

國立虎尾科技大學

機械設計工程系

計算機程式 bg2 期末報告

PyQt5 事件導向計算器

PyQt5 Event-Driven Calculator Project

學生：

設計一乙 40623204 張晏晴

設計一乙 40623206 邱妍蓁

設計一乙 40623214 吳家文

設計一乙 40623215 吳柏毅

設計一乙 40623249 李秉樺

設計一乙 40623251 鄧靖宣

指導教授：嚴家銘

2017.12.1

目錄

第一章

前言	3
----	---

第二章

可攜程式系統介紹	4
----------	---

2.1 啟動與關閉	4
-----------	---

第三章

Python 程式語法	5
-------------	---

3.1 變數命名	5
----------	---

3.2 print 函式	5
--------------	---

3.3 重複迴圈	6
----------	---

3.4 判斷式	6
---------	---

3.5 數列	6
--------	---

第四章

PyQt5 簡介	7
----------	---

第五章

Calculator 程式	8
---------------	---

5.1 建立按鈕	9
----------	---

5.2 建立程式	9
----------	---

5.3 建立迴圈使程式簡單化	10
----------------	----

第六章

6.1 Fossil SCM	11
6.2 網誌心得	12
6.3 說明各學員任務與執行過程	13
6.4 學員心得	14~15

第七章

7.1 結論與建議	16
-----------	----

第一章

計算器程式期末報告前言容。

起初我們對程式邏輯、指令用法完全不了解，不知從何開始，可說是困難重重。

程式製作可說是最嚴謹的一門學問任何的字母大小寫/數字、符號的全形半形/空白鍵的多寡(與層級有關)/絲毫的錯誤都是不容許的。

速度慢是對所有初學者來說必經的過程，不懂得查 goole，不能執行花了大把的時間抓錯，常常都抓不出來哪裡有瑕疵導至於常常全部刪掉重做。

對於所有剛踏進這領域的人們“耐心”是最不可或缺的東西。煩躁!不只對心情也會大受影響，對正在進行的事情更容易出錯。

花得大半天的時間宅在圖書館或咖啡廳，互相討論幫忙解決問題，不小心得發現新的方法可以使用在程式上面(ex:迴圈應用)，大幅地省下時間以及步驟。

新發現是開心也是懊惱，花了很多時間探討、搜尋、解決、終於完成了一個定義及運算，但迴圈的用法省下了我們 2/3 的步驟，這情況林林總總出現了很多次。

再來更深入程式撰寫及開發請往下看我們的製作報告!

第二章可攜程式系統介紹

可攜程式系統介紹

Python 是種通用. 動態. 面向對向的編程語言。Python 的設計目的強調程序員的生產效率和代碼可讀性，Python 靈感來自於 ABC. Haskell. Java. Lisp. 圖標及 Perl 編程語言。Python 不使用大括弧和關鍵詞，他使用空格縮排劃分代碼塊。相較於 C++ 或 Java，Python 使開發者用更少的代碼表達，Python 擁有動態型別系統和垃圾回收功能，可自動管理記憶體，並支援多種編程，包括物件導向. 命令式. 函數式及程式編程

2.1 啟動與關閉 2

第三章 Python 程式語法

Python 程式語法

3.1 變數命名

變數必須以英文字母大寫或小寫或底線開頭

變其餘字元可以是英文大小寫字母、數字、底線

變數區分英文大小寫

變數不限字元長度

不可使用關鍵字當變數名稱

Python3 的程式關鍵字，使用者命名變數時，必須避開下列保留字。

```
[ "False" , " None" , " True" , " and" , " as" , " assert" , " break" ,  
  " class" , " continue" , " def" , " del" , " elif" , " else" , " except" ,  
  " finally" , " for" , " from" , " global" , " if" , " import" , " in" ,  
  " is" , " lambda" , " nonlocal" , " not" , " or" , " pass" , " raise" ,  
  " return" , " try" , " while" , " with" , " yield" ]
```

3.2 print 函式

print() 為 Python 程式語言中用來列印數值或字串的函式，其中有 sep 變數定義分隔符號，sep 內定為 "，"，end 變數則用來定義列印結尾的符號，end 內定為跳行符號。

3.3 重複迴圈-for 迴圈

用於已知重複次數的程式，迴圈結構中指定指定迴圈變數的初始值、終止值語遞增(減)值，迴圈變數將由初始值變化到終止值的前一個數字每次依照遞增(減)的值進行數值遞增或遞減。

