

高一專題統整

郭勝威

2022 年 7 月 29 日

目錄

1	序言	1
2	爬蟲	2
2.1	實作流程	2
2.2	抓取解題網站的資料	3
2.3	自動機器人	5
2.4	密碼破解	8
2.5	校內廣播	10
2.6	排名爬取	12
2.7	學測單字爬取 (一整年的最終成果)	14
2.7.1	序言	14
2.7.2	實作動機	14
2.7.3	遇到的困難和解決方式	15
2.7.4	學到的事情	16
2.7.5	可以改進的方向	16
3	遊戲	17
3.1	序言	17
3.2	遊戲基本流程	17
3.3	貪吃蛇 (c++)	18
3.4	貪吃蛇 (python)	20
4	未來展望	21
5	結語	22

1 序言

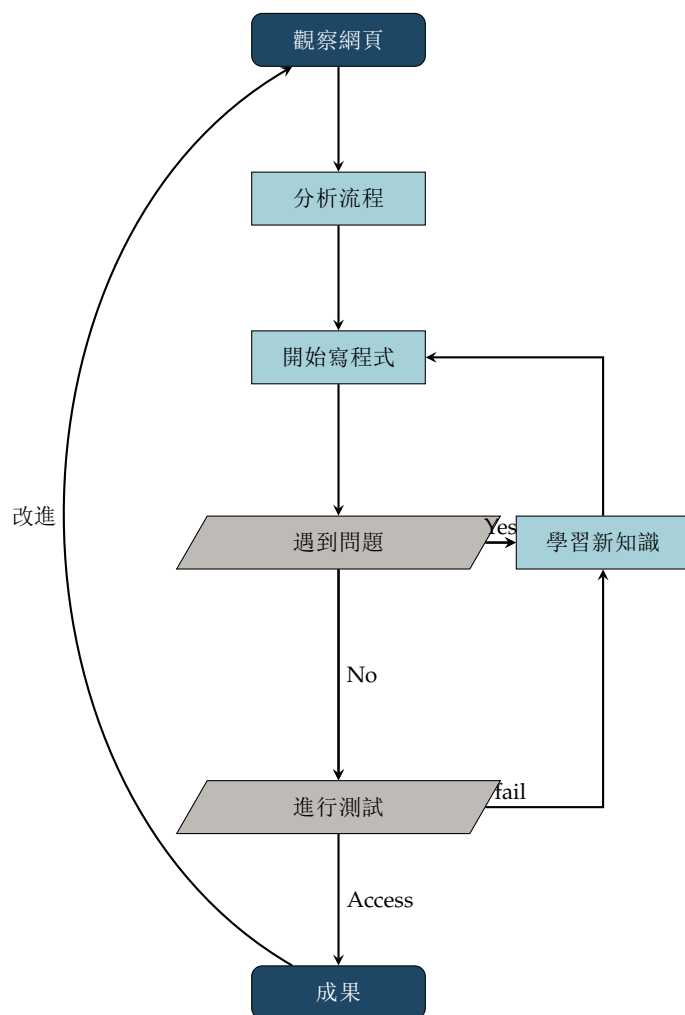
高一轉瞬即逝，當我回顧一年來的收穫，不知不覺中就做了許多小專案，每一個專案都是心血來潮才開始的，實作時，總會遇到許多問題，不過我會投入許多的時間去思考不同的解決方案，抑或從網路上尋找解答，我相信這一點一滴了累積，即使不起眼，但仍代表我的進步

我把這一年所做的專案分成六個部分，大部分都是由`python`實作的，網頁則是用`html`, `css`, `javascript`實作，爬蟲的大部分都是為了讓自己可以輕鬆一點，其他的地方則是可以因為自己的好奇心而去研究的，每一個專案都至少花了**10hr** 以上，不過卻都沒有很完整，都還有許多空間可以改進，不過許多都因為自己的能力未到而尚須時間，就先把這些託付給未來的我了。

