

I. 作者簡介

- A. 姓名：胡瑀真
- B. 系級：測量系 116 級
- C. 學號：F64126147
- D. e-mail：F64126147@gs.ncku.edu.tw

II. 程式簡介

A. 編寫功能

此題我運用三個函式搭配一個主程式來撰寫（圖一）。

```
1  #include <iostream>
2  #include <cstdlib>
3  using namespace std;
4  void printCat(char winner);
5  void printCat2(char winner);
6  void printBoard(string C[0][3]);
7
8  ~ int main() {
```

圖一、標頭檔和函式之程式碼

1. 主程式

(9) 使用 system("color XY") 語法設定小黑窗成淺黃底藍字，以增加美觀度。

(10)~(17) 宣告變數 end 代表遊戲是否結束；X 和 Y 為輸入座標；turn 表示玩家在同一回合的輸入次數；number 表示回合數；scoreA 和 scoreB 分別代表玩家 A 和 B 的分數；字元 type 和 mark 分別表示玩家和玩家所執的棋子。最後令一個二維字串陣列 C 代表棋盤，並用兩個雙重 for 迴圈為陣列賦值。

(18)~(20) 列印歡迎文字並呼叫函式 printBoard 以印出棋盤。

```
9      system("color E9");
10     int end=0,X,Y,turn=0,number=0,scoreA=0,scoreB=0;
11     char type='A',mark='O';
12     string C[3][3];
13     ~ for(int i=0;i<3;i++) {
14     ~     for(int j=0;j<3;j++) {
15         C[i][j] = "_"; // 初始化棋盤
16     }
17     }
18     cout<<"歡迎來到五子棋遊戲！\n遊戲中玩家A執O棋，玩家B執X"<<endl;
19     cout<<"祝你們遊玩愉快！"<<endl;
20     printBoard(C);
```

圖二、主程式之程式碼 1

(22) 使用 while 迴圈以供回合數的增加，當 end 小於 3 時不斷重複。

(23) 令一布林變數 validInput 代表玩家的輸入是否有效，並供後續 do while 迴圈判斷

(25) do-while 迴圈的 do 開始

(26) ~ (53)

首先使用 if 敘述，當玩家在同一回合的輸入次數不為 0 時，判斷四個 if 敘述並印出提示輸入的文字。第一個 if 敘述：當輸入次數 3 次以上時，印出生氣貓咪的圖形。第二、三個 if 敘述：當 X 座標和 Y 座標被輸入的值不在範圍內，則印出提示文字，其中第三個 if 敘述是雙重 if 敘述以增加閱讀時的連貫性。第四個 if 敘述：當輸入的座標在陣列 (棋盤) C 中不是代表空格的「_」，則印出提示文字。四個 if 敘述皆判斷完畢後，統一印出提示輸入的文字。

當輸入次數為 0，表示是第一次且還沒輸入，故不做以上判定，印出提示輸入的文字。最後讓使用者輸入並使輸入次數加 1。

```
22 ~ while (end<3){
23     bool validInput=false;
24     //判別輸入是否有效
25 ~ do {
26 ~     if(turn!=0){
27 ~         if(turn>=3){
28             cout<<endl;
29             cout << "\\      /\\" << "#請玩家好好輸入！"<<endl;
30             cout << " ) ( ' "<<endl;
31             cout << "( > )" <<endl;
32             cout << " \\( __ )" <<endl<<endl;
33         }
34 ~         if (X>3 || X<1) {
35             cout<<"該X座標不在範圍內，";
36         }
37 ~         if (Y>3 || Y<1) {
38 ~             if (X>3 || X<1) {
39                 cout<<"且";
40             }
41             cout<<"該Y座標不在範圍內，";
42         }
43 ~         if (X>=1 && X<=3 && Y>=1 && Y<=3 && C[X-1][Y-1]!="_") {
44             cout<<"該座標已被輸入，";
45         }
46         cout<<"請玩家"<<type<<"重新輸入XY座標（座標間由空格隔開）：";
47     }
48
49 ~     else{
50         cout<<"請玩家"<<type<<"輸入XY座標（座標間由空格隔開）：";
51     }
52     cin >> X >> Y;
53     turn++;
```

