

Lab 2: Android Programming Report

學號：0811562

姓名：何祁恩

1. 請說明下列程式作用

```
<intent-filter>
```

```
    <action android:name="android.intent.action.MAIN" />
```

```
    <category android:name="android.intent.category.LAUNCHER"/>
```

```
</intent-filter>
```

Intent 有分隱式和顯式 intent，隱式 intent 不會直接指定啟動哪一個 activity，而是通過 intent-filter 中的 action、category、data(此程式中沒有 data 這一行)，來篩選符合這些引數的 activity。

action: 是用來說明此 activity 可以被何種功能的 intent 啟動，像是題目上的 "android.intent.action.MAIN" 的功能就是當 app 被啟動時會第一個被啟動的 activity。也就是說，一個 activity 可以有很多個 action。

category: 是用來說明此 activity 會出現在哪一個列表中，一般來說，intent-filter 中至少都會有一個 category name 是 android.intent.category.DEFAULT，也就是會被加到預設的列表中。

而題目上的：`<action android:name="android.intent.action.MAIN" />`,`<category android:name="android.intent.category.LAUNCHER" />` 是用來表明此 activity 是當 app 被啟動時，第一個被啟動的 activity。

當要啟動 app 時，intent-filter 就會作為篩選 activity 機制中的條件，一定要 action 為 MAIN 且有在 LAUNCHER category 中的 activity 才能作為最先被啟動的 activity。

2. 請貼上自己的程式碼並附上註解

MainActivity.java:

```
1  package com.example.guessnumberapp;
2  import androidx.appcompat.app.AppCompatActivity;
3  import android.content.Intent;
4  import android.os.Bundle;
5  import android.view.View;
6  import android.widget.Button;
7  import android.widget.EditText;
8
9  public class MainActivity extends AppCompatActivity {
10
11      EditText studentID;
12      EditText name;
13      Button startButton;
14
15      @Override
16      protected void onCreate(Bundle savedInstanceState) {
17          super.onCreate(savedInstanceState);
18          setContentView(R.layout.activity_main);
19
20
21          studentID = (EditText)findViewById(R.id.studentID);
22          name = (EditText)findViewById(R.id.name);
23          startButton = (Button)findViewById(R.id.startButton);
24
25          startButton.setOnClickListener(new View.OnClickListener(){
26              @Override
27              public void onClick(View v){
28
29                  //using intent to switch activity
30                  Intent intent = new Intent();
31                  intent.setClass(MainActivity.this, MainActivity2.class);
32
33                  //an object to put many variables into
34                  //and pass those variables to the next page through intent
35                  Bundle bundle = new Bundle();
36                  bundle.putString("studentID",studentID.getText().toString());
37                  bundle.putString("name",name.getText().toString());
38
39                  //pass bundle to intent
40                  intent.putExtras(bundle);
41                  startActivity(intent);
42              }
43          });
44      }
45  }
```

在第 30 及 31 行，初始化了一個 intent，透過 intent 切換 activity，從 MainActivity 跳至 MainActivity2(遊戲介面)。

在 35 至 37 行，透過 bundle 將欲傳往下一個 activity 的資料打包，最後透過 intent.putExtras()將整個 bundle 傳給 intent，將其帶入下一個 activity。

