Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

Lập trình Android

Bài 6. *Tài nguyên ứng dụng hình ảnh & giao diện*

Ngành Mạng & Thiết bị di động

2014

Nội dung

1. **Tài nguyên hình ảnh**
   - Thư mục lưu trữ - Định dạng – Truy xuất
   - Các dạng tài nguyên hình ảnh
2. **Tài nguyên giao diện**

1.1 Thư mục lưu trữ - Định dạng – Truy xuất

- **Thư mục lưu trữ:**
  - Các tài nguyên hình ảnh được lưu trữ trong thư mục res/drawable.
  - Có thể có nhiều thư mục drawable theo từ hạn định khác nhau:
    - Ví du: drawable-hdpi, drawable-xhdpi…
- **Định dạng:**
  - Tài nguyên hình ảnh bao gồm cả định dạng *.xml và định dạng hình ảnh (.png, .gif, .jpg).
- **Truy xuất: bao gồm 2 cách thức:**
  - Java: R.drawable.<tên tài nguyên>.
  - XML: @[pakage:]drawable/<tên tài nguyên>.

1.1 Thư mục lưu trữ - Định dạng – Truy xuất

- **Ví dụ truy xuất tài nguyên hình ảnh:**
  - Java:

```
Resources res = getResources();
Drawable drawable = res.getDrawable(R.drawable.ic_launcher);
```

  - XML:

```
<ImageView
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:src=
        "@drawable/ic_launcher" />
```

```
<ImageButton
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:background=
        "@drawable/ic_launcher" />
```

1.2 Các dạng tài nguyên hình ảnh

- **Bao gồm các định dạnh:**
  - Bitmap
  - Shape
  - LayerList
  - StateList
  - LevelList
  - Transition
  - Inset
  - Clip
  - Scale
  - Nine-Patch

1.2 Các dạng tài nguyên hình ảnh

- **Bitmap:**
  - Định dạng ảnh nhị phân, Android hỗ trợ ba định dạng tài nguyên hình ảnh: png, jpg và gif.
  - Các thực thi của Bitmap bao gồm:
    - Sử dụng như tài nguyên thông qua R.drawable.*filename*
    - Tham chiếu biên dịch tài nguyên thông qua đối tượng BitmapDrawable.

1.2 Các dạng tài nguyên hình ảnh

- **Bitmap:**
  - Sử dụng các thuộc tính Bitmap trong XML:.
    - AntinAlias (XML)
    - Dither
    - Filter
    - Gravity
    - Mipmap
    - Tilemode
    - Automirrored

1.2 Các dạng tài nguyên hình ảnh

- **Bitmap:**
  - Ví dụ xây dựng Bitmap trong XML: **mipmap.xml**

```
<bitmap xmlns:android="http://schemas.android.com/apk/res/android"
    android:mipMap="false"
    android:src="@drawable/caro"
    android:tileMode="repeat" >
</bitmap>
```

  - Truy xuất trong Java code:

```
BitmapDrawable drawable = (BitmapDrawable)getResources()
                                    .getDrawable(R.drawable.mipmap);
```

1.2 Các dạng tài nguyên hình ảnh

- **Shape:**
  - Tài nguyên hình ảnh cho các đối tượng đa giác được vẽ bằng XML, bao gồm:
    - Rectangle
    - Oval
    - Line
    - Ring
  - Tham chiếu biên dịch tài nguyên thông qua đối tượng GradientDrawable.

1.2 Các dạng tài nguyên hình ảnh

- **Shape:**
  - Sử dụng các thuộc tính để cấu tạo đối tượng:
    - Corners (Rectangle) - Integer
      - radius
      - topLeftRadius
      - topRightRadius
      - bottomLeftRadius
      - bottomRadius
    - Padding (Rectangle) – Integer
      - left
      - top
      - right
      - bottom

1.2 Các dạng tài nguyên hình ảnh

- **Shape:**
  - Sử dụng các thuộc tính để cấu tạo đối tượng:
    - Gradient
      - angle - integer
      - centerX - integer
      - centerY - integer
      - centerColor - integer
      - endColor - color
      - gradientRadius – integer
      - startColor – color
      - type – linear | radial | sweep
      - useLevel – true | false

1.2 Các dạng tài nguyên hình ảnh

- **Shape:**
  - Sử dụng các thuộc tính để cấu tạo đối tượng:
    - Size – integer
      - width – integer
      - height – integer
    - Solid – integer
      - color – color
    - Stroke – integer
      - width - integer
      - color – color
      - dashWith – integer
      - dashGap - integer

