables
  - data can be accessed and viewed in different ways

Example Wines Database

- **Relational Data:** Data in different tables can be related



Winery Table

Winery ID	Winery name	Address	Region ID
1	Moss Brothers	Smith Rd.	3
2	Hardy Brothers	Jones St.	1
3	Penfolds	Arthurton Rd.	1
4	Lindemans	Smith Ave.	2
5	Orlando	Jones St.	1

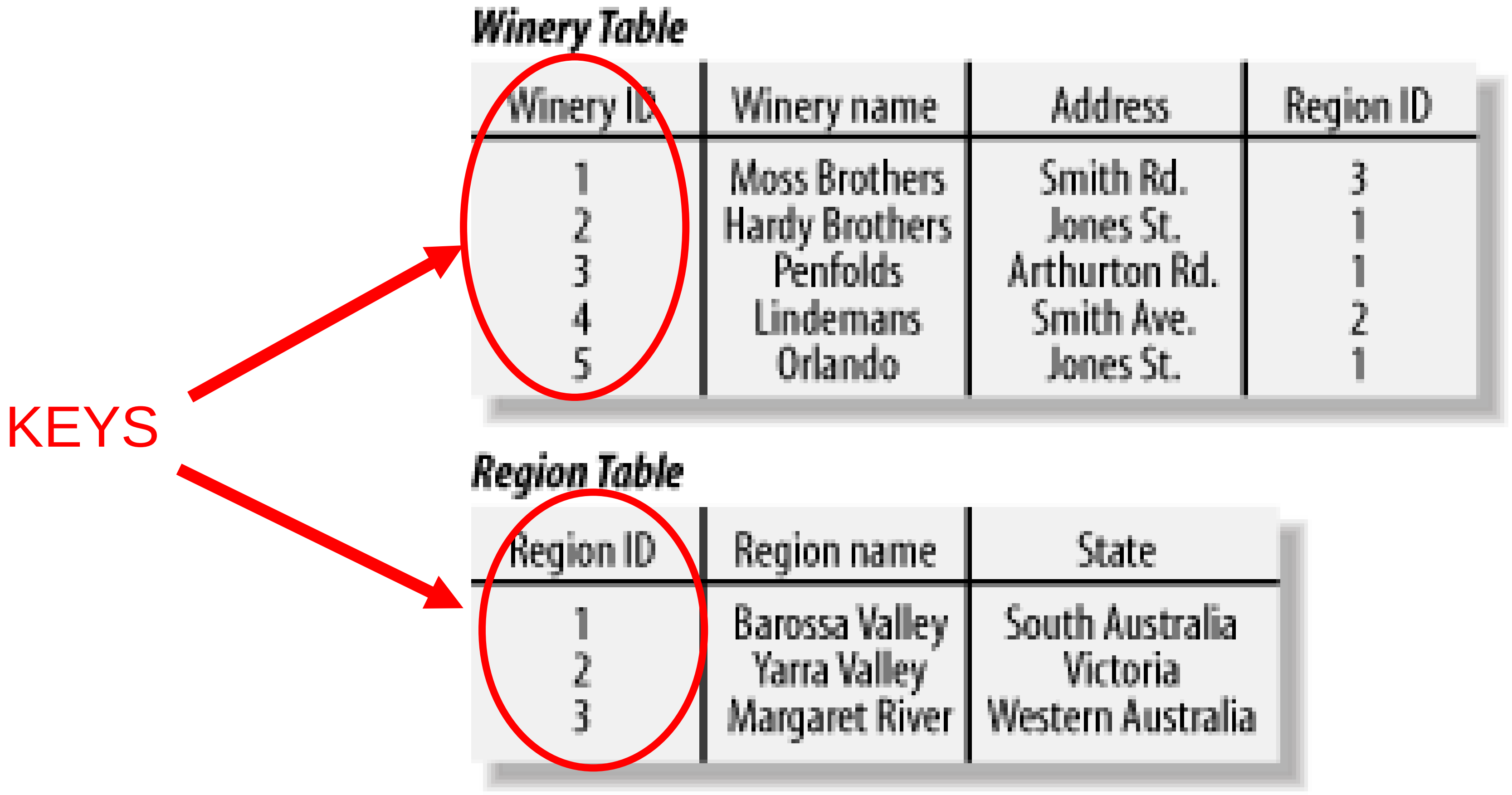
Region Table

Region ID	Region name	State
1	Barossa Valley	South Australia
2	Yarra Valley	Victoria
3	Margaret River	Western Australia

Ref: Web Database Applications with PHP and MySQL, 2nd Edition , by Hugh E. Williams, David Lane

Keys

- Each table has a key
- **Key:** column used to uniquely identify each row



KEYS

Winery Table

Winery ID	Winery name	Address	Region ID
1	Moss Brothers	Smith Rd.	3
2	Hardy Brothers	Jones St.	1
3	Penfolds	Arthurton Rd.	1
4	Lindemans	Smith Ave.	2
5	Orlando	Jones St.	1

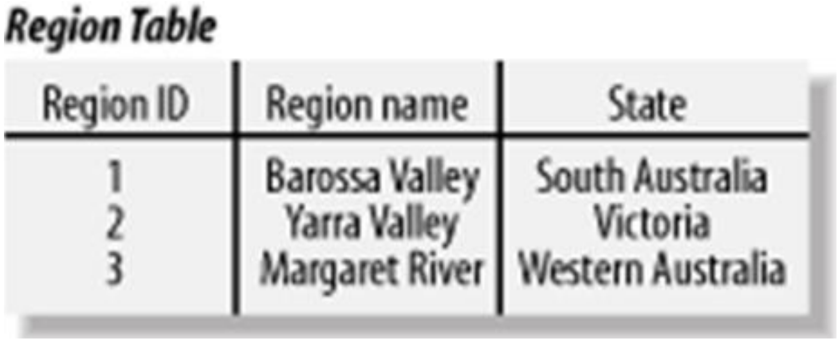
Region Table

Region ID	Region name	State
1	Barossa Valley	South Australia
2	Yarra Valley	Victoria
3	Margaret River	Western Australia

SQL and Databases

- **SQL:** language used to manipulate Relational Database (RDBMS)

- SQL Commands:
  - **CREATE TABLE** - creates new database table
  - **ALTER TABLE** - alters a database table
  - **DROP TABLE** - deletes a database table
  - **SELECT** - get data from a database table
  - **UPDATE** - change data in a database table
  - **DELETE** - remove data from a database table
  - **INSERT INTO** - insert new data in a database table

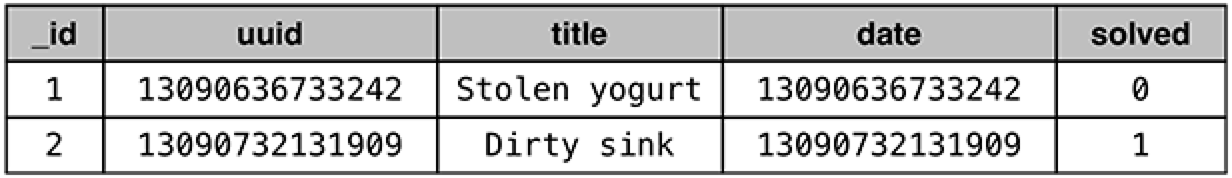


Region Table

Region ID	Region name	State
1	Barossa Valley	South Australia
2	Yarra Valley	Victoria
3	Margaret River	Western Australia

CriminalIntent Database

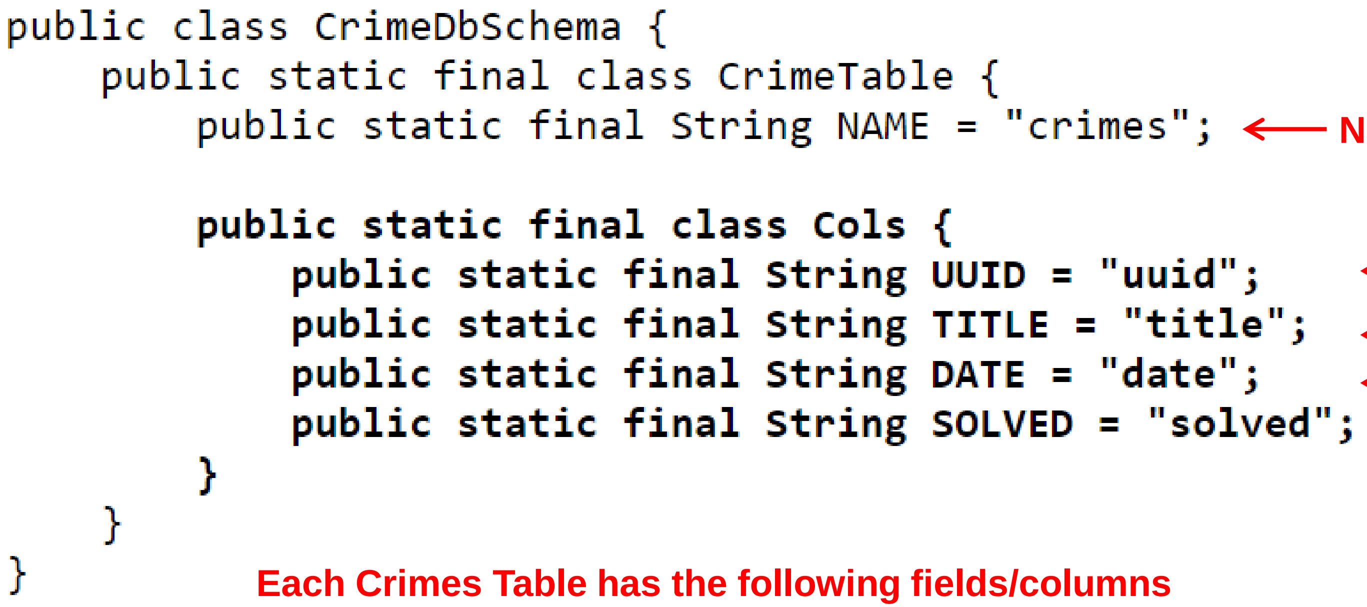
- **SQLite:** open source relational database
- SQLite implements subset of SQL (most but not all)
  - http://www.sqlite.org/
- Android includes a SQLite database
- **Goal:** Store crimes in CriminalIntent in SQLite database
- First step, define database table of **crimes**



_id	uuid	title	date	solved
1	13090636733242	Stolen yogurt	13090636733242	0
2	13090732131909	Dirty sink	13090732131909	1

CriminalIntent Database Schema

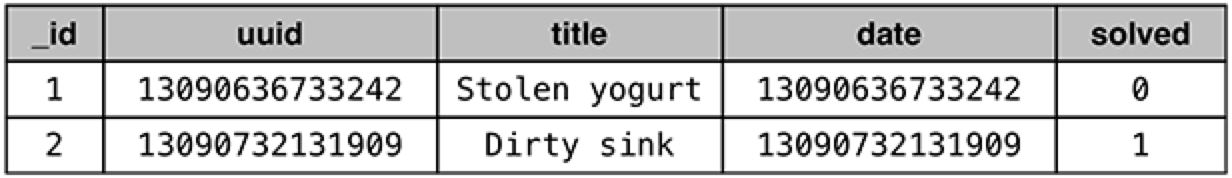
- Create **CrimeDbSchema** class to store **crime** database
- Define fields/columns of the Crimes database table



```
public class CrimeDbSchema {
    public static final class CrimeTable {
        public static final String NAME = "crimes";

        public static final class Cols {
            public static final String UUID = "uuid";
            public static final String TITLE = "title";
            public static final String DATE = "date";
            public static final String SOLVED = "solved";
        }
    }
}
```

