

# Go Go Man

- A Game Written in Jack

Group Members :

B03902024 資工一 鄭筱樺

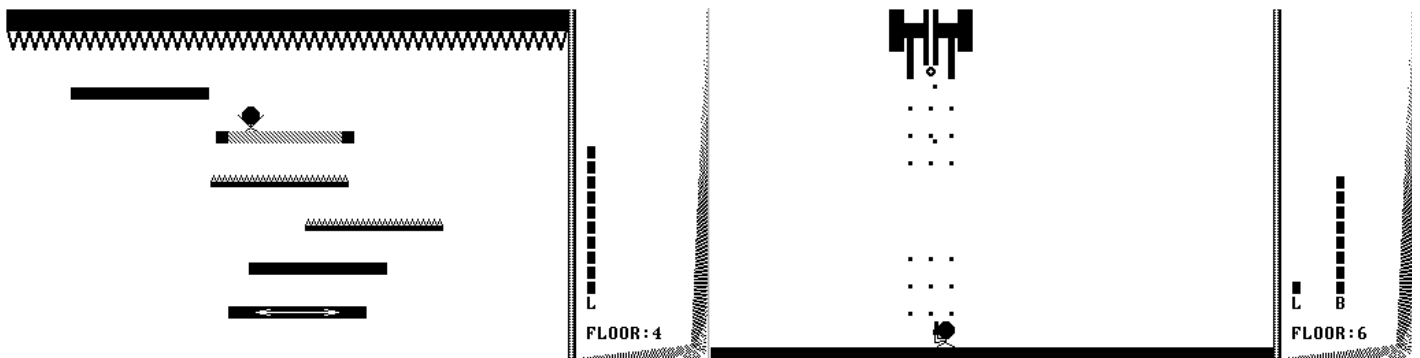
B03902066 資工一 陳俞安

B03902080 資工一 黃子賢

B03902078 資工一 林書瑾

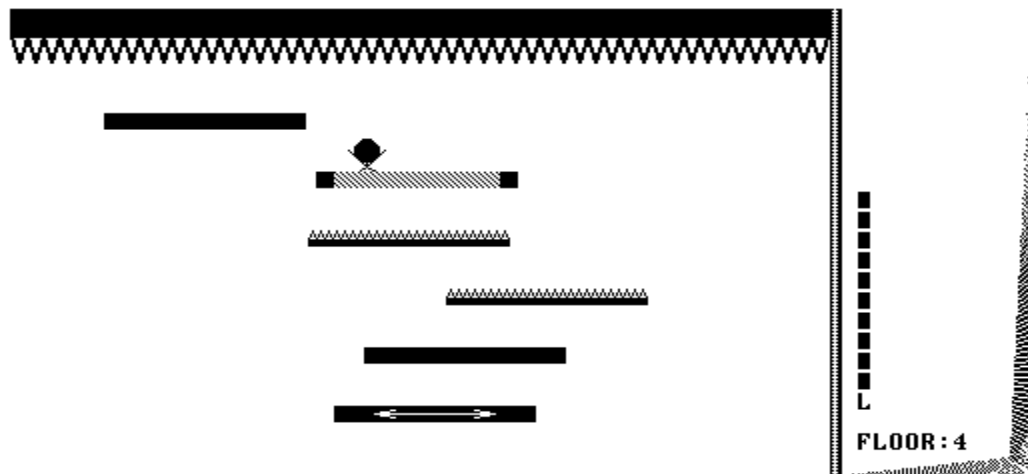
Time:

2015.1

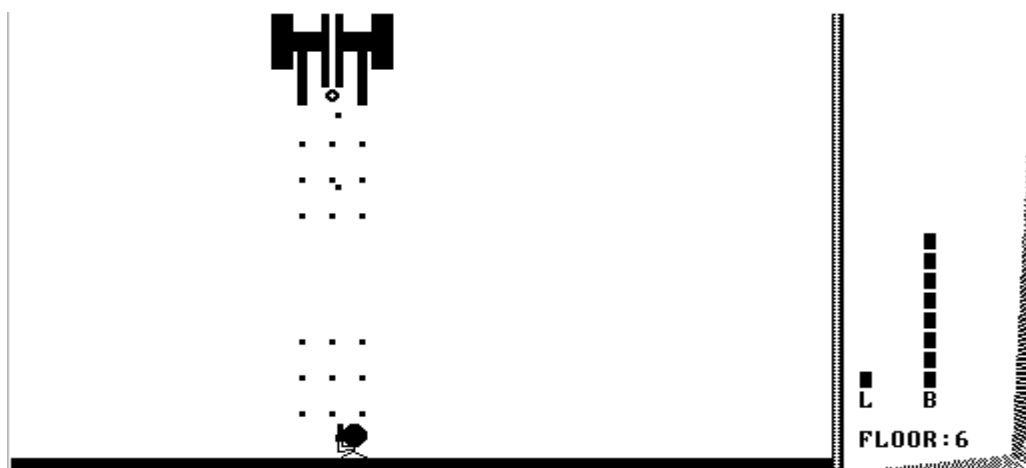


## 一、專案敘述

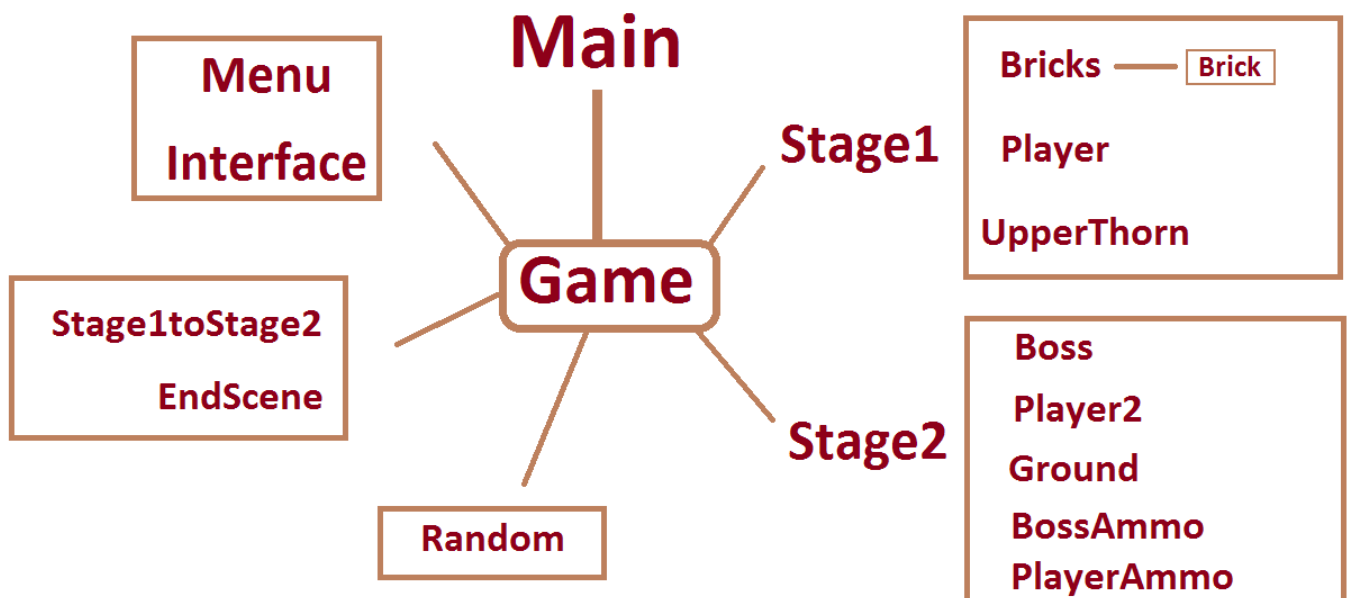
這個專案是我們使用Jack Language 所完成，設計出的一款遊戲，它的名稱叫做「Go Go Man」，象徵不停前進的人。遊戲主體分兩部分，首先是下樓梯，必須要不停移動才能存活，但是路途上會有許多障礙物，如刺磚、反轉磚、黏磚..... 主角一直往下走，磚塊的移動速度會漸漸增加，提高難度，下圖便是第一部分的遊戲場景。