1.2 Các dạng tài nguyên hình ảnh

- **Shape:**
  - Một số thuộc tính chỉ sử dụng cho đối tượng Ring:
    - innerRadius
    - innerRadiusRatio
    - thickness
    - thicknessRatio
    - useLevel (false)

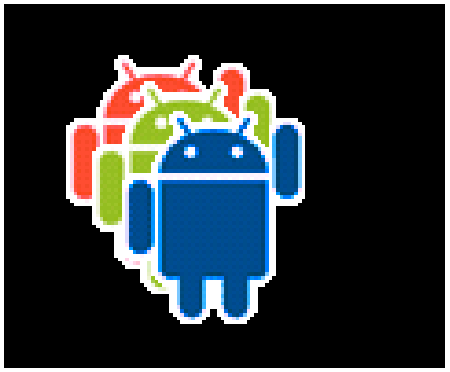
1.2 Các dạng tài nguyên hình ảnh

- **Shape:**
  - Ví dụ:

```
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >
    <gradient
        android:angle="90"
        android:startColor="@android:color/holo_blue_bright"
        android:type="linear" />
    <corners android:radius="20dp"/>
    <size
        android:height="80dp"
        android:width="80dp" />
</shape>
```

1.2 Các dạng tài nguyên hình ảnh

- **LayerList:**
  - Tài nguyên hình ảnh cho phép quản lý mảng các đối tượng hình ảnh được vẽ chồng lên nhau, mỗi đối tượng hình ảnh được qui ước là một item.
  - Mỗi item bao gồm:
    - drawable – resource
    - Id – resource id
    - top - integer
    - right - integer
    - bottom - integer
    - left - integer



  - Tham chiếu biên dịch tài nguyên thông qua đối tượng LayerDrawable.

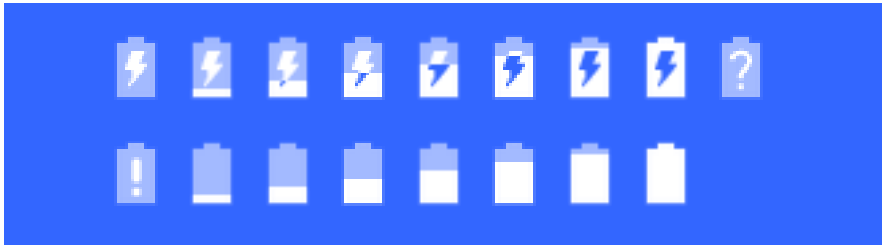
1.2 Các dạng tài nguyên hình ảnh

- **State List:**
  - Tài nguyên hình ảnh cho phép quản lý mảng các đối tượng hình ảnh được vẽ theo trạng thái của đối tượng thể hiện.
  - Một item bao gồm:
    - drawable – resource
    - Tập các trạng thái có thể có:
      - Pressed
      - Focused
      - Hovered
      - Selected
      - Checkable
      - Enable
      - Activated
      - Window focused

  - Tham chiếu biên dịch tài nguyên thông qua đối tượng StateListDrawable.

1.2 Các dạng tài nguyên hình ảnh

- **LevelList:**
  - Tài nguyên hình ảnh cho phép quản lý mảng các đối tượng hình ảnh, mỗi đối tượng hình ảnh được qui ước là một item, hiển thị ảnh theo cấp độ tương ứng đã khai báo.
  - Một item bao gồm:
    - drawable – resource
    - maxLevel
    - minLevel
  - Tham chiếu biên dịch tài nguyên thông qua đối tượng LevelListDrawable.



1.2 Các dạng tài nguyên hình ảnh

- **Transition:**
  - Tài nguyên hình ảnh cho phép thực hiện chuyển đổi (hiệu ứng "biến bóng") giữa hai đối tượng hình ảnh.
  - Mỗi item bao gồm:
    - drawable – resource
    - Id – resource id
    - top - integer
    - right - integer
    - bottom - integer
    - left – integer
  - Các phương thức xử lý chính:
    - startTransition
    - reverserTransition
    - resetTransition.

