

**ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH**



**LẬP TRÌNH NÂNG CAO (MỞ RỘNG)
TRÒ CHƠI Ô ĂN QUAN**

SVTH: Trần Công Minh Quân MSSV: 2012528

GVHD: Trương Tuấn Anh

Tp. HCM, 06/2022

MỤC LỤC

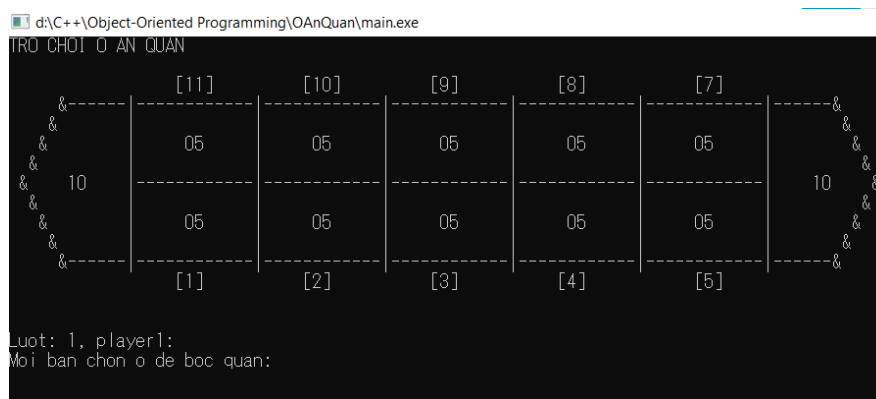
CHƯƠNG 1. MÔ TẢ THIẾT KẾ	1
1.1. Giao diện thiết kế.....	1
1.2. Phương thức chơi.....	1
1.3. Các giá trị quan trọng	1
CHƯƠNG 2. GIẢI THUẬT THỰC HIỆN	3
2.1. Giải thuật OAnQuan.....	3
2.2. Giải thuật moves.....	3
2.3. Giải thuật checkEmptySqr.....	3
2.4. Giải thuật checkPlayer.....	4
2.5. Các giải thuật di chuyển quân	5
2.6. Các giải thuật ăn quân	6
2.7. Giải thuật checkEndGame	7
2.8. Giải thuật summarise	7
CHƯƠNG 3. HƯỚNG DẪN CHƠI	8
3.1. Luật chơi.....	8
3.1.1. Bàn cờ ban đầu	8
3.1.2. Di chuyển quân.....	8
3.1.3. Ăn quân	9
3.1.4. Rải quân đầu lượt	10
3.1.5. Kết thúc ván cờ.....	10
3.2. Các thao tác cơ bản.....	11
3.2.1. Chọn ô để bốc quân	11
3.2.2. Chọn hướng để rải quân	11

CHƯƠNG 1.

MÔ TẢ THIẾT KẾ

1.1. Giao diện thiết kế

Giao diện bàn cờ ô quan được thiết kế dưới dạng text, gồm 10 ô quân và 2 ô quan:



Hình 1.1: Giao diện bàn cờ của chương trình

Giá trị nằm giữa mỗi ô biểu thị số lượng quân trong ô đó. Ban đầu, mỗi ô quân chứa 5 quân, mỗi ô quan chứa 1 quân có giá trị tương đương với 10 quân. Hai người chơi có tên mặc định là Player0 và Player1, được đặt ở hai phía của bàn cờ. Mỗi người chơi quản lý 5 ô quân ở nửa phần bàn cờ tương ứng của mình. Trong chương trình này, quy ước Player0 quản lý các ô có chỉ số từ 1 đến 5, Player1 quản lý các ô có chỉ số từ 7 đến 11.

1.2. Phương thức chơi

Trò chơi diễn ra theo lượt. Khi đến lượt của mình, người chơi nhập chỉ số của ô cần bốc, và hướng rải. Chương trình tính toán và in ra bàn cờ ở cuối mỗi lượt chơi.

1.3. Các giá trị quan trọng

Trong quá trình chơi, người chơi cần để ý đến số lượng quân trên bàn cờ, điểm số, và lượt chơi hiện tại thuộc về ai. Chương trình sử dụng các biến toàn cục sau để lưu các giá trị quan trọng:

- Một mảng `sqr` gồm 12 phần tử được dùng để lưu giá trị của 12 ô trên bàn cờ. Các phần tử `sqr[0]` và `sqr[6]` lưu giá trị của 2 ô quan, các phần tử còn lại lưu giá trị của ô quân có chỉ số tương ứng.

