110360129 林軒至 應用軟體實習 Lab1

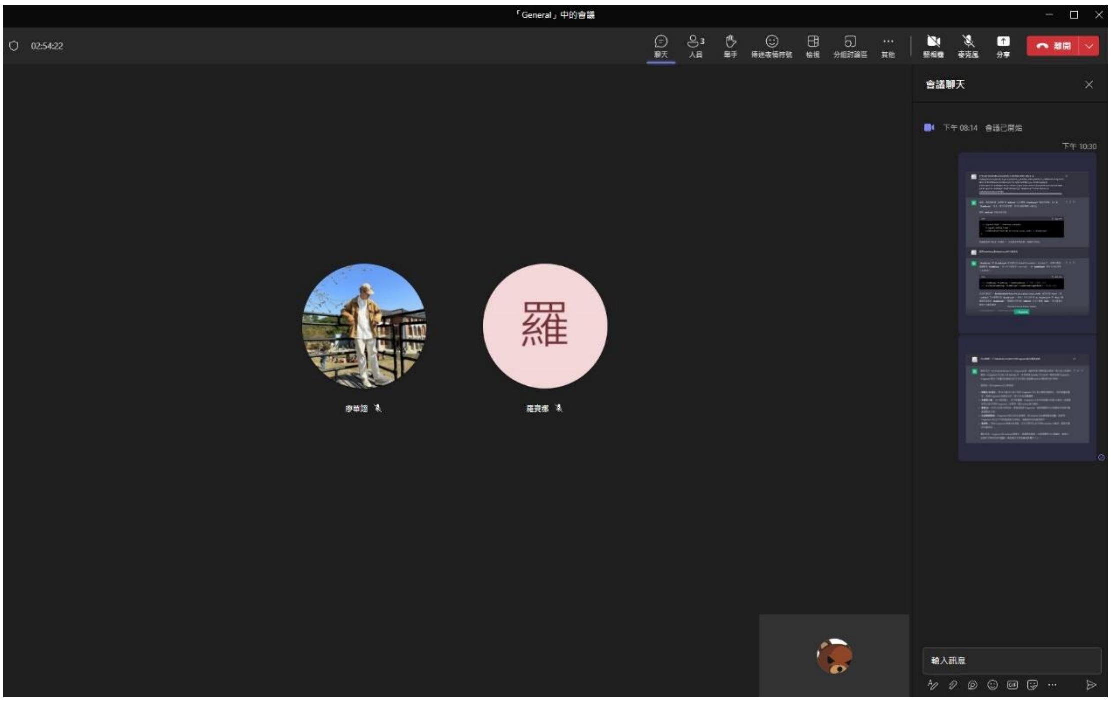
讀書會

組員:
110360121 廖華翊
110360129 林軒至
110360130 楊協盛
110360155 羅寶娜

討論時間: 2023/11/8 19:00~22:00

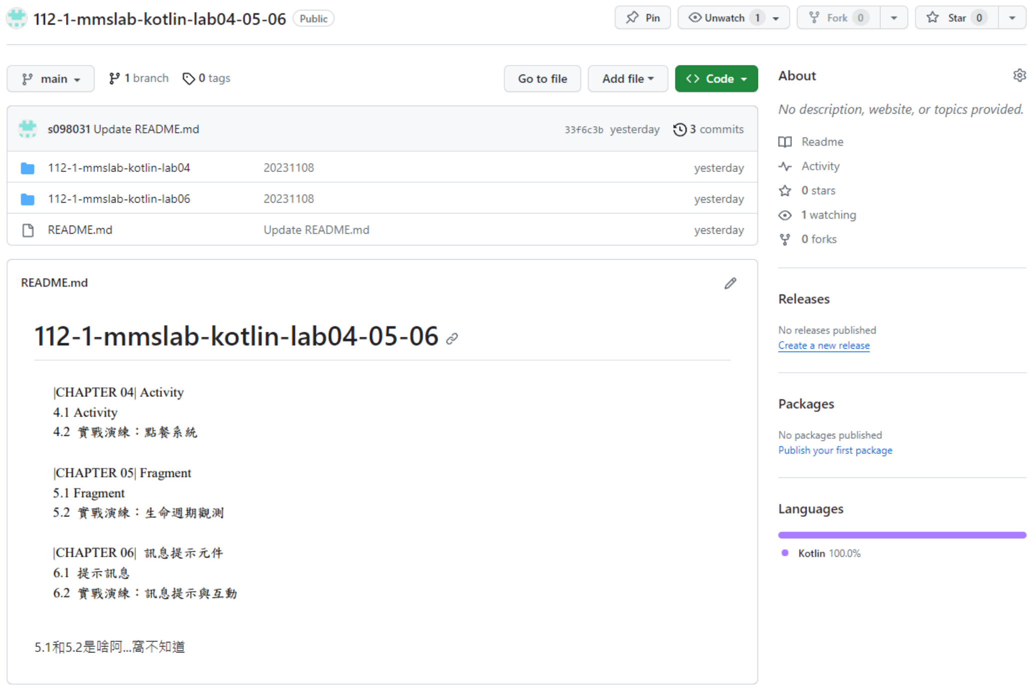
地點:Teams 線上開會 Line 討論

開會截圖:



Github:

https://github.com/s098031/112-1-mmslab-kotlin-lab04-05-06



s098031 Update README.md 33f6c3b yesterday 3 commits

112-1-mmslab-kotlin-lab04	20231108	yesterday
112-1-mmslab-kotlin-lab06	20231108	yesterday
README.md	Update README.md	yesterday

No description, website, or topics provided.

