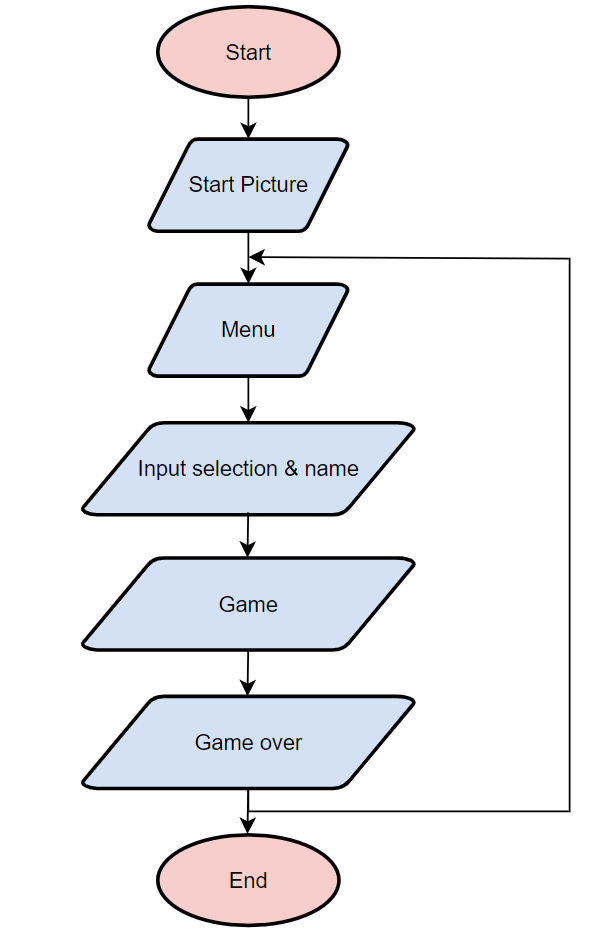
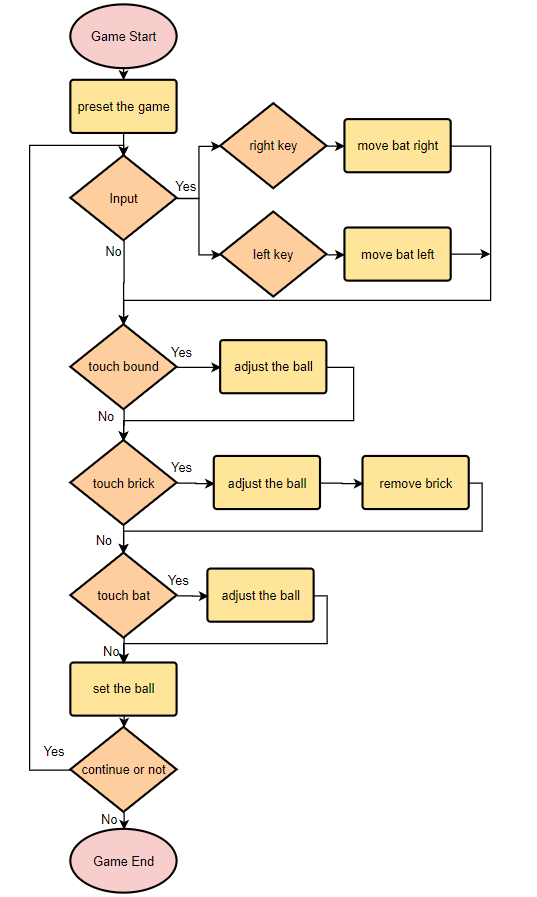
**Assembly Language Final Project**

1. 題目：打磚塊Break Bricks
2. 組別：第五組
3. 組員：陳懷平107201545、陳俊昇107201548、

王詩淳107201544、朱韋丞107409523。

1. 分工：
   1. 陳懷平：遊戲內容製作、打程式、debug、報告製作
   2. 陳俊昇：遊戲封面、menu製作、打程式、debug
   3. 王詩淳：製作ppt、測試程式、報告
   4. 朱韋丞：測試程式、報告
2. 流程圖





