

# BUILD X: ANDROID

Lecture Five: Putting it All Together

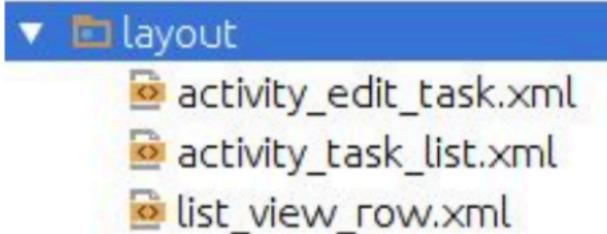
Notes:

<https://tinyurl.com/kcl-tech-android-notes>

# Recap



# Recap – Layouts, Views and View Groups

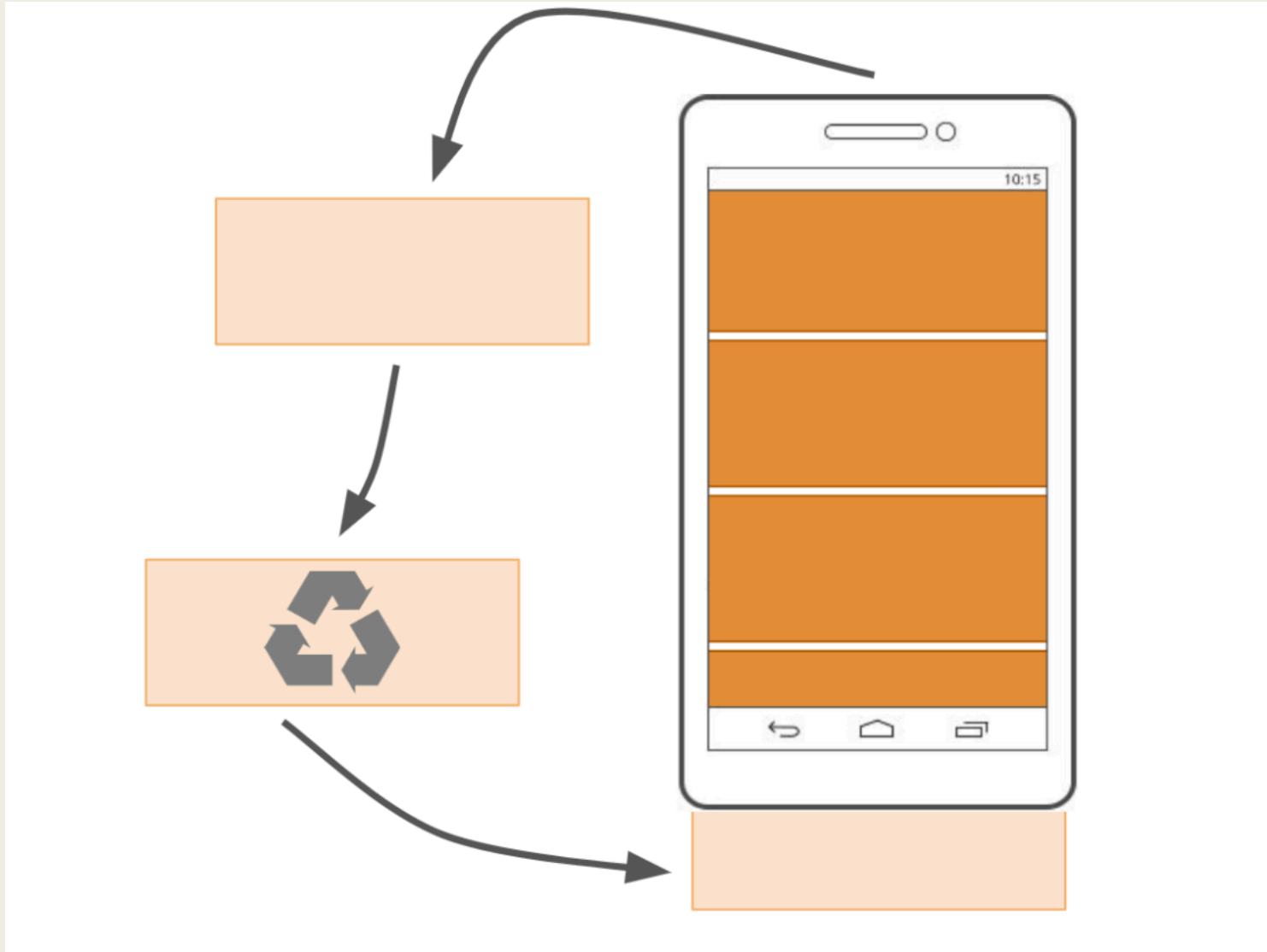


```
<LinearLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical"  
    android:gravity="center">  
  
<ProgressBar  
    android:id="@+id/loading_icon"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    style="android:progressBarStyleInverse"/>  
  
<ScrollView  
    android:id="@+id/main_content"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:paddingTop="16dp"  
    android:paddingBottom="16dp"  
    android:paddingLeft="16dp"  
    android:paddingRight="16dp"  
    android:visibility="gone">
```