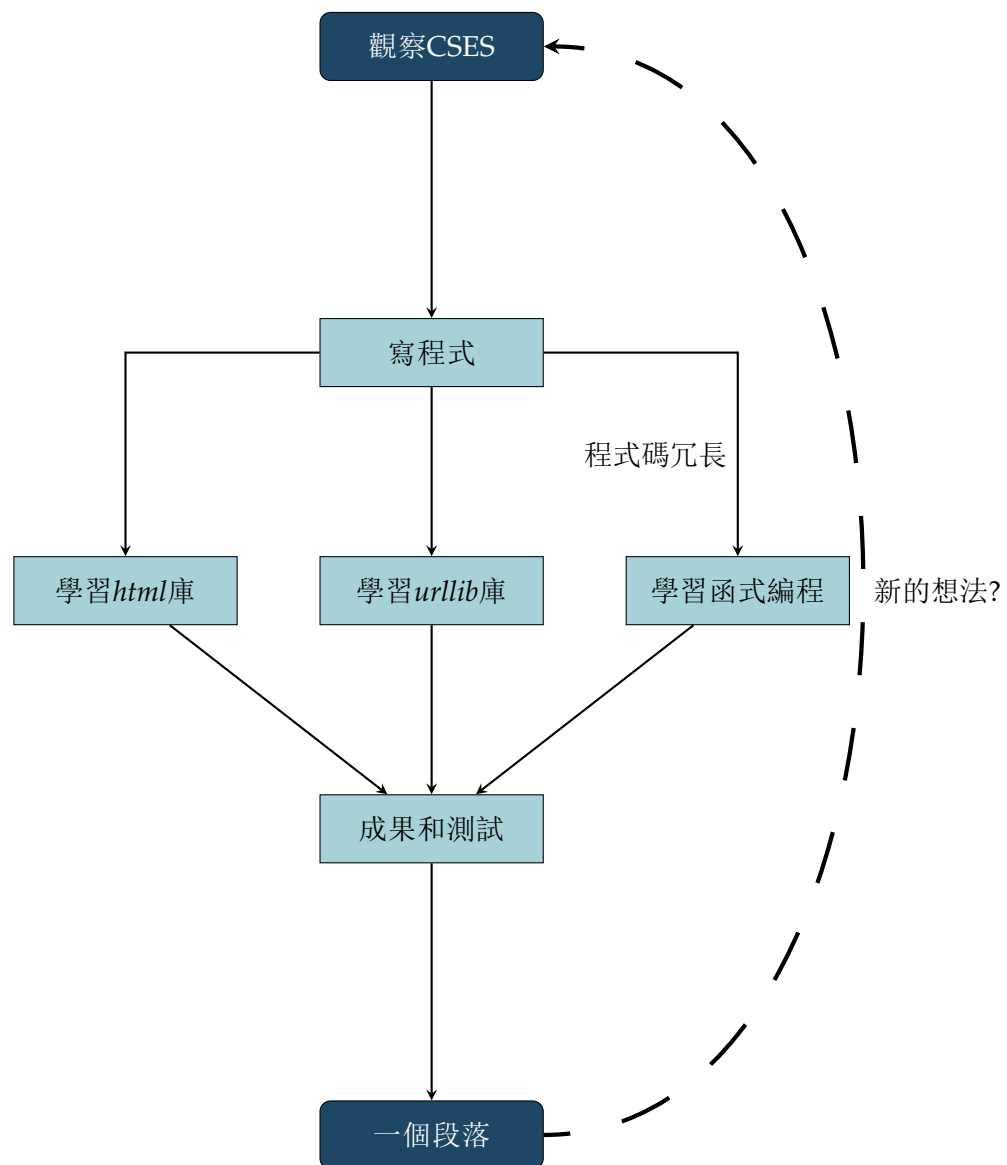
2 爬蟲

2.1 實作流程

實作流程都相差無幾，可以使用下面這張圖來代表



過程



實作動機

因為我有擔任程式設計的教學，所以常常需要出題，而解題網站 [CSES](#) 有許多很好的題目，所以我有時會把裡面的題目翻譯然後搬到 [DDJ](#)上，但是CSES裡面的測資是.txt的，還要一個一個下載下來，而DDJ要求的是.in, .out檔，所以我決定寫個程式把檔案下載下來然後改成.in, .out檔

遇到的困難和解決方式

一開始的時候，因為才剛開始寫python，所以程式碼十分冗長，我自己也看不太懂，所以之後就開始嘗試把許多東西變成函式，意外的發現挺方便的。

學到的事情

1. 開始嘗試用python拼出想要的功能
2. 學會使用urllib庫，
3. 學會使用with...open...來開啟和寫入檔案
4. 如何觀察html文檔