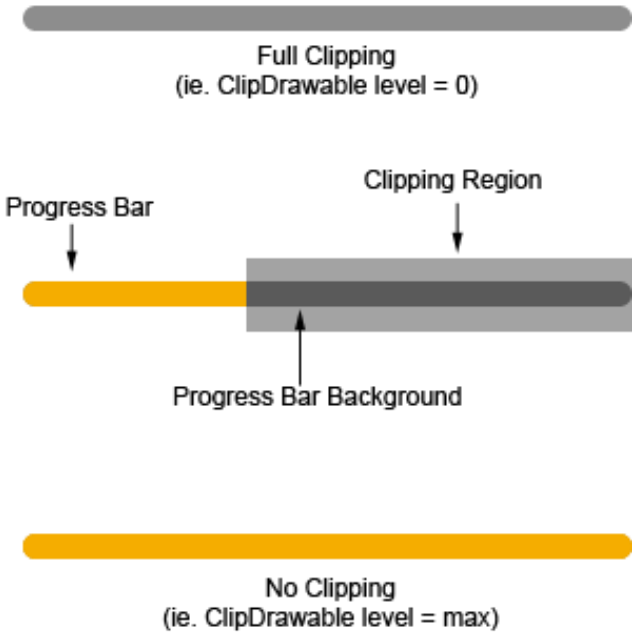
1.2 Các dạng tài nguyên hình ảnh

- **Inset:**
  - Tài nguyên hình ảnh cho phép thực hiện lồng đối tượng hình ảnh theo một ví trí cho trước.
  - Các thuộc tính bao gồm:
    - drawable – resource
    - insetTop - integer
    - insetRight - integer
    - insetBottom - integer
    - insetLeft – integer

  - Tham chiếu biên dịch tài nguyên thông qua đối tượng InsetDrawable.

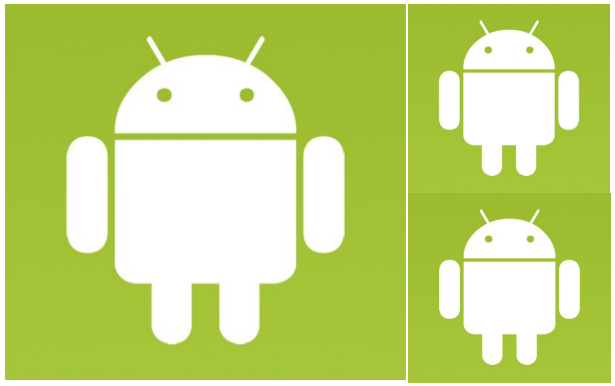
1.2 Các dạng tài nguyên hình ảnh

- **Clip:**
  - Tài nguyên hình ảnh cho phép thực hiện cắt một đối tượng hình ảnh theo thông số vị trí cho trước, có thể thay đổi thông số cắt trong quá trình hoạt động.
  - Tham chiếu biên dịch tài nguyên thông qua đối tượng ClipDrawable.
  - Các thuộc tính bao gồm:
    - drawable – resource
    - clipOrientation – integer
    - Gravity
  - Các phương thức xử lý chính:
    - setLevel (min:0 – max: 10.000)
    - getLevel



1.2 Các dạng tài nguyên hình ảnh

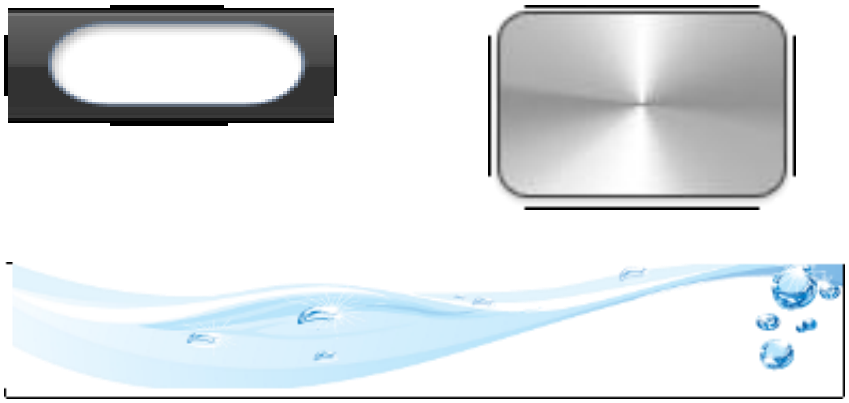
- **Scale:**
  - Tài nguyên hình ảnh cho phép thực hiện phóng to hoặc thu nhỏ một đối tượng hình ảnh theo thông số tỉ lệ cho trước, có thể thay đổi thông số tỉ lệ trong quá trình hoạt động.
  - Các thuộc tính bao gồm:
    - drawable – resource
    - scaleGravity – integer
    - scaleWidth - %
    - scaleHeight - %



  - Tham chiếu biên dịch tài nguyên thông qua đối tượng ScaleDrawable.