- Một mảng `player` gồm 2 phần tử được dùng để lưu điểm ở trong kho của 2 người chơi: `player[0]` là điểm của Player0, và `player[1]` là điểm của Player1.

- Một biến flag mang hai giá trị 0 và 1. Biến flag được dùng để xác định lượt chơi hiện tại thuộc về người chơi nào: $\text{flag} = 0$ tức là lượt chơi của Player0, và $\text{flag} = 1$ tức là lượt chơi của Player1.

Mảng `sqr`, mảng `player` và biến `flag` sẽ được sử dụng thường xuyên trong các giải thuật của chương trình. Các giải thuật sẽ được mô tả ở chương tiếp theo.

CHƯƠNG 2.

GIẢI THUẬT THỰC HIỆN

2.1. Giải thuật OAnQuan

algorithm OAnQuan

Pre: không có

Post: thông báo kết quả của ván cờ

- 1) **procedure** OAnQuan()
- 2) **repeat**
- 3) moves()
- 4) **until** checkEndGame() = true
- 5) **return** summarise()
- 6) **end procedure**

2.2. Giải thuật moves

algorithm moves

Pre: không có

Post: giá trị của các phần tử *sqr* bị thay đổi sau quá trình bốc và rải

- 1) **procedure** moves()
- 2) checkEmptySqr()
- 3) input *index*
- 4) checkPlayer(*index*)
- 5) input *direction*
- 6) **if** *direction* = 0 **then**
- 7) moveClockwise(*index*)
- 8) **else if** *direction* = 1 **then**
- 9) moveCounterClockwise(*index*)
- 10) **end if**
- 11) **end procedure**

2.3. Giải thuật checkEmptySqr

Trong trường hợp người chơi không còn quân nào ở các ô của mình, giải thuật checkEmptySqr sẽ trích 5 quân từ trong kho của người chơi để chia đều cho 5 ô đó.

algorithm checkEmptySqr

Pre: không có

Post: nếu các phần tử *sqr* tương ứng với *player* đều bằng 0, thực hiện “rải quân” rồi kết thúc giải thuật

- 1) **procedure** checkEmptySqr()
- 2) **if** *flag* = 0 **then**
- 3) **if** *sqr*[1] ≠ 0 **or** *sqr*[2] ≠ 0 **or** *sqr*[3] ≠ 0 **or** *sqr*[4] ≠ 0 **or** *sqr*[5] ≠ 0 **then**
- 4) **end procedure**
- 5) **end if**

```

6)    $sqr[1] \leftarrow 1, sqr[2] \leftarrow 1, sqr[3] \leftarrow 1, sqr[4] \leftarrow 1, sqr[5] \leftarrow 1$ 
7)    $player[0] \leftarrow player[0] - 5$ 
8)   else if  $flag = 1$  then
9)     if  $sqr[7] \neq 0$  or  $sqr[8] \neq 0$  or  $sqr[9] \neq 0$  or  $sqr[10] \neq 0$  or  $sqr[11] \neq 0$  then
10)      end procedure
11)    end if
12)     $sqr[7] \leftarrow 1, sqr[8] \leftarrow 1, sqr[9] \leftarrow 1, sqr[10] \leftarrow 1, sqr[11] \leftarrow 1$ 
13)     $player[1] \leftarrow player[1] - 5$ 
14)  end if
15) end procedure

```

2.4. Giải thuật checkPlayer

Ở đầu lượt chơi, người chơi chỉ được phép bốc quân tại 5 ô thuộc quyền quản lí của mình, nếu người chơi bốc ở các ô khác thì được xem là không hợp lệ. Do đó, giải thuật checkPlayer được dùng để kiểm tra người chơi chọn chỉ số của ô hợp lệ hay không; trong trường hợp không hợp lệ, giải thuật sẽ yêu cầu người chơi nhập lại.