改進空間

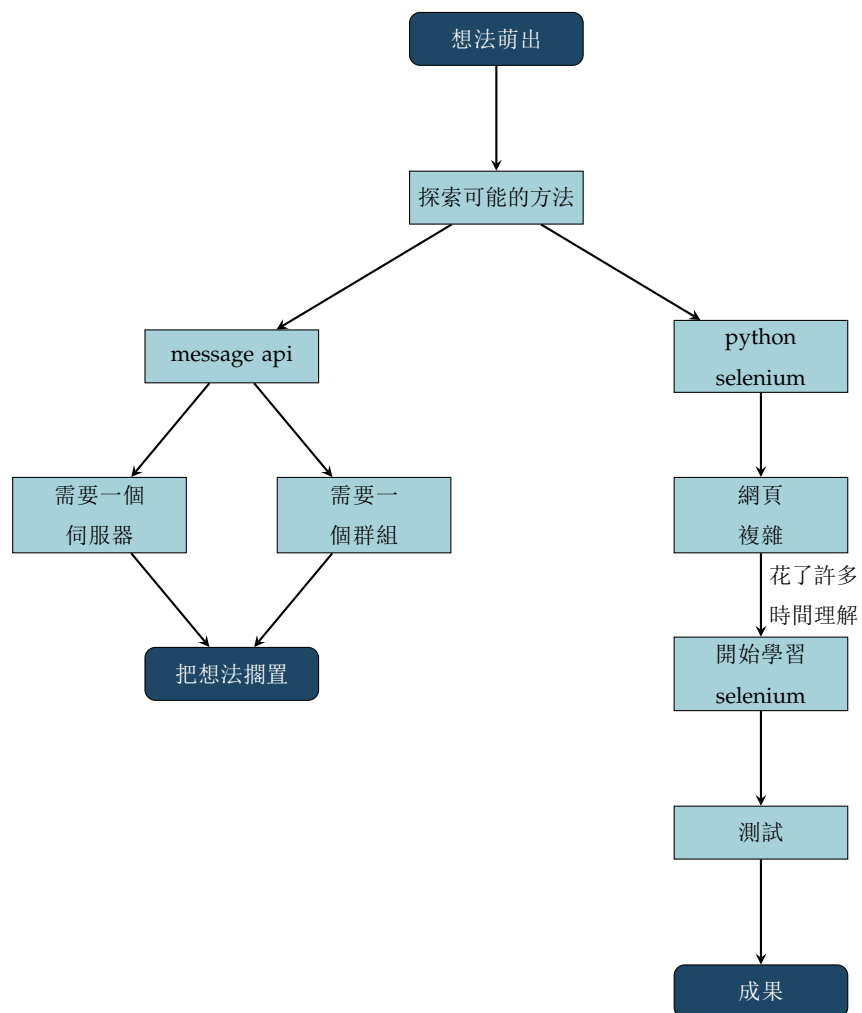
1. 因為我實作登陸是用cookie，每一次都需要複製網站上的cookie，所以可以使用session之類的庫把他改成輸入帳號跟密碼就可以登錄。
2. 通用的改進: 我想要把他弄出圖形化介面

心得

這一次實作的功能十分簡單，只有使用python爬取檔案，然後寫入而已。不過仍花了我許多時間，

2.3 自動機器人

過程



實作動機

之前因為疫情所以開始了線上上課，但是每一次上課的時候都須找尋找 google meet 的連結，十分的不方便，因此我就萌生了一個想法: 可不可以在每節下課的時候發出下一節上課的meet，讓他自動把連結送到聊天室，這樣除了我自己可以方便的進教室外，其他同學也一樣可以方便的使用。

遇到的困難

message api:

1. 實作message api的時候發現要伺服器 \implies 使用[heroku](#)
2. 需要一個群組才能開始傳送訊息 \implies 這時候我決定換一個做法

selenium庫:

1. 網頁十分複雜，不知道哪一個地方代表按鈕 \implies 研究了很久之後發現他們是直接
用div來表示按鈕

學到的事情

1. 學會了如何分析更複雜的網頁
2. 學會了使用selenium庫

心得

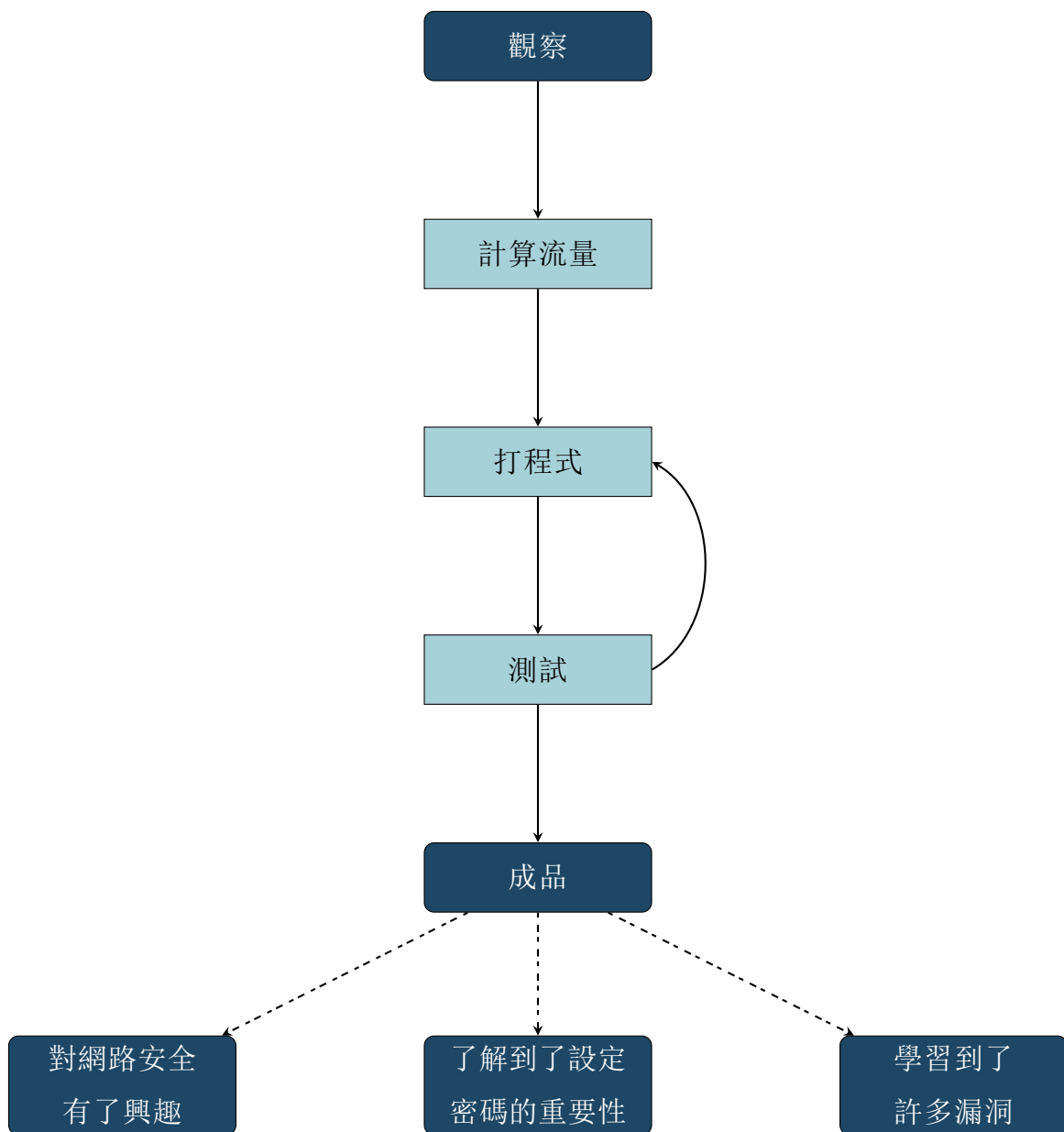
這個專案是我第一次研究大型公司的網頁，我發現他們的網頁都十分的複雜，無論是元素的命名，還是他作用的方式，都大幅的讓我了解到自己的不足，對網頁的了解還只是皮毛，更加深了我對網頁的興趣。

改進

1. 現在的程式仍需要每天清自去點開來執行，有些許的不方便
2. 在修改課程的時候有寫麻煩，在使用的最後幾天因為調課頻繁的原因出了許多差錯，應該可以在多做一個程式來專門修改課程
3. 圖形化介面

2.4 密碼破解

過程



實作動機

在學了許多的知識之後，有時候總會冒出一些奇妙的想法，在學校的時後，我發現有些幹部他們登陸時候的密碼十分的簡單，許多都只有四位數，於是我就萌生了一個想法，是不是可以透過程式來破解密碼，稍微計算了一下，如果每秒鐘請求十次的話，對於四位數的密碼共需要請求約1000秒，也就是16分鐘，是在可以接受的範圍，於是我就開始打程式了

遇到的困難和解決方式

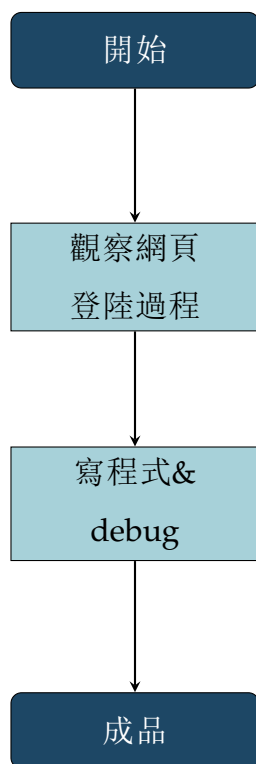
這次因為只是簡單的嘗試所以沒有遇到什麼困難，不過為了確保不會讓學校網站當掉所以有壓低了請求速率。

學到的事情

在實作過後我學到了密碼一定要好好設，如果只有數字而且只有四位數的話，雖然看起來很難被猜到，不過對於電腦來說只是幾分鐘的事情，同時也因為了這次嘗試，我開始對網路安全有了些許興趣，也有開始去學習一些漏洞向觀的知識(譬如: xss, sql注入..., ddos)