圖三、主程式之程式碼 2

(55~59) 接續 do 部分，再使用一個 if 敘述判定輸入是否在範圍且沒有重複輸入同樣數值的問題後，將布林變數 validInput 宣告為 true。do while 迴圈會在 validInput 為 true 時結束，之後初始化輸入次數，令為 0。

(62)~(65) 將玩家輸入的座標加入陣列 C 中，將該座標的值令為玩家所執的棋。之後再次呼叫函式 printBoard 印出棋盤。

```
55         if ((X>=1 && X<=3) && (Y>=1 && Y<=3) && C[X-1][Y-1]=="_") {
56             validInput = true;
57         }
58     } while (!validInput);
59     turn = 0;
60
61     //將OX輸入進陣列中
62     C[(X-1)][(Y-1)]=mark;
63
64     //印出陣列
65     printBoard(C);
```

圖四、主程式之程式碼 3

(68)~(97)

共同使用一個 for 迴圈，令 i 為 1，每次迴圈 i 即加 1，並且在不小於 3 時才結束迴圈。for 迴圈用來判別是否有橫向連線和直向連線。第一個 if 敘述用以判定橫向，第二個用以判定直向。這兩個 if 敘述差別只在行列相反、勝利文字的列印，故僅說明橫向連線：一開始的 if 敘述判定 C[i][0] 是否不為初始的「_」，若是，則判定 C[i][0] 和 C[i][1] 是否相等，若是，再判定 C[i][1] 和 C[i][2] 是否相等，藉此找到橫向連線，找到後，除了印出勝利文字外，也用條件運算字「?:」判斷應該為誰加分，最後印出比分並令 end 為 1。

在直向連線中，用條件運算字「?:」判斷玩家是否有大於一個連線，若是，則在勝利文字的列印上多列印「再次」。

```

68 //判別勝負
69 for(int i=0;i<3;i++){
70     //判別橫向連線
71     if(C[i][0]!="_"){
72         if(C[i][0]==C[i][1]){
73             if(C[i][1]==C[i][2]){
74                 cout<<"透過橫向連線，";
75                 cout<<"玩家"<<type<<"獲勝！"<<endl;
76                 printCat(type);
77                 type=='A'?scoreA++:scoreB++;
78                 cout<<"現在比分是(A:B)："<<scoreA<<":"<<scoreB<<endl;
79                 end=1;
80             }
81         }
82     }
83 }
84
85 //判別直向連線
86 if(C[1][i]!="_"){
87     if(C[0][i]==C[1][i]){
88         if(C[1][i]==C[2][i]){
89             cout<<"透過直向連線，";
90             cout<<"玩家"<<type<<(end==1?"再次":"")<<"獲勝！"<<endl;
91             printCat(type);
92             type=='A'?scoreA++:scoreB++;
93             cout<<"現在比分是(A:B)："<<scoreA<<":"<<scoreB<<endl;
94             end=1;
95         }
96     }
97 }
98 }

```

圖五、主程式之程式碼 4

(99)~(120) 判斷對角線連線和反對角線連線的程式，共用一個 if 敘述，該 if 敘述判定位於棋盤中央的 C[1][1] 是否不為初始的「_」，若是，則同樣用雙重 if 敘述做判定。因兩種情況的差別也只在於行列相反，故僅說明對角線的情況：若一開始的敘述為是，則判定 C[0][0] 和 C[1][1] 是否相等，若是，再判定 C[1][1] 和 C[2][2] 是否相等，藉此找到對角線，找到後的處理方式則和判斷直線連線相同。

