

# Programming Mobile Applications with Android

22-26 September, Albacete, Spain

Jesus Martínez-Gómez



# Programming Mobile Applications with Android

- Lesson 3.- Visual Interfaces
  - Visual elements.- Buttons, Text Areas, Spinners, Lists, etc
  - Layouts and Fragments.- How to organize the visual elements to improve visualization in different devices.
  - Menus and Action bars.- Some basics
  - Events.- How to handle and manage events
  - Android Lab III.- Create, compile and execute an application with different visual interfaces

# Programming Mobile Applications with Android

- Lesson 3.- Visual Interfaces
  - In this lesson, we will learn:
    - What are the visual elements that can be added to a layout, and when to use each one
    - How to use the fragments to improve the graphical user interfaces
    - How to add items to an options menu or action bar
    - How to manage the events that handle the user actions

# Programming Mobile Applications with Android

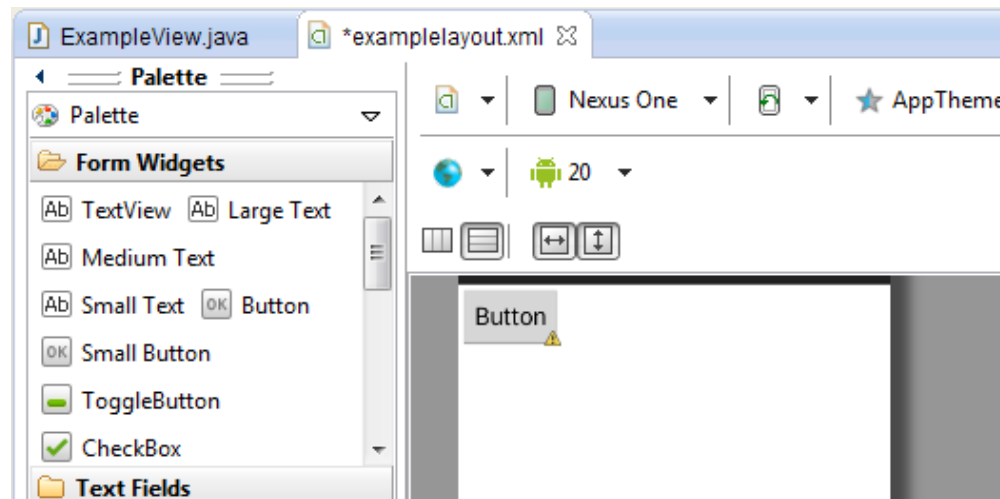
- Visual Elements
  - Button
  - RadioButton
  - CheckBox
  - ToogleBox
  - Spinner
  - TextView
  - EditText
  - ImageView
  - Others

# Programming Mobile Applications with Android

- Visual Elements

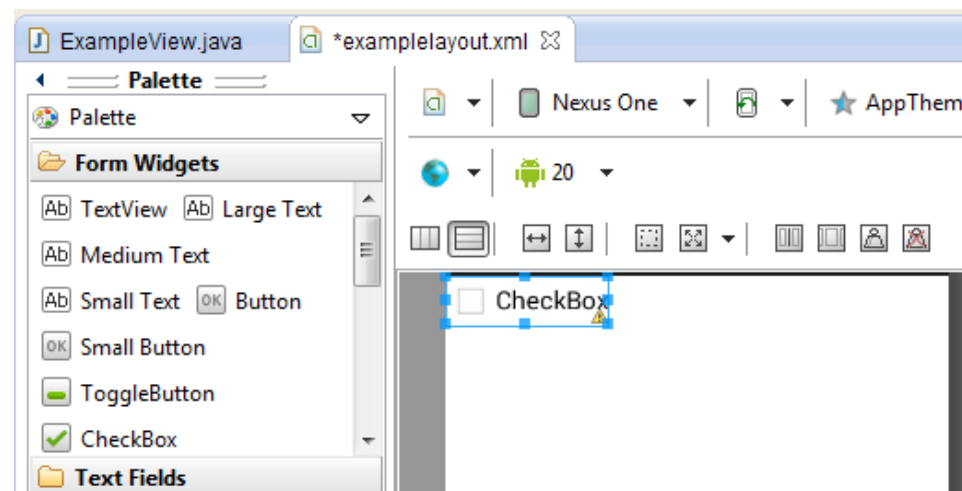
- Button

- Basic element that manages the user pressing
    - It should be used to provoke important changes in the application