MainActivity2.java:

```
1  package com.example.guessnumberapp;
2  import androidx.appcompat.app.AppCompatActivity;
3  import android.content.Intent;
4  import android.os.Bundle;
5  import android.view.View;
6  import android.widget.Button;
7  import android.widget.EditText;
8  import android.widget.TextView;
9  import java.util.regex.Pattern;
10
11  public class MainActivity2 extends AppCompatActivity {
12
13      EditText GuessNumber;
14      Button enter;
15      TextView hint;
16      TextView numGuess;
17      TextView BestRecord;
18      Button restartButton;
19      Button endButton;
20
21      //parameter in guessing game
22      int randNum = 0, history = 999;
23      int min = 1, max = 50, count = 0;
24
25      @Override
26      protected void onCreate(Bundle savedInstanceState) {
27          super.onCreate(savedInstanceState);
28          setContentView(R.layout.activity_main2);
29
30          //using this pointer to get the bundle from the intent
31          //and extract the parameter from the last page
32          Bundle bundle = this getIntent().getExtras();
33          String studentID = (String)bundle.getString("studentID");
34          String name = (String)bundle.getString("name");
35
36          GuessNumber = (EditText)findViewById(R.id.GuessNumber);
37          enter = (Button)findViewById(R.id.enter);
38          hint = (TextView)findViewById(R.id.hint);
39          numGuess = (TextView)findViewById(R.id.numGuess);
40          BestRecord = (TextView)findViewById(R.id.BestRecord);
41          restartButton = (Button)findViewById(R.id.restartButton);
42          endButton = (Button)findViewById(R.id.endButton);
43
44          //get a random number from 0 ~ 50
45          //Math.random return a number between 0 ~ 0.9999999999999999 approach 1
46          randNum = (int)(Math.random() * 50 + 1);
47      }
```

```

48 enter.setOnClickListener(new View.OnClickListener(){
49     @Override
50     public void onClick(View v){
51
52         //assert the user give the non-number input, check if the string isNumber
53         String possibleNumber = GuessNumber.getText().toString();
54         boolean isNumber = Pattern.matches("[0-9]+", possibleNumber);
55         if(!isNumber){
56             hint.setText("請不要輸入數字以外的東西");
57             return;
58         }
59
60         //make sure the user give the number as the input,
61         //otherwise, Integer.parseInt() will make app crash
62         int in = Integer.parseInt(GuessNumber.getText().toString());
63
64         //if the user give a valid input, increase the count
65         count++;
66         numGuess.setText("猜測次數 : " + count);
67
68         //tune the range for the min and max, and give the corresponding hint
69         if(in ≤ max && in ≥ min){
70             if(in > randNum){
71                 max = in;
72                 hint.setText("請輸入"+ min + "~" + max + "的數字");
73             }
74             else if(in < randNum){
75                 min = in;
76                 hint.setText("請輸入"+ min + "~" + max + "的數字");
77             }
78
79             //make a correct guess, compare the current count to the best record (history)
80             else if(in == randNum){
81                 hint.setText("恭喜猜對");
82                 if(count < history){
83                     history = count;
84                     BestRecord.setText("最佳紀錄 : " + history);
85                 }
86             }
87         }
88
89         //make sure the user input the range between min ~ max
90         else{
91             hint.setText("請輸入"+ min + "~" + max + "的數字，請輸入正常值");
92         }
93     }
94 };
95

```

```

96 //reset the guessing number game parameter and reset count
97 restartButton.setOnClickListener(new View.OnClickListener(){
98     @Override
99     public void onClick(View v){
100         randNum = (int)(Math.random() * 50 + 1);
101         min = 0;
102         max = 50;
103         count = 0;
104         hint.setText("請輸入"+ min + "~" + max + "的數字");
105         numGuess.setText("猜測次數 : " + count);
106     }
107 });
108
109 //pass the studentID, name , and best record (history) to the next activity
110 endButton.setOnClickListener(new View.OnClickListener(){
111     @Override
112     public void onClick(View v){
113         Intent intent = new Intent();
114         intent.setClass(MainActivity2.this, MainActivity3.class);
115         Bundle bundle = new Bundle();
116         bundle.putString("studentID", studentID);
117         bundle.putString("name", name);
118         bundle.putInt("history", history);
119         intent.putExtras(bundle);
120         startActivity(intent);
121     }
122 });
123
124 }
125

```

在 32 到 34 行，將前一個 activity 傳來的 bundle 打開，得到前一個介面所得到的學號及姓名，留待未來使用。必須等到 onCreate() 之後才能將其解壓縮出原始的資料，否則 app 會閃退。

在 22 到 23 及 46 行，設定了遊戲所需要的參數，並產生了介於 1~50 之間的隨機數字讓使用者猜測。History 則是用來記錄使用者最佳的紀錄(最少的猜測次數)，count 則是記錄當次猜測已經使用了幾次。

當使用者按下 enter 的按鈕後，在 53~58 行，我會先將使用者輸入的字串分析是否為數字，否則當使用者輸入非數字而原本的 code 中對於非數字的字串使用 Integer.parseInt() 會造成 app 閃退，因此我將這部分的漏洞排除掉了。當使用者輸入非數字的字串，我的提示會改成”請不要輸入數字以外的東西”，以提示使用者好好遵守遊戲規則。

而後 68~94 就開始比較使用者猜測的數字在目前的範圍中為過小或是過大，進而更新範圍提示使用者。當使用者成功猜對數字後，比對 count 與 history 以更新最佳紀錄。

