|  |
| --- |
| 程式設計期末報告 |

|  |
| --- |
| 1082003 張舒涵  2020/6/14 |

目錄

[問題說明 2](#_Toc43065836)

[演算法 2](#_Toc43065837)

[程式碼 3](#_Toc43065838)

[討論 6](#_Toc43065839)

[附錄 7](#_Toc43065840)

# 問題說明

以C++製作雙機對戰黑白棋程式。需使用動態連結檔(.dll)製作，裁判程式會將當前棋盤(A[8][8])及先後手(1或2)傳至學生製作的動態連結檔讀取。

學生製作的程式讀取當前棋盤及先後手後，會判斷我方是1或2，之後會依規則找出可下棋的位置，並回傳此位置給裁判程式。

# 演算法

Step1. 讀取team(包含棋盤A[8][8]及who)

Step2. 若who=1，轉到Step4

Step3. 若who=2，轉到Step4

Step4. 從A[8][8]第一格依序確定能否可下棋(依規則)

Step5. 若此格能下棋，回傳此格位置(x、y)

# 程式碼

流程圖：

程式結束

回傳此位置

否

是

依序判斷位置是否符合下棋規則

2

1

Who是1或2

讀取team

程式開始

是

否

程式結束

回傳此位置

依序判斷位置是否符合下棋規則

假碼：

if who=1 then

for i (0 to 7) do

for j (0 to 7) do

for k (2 to 7) do

for h (1 to 6) do

if A[i][j] =0 and A[i][j+k] =1 and A[i][j+(k-h)] =2 and A[i][j+1] =2 and A[i][j+(k-1)] =2 then

p.x🡨i and p.y🡨j return p

end if

else if A[i][j] =0 and A[i+k][j] =1 and A[i+(k-h)][j] =2 and A[i+1][j] =2 and A[i+(k-1)][j] =2 then

p.x🡨i and p.y🡨j return p

end if

else if A[i][j] =0 and A[i][j-k] =1 and A[i][j-(k-h)] =2 and A[i][j-1] =2 and A[i][j-(k+1)] =2 then

p.x🡨i and p.y🡨j return p

end if

else if A[i][j] =0 and A[i-k][j] =1 and A[i-(k-h)][j] =2 and A[i-1][j] =2 and A[i-(k+1)][j] =2 then

p.x🡨i and p.y🡨j return p

end if

else if A[i][j] =0 and A[i+k][j+k] =1 and A[i+(k-h)][j+(k-h)] =2 and A[i+1][j+1] =2 and A[i+(k+1)][j+(k+1)] =2 then

p.x🡨i and p.y🡨j return p

end if

else if A[i][j] =0 and A[i+k][j-k] =1 and A[i+(k-h)][j-(k-h)] =2 and A[i+1][j-1] =2 and A[i+(k+1)][j-(k-1)] =2 then

p.x🡨i and p.y🡨j return p

end if

else if A[i][j] =0 and A[i-k][j+k] =1 and A[i-(k-h)][j+(k-h)] =2 and A[i-1][j+1] =2 and A[i-(k-1)][j+(k+1)] =2 then

p.x🡨i and p.y🡨j return p

end if

else if A[i][j] =0 and A[i-k][j-k] =1 and A[i-(k-h)][j-(k-h)] =2 and A[i-1][j-1] =2 and A[i-(k-1)][j-(k-1)] =2 then

p.x🡨i and p.y🡨j return p

end if

end for

end for

end for

end for

end if

else if who=2 then

for i (0 to 7) do

for j (0 to 7) do

for k (2 to 7) do

for h (1 to 6) do

if A[i][j] =0 and A[i][j+k] =2 and A[i][j+(k-h)] =1 and A[i][j+1] =1 and A[i][j+(k-1)] =1 then

p.x🡨i and p.y🡨j return p

end if

else if A[i][j] =0 and A[i+k][j] =2 and A[i+(k-h)][j] =1 and A[i+1][j] =1 and A[i+(k-1)][j] =1 then

p.x🡨i and p.y🡨j return p

end if

else if A[i][j] =0 and A[i][j-k] =2 and A[i][j-(k-h)] =1 and A[i][j-1] =1 and A[i][j-(k+1)] =1 then