Name of Table

Each Crimes Table has the following fields/columns

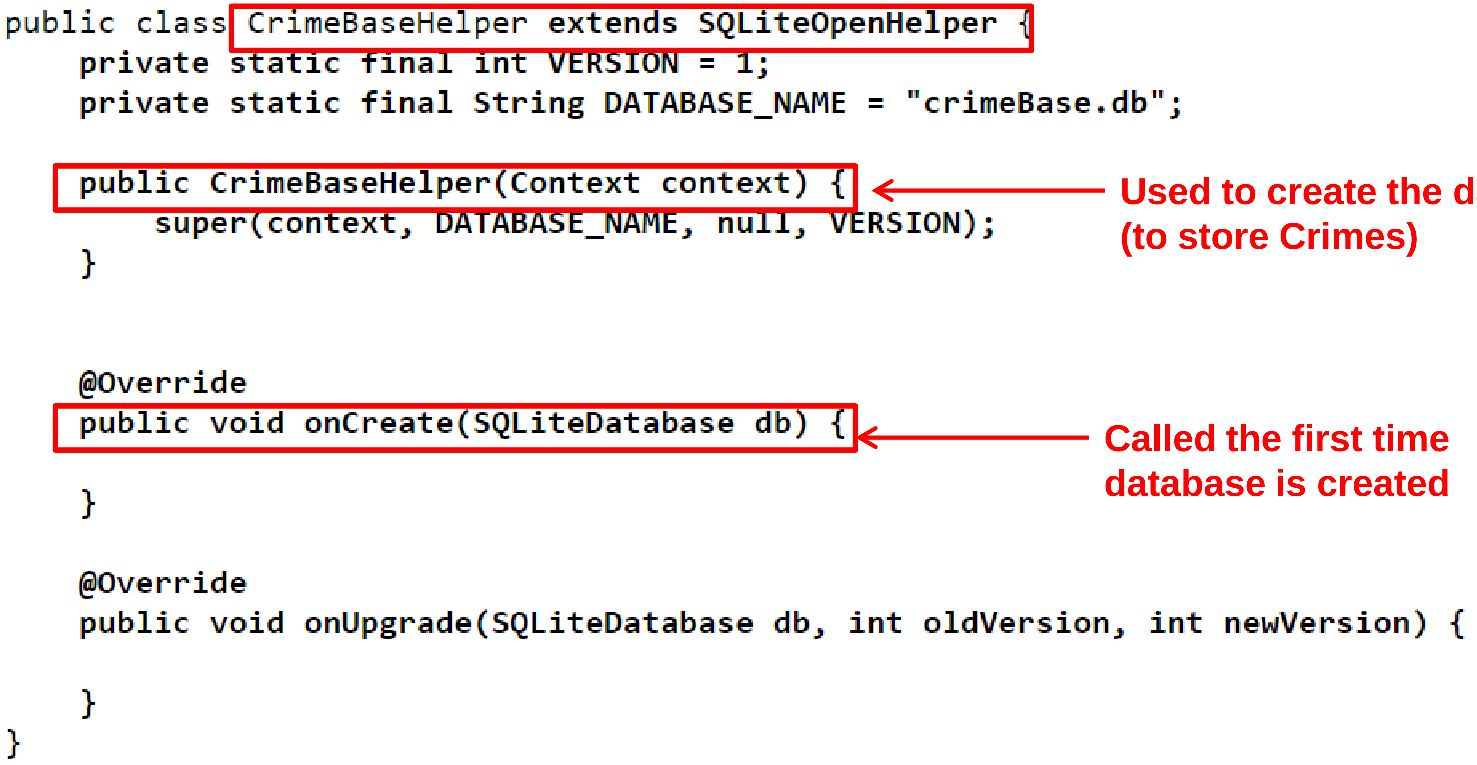


_id	uuid	title	date	solved
1	13090636733242	Stolen yogurt	13090636733242	0
2	13090732131909	Dirty sink	13090732131909	1

Crimes Table

SQLiteOpenHelper

- **SQLiteOpenHelper** class used for database creation, opening and updating a **SQLiteDatabase**
- In **CriminalIntent**, create subclass of **SQLiteOpenHelper** called **CrimeBaseHelper**

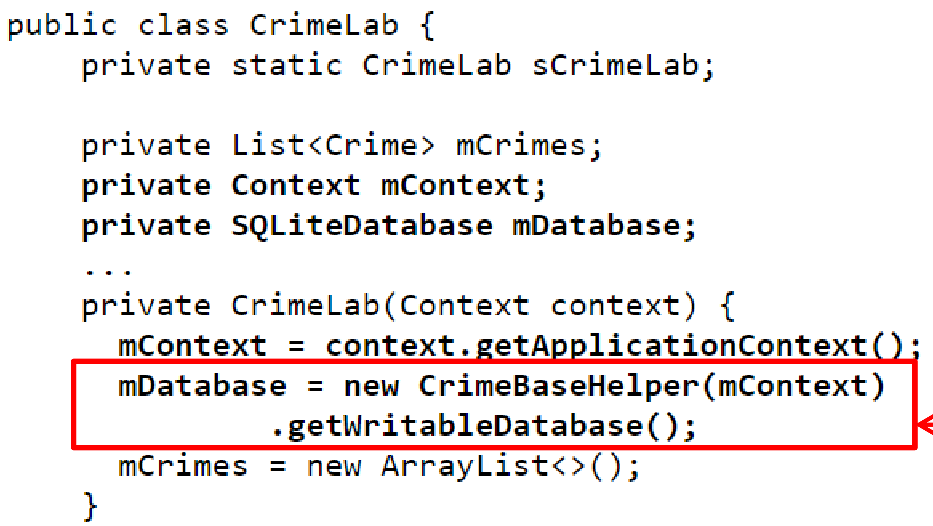


```
public class CrimeBaseHelper extends SQLiteOpenHelper {
    private static final int VERSION = 1;
    private static final String DATABASE_NAME = "crimeBase.db";

    public CrimeBaseHelper(Context context) {
        super(context, DATABASE_NAME, null, VERSION);
    }


    @Override
    public void onCreate(SQLiteDatabase db) {

    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

    }
}
```

Used to create the database (to store Crimes)

Called the first time database is created

Use CrimeBaseHelper to open SQLite Database

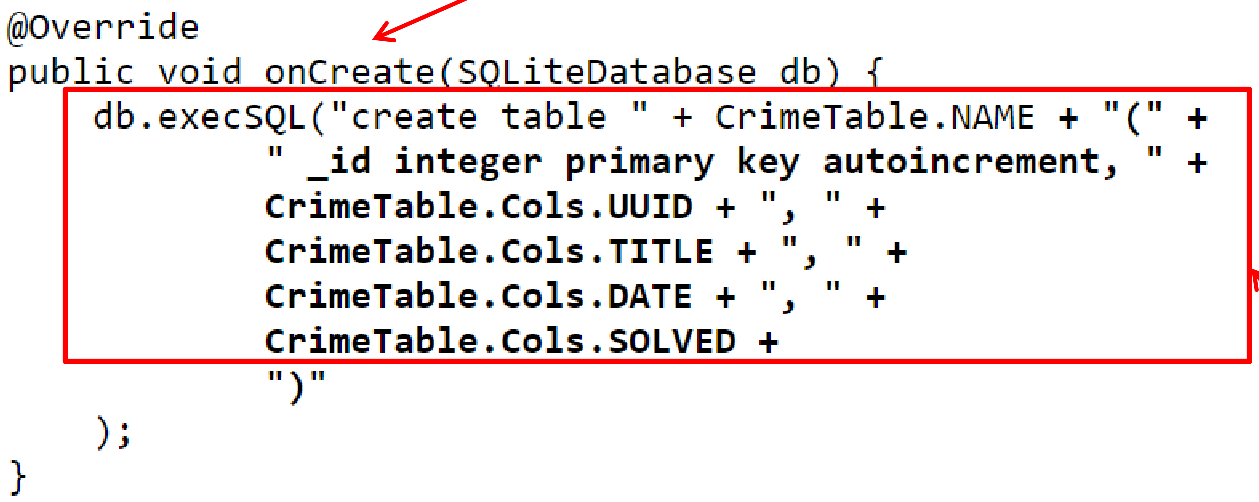


```
public class CrimeLab {
    private static CrimeLab sCrimeLab;

    private List<Crime> mCrimes;
    private Context mContext;
    private SQLiteDatabase mDatabase;
    ...
    private CrimeLab(Context context) {
      mContext = context.getApplicationContext();
      mDatabase = new CrimeBaseHelper(mContext)
              .getWritableDatabase();
      mCrimes = new ArrayList<>();
    }
```

Open new writeable Database

Create CrimeTable in onCreate( )

onCreate called first time
database is created

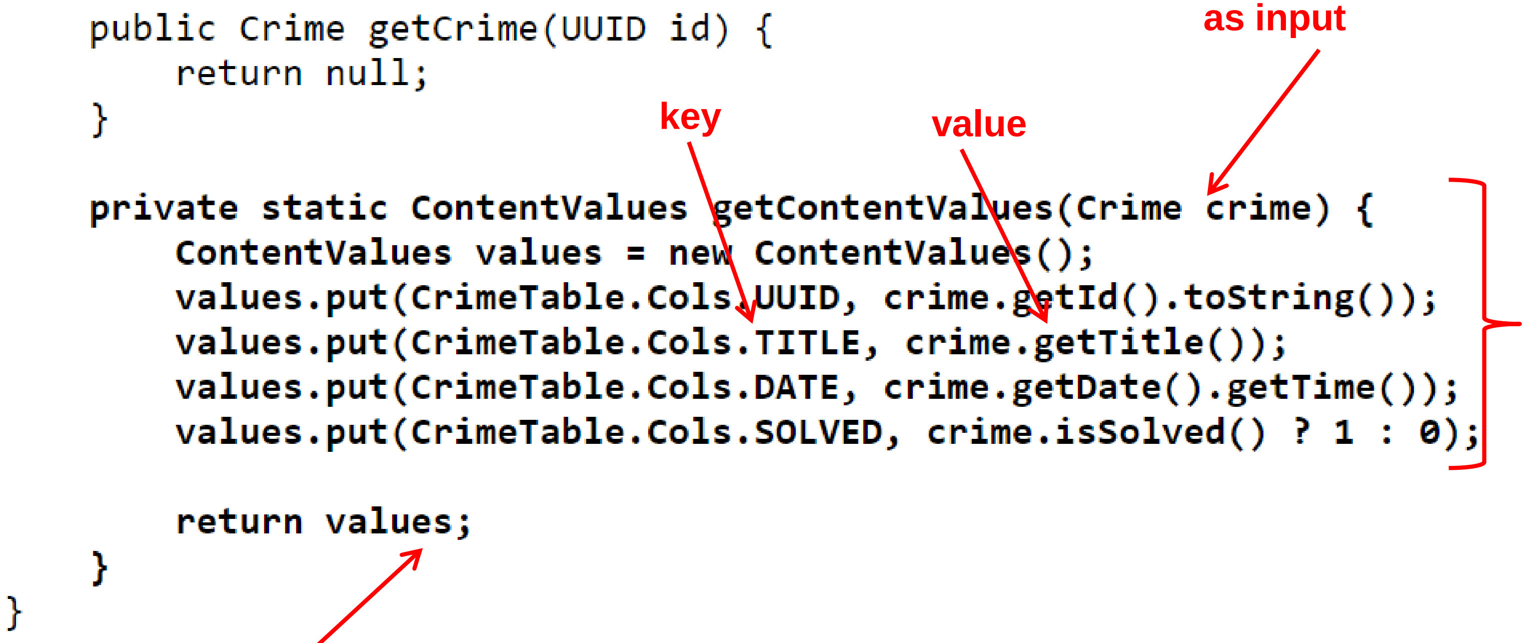


```
@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL("create table " + CrimeTable.NAME + "(" +
            " _id integer primary key autoincrement, " +
            CrimeTable.Cols.UUID + ", " +
            CrimeTable.Cols.TITLE + ", " +
            CrimeTable.Cols.DATE + ", " +
            CrimeTable.Cols.SOLVED +
            ")"
    );
}
```

Create CrimeTable in our new
Crimes Database

Writing Crimes to Database using ContentValues

- In Android, writing to databases is done using class **ContentValues**
- **ContentValues** is key-value pair
- Create method to create **ContentValues** instance from a **Crime**



```
    public Crime getCrime(UUID id) {
        return null;
    }

    private static ContentValues getContentValues(Crime crime) {
        ContentValues values = new ContentValues();
        values.put(CrimeTable.Cols.UUID, crime.getId().toString());
        values.put(CrimeTable.Cols.TITLE, crime.getTitle());
        values.put(CrimeTable.Cols.DATE, crime.getDate().getTime());
        values.put(CrimeTable.Cols.SOLVED, crime.isSolved() ? 1 : 0);

        return values;
    }
}
```

