

一、 需求描述：

本程式為一棒球投打小遊戲，於遊戲一開始由玩家分別擔任攻守方並且創立各種不同「屬性」的投手及打者，在一次次的投球下攻防鬥智。是一款上手容易，變化無窮的遊戲。

二、 程式流程：

1. 執行程式後，系統顯示歡迎訊息並直接開始遊戲
2. 系統顯示需創建一位投手
 - (1) 玩家甲輸入投手的球員號碼
 - (2) 玩家甲輸入投遞棒球的手
 - (3) 玩家甲輸入投遞棒球的投球姿勢
 - (4) 玩家甲輸入投手第一個擅長投的球種
 - (5) 玩家甲輸入投手第二個擅長投的球種
 - (6) 玩家甲輸入投手第三個擅長投的球種
3. 重複步驟，依序創建其餘投手，總共需創建三位投手
4. 系統顯示需創建一位打者
 - (1) 玩家乙輸入打者的球員號碼
 - (2) 玩家乙輸入打者一個最擅長應付的球種
5. 重複步驟 4，依序創建其餘打者，總共需創建六位打者
6. 挑選階段－投手，這個階段系統會和玩家甲互動
 - (1) 系統顯示挑選階段－投手－畫面
 - (2) 玩家甲輸入挑選階段－投手－指令
7. 挑選階段－打者，這個階段系統會和玩家乙互動
 - (1) 系統顯示挑選階段－打者－畫面
 - (2) 玩家乙輸入挑選階段－打者－指令
8. 對決階段－投球，這個階段系統會和玩家甲互動
 - (1) 系統顯示對決階段－投球－畫面
 - (2) 玩家甲輸入對決階段－投球－指令
9. 對決階段－打擊，這個階段系統會和玩家乙互動
 - (1) 系統顯示對決階段－打擊－畫面
 - (2) 玩家乙輸入對決階段－打擊－指令
10. 系統呈現回合結算畫面
11. 重複步驟 6~10 直到遊戲結束，最後系統會顯示玩家乙的分數並結束遊戲

三、 Object/Class 敘述：

1. Pitcher class

Field

- (1) **ID**：球員號碼的 int
- (2) **Energy**：剩餘投球次數，預設 10
- (3) **Side**：左投或右投
- (4) **High**：上肩/低肩投
- (5) **ballA**、**ballB**、**ballC**：第 1 2 3 種擅長的球種
- (6) **onornot**：貯存投手是否在壘上的 boolean

Method

《設定》

- (1) **setID()**：將括號內的值指派為該物件的 ID，利用 If try/catch 檢查指派值是否為正整數。
- (2) **setenergy()**：將括號內的值指派為該物件剩餘的擊球次數，用於甲方更換投手時將剩餘體力歸 0（不得再上場）。
- (3) **setside()**：將括號內的值指派為該物件的 side，先轉成小寫，利用 If/else 檢查指派內容是否為 left/right。
- (4) **sethigh ()**：將括號內的值指派為該物件的 high，先轉成小寫，利用 If/else 檢查指派內容是否為 up/down。
- (5) **setballA ()**、**setballB ()**、**setballC ()**：將括號中的文字存為該投手擅長的三種球種。用 for 迴圈比對是否符合格式。
- (6) **setonornot ()**：設定投手是否在壘上，如果有改變則使用此功能。

《回傳資料》

- (7) **getID**：回傳投手編號，用於提示玩家乙某投手投球／在壘上等訊息
EX：投手 10 號正在場上
- (8) **getenergy**：回傳剩餘投球次數，用於提示玩家甲剩餘的投球次數以及檢查是否有剩餘體力。
EX：投手 10 號在場上，目前剩餘 3 點體力。
- (9) **getside**：回傳投球的手，用於比對左右投以輸出預測進壘點。
- (10) **gethigh**：回傳投球的姿勢，用於比對左右投以輸出預測進壘點。
EX:if("left".equals(pitcherA.getside()))&&"down".equals(pitcherA.gethigh())
- (11) **getballA**、**getballB**、**getballC**：回傳投手擅長球種，用於比對玩家甲輸入的投球指令的球種是否和擅長球種吻合。
- (12) **getdataforstricker**：輸出打者可知的投手資訊（僅含編號、體力）
EX：球員號碼：15 剩餘可投球次數：10
- (13) **getdata**：輸出投手的體力、編號、擅長球種（選擇投手選單用）

