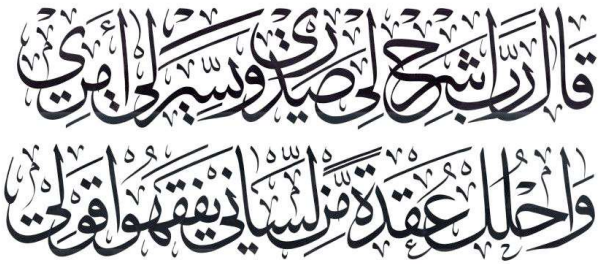
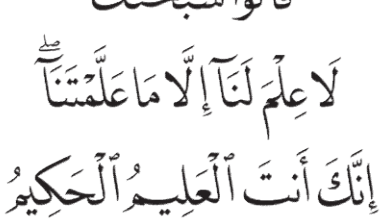
بِسْمِ اللهِ الرَّحْمٰنِ الرَّحِيْمِ



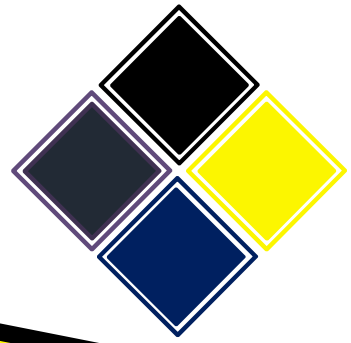


Activities and Intents

Mobile Computing

LECTURE 07

Activities and Intents



Instructor:

Maulana Haq Nawaz

دعا


((اللَّهُمَّ إِنِّي أَسْأَلُكَ
الـمُعَافَـاةَ
فِي الدُّنيَا و الآخِرَةِ))
مجلس الذكر
عن أبي هريرة ﷺ أن النبي ﷺ قال : ((ما من دعوة يدعو بها العبد
أفضل من : اللهم إني أسألك المعافاة ...))
صحيح الجامع (5703)

تصحیح نیت

حضرت محمد صلی اللہ علیہ وسلم نے فرمایا

اِنَّمَا الْاَعْمَالُ بِالنِّـیَّاتِ

ترجمہ

اعمال کا دارومدار نیتوں پر

Little Efforts Daily Will Make You the Greatest

To systematically learn and get excellence in any concept / subject

- روز کا کام روز کریں
- اک مہینے کا کھانا ایک دن میں نہیں کھایا جا سکتا،
  ایسے ہی ایک مہینے کا کام ایک دن میں نہیں ہو سکتا

Little Efforts Daily Will Make You the Greatest

Importance of completing tasks on daily basis

Main Reason of Failure in Life

- یہ کام کل کریں گے
- جو کام کبھی بھی ہو سکتا ہے وہ کبھی نہیں ہوتا
- زندگی ایک دن ہے اور وہ ہے آج۔ زندگی میں کل نام کی کوئی چیز نہیں ہے
- جو دن آپ کی زندگی سے چلا گیا اب واپس نہیں آئے گا
- آج کا کام آج ہی ہوسکتا ہے
- جو گز گیا وہ آنا نہیں ، آنے والے دن کا پتہ نہیں ، آج میدان جما ہے تو اینے جوہر دکھاؤ

How to Achieve BIG Goals in Life

Balanced Life is Ideal Life ☺

To achieve BIG Goals in Life

- **Make a Schedule of 24 Hours with a focus on Five main components of Human Life**
  - **Health**
    - **Physical Health**
    - **Mental Health**
    - **Social Health**
  - **Spirituality**
  - **Work**
  - **Family**
  - **Friends**

جو کام کریں دل سے کریں

کام کرنا.

خوشی خوشی کام کرنا.

اللہ کو ساتھ لے کرخوشی خوشی کام کرنا.

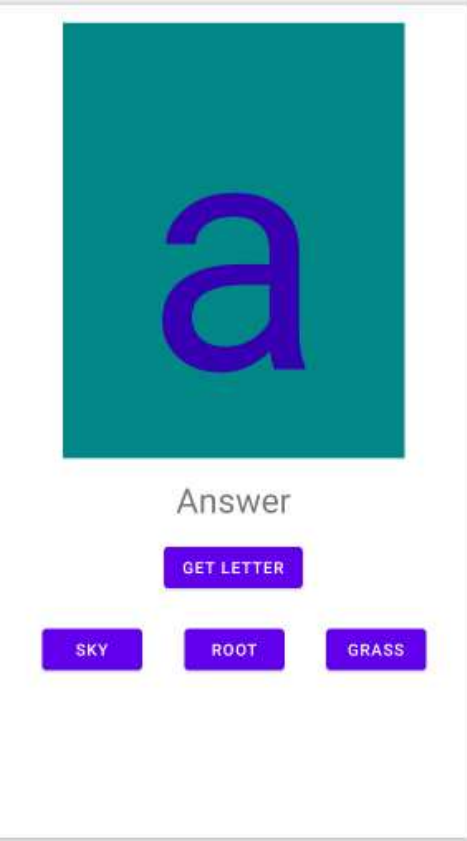
آیت: إِيَّاكَ نَعْبُدُ وإِيَّاكَ نَسْتَعِينُ

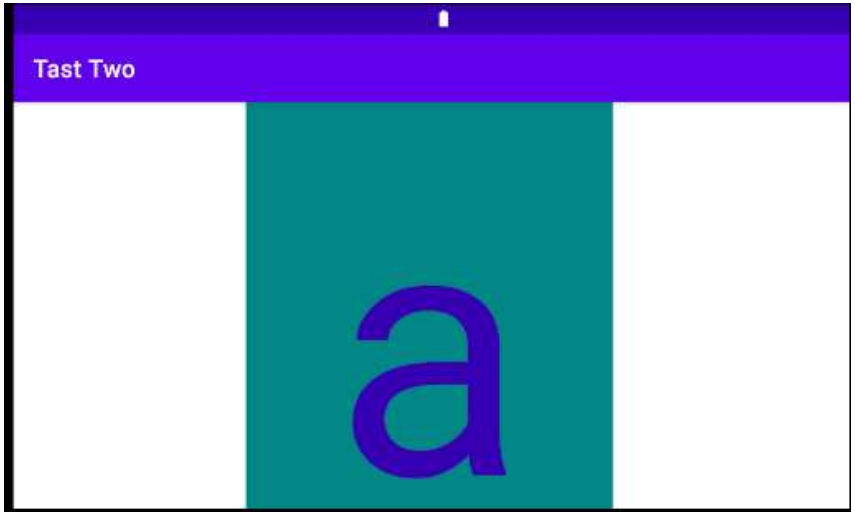
ترجمہ: یا اللہ ہم تیری ہی عبادت کرتے ہیں.
اور تجھ ہی سے مد د مانگتے ہیں