Readme
Activity
0 stars
1 watching
0 forks

No releases published
Create a new release

No packages published
Publish your first package

README.md

112-1-mmslab-kotlin-lab04-05-06

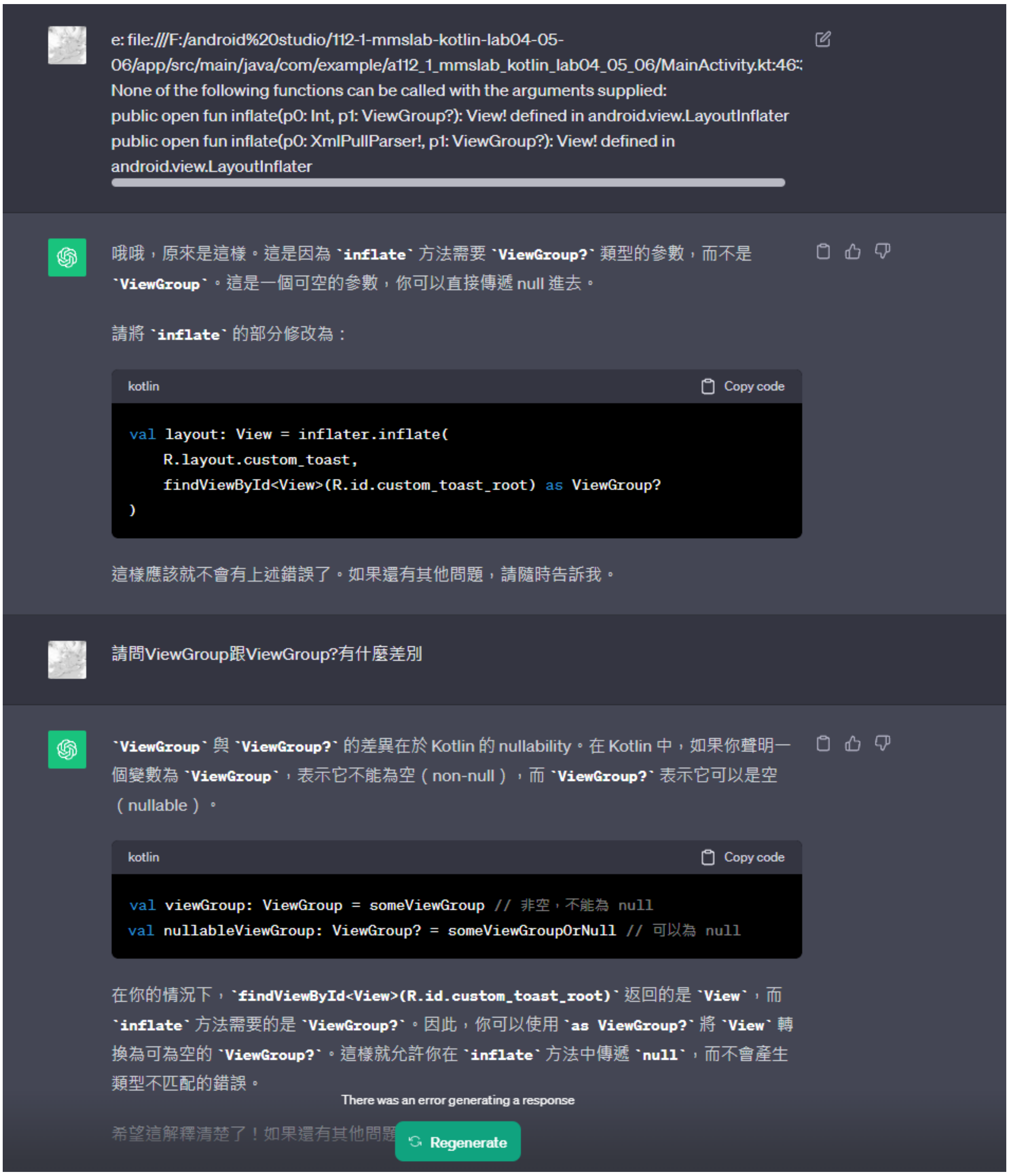
|CHAPTER 04| Activity
4.1 Activity
4.2 實戰演練：點餐系統