1.2 Các dạng tài nguyên hình ảnh

- **NinePatch:**
  - Tài nguyên hình ảnh cho phép thực hiện tạo đối tượng hình ảnh (PNG) có kích thước co dãn theo tỉ lệ đối tượng thể hiện.
  - Các thuộc tính bao gồm:
    - src– resource
    - dither– integer
  - Tham chiếu biên dịch tài nguyên thông qua đối tượng NinePatchDrawable.



Nội dung

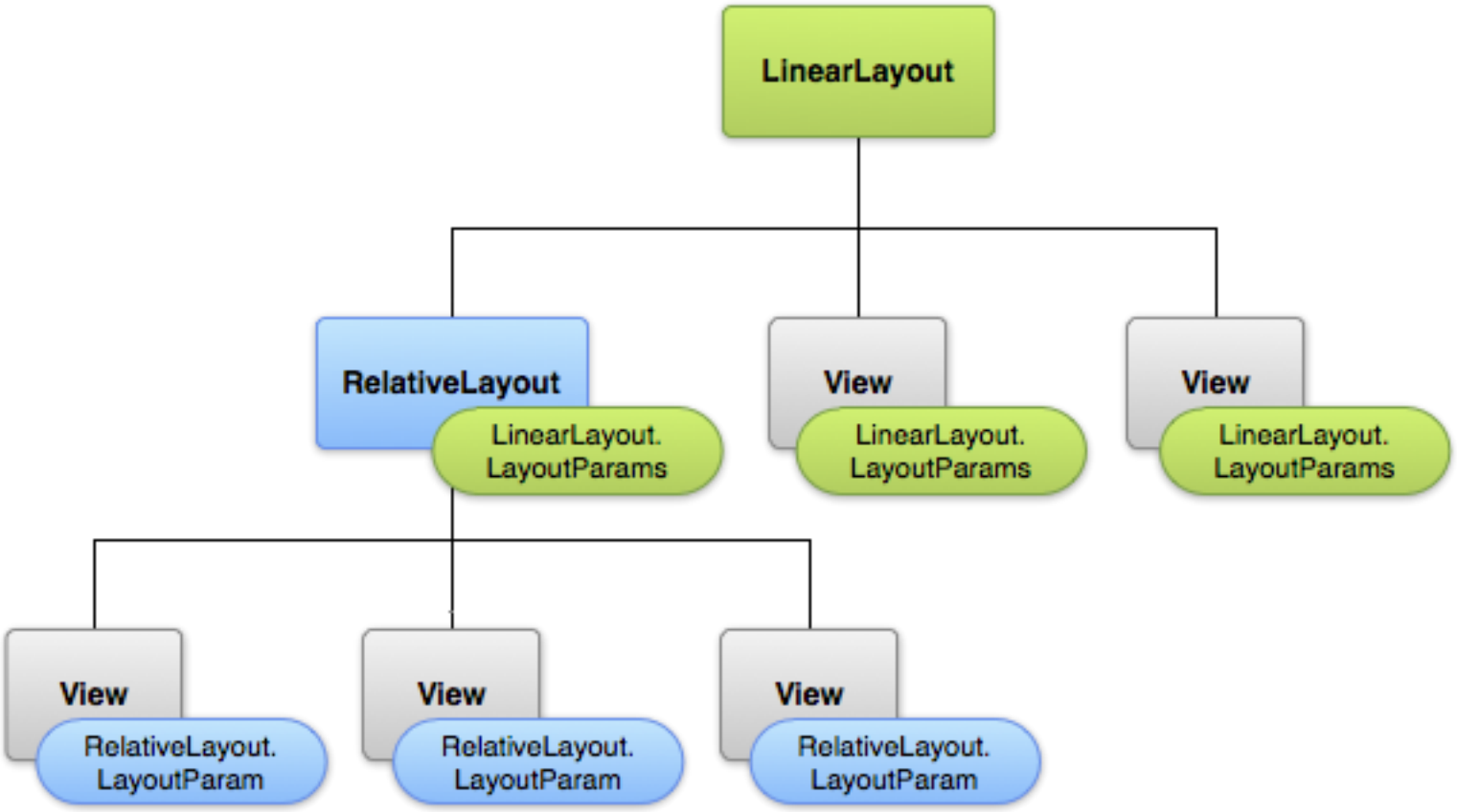
1. Tài nguyên hình ảnh

2. **Tài nguyên giao diện**
   - Thư mục lưu trữ – Truy xuất
   - Các định dạng Layout