Lecture Outline

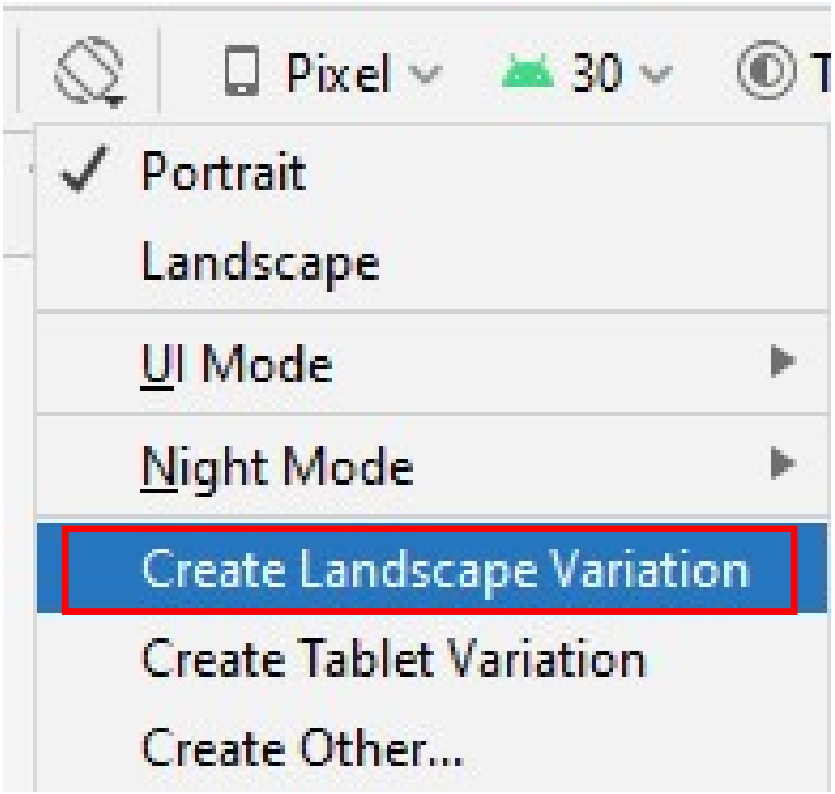
1. Revision
2. Activity
3. Function of Activity
4. Addition of New Activity
5. Manifest file
6. Intents
7. Explicit and implicit intents
8. Working of Activities
9. Activity Lifecycle
10. Stop Complaining! Stop Criticizing! Let's Start Contributing
11. Lecture Summary

