Android程式設計

動態密碼顯示

王昱景 Brian Wang brian.wang.frontline@gmail.com



1. 題目說明:

請開啓 COM.TQC.GDD01 專案,設計「動態密碼顯示」應用程式。欄位包 含輸入帳號與輸入密碼的 EditText 元件,以及設定顯示密碼的 CheckBox 元件。輸入帳號的欄位預設只能輸入數字,輸入密碼的欄位預設以密碼方 式顯示,當【顯示密碼】的 CheckBox 被勾選時,則將所輸入的密碼直接以 明碼顯示出來,請依下列題意完成作答。

デ 記 動態密碼顕示	显 上午 2:45
請輸入帳號	
請輸入密碼	
舅示密碼	

2. 設計說明:

- (1) main.xml 為 Activity 所使用的 Layout, 已設計好程式内所需使用的元件。
- (2) color.xml 及 strings.xml 中已定義好程式内所需使用的顏色及文字。
- (3) 開啓 main.xml,設定 Layout 的背景色為已定義好的 white。
- (4) 設定 id 為 mAccount 的 EditText 元件,使 EditText 在還未輸入文字時,顯示已定義好的字串 str1,當有文字被輸入時,顯示的字串 str1 便會消失,同時設定此 EditText 僅能輸入數字。

- (5) 設定 id 為 mPassword 的 EditText 元件,使 EditText 在還未輸入文字時,顯示已定義好的字串 str2,當有文字被輸入時,顯示的字串 str2 便會消失,同時設定此 EditText 以密碼的型態顯示(輸入值會以【·】呈現)。
- (6) 開啓 GDD01.java,實作 CheckBox 的 OnCheckedChangeListener,當【顯示密碼】的核取方塊被勾選時,輸入的密碼將以明碼直接顯示。
- (7) 當【顯示密碼】的核取方塊未被勾選時,輸入的密碼將以密碼方式顯示 (輸入值以【·】呈現)。

3. 執行結果參考畫面:

(1) 當【顯示密碼】的核取方塊未被勾選時,輸入的密碼將以密碼方式顯示(輸入值以·呈現)。

学和 動態密码研示	□ 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1234567	
顯示密碼	

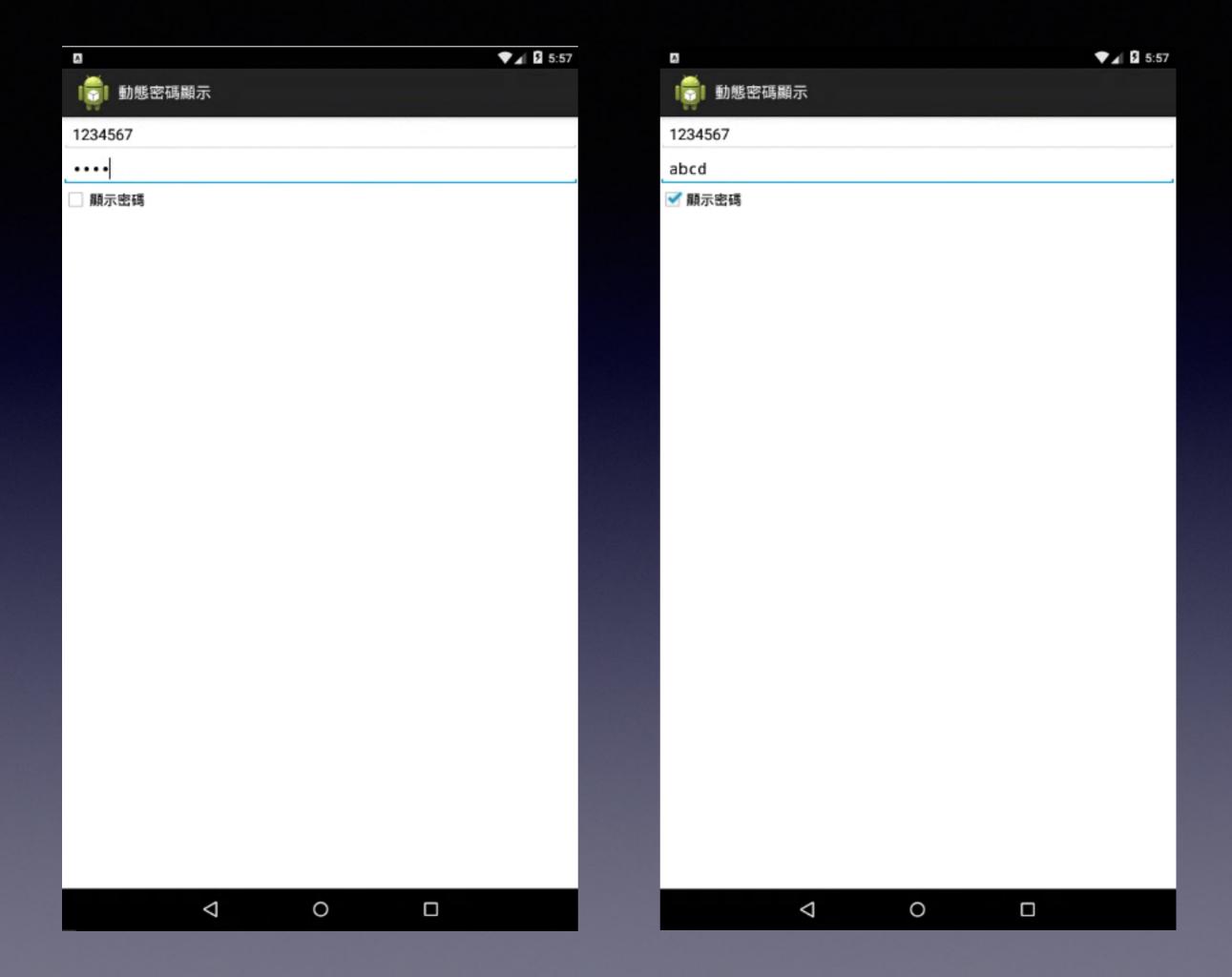
(2) 當【顯示密碼】的核取方塊被勾選時,輸入的密碼將以明碼直接顯示。



- 4. 自行測試程式是否達成下列結果:
 - (1) 程式執行時畫面背景為白色。
 - (2) 輸入帳號的 EditText 在未輸入值前會顯示【請輸入帳號】,當有文字被輸入時,【請輸入帳號】字樣會消失。
 - (3) 輸入密碼的 EditText 在未輸入值前會顯示【請輸入密碼】,當有文字被輸入時,【請輸入密碼】字樣會消失。
 - (4) 輸入帳號的 EditText 僅能輸入數字。
 - (5) 輸入密碼的 EditText 預設以密碼方式呈現。
 - (6) 當【顯示密碼】的核取方塊被勾選時,輸入的密碼以明碼方式呈現。
 - (7) 當【顯示密碼】的核取方塊被取消勾選時,輸入的密碼被再度隱藏。

5. 評分項目:

項	配分	得分
(1) 程式執行時畫面背景為白色		
(2) 輸入帳號的 EditText 在未輸入値前會顯示【請輸入帳號】, 當有文字被輸入時, 【請輸入帳號】字 樣會消失		
(3) 輸入密碼的 EditText 在未輸入值前會顯示【請輸入密碼】, 當有文字被輸入時, 【請輸入密碼】字 樣會消失		
(4) 輸入帳號的 EditText 僅能輸入數字		
(5) 輸入密碼的 EditText 預設以密碼方式呈現		
(6) 當【顯示密碼】的核取方塊被勾選時,輸入的密碼以明碼方式呈現		
(7) 當【顯示密碼】的核取方塊被取消勾選時,輸入 的密碼被再度隱藏		
總	20	



DynamicPassword

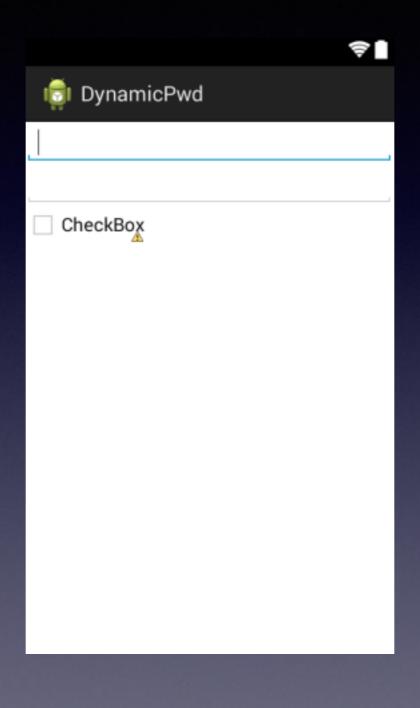
- 開啟和執行 Android 專案
- 建立元件的版面配置
- 建立 Activity 活動類別