Takes Crime as input

key

value

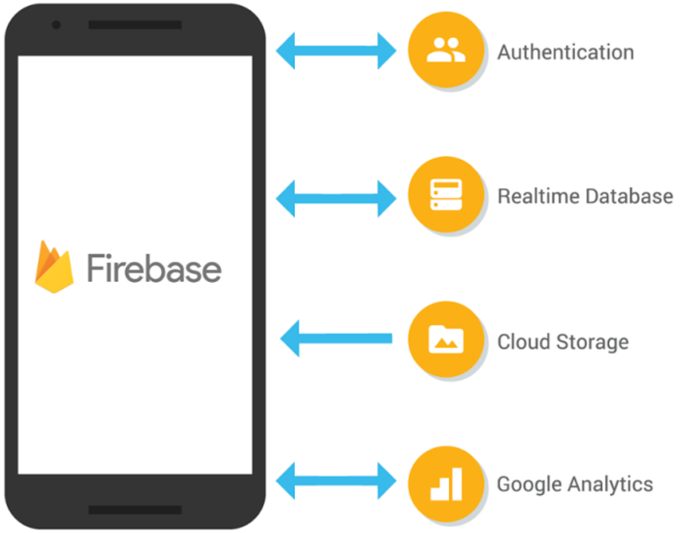
Converts Crime to ContentValues

Returns values as output

Firebase Cloud API

Firebase

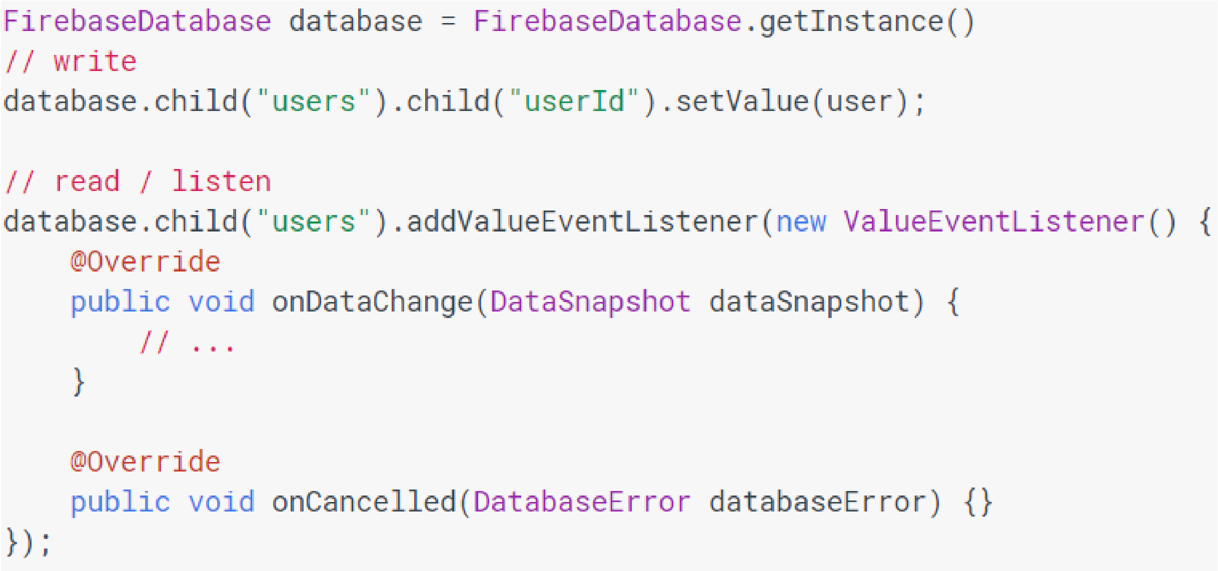
- Mobile cloud backend service for
  - Analytics
  - Messaging
  - Authentication
  - Database
  - Crash reporting, etc



- Previously 3rd party company
- Acquired by Google in 2014
  - Now part of Google. See https://firebase.google.com/
  - Fully integrated, could speed up development. E.g. final project

Firebase

- Relatively easy programming, few lines of code
- E.g. to create database



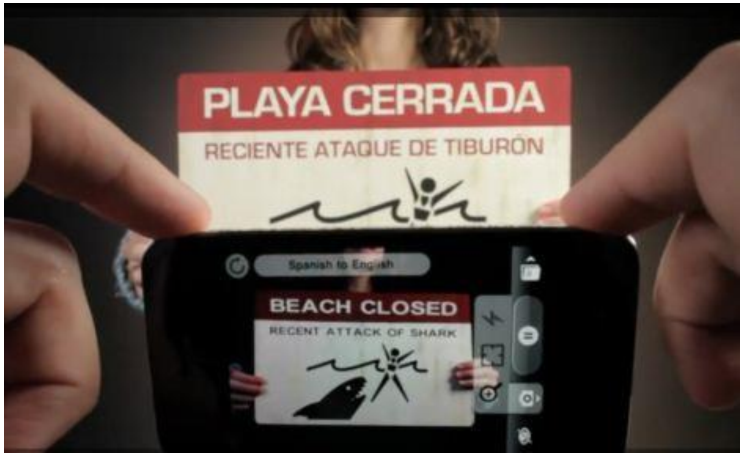
```
FirebaseDatabase database = FirebaseDatabase.getInstance()
// write
database.child("users").child("userId").setValue(user);

// read / listen
database.child("users").addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        // ...
    }

    @Override
    public void onCancelled(DatabaseError databaseError) {}
});
```

The Mobile Camera

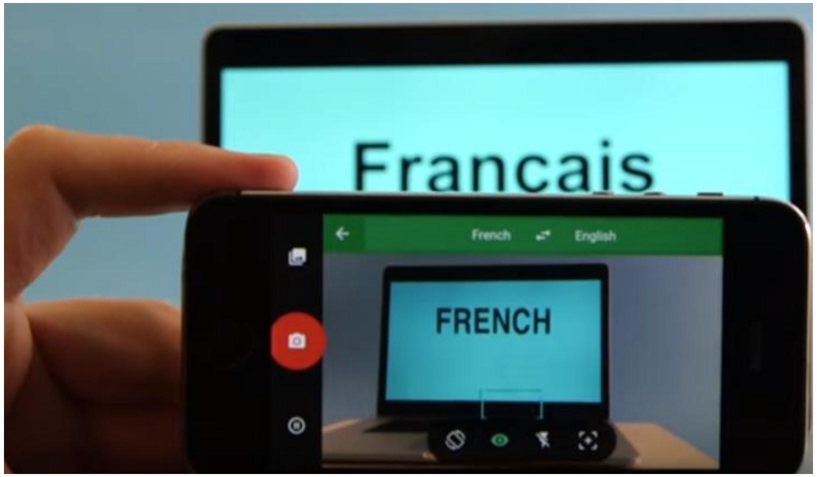
Interesting application

Word Lens Feature of Google Translate

- Word Lens: translates text/signs in foreign Language in real time
- Example use case: tourist can understand signs, restaurant menus
- Uses Optical Character Recognition technology
- Google bought company in 2014, now part of Google Translate



[Original Word Lens App]



[Word Lens as part of Google Translate]

Camera: Taking Pictures

Taking Pictures with Camera

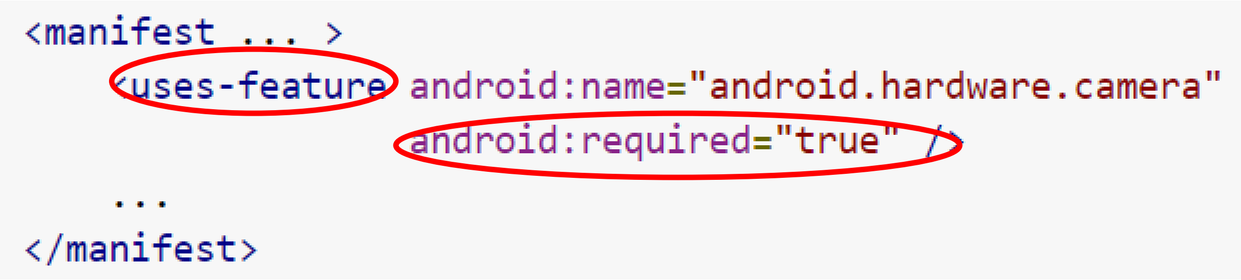
Ref: https://developer.android.com/training/camera/photobasics.html

- How to take photos from your app using Android Camera app
- 4 Steps:
  1. Request the camera feature
  2. Take a Photo with the Camera App
  3. Get the Thumbnail
  4. Save the Full-size Photo

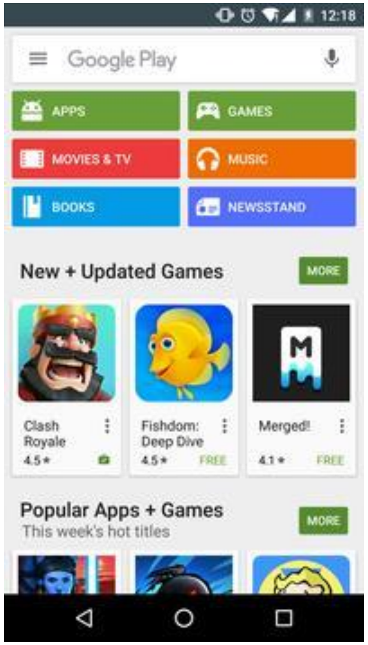
1. Request the Smartphone Camera Feature

Ref: https://developer.android.com/training/camera/photobasics.html

- If your app takes pictures using the phone's Camera, you can allow only devices with a camera find your app while searching Google Play Store
- How?
- Make the following declaration in AndroidManifest.xml



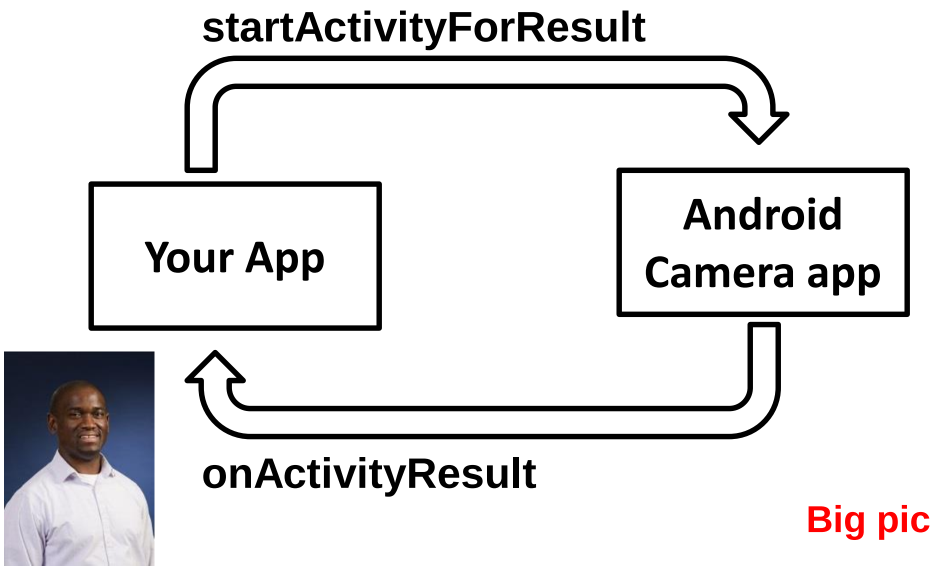
```
<manifest ... >
    <uses-feature android:name="android.hardware.camera"
                  android:required="true" />
    ...
</manifest>
```



2. Capture an Image with the Camera App

Ref: https://developer.android.com/training/camera/photobasics.html

- To take picture, your app needs to send **implicit Intent** requesting for a picture to be taken (i.e. action = capture an image)
- Call **startActivityForResult( )** with Camera intent since picture sent back
- Potentially, multiple apps/activities can handle this/take a picture
- Check that at least 1 Activity that can handle request to take picture using **resolveActivity**



startActivityForResult

Your App

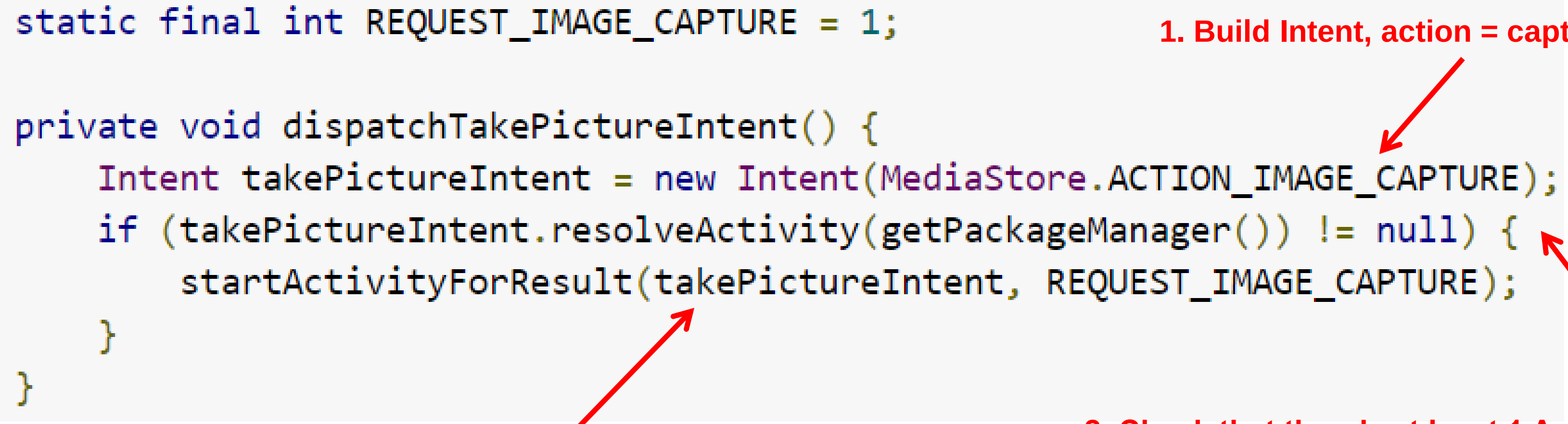
Android Camera app

onActivityResult

Big picture: taking a picture

Code to Take a Photo with the Camera App

Ref: https://developer.android.com/training/camera/photobasics.html

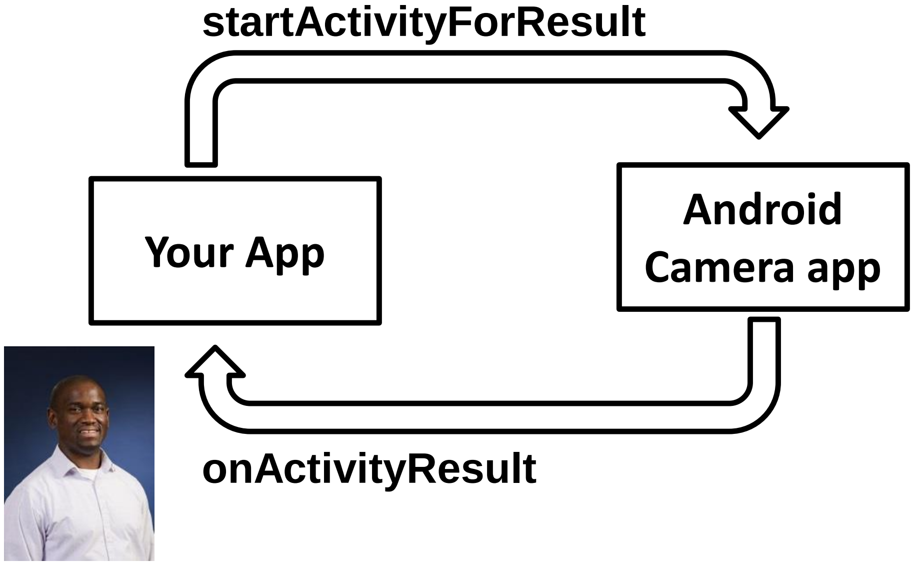


```
static final int REQUEST_IMAGE_CAPTURE = 1;

private void dispatchTakePictureIntent() {
    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
        startActivityForResult(takePictureIntent, REQUEST_IMAGE_CAPTURE);
    }
}
```

