

Lab 3: Android Programming Report

學號：0811562

姓名：何祁恩

1. 請說明 Android Studio 中的 Layout 元件種類，簡單介紹不同 Layout 之間的主要差異。

在 Android Studio 中常見的 Layout 元件有以下五種：

- Linear Layout (線性佈局)

線性佈局又可以根據 Orientation 的不同分成水平或是垂直。垂直表示內部的物件會由上往下堆疊，而水平表示內部的物件會由左而右堆疊。

- Relative Layout (相對佈局)

透過物件與物件間的相對位置關係來排放物件。透過 id 來定位物件要擺放在這個物件的上下左右。並不是給定絕對的位置，而是使用相對的位置來擺放，這樣比較不會因為不同品牌的手機而造成佈局失效。

- Table Layout (表格佈局)

讓物件像表格一樣排列的佈局，長得很像 excel，看起來不太討喜，但是簡單容易上手。透過<TableRow> </TableRow>夾住該行要放置的物件，相當直覺的一種佈局方式。

- Absolute Layout (絕對佈局)

給定絕對的 x,y 座標來指定該物件要被放置在哪邊，倘若開發的 app 是需要跑在不同的機型上，因為畫面大小的不同，使用絕對佈局便可能僅僅侷限在特定機種上有效，而在其他機種上失效。

- Frame Layout (框架佈局)

先出現在.xml 檔的物件會出現在最後面，而後出現的會堆疊在上面。

2. 請貼上自己的程式碼並附上註解

MainActivity.java

```
1 package com.example.lab3_set;
2 import androidx.annotation.NonNull;
3 import androidx.appcompat.app.ActionBarDrawerToggle;
4 import androidx.appcompat.app.AppCompatActivity;
5 import androidx.core.view.GravityCompat;
6 import androidx.drawerlayout.widget.DrawerLayout;
7 import android.os.Bundle;
8 import android.view.Menu;
9 import android.view.MenuItem;
10 import androidx.appcompat.widget.Toolbar;
11 import com.google.android.material.navigation.NavigationView;
12 import org.jetbrains.annotations.NotNull;
13
14 public class MainActivity extends AppCompatActivity implements NavigationView.OnNavigationItemSelectedListener {
15     private DrawerLayout drawer;
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_main);
21
22         Toolbar toolbar = findViewById(R.id.toolbar);
23         setSupportActionBar(toolbar);
24
25         drawer = findViewById(R.id.drawer_layout);
26         NavigationView navigationView = findViewById(R.id.nav_view);
27         navigationView.setNavigationItemSelectedListener(this);
28
29         ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(this, drawer, toolbar, R.string.navigation_drawer_open, R.string.navigation_drawer_close);
30         drawer.addDrawerListener(toggle);
31         toggle.syncState();
32
33         if(savedInstanceState == null) {
34             getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container, new time_fragment()).commit();
35             navigationView.setCheckedItem(R.id.time);
36         }
37     }
38
39     @Override
40     public boolean onNavigationItemSelected(@NonNull @NotNull MenuItem item) {
41         switch (item.getItemId()){
42             case R.id.time:
43                 getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container, new time_fragment()).commit();
44                 break;
45             case R.id.stopwatch:
46                 getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container, new stopwatch_fragment()).commit();
47                 break;
48             case R.id.timer:
49                 getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container, new timer_fragment()).commit();
50                 break;
51         }
52
53         drawer.closeDrawer(GravityCompat.START);
54         return true;
55     }
56
57     @Override
58     public void onBackPressed() {
59         if (drawer.isDrawerOpen(GravityCompat.START)){
60             drawer.closeDrawer(GravityCompat.START);
61         }else {
62             super.onBackPressed();
63         }
64     }
65 }
66 }
```

持續的監測使用者典籍的是側選單中的哪一個 item，而後跳轉到該頁面。

Time_fragment.java

```
1 package com.example.lab3_set;
2 import android.os.Bundle;
3 import android.view.LayoutInflater;
4 import android.view.View;
5 import android.view.ViewGroup;
6 import androidx.annotation.NonNull;
7 import androidx.annotation.Nullable;
8 import androidx.fragment.app.Fragment;
9
10 public class time_fragment extends Fragment {
11     @Nullable
12     @Override
13     public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
14         return inflater.inflate(R.layout.time_fragment, container, false);
15     }
16 }
```