2.1 Thư mục lưu trữ – Truy xuất

- **Thư mục lưu trữ:**
  - Các tài nguyên giao diện được lưu trữ trong thư mục res/layout.
  - Có thể có nhiều thư mục layout theo từ hạn định khác nhau:
    - Ví du: layout-land, layout-xhdpi…
- **Truy xuất: bao gồm 2 cách thức:**
  - Java: R.layout.<tên tài nguyên>.
  - XML: @[pakage:]layout/<tên tài nguyên>.

2.1 Thư mục lưu trữ – Truy xuất



2.1 Thư mục lưu trữ – Truy xuất

- **Ví dụ khai báo trong XML: activity_main.xml**

```
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical" >

      <ImageView
         android:id="@+id/imgScale"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:contentDescription="@null" />

</LinearLayout>
```

2.2 Các định dạng Layout

- **Bao gồm các lớp kế thừa từ ViewGroup:**
  - AbsoluteLayout (**Deprecated**)
  - AdapView (ListView, Gridview…)
  - DrawerLayout
  - FragmentBreadCrumbs
  - **FrameLayout**
  - GridLayout
  - **LinearLayout**
  - PagerTitleStrip
  - **RelativeLayout**
  - SlidingDrawer
  - SlidingPaneLayout
  - SwipeRefreshLayout
  - ViewPager

2.2 Các định dạng Layout

- **FrameLayout:**
  - Sử dụng trong các trường hợp xây dựng bố cục tổ chức hiển thị một đối tượng duy nhất.
  - Đối tượng mặc định vị trí top-left trên FrameLayout, có thể sử dụng thuộc tính Gravity để thiết lập lại vị trí.
  - Ví dụ khai báo:

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

2.2 Các định dạng Layout

- **FrameLayout:**
  - Các đối tượng kế thừa phổ biến:
    - **ViewFlipper**: đối tượng cho phép thực hiện hiển thị các đối tượng ở chế độ phân trang, chỉ hiển thị một đối tượng ở một thời điểm.
      - Ví dụ khai báo:

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

      - Các phương thức sử dụng:
        - startFlipping
        - setAutoStart
        - showNext
        - showPrevious

2.2 Các định dạng Layout

- **FrameLayout:**
  - Các đối tượng kế thừa phổ biến:
    - **ScrollView**: đối tượng cho phép thực hiện hiển thị các đối tượng ở chế độ cuộn màn hình, chỉ cho phép chứa một đối tượng ở một thời điểm.
      - Ví dụ khai báo:

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

      - Các phương thức sử dụng:
        - setFillViewPort
        - scrollBy
        - scrollTo
        - smoothScrollBy
        - smoothScrollTo

2.2 Các định dạng Layout

- **LinearLayout:**
  - Sử dụng trong các trường hợp xây dựng bố cục tổ chức hiển thị các đối tượng theo một chiều duy nhất (chiều dọc hoặc ngang).
  - Đối tượng mặc định vị trí top left trên LinearLayout , có thể sử dụng thuộc tính Gravity để thiết lập lại vị trí.
  - Ví dụ khai báo:

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
</LinearLayout>
```

2.2 Các định dạng Layout

- **LinearLayout:**
  - **TableLayout**: đối tượng layout kế thừa từ LinearLayout, cho phép hiển thị các đối tượng theo nhiều dòng (TableRow).
  - Mỗi dòng có thể chứa nhiều View, mỗi View được xem là một cột.
  - Ví dụ khai báo:

```
<TableLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
        <Tablerow>
            <Button/>
        </Tablerow>
</TableLayout>
```

2.2 Các định dạng Layout

- **RelativeLayout:**
  - Sử dụng trong các trường hợp xây dựng bố cục tổ chức hiển thị các đối tượng theo mối quan hệ vị trí.
  - Đối tượng được đặt vào RelativeLayout đầu tiên sẽ xác định vị trí cho các đối tượng sau đó.
  - Ví dụ khai báo:

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

Thảo luận