

Embedded System Project Laboratory

嵌入式系統專題實驗

實習 (1)

# 嵌入式系統開發環境及 Linux 使用介紹

授課教師：曾嘉影 老師

班級：I4A

座號：28

姓名：黃昭維

題目：

試建立一專案,修改 layout 和 values,使得模擬器畫面輸出” Hello World, Hello 你的學號!” 及不同的字體大小、字體顏色、背景顏色...等豐富的畫面。

實習內容及問題說明：

藉由第一次的 Android 程式開發，來熟悉 Android Project 的架構以及流程。讓我們能夠了解到以下這些檔案或路徑再 Android Project 中的意義：

src/ = 程式放置的地方

gen/R.java = ADT 自動生成的資源文件,不允許用戶編輯的,上面保存有各個資源的 ID

res/ drawable/ = 用來存放圖片文件

res/ layout/ = 用來存放版面配置文件

res/ values/ = 用來存放一些變數、參數等文件

AndroidManifest.xml = 應用程式功能清單，描述程式主要功能

程式架構或類別繼承圖：



## Main.xml 主程式版面：

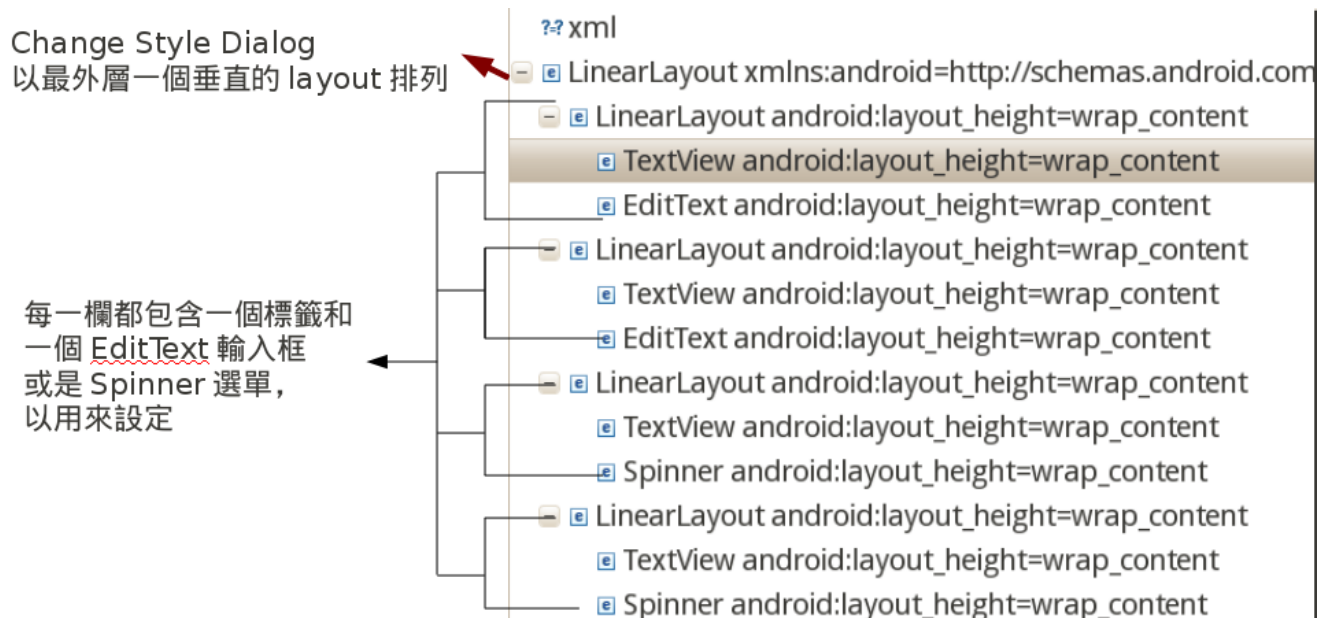
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:id="@+id/view">
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello"
        android:id="@+id/textView_msg"
    />

    <RelativeLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">
        <Button
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_alignParentBottom="true"
            android:text="Change Style"
            android:id="@+id/btn_style"
        />
    </RelativeLayout>
</LinearLayout>
```

主程式畫面非常單純，只有一個 TextView 用來顯示輸入的內容，

以及在畫面最下方產生一個 Button，點擊即可修改變更風格

## dialog.xml 對話框版面：



實習程式及輸出畫面：