說明：第一張圖為整個程式運作的流程，第二張圖為程式運作進入遊戲的流程。

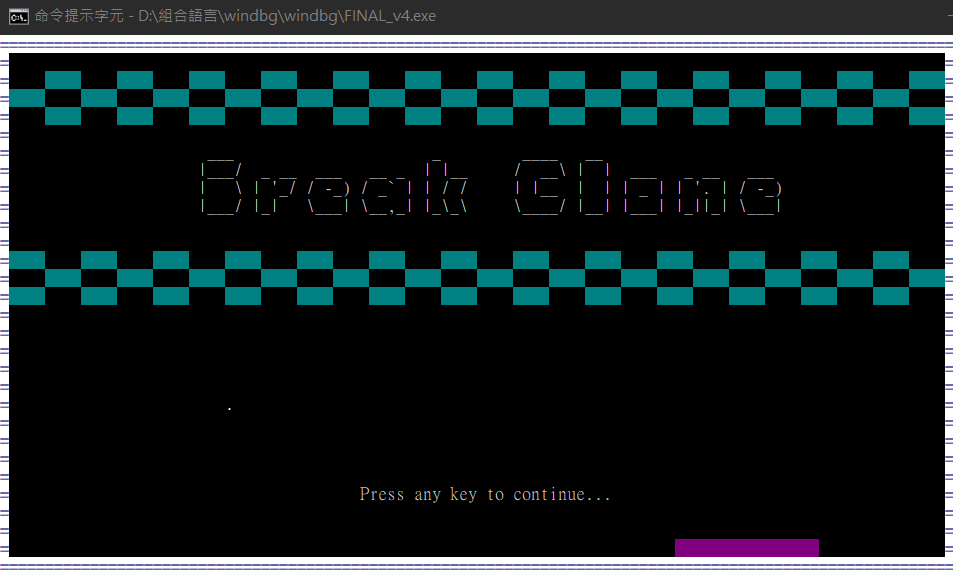
1. 程式碼與各子程式說明

備註：子程式說明順序按照在程式碼中流程的順序。

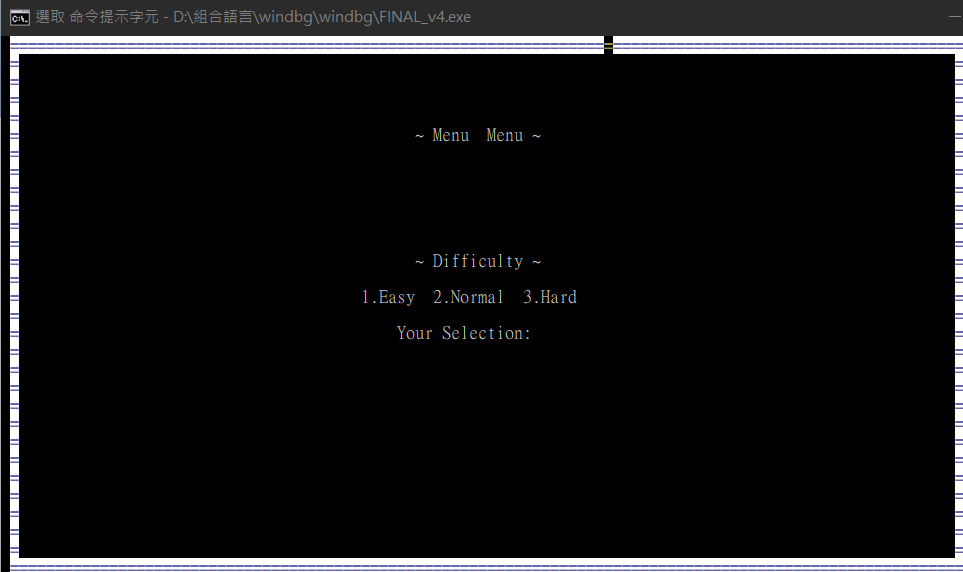
1. mGotoxy：設定現在輸出在螢幕的座標位置。
2. mWrite：將傳進來的字串輸出到螢幕上。
3. mRead：讀取使用者輸入的字串內容，並且作長度檢查，使得輸入的字串長度介於1~13之間。
4. mWriteString：與上述mWrite類似，都是輸出字串至螢幕上，只是使用不同方式。
5. StartGame：只是單純的控管主程式所需的所有子程式。
6. TitleScreen：繪製程式開始時的進入畫面。
7. Menu：繪製程式中Menu的畫面，並且讀取使用者對於難度設定的輸入&姓名字串輸入。
8. Walloutput：輸出大部分的外牆，以美化在各畫面中轉換的狀態。
9. SetConsoleTitle：設定CMD的Title。
10. GetStdHandle：取得輸出與輸入的握把。
11. GetConsoleScreenBufferInfo：取得當前CMD的資訊。
12. PreSet：首先設定CMD的長與寬，之後甩出亂數種子，設定磚塊位置，出現磚塊的機率為75%，並且設定橫桿與球與空白位置。
13. FillTheWindows：根據State中的狀態內容，將遊戲畫面輸出至螢幕上，其中0表示空白1表示球2表示橫桿3表示磚塊，並且針對磚塊位置做出不同的外觀顯示。
14. Game：首先初始化各項變數與畫面設定，接者讀取鍵盤輸入->設定橫桿，在目前位置擦除球，對球做外來預測；做球的外牆碰撞檢測->若撞到外牆則反轉球，若狀到下界則設定死亡；做球的磚塊碰撞檢測->撞到磚塊則反轉球並且擦除磚塊，其中撞到磚塊有三種反彈模式：橫向、垂直、斜角碰撞；做球的橫桿碰撞檢測->若撞到橫桿則反轉球；最後在未來位置畫上球，並且對遊戲狀態做檢查，若為繼續則跳回若為勝利或失敗則跳出。
15. Sleep：設定延時，類似於設定FPS。
16. WriteConsoleOutputAttribute：設定輸出背景。
17. WriteConsoleOutputCharacter：設定輸出文字。
18. Gameover：繪製程式中Gameover的畫面，並且輸出當前的分數與經過排序過後的計分表，並且讀取使用者繼續遊玩與否的選擇。
19. 詳細操作說明