|CHAPTER 05| Fragment
5.1 Fragment
5.2 實戰演練：生命週期觀測

|CHAPTER 06| 訊息提示元件
6.1 提示訊息
6.2 實戰演練：訊息提示與互動

5.1和5.2是啥阿...窩不知道

問 ChatGPT 問題



```
e: file:///F:/android%20studio/112-1-mmslab-kotlin-lab04-05-06/app/src/main/java/com/example/a112_1_mmslab_kotlin_lab04_05_06/MainActivity.kt:46:
None of the following functions can be called with the arguments supplied:
public open fun inflate(p0: Int, p1: ViewGroup?): View! defined in android.view.LayoutInflater
public open fun inflate(p0: XmlPullParser!, p1: ViewGroup?): View! defined in android.view.LayoutInflater
```

哦哦，原來是這樣。這是因為 `inflate` 方法需要 `ViewGroup?` 類型的參數，而不是 `ViewGroup`。這是一個可空的參數，你可以直接傳遞 null 進去。

請將 `inflate` 的部分修改為：

```kotlin
val layout: View = inflater.inflate(
    R.layout.custom_toast,
    findViewById<View>(R.id.custom_toast_root) as ViewGroup?
)
```

這樣應該就不會有上述錯誤了。如果還有其他問題，請隨時告訴我。

請問ViewGroup跟ViewGroup?有什麼差別

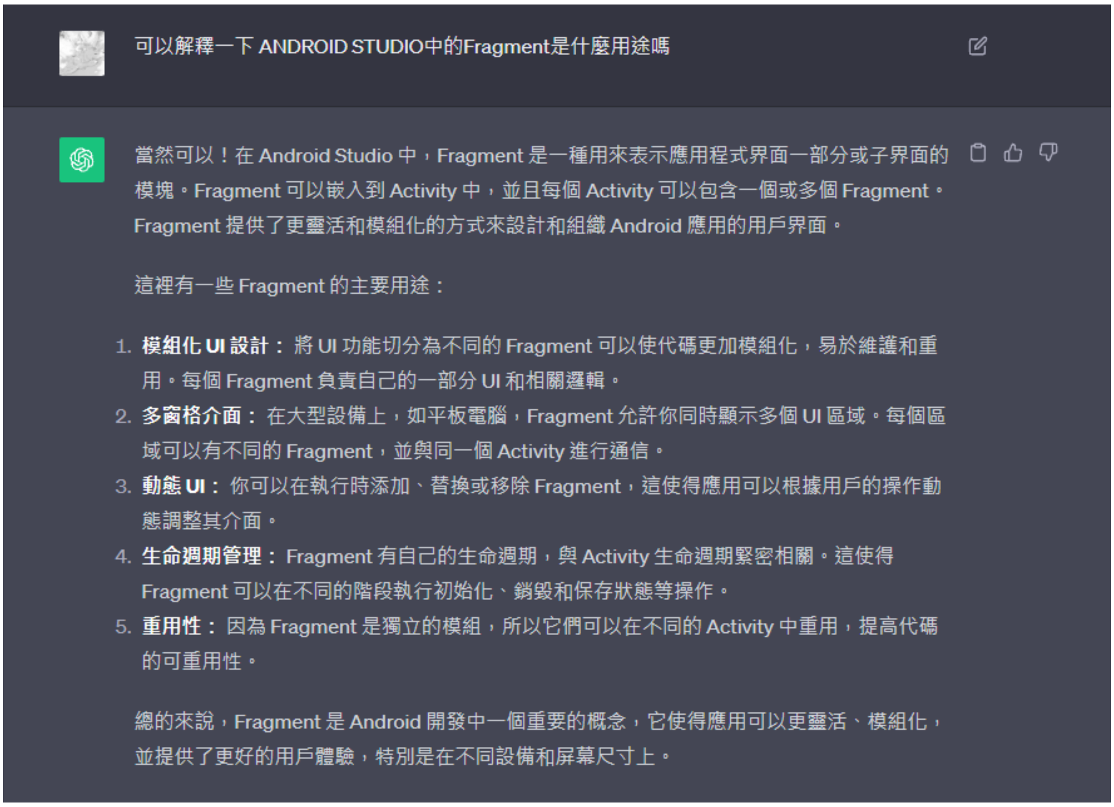
`ViewGroup` 與 `ViewGroup?` 的差異在於 Kotlin 的 nullability。在 Kotlin 中，如果你聲明一個變數為 `ViewGroup`，表示它不能為空（non-null），而 `ViewGroup?` 表示它可以是空（nullable）。

```kotlin
val viewGroup: ViewGroup = someViewGroup // 非空，不能為 null
val nullableViewGroup: ViewGroup? = someViewGroupOrNull // 可以為 null
```