# Programming Mobile Applications with Android

- Visual Elements
  - CheckBox
    - Just two states: checked or not
    - Actions over these elements should no invoke changes
    - Used to create yes/no questionnaires



# Programming Mobile Applications with Android

- Visual Elements
  - Spinner
    - Selection from a large number of options
    - The options should be defined
      - By means of programming
      - Using resource string-arrays

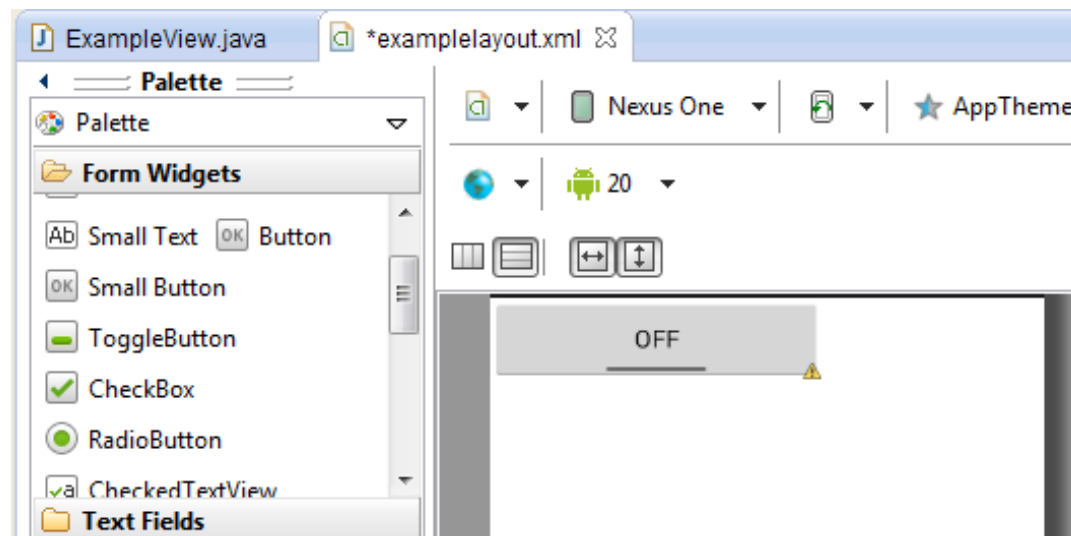
# Programming Mobile Applications with Android

- Visual Elements
  - Spinner
    - Using resource string arrays
    - Edit the strings.xml file in the /res/values folder and add
      - `<string-array name="tempusSeasons">`
      - `<item>First</item>`
      - `<item>Second</item>`
      - `<item>Third</item>`
      - `<item>Fourth</item>`
      - `</string-array>`
    - Then, select the tempusSeasons as entries



# Programming Mobile Applications with Android

- Visual Elements
  - ToogleButtons
    - Just two states: checked or not
    - Used as check box with associated changes