首先請輸入難度，1為簡單2為普通3為困難，接者請輸入姓名，輸入長度請在1~13之間，進入遊戲，以鍵盤左與右方向鍵，分別操控橫桿向左與右移動。(詳細流程請見流程圖or Demo截圖。)

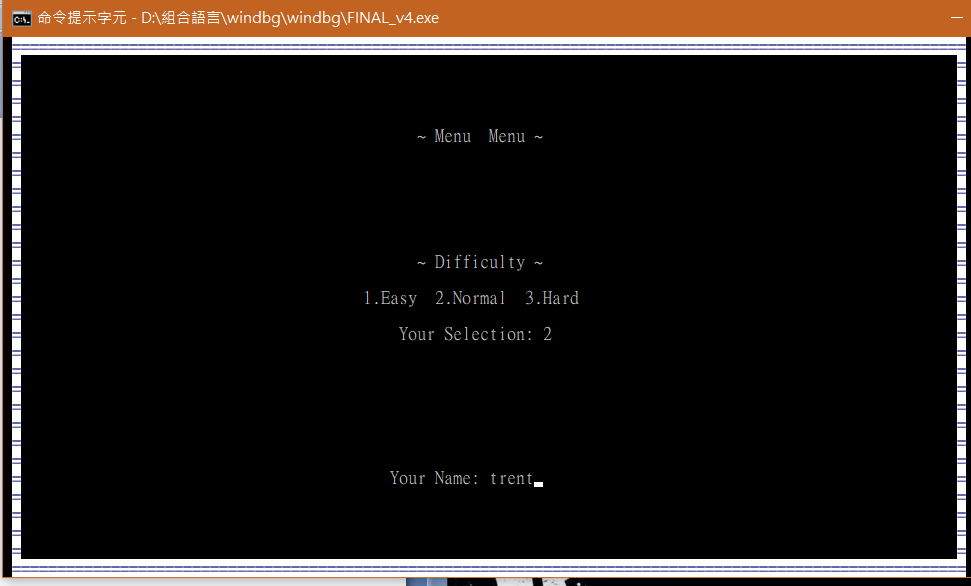
1. Demo截圖
2. 程式進入畫面



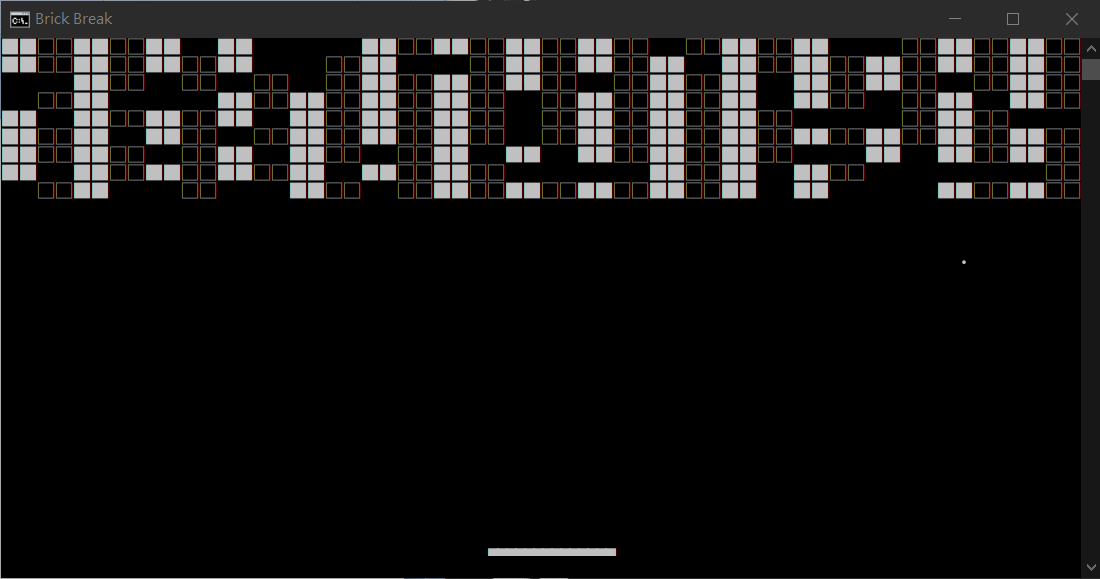
1. 程式Menu：難度選擇



1. 程式Menu：姓名填入



1. 遊戲畫面(使用鍵盤左右方向鍵操控橫桿)



1. 遊戲結束畫面、計分表(會對分數做排序)