用條件運算字「?:」判斷玩家是否有大於一個連線，若是，則在勝利文字的列印上多列印「再次」。

(122)~(128) 判斷平手的情況，利用一開始令的變數 number (代表回合數)，若 number 為 9 則表示玩家已為陣列 C 的所有值賦上新的字串，又 end 為 0 表示為分出勝負，故用 if 敘述，當同時達成以上條件，則印出宣布平手的文字與獎勵，最後令 end 為 2。

```

99      //判別斜連線
100  ✓   if(C[1][1]!="_"){
101  ✓       if(C[0][0]==C[1][1]){
102  ✓           if(C[1][1]==C[2][2]){
103               cout<<"透過斜向連線，";
104               cout<<"玩家"<<type<<(end==1?"再次":"")<<"獲勝！"<<endl;
105               printCat(type);
106               type=='A'?scoreA++:scoreB++;
107               cout<<"現在比分是(A:B)："<<scoreA<<":"<<scoreB<<endl;
108               end=1;
109           }
110       }
111  ✓   if(C[0][2]==C[1][1]){
112  ✓       if(C[1][1]==C[2][0]){
113           cout<<"透過斜向連線，";
114           cout<<"玩家"<<type<<(end==1?"再次":"")<<"獲勝！"<<endl;
115           printCat(type);
116           type=='A'?scoreA++:scoreB++;
117           cout<<"現在比分是(A:B)："<<scoreA<<":"<<scoreB<<endl;
118           end=1;
119       }
120   }
121 }
122 //判別平手
123  ✓   if(number==9 && end==0){
124       cout<<"\t雙方平手！"<<endl;
125       printCat2('A');
126       printCat2('B');
127       cout<<"現在比分是(A:B)："<<scoreA<<":"<<scoreB<<endl;
128       end=2;
129   }

```

圖六、主程式之程式碼 5

(130)~(156)

使用一個 if 敘述，當 end 不為 0 時（表示有勝負或平手的情況產生），則執行以下程式：詢問玩家是否要再玩一局遊戲，並令一字串 n 儲存玩家輸入的字串。

再使用一個 if 敘述，若 n 是 yes，則用兩個 for 迴圈把陣列 C（棋盤）初始化，再呼叫函式 printBoard 以印出棋盤。最後把 end 和 number 初始化另為 0，並用條件運算字「?:」判斷上一局是否有勝負，若有勝負（end 為 0）則由輸家當先手，若無勝負則由上一局的後手先開始。

若 n 為 no，則用一個 if 敘述判斷 scoreA 是否大於 socreB，再用兩個 else if 敘述分別表示 scoreA 是否小於、等於 socreB 的情況，再呼叫 printCat2 作為給勝者的獎勵。最後令 end=3 以結束 while 迴圈。

```

130  ✓   if(end!=0){
131      cout<<"要不要再玩一局呢？如果要的話，請輸入yes，如果不要，請輸入no"<<endl;
132      string n;
133      cin>>n;
134  ✓   if(n=="yes"){
135  ✓       for (int i = 0; i < 3; i++) {
136  ✓           for (int j = 0; j < 3; j++) {
137               C[i][j] = "-";
138           }
139       }
140       printBoard(C);
141       cout<<"換"<<(end==1?"輸":"前一局後手")<<"的玩家先開始!"<<endl;
142       end=0;
143       number=0;
144   }
145  ✓   else if(n=="no"){
146  ✓       if(scoreA>scoreB){
147           cout<<"恭喜A獲得最終勝利!! "<<endl;
148           printCat2('A');
149       }
150  ✓       else if(scoreA<scoreB){
151           cout<<"恭喜B獲得最終勝利!! "<<endl;
152           printCat2('B');
153       }
154  ✓       else if(scoreA==scoreB){
155           cout<<"A和B平手!! "<<endl;
156           printCat2('A');
157           printCat2('B');
158       }
159       end=3;
160   }
161   }

```

圖七、主程式之程式碼 6

(162)~(165) 在每一次 while 迴圈的最後都使 number (回合數) 加 1，並用條件運算字「?:」改變 type 和 mark 各自代表的值，以因應下一次執行 while 迴圈。

(166) 在主程式執行到最後傳回 0。

```

162      number++;
163      type = (type == 'A' ? 'B' : 'A');
164      mark = (mark == '0' ? 'X' : '0');
165  }
166  return 0;
167  }

```

圖八、主程式之程式碼 7

2. 函式 printCat、printCat2

使用跳脫字元\t等字元印出貓咪圖形 printCat 出現於一局勝負分曉時，printCat2 出現於玩家不在繼續玩時，獎勵最終勝者。

```

164 v void printCat(char winner) {
165     cout << "\t\t /\_/\ \" << endl;
166     cout << "\t\t ( o.o ) \" << endl;
167     cout << "\t\t > ^ < \" << endl;
168     cout << "\t\t / \ \" << endl;
169     cout << "\t\t / I ♥ \" << winner << " \ \" << endl;
170     cout << "\t\t /_____ \" << endl;
171 }
172
173 v void printCat2(char winner) {
174     cout << "\t\t /\_/\ \" << endl;
175     cout << "\t\t ( o.o ) \" << endl;
176     cout << "\t\t > ^ < \" << endl;
177     cout << "\t\t / \_/\ \" << endl;
178     cout << "\t\t/_____ \" << endl;
179     cout << "\t\t ♥♥♥\" << winner << "♥♥♥\" << endl;
180     cout<<"\t歡迎下次再度遊玩！"<<endl;
181 }