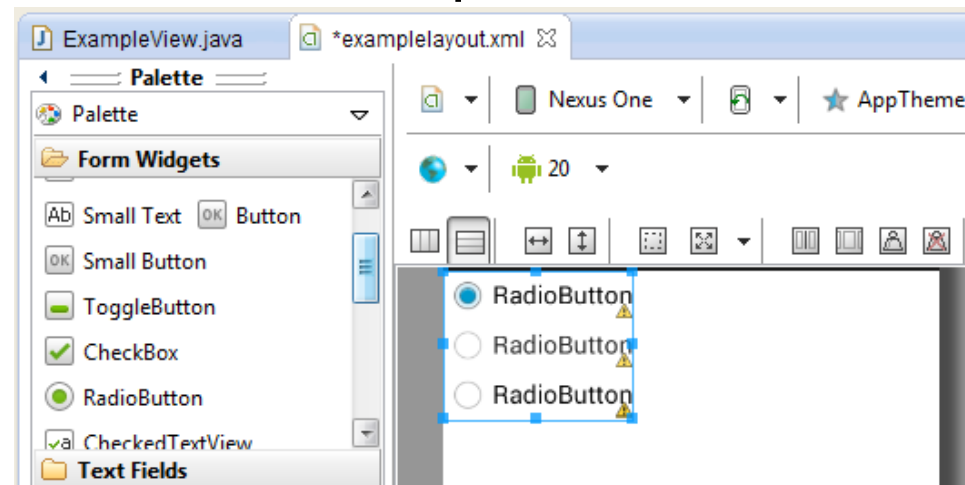


# Programming Mobile Applications with Android

- Visual Elements

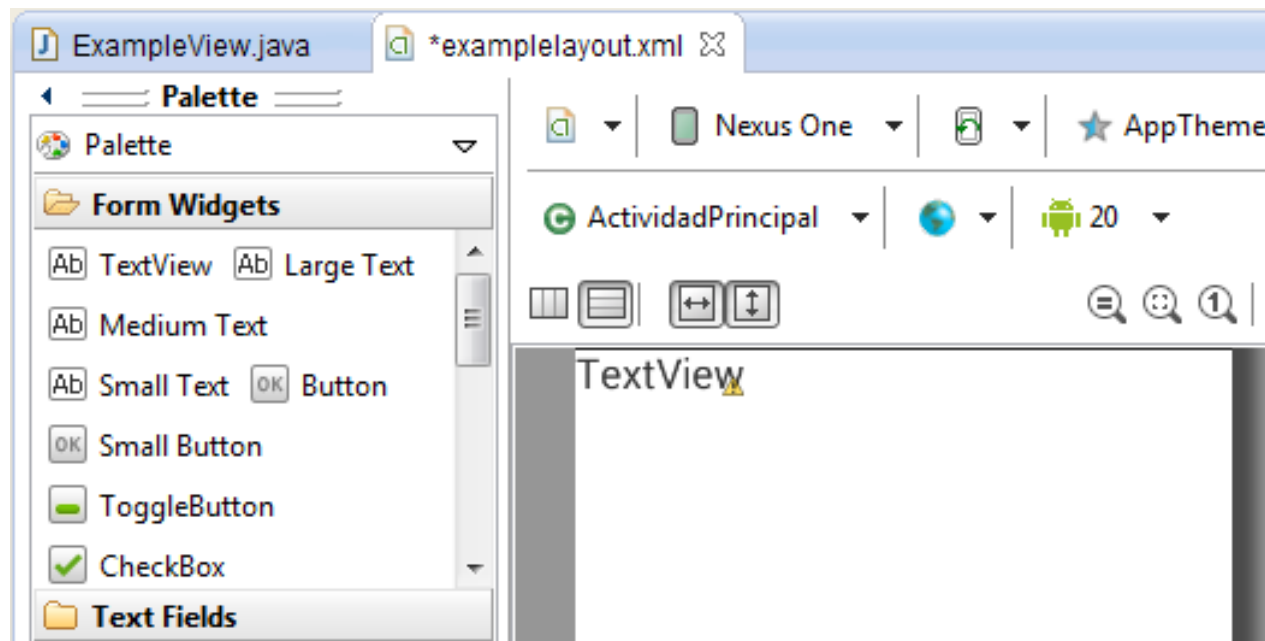
- RadioButton

- Just two states: checked or not
    - Several radio buttons can belong to the same group and then, only one can be selected
    - They should be used for options with few answers



# Programming Mobile Applications with Android

- Visual Elements
  - TextView
    - Used for static text that requires no changes



# Programming Mobile Applications with Android

- Visual Elements
  - EditText
    - Used for the introduction of text from the user
    - When it is focused, the android operative system loads the keyboard
    - Several restrictions to the text introduced can be defined
      - Passwords (The text written is not visualized)
      - Emails
      - Telephone Numbers
      - Postal Address

