Summer 2019 / Android Class Remote Learning Assignment - Week 4

Answer ​Q1​ to ​Q5​ in any text file. Upload the Android project includes ​Q6​ and the text file (​Q1 to ​Q5​) to your GitHub and ​send a direct message to us that you have completed.

//在任何文本文件中回答 Q1 到 Q5。 將包含 Q6 和文本文件（Q1 到 Q5）的 Android 項目上傳到您的 GitHub，並向我們發送您已完成的直接消息。

* 1. **What is ​Polymorphism​? Try to explain in Mandarin.**

//Polymorphism是什麼？ 嘗試用一般方式解釋

Polymorphism 中文意思為「多型」，物件能夠依照不同情形變換資料型態，以利減少開發者反覆建立類似概念的物件。例如遊戲中，角色/職業皆有「跑」、「跳」、「攻擊」模組，透過 Polymorphism 概念即可快速針對不同角色/職業調整其配套模組，而不用為了建立新角色而重頭搭建模組。

* 1. **Here are the 7 important lifecycle methods of an ​Activity​:**

//以下是lifecycle的七個重要 methods ，請嘗試解釋Activity在生命週期哪些時候會使用到。

* 1. onCreate()

當android系統第一次開啟應用程式時，系統會透過onCreate()初始化應用程式。

* 1. onStart()

當系統已經開啟應用程式、或者使用者從其他頁面回到應用程式時，onStart()準備讓使用者能看到應用程式前台介面，並準備應用程式使其成為可操作形式。

* 1. onResume()

當應用程式進入onResume()階段，使用者已經可以透過前台操作應用程式。

* 1. onPause()

onPause是離開應用程式的首要指令，代表前台已經離開此應用程式。

* 1. onStop()

當使用者離開原應用程式，開始操作其他應用程式，即進入onStop()指令。

* 1. onDestroy()

當應用程式被完全關閉時，就會使用onDestroy()，原因可包含(1)使用者自主關閉軟體、(2)軟體操作完畢、(3)其他系統情形，包含旋轉手機、多視窗模式等使系統暫時關係軟體。

* 1. onRestart()

當應用程式在Stop階段，而使用者要重新回到原應用程式時，系統使用onRestart()。

* 1. **What is the ​Android Jetpack​?**

//Android Jetpack是什麼？

Google於2018年發佈全新的Android Jetpack，此全新套件包含大量開發Android所需的Library，包含 Navigation 、 Lifecycle mangement ，未來也將持續新增套件內容，使開發者能快速透過模組開發應用程式。

* 1. **Key concepts in ​Android Development​: Try to research them and explain briefly.**

//Android 開發的關鍵概念？嘗試研究並且簡要說明

以集訓前的課程來看，認為android 的關鍵概念包含：

* 1. Kotlin：Kotlin 是由 JetBrains 團對所開發，不僅可以與 JAVA 程式碼相互運作，且向各個程式語言學習各優點，節省許多撰寫程式碼的時間，使開發者更專注研發軟體，許多開發者都說透過 Kotlin 重拾撰寫軟體的熱情，也代表著開發 Android 時熟悉 Kotlin 會是未來的必要關鍵。
  2. SDK / Jetpack / Library ：Android Studio 可以透過相關套件加速軟體開發，若能熟悉相關工具能更節省時間、有效開發。
  3. Layout / XML / Navigation ：透過 XML 語法或者可視介面能快速架設 Layout 等 UI ，其中又陸續提供 ConstraintLayout 等製作概念，使開發能更適應不同硬體設備；而操作各界面的切換時，Android Studio更提供 Navigation 讓開發者能流暢建立各 Layout 之間的連結。
  4. 不斷的學習/進化( Room / ViewModel / Navigation / LiveData … )：為能更加提升開發者及 android 的工作效率，Google 不斷地推出更多的開發工具及概念，也代表除了上述的關鍵概念以外，開發者需不斷地吸收新知才能跟上未來開發所需。

* 1. **What is ​Coroutines​? Why we use it? Try to explain in Mandarin**

//Coroutines 是什麼？ 為什麼要使用他？嘗試用普通話解釋

Coroutines 能視為「呼叫可以中斷及繼續執行的函式」，有時一個函數包含大量的執行程序，為避免執行造成應用程式的延緩，或期待逐步執行及輸出處理結果，即可使用 Couroutines 。