國立虎尾科技大學

機械設計工程系

計算機程式 bg6 期末報告

PyQt5 事件導向計算器

PyQt5 Event-Driven Calculator Project

學生：

設計一乙 40623225 卓昆峰

設計一乙 40623226 鄭清詮

設計一乙 40623227 張耀元

設計一乙 40623234 洪一木

設計一乙 40623235 黃昱誠

設計一乙 40623236 黃子峰

指導教授：嚴家銘

2017.12.18

摘要

本研究的重點在於每組寫一個計算機程式，並分配每組每個組員需要負責的按鍵，然後依照老師所給的範例寫入各按鍵的程式，這並非只是抄襲，而是在寫入每行程式之後，並且去理解每行的程式運作及方法，然後將遇到的問題組別一起討論，並將所遇到的問題解決，讓自己本身以及自己組員對計算機程式有更深的了解。

- 我們使用 eric6 來做計算機程式
- 使用 github 來做協同

目錄

表目錄

圖目錄

第一章 前言

前言

為了讓我們知道以後協同的慨念和整體運作的方法所以使用 MIKTex 做報告把我們一學期學到的做出整合分配協同出一個可用的計算機

機械設計就是靈活運用六種表達, 明確說明如何透過固體、流體與軟體元件之互動運作, 而能達成預定結果之明確與具體表達.

設計是一種明確與具體的表達, 而且是在仔細思考、多方考量後所完成的表達, 表達具有六種形式, 包括口語、文字、2D、3D、數學與實體表達, 設計的結果可以讓執行者有所依循, 根據指示執行後, 可得預期之結果.

機械是一種器物, 而且是由固體、流體與軟體元件精巧組合而成, 可互動運作, 達成特定功能之器物

三種類型的電腦與相關交互輔助設計模式:

近端 (local) - 工程師面前的電腦, 以及機械設計系內部網路上連線的電腦.

遠端 (cloud) - 廣域網路上的電腦.

可攜系統 (mobile) - 工程師可以隨身啟動, 在各近端與遠端間移動, 仍能保有客製化的設定環境.

第二章 可攜程式系統介紹

可攜程式系統介紹

2.1 啟動與關閉

start.bat 會開啟 cover_and_abstract cmd exe

可以從 cover_and_abstract 中修改 start.bat 讓他開啟時可以一起開啟 leo

從 cover_and_abstract 中修改 leo 讓他在 start.bat 開啟時可以找到要開的檔案

stop.bat 的作用關閉全部的檔案

2.2 可攜式系統介紹

可攜式系統的意義: 為了能讓自己隨身隨地完成自己的工作和建立自己習慣的開發方式

miktex: 讓文字檔跟圖檔轉成 PDF 檔

GIMP2: 修剪圖片裁切圖片

DiaPortable: 可繪製圖形註解圖片

github: 組態管理的一種能讓多人協同優點公開不用錢

Python36: 在不同電腦能進行 Python 程式開發

Share-X: 錄製影片截圖

第三章 Calculator 程式

Calculator 程式細部說明

3.1 計算機程式內容

for i in num_button: i.clicked.connect(self.digitClicked)

迴圈數字鍵點案的時候連結到 digitClicked

self.display.setText("0")

起始 display 為 0

class Dialog(QDialog, Ui_Dialog)

Dialog 類別同時繼承 QDialog 與 Ui_Dialog 類別

self.equalButton.clicked.connect(self.equalClicked)

當你按到 equalButton 連結到 equalClicked 方法

self.waitingForOperand = True

起始時, 等待使用者輸入運算數值變數為真

def digitClicked(self)

使用者點擊按鈕時送出的按鈕指標類別, 在此利用此按鍵類別建立案例

clickedButton = self.sender()

clickedButton 即為當下使用者所按下的按鈕物件

```
if self.pendingAdditiveOperator: if not self.calculate(operand, self.pendingAdditiveOperator):
        self.abortOperation()

        return
```

```
        self.display.setText(str(self.sumSoFar))
    else:
    self.pendingAdditiveOperator = clickedOperator
    self.waitingForOperand = True
        顯示目前的運算結果
        假如 self.pendingAdditiveOperator 為 False, 則將運算數與 self.fumSoFar 對應
        self.sumSoFar = operand
    能夠重複按下加或減, 以目前的運算數值執行重複運算
    進入等待另外一個運算數值的階段, 設為 True 才會清空 LineEdit
if self.waitingForOperand: return
    self.display.setText('0')
    self.waitingForOperand = True
在等待運算數階段, 直接跳出 slot, 不會清除顯示幕
清除顯示幕後, 重置等待運算數狀態變數
```

第四章 Python 程式語法

Python 程式語法

4.1 變數命名

Python3 變數命名規則與關鍵字 Python 英文變數命名規格

變數必須以英文字母大寫或小寫或底線開頭變數其餘字元可以是英文大小寫字母, 數字或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作變數名稱

Python3 的程式關鍵字, 使用者命名變數時, 必須避開下列保留字.

