Software Practice 3

- Today’s lecture
- Today’s Task
- Porting Android App. in real device

Prof. Hwansoo Han
T.A. Jeonghwan Park 43
ADVANCED UI
Past Practices

- We learned about
  - single **Activity** and **Layout**
  - dynamic actions from an **Activity**
Today’s work

- Still work in a single Activity, but do lots of intelligent things
  - Split an Activity into multiple Layouts
  - Create and manage ListView
    - Inflater
    - Adapter
Issues for Multiple Layouts

- A **Layout** is allowed to contain other **Layouts**
- All **Layouts** in a **Layout** are not automatically separated (Manually declared)
  - Padding and Margin
- **Generate View from layout written in XML**
  - Inflater
Padding and Margin

Hello Android
HorizontalScrollView

- Horizontal version of **LinearLayout**
- Scrollable but not focusable and clickable
- Not **Layout**!
  - Only one **View/Layout** can be located in this component
  - Then, insert **Layout** to get multiple **Views**!
Multiple Layout

- layout/layout_main.xml
- layout_marginBottom
  - set margin with given value
  - marginBottom(@id/linearLayout) = height(@id/horizontalScrollView)
- LinearLayout for main content
  - will be changed via menu
- HorizontalScrollView for menu bar
Multiple Layout

- layout/layout_main.xml
  - @string/main_page = main
  - @string/animal_page = settings
  - @string/my_page = my

- clickable must be collocated with focusable
Multiple Layout

- layout/view_content.xml
- This **Layout** will be located in @id/linearLayout
- Change the content defined by action of menu
- choiceMode=singleChoice
  - Each element enabled to click only once at a time
- Initially have nothing in ListView
Activity for Multiple Layout

- **LayoutInflater**
  - Convert resources which declared in XML to View
  - Commonly used when generating new View or ViewGroup in Java source code, such as Activity or Adapter
  - setContentView method of Activity class is also implemented using LayoutInflater

- **Warning!**
  - Activity cannot directly get the attributes defined in XML
  - You have to parse XML by yourselves for doing so
Four Ways to Create Inflater

- **Context # getSystemService()**

  ```java
  LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
  View view = inflater.inflate(R.layout.my_layout, parent, false);
  ```

- **Activity # getLayoutInflater()**

  ```java
  LayoutInflater inflater = getLayoutInflater();
  ```

- **LayoutInflater # from**

  ```java
  LayoutInflater inflater = LayoutInflater.from(context);
  View view = inflater.inflate(R.layout.my_layout, parent, false);
  ```

- **View # inflate**

  ```java
  View view = View.inflate(context, R.layout.my_layout, parent);
  ```
Example of Inflating

- **Definition of inflate()**

  ```java
  inflate(int resource, ViewGroup root, boolean attachToRoot);
  ```

- **Example source code**

  ```java
  // /android/view/LayoutInflator.java
  // Temp is the root view that was found in the xml View temp;
  
  ... 
  ViewGroup.LayoutParams params = null;
  if (root != null) {
    // Create layout params that match root, if supplied params = root.generateLayoutParams(attrs);
    if (!attachToRoot) {
      // Set the layout params for temp if we are not attaching.
      temp.setLayoutParams(params);
    }
  }
  ...
  
  if (root != null && attachToRoot) {
    root.addView(temp, params);
  }
  ```
ListView

- A **ViewGroup** which groups several items and display them in vertical scrollable list
- The list items are automatically inserted to the list using an **Adapter**
  - Adapter pulls content from a source such as an array or database
- **Constructor of Adapter**
  - `ArrayAdapter(Context context, int resource, List<T> objects);`
  - context: super class of Activity
  - resource: id of a resource
  - objects: data to be inserted
public class MainActivity extends Activity {
    ListView mListView;
    @Override
    public void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        setContentView(R.layout.layout_main);
        mListView = findViewById(R.id.listView);

        ArrayList<String> data = new ArrayList<>();
        data.add("lion");
        data.add("tiger");
        data.add("dog");
        data.add("cat");
        ArrayAdapter<String> adapter = new ArrayAdapter<>(this, R.layout.view_item, data);
        mListView.setAdapter(adapter);
    }
}
[Lab – Practice #3]

- Implement your application satisfying the following conditions
  - Three buttons on the bottom of Activity for changing main_content area (LinearLayout)
    - first menu: list integers of range \([1, 10]\) in increasing order
    - second menu: list images which are given at i-campus
    - third menu: list information of you with following order
      - student id -> name -> department -> college -> university
  - These buttons must be resized to be scrollable
  - Use default configuration for implementing ListView in main_content area
**Example**

Third
2018111111
Gildong Hong
Department of Software
College of Software
Sungkyunkwan University