當使用者點擊重新開始的按鍵，會將遊戲的參數初始化，但是最佳紀錄仍需留存。當使用者點擊結束遊戲的按鈕後，便會將學號姓名及最佳紀錄打包進 bundle 而後透過 intent 傳至最後的結算畫面。

MainActivity3.java:

```
1  package com.example.guessnumberapp;
2  import androidx.appcompat.app.AppCompatActivity;
3  import android.content.Intent;
4  import android.os.Bundle;
5  import android.view.View;
6  import android.widget.Button;
7  import android.widget.TextView;
8
9  public class MainActivity3 extends AppCompatActivity {
10
11      TextView studentID;
12      TextView name;
13      TextView history;
14      Button BackButton;
15
16      @Override
17      protected void onCreate(Bundle savedInstanceState) {
18          super.onCreate(savedInstanceState);
19          setContentView(R.layout.activity_main3);
20
21          studentID = (TextView)findViewById(R.id.studentID);
22          name = (TextView)findViewById(R.id.name);
23          history = (TextView)findViewById(R.id.history);
24          BackButton = (Button)findViewById(R.id.BackButton);
25
26          //extract the data from last 2 activity
27          Bundle bundle = this getIntent().getExtras();
28          String studentIDTEXT = (String)bundle.getString("studentID");
29          String nameTEXT = (String)bundle.getString("name");
30          int historyTEXT = (int)bundle.getInt("history");
31
32          //show the data
33          studentID.setText("學號:      " + studentIDTEXT);
34          name.setText("姓名:      " + nameTEXT);
35          history.setText("最佳紀錄:      " + historyTEXT);
36
37          //go back to the main activity
38          BackButton.setOnClickListener(new View.OnClickListener(){
39              @Override
40              public void onClick(View v){
41                  Intent intent = new Intent();
42                  intent.setClass(MainActivity3.this, MainActivity.class);
43                  startActivity(intent);
44              }
45          });
46      }
47  }
```

在 27~30 行，將 activity2 的 bundle 打開得到學號姓名及最佳紀錄，而後將其秀在結算畫面上。當使用者點擊返回的按鈕便會跳回到第一個開始的介面，將遊戲重新開始。

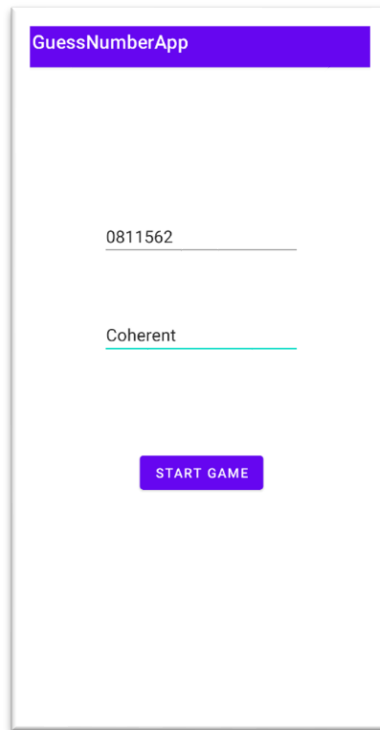
3. 心得

這次的實驗還蠻有趣的，學到如何製作跳轉頁面的 app，除此之外，也學到了 intent 及 bundle 的用法，其實有點像是過去在 C/C++ 程式設計中，透過 struct 將東西打包，而後透過指向 struct 的指標將其傳遞到想要使用的 function。透過這幾次的實驗，我更加了解在 android studio 是如何透過物件導向程式設計控制個別的物件，並且透過 getter 及 setter 設定這些物件的 property。

這次的實驗蠻有趣的，製作出了一個猜數字的小遊戲，為了快速猜到結果，每次在猜數字都用 binary search 的方式，希望能夠盡快猜到數字。假如真的很衰的話，使用 binary search 最多也只需要 6 次便可以猜到答案。

4. 結果

初始畫面:



屏蔽非數字:



用生命在猜 (一次就對):

GuessNumberApp

重新開始

結束遊戲

猜數字

恭喜猜對

25

送出

猜測次數: 1

最佳紀錄: 1

猜在範圍外:

GuessNumberApp

重新開始

結束遊戲

猜數字

請輸入13~50的數字，請輸入正常值

12

送出

猜測次數: 2

最佳紀錄: 1

結算畫面:

GuessNumberApp

BACK

學號:0811562

姓名:Coherent

最佳紀錄:1