1. Build Intent, action = capture an image

2. Check that there's at least 1 Activity that can handle request to capture an image (Avoids app crashing if no camera app available)

3. Send Intent requesting an image to be captured (usually handled by Android's Camera app)

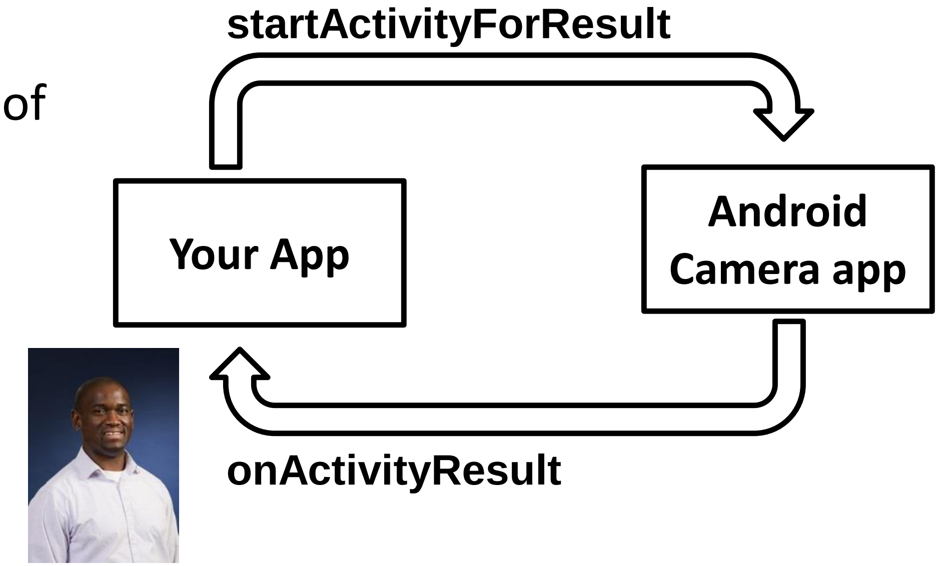


3. Get the Thumbnail

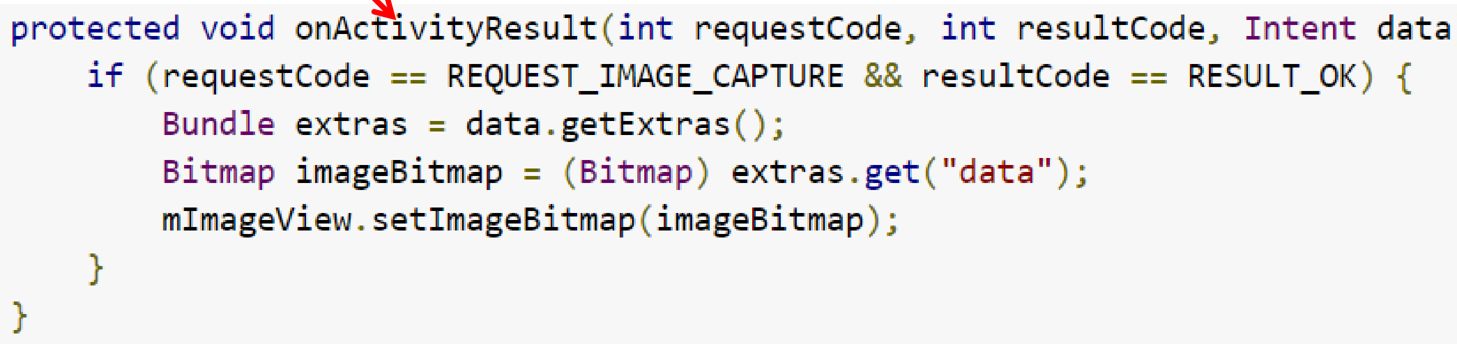
Ref: https://developer.android.com/training/camera/photobasics.html

- Android Camera app returns thumbnail of photo (small bitmap)

- Thumbnail bitmap returned in “extra” of **Intent** delivered to **onActivityResult( )**



In onActivityResult( ), receive thumbnail picture sent back



```
protected void onActivityResult(int requestCode, int resultCode, Intent data
    if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
        Bundle extras = data.getExtras();
        Bitmap imageBitmap = (Bitmap) extras.get("data");
        mImageView.setImageBitmap(imageBitmap);
    }
}
```

4. Save Full-Sized Photo

Ref: https://developer.android.com/training/basics/data-storage/files.html

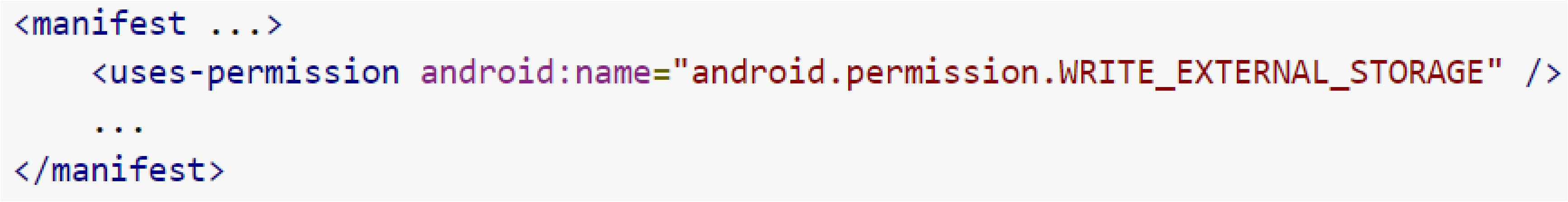
- Android Camera app saves full-sized photo in a filename you give it
- We need phone owner's permission to write to external storage
- Android systems have:
  - **Internal storage:** data stored here is available by only your app
  - **External storage:** available stored here is available to all apps
- Would like all apps to read pictures this app takes, so use external storage

Save Full-Sized Photo

Ref: https://developer.android.com/training/basics/data-storage/files.html

- Android Camera app can save full-size photo to
  1. **Public external storage** (shared by all apps)
     - **getExternalStoragePublicDirectory( )**
     - Need to get permission

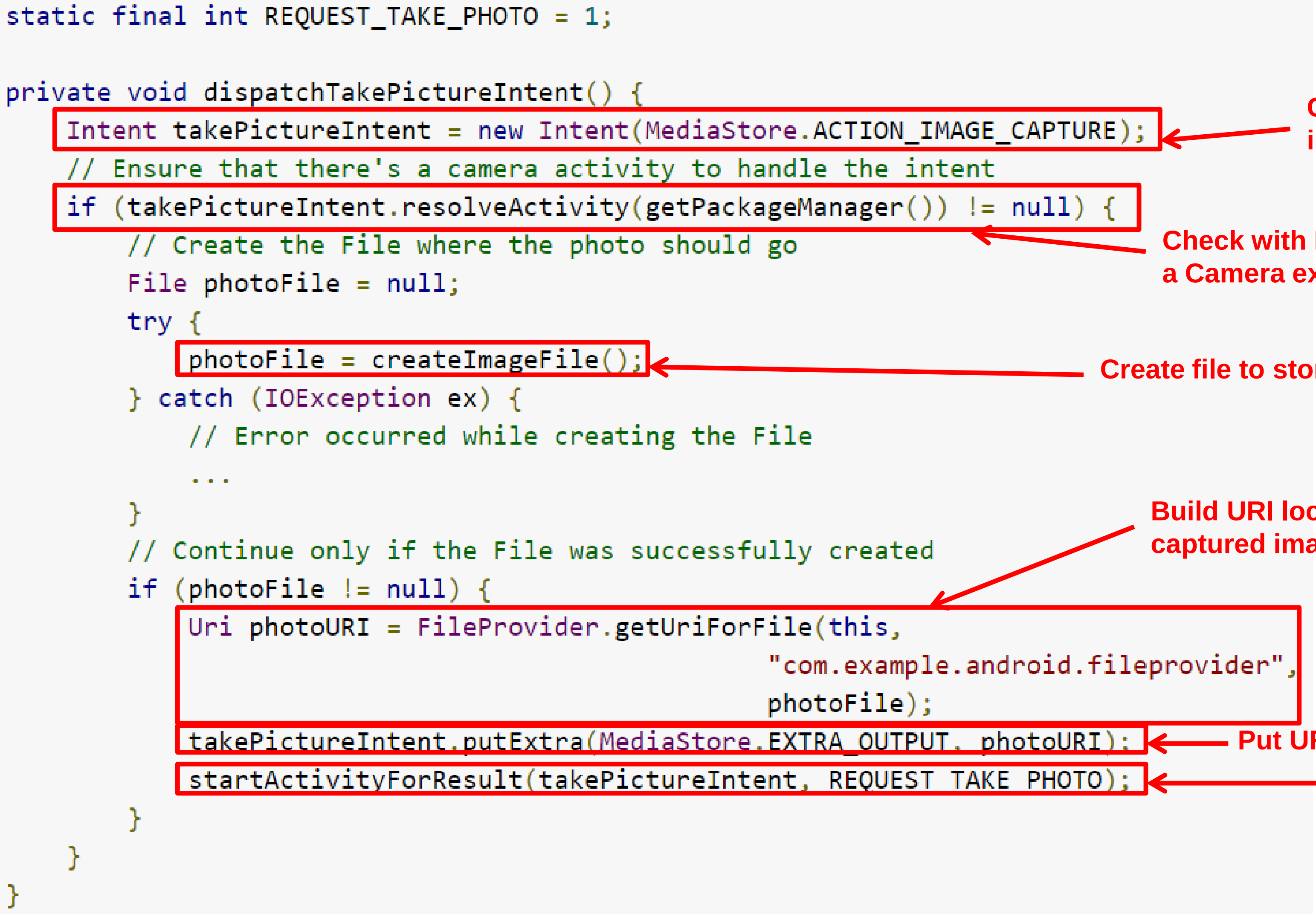
  2. **Private storage** (Seen by only your app, deleted when your app uninstalls):
     - **getExternalFilesDir( )**
- Either way, need phone owner's permission to write to external storage
- In AndroidManifest.xml, make the following declaration



```
<manifest ...>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    ...
</manifest>
```

Saving Full Sized Photo

Ref: https://developer.android.com/training/camera/photobasics.html



```
static final int REQUEST_TAKE_PHOTO = 1;

private void dispatchTakePictureIntent() {
    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    // Ensure that there's a camera activity to handle the intent
    if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
        // Create the File where the photo should go
        File photoFile = null;
        try {
            photoFile = createImageFile();
        } catch (IOException ex) {
            // Error occurred while creating the File
            ...
        }
        // Continue only if the File was successfully created
        if (photoFile != null) {
            Uri photoURI = FileProvider.getUriForFile(this,
                                                  "com.example.android.fileprovider",
                                                  photoFile);
            takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, photoURI);
            startActivityForResult(takePictureIntent, REQUEST_TAKE_PHOTO);
        }
    }
}
```

Create new intent for image capture

Check with PackageManager that a Camera exists on this phone

Create file to store full-sized image

Build URI location to store captured image (E.g. file//xyz)

Put URI into Intents extra

Take picture

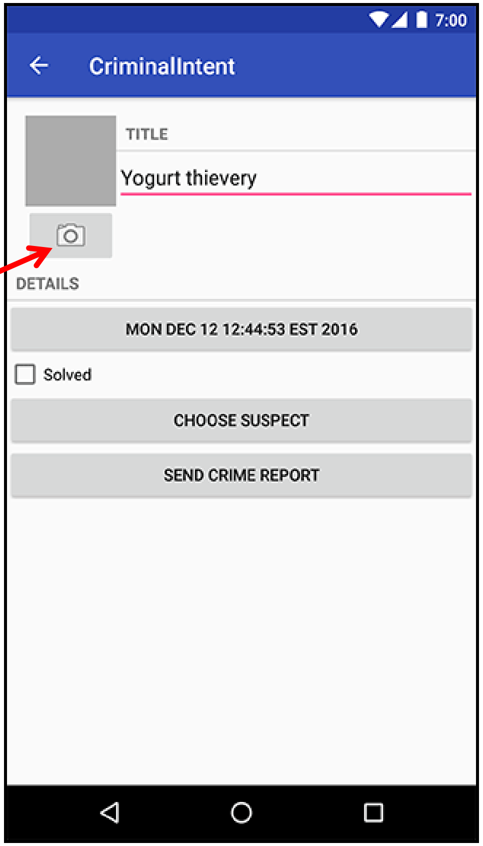
Taking Pictures: Bigger Example

Taking Pictures with Intents

Ref: Ch 16 Android Nerd Ranch 3rd edition

- Would like to take picture of “Crime” to document it
- Use implicit intent to start Camera app from our CrimeIntent app
- **Recall:** Implicit intent used to call component in different activity