```

圖九、printCat 和 printCat2 之程式碼

3. 函式 printBoard

除使用多個跳脫字元\t外，還使用了一個 for 迴圈和一個雙重 for 迴圈分別印出棋盤的行列標籤、棋盤本身字元。

```

183 v void printBoard(string C[3][3]){
184     cout<<endl;
185     cout<<"\t\t ";
186 v     for(int i=1;i<=3;i++) {
187         cout<<i<<" "; // 列印列號
188     }
189     cout<<endl;
190 v     for(int i=0;i<3;i++){
191         cout<<"\t\t"<<i+1<<" ";
192 v         for(int j=0;j<3;j++){
193             cout<<C[i][j]<<" ";
194         }
195         cout<<endl;
196     }
197     cout<<endl;
198 }

```

圖十、printBoard 之程式碼

B. 編寫創意

1. 使用 `system("color E9")` 增加版面的美觀度。
2. 若玩家輸入錯誤的次數在三次（含）以上，則印出生氣貓咪的圖案，並提醒玩家輸入時多加留意。
3. 在為變數命名時，多使用「駝峰式命名」，增加程式的易讀性。
4. 增加 `scoreA` 和 `scoreB` 兩變數為玩家 A 和 B 計分，並用 `printCat` 的函式印出可愛的貓咪給予贏家獎勵。
5. 詢問玩家是否還要再玩一局，若玩家回答 `yes`，則會繼續延續原本玩家 A 和 B 的計分（`scoreA` 和 `scoreB` 不變），讓兩玩家可以繼續競爭，並且在新的一局會由輸家或是上一局的後手當先手，增加遊戲的趣味性。
6. 即使玩家不願意再玩一局，回答 `no`，仍然比較玩家 A 和 B 的分數高低，並且用 `printCat2` 的函式給予分數較高方可愛的貓咪圖案作為獎勵。
7. 在小黑窗印出東西時，盡量以函式處理讓主程式的畫面更加乾淨。
8. 使用 `for` 迴圈時，運用老師在課堂所講的概念「在迴圈中命名的參數隨迴圈結束而消失」，令每個 `for` 迴圈的初始值皆為 `i`，若有第二層責令其初始值為 `j`，以方便編寫和查閱。

III. 結果簡介

A. 輸入格式與運作流程

默認玩家 A 為先手。看到提示輸入的文字後，輸入座標 X（行數）和 Y（列數），兩座標以空格隔開（如圖十一）。輸入完畢按下 `enter` 鍵後，若為有效輸入，即可看到給另一個玩家的提示輸入文字；若為無效輸入，即可看到描述無效原因和提示重新輸入的文字（如圖十二），玩家需重新輸入。

一局的對弈結束後，會出現兩玩家比分和獎勵贏家的文字和圖片，最後出現詢問是否要再玩一局的提示（如圖十三），若玩家輸入 `yes` 則沿用比分並重新開始一局遊戲，但換輸家或原本的後手當先手（如圖十四）；若玩家輸入 `no` 則出現獎勵最終贏家的圖片。


```

歡迎來到五子棋遊戲！
遊戲中玩家A執O棋，玩家B執X
祝你們遊玩愉快！

      1 2 3
    1 - - -
    2 - - -
    3 - - -

請玩家A輸入XY座標（座標間由空格隔開）：

```

圖十一、輸入格式 1

```

歡迎來到五子棋遊戲！
遊戲中玩家A執O棋，玩家B執X
祝你們遊玩愉快！

      1 2 3
    1 - - -
    2 - - -
    3 - - -

請玩家A輸入XY座標（座標間由空格隔開）：1 2

      1 2 3
    1 - 0 -
    2 - - -
    3 - - -

請玩家B輸入XY座標（座標間由空格隔開）：4 2
該X座標不在範圍內，請玩家B重新輸入XY座標（座標間由空格隔開）：2 4
該Y座標不在範圍內，請玩家B重新輸入XY座標（座標間由空格隔開）：4 4

\      /\ #請玩家好好輸入！
 )    (')
(      > )
 \(\_)|

該X座標不在範圍內，且該Y座標不在範圍內，請玩家B重新輸入XY座標（座標間由空格隔開）：