# Programming Mobile Applications with Android

- Visual Elements

- ImageView

- Used for the visualization of images

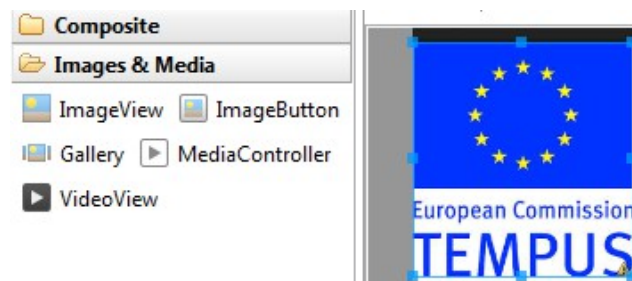
- System Images

- or

- Resource Images

- Example: copy a new image to the res/drawable folder

- Add a new ImageView and select such Image



# Programming Mobile Applications with Android

- Visual Elements
  - Others
    - Rating Bars
    - Progress Bars
    - Seek Bars
    - Gallery
    - Digital Clock
    - Calendar View
    - ...

# Programming Mobile Applications with Android

- Visual Elements

- Visual elements or Views are added in two steps
  - Step 1.- Load them into the layout and establish the preferences (visualization, text, appearance, etc) and remember the Id for future use.

```
<Button  
android:id="@+id/myButton"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Press" />
```

# Programming Mobile Applications with Android

- Visual Elements

- Visual elements or Views are added in two steps

- Step 2.- Create a new reference in the code of the activity to perform changes in the element

- Create a global reference and link it to the name used on the layout

*final Button myButton = (Button)findViewById(R.id.myButton);*

- Now, we can access a set of methods from the class View

```
final Button myButton = (Button)findViewById(R.id.myButton);  
myButton.  
● addChildenForAccessibility(ArrayList<View> children) : v  
● addFocusables(ArrayList<View> views, int direction) : voi  
● addFocusables(ArrayList<View> views, int direction, int fo  
● addOnAttachStateChangeListener(OnAttachStateChange  
● addOnLayoutChangeListener(OnLayoutChangeListener li  
● addTextChangedListener(TextWatcher watcher) : void - T  
● addTouchables(ArrayList<View> views) : void - View  
● animate() : ViewPropertyAnimator - View  
● announceForAccessibility(CharSequence text) : void - Vie
```