p.x🡨i and p.y🡨j return p

end if

else if A[i][j] =0 and A[i-k][j] =2 and A[i-(k-h)][j] =1 and A[i-1][j] =1 and A[i-(k+1)][j] =1 then

p.x🡨i and p.y🡨j return p

end if

else if A[i][j] =0 and A[i+k][j+k] =2 and A[i+(k-h)][j+(k-h)] =1 and A[i+1][j+1] =1 and A[i+(k+1)][j+(k+1)] =1 then

p.x🡨i and p.y🡨j return p

end if

else if A[i][j] =0 and A[i+k][j-k] =2 and A[i+(k-h)][j-(k-h)] =1 and A[i+1][j-1] =1 and A[i+(k+1)][j-(k-1)] =1 then

p.x🡨i and p.y🡨j return p

end if

else if A[i][j] =0 and A[i-k][j+k] =2 and A[i-(k-h)][j+(k-h)] =1 and A[i-1][j+1] =1 and A[i-(k-1)][j+(k+1)] =1 then

p.x🡨i and p.y🡨j return p

end if

else if A[i][j] =0 and A[i-k][j-k] =2 and A[i-(k-h)][j-(k-h)] =1 and A[i-1][j-1] =1 and A[i-(k-1)][j-(k-1)] =1 then

p.x🡨i and p.y🡨j return p

end if

end for

end for

end for

end for

end if

# 討論

此程式使用了動態連結檔(.dll)，這種平常較少接觸存檔方式，因此為了確保檔案能被讀檔成功，還需例外寫一個裁判程式做測試。

因為之前並沒有使用過動態連結檔過，所以有尋找了動態連結檔如何開檔。

決賽的程式有針對預賽程式所有的邏輯bug，作改進加強。

# 附錄

完整程式碼：

#include <iostream>

#include <iomanip>

using namespace std;

struct team

{

int who;

int A[8][8];

};

struct point

{

int x;

int y;

};

extern "C"

{

\_\_declspec(dllexport) struct point play(struct team t)

{

int i, j;

int k, h;

struct point p;

if (t.who == 1)

{

for (i = 0; i < 8; i++)

for (j = 0; j < 8; j++)

for (k = 2; k < 8; k++)

for (h = 1; h < 7; h++)

{

if ((t.A[i][j + k] == 1) && (j + k < 8) && (t.A[i][j + (k - h)] == 2) && (t.A[i][j] == 0) && (t.A[i][j + 1] == 2) && (k - h > 0) && (t.A[i][j + (k - 1)] == 2) && (k - 1 < k))

{

p.x = i;

p.y = j;

return p;

}

else if ((t.A[i + k][j] == 1) && (i + k < 8) && (t.A[i + (k - h)][j] == 2) && (t.A[i][j] == 0) && (t.A[i + 1][j] == 2) && (k - h > 0) && (t.A[i + (k - 1)][j] == 2) && (k - 1 < k))

{

p.x = i;

p.y = j;

return p;

}

else if ((t.A[i][j - k] == 1) && (j - k > 0) && (t.A[i][j - (k - h)] == 2) && (t.A[i][j] == 0) && (t.A[i][j - 1] == 2) && (k - h > 0) && (t.A[i][j - (k + 1)] == 2) && (k + 1 > k))

{

p.x = i;

p.y = j;

return p;

}

else if ((t.A[i - k][j] == 1) && (i - k > 0) && (t.A[i - (k - h)][j] == 2) && (t.A[i][j] == 0) && (t.A[i - 1][j] == 2) && (k - h > 0) && (t.A[i - (k + 1)][j] == 2) && (k + 1 > k))

{

p.x = i;

p.y = j;

return p;

}

else if ((t.A[i - k][j + k] == 1) && (i - k > 0) && (j + k < 8) && (t.A[i - (k - h)][j + (k - h)] == 2) && (t.A[i][j] == 0) && (t.A[i - 1][j + 1] == 2) && (k - h > 0) && (t.A[i - (k - 1)][j + (k + 1)] == 2))

{

p.x = i;

p.y = j;

return p;

}

else if ((t.A[i - k][j - k] == 1) && (i - k > 0) && (j - k > 0) && (t.A[i - (k - h)][j - (k - h)] == 2) && (t.A[i][j] == 0) && (t.A[i - 1][j - 1] == 2) && (k - h > 0) && (t.A[i - (k - 1)][j - (k - 1)] == 2))

