

CS355

Mobile Application Development การพัฒนาโปรแกรมประยุกต์สำหรับอุปกรณ์พกพา



Pakorn Leesutthipornchai, Ph.D.
Assistant Professor
ผศ.ดร.ปกรณ์ ลีสุทธิพรชัย
pakornl@cs.tu.ac.th



MA07: Handling Runtime Changes
การรองรับการเปลี่ยนแปลงขณะประมวลผล

1

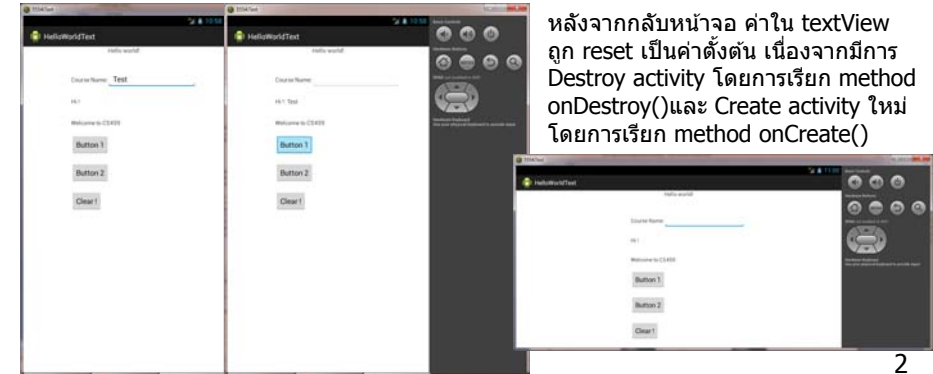
Activity Life Cycle

onCreate()



onDestroy()

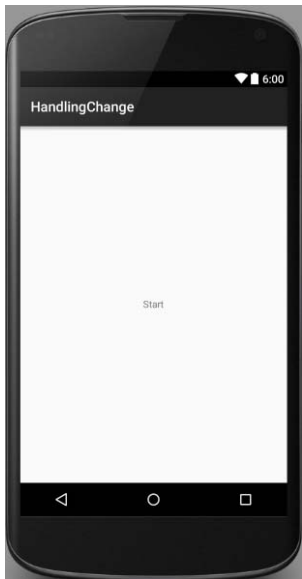
เมื่อ mobile device มีการเปลี่ยนค่า configuration เช่น available width หรือ orientation Activity จะถูก destroy และ create ใหม่



หลังจากกลับหน้าจอ ค่าใน textView ถูก reset เป็นค่าตั้งต้น เนื่องจากการ Destroy activity โดยการเรียก method onDestroy() และ Create activity ใหม่ โดยการเรียก method onCreate()

2

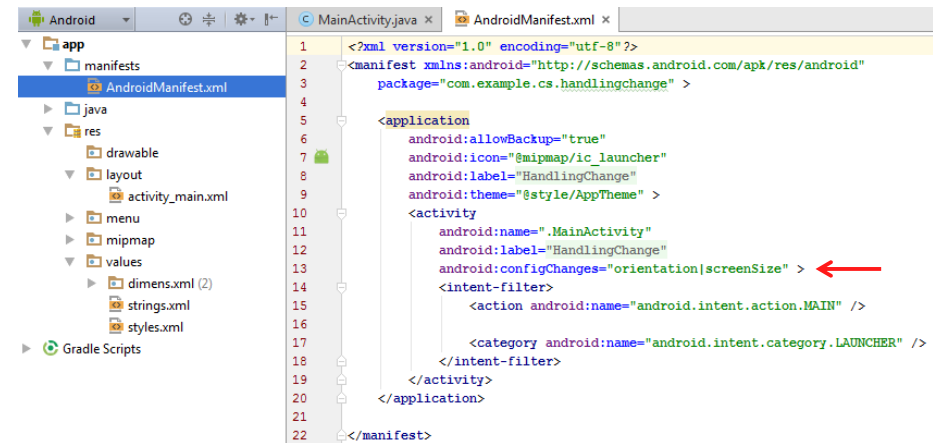
Handling Runtime Changes : Layout



```
1. <LinearLayout
2.     xmlns:android="http://schemas.android.com/apk/res/android"
3.     xmlns:tools="http://schemas.android.com/tools"
4.     android:layout_width="match_parent"
5.     android:layout_height="match_parent"
6.     android:paddingLeft="@dimen/activity_horizontal_margin"
7.     android:paddingRight="@dimen/activity_horizontal_margin"
8.     android:paddingTop="@dimen/activity_vertical_margin"
9.     android:paddingBottom="@dimen/activity_vertical_margin"
10.    android:gravity="center"
11.    tools:context=".MainActivity">
12.
13.    <TextView
14.        android:id="@+id/textView"
15.        android:text="@string/msg"
16.        android:layout_width="wrap_content"
17.        android:layout_height="wrap_content" />
18. </LinearLayout>
```