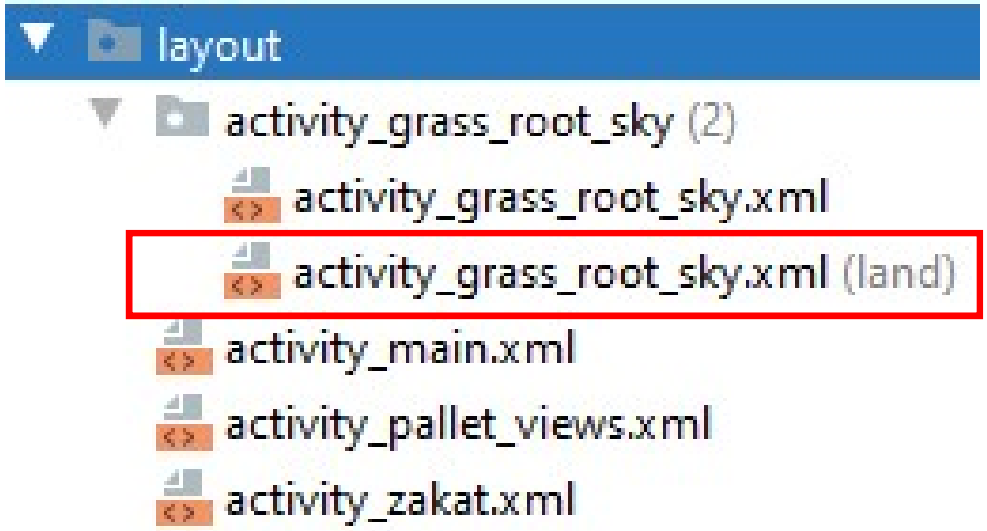
Issue we Find previously





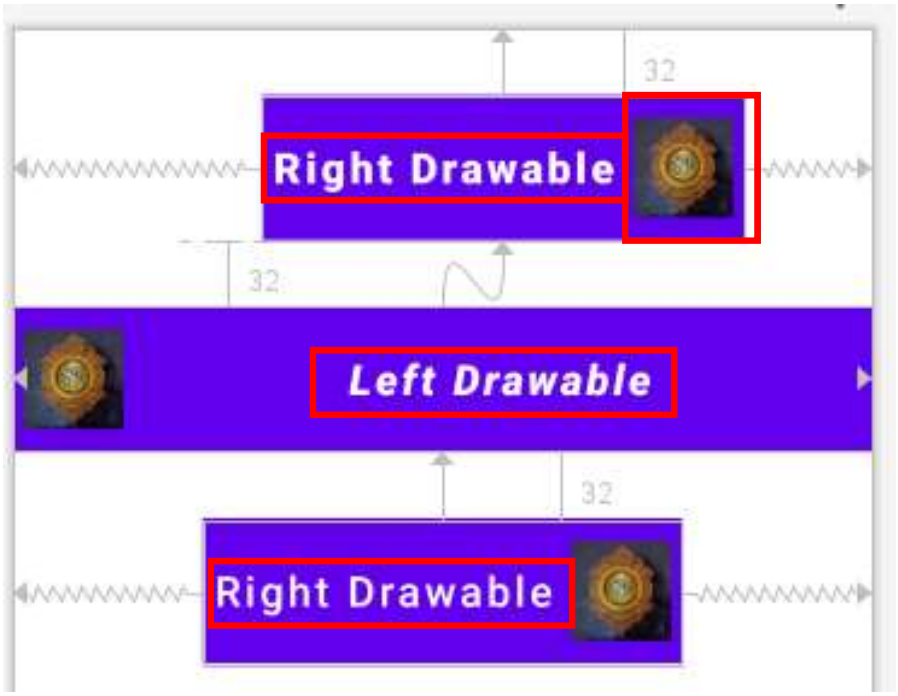
Create Landscape View





Create Landscape View

Button



Button Code

```
<Button
    android:id="@+id/simpleButton1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:background="@color/purple_200"
    android:drawableRight="@drawable/quran_fourtyeight"
    android:hint="Right Drawable"
    android:padding="5dp"
    android:textColorHint="#fff"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.66"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

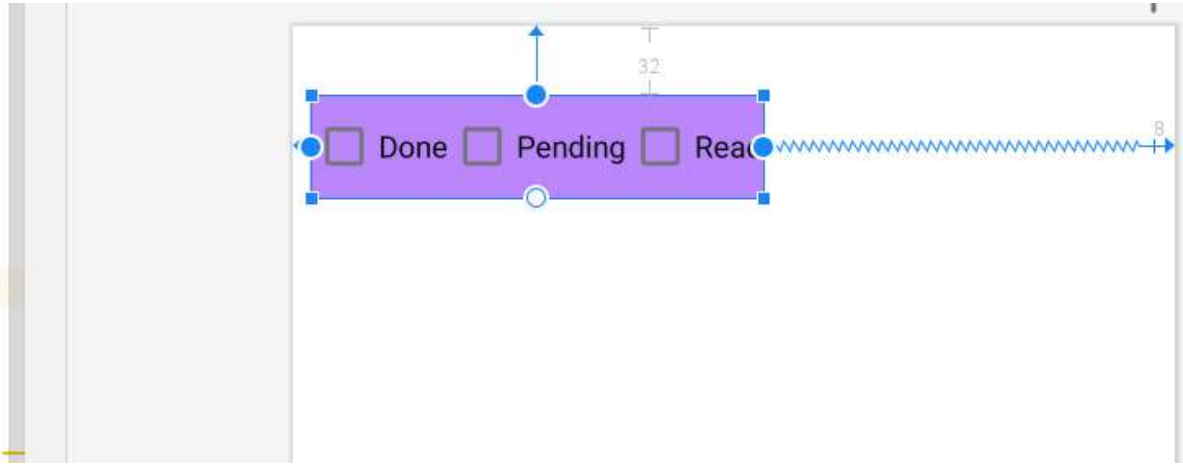
```
<Button
    android:id="@+id/simpleButton2"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_centerInParent="true"
    android:layout_marginTop="32dp"
    android:background="@color/white"
    android:drawableLeft="@drawable/quran_fourtyeight"
    android:hint="Left Drawable"
    android:padding="5dp"
    android:textColorHint="#fff"
    android:textSize="20sp"
    android:textStyle="bold|italic"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/simpleButton1" />
```

Button Code

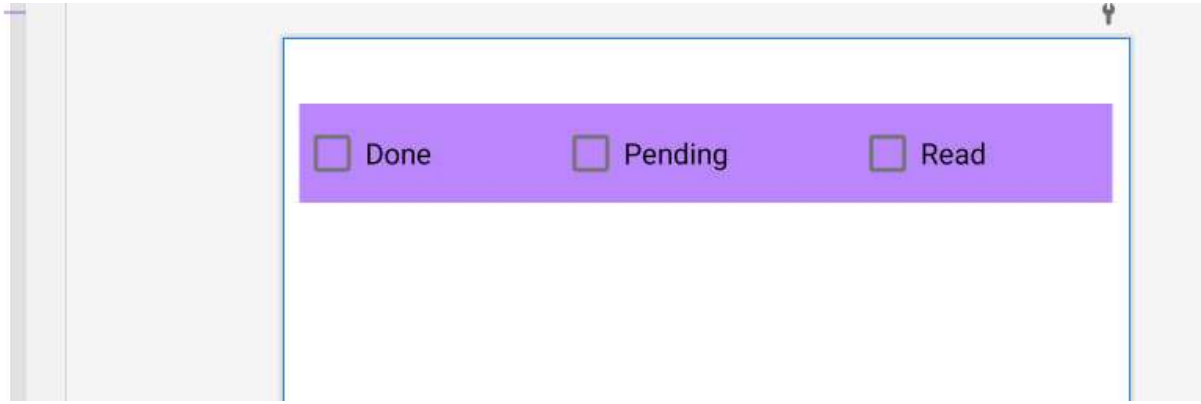
```
<Button
    android:id="@+id/simpleButton3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:background="@color/purple_200"
    android:drawableRight="@drawable/quran_fourtyeight"
    android:hint="Right Drawable"
    android:padding="5dp"
    android:textColorHint="#fff"
    android:textSize="20sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/simpleButton2" />
```

LinearLayout CheckBox

```
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="32dp"
    android:layout_marginEnd="8dp"
    android:background="@color/purple_200"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toStartOf="parent"
```

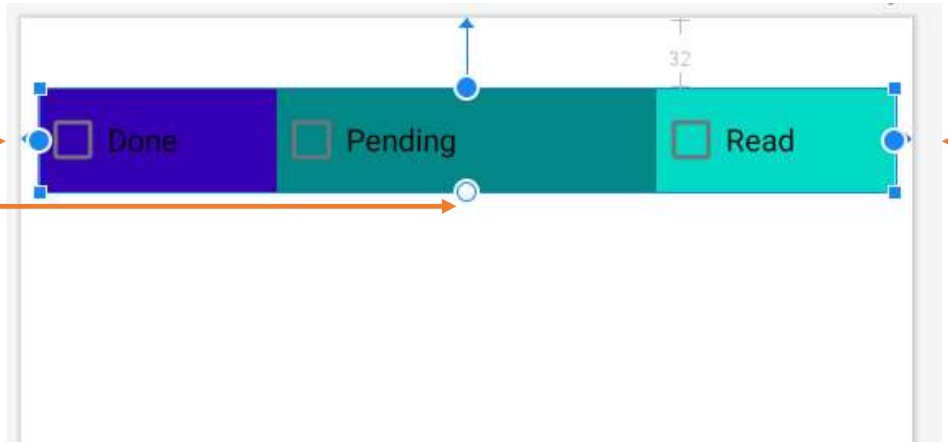



```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="32dp"
    android:layout_marginEnd="8dp"
    android:background="@color/purple_200"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
```



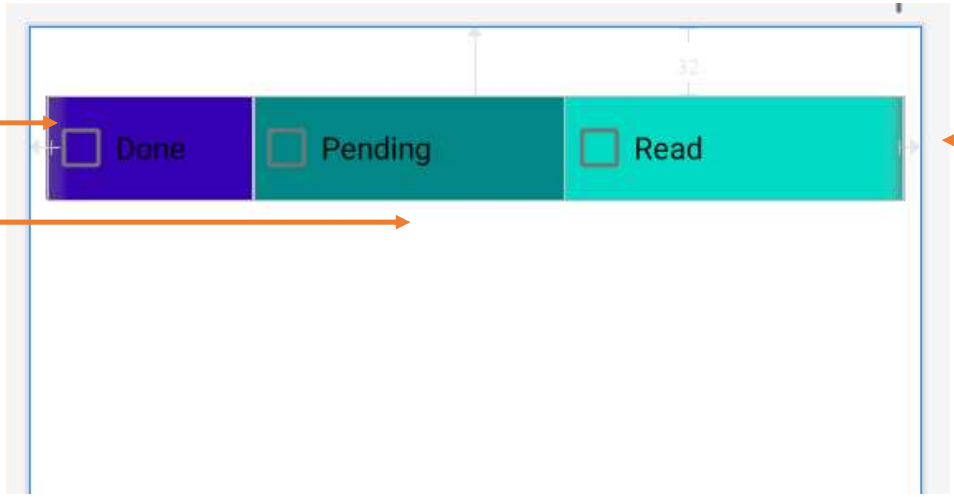
LinearLayout CheckBox

