Building User Interface

Session 3

Objective

- The View Class
- Basic Widgets
  - Labels
  - Buttons
  - Images
  - EditText
  - CheckBox
  - RadioButtons
- Android Layouts
- Five basic types of Layouts
- Attaching Layouts to Java Code

The View Class

- Lớp View biểu diễn các thành phần giao diện cho người dùng.
- Một View chiếm một vùng chữ nhật trên màn hình và nhận vào cũng như xử lý các tương tác.
- View là lớp cơ bản của widgets, được dùng để tạo các thành phần UI tương tác (buttons, text fields, ...).
- Lớp con ViewGroup là lớp cơ bản cho layouts, là container ẩn chứa các Views khác (hoặc ViewGroups khác) và định nghĩa các thuộc tính layout của nó.

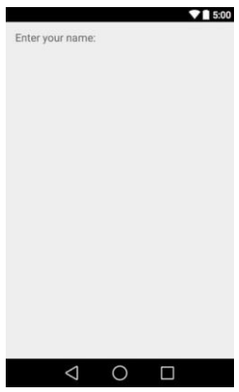
Using Views

- Tất cả các views trong cửa sổ được sắp xếp dưới dạng một cây đơn
- Ta có thể thêm các view bằng code hoặc kéo thả và tùy chỉnh thuộc tính trong XML.
- Sau khi tạo view, có bốn kiểu thao tác phổ biến:
  - Set properties
  - Set focus
  - Set up listeners
  - Set visibility

Basic Widgets: Labels

- Một nhãn trong android gọi là TextView.
- TextViews thường sử dụng để hiển thị.
- TextViews không thể edit, do đó không thể nhập dữ liệu

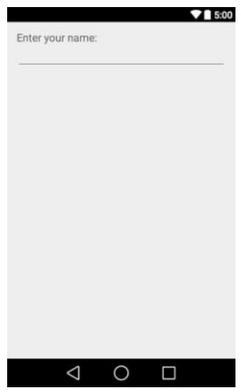
```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="Enter your name:"
    android:id="@+id/textView"
    android:layout_alignParentTop="true"
    android:layout_alignParentStart="true" />
```



Basic Widgets: EditText

- EditText (TextBox) là mở rộng của TextView và cho phép update
- Có thể cấu hình dữ liệu nhập
- Hai phương thức java quan trọng : txtBox.setText("giá trị") và txtBox.getText().toString()

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/editText"
    android:layout_below="@+id/textView"
    android:layout_alignParentStart="true" />
```



Basic Widgets: Buttons

- Button widget cho phép thực hiện hành động click trên GUI.
- Button là lớp con của TextView. Tạo và lập các thuộc tính tương tự TextView.

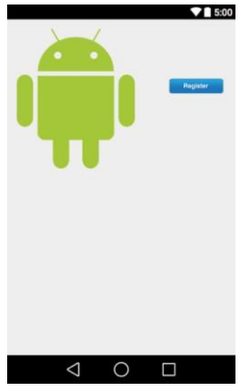
```
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Register"
    android:id="@+id/button"
    android:layout_marginTop="37dp"
    android:layout_below="@+id/editText"
    android:layout_alignParentStart="true" />
```

Basic Widgets: Images

- ImageView và ImageButton là hai Android widgets cho phép nhúng ảnh vào ứng dụng.
- Cả hai image-based widgets tương tự TextView và Button.
- Mỗi widget lấy ảnh từ android:src và android:background attribute (trong XML layout).
- Ảnh thường đặt trong thư mục tài nguyên *drawable*.
- Ta cũng có thể đặt nội dung ảnh bằng URI từ một content provider qua setImageURI().
- ImageButton là lớp con của ImageView, nó cũng nhận sự kiện *click*.

Basic Widgets: Images

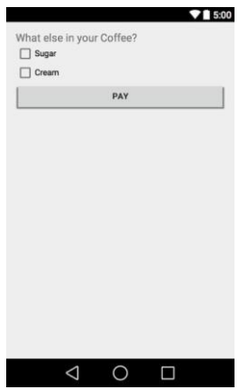
```
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/imageView"
    android:src="@drawable/AndroidRobot"
    android:layout_alignParentTop="true"
    android:layout_alignParentStart="true" />

<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/imageButton"
    android:padding="0dp"
    android:src="@drawable/RegisterButton"
    android:layout_alignTop="@+id/imageView"
    android:layout_alignParentEnd="true"
    android:layout_marginTop="79dp" />
```



Basic Widgets: CheckBox

- Checkbox là một loại nút bấm đặc biệt với hai trạng thái *checked hoặc unchecked*.
- Ví dụ:

```
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Sugar"
    android:id="@+id/checkBox"
    android:checked="false"
    android:layout_below="@+id/textView"
    android:layout_alignParentStart="true" />
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Cream"
    android:id="@+id/checkBox2"
    android:layout_below="@+id/checkBox"
    android:layout_alignParentStart="true"
    android:checked="false" />
```

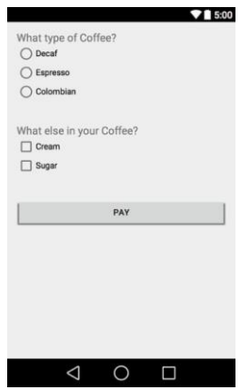


Basic Widgets: RadioButtons

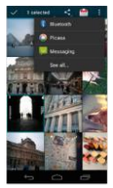
- Radio button là nút bấm chỉ nhận một trong hai trạng thái *checked* ***hoặc*** *unchecked*.
- Khi một radio button ở trạng thái unchecked, user có thể bấm hoặc click để chuyển sang trạng thái check.
- Radio buttons thông thường được nhóm với nhau thành RadioGroup**.**
- Khi một số radio buttons đặt trong một radio group, chọn một radio button sẽ *unchecks* ***tất cả các radio button khác***.
- RadioButton thừa kế … TextView, do đó các thuộc tính tương tự TextView.
- Tương tự, ta có thể gọi isChecked() trên một RadioButton để kiểm tra nếu nó được chọn (selected), toggle() để chọn tương tự Checkbox

Basic Widgets: RadioButtons

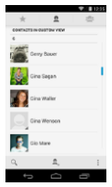
```
<RadioButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Decaf"
    android:id="@+id/radioButton"
    android:layout_below="@+id/textView2"
    android:layout_alignParentStart="true"
    android:checked="false" />

<RadioButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Espresso"
    android:id="@+id/radioButton2"
    android:layout_below="@+id/radioButton"
    android:checked="false" />

<RadioButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Colombian"
    android:id="@+id/radioButton3"
    android:checked="false"
    android:layout_below="@+id/radioButton2"
    android:layout_alignParentStart="true" />
```



Sample UI



Action Bar



Tab Bar

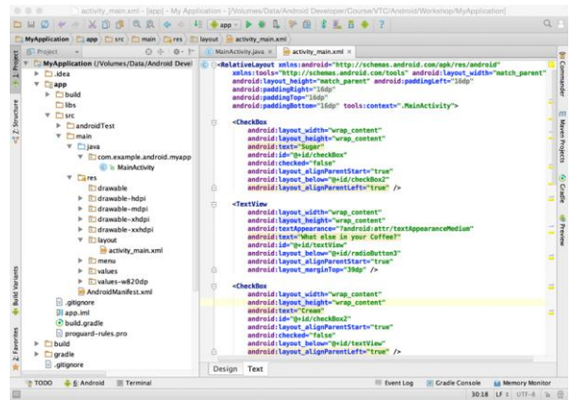
Swipe View

Navigation Drawer

XML Layout

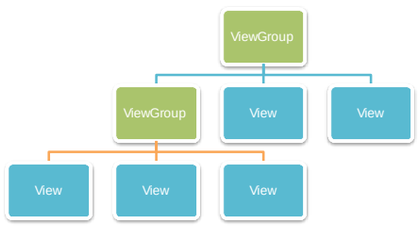
- Một XML-based layout xác định các thành phần UI (widgets) và quan hệ giữa chúng – tất cả được viết theo khuôn dạng XML
- Android xem XML-based layouts là tài nguyên (resources), và các file layout được lưu trữ trong thư mục res/layout của Android project
- Mỗi file XML chứa một *tree of elements* xác định một layout của cacs widgets và containers

Declaring Application Requirements



Android Layouts

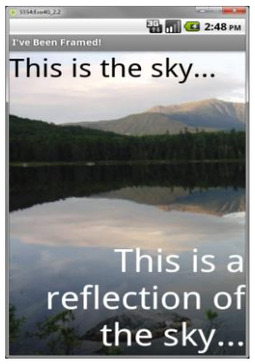
- Cách phổ biến nhất để định nghĩa layout và biểu diễn quan hệ các view là thông qua XML layout file.
- XML cho phép đọc cấu trúc, tương tự HTML.
- Mỗi thành phần trong XML là một đối tượng *View hoặc ViewGroup*



Frame Layout

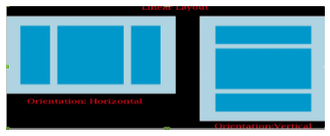
dùng để hiển thị một view bên trong

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <ImageView
        android:src="@drawable/ic_launcher"
        android:scaleType="fitCenter"
        android:layout_height="250px"
        android:layout_width="250px"/>

    <TextView
        android:text="Frame Layout Demo"
        android:textSize="30px"
        android:textStyle="bold"
        android:layout_height="fill_parent"
        android:layout_width="fill_parent"
        android:gravity="center"/>
</FrameLayout>
```



