測試环境：

1）Android Studio

2) java&Kotlin

電子三甲

裘翀皓

108360150

大概總結所遇到的java和kotlin在MainActivity檔案上的不同用法:

1.定義變數上:

```kotlin
val ed_name = findViewById<EditText>(R.id.ed_name)
val tv_text = findViewById<TextView>(R.id.tv_text)
val tv_name = findViewById<TextView>(R.id.tv_name)
val tv_winner = findViewById<TextView>(R.id.tv_winner)
val tv_mmora = findViewById<TextView>(R.id.tv_mmora)
val tv_cmora = findViewById<TextView>(R.id.tv_cmora)
val btn_scissor = findViewById<RadioButton>(R.id.btn_scissor)
val btn_stone = findViewById<RadioButton>(R.id.btn_stone)
val btn_paper = findViewById<RadioButton>(R.id.btn_paper)
val btn_mora = findViewById<Button>(R.id.btn_mora)
```

上圖爲kotlin定義函數

```java
public class MainActivity extends AppCompatActivity {
    private EditText ed_name;
    private TextView tv_text,tv_name,tv_winner, tv_mmora ,tv_cmora;
    private RadioButton btn_scissor, btn_stone, btn_paper;
    private Button btn_mora;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ed_name=findViewById(R.id.ed_name);
        tv_text=findViewById(R.id.tv_text);
        tv_name=findViewById(R.id.tv_name);
        tv_winner=findViewById(R.id.tv_winner);
        tv_mmora=findViewById(R.id.tv_mmora);
        tv_cmora=findViewById(R.id.tv_cmora);
        btn_scissor=findViewById(R.id.btn_scissor);
        btn_stone=findViewById(R.id.btn_stone);
        btn_paper=findViewById(R.id.btn_paper);
        btn_mora=findViewById(R.id.btn_mora);
        btn_mora.setOnClickListener(new View.OnClickListener()
```

上圖爲java定義函數

可以看到kotlin更加簡潔

當然kotlin也可以仿照java分成兩部分定義，但是就略顯重複

如下圖所示:

```kotlin
private lateinit var btn_mora: Button
```

```kotlin
btn_mora = findViewById<Button>(R.id.btn_mora)
```

2.when語句:

在Java中我們用if語句對變量做判斷:

```
if (btn_scissor.isChecked)
    tv_mmora.text = "我方出拳\n剪刀"
else if (btn_stone.isChecked)
    tv_mmora.text = "我方出拳\n石頭"
else
    tv_mmora.text = "我方出拳\n布"
```

在kotlin中我們用when語句對變量做判斷和賦值，讓程式更加簡介：

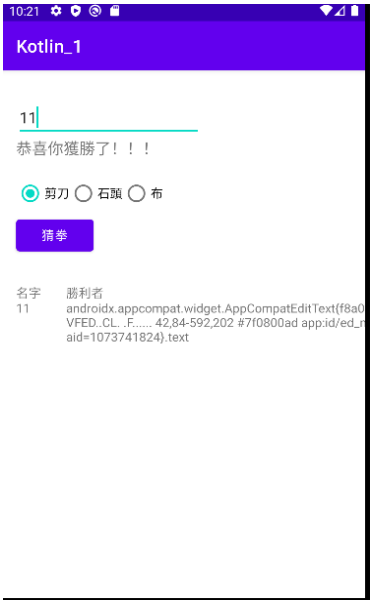
```
val playerMoraText = when {
    btn_scissor.isChecked->"剪刀"
    btn_stone.isChecked->"石頭"
    else ->"布"
}
tv_mmora.text="名字\n$playerMoraText"
```

在整個java程式中有很多if語句，都可以改成kotlin的when語句，這邊就不重複放圖片了。

3.**kotlin**語句中，在对tv_winner賦值的时候我发现，如果$ed_name.text不加{}的话

```
if (btn_scissor.isChecked && computer == 2 || bt
    tv_winner.text = "勝利者\n"+"$ed_name.text"
    tv_text.text = "恭喜你獲勝了！！！"
}
```

編譯後會變成地址:



正確的寫法如下:

```
//用三個
if (btn_scissor.isChecked && computer == 2 || btn_stone.isChecked && computer == 0 |
    tv_winner.text = "勝利者\n"+"${ed_name.text}"
    tv_text.text = "恭喜你獲勝了！！！"
}
```

如果更換成變量的話就不用:

```
val playername=ed_name.text
if (btn_scissor.isChecked && computer == 2 || btn_stone.isChecked && computer
    tv_winner.text = "勝利者\n"+"$playername"
    tv_text.text = "恭喜你獲勝了！！！"
}
```

查閱資料後得到：

$ 表示一个變量名或者變量值

$varName 表示變量值

${varName.fun()} 表示變量的方法返回值:

可变变量定义：var 关键字

```
var <标识符> : <类型> = <初始化值>
```

不可变变量定义：val 关键字，只能赋值一次的变量(类似Java中final修饰的变量)

```
val <标识符> : <类型> = <初始化值>
```

常量与变量都可以没有初始化值,但是在引用前必须初始化

编译器支持自动类型判断,即声明时可以不指定类型,由编译器判断。

```
val a: Int = 1
val b = 1    // 系统自动推断变量类型为Int
val c: Int   // 如果不在声明时初始化则必须提供变量类型
c = 1        // 明确赋值


var x = 5    // 系统自动推断变量类型为Int
x += 1       // 变量可修改
```

專案已經上傳到github，地址在最後

參考資料:

https://www.runoob.com/kotlin/kotlin-basic-syntax.html

Github:

https://github.com/547320328/Android_project_108360150/blob/master/Kotlin_1/app/src/main/java/com/example/kotlin_1/MainActivity.kt