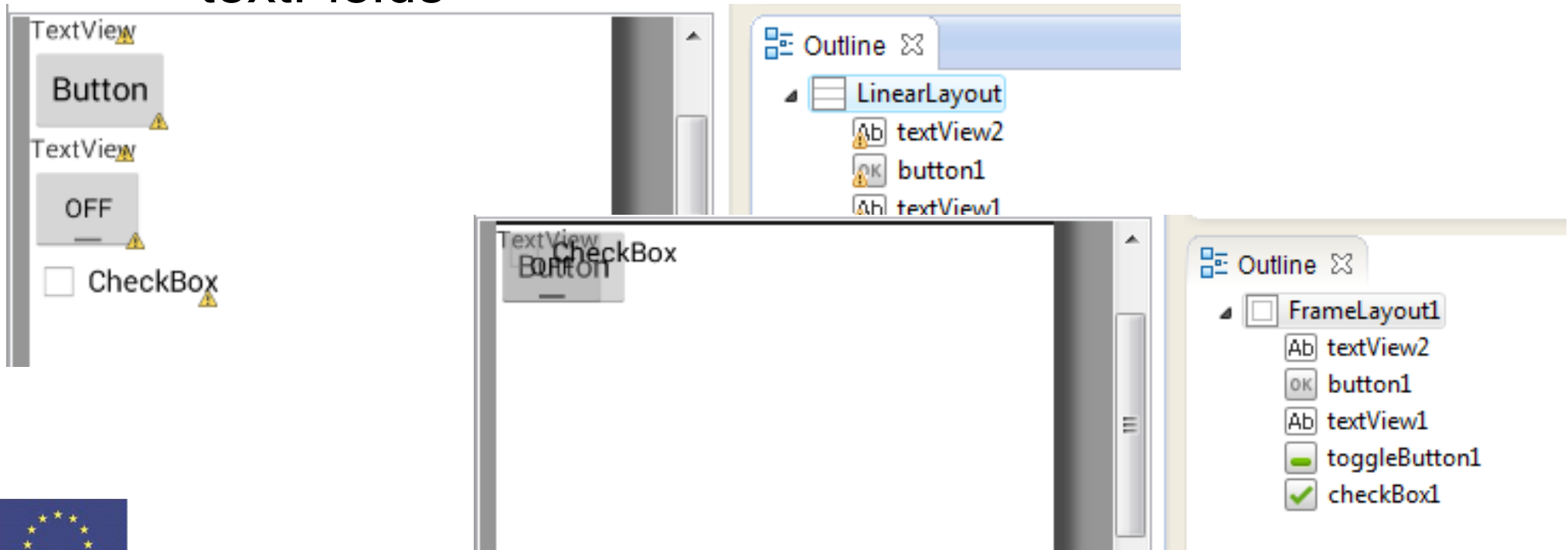


# Programming Mobile Applications with Android

- Layout and Fragments
  - Layouts define the visualization of the elements included in the layout
    - Different types: relative, linear, grid, frame, table
  - Fragments are used to describe a partial visualization of the activity
    - A fragment has its own life-cycle
      - Resumed
      - Paused
      - Stopped
  - Fragments are more complicated but complex

# Programming Mobile Applications with Android

- Layouts
  - Please experiment with the different types of layouts adding some elements like buttons and textFields



# Programming Mobile Applications with Android

- Fragments
  - They started with the release of bigger devices like tablets
  - There two different ways to add a fragment
    - Define it in the layout file → static ones
    - Create them by programming → dynamic ones
  - Must be associated with a subclass of the Fragment class → new methods
    - OnAttach(), onCreate(), onCreateView(), onActivityCreated(), onDetach(),etc

# Programming Mobile Applications with Android

- Fragments – basic usage
  - Fragments can be used to create menus
    - For instance: we create a menu where the first level shows the course days and once a year has been selected, the overall description is shown.
  - What do we need ?
    - Two different .java fragment files
    - A single .java activity class
    - Three layouts: 1 for the main activity and 2 for the fragments

# Programming Mobile Applications with Android

- Fragments – basic usage
  - MainClass
    - Layout: linear layout with two fragments (A and B)
    - Class: extends activity and implements `ListSelectionListener`
      - Receives the index of the element selected in Fragment A
      - Calls a method in Fragment B with the index

# Programming Mobile Applications with Android

- Fragments – basic usage
  - Fragment A
    - Layout: just and empty TextView (for ListFragment cast)
    - Class: extends ListFragment
      - Each time a element is selected, it is invoked a function in the main class
      - When the activity is created, it is filled with the name of the days as a List