```
public class ESHW1Activity extends Activity {
    /** Called when the activity is first created. */
    private String[] colors = {"Red", "Yellow", "Green", "Cyan", ...
    private String[] hexColors = {"#FF0000", "#FFFF00", "#00FF00", "#00FFFF", ...
    private String ui_text = "Hello World, Hello 410006128.";
    private int ui_size = 40;
    private int ui_foreground = 0; // red
    private int ui_background = 6; // white

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        // get textView component
        TextView textView = (TextView) findViewById(R.id.textView_msg);
        LinearLayout view = (LinearLayout) findViewById(R.id.view);
        // set default text
        textView.setText(ui_text);
        // set font size
        textView.setTextSize(TypedValue.COMPLEX_UNIT_DIP, ui_size);
        // set foreground color
        textView.setTextColor(Color.parseColor(hexColors[ui_foreground]));
        // set background color
        view.setBackgroundColor(Color.parseColor(hexColors[ui_background]));

        Button btn_style = (Button) findViewById(R.id.btn_style);
        btn_style.setOnClickListener(btnHandler);
    }

    View.OnClickListener btnHandler = new View.OnClickListener() {
        public void onClick(View v) {
            // 點擊 Button 後
            // 產生一個以 dialog.xml 為版面的對話框
            LayoutInflater inflater = getLayoutInflater();
            View layout = inflater.inflate(R.layout.dialog, (ViewGroup) findViewById(R.id.layout_dialog));
            AlertDialog.Builder dialog = new AlertDialog.Builder(ESHW1Activity.this);

            ArrayAdapter<String> colorList = new ArrayAdapter<String>(ESHW1Activity.this, ...);
            colorList.setDropDownViewResource(android.R.layout.select_dialog_item);

            final Spinner spinner_foreground = (Spinner) layout.findViewById(R.id.spinner_foreground);
            spinner_foreground.setAdapter(colorList);
            spinner_foreground.setSelection(ui_foreground); // 將目前 TextView 的相關資訊顯示在對話框之中

            final Spinner spinner_background = (Spinner) layout.findViewById(R.id.spinner_background);
            spinner_background.setAdapter(colorList);
            spinner_background.setSelection(ui_background);

            final EditText editText_text = (EditText) layout.findViewById(R.id.editText_text);
            final EditText editText_fontSize = (EditText) layout.findViewById(R.id.editText_fontSize);
            editText_text.setText(ui_text);
            editText_fontSize.setText("" + ui_size);

            dialog.setTitle("Style")
                .setView(layout)
                .setPositiveButton("OK", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface d, int arg1) {
                        // 按下 OK 的選項，則會套用新設定的 Style
                        TextView textView = (TextView) findViewById(R.id.textView_msg);
                        ui_text = editText_text.getText().toString();
                        textView.setText(ui_text);
                        ui_size = Integer.parseInt(editText_fontSize.getText().toString());
                        textView.setTextSize(TypedValue.COMPLEX_UNIT_DIP, ui_size);