3.4 判斷式

If self.display.text() == '0' and digitValue == 0.0:

Return

==判斷式的等於要兩個

'0' 表示字串

and 有時候用||

:還沒說完

3.5 數列

Digits = [self.one, ...\]

[]表示數列

\表示還沒講完

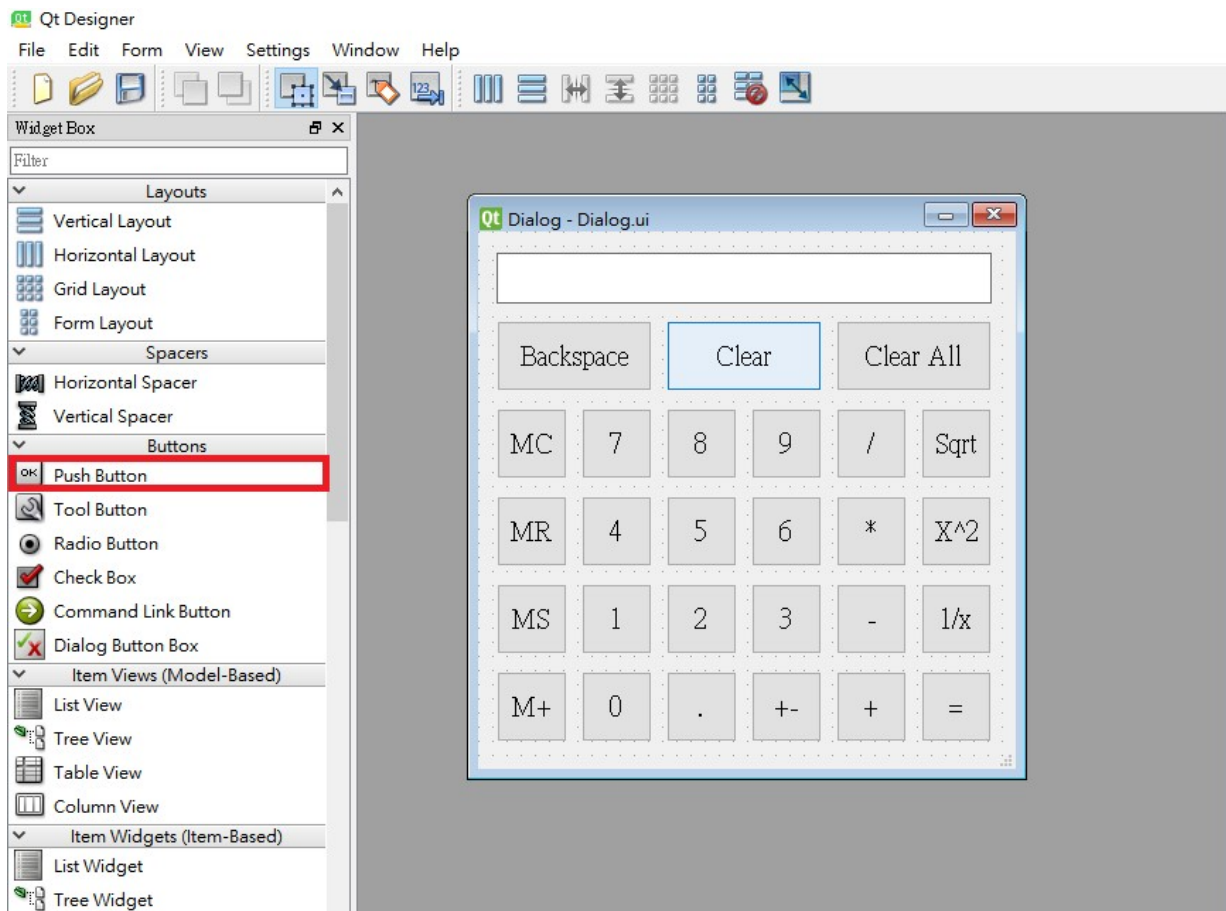
第四章 PyQt5 簡介

Pyqt5 是一套綁定 Sogia QT5 應用的框架，可用於 Python2 和 3，Pyqt5 作為 Python 的一個模塊，它有 620 多種和 6000 個函數和方法。Pyqt5 是一個跨越平台到工具包，可運行在所有主要的操作系統，包括 UNIX. MacOS。Pyqt5 是雙重許可，開發者可在 GPL 和商業許可間選擇