在你的情況下，`findViewById<View>(R.id.custom_toast_root)` 返回的是 `View`，而 `inflate` 方法需要的是 `ViewGroup?`。因此，你可以使用 `as ViewGroup?` 將 `View` 轉換為可為空的 `ViewGroup?`。這樣就允許你在 `inflate` 方法中傳遞 `null`，而不會產生類型不匹配的錯誤。

There was an error generating a response

希望這解釋清楚了！如果還有其他問題



可以解釋一下 ANDROID STUDIO中的Fragment是什麼用途嗎

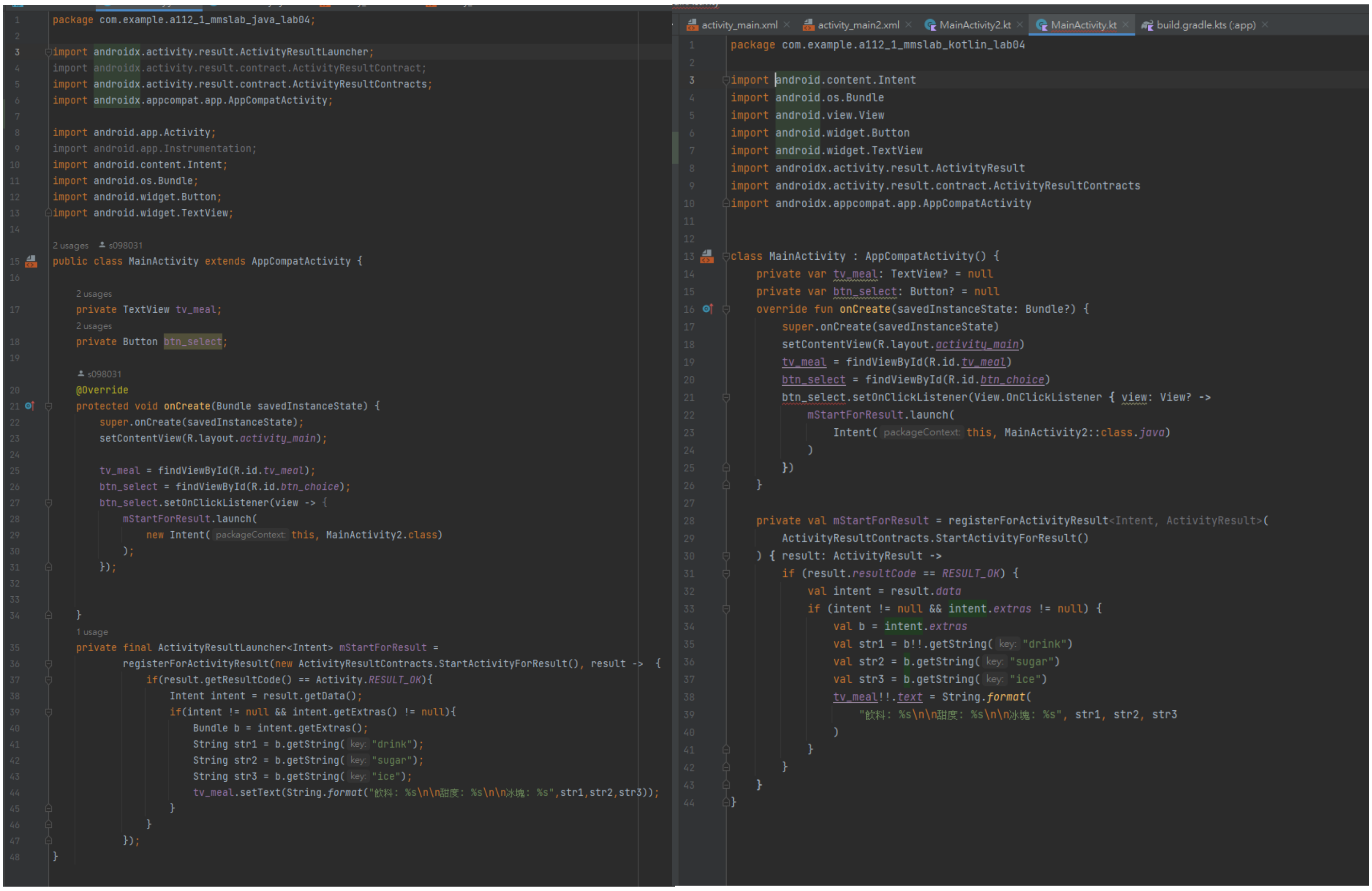
當然可以！在 Android Studio 中，Fragment 是一種用來表示應用程式界面一部分或子界面的模塊。Fragment 可以嵌入到 Activity 中，並且每個 Activity 可以包含一個或多個 Fragment。Fragment 提供了更靈活和模組化的方式來設計和組織 Android 應用的用戶界面。