Click here to take picture



7:00

CriminalIntent

TITLE

Yogurt thievery

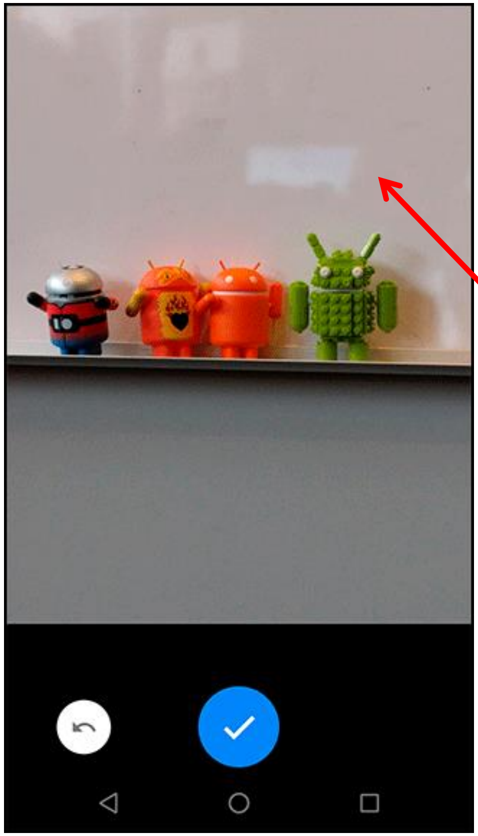
DETAILS

MON DEC 12 12:44:53 EST 2016

□ Solved

CHOOSE SUSPECT

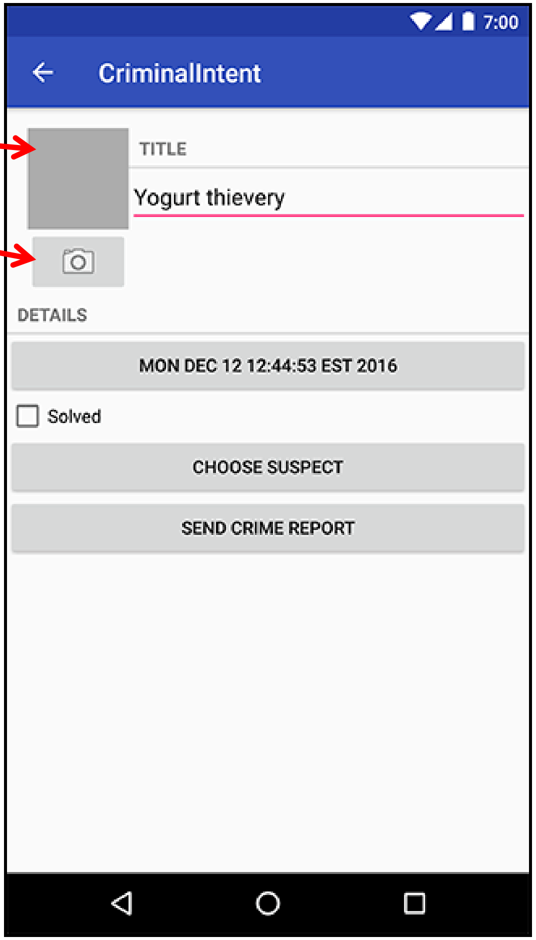
SEND CRIME REPORT



Launches Camera app

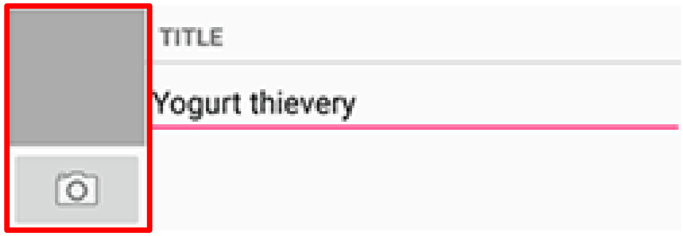
Create Placeholder for Picture

- Modify layout to include
  - ImageView for picture
  - Button to take picture



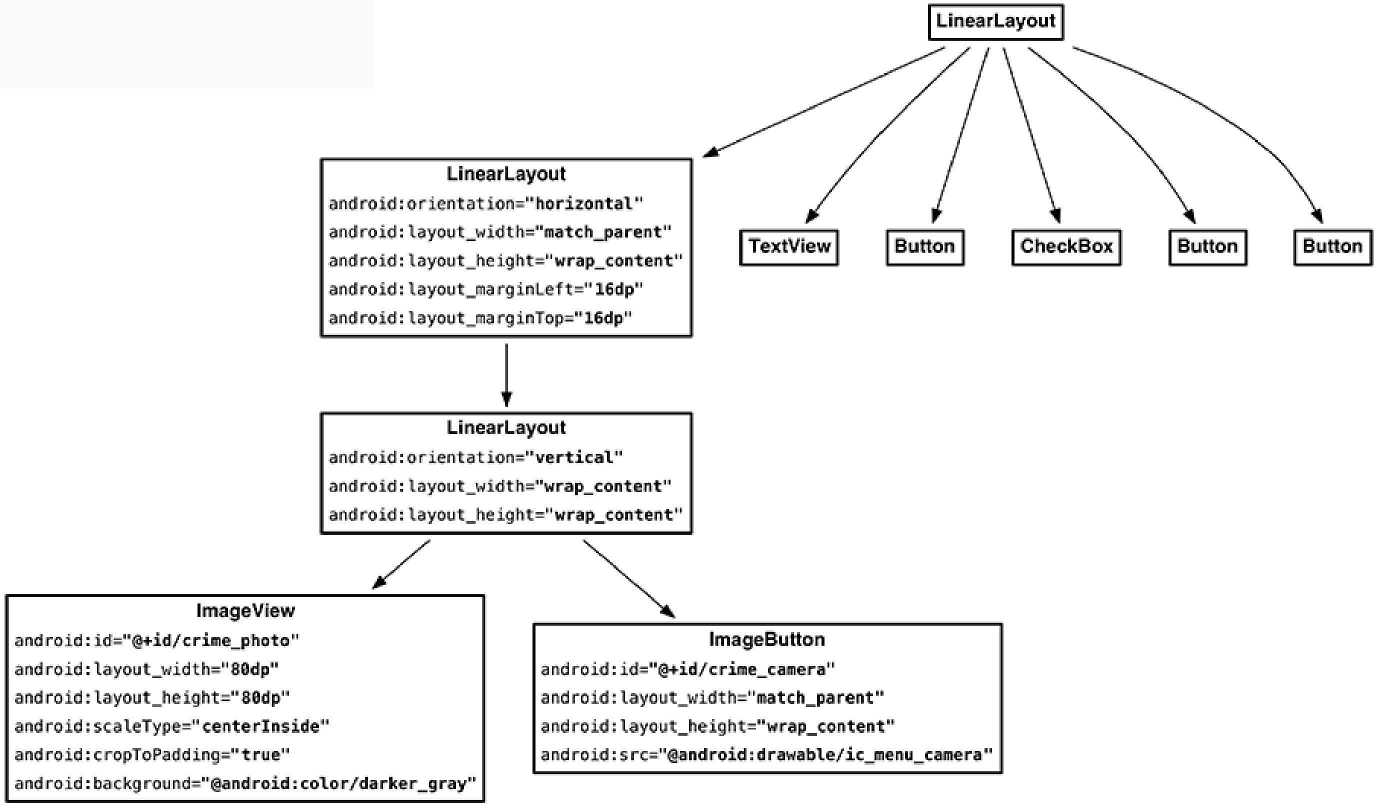
Create Layout for Thumbnail and Button

- First, build out left side



TITLE

Yogurt thievery



LinearLayout

LinearLayout

```
android:orientation="horizontal"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginLeft="16dp"
android:layout_marginTop="16dp"
```

TextView

Button

CheckBox

Button

Button

LinearLayout

```
android:orientation="vertical"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
```

ImageView

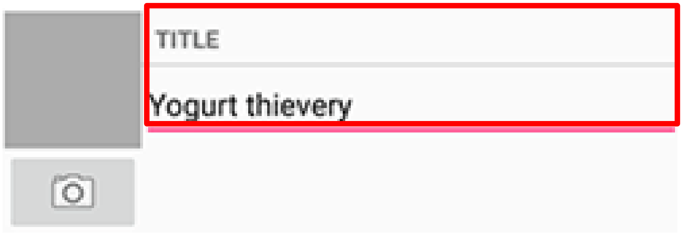
```
android:id="@+id/crime_photo"
android:layout_width="80dp"
android:layout_height="80dp"
android:scaleType="centerInside"
android:cropToPadding="true"
android:background="@android:color/darker_gray"
```

ImageButton

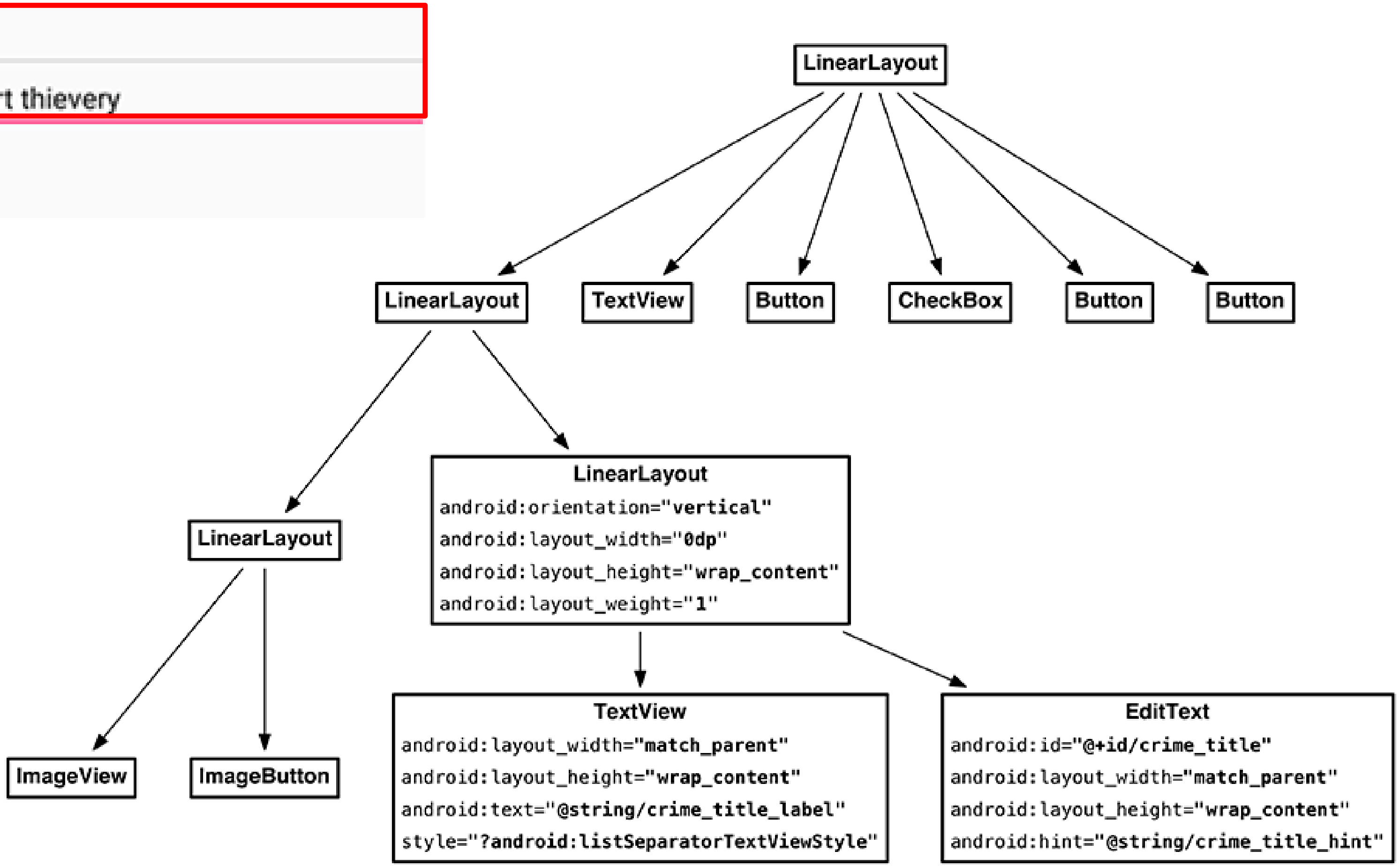
```
android:id="@+id/crime_camera"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:src="@android:drawable/ic_menu_camera"
```

