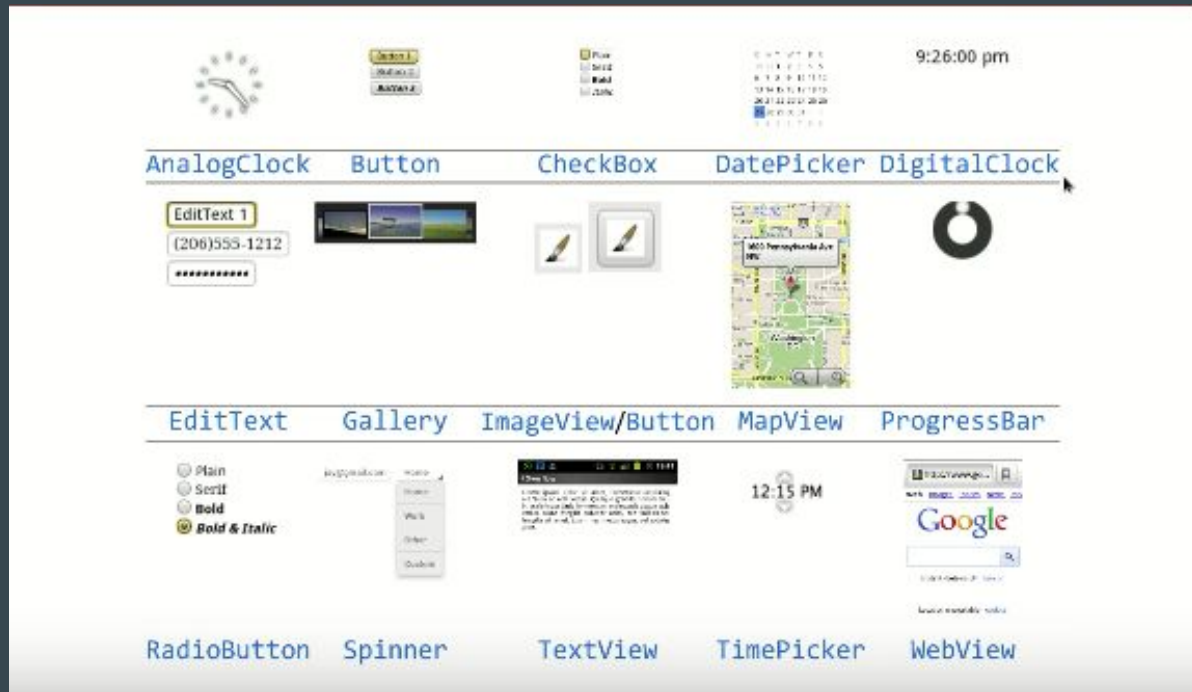


GUI Widgets



Lecture 2

Android widgets



Button

A clickable widget with a text label



- key attributes:

<code>android:clickable="bool"</code>	set to false to disable the button
<code>android:id="@+id/theID"</code>	unique ID for use in Java code
<code>android:onClick="function"</code>	function to call in activity when clicked (must be public, void, and take a View arg)
<code>android:text="text"</code>	text to put in the button

- represented by Button class in Java code

```
Button b = (Button) findViewById(R.id.theID);  
...
```

TextView

<code>android:id="@+id/<i>theID</i>"</code>	unique ID for use in Java code
<code>android:text="<i>string</i>"</code>	text to display

key attributes in XML

```
1 // to change the visible text in Java code
2 TextView myTextView = (TextView) findViewById(R.id.theID);
3 myTextView.setText("text");
```

ScrollView

A container with scrollbars around another widget or container

```
<LinearLayout ...>
    ...
    <ScrollView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
        <TextView ... android:id="@+id/turtle_info" />
    </ScrollView>
</LinearLayout>
```

Michelangelo, Mike or Mikey (as he is usually called), is a fictional character and one of the four protagonists of the Teenage Mutant Ninja Turtles comics and all related media. His mask is typically portrayed as orange outside of the Mirage/Image Comics and his weapons are dual nunchucks, though he has also been portrayed using other weapons, such as a grappling hook. manriki-ousari.