```
<CheckBox
    android:id="@+id/checkBoxDone"
    android:background="@color/purple_700"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Done" />

<CheckBox
    android:id="@+id/checkBoxPending"
    android:background="@color/teal_700"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:text="Pending" />

<CheckBox
    android:id="@+id/checkBoxRead"
    android:background="@color/teal_200"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Read" />
```



LinearLayout CheckBox

```
<CheckBox
    android:id="@+id/checkBoxDone"
    android:background="@color/purple_700"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Done" />

<CheckBox
    android:id="@+id/checkBoxPending"
    android:background="@color/teal_700"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:text="Pending" />

<CheckBox
    android:id="@+id/checkBoxRead"
    android:background="@color/teal_200"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="3"
    android:text="Read" />
```



LinearLayout CheckBox Code

```
<LinearLayout
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:background="@color/purple_200"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/simpleButton3">

    <CheckBox
        android:id="@+id/checkBox4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Done" />

    <CheckBox
        android:id="@+id/checkBox5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Pending" />

    <CheckBox
        android:id="@+id/checkBox3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Read" />

</LinearLayout>
```

Implements OnClickListenerView

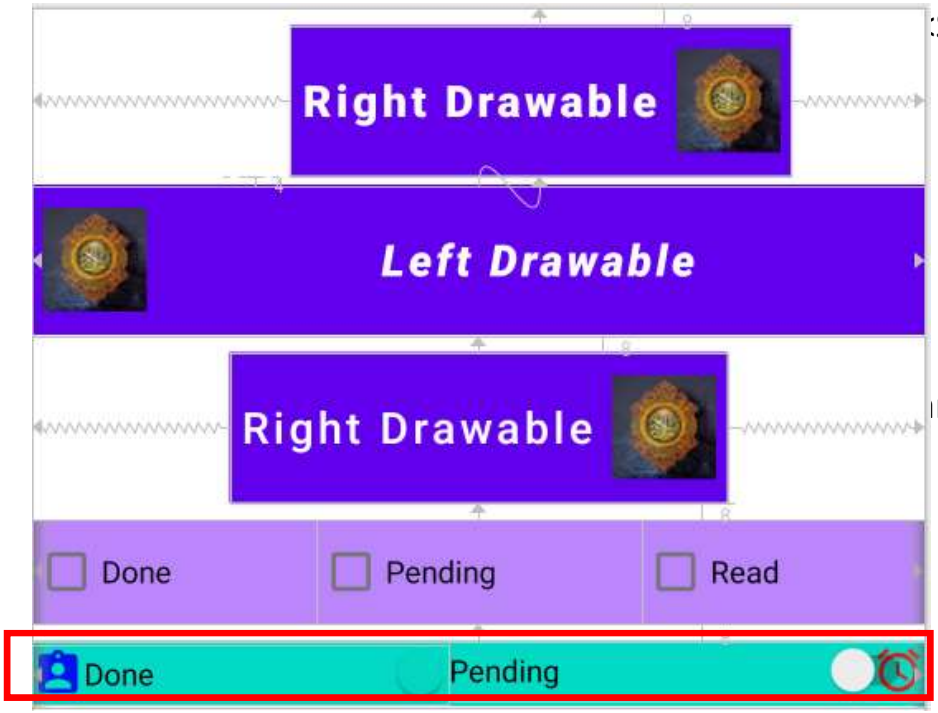
```
public class PalletViews extends AppCompatActivity implements View.OnClickListener {

 CheckBox checkBoxDone, checkBoxPending, checkBoxRead;

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

      checkBoxDone = findViewById(R.id.checkBoxDone);
      checkBoxDone.setOnClickListener(this);
   }

   @Override
   public void onClick(View v) {
   }
```

implements OnClickListenerView.

```
public class PalletViews extends AppCompatActivity implements
View.OnClickListener {
   CheckBox checkBoxDone, checkBoxPending, checkBoxRead;
   Button simpleButton2;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_pallet_views);

      checkBoxDone = findViewById(R.id.checkBoxDone);
      checkBoxDone.setOnClickListener(this);

      checkBoxPending = findViewById(R.id.checkBoxPending);
      checkBoxPending.setOnClickListener(this);

      checkBoxRead = findViewById(R.id.checkBoxRead);
      checkBoxRead.setOnClickListener(this);

      simpleButton2 = findViewById(R.id.simpleButton2);
   }
```

```
@Override
   public void onClick(View v) {
      switch (v.getId( )) {
         case R.id.checkBoxDone:
            if (checkBoxDone.isChecked())
               simpleButton2.setText("Check Box Done");
            break;
         case R.id.checkBoxPending:
            if (checkBoxPending.isChecked())
               simpleButton2.setText("Check Box Pending");
            break;
         case R.id.checkBoxRead:
            if (checkBoxRead.isChecked())
               simpleButton2.setText("Check Box Read");
            break;
      }
   }
```

LinearLayout Switch



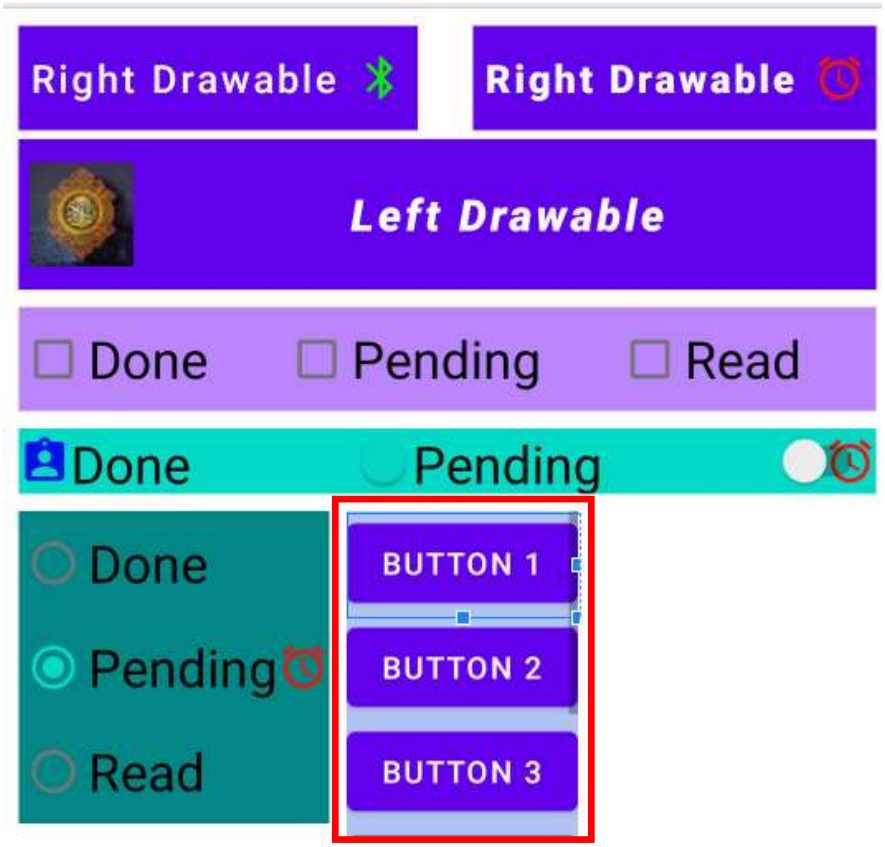