```

圖十二、輸入格式 2

```

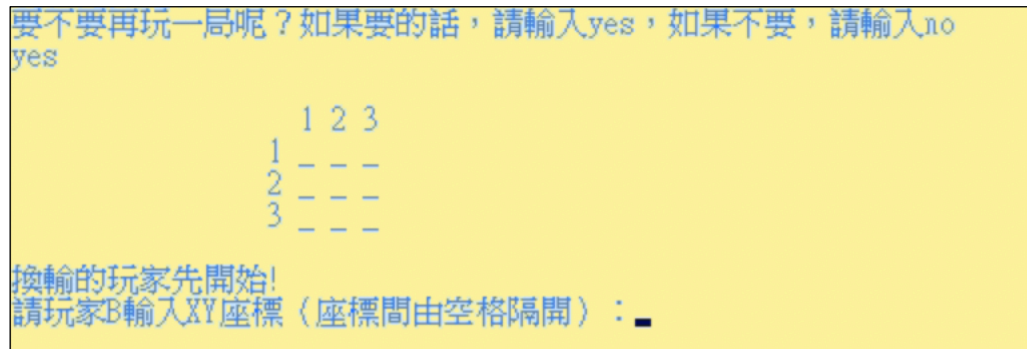
透過斜向連線，玩家A獲勝！

      /\_/\
     ( o.o )
      > ^ <
     /\ loveA \
    /\_/_/_/\

現在比分是(A:B)：1:0
要不要再玩一局呢？如果要的話，請輸入yes，如果不要，請輸入no

```

圖十三、輸入格式 3



圖十四、輸入格式 4

B. 結束畫面

第一局遊戲之測資為範例測資，結果為 A 勝得 1 分，而第二、三局則為自訂測資，皆延續前一筆測資的比分，結果分別為 B 同時連兩條線得 2 分和 A 再次勝利，最後雙方平手（2：2）。

歡迎來到五子棋遊戲！
 遊戲中玩家A執O棋，玩家B執X
 祝你們遊玩愉快！

	1	2	3
1	-	-	-
2	-	-	-
3	-	-	-

請玩家A輸入XY座標（座標間由空格隔開）：2 2

	1	2	3
1	-	-	-
2	-	O	-
3	-	-	-

請玩家B輸入XY座標（座標間由空格隔開）：2 2
 該座標已被輸入，請玩家B重新輸入XY座標（座標間由空格隔開）：1 2

	1	2	3
1	-	X	-
2	-	O	-
3	-	-	-

請玩家A輸入XY座標（座標間由空格隔開）：1 1

	1	2	3
1	O	X	-
2	-	O	-
3	-	-	-

請玩家B輸入XY座標（座標間由空格隔開）：3 3

	1	2	3
1	O	X	-
2	-	O	-
3	-	-	X

請玩家A輸入XY座標（座標間由空格隔開）：3 1

	1	2	3
1	O	X	-
2	-	O	-
3	O	-	X

請玩家B輸入XY座標（座標間由空格隔開）：1 3

	1	2	3
1	O	X	X
2	-	O	-
3	O	-	X

請玩家A輸入XY座標（座標間由空格隔開）：2 1

	1	2	3
1	O	X	X
2	O	O	-
3	O	-	X

透過直向連線，玩家A獲勝！

```
  /\_/\
 (  o.o  )
  > ^ <
 /I loveA\
/_____\  

```

現在比分是(A:B)：1:0

要不要再玩一局呢？如果要的話，請輸入yes，如果不要，請輸入no
yes

```
   1 2 3
  1 - - -
  2 - - -
  3 - - -
```

換輸的玩家先開始!