Create Title and Crime Entry EditText

- Build out right side


TITLE

Yogurt thievery


LinearLayout

LinearLayout

TextView

Button

CheckBox

Button

Button

LinearLayout

ImageView

ImageButton

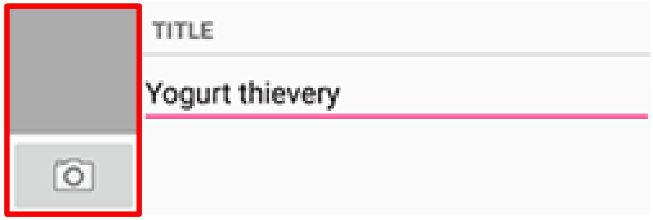
LinearLayout
```
android:orientation="vertical"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_weight="1"
```

TextView
```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="@string/crime_title_label"
style="?android:listSeparatorTextViewStyle"
```

EditText
```
android:id="@+id/crime_title"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="@string/crime_title_hint"
```

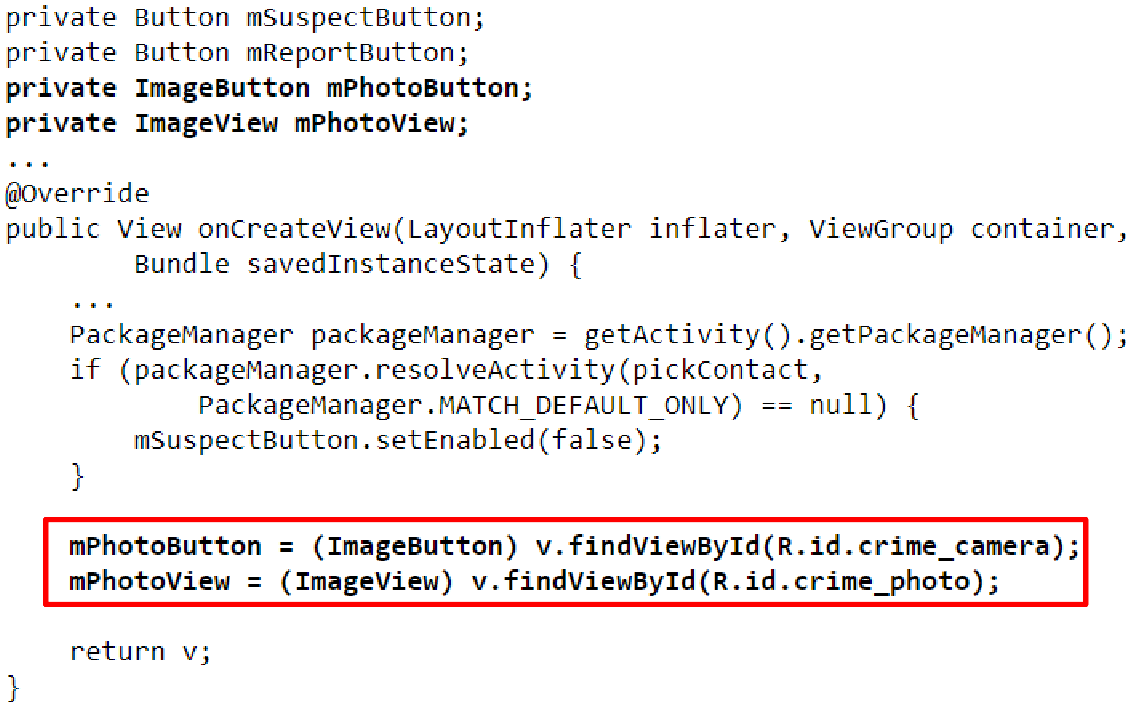
Get Handle of Camera Button and ImageView

- To respond to Camera Button click, in camera fragment, need handles to
  - Camera button
  - ImageView



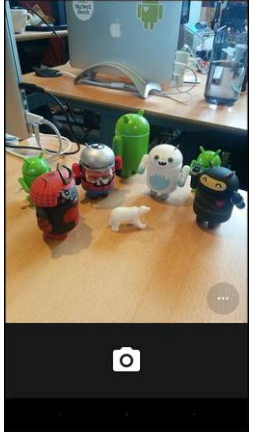
TITLE

Yogurt thievery



```
private Button mSuspectButton;
private Button mReportButton;
private ImageButton mPhotoButton;
private ImageView mPhotoView;
...
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
    ...
    PackageManager packageManager = getActivity().getPackageManager();
    if (packageManager.resolveActivity(pickContact,
            PackageManager.MATCH_DEFAULT_ONLY) == null) {
        mSuspectButton.setEnabled(false);
    }

    mPhotoButton = (ImageButton) v.findViewById(R.id.crime_camera);
    mPhotoView = (ImageView) v.findViewById(R.id.crime_photo);

    return v;
}
```

Firing Camera Intent



```
private static final int REQUEST_DATE = 0;
private static final int REQUEST_CONTACT = 1;
private static final int REQUEST_PHOTO= 2;
...
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
    ...
    mPhotoButton = (ImageButton) v.findViewById(R.id.crime_camera);
    final Intent captureImage = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);

    boolean canTakePhoto = mPhotoFile != null &&
            captureImage.resolveActivity(packageManager) != null;
    mPhotoButton.setEnabled(canTakePhoto);

    mPhotoButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Uri uri = FileProvider.getUriForFile(getActivity(),
                    "com.bignerdranch.android.criminalintent.fileprovider",
                    mPhotoFile);
            captureImage.putExtra(MediaStore.EXTRA_OUTPUT, uri);

            List<ResolveInfo> cameraActivities = getActivity()
                    .getPackageManager().queryIntentActivities(captureImage,
                            PackageManager.MATCH_DEFAULT_ONLY);

            for (ResolveInfo activity : cameraActivities) {
                getActivity().grantUriPermission(activity.activityInfo.packageName,
                        uri, Intent.FLAG_GRANT_WRITE_URI_PERMISSION);
            }

            startActivityForResult(captureImage, REQUEST_PHOTO);
        }
    });

    mPhotoView = (ImageView) v.findViewById(R.id.crime_photo);

    return v;
}
```

Create new intent for image capture

Check with PackageManager that a Camera exists on this phone

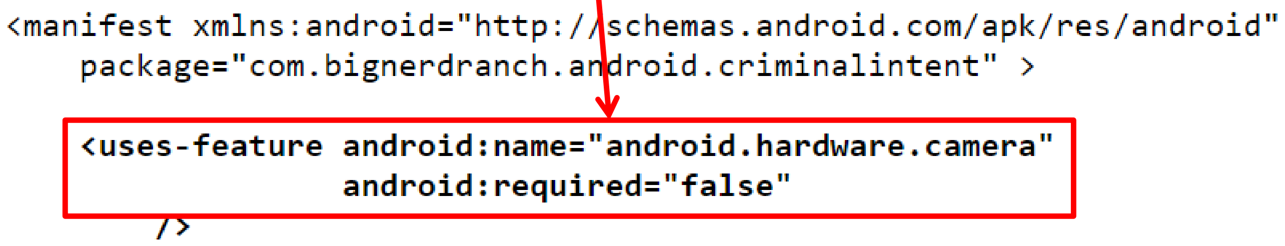
Build Uri location to store image,

Put image URI into Intents extra

Take picture

Declaring Features

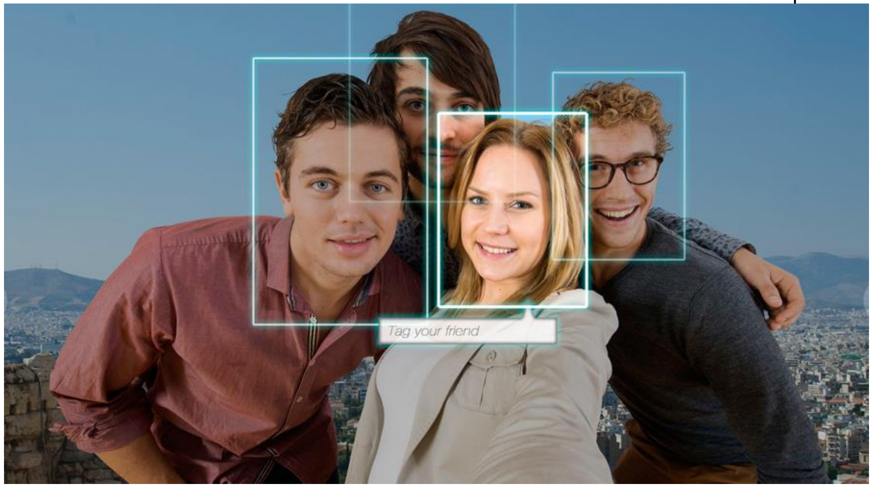
- Declaring “uses-features”.. But “android:required=false” means app prefers to use this feature
- Phones without a camera will still “see” and on Google Play Store and can download this app



```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.bignerdranch.android.criminalintent" >

    <uses-feature android:name="android.hardware.camera"
                  android:required="false"
        />
```

Face Recognition

Face Recognition



- Answers the question:

Who is this person in this picture?

Example answer: John Smith

- Compares unknown face to database of faces with known identity
- Neural networks/deep learning now makes comparison faster

FindFace App: Stalking on Steroids?

- See stranger you like? Take a picture
- App searches 1 billion pictures using neural networks < 1 second
- Finds person's picture, identity, link on VK (Russian Facebook)
  - You can send friend Request
- ~ 70% accurate!
- Can also upload picture of celebrity you like
- Finds 10 strangers on Facebook who look similar, can send friend request

FindFace App

- Also used in law enforcement
  - Police identify criminals on watchlist

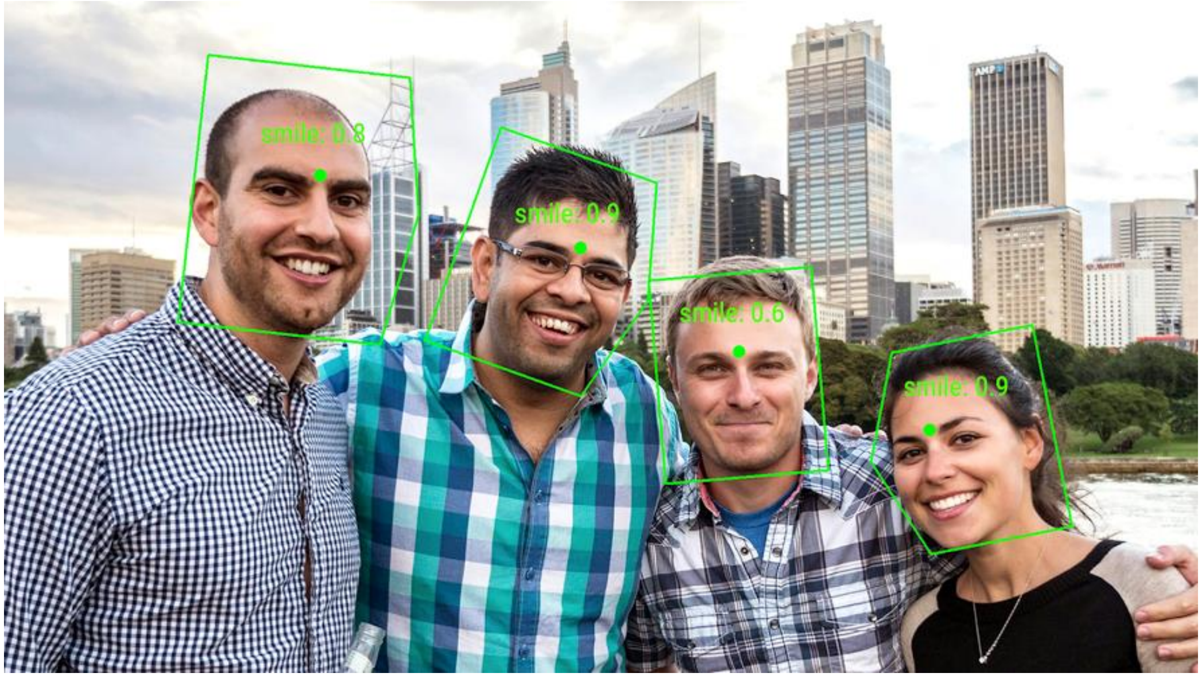
Ref: http://www.computerworld.com/article/3071920/data-privacy/face-recognition-app-findface-may-make-you-want-to-take-down-all-your-online-photos.html

Face Detection

Mobile Vision API

https://developers.google.com/vision/

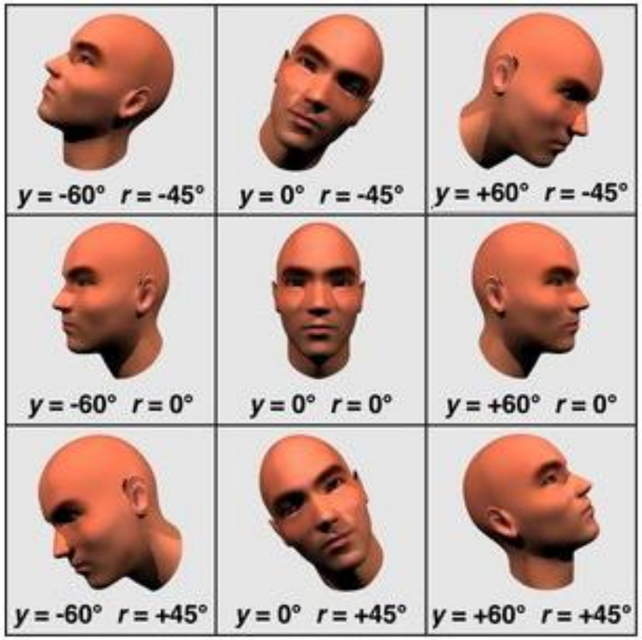
- **Face Detection:** Are there [any] faces in this picture?
- **How?** Locate face in photos and video and
  - **Facial landmarks:** Eyes, nose and mouth
  - **State of facial features:** Eyes open? Smiling?