# Programming Mobile Applications with Android

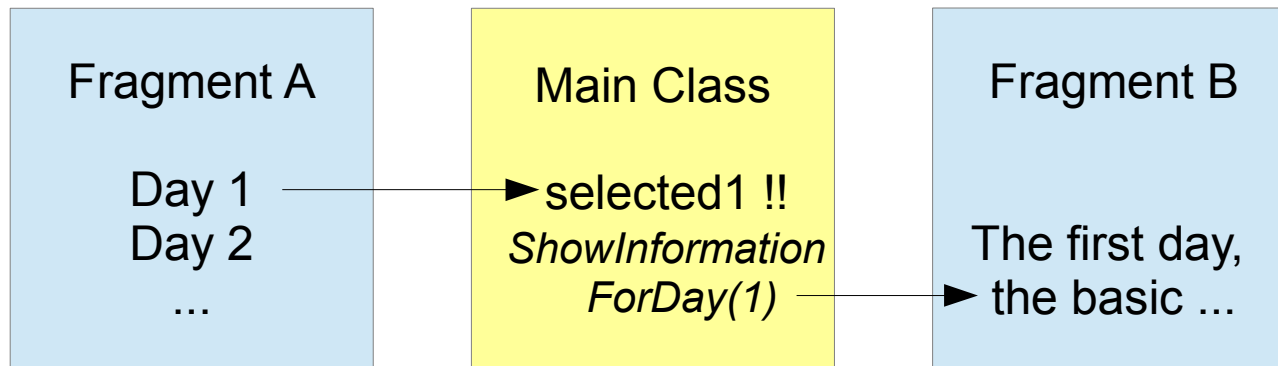
- Fragments – basic usage

- Fragment B

- Layout: just and empty TextView

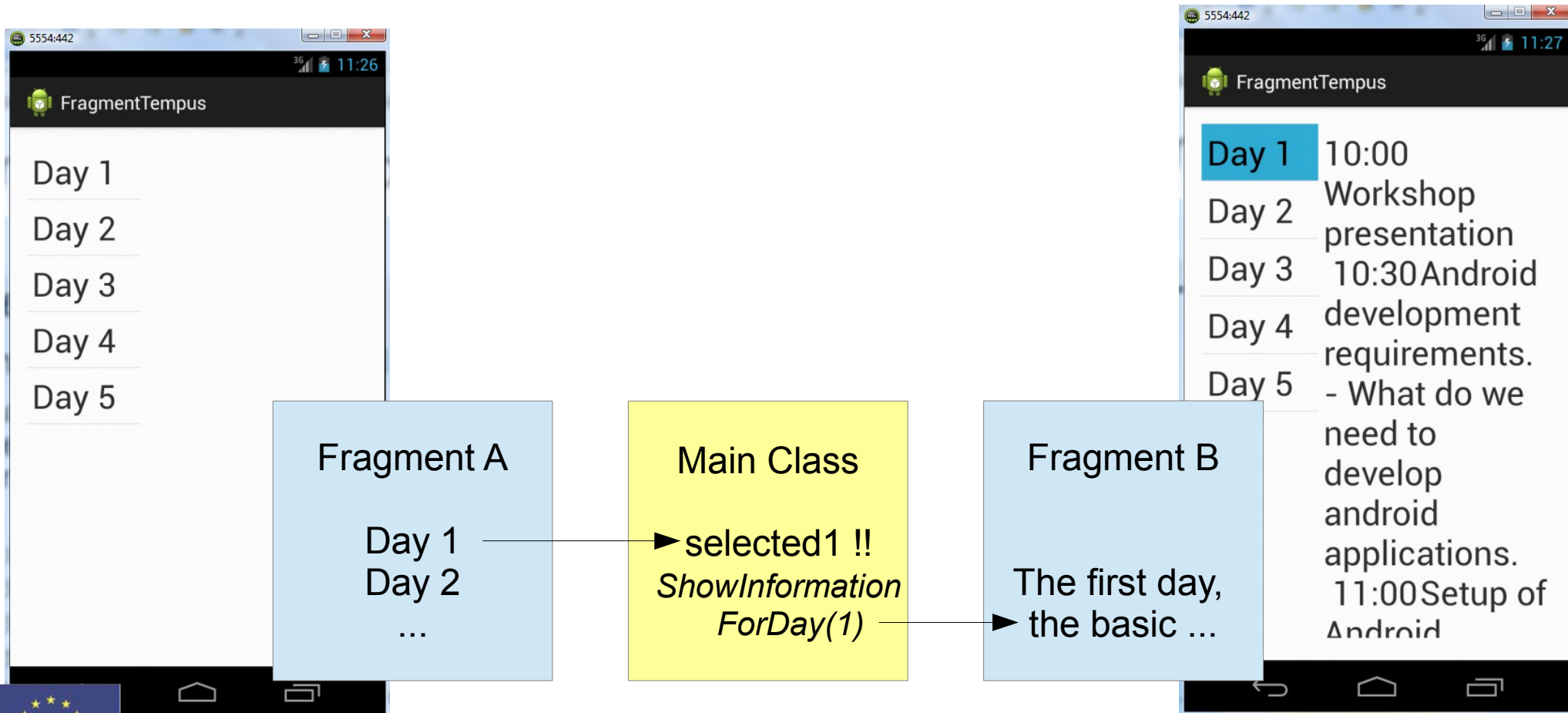
- Class: extends Fragment

- Key method: `showInformationForDay(i)`, establish the text of the TextView when called from the main Activity