ImageButton

A clickable widget with an image label



- key attributes:

<code>android:clickable="bool"</code>	set to false to disable the button
<code>android:id="@+id/theID"</code>	unique ID for use in Java code
<code>android:onClick="function"</code>	function to call in activity when clicked (must be public, void, and take a View arg)
<code>android:src="@drawable/img"</code>	image to put in the button (must correspond to an image resource)

- to set up an image resource:
 - put image file in project folder **app/src/main/res/drawable**
 - use `@drawable/foo` to refer to `foo.png`
 - use simple file names with only letters and numbers

ImageView

Displays an image without being clickable



- key attributes:

<code>android:id="@+id/<i>theID</i>"</code>	unique ID for use in Java code
<code>android:src="@drawable/<i>img</i>"</code>	image to put in the screen (must correspond to an image resource)

- to change the visible image, in Java code:
 - get the ImageView using `findViewById`
 - call its **`setImageResource`** method and pass `R.drawable.filename`

EditText

An editable text input box



- key attributes:

<code>android:hint="text"</code>	gray text to show before user starts to type
<code>android:id="@+id/theID"</code>	unique ID for use in Java code
<code>android:inputType="type"</code>	what kind of input is being typed; number, phone, date, time, ...
<code>android:lines="int"</code>	number of visible lines (rows) of input
<code>android:maxLines="int"</code>	max lines to allow user to type in the box
<code>android:text="text"</code>	initial text to put in box (default empty)
<code>android:textSize="size"</code>	size of font to use (e.g. "20dp")

- others: `capitalize`, `digits`, `fontFamily`, `letterSpacing`, `lineSpacingExtra`, `minLines`, `numeric`, `password`, `phoneNumber`, `singleLine`, `textAllCaps`, `textColor`, `typeface`

CheckBox

An individual toggleable on/off switch



- key attributes:

<code>android:checked="bool"</code>	set to true to make it initially checked
<code>android:clickable="bool"</code>	set to false to disable the checkbox
<code>android:id="@+id/theID"</code>	unique ID for use in Java code
<code>android:onClick="function"</code>	function to call in activity when clicked (must be public, void, and take a View arg)
<code>android:text="text"</code>	text to put next to the checkbox

- In Java code:

```
CheckBox cb = (CheckBox) findViewById(R.id.theID);  
cb.toggle();  
cb.setChecked(true);  
cb.performClick();
```

RadioButton

A toggleable on/off switch; part of a group

- ☐ Plain
- ☐ Serif
- ☒ **Bold**
- ☐ ***Bold & Italic***

- key attributes:

android:checked=" bool "	set to true to make it initially checked
android:clickable=" bool "	set to false to disable the button
android:id="@+id/ <i>theID</i> "	unique ID for use in Java code
android:onClick=" <i>function</i> "	function to call in activity when clicked (must be public, void, and take a View arg)
android:text=" text "	text to put next to the button

- need to be nested inside a RadioGroup tag in XML
so that only one can be selected at a time

RadioGroup

```
<LinearLayout ...  
    android:orientation="vertical"  
    android:gravity="center|top">  
    <RadioGroup ...  
        android:orientation="horizontal">  
        <RadioButton ... android:id="@+id/lions"  
            android:text="Lions"  
            android:onClick="radioClick" />  
        <RadioButton ... android:id="@+id/tigers"  
            android:text="Tigers"  
            android:checked="true"  
            android:onClick="radioClick" />  
        <RadioButton ... android:id="@+id/bears"  
            android:text="Bears, oh my!"  
            android:onClick="radioClick" />  
    </RadioGroup>  
</LinearLayout>
```



Reusing onClick handler

```
// in MainActivity.java
public class MainActivity extends Activity {

    public void radioClick(View view) {
        // check which radio button was clicked
        if (view.getId() == R.id.lions) {
            // ...
        } else if (view.getId() == R.id.tigers) {
            // ...
        } else {
            // bears ...
        }
    }
}
```



Switch

Switch (link)



An individual toggleable boolean on/off switch

<code>android:id="@+id/theID"</code>	unique ID for use in Java code
<code>android:textOn="text"</code>	text to show next to "on" (right) side
<code>android:textOff="text"</code>	text to show next to "off" (left) side
<code>android:showText="bool"</code>	whether to show text on L/R sides (default false)
<code>android:onClick="function"</code>	function to call in activity when clicked
<code>android:text="text"</code>	text to put on left side next to the switch

key attributes in XML

```
1 // to interact with a Switch in Java code
2 Switch sw = (Switch) findViewById(R.id.theID);
3 if (sw.isChecked()) { ... }
```

