

# Android UI Part 3

Menu / Spinner / ListView

# Menu

- Physical or Logical
- Menu(3.0+ icon won't display)
  - 選項功能表(OptionsMenu)
  - 上下文功能表(ContextMenu)
  - 子功能表(SubMenu)
- Ch 6-2

# OptionsMenu

- Override **onCreateOptionsMenu(...)**
- Menu **add()**方法的四個參數,依次是:
  - 組別,如果不分組的話就寫 Menu.NONE
  - Id,用來確定不同的功能表
  - 順序,那個功能表現在在前面由這個參數的大小決定
  - 文字,功能表的顯示文字
- **setIcon()**方法為功能表設置圖示
- override **onOptionsItemSelected(...)** 處理被選擇時的邏輯

# OptionsMenu - Code

@Override

```
public boolean onCreateOptionsMenu(Menu menu) {  
    menu.add(Menu.NONE, Menu.FIRST + 1, 5, "刪除").setIcon(android.R.drawable.ic_menu_delete);  
    menu.add(Menu.NONE, Menu.FIRST + 2, 2, "儲存").setIcon(android.R.drawable.ic_menu_edit);  
    menu.add(Menu.NONE, Menu.FIRST + 3, 6, "說明").setIcon(android.R.drawable.ic_menu_help);  
    menu.add(Menu.NONE, Menu.FIRST + 4, 1, "新增").setIcon(android.R.drawable.ic_menu_add);  
    menu.add(Menu.NONE, Menu.FIRST + 5, 4, "明細").setIcon(android.R.drawable.ic_menu_info_details);  
    menu.add(Menu.NONE, Menu.FIRST + 6, 3, "傳送").setIcon(android.R.drawable.ic_menu_send);  
  
    return true;  
}
```

# OptionsMenu - XML

@Override

```
public boolean onCreateOptionsMenu(Menu menu) {  
    // Inflate the menu; this adds items to the action bar if it is present.  
    getMenuInflater().inflate(R.menu.menu_options, menu);  
    return true;  
}
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
```

```
    <item android:id="@+id/item01" android:icon="@android:drawable/ic_menu_add" android:title="新增"></item>
```

```
    <item android:id="@+id/item02" android:icon="@android:drawable/ic_menu_send" android:title="傳送"></item>
```

```
    ....
```

```
</menu>
```

# 功能表選項被選擇事件

// 功能表選項被選擇事件

@Override

public boolean onOptionsItemSelected(MenuItem item) {

    Log.i("MENU",item.getItemId()+"");

    String title = item.getTitle().toString();

    Toast.makeText(this, title+"選項被點擊了",Toast.LENGTH\_LONG).show();

}

# ContextMenu

- 註冊 **registerForContextMenu(:View)**
- Override **onCreateContextMenu(...)**
- Menu **add()**方法的四個參數,依次是:
  - 組別,如果不分組的話就寫 Menu.NONE
  - Id,用來確定不同的功能表
  - 順序,那個功能表現在在前面由這個參數的大小決定
  - 文字,功能表的顯示文字
- Override **onContextItemSelected(..)** 處理被選擇時的邏輯

# ContextMenu - 1

```
private void initView() {  
  
    wholeWiew = (LinearLayout) findViewById(R.id.wholeview);  
  
    // 註冊前後文選單在整個螢幕上(LinearLayout)  
  
    registerForContextMenu(wholeWiew);  
  
}  
  
@Override  
  
public void onCreateContextMenu(ContextMenu menu, View v, ContextMenuInfo menuInfo) {  
  
    super.onCreateContextMenu(menu, v, menuInfo);  
  
    menu.add(0, 0, 1, "Option 1");  
  
    menu.add(0, 1, 2, "Option 2");  
  
    menu.add(0, 2, 3, "Option 3");  
  
}
```



# ContextMenu - 2

@Override

```
public boolean onContextItemSelected(MenuItem item) {  
  
    Toast.makeText(this, item.getTitle().toString(),  
    Toast.LENGTH_LONG).show();  
  
    return true;  
  
}
```

# SubMenu

- Override **onCreateOptionsMenu(...)**
- **Menu add()**方法的四個參數,依次是:
  - 組別,如果不分組的話就寫 Menu.NONE
  - Id,用來確定不同的功能表
  - 順序,那個功能表現在在前面由這個參數的大小決定
  - 文字,功能表的顯示文字
- **menu.addSubMenu():SubMenu**
- override **onOptionsItemSelected(...)** 處理被選擇時的邏輯

# SubMenu - Code

@Override

```
public boolean onCreateOptionsMenu(Menu menu) {  
    boolean result = super.onCreateOptionsMenu(menu);  
  
    SubMenu Menu_1 = menu.addSubMenu(0, 0, 0, "Menu_1");  
  
    Menu_1.add(0, 1, 3, "Sub Menu 1 1");  
  
    Menu_1.add(0, 2, 2, "Sub Menu 1 2");  
  
    Menu_1.add(0, 3, 1, "Sub Menu 1 3");  
  
    SubMenu Menu_2 = menu.addSubMenu(1, 4, 0, "Menu_2");  
  
    Menu_2.add(1, 5, 2, "Sub Menu 2 1");  
  
    Menu_2.add(1, 6, 1, "Sub Menu 2 2");  
  
    return result;  
}
```