這裡有一些 Fragment 的主要用途：

1. **模組化 UI 設計：** 將 UI 功能切分為不同的 Fragment 可以使代碼更加模組化，易於維護和重用。每個 Fragment 負責自己的一部分 UI 和相關邏輯。
2. **多窗格介面：** 在大型設備上，如平板電腦，Fragment 允許你同時顯示多個 UI 區域。每個區域可以有不同的 Fragment，並與同一個 Activity 進行通信。
3. **動態 UI：** 你可以在執行時添加、替換或移除 Fragment，這使得應用可以根據用戶的操作動態調整其介面。
4. **生命週期管理：** Fragment 有自己的生命週期，與 Activity 生命週期緊密相關。這使得 Fragment 可以在不同的階段執行初始化、銷毀和保存狀態等操作。
5. **重用性：** 因為 Fragment 是獨立的模組，所以它們可以在不同的 Activity 中重用，提高代碼的可重用性。

總的來說，Fragment 是 Android 開發中一個重要的概念，它使得應用可以更靈活、模組化，並提供了更好的用戶體驗，特別是在不同設備和屏幕尺寸上。

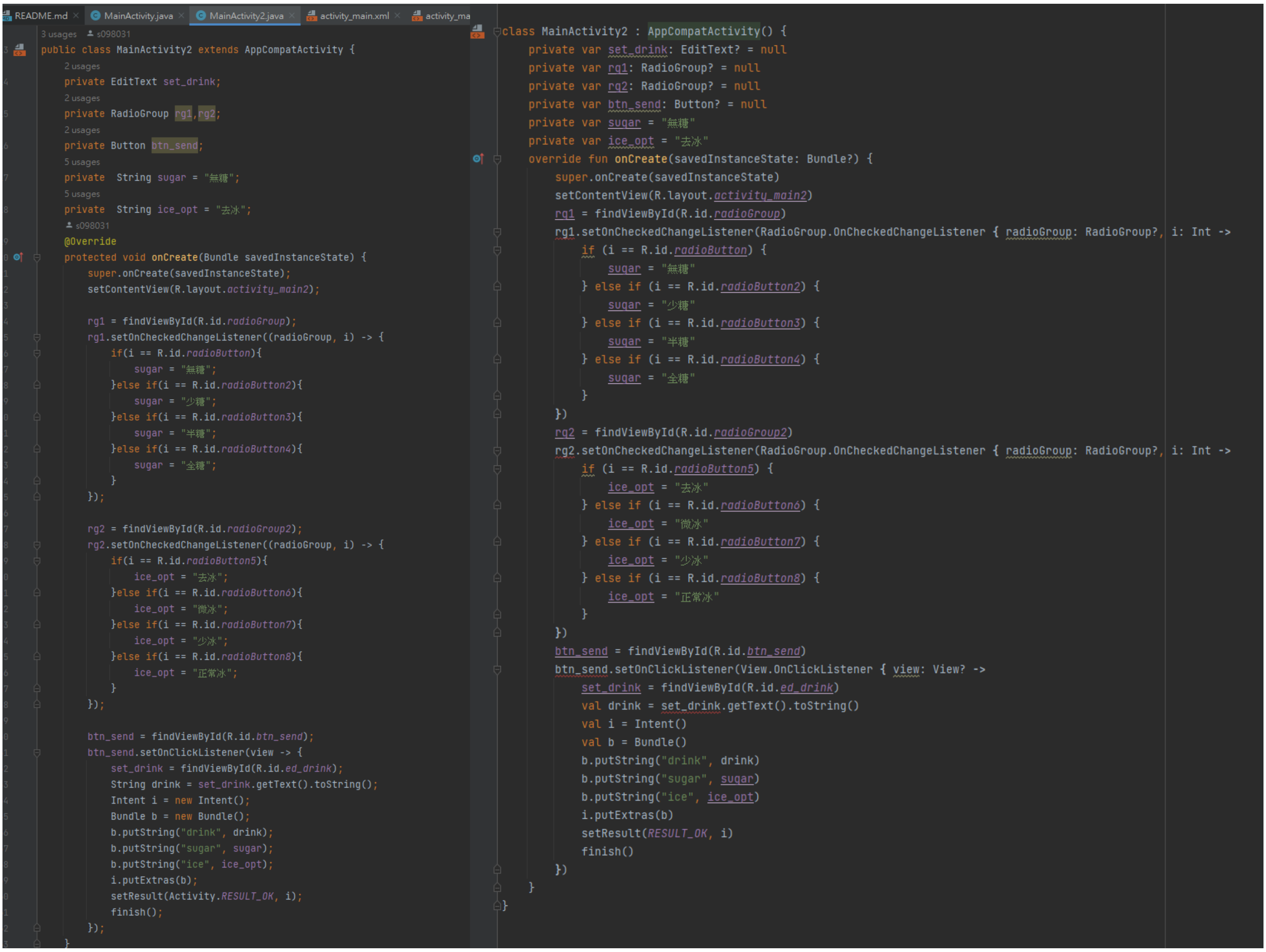
JAVA 與 KOTLIN 語法差異(左 JAVA、右 KOTLIN)



```java
package com.example.a112_1_mmslab_java_lab04;

import androidx.activity.result.ActivityResultLauncher;
import androidx.activity.result.contract.ActivityResultContract;
import androidx.activity.result.contract.ActivityResultContracts;
import androidx.appcompat.app.AppCompatActivity;

import android.app.Activity;
import android.app.Instrumentation;
import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;

2 usages  s098031
public class MainActivity extends AppCompatActivity {

    2 usages
    private TextView tv_meal;
    2 usages
    private Button btn_select;

    s098031
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        tv_meal = findViewById(R.id.tv_meal);
        btn_select = findViewById(R.id.btn_choice);
        btn_select.setOnClickListener(view -> {
            mStartForResult.launch(
                new Intent( packageContext: this, MainActivity2.class)
            );
        });


    }
    1 usage
    private final ActivityResultLauncher<Intent> mStartForResult =
            registerForActivityResult(new ActivityResultContracts.StartActivityForResult(), result ->  {
                if(result.getResultCode() == Activity.RESULT_OK){
                    Intent intent = result.getData();
                    if(intent != null && intent.getExtras() != null){
                        Bundle b = intent.getExtras();