這頁只是簡單的秀出 Analog Clock, Digital Clock, Text Clock，因此沒有與這些物件有互動。

Stopwatch_fragment.java

```
1 package com.example.lab3_set;
2 import android.os.Bundle;
3 import android.os.SystemClock;
4 import android.view.LayoutInflater;
5 import android.view.View;
6 import android.view.ViewGroup;
7 import android.widget.Button;
8 import android.widget.Chronometer;
9 import androidx.annotation.NonNull;
10 import androidx.annotation.Nullable;
11 import androidx.fragment.app.Fragment;
12
13 public class stopwatch_fragment extends Fragment {
14     private
15     View root;
16     Button start_btn;
17     Button stop_btn;
18     Chronometer chronometer;
19
20     @Nullable
21     @Override
22     public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
23         root = inflater.inflate(R.layout.stopwatch_fragment, container, false);
24         chronometer = (Chronometer) root.findViewById(R.id.chronometer);
25         start_btn = (Button) root.findViewById(R.id.start_btn);
26         stop_btn = (Button) root.findViewById(R.id.stop_btn);
27
28         start_btn.setOnClickListener(new View.OnClickListener() {
29             @Override
30             public void onClick(View view) {
31                 chronometer.setBase(SystemClock.elapsedRealtime());
32                 chronometer.start();
33             }
34         });
35
36         stop_btn.setOnClickListener(new View.OnClickListener() {
37             @Override
38             public void onClick(View view) {
39                 chronometer.stop();
40             }
41         });
42
43         chronometer.setOnChronometerTickListener(new Chronometer.OnChronometerTickListener() {
44             @Override
45             public void onChronometerTick(Chronometer chronometer) {
46                 long time = SystemClock.elapsedRealtime() - chronometer.getBase();
47                 int s = (int)time / 1000; //ms -> s
48                 int m = (int) s / 60; //sec -> min
49                 int h = (int) m / 60; //min -> hour
50                 s = s - m * 60;
51                 m = m - h * 60;
52                 chronometer.setText(String.format("%02d:%02d:%02d", h, m, s));
53             }
54         });
55         return root;
56     }
57 }
```

當按下開始的按鈕後，設置基準時間，而後透過當下的時間與基準時間的差，便可以知道經過多久，而後將時間換算為對應的時間單位。

Timer_fragment.java

```
1 package com.example.lab3_set;
2 import android.os.Bundle;
3 import android.os.CountDownTimer;
4 import android.view.LayoutInflater;
5 import android.view.View;
6 import android.view.ViewGroup;
7 import android.widget.Button;
8 import android.widget.NumberPicker;
9 import android.widget.TextView;
10 import androidx.annotation.NonNull;
11 import androidx.annotation.Nullable;
12 import androidx.fragment.app.Fragment;
13 import org.w3c.dom.Text;
14
15 public class timer_fragment extends Fragment {
16
17     View root;
18     NumberPicker minutes;
19     NumberPicker seconds;
20     TextView counter;
21     Button begin;
22     Button reStart;
23     CountDownTimer countdown;
24     Boolean flag; //avoid multi countdown simultaneously
25
26
27     @Nullable
28     @Override
29     public View onCreateView(@NonNull @org.jetbrains.annotations.NotNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
30         root = inflater.inflate(R.layout.timer_fragment, container, false);
31         minutes = (NumberPicker) root.findViewById(R.id.minutes);
32         minutes.setMinValue(0);
33         minutes.setMaxValue(59);
34
35         seconds = (NumberPicker) root.findViewById(R.id.seconds);
36         seconds.setMinValue(0);
37         seconds.setMaxValue(59);
38
39         counter = (TextView) root.findViewById(R.id.counter);
40         begin = (Button) root.findViewById(R.id.begin);
41         reStart = (Button) root.findViewById(R.id.reStart);
42         flag = true;
43
44         begin.setOnClickListener(new View.OnClickListener(){
45             @Override
46
47             public void onClick(View v){
48                 if(flag){
49                     flag = false;
50
51                     //get the user wanted time duration
52                     int m = minutes.getValue();
53                     int s = seconds.getValue();
54
55                     //transfer into msec
56                     long time_left = (m * 60 + s) * 1000;
57                     counter.setText(String.format("%02d:%02d", m, s));
58
59                     countdown = new CountDownTimer(time_left, 1000) {
60                         @Override
61                         public void onTick(long millisUntilFinished) {
62                             int s = (int)(millisUntilFinished/1000); //ms -> s
63                             int m = (int)(s / 60); //s -> m
64                             s = s - m * 60;
65                             counter.setText(String.format("%02d:%02d", m, s));
66                         }
67
68                         @Override
69                         public void onFinish() {
70                             //when countdown finished
71                             flag = true;
72                             counter.setText("00:00");
73                         }
74                     }.start();
75                 }
76             }
77         });
78
79         reStart.setOnClickListener(new View.OnClickListener() {
80             @Override
81             public void onClick(View v) {
82                 int m = minutes.getValue();
83                 int s = seconds.getValue();
84                 counter.setText(String.format("%02d:%02d", m, s));
85                 countdown.cancel();
86                 flag = true;
87             }
88         });
89
90         return root;
91     }
92 }
```

讓使用者設定完要倒數的時間後，透過 `getValue()` 取得資料後開始倒數。布林 `flag` 用來避免多個 `countdown timer` 同時計時，只有當 `restart` 時才會重設為 `true`。

3. 心得或建議

相比 Lab1 而言，Lab2 和 Lab3 的架構比較像是一般手機 app 會出現的架構，有頁面跳轉，導航欄以清楚明瞭這個 app 的架構。經過這三次 lab，我對 Android Studio 也了解許多，感謝助教在投影片上的用心付出，寫得非常詳細，讓我們自己不用花太多時間在上網查找語法相關問題，將會用到的 function 都整理出來，非常方便我們使用!對於新手而言也非常好上手。

接下來兩次實驗就換到我比較熟悉的 python 了，不過之前都是拿來做資料視覺化或是機器學習，比較沒有使用到 GUI 的部分，非常期待下次的實驗~