# 功能表選項被選擇事件

// 功能表選項被選擇事件

@Override

public boolean onOptionsItemSelected(MenuItem item) {

    Log.i("MENU",item.getItemId()+"");

    String title = item.getTitle().toString();

    Toast.makeText(this, title+"選項被點擊了",Toast.LENGTH\_LONG).show();

}

# Spinner

- 透過Adapter設定呈現資料與方式
- 資料來源可以設定在res/values/strings.xml
- Ch 7-1-8

# String Array

```
<resources>
```

```
.....
```

```
<string-array name="colors">
```

```
<item>黑色 | Black</item>
```

```
<item>藍色 | Blue</item>
```

```
<item>棕色 | Brown</item>
```

```
<item>綠色 | Green</item>
```

```
<item>灰色 | Grey</item>
```

```
<item>粉色 | Pink</item>
```

```
<item>紫色 | Purple</item>
```

```
<item>紅色 | Red</item>
```

```
.....
```

```
</string-array>
```

```
</resources>
```

# Spinner 初始化

```
private void initView() {  
  
    spinner = (Spinner) findViewById(R.id.spinner1);  
  
    // 準備一個陣列配接器  
  
    ArrayAdapter adapter = ArrayAdapter.createFromResource(this,  
        R.array.colors, android.R.layout.simple_spinner_item);  
  
    // 設置下拉樣式  
  
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);  
  
    // 為下拉清單設置配接器  
  
    spinner.setAdapter(adapter);  
  
}
```

# Spinner 選項被選到的

```
spinner.setOnItemSelectedListener(new OnItemSelectedListener() {  
  
    @Override  
  
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {  
  
        Toast.makeText(MainActivity.this,"選擇的色彩: "+  
parent.getItemAtPosition(position).toString(),Toast.LENGTH_LONG).show();  
  
    }  
  
    @Override  
  
    public void onNothingSelected(AdapterView<?> parent) {  
  
    }  
  
});
```



# ListView

- 透過Adapter設定呈現資料與方式(可以客製化)
- 資料來源可以設定在res/values/strings.xml
- setOnItemSelectedListener
- setOnItemLongClickListener
- Ch 7-1-1 ~ 7-1-3

# ListView - 1

```
String[] data = {.... };
```

```
lv.setAdapter(new ArrayAdapter(this,  
                                android.R.layout.simple_list_item_1,  
                                data));
```

# ListView - 2

```
lv.setOnItemClickListener(new ListView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> parent, View view,  
        int position, long id) {  
        Toast.makeText(getApplicationContext(),  
            "您點擊的是:" + ((TextView) view).getText(),  
            Toast.LENGTH_SHORT).show();  
    }  
});
```

# ListView - 3

```
lv.setOnItemLongClickListener(new ListView.OnItemLongClickListener() {  
  
    @Override  
  
    public boolean onItemLongClick(AdapterView<?> parent, View view,  
  
        int position, long id) {  
  
        Toast.makeText(getApplicationContext(),  
  
            "您長按的是:" + ((TextView) view).getText(),  
  
            Toast.LENGTH_SHORT).show();  
  
        return false;  
  
    }  
  
});
```

# Custom Adapter - 1

```
ArrayList<String> list = new ArrayList<String>(10);
```

```
MyAdapter adapter = new MyAdapter(this,list);
```

```
lv.setAdapter(adapter);
```

# Custom Adapter -2

```
public class MyAdapter extends MyBaseAdapter<String> {  
    public MyAdapter(Context context, ArrayList<String> dataList) {  
        super(context, dataList);  
    }  
    ...  
    public View getView(final int position, View convertView, ViewGroup parent) {  
        View view = inflater.inflate(R.layout.adapter_mycell, null);  
        TextView courseNameText = (TextView) view.findViewById(R.id.textView1);  
        courseNameText.setText(dataList.get(position));  
        .....  
        return view;  
    }  
}
```

# Custom Adapter - 3

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" android:id="@+id/rowLayout"
    android:layout_width="match_parent" android:layout_height="match_parent">

    <ImageView

        android:id="@+id/imageView1" android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:src="@drawable/ic_launcher" />

    <TextView android:id="@+id/textView1" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="Large Text"
        android:textAppearance="?android:attr/textAppearanceLarge" android:textSize="36sp" />

</LinearLayout>
```

# Custom Adapter - 4

```
public abstract class MyBaseAdapter<T> extends BaseAdapter{  
  
    protected Context context;  
  
    protected ArrayList<T> dataList;  
  
    public void setDataList(ArrayList<T> dataList) {  
  
        this.dataList = dataList;  
  
        //notifyDataSetChanged();  
  
    }  
  
    protected LayoutInflater inflater;  
  
    protected abstract void rowSelected(T song, int index);  
  
    ....(continue)
```



# Custom Adapter - 5

```
public MyBaseAdapter(Context context, ArrayList<T> dataList) {  
    this.context = context;  
    this.inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);  
    this.dataList = dataList;  
}  
  
@Override  
public int getCount() {  
    return dataList.size();  
}  
  
@Override  
public T getItem(int position) {  
    return dataList.get(position);  
}
```

# Custom Adapter - 6

@Override

```
public long getItemId(int position) {  
    return position;  
}
```

```
public ArrayList<T> getDataList() {  
    return dataList;  
}
```

```
}
```