Spinner

A drop-down menu of selectable choices

- key attributes:

<code>android:clickable="bool"</code>	set to false to disable the spinner
<code>android:id="@+id/theID"</code>	unique ID for use in Java code
<code>android:entries="@array/array"</code>	set of options to appear in spinner (must match an array in <code>strings.xml</code>)
<code>android:prompt="@string/text"</code>	title text when dialog of choices pops up

- also need to handle events in Java code (see later)
 - must get the Spinner object using `findViewById`
 - then call its `setOnItemSelectedListener` method (see example)

Home
Home
Work
Other
Custom

String resources

- Declare constant strings and arrays in res/values/**strings.xml**:

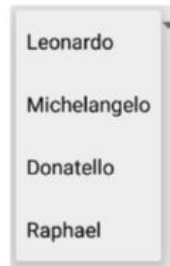
```
<resources>
  <string name="name">value</string>
  <string name="name">value</string>

  <string-array name="arrayname">
    <item>value</item>
    <item>value</item>
    <item>value</item>    <!-- must escape ' as \' in values -->
    ...
    <item>value</item>
  </string-array>
</resources>
```

- Refer to them in Java code:
 - as a resource: `R.string.name`, `R.array.name`
 - as a string or array: `getResources().getString(R.string.name)`,
`getResources().getStringArray(R.array.name)`

Spinner Example

```
<LinearLayout ...>
    <Spinner ... android:id="@+id/tmnt"
        android:entries="@array/turtles"
        android:prompt="@string/choose_turtle" />
    <TextView ... android:id="@+id/result" />
</LinearLayout>
```



- in res/values/strings.xml:

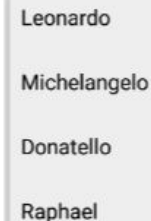
```
<resources>
    <string name="choose_turtle">Choose a turtle:</string>
    <string-array name="turtles">
        <item>Leonardo</item>
        <item>Michelangelo</item>
        <item>Donatello</item>
        <item>Raphael</item>
    </string-array>
</resources>
```


Spinner event example

```
// in MainActivity.java
public class MainActivity extends Activity {
    ...
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Spinner spin = (Spinner) findViewById(R.id.tmnt);
        spin.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
            public void onItemSelected(AdapterView<?> spin, View v, int i, long id) {
                TextView result = (TextView) findViewById(R.id.turtle_result);
                result.setText("You chose " + spin.getSelectedItem());
            }

            public void onNothingSelected(AdapterView<?> parent) {} // empty
        });
    }
}
```



- Leonardo
- Michelangelo
- Donatello
- Raphael

Resources

- In the project directory structure:
 - `res/type/name.extension`
 - *Example: `res/drawable/miranda.png`*
- Referring to a resource, in the XML:
 - `@type/name`
 - *Example: `@drawable/miranda`*
- Referring to a resource ID, in the Java code:
 - `R.type.name`
 - *Example: `R.drawable.miranda`*

Converting resource IDs

- If you have a resource name and want its ID:

```
1 // set pikaID to the integer value of R.drawable.pikachu
2 int pikaID = getResources().getIdentifier(
3     "pikachu", "drawable", getPackageName());
```

- If you have a resource ID and want its name:

```
1 // set pika to "pikachu"
2 String pika = getResources().getResourceEntryName(
3     R.drawable.pikachu);
4
5 // set full to "R.drawable.pikachu"
6 String full = getResources().getResourceName(
7     R.drawable.pikachu);
```

TMNT app exercise

- Write an app to select TMNT characters from a spinner.
 - When a character is selected, an image about that character and other information is presented to the user.
 - Assume that relevant image files are already available for each character.



TMNT app exercise 2

Let's use the radio buttons this time...!

List

A visible menu of selectable choices

- lists are more complicated, so we'll cover them later ...

Android

iPhone

WindowsMobile

Blackberry

WebOS

Ubuntu

Windows7

Max OS X