遊戲的第二部分是會在一定層數出現的魔王，魔王會射出大量子彈攻擊玩家，而玩家必須使用槍枝來對付它，怪物的血量也會隨著層數的增加而變多，對主角造成更大的威脅。



## 二、專案架構



- **主體 - Main、Game、Menu、Stage1、Stage2**

遊戲的主體架構，也就是 Game，包含了遊戲的兩大舞台 Stage1 和 Stage2，其中 Stage 的架構主要是分別呼叫各個物件的 run 函式，讓每個物件都能執行一個影格，它只負責呼叫，控制整體遊戲流程與延遲，不負責處理物件詳細判斷。

選單裡設計了說明，是為了解使用者明白遊戲玩法，更能享受樂趣，以及最高紀錄的設計，讓玩家清楚明白目前的所達到的最高成績。還有選單可以藉由判斷使用者進入遊戲的時間，生成亂數種子並傳給 Random，讓後續函式使用。

- **人物 - Player、Player2**

Player 及 Player2 是屬於比較困難及複雜的部份，因為它處處與其它 Class 相關，需要花最多的時間與組員寫的其他程式溝通，有時要幫別人寫一些函式，有時則相反，還有要處理的細節也比較多，是我們這次 Project 花最多心思的地方。

- **動畫 - Stage1toStage2、EndScene**

Stage1toStage2、EndScene 是屬於美工的部份，用 Jack 畫圖及動畫真的很不容易，在只能使用圓、長方形、直線及點的情況下，畫出精美的圖需要時間及耐心，慢慢地用座標算出邊界及長度，但在遊戲中，介面非常重要，玩家都喜歡酷炫、有創意的動畫或圖像，所以這個部分是不可或缺的。

- **右側顯示介面 - Interface**

這部分所需要的資訊(例如：玩家血量、魔王血量)大多都可以從其他 class 裡取得，Interface 的任務只是將所有資訊顯示出來，不過慢慢討論後，小組中開始出現有些資訊想放在這個 class 的聲音，老實說，這是蠻違背我的分類概念，我一直堅持，Interface 就像你的電腦螢幕，他怎麼會存任何和遊戲有關的資訊呢？這也讓我和其他組員溝通上產生一點問題，我也提出過新建一個資料庫來存遊戲中任何的資訊的解決方案，不過因為當時很多資訊已經分散在不同的地方，要統合花費的時間成本太高，後來為了讓整體完成速度加快，我放棄原先的執著，成為部分資訊的中繼站。

- **亂數 - Random**

現在有很多種產生亂數的方法，但比較易懂與 Jack 能實作的大概是線性同餘，不過理解他的實作概念是一回事，能作出好的亂數器又是另外一回事(嚴格來講應該稱為偽亂數器)，引用某篇文章「RANDOM NUMBER GEUERATORS: GOOD ONES ARE HARD TO FIND (<http://goo.gl/00ngRJ>)」，此外亂數這個議題牽涉到很多數論的概念，我們也不太能明白那些好的亂數因子有什麼數學上的論證支持，最後只能找別人現成亂數因子來做，唯一加的功能就只有設定亂數的上下限

(mod 運算)，種子來源為按下遊戲開始的經過時間，雖然亂數還有許多改進的地方，但已經符合我們需求。

- **魔王與子彈 – Boss、BossAmmo、PlayerAmmo**

魔王的移動模式我們採取根據玩家的移動方向來決定，可是因為人物若和 BOSS 移動速度一樣，遊戲困難度過大，所以後來決定讓怪物慢一步。在攻擊方面，我們以變數做控制，當作攻擊間隔。在子彈的部分，遇到比較大的問題是，Jack 的陣列不完整，結構陣列很難用(無法直接呼叫 method)，所以我們使用 2 個陣列紀錄 X 和 Y，當作子彈的結構陣列。除此之外遇到最大的問題是在子彈移動的過程中，有時會有殘餘的沒有被清掉，可是在程式碼中又沒有錯，後來我們發現應該是 Jack Machine 本身模擬螢幕的問題，只好將子彈從圓形換為正方形，減輕運算壓力。

- **上方的尖刺 – UpperThorn**

這個實作最困難的部分主要在構圖，我們花費了時間計算尖刺的位置，整體的美觀、尖刺的長度和尖銳程度等等。因為這個物件同時也定義了遊戲整體的上界，我們也撰寫了函式，能夠傳回遊戲畫面的上界。