Face Detection: Google Mobile Vision API

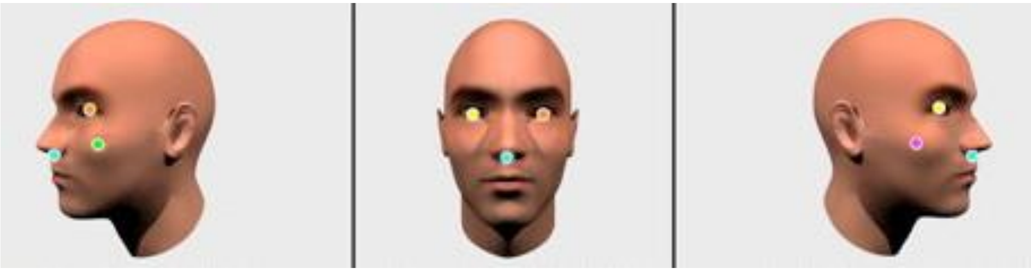
Ref: https://developers.google.com/vision/face-detection-concepts

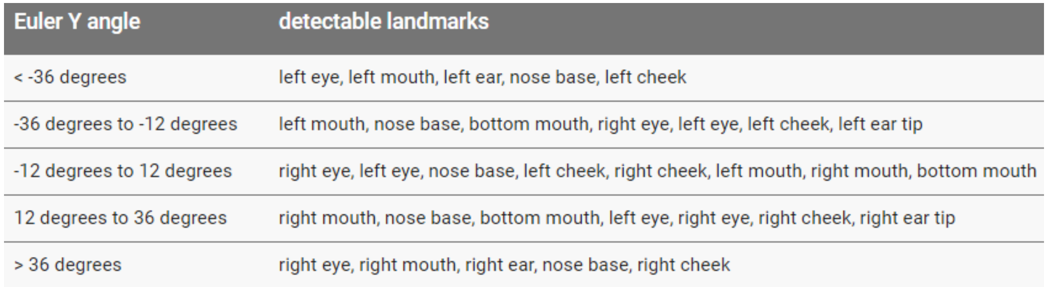
- Detects faces:
  - reported at a position, with size and orientation
  - Can be searched for landmarks (e.g. eyes and nose)



Orientation

Landmarks





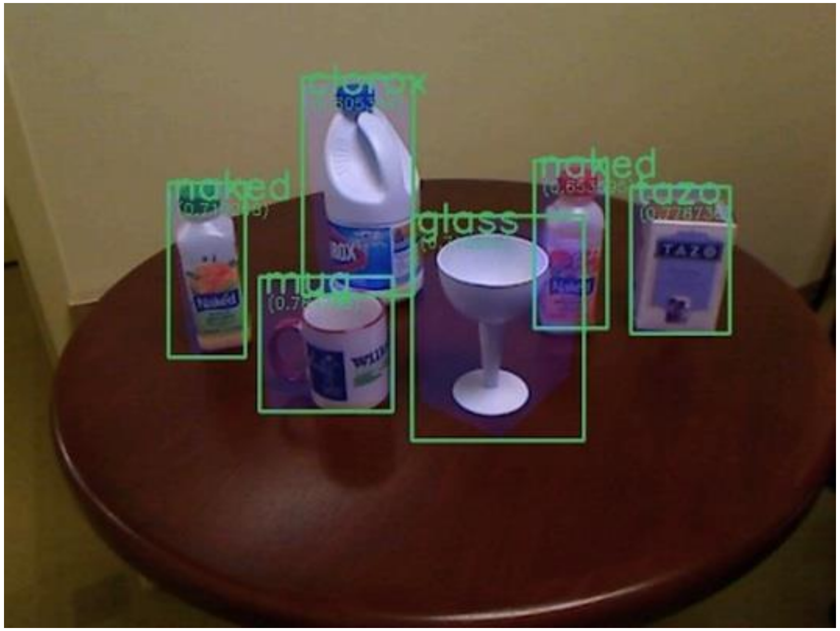
Euler Y angle	detectable landmarks
< -36 degrees	left eye, left mouth, left ear, nose base, left cheek
-36 degrees to -12 degrees	left mouth, nose base, bottom mouth, right eye, left eye, left cheek, left ear tip
-12 degrees to 12 degrees	right eye, left eye, nose base, left cheek, right cheek, left mouth, right mouth, bottom mouth
12 degrees to 36 degrees	right mouth, nose base, bottom mouth, left eye, right eye, right cheek, right ear tip
> 36 degrees	right eye, right mouth, right ear, nose base, right cheek

Google Mobile Vision API

- Mobile Vision API also does:
  - **Face tracking:** detects faces in consecutive video frames
  - **Classification:** Eyes open? Face smiling?
- Classification:
  - Determines whether a certain facial characteristic is present
  - API currently supports 2 classifications: eye open, smiling
  - Results expressed as a confidence that a facial characteristic is present
    - Confidence > 0.7 means facial characteristic is present
    - E.g. > 0.7 confidence means it’s likely person is smiling
- Mobile vision API does face **detection** but NOT **recognition**

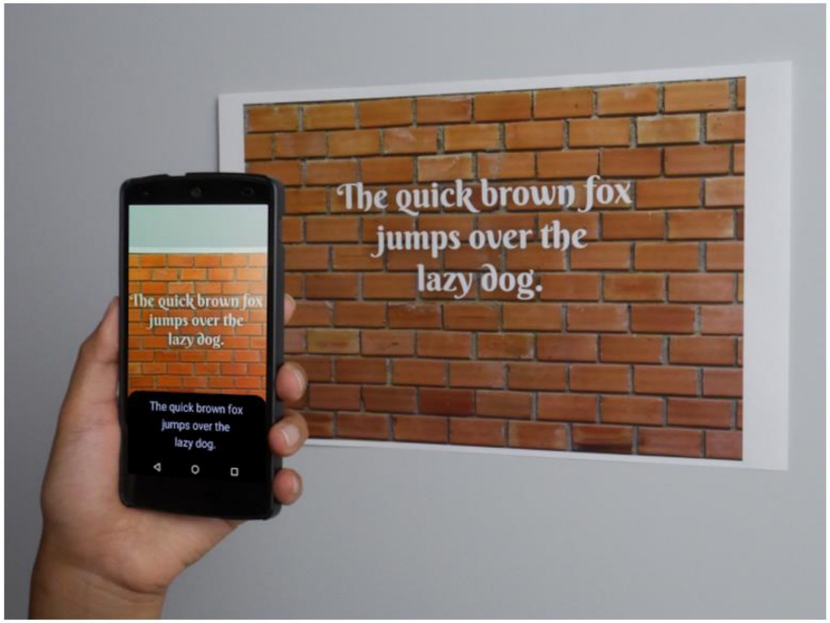
Face Detection

- **Face detection:** Special case of object-class detection
- **Object-class detection task:** find locations and sizes of all objects in an image that belong to a given class.
    - E.g: bottles, cups, pedestrians, and cars
- **Object matching:** Objects in picture compared to objects in database of labelled pictures



Mobile Vision API: Other Functionality

- Barcode scanner
- Recognize text

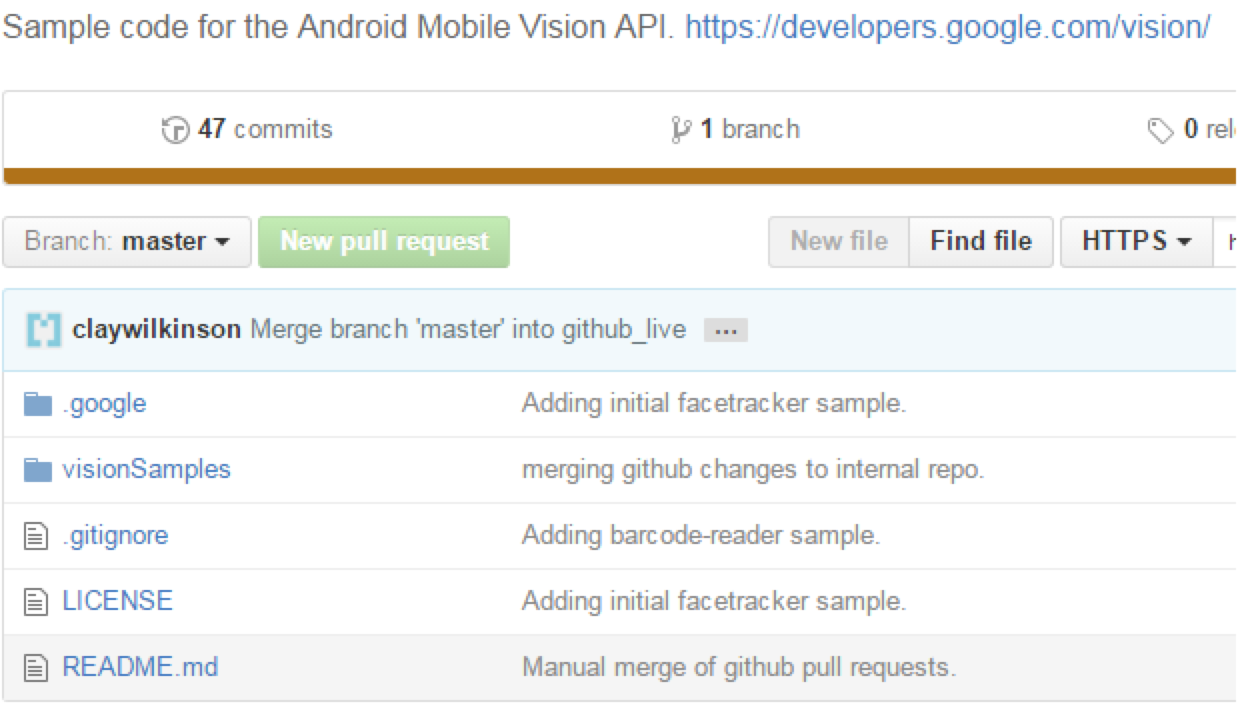


Face Detection Using Google's Mobile Vision API

Getting Started with Mobile Vision Samples

https://developers.google.com/vision/android/getting-started

- Get **Android Play Services SDK** level 26 or greater
- Download mobile vision samples from github



Sample code for the Android Mobile Vision API. https://developers.google.com/vision/

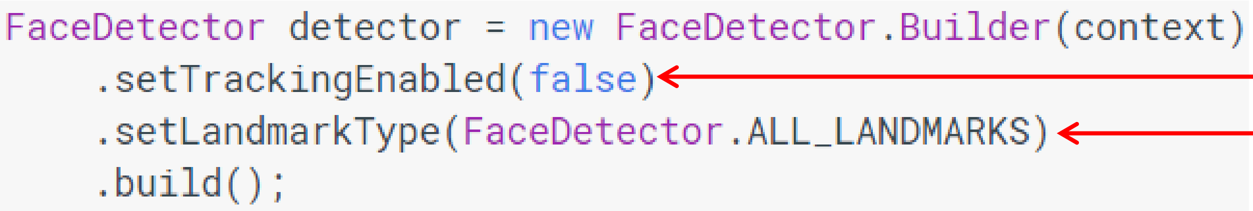
claywilkinson Merge branch 'master' into github_live …

.google	Adding initial facetracker sample.
visionSamples	merging github changes to internal repo.
.gitignore	Adding barcode-reader sample.
LICENSE	Adding initial facetracker sample.
README.md	Manual merge of github pull requests.