                        String str1 = b.getString( key: "drink");
                        String str2 = b.getString( key: "sugar");
                        String str3 = b.getString( key: "ice");
                        tv_meal.setText(String.format("飲料: %s\n\n甜度: %s\n\n冰塊: %s",str1,str2,str3));
                    }
                }
            });
}
```

```kotlin
package com.example.a112_1_mmslab_kotlin_lab04

import android.content.Intent
import android.os.Bundle
import android.view.View
import android.widget.Button
import android.widget.TextView
import androidx.activity.result.ActivityResult
import androidx.activity.result.contract.ActivityResultContracts
import androidx.appcompat.app.AppCompatActivity


class MainActivity : AppCompatActivity() {
    private var tv_meal: TextView? = null
    private var btn_select: Button? = null
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        tv_meal = findViewById(R.id.tv_meal)
        btn_select = findViewById(R.id.btn_choice)
        btn_select.setOnClickListener(View.OnClickListener { view: View? ->
            mStartForResult.launch(
                Intent( packageContext: this, MainActivity2::class.java)
            )
        })
    }

    private val mStartForResult = registerForActivityResult<Intent, ActivityResult>(
        ActivityResultContracts.StartActivityForResult()
    ) { result: ActivityResult ->
        if (result.resultCode == RESULT_OK) {
            val intent = result.data
            if (intent != null && intent.extras != null) {
                val b = intent.extras
                val str1 = b!!.getString( key: "drink")
                val str2 = b.getString( key: "sugar")
                val str3 = b.getString( key: "ice")
                tv_meal!!.text = String.format(
                    "飲料: %s\n\n甜度: %s\n\n冰塊: %s", str1, str2, str3
                )
            }
        }
    }
}
```