Linear Layout

- LinearLayout sắp xếp các view con bên trong nó theo một cột (từ trên xuống dưới) hoặc theo một hàng (từ trái qua phải)
- Các view con được xếp dọc hoặc ngang tùy thuộc vào tham số android:orientation của LinearLayout



Linear Layout

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:paddingLeft="16dp"
   android:paddingRight="16dp"
   android:orientation="vertical" >
   <EditText
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="To" />
   <EditText
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Subject" />
   <EditText
      android:layout_width="match_parent"
      android:layout_height="0dp"
      android:layout_weight="1"
      android:gravity="top"
      android:hint="Message" />
   <Button
      android:layout_width="100dp"
      android:layout_height="wrap_content"
      android:layout_gravity="right"
      android:text="Send" />
</LinearLayout>
```

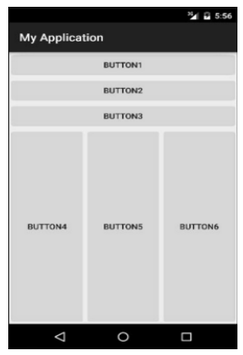
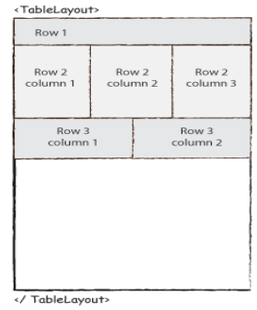


Table Layout

- TableLayout positions its children into rows and columns.
- TableLayout containers do not display border lines for their rows, columns, or cells.
- The table will have as many columns as the row with the most cells.
- A table can leave cells empty, but cells cannot span columns, as they can in HTML.
- *TableRow* objects are the child views of a TableLayout (each TableRow defines a single row in the table).
- Each row has zero or more cells, each of which is defined by any kind of other Views (Image View, TextView,…)
- A cell may also be a ViewGroupobject (for example, you can nest another TableLayout as a cell).

Table Layout

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:stretchColumns="1">
   <TableRow>
      <TextView
         android:layout_column="1"
         android:text="Open..."
         android:padding="3dip" />
      <TextView
         android:text="Ctrl-O"
         android:gravity="right"
         android:padding="3dip" />
   </TableRow>
   <TableRow>
      <TextView
         android:layout_column="1"
         android:text="Save..."
         android:padding="3dip" />
      <TextView
         android:text="Ctrl-S"
         android:gravity="right"
         android:padding="3dip" />
   </TableRow>
</TableLayout>
```



Relative Layout

- RelativeLayout là một ViewGroup có hiển thị các View con ở các vị trí tương đối.
- Vị trí của mỗi View có thể được quy định liên quan đến các View anh em (như bên trái của hoặc bên dưới một View khác).
  - Hoặc ở các vị trí tương đối với khu vực cha **RelativeLayout** (chẳng hạn như sắp xếp ngay phía dưới, bên trái hoặc trung tâm).

Relative Layout

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:paddingLeft="16dp"
   android:paddingRight="16dp" >
   <EditText
      android:id="@+id/name"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Enter your name" />
  <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Done"
      android:id="@+id/button"
      android:layout_below="@+id/timePicker"
      android:layout_alignEnd="@+id/name" />
  <TimePicker
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/timePicker"
      android:layout_below="@+id/name"
      android:layout_alignParentStart="true" />
</RelativeLayout>
```

Absolute Layout

- AbsoluteLayout cho phép đặt các view con bên trong nó tại vị trí chính xác, cố định, tính theo px hoặc dp.
- Layout loại này không linh hoạt và không thích nghi được với sự thay đổi của độ phân giải màn hình.

```
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent">
   <Button
      android:layout_width="100dp"
      android:layout_height="wrap_content"
      android:text="OK"
      android:layout_x="50px"
      android:layout_y="361px" />
   <Button
      android:layout_width="100dp"
      android:layout_height="wrap_content"
      android:text="Cancel"
      android:layout_x="280px"
      android:layout_y="361px" />
</AbsoluteLayout>
```

Attaching Layouts to Java Code

- Assume *res/layout/main.xml* has been created. This layout could be called by an application using the statement

```
setContentView(R.layout.main)
```

- Individual widgets, such as myButtoncould be accessed by the application using the statement *findViewByID(...)* as in

```
Button button = (Button) findViewById(R.id.myButton)
```

- Where R is a class automatically generated to keep track of resources available to the application. In particular *R.id...* is the collection of widgets defined in the XML layout.

Summary

- In this session, we learnt:
  - The View Class
  - Basic Widgets
    - Labels
    - Buttons
    - Images
    - EditText
    - CheckBox
    - RadioButtons
  - Android layouts
  - Five basic types of layouts
  - Attaching layouts to Java code