```
:Switch
        android:id="@+id/switchDone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_weight="1"
        android:checked="true"
```


```
ndroid:drawableStart="@drawable/ic_baseline_assignment_ind_24"
        android:text="Done" />
```

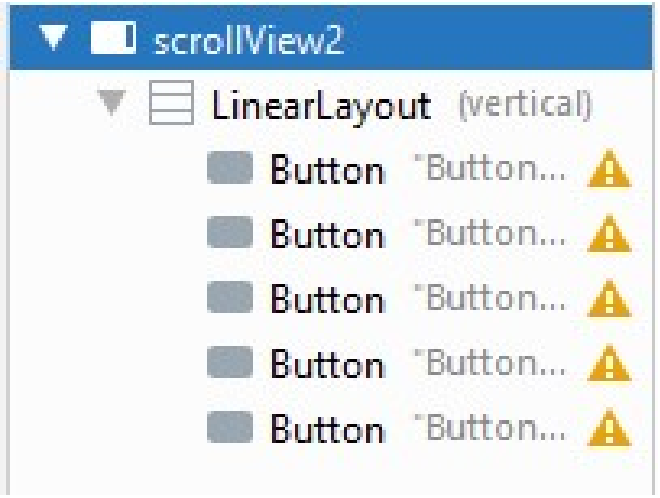
```
<Switch
        android:id="@+id/switchPending"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
```


```
    android:drawableRight="@drawable/ic_baseline_alarm_24"
        android:text="Pending" />
```

ScrollView


Right Drawable
Right Drawable
Left Drawable
Done
Pending
Read
Done
Pending
Done
Pending
Read
BUTTON 1
BUTTON 2
BUTTON 3


BUTTON 1
BUTTON 2
BUTTON 3


scrollView2
LinearLayout (vertical)
Button "Button...
Button "Button...
Button "Button...
Button "Button...
Button "Button...

ScrollView Code

```
<ScrollView
      android:id="@+id/scrollView2"
      android:layout_width="wrap_content"
      android:layout_height="150dp"
      android:layout_marginStart="8dp"
      android:layout_marginTop="8dp"
      android:scrollbars="vertical"
      app:layout_constraintStart_toEndOf="@+id/linearLayout3"

app:layout_constraintTop_toBottomOf="@+id/linearLayout2">

      <LinearLayout
         android:layout_width="wrap_content"
         android:layout_height="150dp"
         android:background="#B1C3F1"
         android:orientation="vertical"
         app:layout_constraintEnd_toEndOf="parent"
         app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toEndOf="@+id/linearLayout3"
   app:layout_constraintTop_toBottomOf="@+id/linearLayout2">
<Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 1" />
         <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 2" />
         <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 3" />
         <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 4" />
         <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 5" />
      </LinearLayout>
</ScrollView>
```

Activity

- Desktop paradigm vs Android
- One App invokes another
- Splash Activity
- Main Activity / Entry Point

Activity

- An Activity is an application component
- Represents one window, one hierarchy of views
- Typically fills the screen, but can be embedded in other Activity or a appear as floating window
- Java class, typically one Activity in one file

Activity

An activity is a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window for you in which you can place your UI with setContentView(View). While activities are often presented to the user as full-screen windows, they can also be used in other ways: as floating windows (via a theme with R.attr.windowIsFloating set), Multi-Window mode or embedded into other windows. There are two methods almost all subclasses of Activity will implement:

Configuring the manifest

- For your app to be able to use activities, you must declare the activities, and certain of their attributes, in the manifest.

Configuring the manifest

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="haqnawaz.org.test">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Test"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Test">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

- intent-filter to start from launcher

Configuring the manifest

Declare activities

To declare your activity, open your manifest file and add an <activity> element as a child of the <application> element. For example:

```
<manifest ... >
  <application ... >
      <activity android:name=".ExampleActivity" />
      ...
  </application ... >
  ...
</manifest >
```

The only required attribute for this element is android:name, which specifies the class name of the activity. You can also add attributes that define activity characteristics such as label, icon, or UI theme. For more information about these and other attributes, see the <activity> element reference documentation.

Declare intent filters

- They provide the ability to launch an activity
  - Explicit
  - Implicit
- <intent-filter> attribute in the <activity> element

Start a task

Start a task

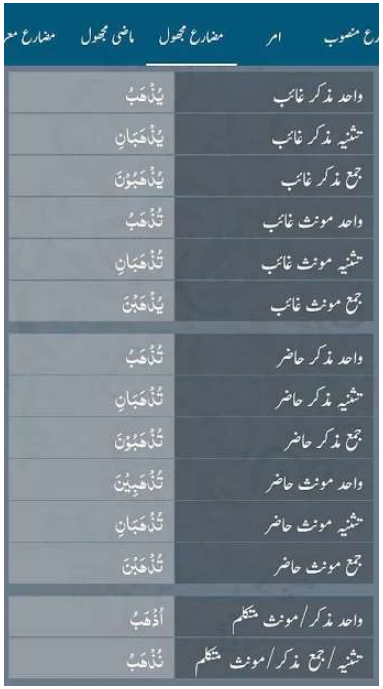
You can set up an activity as the entry point for a task by giving it an intent filter with `"android.intent.action.MAIN"` as the specified action and `"android.intent.category.LAUNCHER"` as the specified category. For example:

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

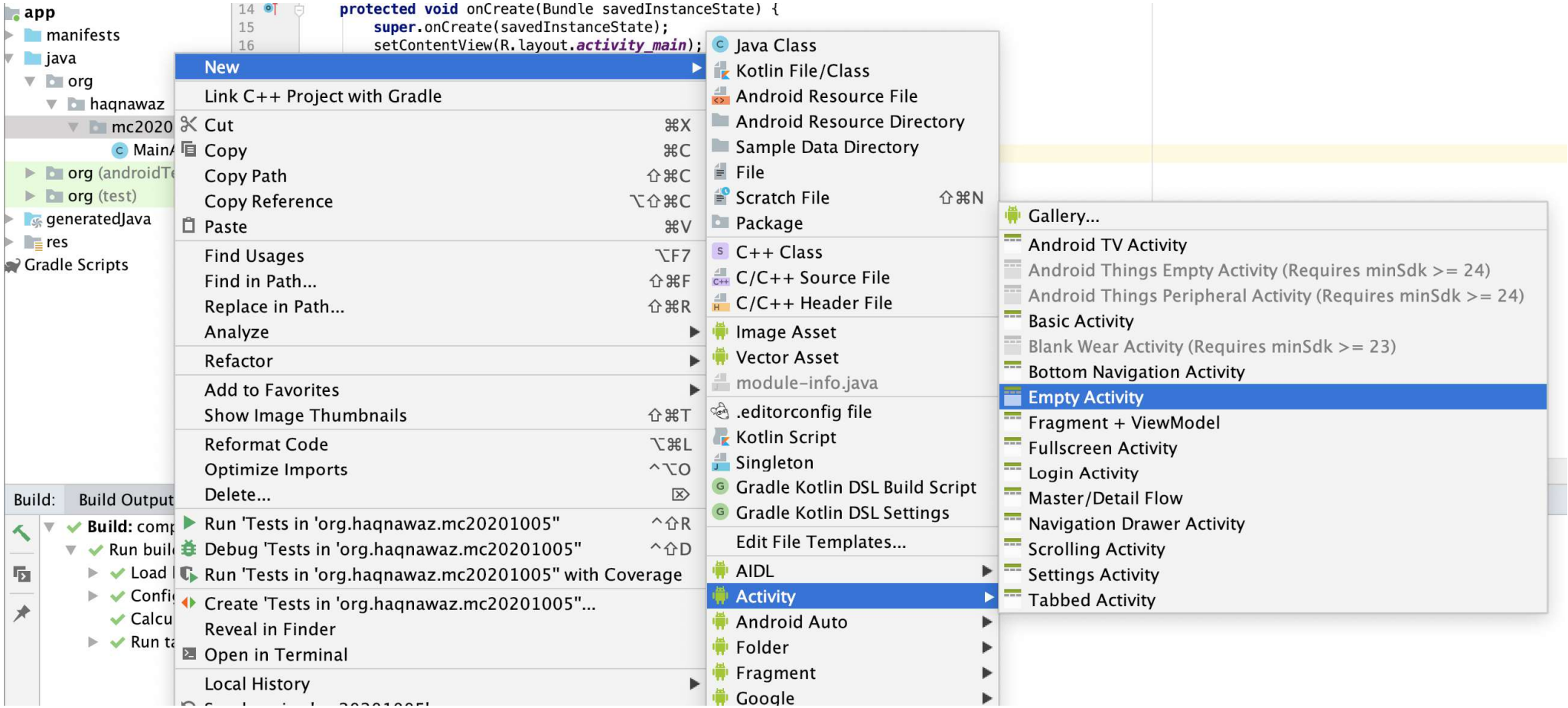
Functions of Activity

- Represents an activity, reciting “surah”, sending email, dialing number
- Handles user interactions, such as button clicks, text entry, or login verification
- Can start other activities in the same or other apps
- Has a life cycle—is created, started, runs, is paused, resumed, stopped, and destroyed