3

Handling Runtime Changes : AndroidManifest.xml



4

Handling Runtime Changes : android:configChanges

android:configChanges

Lists configuration changes that the activity will handle itself. When a configuration change occurs at runtime, the activity is shut down and restarted by default, but declaring a configuration with this attribute will prevent the activity from being restarted. Instead, the activity remains running and its `onConfigurationChanged()` method is called.

mcc	Mobile country code (MCC) has changed — a SIM has been detected and updated the MCC.
mnc	Mobile network code (MNC) has changed — a SIM has been detected and updated the MNC.
locale	The locale has changed — the user has selected a new language that text should be displayed in.
fontScale	The font scaling factor has changed — the user has selected a new global font size.

5

uiMode	The user interface mode has changed — this can be caused when the user places the device into a desk/car dock or when the night mode changes. <i>Added in API level 8.</i>
orientation	The screen orientation has changed — the user has rotated the device. Note: If your application targets API level 13 or higher (as declared by the <code>minSdkVersion</code> and <code>targetSdkVersion</code> attributes), then you should also declare the "screenSize" configuration, because it also changes when a device switches between portrait and landscape orientations.
screenSize	The current available screen size has changed. This represents a change in the currently available size, relative to the current aspect ratio, so will change when the user switches between landscape and portrait. (this configuration change does not restart your activity, even when running on an Android 3.2 or higher device). <i>Added in API level 13.</i>
smallestScreenSize	The physical screen size has changed. This represents a change in size regardless of orientation, so will only change when the actual physical screen size has changed such as switching to an external display. A change to this configuration corresponds to a change in the <code>smallestWidth configuration</code> . (this configuration change does not restart your activity, even when running on an Android 3.2 or higher device). <i>Added in API level 13.</i>
layoutDirection	The layout direction has changed. For example, changing from left-to-right (LTR) to right-to-left (RTL). <i>Added in API level 17.</i>

Handling Runtime Changes : MainActivity.java

```
1.import android.widget.TextView;
2.import android.content.res.Configuration;
3.import android.widget.Toast;
```

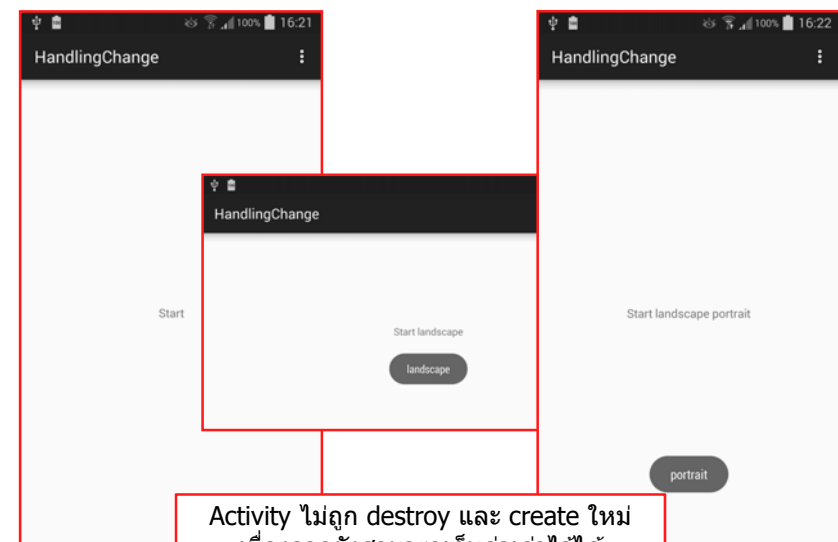
```
1.@Override
2.protected void onCreate(Bundle savedInstanceState) {
3.    super.onCreate(savedInstanceState);
4.    setContentView(R.layout.activity_main);

5.    final TextView txtView=(TextView)findViewById(R.id.textView);
6.    txtView.setText("Start");
7.}
```

```
1.@Override
2.public void onConfigurationChanged(Configuration newConfig) {
3.    super.onConfigurationChanged(newConfig);
4.    final TextView txtView=(TextView)findViewById(R.id.textView);

5.    if (newConfig.orientation == Configuration.ORIENTATION_LANDSCAPE) {
6.        txtView.setText(txtView.getText() + " landscape");
7.        Toast.makeText(this, "landscape", Toast.LENGTH_SHORT).show();
8.    } else if (newConfig.orientation == Configuration.ORIENTATION_PORTRAIT) {
9.        txtView.setText(txtView.getText() + " portrait");
10.        Toast.makeText(this, "portrait", Toast.LENGTH_SHORT).show();
11.    }
12.}
```