球員號碼：1 剩餘可投球次數：10 擅長的球種：a,b,c
《動作》

(14) **pitch**：投球動作，將能量值-1 若扣完後的能量值為 0 則將
onornot 由 **true** 改成 **false**

2. Stricker Class (應為 batter class)

Field

- (1) **ID**：球員號碼的 int
- (2) **Energy**：剩餘投球次數，預設 3
- (3) **Ball**:打者擅長的球種

Method

《設定》

- (1) **setID**：同 pitcher class
- (2) **setball**：同 pitcher class

《回傳資料》

- (3) **getID**：回傳打者編號，用於回合結算畫面
EX：場上打者球員號碼：4
 - (4) **getenergy**：回傳剩餘投球次數，用於提示玩家乙剩餘的打擊次數
以及檢查是否有剩餘體力。
 - (5) **getdataforpitcher**：輸出投手可知的打者資訊（僅含編號、體力），挑選投手畫面時使用
EX：球員號碼：15 剩餘可投球次數：10
 - (6) **getdata**：輸出投手的體力、編號、擅長球種（選擇投手選單用）
- 《動作》
- (7) **strick**：將體力-1

3. Main function

(1) 執行程式後，系統顯示歡迎訊息並直接開始遊戲

創建九個 object 分別為 pitcher1-3、stricker1-6，以及 int point 存玩家乙的總得分

(2) 系統顯示需創建一位投手

- 玩家甲輸入投手的球員號碼：使用 Pitcher.setID()
- 玩家甲輸入投遞棒球的手：使用 Pitcher.setside()
- 玩家甲輸入投遞棒球的投球姿勢：使用 Pitcher.sethigh()
- 玩家甲輸入投手第一個擅長投的球種：使用 Pitcher.setballA()
- 玩家甲輸入投手第二個擅長投的球種：使用 Pitcher.setballB()
- 玩家甲輸入投手第三個擅長投的球種：使用 Pitcher.setballC()

(3) 重複步驟，依序創建其餘投手，總共需創建三位投手

(4) 系統顯示需創建一位打者

- 玩家乙輸入打者的球員號碼：使用 Stricker.setID()
- 玩家乙輸入打者最擅長應付的球種：使用 Stricker.setball()

(5) 重複步驟 4，依序創建其餘打者，總共需創建六位打者

(6) 挑選階段－投手，這個階段系統會和玩家甲互動

- 系統顯示挑選階段－投手－畫面

《在場上的投手》

用 for 迴圈，當 pitcher class 的 OnorNot 為 true 時呼叫 getdata()

《不在場上的投手》

用 for 迴圈，當 pitcher class 的 OnorNot 為 false 時呼叫 getdata()

- 玩家甲輸入挑選階段－投手－指令

創建 Pitcher、stricker 陣列名為 pitcherlist、strickerlist

（以下簡稱為投手陣列、打者陣列）

，以及 int CountofPitcherOn/CountofStrickerOn 記錄在場上的球員在陣列中的編號，初始值=0。（以下簡稱為上場投手編號、上場打者編號）

將玩家甲輸入的指令命為 Command，用 split 將其分解用長度粗略除錯，詳細部分則用.euqals（檢查 select pitcher continue 拼字）for 迴圈（檢查投手編號是否存在）try/catch（檢查 int 該項是否為整數，避免程式跳出）檢查。

如果上述都通過則將投手編號吻合的投手於投手陣列，的位置寫入上場投手編號中。

註：遊戲有一設定為「更換投手時則不能再上場」，本程式設計將其體力值改為 0 以達成其效果，具體方法為再將新的上場投手編號複寫之前，

先比對兩者是否相同，若不同則將舊的投手體力歸零後再複寫上場球員編號，如遇到舊值為零（第一回合）則直接複寫。

(7) 挑選階段－打者，這個階段系統會和玩家乙互動

a. 系統顯示挑選階段－打者－畫面

《尚有機會上場的打者》

使用 if（檢查 `energy>0`）和 for 迴圈搭配 `stricker.getdata()` 輸出可用的打者。

《目前在場上的投手》

用 `pitcher.getdataforstricker()`，將上場投手的資料印出。

b. 玩家乙輸入挑選階段－打者－指令

同理於選擇投手，只是不考慮換人則體力歸 0 的問題

(8) 對決階段－投球，這個階段系統會和玩家甲互動

宣告 `int pitchpoint`、`swingpoint` 為進壘點、打擊點，`string balltype` 為投手該回合投出的球種

a. 系統顯示對決階段－投球－畫面

用 `pitcher.getdata()`、`stricker.getdataforpitcher()`，將上場投手的資料印出。

b. 玩家甲輸入對決階段－投球－指令

同理於選擇投手，多考慮僅能投出擅長球種（使用 `if`、`getballA()`、`getballB()`、`getballC()`）、進壘點只能為 1-9，並將投出球種、打擊點寫入對應變數。

(9) 對決階段－打擊，這個階段系統會和玩家乙互動

a. 系統顯示對決階段－打擊－畫面

比對該回合投出球種是否跟該打者擅長的球種吻合，若一樣直接顯示進壘點位置。

若進壘點於角落（1 3 7 9）則直接顯示該點附近的數字

如果在其他位置（2 4 5 6 8）則先用 `pitcher.gethigh()`和 `pitcher.getside()`判斷投手球路（上肩右投、上肩左投……）再用 `if` 給出可能的進壘點。