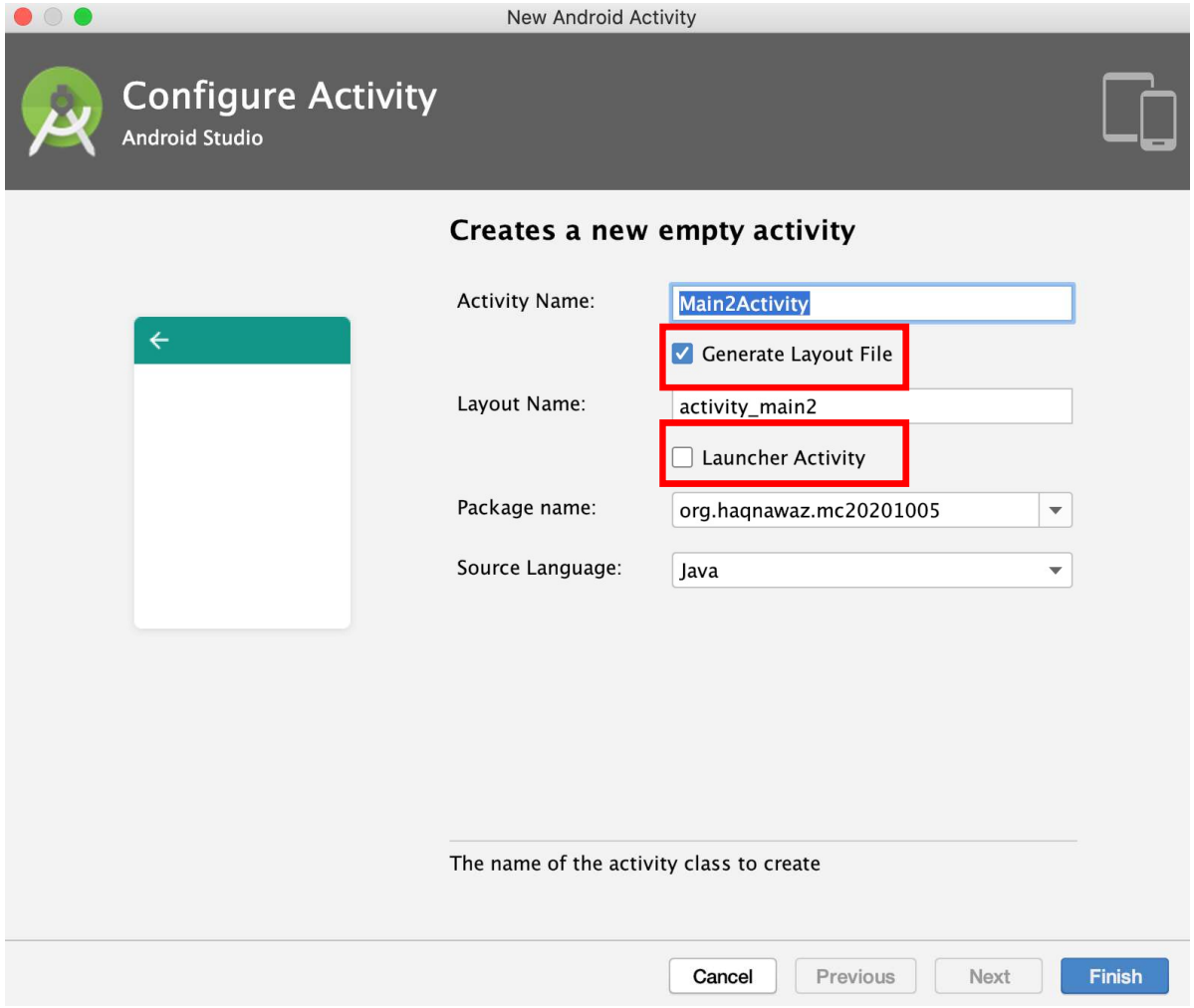
Examples of Activities



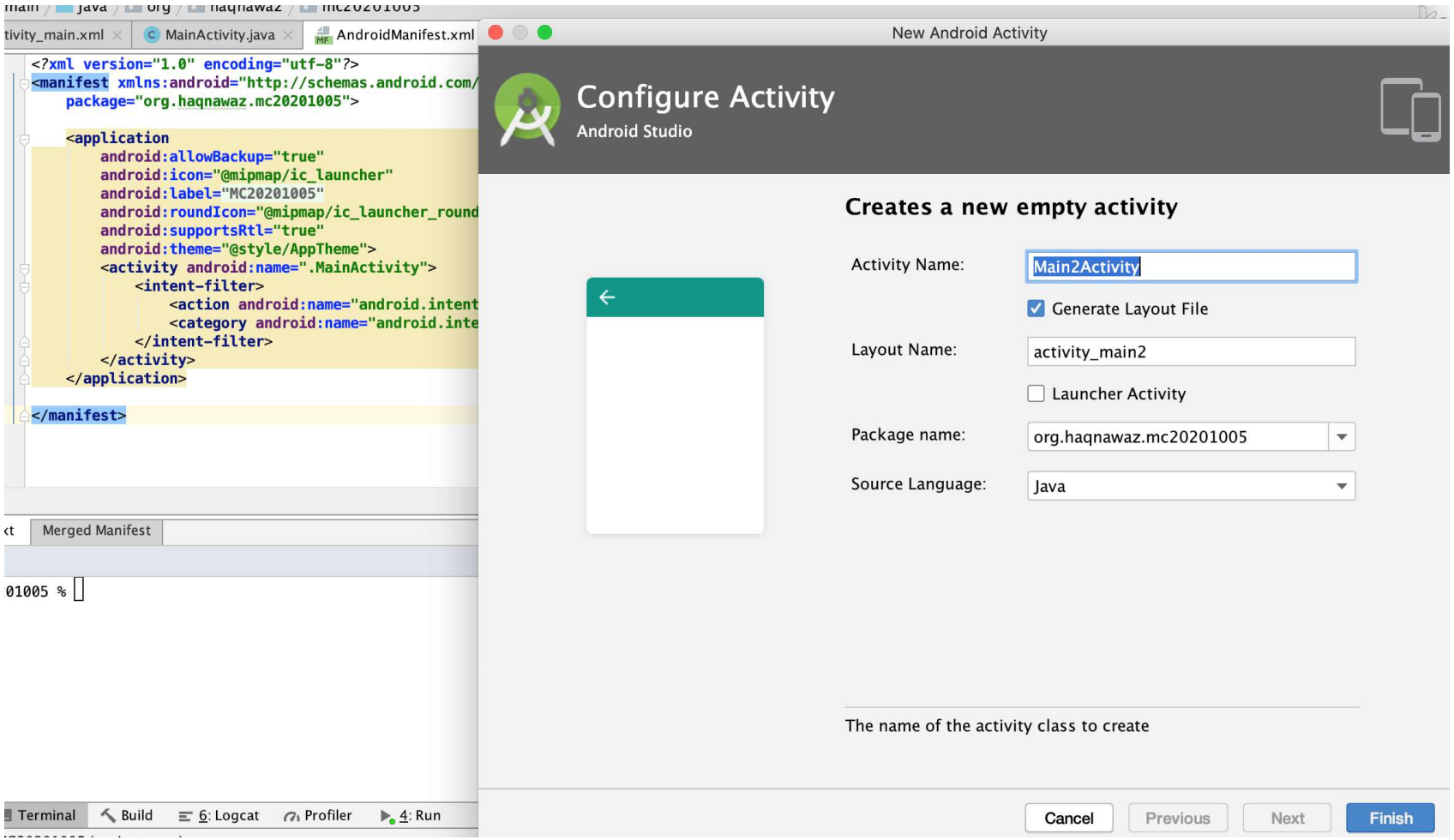
Add New Activity



Add New Activity



Add New Activity

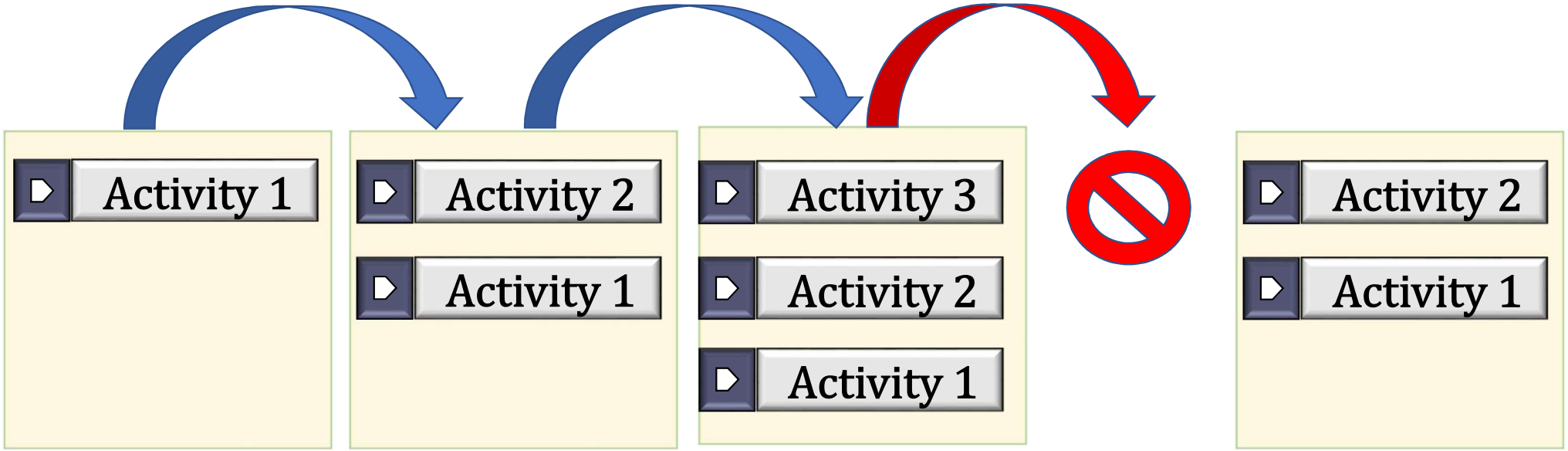


Add New Activity

Tasks and the back stack

- A task is a collection of activities that users interact with when trying to do something in your app. These activities are arranged in a stack—the back stack—in the order in which each activity is opened. For example, an email app might have one activity to show a list of new messages. When the user selects a message, a new activity opens to view that message. This new activity is added to the back stack. Then, if the user presses or gestures Back, that new activity is finished and popped off the stack.

Tasks and the back stack



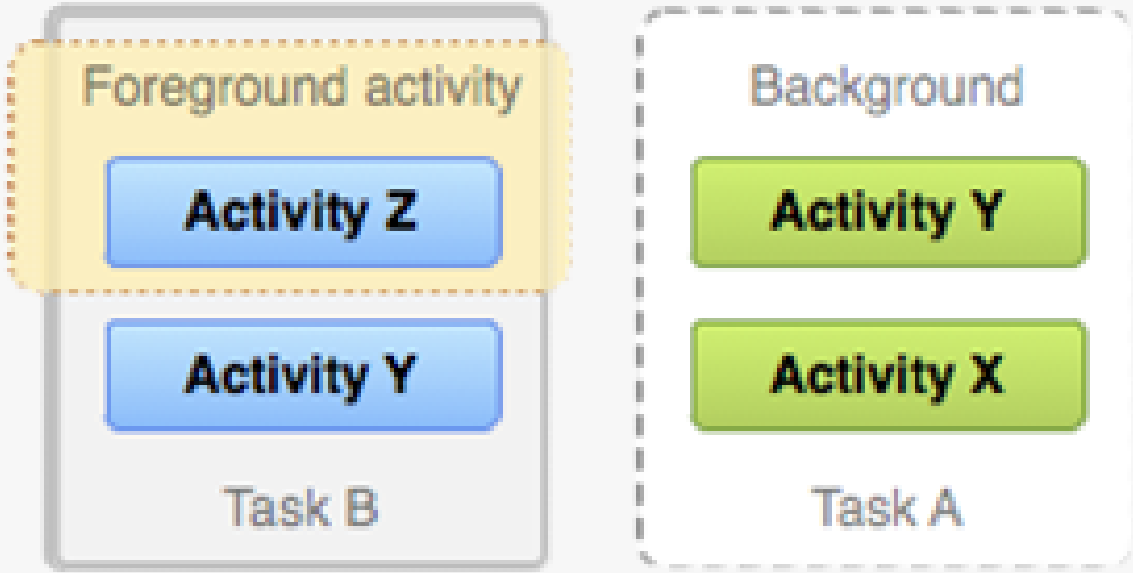
Tasks and the back stack

- Root launcher activities are activities that declare an Intent filter with both ACTION_MAIN and CATEGORY_LAUNCHER. These activities are unique because they act as entry points into your app from the app launcher and are used to start a task.
- When a user presses or gestures Back from a root launcher activity, the system handles the event differently depending on the version of Android that the device is running.