2.5 校內廣播

過程



實作動機

因為我是資訊股長，所以要負責看廣播，但有時候會漏看許多的廣播，這時候腦袋內又出現了一個想法，是不是可以透過python來進行廣播的確認。

遇到的困難和解決方式

一開始我一直在煩惱要怎麼樣才可以解決登入的問題，在經過了一段時間的尋找之後，我發現session這個函式庫可以很好的解決我的問題。

學到的事情

1. 學會使用session庫
2. 透過觀察網頁了解到學校登陸功能的實作 \Rightarrow 透過承載資訊直接傳給網頁，

意外的發現

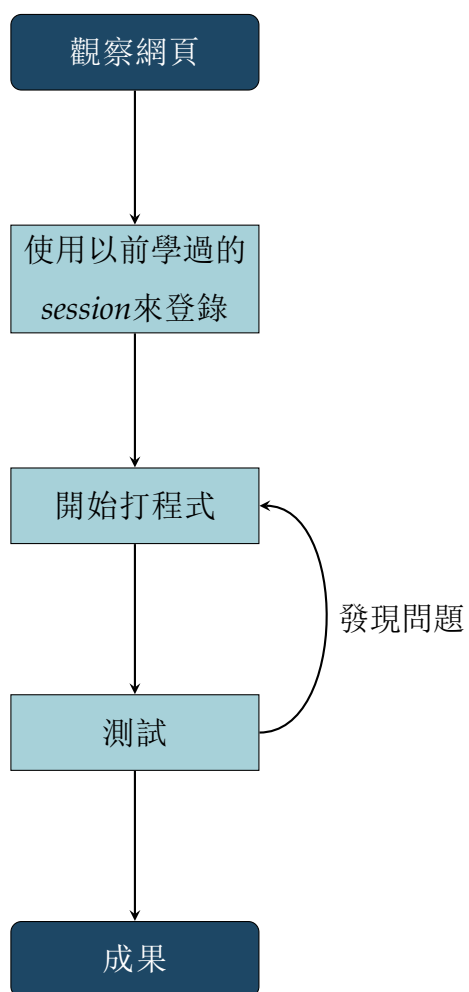
在實作的過程中，使用控制台觀察資料傳遞的過程的時候，發現學校網頁在傳遞的過程中沒有把承載的資料進行收回，雖然不知道會不會造成安全性的問題，不過也是有趣的收穫。

改進的方向

1. 希望可以在把資訊爬取下來之後把得到的資訊整理過後再重新放到網頁上面讓班級同學查看
2. 透過增加一些程式讓使用程式的時候可以更加方便的更改想要資訊，目前是希望透過一個setting.json來進行實作，
3. 圖形化介面

2.6 排名爬取

過程



實作動機

由於我是進階助教，所以每次考試之後都需要統計他們的分數，每一次都要逐一的去登記，由於我覺得他十分麻煩，不如直接寫程式把他們的成績爬取下來。

遇到的困難和解決方式

因為登陸的問題在前面已經解決了，所以這次沒有發生什麼特別的困難，只是對於改進沒有什麼想法

學到的事情

1. 對於python的實作更加的了解

可改進的方向

1. 這次實作的只有獲得總分，因為如果要獲取每一題的分數的話難度會增加很多，因為比賽時如果拿到滿分的話，顯示出來的是圖片，所以要做的事情就需要多許多。
2. 圖形化介面
3. 改善操作使用，因為現在需要把網址一個一個貼上才可以，希望以後可以直接使用比賽的命稱來直接獲得分數

2.7 學測單字爬取 (一整年的最終成果)