開啟和執行 Android 專案

- 請啟動 Eclipse IDE
- 建立 Android 專案
 - Project Name: DynamicPwd
 - Build Target: Android 4
 - Package Name: tw.edu.vnu.dynamicpwd

建立元件的版面配置

- 刪除 "Hello World" TextView
- 刪除 string.xml 的 hello_world,修改 app_name 為"動態密碼顯示"
- 將 RelativeLayout 改為 LinearLayout (Vertical)
- 在 LinearLayout (Vertical) 依序垂直編排
 EditText (Number) 、EditText (Password) 和
 CheckBox 元件



- 在 string.xml 新增以下字串
 - str1 為 "請輸入帳號"
 - str2 為 "請輸入密碼"
 - str3 為 "顯示密碼"

```
<pre
```

• 新增 color.xml 並增加 drawable 屬性設定

• black 為 "#000000"

• white 為 "#FFFFFF"

- LinearLayout 的 background 設定為 white
- 將第一個 EditText 的 id 設定為 mAccount, hint 設定為 str1, textSize 設定為 18sp
- 第二個 EditText 的 id 設定為 mPassword, hint 設定為 str2, textSize 設定為 18sp
- CheckBox 的 id 設定為 mCheck, text 設定為 str3, textColor 設定為 black



```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"</pre>
  1
          xmlns:tools="http://schemas.android.com/tools"
          android:layout_width="match_parent"
          android: layout height="match parent"
  4
  5
          android:background="@drawable/white"
  6
          android:orientation="vertical" >
  7
  8
         <EditText
  9
              android:id="@+id/mAccount"
 10
              android:layout_width="match_parent"
 11
              android: layout_height="wrap_content"
              android:hint="@string/str1"
 12
              android:inputType="number"
 13
              android:textSize="18sp" />
 14
 15
 16
         <EditText
 17
              android:id="@+id/mPassword"
 18
              android: layout width="match parent"
 19
              android: layout_height="wrap_content"
              android:hint="@string/str2"
 20
              android:password = "true"
60n21
              android:textSize="18sp" 7>
 22
 23
         <CheckBox
 24
 25
              android:id="@+id/mCheck"
 26
              android: layout_width="wrap_content"
 27
              android: layout_height="wrap_content"
 28
              android:text="@string/str3"
 29
              android:textColor="@drawable/black" />
 30
 31
     </LinearLayout>
 32
```

建立 Activity 活動類別

- 宣告成員變數
- onCreate() 方法
- setOnClickListener() 事件處理方法

宣告成員變數

```
package tw.edu.vnu.dynamicpwd;
 2
 3
    import android.app.Activity;
   import android.os.Bundle;
 4
   import android.text.method.HideReturnsTransformationMethod;
   import android.text.method.PasswordTransformationMethod;
    import android.view.View;
    import android.view.View.OnClickListener;
    import android.widget.CheckBox;
    import android.widget.EditText;
10
11
12
    public class MainActivity extends Activity {
13
        private EditText pwd;
14
15
        private CheckBox cb;
16
        @Override
17
18
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
19
            setContentView(R.layout.activity main);
20
21
```

onCreate() 方法

```
@Override
17
        protected void onCreate(Bundle savedInstanceState) {
18
            super.onCreate(savedInstanceState);
19
            setContentView(R.layout.activity_main);
20
21
            pwd = (EditText) findViewById(R.id.mPassword);
22
            cb = (CheckBox) findViewById(R.id.mCheck);
23
24
25
26
                @Override
                public void onClick(View view) {
27
28
                     if(cb.isChecked()) {
                         pwd.setTransformationMethod(HideReturnsTransformationMethod.getInstance());
29
30
                     } else {
31
                         pwd.setTransformationMethod(PasswordTransformationMethod.getInstance());
32
33
            });
34
35
36
```

setOnClickListener() 事件處理方法

```
@Override
17
18
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
19
20
            setContentView(R.layout.activity_main);
21
            pwd = (EditText) findViewById(R.id.mPassword);
22
                   CheckBox) findViewByTd(R id mCheck):
23
24
            cb.setOnClickListener(new OnClickListener() {
25
26
                @Override
                public void onClick(View view) {
27
28
                     if(cb.isChecked()) {
                         pwd.setTransformationMethod(HideReturnsTransformationMethod.getInstance());
29
30
                     } else {
31
                         pwd.setTransformationMethod(PasswordTransformationMethod.getInstance());
32
33
34
35
36
```