{

p.x = i;

p.y = j;

return p;

}

else if ((t.A[i + k][j + k] == 1) && (i + k < 8) && (j + k < 8) && (t.A[i + (k - h)][j + (k - h)] == 2) && (t.A[i][j] == 0) && (t.A[i + 1][j + 1] == 2) && (k - h > 0) && (t.A[i + (k + 1)][j + (k + 1)] == 2))

{

p.x = i;

p.y = j;

return p;

}

else if ((t.A[i + k][j - k] == 1) && (i + k < 8) && (j - k > 0) && (t.A[i + (k - h)][j - (k - h)] == 2) && (t.A[i][j] == 0) && (t.A[i + 1][j - 1] == 2) && (k - h > 0) && (t.A[i + (k + 1)][j - (k - 1)] == 2))

{

p.x = i;

p.y = j;

return p;

}

}

}

else if (t.who == 2)

{

for (i = 0; i < 8; i++)

for (j = 0; j < 8; j++)

for (k = 2; k < 8; k++)

for (h = 1; h < 7; h++)

{

if ((t.A[i][j + k] == 2) && (j + k < 8) && (t.A[i][j + (k - h)] == 1) && (t.A[i][j] == 0) && (t.A[i][j + 1] == 1) && (k - h > 0) && (t.A[i][j + (k - 1)] == 1) && (k - 1 < k))

{

p.x = i;

p.y = j;

return p;

}

else if ((t.A[i + k][j] == 2) && (i + k < 8) && (t.A[i + (k - h)][j] == 1) && (t.A[i][j] == 0) && (t.A[i + 1][j] == 1) && (k - h > 0) && (t.A[i + (k - 1)][j] == 1) && (k - 1 < k))

{

p.x = i;

p.y = j;

return p;

}

else if ((t.A[i][j - k] == 2) && (j - k > 0) && (t.A[i][j - (k - h)] == 1) && (t.A[i][j] == 0) && (t.A[i][j - 1] == 1) && (k - h > 0) && (t.A[i][j - (k + 1)] == 1) && (k + 1 > k))

{

p.x = i;

p.y = j;

return p;

}

else if ((t.A[i - k][j] == 2) && (i - k > 0) && (t.A[i - (k - h)][j] == 1) && (t.A[i][j] == 0) && (t.A[i - 1][j] == 1) && (k - h > 0) && (t.A[i - (k + 1)][j] == 1) && (k + 1 > k))

{

p.x = i;

p.y = j;

return p;

}

else if ((t.A[i - k][j + k] == 2) && (i - k > 0) && (j + k < 8) && (t.A[i - (k - h)][j + (k - h)] == 1) && (t.A[i][j] == 0) && (t.A[i - 1][j + 1] == 1) && (k - h > 0) && (t.A[i - (k - 1)][j + (k + 1)] == 1))

{

p.x = i;

p.y = j;

return p;

}

else if ((t.A[i - k][j - k] == 2) && (i - k > 0) && (j - k > 0) && (t.A[i - (k - h)][j - (k - h)] == 1) && (t.A[i][j] == 0) && (t.A[i - 1][j - 1] == 1) && (k - h > 0) && (t.A[i - (k - 1)][j - (k - 1)] == 1))

{

p.x = i;

p.y = j;

return p;

}

else if ((t.A[i + k][j + k] == 2) && (i + k < 8) && (j + k < 8) && (t.A[i + (k - h)][j + (k - h)] == 1) && (t.A[i][j] == 0) && (t.A[i + 1][j + 1] == 1) && (k - h > 0) && (t.A[i + (k + 1)][j + (k + 1)] == 1))

{

p.x = i;

p.y = j;

return p;

}

else if ((t.A[i + k][j - k] == 2) && (i + k < 8) && (j - k > 0) && (t.A[i + (k - h)][j - (k - h)] == 1) && (t.A[i][j] == 0) && (t.A[i + 1][j - 1] == 1) && (k - h > 0) && (t.A[i + (k + 1)][j - (k - 1)] == 1))

{

p.x = i;

p.y = j;

return p;

}

}

}

else

{

cout << "t.who的值給錯了" << endl;

p.x = 8;

p.y = 8;

return p;

}

}

}