```
<activity android:name=".PalletViews">
    <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<activity android:name=".GrassRootSky"></activity>
<activity android:name=".ZakatActivity"></activity>
<activity android:name=".MainActivity"></activity>
```

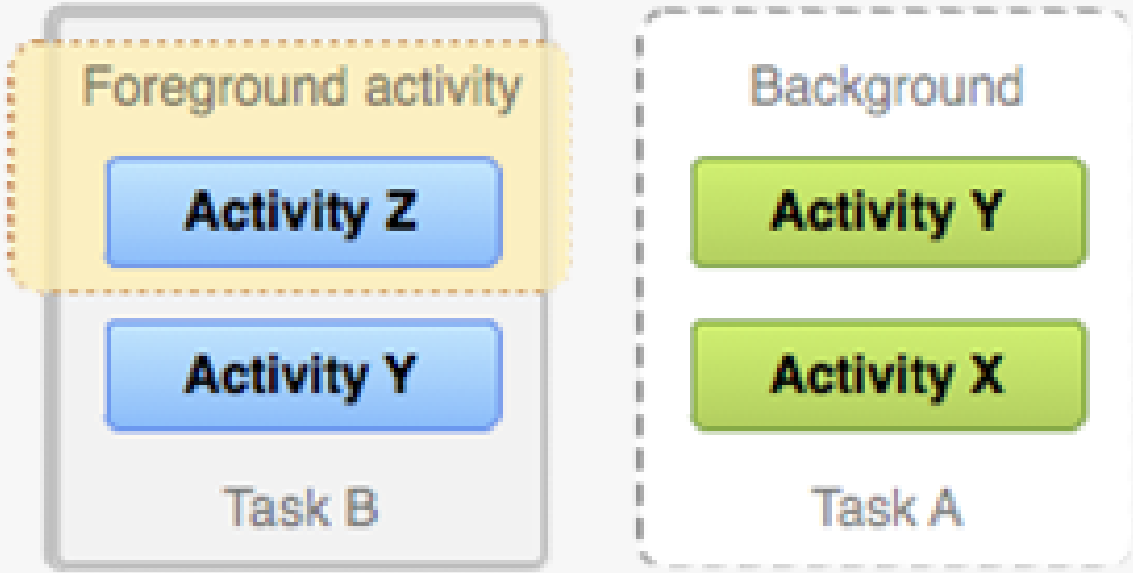
Background and foreground tasks

- The user uses the Home button or gesture, then starts a new app from the app launcher. When the Home screen appears, Task A goes into the background. When the new app starts, the system starts a task for that app (Task B) with its own stack of activities.



Background and foreground tasks

- After interacting with that app, the user returns Home again and selects the app that originally started Task A. Now, Task A comes to the foreground



Extract Resources

https://developer.android.com/guide/components/activities/intro-activities
https://medium.com/androiddevelopers/tasks-and-the-back-stack-dbb7c3b0f6d4#.g6dck3mde
https://developer.android.com/guide/components/activities/tasks-and-back-stack