請玩家B輸入XY座標（座標間由空格隔開）：1 3

```
   1 2 3
  1 - - X
  2 - - -
  3 - - -
```

請玩家A輸入XY座標（座標間由空格隔開）：1 2

```
   1 2 3
  1 - 0 X
  2 - - -
  3 - - -
```

請玩家B輸入XY座標（座標間由空格隔開）：1 1

```
   1 2 3
  1 X 0 X
  2 - - -
  3 - - -
```

請玩家A輸入XY座標（座標間由空格隔開）：3 1

```
   1 2 3
  1 X 0 X
  2 - - -
  3 0 - -
```

請玩家B輸入XY座標（座標間由空格隔開）：2 2

```
   1 2 3
  1 X 0 X
  2 - X -
  3 0 - -
```

請玩家A輸入XY座標（座標間由空格隔開）：2 1

```
   1 2 3
  1 X 0 X
  2 0 X -
  3 0 - -
```

請玩家B輸入XY座標〈座標間由空格隔開〉：2 3

```
  1 2 3
1 X 0 X
2 0 X X
3 0 _ _
```

請玩家A輸入XY座標〈座標間由空格隔開〉：3 2

```
  1 2 3
1 X 0 X
2 0 X X
3 0 0 _
```

請玩家B輸入XY座標〈座標間由空格隔開〉：3 3

```
  1 2 3
1 X 0 X
2 0 X X
3 0 0 X
```

透過直向連線，玩家B獲勝！

```
  /\_/\
 ( o.o )
  > ^ <
  /\_/\
 /I loveB\
/_____\  
|
```

現在比分是(A:B)：1:1

透過斜向連線，玩家B再次獲勝！

```
  /\_/\
 ( o.o )
  > ^ <
  /\_/\
 /I loveB\
/_____\  
|
```

現在比分是(A:B)：1:2

要不要再玩一局呢？如果要的話，請輸入yes，如果不要，請輸入no
yes

```
  1 2 3
1 _ _ _
2 _ _ _
3 _ _ _
```

換輸的玩家先開始!

請玩家A輸入XY座標〈座標間由空格隔開〉：2 2

```
  1 2 3
1 _ _ _
2 _ 0 _
3 _ _ _
```

請玩家B輸入XY座標〈座標間由空格隔開〉：2 2
 該座標已被輸入，請玩家B重新輸入XY座標〈座標間由空格隔開〉：2 2
 該座標已被輸入，請玩家B重新輸入XY座標〈座標間由空格隔開〉：2 2

\ / \ #請玩家好好輸入！
) (')
 (>)
 \ (_)

該座標已被輸入，請玩家B重新輸入XY座標〈座標間由空格隔開〉：1 1

	1	2	3
1	X	-	-
2	-	0	-
3	-	-	-

請玩家A輸入XY座標〈座標間由空格隔開〉：1 2

	1	2	3
1	X	0	-
2	-	0	-
3	-	-	-

請玩家B輸入XY座標〈座標間由空格隔開〉：3 2

	1	2	3
1	X	0	-
2	-	0	-
3	-	X	-

請玩家A輸入XY座標〈座標間由空格隔開〉：1 3

	1	2	3
1	X	0	0
2	-	0	-
3	-	X	-

請玩家B輸入XY座標〈座標間由空格隔開〉：3 1

	1	2	3
1	X	0	0
2	-	0	-
3	X	X	-

請玩家A輸入XY座標〈座標間由空格隔開〉：3 3

	1	2	3
1	X	0	0
2	-	0	-
3	X	X	0

請玩家B輸入XY座標〈座標間由空格隔開〉：2 3

	1	2	3
1	X	0	0
2	-	0	X

請玩家B輸入XY座標〈座標間由空格隔開〉：2 3

```
  1 2 3
1 X 0 0
2   0 X
3 X X 0
```

請玩家A輸入XY座標〈座標間由空格隔開〉：2 1

```
  1 2 3
1 X 0 0
2 0 0 X
3 X X 0
```

雙方平手！

```
  /\_/\
 ( o.o )
  > ^ <
 /\_/\
/_____\
loveA love
```

歡迎下次再度遊玩！

```
  /\_/\
 ( o.o )
  > ^ <
 /\_/\
/_____\
loveB love
```

歡迎下次再度遊玩！

現在比分是(A:B)：1:2

要不要再玩一局呢？如果要的話，請輸入yes，如果不要，請輸入no

no

恭喜B獲得最終勝利！！

```
  /\_/\
 ( o.o )
  > ^ <
 /\_/\
/_____\
loveB love
```

歡迎下次再度遊玩！

Process exited after 116.8 seconds with return value 0
請按任意鍵繼續 . . .

圖十五、範例和自訂測資