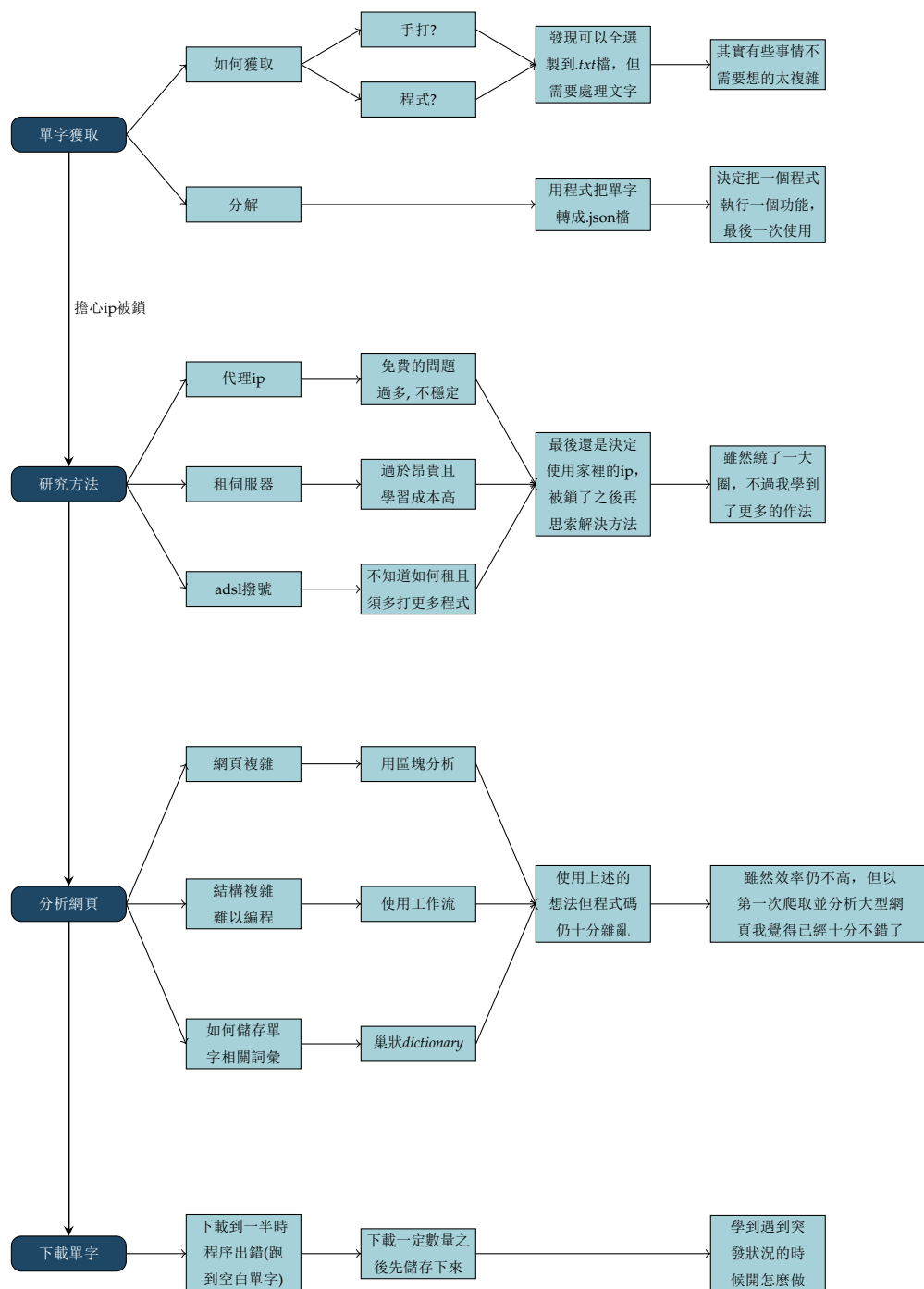
2.7.1 序言

這一個專案是我目前做過最大的一個專案，程式碼超過了500行，也是我第一次把我學到的幾乎所有東西都用上了，也學到了更多的新事物，雖然這個專案可能沒有如此的龐大，但對於我在探索程式的道路上還是有了深遠的影響。

2.7.2 實作動機

一切的起因是因為同學一句：「學測英文是不是沒有範圍阿?」，這引起了我的好奇心，所以我就很認真的去查了學測的範圍，發現其實每一科都有很明確的範圍的，我也發現到英文科其實有特別明定7000單的範圍，所以我就萌生了想法，我是不是可以先把7000單下載下來，然後透過劍橋字典，把每個單字都找出他的用法、片語、詞性、同義字....，存下來，之後就看要做網頁、應用程式、單純的執行檔，都可以，於是我就開始了盡情揮灑創意的實作了。

2.7.3 遇到的困難和解決方式



2.7.4 學到的事情

1. 透過把程式碼變成許多函式+檔案來讓自己可以快速的找到使用的函式
2. 學到了許多關於ip的事情
3. 對於中小型項目的檔案分布跟程式碼實作方式有了基本的了解
4. 開始嘗試使用許多小小的區塊，透過工作流的方式，把小小的功能集結成一個完整的項目。

2.7.5 可以改進的方向

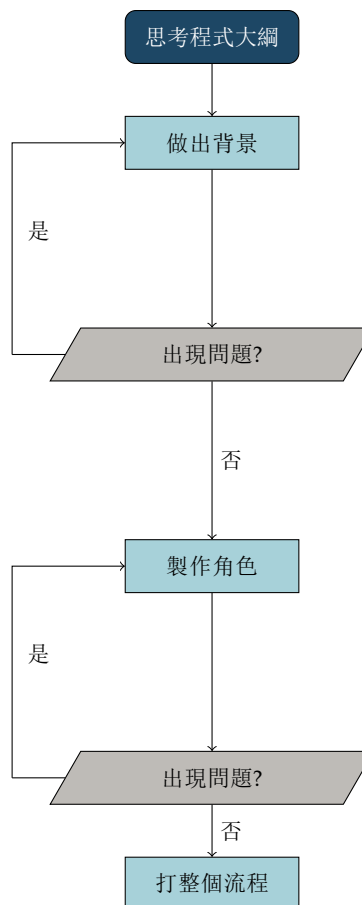
1. 因為對於一些名詞不太熟悉的緣故，命名命的不太好，只後應該可以把函式命名簡化配合工作流(譬如: `work_flow_1`, `work_flow_2`, `work_flow_3`)
2. 之後可以使用應用程式、網頁、執行檔來把單字使用上去。
3. 對於檔案名稱和資料夾的名稱可以再加強，
4. 在實作專案之前，可以先把整個檔案的執行過程使用流程圖表示出來，然後把檔案的名稱分配好，讓實作可以變得更加的容易。讓整個檔案可以變得更加的易讀