Run on Mobile Device



Activity ไม่ถูก destroy และ create ใหม่
เนื่องจากยังสามารถเก็บค่าเอาไว้ได้

8

Toasts (1)

- Small popup
 - requires few amount of space
 - automatically disappear after timeout
- Current activity remains visible and interactive

■ Example code:

```
1) Context context = getApplicationContext();
2) CharSequence text = "Hello toast!";
3) int duration = Toast.LENGTH_SHORT;
4) Toast toast = Toast.makeText(context, text, duration);
5) toast.show();
```

LENGTH_SHORT ≈ 2 วินาที
LENGTH_LONG ≈ 4 วินาที

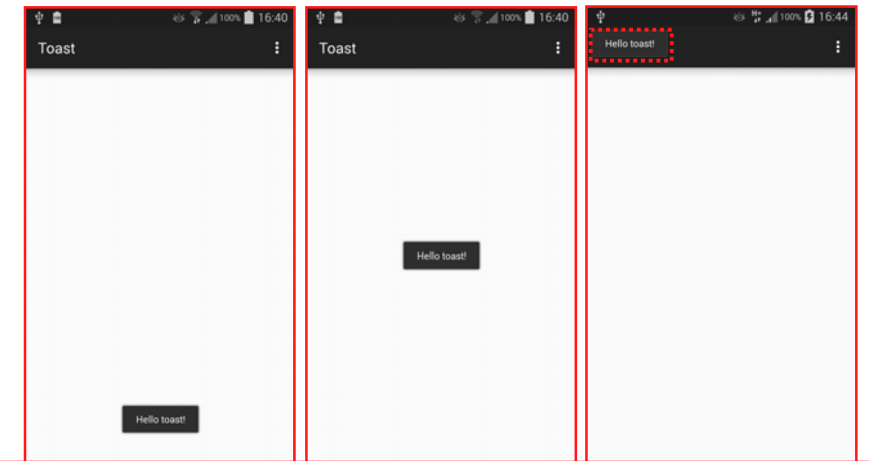
■ Avoid holding on to the Toast object:

- Convert Lines 4 and 5 to
`Toast.makeText(context, text, duration).show();`

9

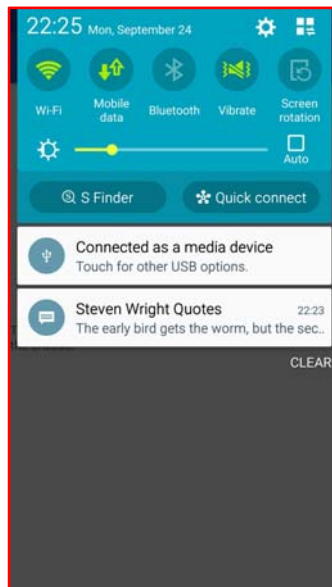
Toasts (2)

- Standard toast notification จะปรากฏอยู่ด้านล่าง ตรงกลาง ของหน้าจอส่วนแสดงผล
- เราสามารถเปลี่ยนตำแหน่งการแสดงผลได้ โดยใช้คำสั่ง
 - `toast.setGravity(Gravity.CENTER, 0, 0);`
 - `toast.setGravity(Gravity.TOP|Gravity.LEFT, 0, 0);`



`toast.setGravity(Gravity.TOP|Gravity.LEFT, Gravity.CENTER_VERTICAL, Gravity.CENTER_HORIZONTAL); ???`