- **磚塊與地板 – Bricks、Brick、Ground**

迫於 Jack 在結構陣列方面的不足(無法直接呼叫 method)，所以我實作了一個類似結構陣列的 Bricks 來放置每個 Brick。這個 Class 與 Player 的關係非常密切，要隨時回傳目前磚塊的位置、狀態給 Player，也要靠變數紀錄時間，判斷是否重新出現磚塊。我們實作的磚塊有四種：普通磚塊、踩了移動速度會變慢的黏磚、有尖刺的磚塊、踩了左右會相反的反轉磚塊，增加遊戲樂趣與難度。

### 三、學習收穫

**子賢：**學到東西蠻多的，主要是團隊合作吧，在那時我們一起討論一起打 code 那時候真的很熱血，在團隊合作中最重要的是「溝通」，如何跟隊友說自己的要求很重要，而且最好先想好，因為我有一個函式寫了 3 遍刪了 2 遍，這樣會很不爽，所以思路要先想清楚再要求比較好，在討論的過程中，難免會有所分歧，那時要先多聽少說，等到真的了解對方的意思之後再互相討論，這樣進度會比較快，在 DEBUG 的時候會有點麻煩，因為這樣要去找是誰的錯，可是只要找到了，也只要他改就好，其他的不用動，這應該是寫物件的好處吧，原本在寫 C 的時候雖懂但是覺得沒必要，真的在寫時才覺得很好用，因為是彼此串聯再一起，所以要完成需要大家都完成，可是難免會怕自己雷隊友，所以有點壓力。

**筱樺：**這是我第一次用打程式碼的方式寫遊戲，覺得很新鮮也很有趣。跟著組員們，從構思、實作到美化，我們一起討論、一起寫程式，完成了這個遊戲。從中我學到了很多，程式技巧層面包括了物件的分工和執行的先後順序，要分成多少個 Class，要誰控制，Game 要先呼叫誰等等，這些都是學 C 語言時學不到的部份，做這次的 project 使我對物件有些概念；團體合作層面則是學到了如何和組員溝通、如何互相配合及妥協，每個人都有自己的想法，但是結果只能有一種，所以在意見不同時，溝通和協調是非常重要的，例如，在構想遊戲時，我們花了很多時間在討論子彈的取得方式及數量，有人希望有限個數，有人希望無限，有人希望玩家自己取得子彈，有人認為這樣會有缺失，在分析利與弊後，我們各退一步，達成了共識；我還學習到如何利用網路資源，當我們遇到大困難時，網路就是救星了。

很感謝組員們的互助，在困境中一起 debug，一起腦力激盪，大家一起努力的感覺真的很好。在我們這兩週的努力之下終於完成了 go go man 的遊戲，我覺

得成就感十足，不僅體驗到程式設計師作 Project 的感覺，也開啟了製作遊戲程式的那扇窗。我發現我對這塊領域是有興趣的，希望在系上學習更多知識後，寒暑假有空時，也可以自己寫一些小遊戲或者是 APP。

**俞安：**這次的 project 是我第一次和其他人合作寫程式，也是第一次寫類似物件導向的程式，它的思考邏輯和 C(程式程序導向)差別蠻大的，讓我剛開始接觸時遭遇到一些困難，為了貪圖方便，我的程式中也出現很多冗贅的寫法，導致後來修正的困難度提高。我想我在寫程式的能力可能沒有太大改變，但對學習和別人合作這塊倒是有不少的體會，透過和大家一起合作，也更能發現自己所不能看見的缺失。

**書瑾：**能夠與一個 4 人的 team 合作真的很有收穫，其中我學習到最多當然是在團隊溝通方面。面對一個這麼大的 project，該如何善加發揮四人的力量，以最有效率的方式分工，撰寫程式碼，再以最小的成本結合每個人的 code，合成出 project。這方面需要許多軟體工程的技術，但我們依靠僅有的物件導向概念，設計程式整體，再藉由物件整合，最後還是順利的完成這個專案。

我學習到的，是面對一個這麼龐大的 project，必須要如何改善我的 programming style，以及該如何分工給組員們好讓工作量相同，也要同時顧及不致使工作過於分散，當然也學習到在一個 team 裡與他人合作時該怎麼行事、溝通，還有最後將大家寫好的各個部分統整……這些都是我豐富的收穫。

#### 四、統計資料

原始碼總行數：2915 行

原始碼總大小：57.6KB

專案總工作時數約：120 hour

專案每人工作時數約：30 hour