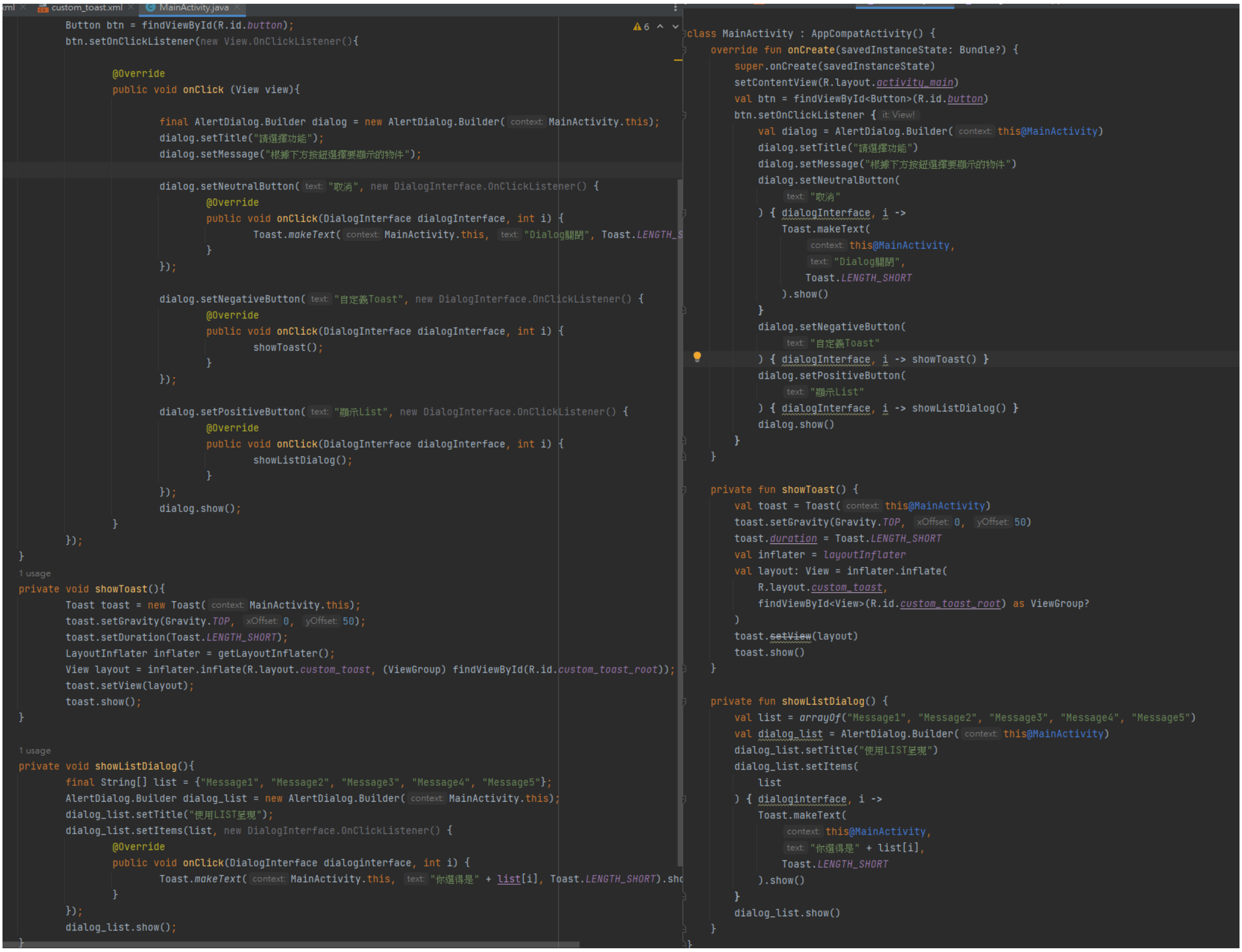
基本上差異跟上次的猜拳程式差異差不多，主要是變數和 IF 判斷語法不太一樣而已。



```java
3 usages  s098031
public class MainActivity2 extends AppCompatActivity {
    2 usages
    private EditText set_drink;
    2 usages
    private RadioGroup rg1,rg2;
    2 usages
    private Button btn_send;
    5 usages
    private  String sugar = "無糖";
    5 usages
    private  String ice_opt = "去冰";
    s098031
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);

        rg1 = findViewById(R.id.radioGroup);
        rg1.setOnCheckedChangeListener((radioGroup, i) -> {
            if(i == R.id.radioButton){
                sugar = "無糖";
            }else if(i == R.id.radioButton2){
                sugar = "少糖";
            }else if(i == R.id.radioButton3){
                sugar = "半糖";
            }else if(i == R.id.radioButton4){
                sugar = "全糖";
            }
        });

        rg2 = findViewById(R.id.radioGroup2);
        rg2.setOnCheckedChangeListener((radioGroup, i) -> {
            if(i == R.id.radioButton5){
                ice_opt = "去冰";
            }else if(i == R.id.radioButton6){
                ice_opt = "微冰";
            }else if(i == R.id.radioButton7){
                ice_opt = "少冰";
            }else if(i == R.id.radioButton8){
                ice_opt = "正常冰";
            }
        });

        btn_send = findViewById(R.id.btn_send);
        btn_send.setOnClickListener(view -> {
            set_drink = findViewById(R.id.ed_drink);
            String drink = set_drink.getText().toString();
            Intent i = new Intent();
            Bundle b = new Bundle();
            b.putString("drink", drink);
            b.putString("sugar", sugar);
            b.putString("ice", ice_opt);
            i.putExtras(b);
            setResult(Activity.RESULT_OK, i);
            finish();
        });
    }
```

```kotlin
class MainActivity2 : AppCompatActivity() {
    private var set_drink: EditText? = null
    private var rg1: RadioGroup? = null
    private var rg2: RadioGroup? = null
    private var btn_send: Button? = null
    private var sugar = "無糖"
    private var ice_opt = "去冰"
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main2)
        rg1 = findViewById(R.id.radioGroup)
        rg1.setOnCheckedChangeListener(RadioGroup.OnCheckedChangeListener { radioGroup: RadioGroup?, i: Int ->
            if (i == R.id.radioButton) {
                sugar = "無糖"
            } else if (i == R.id.radioButton2) {
                sugar = "少糖"
            } else if (i == R.id.radioButton3) {
                sugar = "半糖"
            } else if (i == R.id.radioButton4) {
                sugar = "全糖"
            }
        })
        rg2 = findViewById(R.id.radioGroup2)
        rg2.setOnCheckedChangeListener(RadioGroup.OnCheckedChangeListener { radioGroup: RadioGroup?, i: Int ->
            if (i == R.id.radioButton5) {
                ice_opt = "去冰"
            } else if (i == R.id.radioButton6) {
                ice_opt = "微冰"
            } else if (i == R.id.radioButton7) {
                ice_opt = "少冰"
            } else if (i == R.id.radioButton8) {
                ice_opt = "正常冰"
            }
        })
        btn_send = findViewById(R.id.btn_send)
        btn_send.setOnClickListener(View.OnClickListener { view: View? ->
            set_drink = findViewById(R.id.ed_drink)
            val drink = set_drink.getText().toString()
            val i = Intent()
            val b = Bundle()
            b.putString("drink", drink)
            b.putString("sugar", sugar)
            b.putString("ice", ice_opt)
            i.putExtras(b)
            setResult(RESULT_OK, i)
            finish()
        })
    }
}
```