3 遊戲

3.1 序言

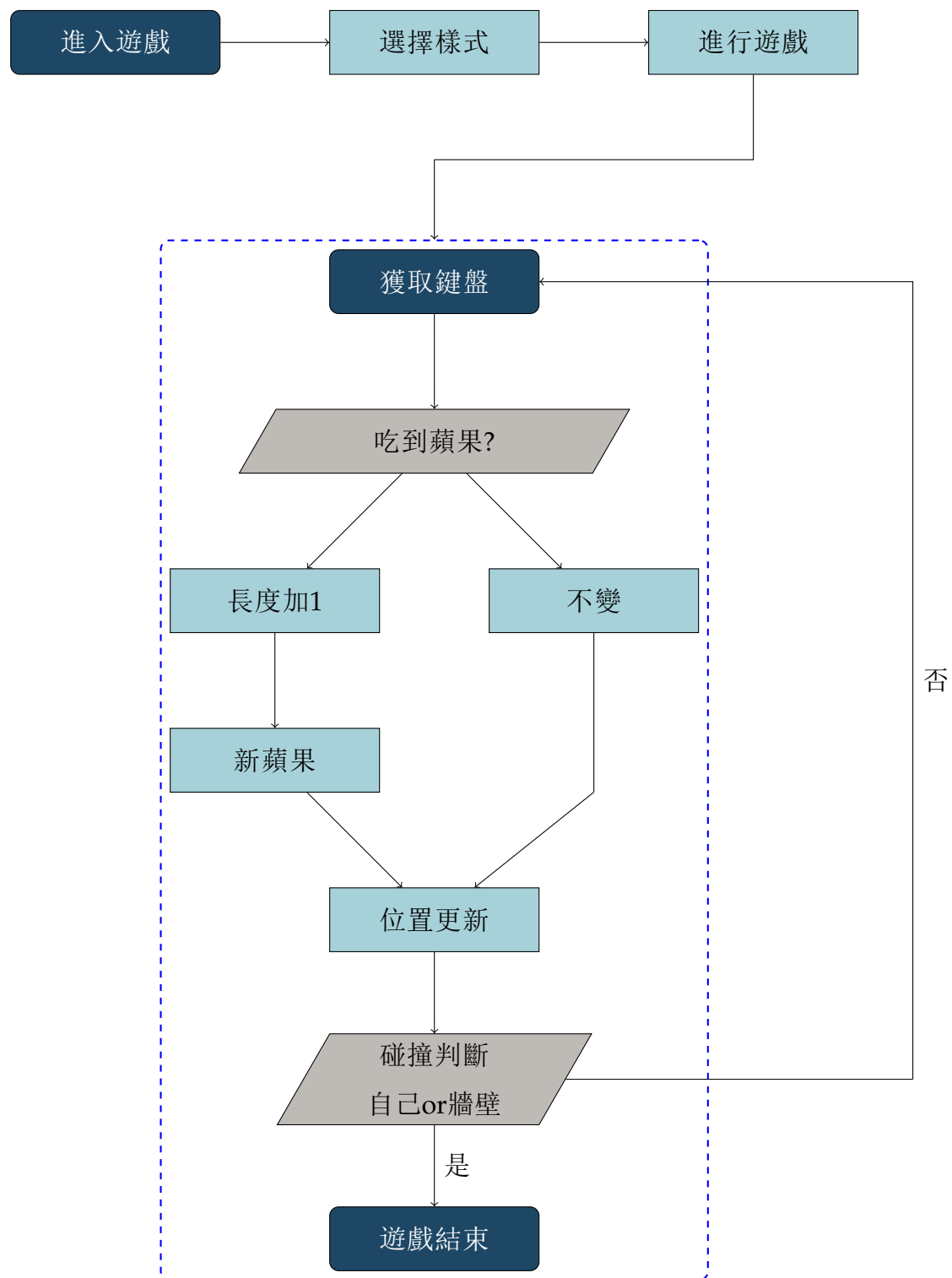
做遊戲件事情我還在逐漸摸索當中，目前還停留在熟悉製作過程當中，因為大部分的遊戲整個流程都會十分的複雜，我現在摸索的還只是停留在實作數十年前簡單的遊戲上面，就已經接近目前我能處理複雜程式的極限了，無論是角色的控制、遊戲進行的過程、獎勵機制，每一個都需要縝密的思考，還有在把許多的部件一遍遍的組成起來的時候也會遇到許多問題。

3.2 遊戲基本流程



3.3 貪吃蛇 (c++)