第五章 Calculator 程式

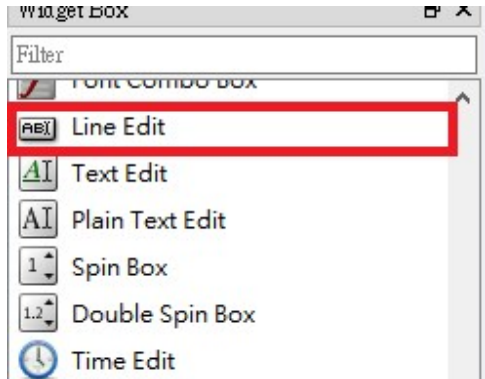
5.1 建立按鈕

從左邊工具列拉按鈕"Push Button"



並更改每一項的命名"objectName"

以及拉顯示欄"Line Edit"



更改其名為"display"

5.2 建立程式碼

導入 sys 模組→`import sys`

從 PyQt5 模組導入 QtWidgets 模組→`from PyQt5 import QtWidgets`

從 ui 目錄導入 Dialog.py, Python 程式檔案本身就是一個模組, 且文件名就是模組名→`from ui import Dialog`

定義各數字名稱→`number = [self.one, self.two, self.three, self.four, self.five, self.six, self.seven, self.eight, self.nine, self.zero]`

以及符號名稱→`plus_minus = [self.plusButton, self.minusButton]`

`times_division = [self.timesButton, self.divisionButton]`

```
unary = [self.reciprocalButton, self.squareRootButton,  
self.powerButton]
```

5.3 建立迴圈使程式簡單化

```
39  
40 - for i in number :  
41     ..... i.clicked.connect (self.digitClicked)  
42  
43 - for i in plus_minus:  
44     ..... i.clicked.connect (self.additiveOperatorClicked)  
45  
46 - for i in times_division:  
47     ..... i.clicked.connect(self.multiplicativeOperatorClicked)  
48  
49 - for i in unary :  
50     ..... i.clicked.connect(self.unaryOperatorClicked)  
51
```

其餘建立各個 Button 的函式

第六章心得

期末報告心得

6.1 Fossil SCM

40623204-

剛開始真的什麼都不會，都不知道上課要做什麼，經過努力學習，大概知道怎麼使用，經過長期使用倉儲，可以透過 timeline 知道自己每次什麼時候上傳資料。

40623206-

從無到有用幾個簡單的 fossil 指令建倉儲，利用近端遠端維護網誌，使得我更了解 fossil 的運用

40623214-

Fossil SCM 的倉儲很方便，裡面可以儲存連結及網誌，又有 timeline 可以查詢我們上傳檔案的時間及頻率，讓其他人知道你做檔案的次數，都寫得一清二楚

40623215-

Fossil 倉儲真的很方便，有 Timeline 可以清楚記錄自己在幹什麼，雖然一開始很難，不知道在幹嘛，沒有進入狀況，但現在總算是熟練多了，希望以後不論是作業、活動紀錄... 等能夠善用這個方便的功能。

40623249-

Fossilscm 很想一個網頁版的隨身碟，也可以隨時得看到我過去到底做了些甚麼東西。也很像小日記的感覺！

40623251-

了解用法與其語法的意義，並配合 lieo 來管理倉儲與網誌。

6.2 網誌心得

40623204-

可以記錄每週學到什麼東西、有什麼學習心得，有需要時也可以打開來看，迅速知道自己之前學到什麼、做了什麼。

40623206-

將上課所寫的筆記整理完後，以最簡潔的方式放到網誌上，記錄每週所學東西，雖然不是很熟悉，但大致上了解得差不

40623214-

Fossil SCM 的網誌很方便，裡面可以儲存連結及網誌，又有 timeline 可以查詢我們上傳檔案的時間及頻率，讓其他人知道你做檔案的次數，都寫得一清二楚

40623215-

Fossil 網誌真的很方便，有 Timeline 可以清楚記錄自己在幹什麼，雖然一開始很難，不知道在幹嘛，沒有進入狀況，但現在總算是熟練多了，希望以後不論是作業、活動紀錄... 等能夠善用這個方便的功能。

40623249-

Fossil 這個網誌可以記錄我每個禮拜上課的筆記，心得。

養成一個紀錄的習慣，這跟一個備忘錄得太念是一樣的。

40623251-

透過先前的 fossil 與 lieo 來達到網誌的管理，跳脫以前從網頁更新網誌的觀念與做法，並透過網誌來紀錄學習歷程。

Github 協同倉儲

bg2 : https://github.com/40623206/bg2_pyqt5_calculator

6.3 說明各學員任務與執行過程

40623204-

協同做計算機，剛開始以為很難，不知道有哪些可以我可以做到，後來努力去理解老師寫的東西，有大概知道怎麼做，但中間做小數點還是遇到很多問題，但慢慢推，就把遇到的問題解決了。