Python keywords: [‘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’]

使用有意義且適當長度的變數名稱，例如: 使用 length 代表長度, 不要單獨使用 l 或 L, 也不要使用 this_is_the_length 程式前後變數命名方式盡量一致, 例如: 使用 rect_length 或 RectLength 用底線開頭的變數通常具有特殊意義

4.2 print 函式

—def hi (): print(hi)

4.3 重複迴圈

—for i in num_button: i.clicked.connect(self.digitClicked)

4.4 判斷式

—if self.pendingMultiplicativeOperator: if not self.calculate(operand, self.pendingMultiplicativeOpe
self.abortOperation() return

4.5 數列

—num_button = [self.one, self.two,
self.three, self.four, self.five, self.six, self.seven, self.eight, self.nine, self.zero]

第五章 PyQt5

5.1 PyQt5 架構

精巧的電子零件，是透過輸入 010101，稱為「機器碼 (Machine code)」的指令才能運作的。電子裝置可以直接辨識與執行機器碼的指令，而且不同的裝置使用的機器碼語言是不一樣的。

想當然爾，人類不可能一直查詢機器碼，再輸入給電子零件，因此才產生「程式 (Program)」。

組合語言 (Combination language) 是一種用於可編成組件的程式語言，一種組合語言專用於某種電腦的系統結構。換句話說，組合語言僅次於機器碼，可以以較簡單的方式命令，而且開發人員可以輕易辨別位址與數值

C 語言透過「編譯 (Compile)」這個流程，將程式碼轉為機器碼執行。不同作業系統存在相應的「編譯器 (Compiler)」，對應到硬體的機器碼執行，而某些編譯器正是組合語言撰寫的。

Python 語言是透過 C 語言編寫的直譯器進行轉譯與執行，因此不須經過編譯動作，甚至可以藉由伺服器在網路端執行，將結果回傳給客戶端。

低階語言 (C) 有以下特性：

處理範圍廣，而且可以直接影響運作效能。較不易閱讀與開發，一件工作必須寫得較為詳細。

高階語言 (Python) 則反之：

處理範圍和運作效能端看基礎架構而定，大量運算時效能不如低階語言。簡單且優化過的內建命令，往往會有簡單的表示法與程式庫、模組。

PyQt 幾乎支援 Qt 大部分的功能，並且將較專門的功能另外分成 PyQt Chart（2D 圖表）、PyQt Data Visualization（3D 圖表）、PyQt Purchasing（應用程式購買功能）。

另外 QScintilla 是一個將 Scintilla 連結至 PyQt 的套件（在 C++ 可以直接用 Qt 和

Scintilla 即可），用途是辨識文字中的程式語言，以亮顯 (highlight) 的方式呈現，可以用作程式語言的辨識功能。

導入 sys 模組

import sys

導入 keyword 模組

import keyword

print() 函式用法

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, s ep 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.

簡單的 for 迴圈範例

for i in range(10): print(i)

函式用法與呼叫使用者可以利用下列程式, 練習 def 函式定義與呼叫的用法.

定義函式

def square_of_x(x): return x*x

呼叫函式

y = square_of_x(3) print(y)

列印 y 對應內容

第六章 心得

期末報告心得

6.1 Fossil SCM

40623236: 組成內容與狀態的配置, 因此軟體組態管理就是針對軟體開發過程, 有關組成內容與狀態配置的管理

40623235:

40623234:

40623227:

40623226:

40623225:

6.2 網誌心得

40623236: 更新網誌是一件很麻煩的事但是在打的過程中可以回復今天上的東西以後旺季的時候也可以回去觀看複習走過必留下痕跡在網誌中遇到的問題難保以後不會遇到這樣多去看已定可以更加進步

40623235:

40623234:

40623227:

40623226:

40623225:

6.3 Github 協同倉儲

bg6 協同倉儲:https://github.com/40623234-1/bg6_pyqt5_calculator

40623236: 在推東西的時後要先記得 pull 雖然協同很方便但是溝通是不可或缺的

40623235:

40623234:

40623227:

40623226:

40623225:

6.4 學員心得

40623236: 協同是一個很重要的東西一個東西完成需要很多人的幫忙和修改

40623235:

40623234: 在一開始時，我剛進這間學校後發現有計算機程式這堂課，那一瞬間想到高職時尚的計算機概論，差不多是了解電腦程式和灌灌程式，後來知道其實 fossil 與 github 的方便性與快速性，了解為何世界的科技能進步如此快速。

40623227: 因為是初學者，所以一個人製作是非常困難的，因此需要組員間的討論才能完成

40623226:

40623225: 本來從一開始甚麼東西都是自己來，到現在變成一個團隊憶起分工合作，一起將一個計算機做出來別新奇

說明各學員任務與執行過程

40623236: 數字鍵等號計算清除等號跟計算最麻煩因為會影響到很多人當有問題時我都要去幫忙

40623235:

40623234:

40623227:+-號報告因為報告要集結大家意見，所以在寫的過程中幫忙我的其實很多

40623226:

40623225: 任務分配是乘跟除，需要去了解各個字所代表的意思，覺得這個部分特別的難

第七章 結論

期末報告結論

7.1 結論與建議

要寫一個計算機比想像中的困難，程式的邏輯以及運用方法，都是非常的困難，但在製作過程中，有大家的參與，組員間的討論溝通，解決遇到的問題，這都讓在寫計算機的時候變得簡單許多，能夠完成自己組別的一個計算機。建議: 需要隨時注意完成度

第八章 參考文獻