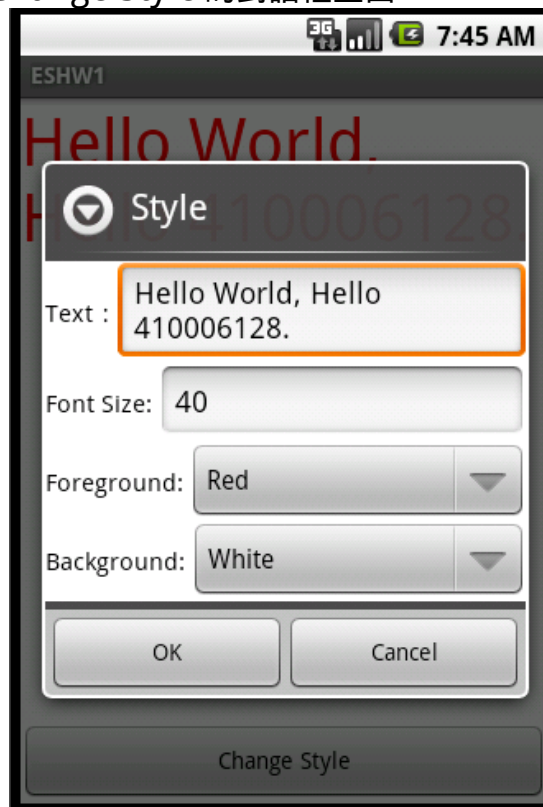
                        ui_foreground = spinner_foreground.getSelectedItemPosition();
                        textView.setTextColor(Color.parseColor(hexColors[ui_foreground]));
                        ui_background = spinner_background.getSelectedItemPosition();
                        LinearLayout view = (LinearLayout) findViewById(R.id.view);
                        view.setBackgroundColor(Color.parseColor(hexColors[ui_background]));
                    }
                })
                .setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface d, int arg1) {
                        // 反之，按下 Cancel 的選項，則不會有任何改變，回到原本的畫面
                    }
                })
                .show();
        }
    };
}
```

1. 程式初始畫面：



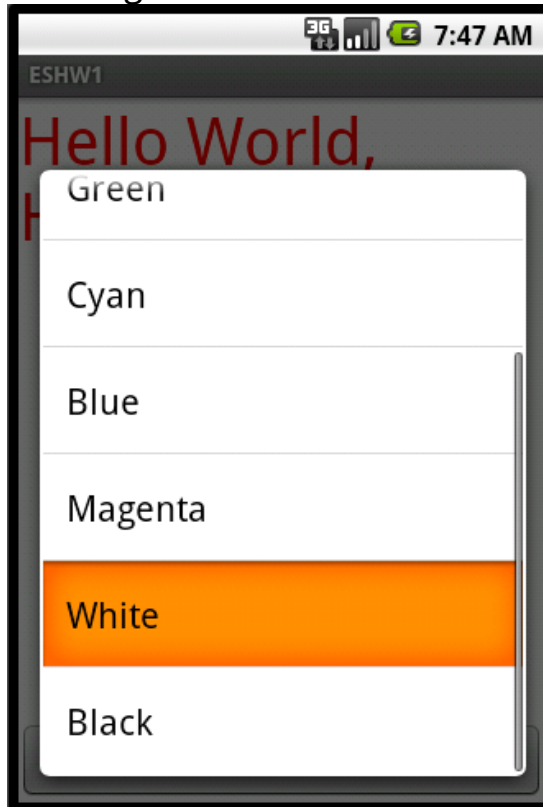
程式一開始的初始畫面，可以點擊下方的 Change Style 按鈕來設定新的風格。

2. Change Style 的對話框畫面：



有四個欄位，由上至下可以設定字串內容、字體大小（單位 dp）、字體顏色、背景顏色等

3. 選擇 foreground 顏色的畫面：



利用清單的方式列出可以選擇的顏色，有 Red, Yellow, Green, Cyan, Blue, Magenta, White, Black 共 8 種顏色可選擇。

4. 設定新的內容、大小、顏色：



將字串內容改為 Hello, I am 410006128. 字體大小 35dp，字體顏色為白色，背景為洋紅。

5. 按下 OK 套用新的設定畫面：



按下 OK，  
對話框消失且成功套用新的 Style。

學習紀錄：

因為之前已經有過 Android 的程式開發經驗，所以第一次作業沒有什麼太大的問題，唯一的問題是之前的 Android 開發都是在 Windows 的平台下作業而且使用得 Android SDK 版本也比較新（4.x 版本），第一次使用 Linux 平台上的 Eclipse 進行 2.X 版本的開發，有一些變動跟習慣跟之前都不一樣，需要一些時間來適應。

思考分析：

第一次的作業難度不高，很適合沒有開發過 Android 的新手，且對於 Linux 的環境不是很熟悉的情況下，也使用了常見的 Eclipse IDE 環境來進行開發，這樣大大地減少了不同作業系統上的差異，讓我們不熟悉 Linux 的初學者也能很快的開發 Android，而本次作業的自由度很高，只要求我們改變內容、字體的大小顏色，還有背景顏色，但是每個人所作出的結果一定是不盡相同的，我認為讓每個人在這次作業上加入自己的設計風格和不同的樣貌，也是這次作業的重點之一。