# Programming Mobile Applications with Android

- Fragments – basic usage





# Programming Mobile Applications with Android

- Other visual elements: Menus and Action Bar
  - Menus can be easily defined using a resource file, within the /res/menu folder
  - For instance, we can modify the main.xml file created as default and add the following lines

```
<item android:id="@+id/item1" android:title="Menu Option A"/>
```

```
<item android:id="@+id/item2" android:title="Menu Option B"/>
```

```
<item android:id="@+id/item3" android:title="Menu Option C"/>
```

# Programming Mobile Applications with Android

- Other visual elements: Menus
  - Then, there are two methods to implement

```
public boolean onCreateOptionsMenu(Menu menu) {  
    MenuInflater inflater = getMenuInflater();  
    inflater.inflate(R.menu.main, menu);  
    return true; }
```

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.item1:  
            Toast.makeText(getApplicationContext(), "Option 1  
selected", Toast.LENGTH_SHORT).show();
```

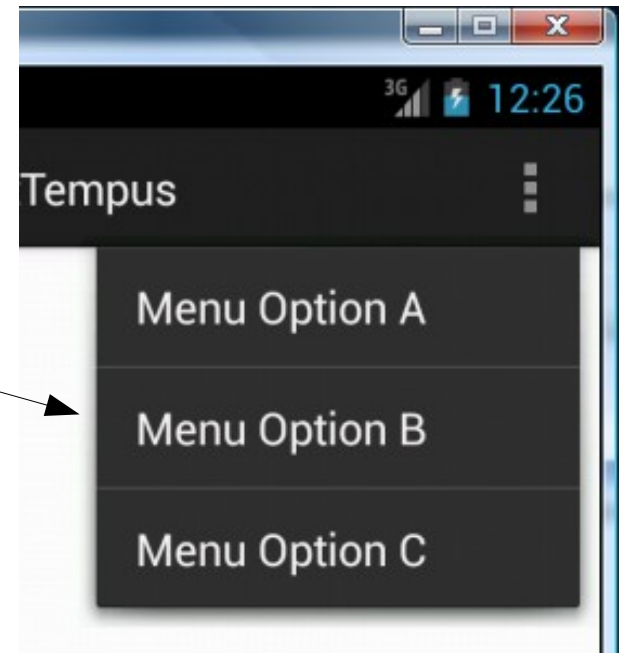
...