# Recap – Events, Intents and the Activity Lifecycle

```
public class TaskListActivity extends AppCompatActivity {  
  
    public class EditTaskActivity extends AppCompatActivity {  
  
        <activity  
            android:name=".TaskListActivity"  
            android:label="@string/app_name">  
            <intent-filter>  
                <action android:name="android.intent.action.MAIN" />  
                <category android:name="android.intent.category.LAUNCHER" />  
            </intent-filter>  
        </activity>  
        <activity android:name=".EditTaskActivity" />
```

# Recap - ListView



# Recap - Models

```
public class Task {  
  
    private int id = -1; // ID defaults to -1, meaning "no ID has been set yet"  
    private String title;  
    private String notes;  
    private DateTime dueDate;  
    private boolean isComplete;
```

# Recap - Databases

```
public class DBHelper extends SQLiteOpenHelper {  
    public static final String DB_NAME = "DB";  
    public static final int DB_VERSION = 1;  
    public DBHelper(Context context) { super(context, DB_NAME, null, DB_VERSION); }  
    @Override  
    public void onCreate(SQLiteDatabase db) {...}  
    @Override  
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {...}
```

# What are we doing?

Course Notes:

<https://tinyurl.com/kcl-tech-android-notes>

We're want to build a basic todo list app, **KCL Tech Todo**.

If you want a preview, it is available on **Google Play**.

- Search for "KCL Tech Todo" on Google or on the Play Store.



Time to put it all together!



What features does our  
TODO app need to have?



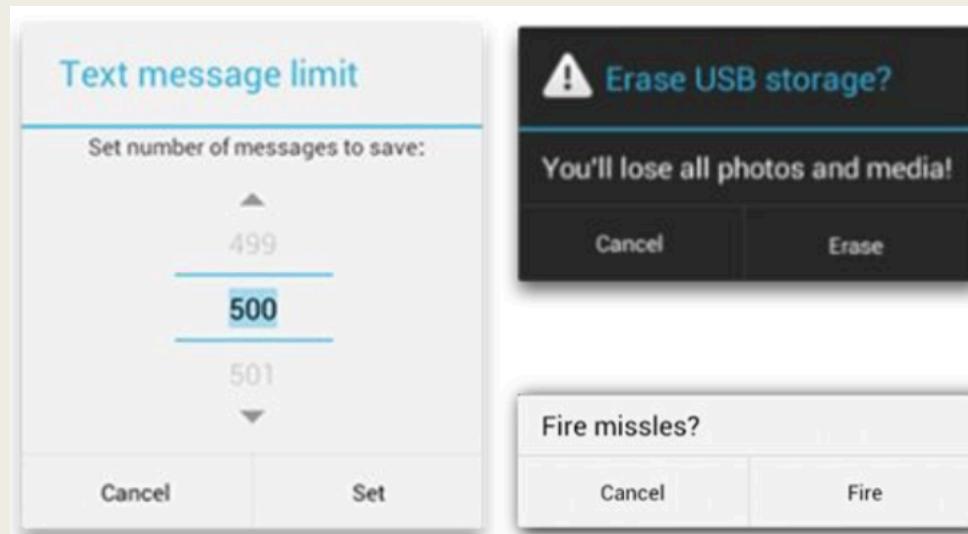
What do YOU have to do?



# Alert Dialogs

An alert dialog is a “pop-up” on Android that can contain pretty much anything that you want.

- Text, an image, input fields, progress bars, buttons, etc.



# Alert Dialogs: Builders

We could create one from scratch, but it's easier to use a **design pattern** called a **builder**.

The **builder pattern** is used in many languages and works like a real builder: you tell them how you want a complex object to look/work, and they build it for you.

```
AlertDialog.Builder builder =  
    new AlertDialog.Builder(<<activity ref>>);
```

```
builder.setOption( ... )  
    .setOption( ... )  
    .setOption( ... );
```

```
AlertDialog dialog = builder.create();  
dialog.show();
```

```
AlertDialog.Builder builder =  
    new AlertDialog.Builder(<<activity ref>>);
```

```
builder.setOption( ... )  
    .setOption( ... )  
    .setOption( ... );
```

```
AlertDialog dialog = builder.create();  
dialog.show();
```

```
AlertDialog.Builder builder =  
    new AlertDialog.Builder(<<activity ref>>);
```

```
builder.setOption( ... )  
    .setOption( ... )  
    .setOption( ... );
```

```
AlertDialog dialog = builder.create();  
dialog.show();
```

```
AlertDialog.Builder builder =  
    new AlertDialog.Builder(<<activity ref>>);  
  
builder.setOption( ... )  
    .setOption( ... )  
    .setOption( ... );  
  
AlertDialog dialog = builder.create();  
dialog.show();
```

```
AlertDialog.Builder builder =  
    new AlertDialog.Builder(<<activity ref>>);  
  
builder.setOption( ... )  
    .setOption( ... )  
    .setOption( ... );  
  
AlertDialog dialog = builder.create();  
dialog.show();
```