Creating the Face Detector

Ref: https://developers.google.com/vision/android/detect-faces-tutorial

- In app’s **onCreate** method, create face detector



```
FaceDetector detector = new FaceDetector.Builder(context)
    .setTrackingEnabled(false)
    .setLandmarkType(FaceDetector.ALL_LANDMARKS)
    .build();
```

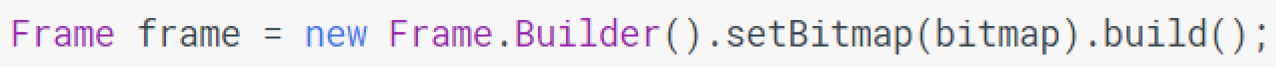
Don’t track points

Detect all landmarks

- **detector** is base class for implementing specific detectors. E.g. face detector, bar code detector
- Tracking finds same points in multiple frames (continuous)
- Detection works best in single images when **trackingEnabled** is false

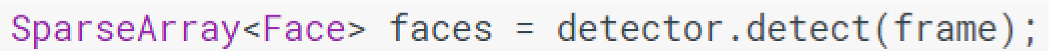
Detecting Faces and Facial Landmarks

- Create Frame (image data, dimensions) instance from bitmap supplied



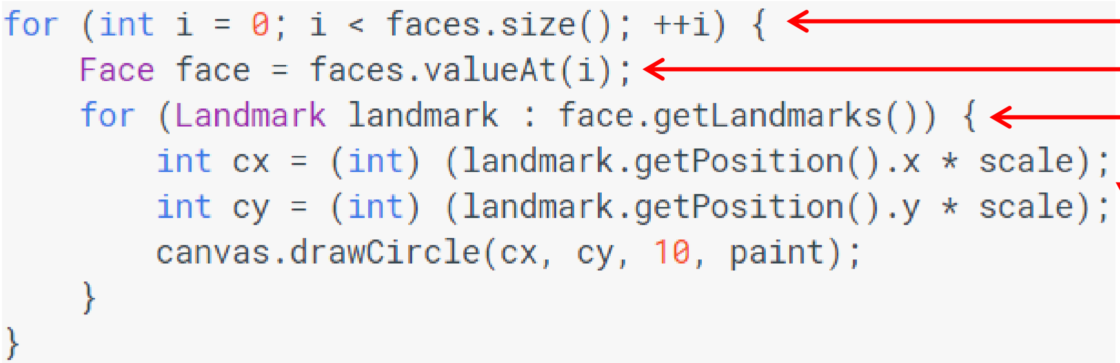
```
Frame frame = new Frame.Builder().setBitmap(bitmap).build();
```

- Call detector synchronously with frame to detect faces



```
SparseArray<Face> faces = detector.detect(frame);
```

- Detector takes **Frame** as input, outputs array of **Faces** detected
- **Face** is a single detected human face in image or video
- Iterate over array of faces, landmarks for each face, and draw the result based on each landmark's position



```
for (int i = 0; i < faces.size(); ++i) {
    Face face = faces.valueAt(i);
    for (Landmark landmark : face.getLandmarks()) {
        int cx = (int) (landmark.getPosition().x * scale);
        int cy = (int) (landmark.getPosition().y * scale);
        canvas.drawCircle(cx, cy, 10, paint);
    }
}
```

Iterate through face array

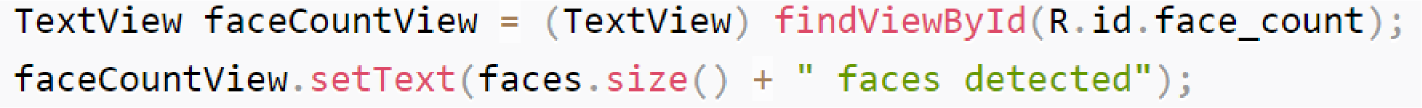
Get face at position i in Face array

Return list of face landmarks (e.g. eyes, nose)

Returns landmark's (x, y) position where (0, 0) is image's upper-left corner

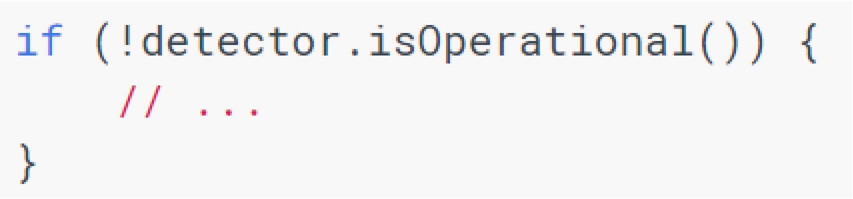
Other Stuff

- To count faces detected, call **faces.size( )**. E.g.



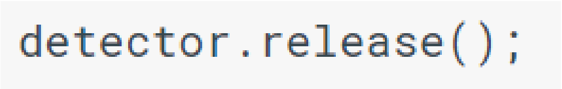
```
TextView faceCountView = (TextView) findViewById(R.id.face_count);
faceCountView.setText(faces.size() + " faces detected");
```

- Querying Face detector’s status



```
if (!detector.isOperational()) {
    // ...
}
```

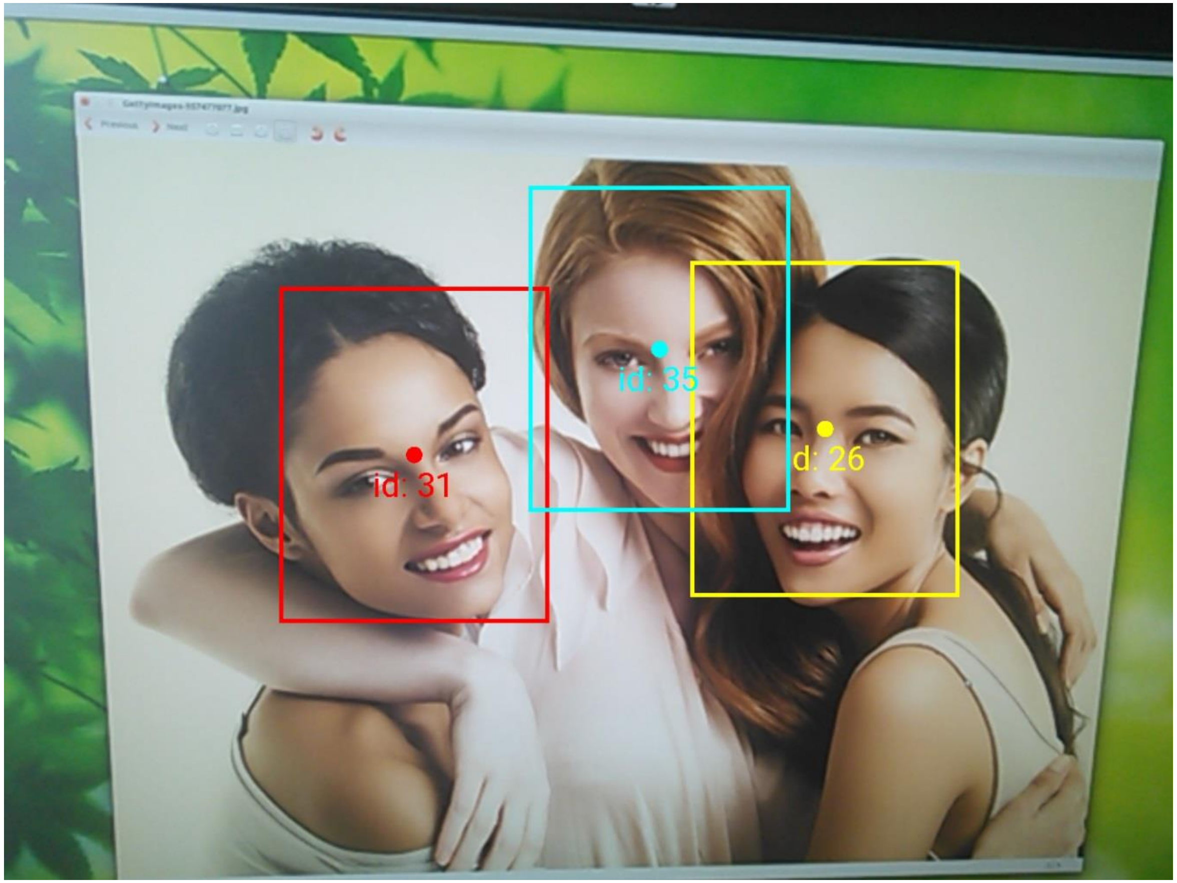
- Releasing Face detector (frees up resources)



```
detector.release();
```

Detect & Track Multiple Faces in Video

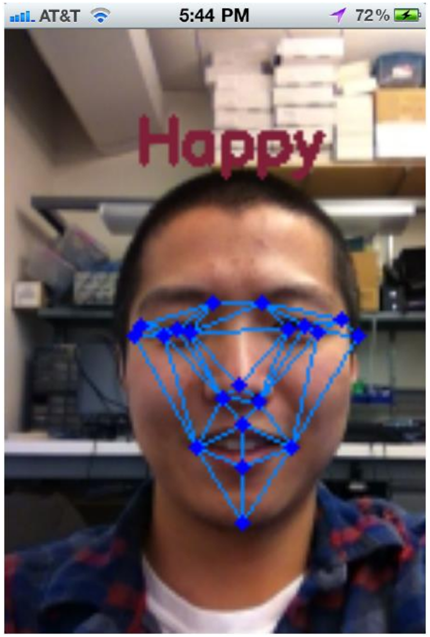
- Can also track multiple faces in image sequences/video, draw rectangle round each one



Face Interpretation

Visage Face Interpretation Engine

- Real-time face interpretation engine for smart phones
  - Tracking user’s 3D head orientation + facial expression

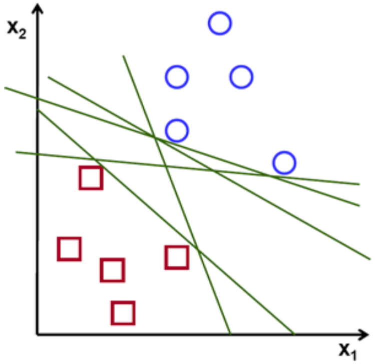


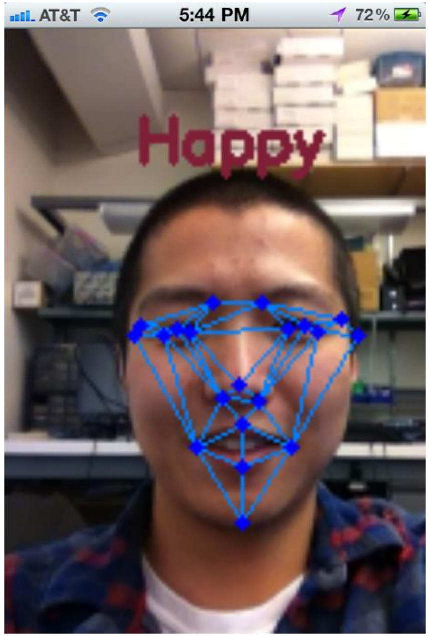
- Facial expression?
  - angry, disgust, fear, happy, neutral, sad, surprise
  - Use? Can be used in Mood Profiler app

Yang, Xiaochao, et al. "Visage: A face interpretation engine for smartphone applications." *Mobile Computing, Applications, and Services Conference*. Springer Berlin Heidelberg, 2012. 149-168.

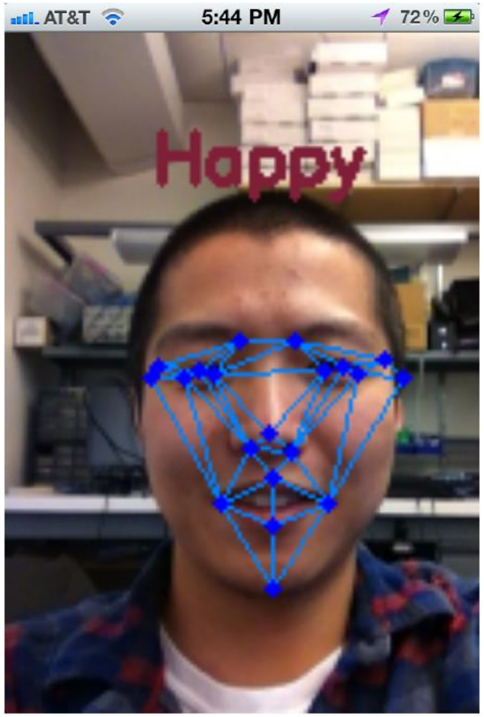
Facial Expression Inference

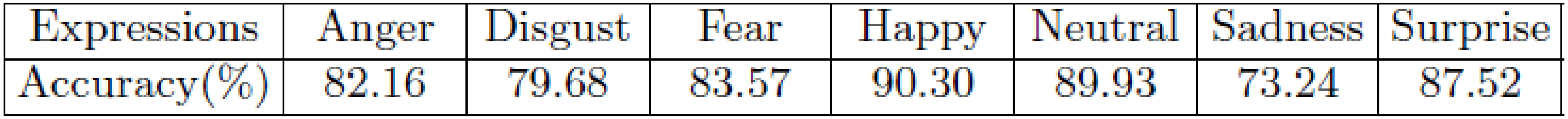
- Active appearance model
  - Describes 2D image as triangular mesh of landmark points
- 7 expression classes: angry, disgust, fear, happy, neutral, sad, surprise
- Extract triangle shape, texture features
- Classify features using Machine learning





Classification Accuracy





Expressions	Anger	Disgust	Fear	Happy	Neutral	Sadness	Surprise
Accuracy(%)	82.16	79.68	83.57	90.30	89.93	73.24	87.52

References

- Google Camera “Taking Photos Simply” Tutorials, http://developer.android.com/training/camera/photobasics.html
- Busy Coder’s guide to Android version 4.4
- CS 65/165 slides, Dartmouth College, Spring 2014
- CS 371M slides, U of Texas Austin, Spring 2014

References

- Android Nerd Ranch, 1st edition
- Busy Coder's guide to Android version 4.4
- CS 65/165 slides, Dartmouth College, Spring 2014
- CS 371M slides, U of Texas Austin, Spring 2014