實作流程



實作動機

那時候因為資訊課需要自己實作出一個專案，也因為我對c++已經熟練，所以我就想說可不可以透過我學過的各種語法，來完成最經典的遊戲貪吃蛇。

遇到的困難

1. 一開始不知道怎們偵測鍵盤
⇒ 上網找函式庫
2. 因為更新全局導致畫面閃爍
⇒ 把更新改成局部的
3. 貪吃蛇長度太長也導致更新的次數太多
⇒ 觀察貪吃蛇頭尾的移動的性質來縮減更新次數

心得

因為是第一次做遊戲的原因，所以我許多地方都做的十分冗長，像是沒有好好的把節點的更新的邏輯做好，導致在後面需要多加許多程式碼來才能達成一樣的效果，對於這個簡單的小遊戲卻也用了200多行，還有許多可以進步的空間。

可以改進的方向

1. 之後我有學到如何變換字體顏色，可以來弄一個彩色的遊戲。
2. 其實許多函示可以把他變成.h檔案，讓主程式更加的簡潔。
3. 對於開始遊戲有點太過於突然了，可能可以加上一個教學過程
4. 可以透過寫入檔案來把遊玩的過程儲存下來，讓人可以感覺到自己的進步。
5. 之後再實作的過程中應該要先把遊戲的執行流程畫出來，然後依次的實作每一個小功能，一步步的測試，這樣才可以有效率的寫出程式。

3.4 貪吃蛇 (python)

實作動機

因為有用c++做過貪吃蛇，之後又因為python越學的越多，所以就開始學習pygame。也有把過程都錄成影片，影片裡面也又放上我個人學習程式的感想。

遇到的困難和解決方式

因之前有稍微學習過pygame，也有試過做出一點小專案，困難相對沒有那麼多。

1. 因為在pygame裡是直接進行畫面渲染，所以要加上文字就變得相對較困難。

⇒ 從網路上找到可以執行的程式碼學習。

心得

在兩次的比較之中，我很明確的感受到python和c++的差別，使用c++時，我寫了**200多行**，python則是**100多行**，做出來的成品也漂亮了許多，不但有**色彩**、**操作也更順暢**、也可以畫出**圓形**。但也不是說c++就一無是處，c++產生出來的檔案十分的小，不到**1MB**，而python則是**2、30MB**，所以我想如果遊戲在做大一點的話，可能c++的優勢會比python好許多。

項目 語言	學習難度	程式碼複雜度	執行檔大小	執行速度	使用時機
c++	高	高	較小	快	需要效能、大專案
python	低	低	較大(使用pyinstasller)	慢	小遊戲、中小型專案

4 未來展望

在經過這一年之後，我逐漸的清楚接下來需要做的事情。大概有以下幾點

1. 想學的程式語言
 - a. dark (0%)
 - b. javascript (50%)
2. 想做的專案
 - a. 寫出一個2D遊戲
 - b. 架設一個自己的網站
 - c. 寫出一個物理模擬器
 - d. 做出一個自己的應用程式
3. 想參加的比賽
 - a. 資訊奧林匹亞(TOI)
 - b. 全國資訊學科能力競賽
 - c. 網際網路程式設計全國大賽(NPSC)
4. 其他
 - a. 紀錄寫專案的過程
 - b. 把之前的專案完善好
 - c. 記錄下每一個專案的時間
 - d. 在每一次做專案之前都先做計畫表和流程圖
 - e. 寫完專案後都要寫一篇心得統整自己的收穫

5 結語

在這一年的旅途中，我的好奇心不斷的驅使我去做我從未做過的事物，如同冒險一般，雖然旅途中總是會遇到許多的不如意，但只要花時間，一次次的去嘗試錯誤、思考新作法，把自己的創意投入每一件事物，最後的收穫一定都是絢麗的。

我十分的感謝同學，同學們不經意間說出的話語總是能讓我豁然開朗，在我遇到瓶頸的時候也會聽我訴苦，在debug很無聊的時候也會陪伴我，讓我可以完成一件件我喜歡的事情，把內心的想法逐漸塑造。

我也十分慶幸我生活在這資訊充足的世界，無止盡的滿足我的好奇心，無論是什麼千奇百怪的問題，總是存在解答，也提供了無數的方向讓我去探索，縱使我現在的能力還不足以成為回答者，但我仍會持續的精進自我，期許能為世界帶來改變。

時光荏苒，高中生活如流水飛逝而去，在這短暫的時光當中，我想....

以靈動的指尖，
編織出斑斕的生活。
以自身的渺小，
緊握住龐大的夢想。
以無數的挫折，
澆灌出成功的花田。
以獨有的創意，
遨遊在資訊的海洋。

在剩下的兩年中，我會盡全力的做好每一件事，愉快的遨遊在資訊的海洋中。

我想，無論成功與否。

往後回首過去時，一定都是一段快樂的回憶。