```
return false; }}
```

# Programming Mobile Applications with Android

- Other visual elements: Menus

- Menu items can also be associated to shortcut keys
- There are option menus and context menus (associated to view elements)



- Step A for context menus

- Link ViewElement ↔ Menu: `registerForContextMenu(elemVar)`

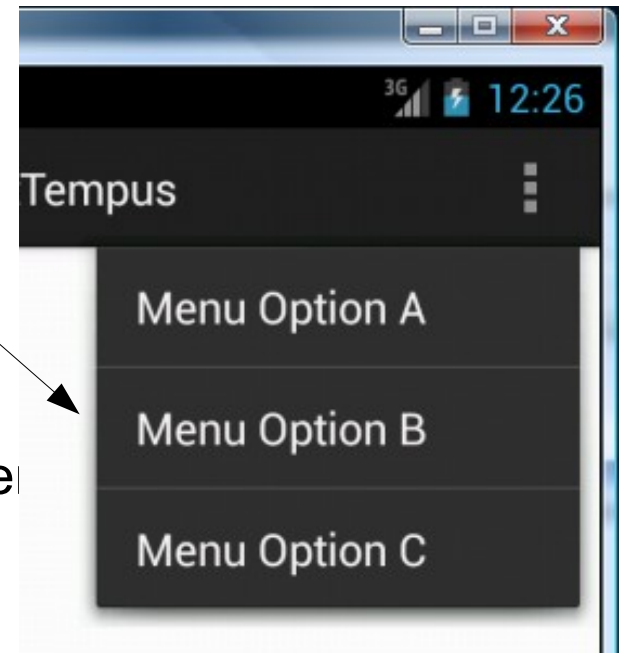
# Programming Mobile Applications with Android

- Other visual elements: Menus

- There are option menus and context menus (associated to view elements)

- Step B for context menus

- Create the menu and associate elements
  - onCreateContextMenu
  - onContextItemSelected



# Programming Mobile Applications with Android

- Other visual elements: Action Bar
  - It is located at the top and can display icons, a descriptive title or interactive items
  - Items should be used to provide direct links to operations with high interest
  - The methods to add and manage items are the same as for the options menu
    - You can add `android:showAsAction="never"` to avoid show actions in the menu
  - You can play with the code using a direct reference: `ActionBar actionBar = getActionBar();`

# Programming Mobile Applications with Android

- Events.- How to handle and manage events?
  - Events: dynamic part of the android applications
  - There are two overall types of events
    - Those performed over specific View elements
      - Button is pressed
      - Some text introduced in a text field
      - An image has been clicked
      - ...

# Programming Mobile Applications with Android

- Events.- How to handle and manage events?
  - There are two overall types of events
    - Those performed over the whole activity
      - Click
      - A key has been pressed
  - Classes in charge of the management of events
    - Listener → are executed automatically

# Programming Mobile Applications with Android

- Events.- List of common events
  - Onclick
  - ScrollX
  - ScrollY
  - ScrollBarSize
  - LongClickable
  - OverScrollMode
  - Padding
  - Focusable



# Programming Mobile Applications with Android

- Events.- View element events
  - A button has been pressed

```
final Button myButton = (Button)findViewById(R.id.myButton);
    myButton.setOnClickListener(new OnClickListener()
    {
        public void onClick(View v)
        {
            Toast.makeText(MainActivity.this, "Button
Pressed", Toast.LENGTH_LONG).show();
        }
    });
```

# Programming Mobile Applications with Android

- Events.- View element events
  - Enter key has been inserted in a text edit

```
final EditText myEditText = (EditText) findViewById(R.id.editText1);
    myEditText.setOnKeyListener(new OnKeyListener() {

        public boolean onKeyDown(View v, int keyCode, KeyEvent event) {
            if(keyCode==KeyEvent.KEYCODE_ENTER)
                Toast.makeText(MainActivity.this,
                "Enter", Toast.LENGTH_LONG).show();
            return false;
        }
    });
```

# Programming Mobile Applications with Android

- Events.- View element events
  - The element of a spinner has been selected

```
final Spinner mySpinner = (Spinner) findViewById(R.id.spinner1);
    mySpinner.setOnItemSelectedListener(new OnItemSelectedListener()
    {
public void onItemSelected(AdapterView<?> parent, View view,int position, long id)
    {
    Toast.makeText(MainActivity.this, "It has been selected the item at position
    "+position,Toast.LENGTH_LONG).show();
    }
public void onNothingSelected(AdapterView<?> parent) {
    Toast.makeText(MainActivity.this, "No item has been
    selected",Toast.LENGTH_LONG).show();
    }
    });
```

# Programming Mobile Applications with Android

- Events.- View element events
  - The whole activity can also be prepared to respond to events.
  - In this case, the activity class has to implement some of the following methods
    - OnClickListener
    - OnLongClickListener
    - OnKeyListener
    - onTouchListener
  - The class has to include the related methods

# Programming Mobile Applications with Android

- Android Lab III.- Create, compile and execute an application with different visual interfaces
  - Follow the instructions to create the same android application with events management and different visualizations

# Programming Mobile Applications with Android

22-26 September, Albacete, Spain

Jesus Martínez-Gómez