Notification



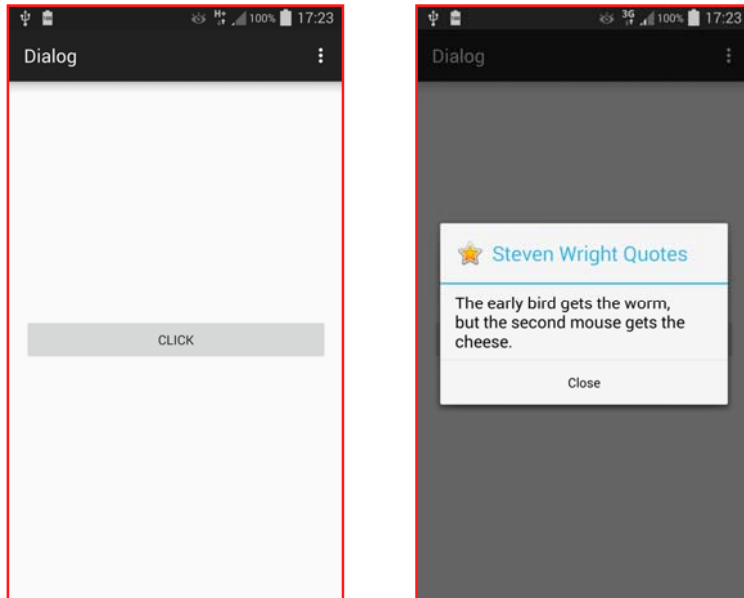
- Small floating message that does not annoy mobile user
- Just show message, or show message and intent to another activity

11

```
1. public class MainActivity extends AppCompatActivity {
2.     String CHANNEL_ID = "1";
3.     public static final int NOTIFICATION_ID = 1;
4.     private NotificationManager mNotificationManager;
5.     NotificationCompat.Builder mBuilder;
6.
7.     @Override
8.     protected void onCreate(Bundle savedInstanceState) {
9.         super.onCreate(savedInstanceState);
10.        setContentView(R.layout.activity_main);
11.        mNotificationManager = (NotificationManager)
12.            this.getSystemService(Context.NOTIFICATION_SERVICE);
13.        PendingIntent contentIntent = PendingIntent.getActivity(this, 0,
14.            new Intent(this, MainActivity2.class), 0);
15.        mBuilder = new NotificationCompat.Builder(this, CHANNEL_ID)
16.            .setSmallIcon(android.R.drawable.stat_notify_chat)
17.            .setContentTitle(getString(R.string.notification_title))
18.            .setContentText(getString(R.string.notification_content))
19.            .setPriority(NotificationCompat.PRIORITY_DEFAULT);
20.        mBuilder.setContentIntent(contentIntent);
21.    }
22.    protected void onResume(){
23.        super.onResume();
24.        mNotificationManager.notify(NOTIFICATION_ID, mBuilder.build());
25.    }
26.}
```

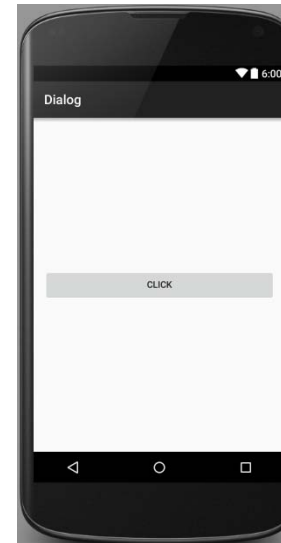
12

Alert Dialog (1)



13

Alert Dialog (2) : Layout



```

1. <LinearLayout
2.     xmlns:android="http://schemas.android.com/apk/res/android"
3.     xmlns:tools="http://schemas.android.com/tools"
4.     android:layout_width="match_parent"
5.     android:layout_height="match_parent"
6.     android:paddingLeft="@dimen/activity_horizontal_margin"
7.     android:paddingRight="@dimen/activity_horizontal_margin"
8.     android:paddingTop="@dimen/activity_vertical_margin"
9.     android:paddingBottom="@dimen/activity_vertical_margin"
10.    android:gravity="center"
11.    tools:context=".MainActivity">
12.
13.        <Button
14.            android:id="@+id/button"
15.            android:layout_width="match_parent"
16.            android:layout_height="wrap_content"
17.            android:text="@string/click"
18.            android:onClick="show"/>
19.    </LinearLayout>

```

14

Alert Dialog (3) : MainActivity.java

```

1. public void show(View v){
2.     final AlertDialog.Builder dialog = new AlertDialog.Builder(this);
3.     dialog.setTitle("Steven Wright Quotes");
4.     dialog.setIcon(android.R.drawable.btn_star_big_on);
5.     dialog.setMessage("The early bird gets the worm, \n
        but the second mouse gets the cheese.");
6.     dialog.setPositiveButton("Close", null);
7.     dialog.show();
8. }

```

15

References

- Handling Runtime Changes, <http://developer.android.com/guide/topics/resources/runtime-changes.html>
- Toast, <http://developer.android.com/guide/topics/ui/notifiers/toasts.html>
- Notification, <https://developer.android.com/training/notify-user/build-notification>
- Dialog, <https://developer.android.com/guide/topics/ui/dialogs>

16