EX: `if(上肩右投){`

`if(進壘點=2){...print("1 2 4 5")}`

`}`

b. 玩家乙輸入對決階段－打擊－指令：同理於投球指令

(10) 系統呈現回合結算畫面

比對進壘點是否等於打擊點，如果是則加一分，再印出回合結束畫面

(11) 重複步驟 6～10 直到遊戲結束，最後系統會顯示玩家乙的分數並結束遊戲

使用 **while** 迴圈，判斷依據為當投手或打者的能量和均不為 0 時，持續進行。

當 **while** 迴圈中斷時，若打者的能量和 >0 則將其全部加入 **point** 中（投手操作不當的加分處理）

四、使用說明：

本程式為一棒球投打小遊戲，於遊戲一開始由玩家分別擔任攻守方並且創立各種不同「屬性」的投手及打者，在一次次的投球下攻防鬥智。

1. 遊戲一開始請由玩家甲擔任投手並開始操作
2. 玩家甲按照下列指示輸入投手資料
 - (1) 設定投手 ID：ID 須為正整數。註：投手 ID 不可與其他投手重複
 - (2) 設定投手投球的手：只能為“left”或“right”（左投、右投）
 - (3) 設定投手投球姿勢：只能為“up”或“down”（上肩投、低肩投）
a 快速球 b 曲球 c 滑球 d 變速球 e 指叉球 f 伸卡球
 - (4) 設定投手第一種擅長的球種：依上表輸入相對應的字母，如快速球輸入 a，投手在遊戲中只會投出他擅長的球種。
 - (5) 依此類推輸入第二、三種擅長的球種。註：同一個投手的擅長球種不可重複
 - (6) 以上內容如果輸入有誤，系統均會要求重新輸入
3. 重複步驟 2.共將創立三位投手，之後玩家乙擔任打者方並開始操作。
4. 玩家乙按照下列指示輸入打者資料
 - (1) 設定打者 ID：ID 須為正整數。註：打者 ID 不可與其他打者重複
a 快速球 b 曲球 c 滑球 d 變速球 e 指叉球 f 伸卡球
 - (2) 設定打者擅長應付的球種：依上表輸入相對應的字母，如快速球輸入 a
 - (3) 以上內容如果輸入有誤，系統均會要求重新輸入
5. 重複步驟 4.共將創立六位打者，並且轉為玩家甲操作。
6. 系統會顯示玩家甲目前上場的投手（第一回合或是原投手體力耗盡則顯示無）、以及其他位上場的投手，他們擅長的球種及剩餘體力值（每名投手初始體力值為 10 且只能上場一次）。玩家甲需輸入指令選擇投手，指令格式如下 **select pitcher [投手球員號碼]** EX: **select pitcher 5** 或是在場上有投手的情況下輸入 **continue** 則可繼續使用該投手。
輸入成功後則交給玩家乙操作。
注意：每名投手只能上場一次，一經撤換該名投手的體力值即會歸零，並不得再上場
7. 系統會提示玩家乙目前可以使用的打者，以及它們各自的能量、體力值、擅長應付的球種。還有當前玩家甲派出的投手編號和他剩餘的投球次數。玩家乙需要輸入指令選擇打者，指令格式如下
select batter [投手球員號碼] EX: **select batter 5**
輸入成功後，則交給玩家甲操作
8. 玩家甲須命令場上投手該投出何種擅長球種，具體指令如下：
pitch [球種編號] [進壘點編號] EX **pitch a 5**。球種編號須為擅長球種，進壘點為 1-9 的其中一個正整數。輸入成功後則交給玩家乙操作。

9. 系統將提示玩家乙投手投的球可能的進壘點，如果投手投出的球種正好為場上打者擅長對付的球種，則可以精確知道進壘點。

1	2	3
4	5	6
7	8	9

反之系統則會提示玩家乙可能的進壘點（如圖，圖片來源 hw3.pdf），當玩家甲投球投向 1 時，系統則會顯示「可能的進壘點為：1、2、4、5」提示的點位會因為投手投球的姿勢而有所不同，左投投手球路偏右，上肩投手球路偏低，反之亦然

知悉可能進壘點後，玩家乙將給出打擊指令，指令格式如下：

swing [進壘點編號] EX.:swing 9 輸入成功後，本回合結束。

10. 回合結束後系統將顯示本回合結果，包含：擊中與否、場上投手編號、場上打者編號、目前累計得分
11. 重複步驟 6-10 直到投手／打者其中一方體力為零。

注意：若因玩家甲操作不當，頻繁更換投手，導致三個投手皆不能上場時。玩家乙剩餘打擊機會，直接換算成分數。如打者剩 7 次打擊機會，直接+7 分。

五、其他：

1. 本系統所有指令均不分大小寫，程式運行過程中已經詳細的處理各種可能發生的錯誤。