40623206-

數字、數字疊加、+、-、=、計算、中斷運算、clearAll 因為 waitingForOperand 的 waiting 少了 ing，所以一直無法順利執行+ 跟-的部分用了 For 迴圈及各自跟下面做連結，導致+按下去*2，-按下去*0。

40623214-

分組做了計算機程式，我負責 clearmemory 和正負號，做正負號的過程中遇到比較多問題，一開始不懂那些指令是什麼意思，後來才知道是說假設變數大於0，按下正負號鍵會顯示成正號，當函數小於0，按下正負號會顯示除了第一個字元以外的數字。

40623215-

在做根號和退格(Backspace)的函式過程還算順暢，可以快速理解，但是在正負號函式的[1:]搞不懂是什麼，由退格函式的[:-1], 猜想[1:]的意義應該是"保留除了第一個字元以外的字串"，總之討論出來很有成就感！

40623249-

每次做完測試的時候，總會跳出一個對話框說錯誤。我一直以為都是我得問題。就不停地反覆尋找，或是刪掉重做。差不多反覆個5到6次吧，就聽到有人在講說常常你在做要測試的石鈎都會說錯誤，但不一定是自己的問題可能是前面同學做的可能有做錯誤導致自己的程式無法順利的運行。

40623251-

一開始以為 memory 系列的計算機按鍵的程式碼會很複雜，會牽涉很多其他的程式碼，像是+*/=-之類的，後來發現其實很簡單，變成比較多的時間在處理程式碼之間的衝突與問題，因為如此，漸漸的了解計算機的程式編寫與其程式碼之間的運行。學員心得

40623204-

協同做計算機，剛開始以為很難，不知道有哪些可以我可以做到，後來努力去理解老師寫的東西，有大概知道怎麼做，但中間做小數點還是遇到很多問題，但慢慢推，就把遇到的問題解決了。

40623206

從一開始的 eric6 建立 Dialog 拉按鈕及顯示幕到使用物件導向編寫程式，用運老師所教的，利用 For 迴圈連結同屬性的東西，才不會使程式複雜化。

40623214-

整個過程中，感覺有認識到一些字元和指令，但還是有很多地方看不懂，應該要多上網看影片或查詢意思，並且多做筆記來加強。

40623215-

多人協同做計算機，一開始覺得很難不知該從何下手，但是時間一到，大家都在時間內做到自己該做的事，雖然小組報告協同到一半出了差錯，但是換了方式還是解決了問題，小組合作起來真的是比隻身一人強太多了！

40623249-

這件事情真的需要大家的同心協力一起去完成。從剛凱是自己做自己的沒有溝通導致製作*/時一直無法執行，沒有意識到說是別人負責的部分沒有弄好進而牽連到我的部分無法執行。

40623251-

先從 fossil 來作為計算機程式的入門，等熟悉其系統後再銜接到更進階卻很相似的 github，透過 github 與 eric6 來完成計算機的協同編寫，並透過 lieo 管理網誌來紀錄學習過程

第七章結論

7.1 結論與建議

經過許多得波折，終於完成簡易版的計算機了。

大家分工合作，省下大把大把的時間，有人校閱/錯誤、有人做+、有人做=、等……，運用手邊資源查詢/參考文本。

多錯幾次速度就會變得很快，而且常常重做總能發現很多意外的小驚喜，因為做越多就越知道裡面程式的含意，想表達的東西。

雖然對老師來說沒有甚麼，但對來說慢慢能看得懂 PYQT5 的成就感不容小覷，這些成就感洪流讓我們對程式的興趣正在一點一滴的萌芽，但是多次的失敗都曾經讓我們心情盪到谷底。

失敗後的成功學到的東西總是最多的，沒有失敗的襯托那成功有何意義！

對於之後的課程以及發展對程式這塊迅速學習的方法，邊做邊學習緊接著連續多次的失敗是不二法門。

細心!細心!!再細心!!!不是需要而是必要，1 就是 1，全形就是全形，電腦絕不妥协。

對這些程式新手來說正確的使用比精準使用都來的重，先會走還要走的穩才可以在想該如何跑！