點餐的頁面也是沒甚麼大改變，除了語法不同外其他都一樣的架構。



```java
        Button btn = findViewById(R.id.button);
        btn.setOnClickListener(new View.OnClickListener(){

                @Override
                public void onClick (View view){

                        final AlertDialog.Builder dialog = new AlertDialog.Builder( context: MainActivity.this);
                        dialog.setTitle("請選擇功能");
                        dialog.setMessage("根據下方按鈕選擇要顯示的物件");

                        dialog.setNeutralButton( text: "取消", new DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface dialogInterface, int i) {
                                        Toast.makeText( context: MainActivity.this,  text: "Dialog關閉", Toast.LENGTH_S
                                }
                        });

                        dialog.setNegativeButton( text: "自定義Toast", new DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface dialogInterface, int i) {
                                        showToast();
                                }
                        });

                        dialog.setPositiveButton( text: "顯示List", new DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface dialogInterface, int i) {
                                        showListDialog();
                                }
                        });
                        dialog.show();
                }
        });
}
1 usage
private void showToast(){
        Toast toast = new Toast( context: MainActivity.this);
        toast.setGravity(Gravity.TOP,  xOffset: 0,  yOffset: 50);
        toast.setDuration(Toast.LENGTH_SHORT);
        LayoutInflater inflater = getLayoutInflater();
        View layout = inflater.inflate(R.layout.custom_toast, (ViewGroup) findViewById(R.id.custom_toast_root));
        toast.setView(layout);
        toast.show();
}

1 usage
private void showListDialog(){
        final String[] list = {"Message1", "Message2", "Message3", "Message4", "Message5"};
        AlertDialog.Builder dialog_list = new AlertDialog.Builder( context: MainActivity.this);
        dialog_list.setTitle("使用LIST呈現");
        dialog_list.setItems(list, new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialoginterface, int i) {
                        Toast.makeText( context: MainActivity.this,  text: "你選得是" + list[i], Toast.LENGTH_SHORT).sho
                }
        });
        dialog_list.show();
}
```

```kotlin
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val btn = findViewById<Button>(R.id.button)
        btn.setOnClickListener { it: View!
            val dialog = AlertDialog.Builder( context: this@MainActivity)
            dialog.setTitle("請選擇功能")
            dialog.setMessage("根據下方按鈕選擇要顯示的物件")
            dialog.setNeutralButton(
                text: "取消"
            ) { dialogInterface, i ->
                Toast.makeText(
                    context: this@MainActivity,
                    text: "Dialog關閉",
                    Toast.LENGTH_SHORT
                ).show()
            }
            dialog.setNegativeButton(
                text: "自定義Toast"
            ) { dialogInterface, i -> showToast() }
            dialog.setPositiveButton(
                text: "顯示List"
            ) { dialogInterface, i -> showListDialog() }
            dialog.show()
        }
    }

    private fun showToast() {
        val toast = Toast( context: this@MainActivity)
        toast.setGravity(Gravity.TOP,  xOffset: 0,  yOffset: 50)
        toast.duration = Toast.LENGTH_SHORT
        val inflater = layoutInflater
        val layout: View = inflater.inflate(
            R.layout.custom_toast,
            findViewById<View>(R.id.custom_toast_root) as ViewGroup?
        )
        toast.setView(layout)
        toast.show()
    }

    private fun showListDialog() {
        val list = arrayOf("Message1", "Message2", "Message3", "Message4", "Message5")
        val dialog_list = AlertDialog.Builder( context: this@MainActivity)
        dialog_list.setTitle("使用LIST呈現")
        dialog_list.setItems(
            list
        ) { dialoginterface, i ->
            Toast.makeText(
                context: this@MainActivity,
                text: "你選得是" + list[i],
                Toast.LENGTH_SHORT
            ).show()
        }
        dialog_list.show()
    }
}
```

可以看到在呼叫提示視窗的時候，JAVA 與 KOTLIN 有較大的差異，而且我認為 KOTLIN 比 JAVA 更好理解程式的目的且簡潔許多。

心得:

除了在最後幾的練習的矩正有點問題，剩下的幾乎都可以理解是怎麼運作的。