algorithm checkPlayer

Pre: $index$ được input từ hàm moves

Post: kiểm tra nếu $index$ hợp lệ thì kết thúc giải thuật; không thì cập nhật giá trị $index$ rồi tiếp tục kiểm tra lại cho đến khi hợp lệ

```

1) procedure checkPlayer( $index$ )
2)   while true do
3)     if  $index \geq 1$  and  $index \leq 5$  then
4)       if  $flag = 0$  then
5)         end procedure
6)       else
7)         re-input  $index$ 
8)       end if
9)     else if  $index \geq 7$  and  $index \leq 11$  then
10)      if  $flag = 1$  then
11)        end procedure
12)      else
13)        re-input  $index$ 
14)      end if
15)    end if
16)  end while
17) end procedure

```

2.5. Các giải thuật di chuyển quân

algorithm moveClockwise

Pre: *start* là chỉ số của ô được bốc quân

Post: Các quân của ô được chọn sẽ được rải đều theo chiều kim đồng hồ

```

1) procedure moveClockwise(start)
2)   temp  $\leftarrow$  sqr[start]
3)   sqr[start]  $\leftarrow$  0
4)   while temp > 0 do
5)     if start = 0 then
6)       start  $\leftarrow$  11
7)     else
8)       start  $\leftarrow$  start - 1
9)     end if
10)    sqr[start]  $\leftarrow$  sqr[start] + 1
11)    temp  $\leftarrow$  temp - 1
12)  end while
13)  if start = 0 then
14)    next  $\leftarrow$  11
15)  else
16)    next  $\leftarrow$  start - 1
17)  end if
18)  if next = 0 or next = 6 then //TH1
19)    end procedure
20)  else if sqr[next]  $\neq$  0 then //TH2
21)    moveClockwise(next)
22)  else
23)    if next = 0 then
24)      next  $\leftarrow$  11
25)    else
26)      next  $\leftarrow$  next - 1
27)    end if
28)    if sqr[next] = 0 then //TH3a
29)      end procedure
30)    else //TH3b
31)      consumeClockwise(next)
32)    end if
33)  end if
34) end procedure

```

algorithm moveCounterClockwise

Pre: *start* là chỉ số của ô được bốc quân

Post: Các quân của ô được chọn sẽ được rải đều ngược chiều kim đồng hồ

```

1) procedure
   moveCounterClockwise(start)
2) temp  $\leftarrow$  sqr[start]
3) sqr[start]  $\leftarrow$  0
4) while temp > 0 do
5)   if start = 11 then
6)     start  $\leftarrow$  0
7)   else
8)     start  $\leftarrow$  start + 1
9)   end if
10)  sqr[start]  $\leftarrow$  sqr[start] + 1
11)  temp  $\leftarrow$  temp - 1
12) end while
13) if start = 11 then
14)  next  $\leftarrow$  0
15) else
16)  next  $\leftarrow$  start + 1
17) end if
18) if next = 0 or next = 6 then //TH1
19)  end procedure
20) else if sqr[next]  $\neq$  0 then //TH2
21)  moveCounterClockwise(next)
22) else
23)  if next = 11 then
24)    next  $\leftarrow$  0
25)  else
26)    next  $\leftarrow$  next + 1
27)  end if
28)  if sqr[next] = 0 then //TH3a
29)    end procedure
30)  else //TH3b
31)    consumeCounterClockwise(next)
32)  end if
33) end if
34) end procedure

```

2.6. Các giải thuật ăn quân

Bên cạnh việc ăn quân tại ô được chọn, các giải thuật có thể tự gọi lại chính nó trong trường hợp ăn liên hoàn xảy ra.

algorithm consumeClockwise

Pre: *index* là chỉ số của ô bị ăn

Post: tất cả quân ở ô được chọn sẽ bị bỏ vào kho điểm của người chơi; thực hiện ăn liên hoàn theo chiều kim đồng hồ nếu có

- 1) **procedure** consumeClockwise(*index*)
- 2) $player[flag] \leftarrow player[flag] + sqr[index]$
- 3) $sqr[index] \leftarrow 0$
- 4) **if** *index* = 1 **or** *index* = 7 **then**
- 5) **end procedure**
- 6) **else if** *index* = 0 **then**
- 7) **if** $sqr[11] = 0$ **and** $sqr[10] \neq 0$ **then**
- 8) consumeClockwise(10)
- 9) **end if**
- 10) **else if** $sqr[index - 1] = 0$
- 11) **and** $sqr[index - 2] \neq 0$ **then**
- 12) consumeClockwise(*index* - 2)
- 13) **end if**
- 14) **end procedure**

algorithm consumeCounterClockwise

Pre: *index* là chỉ số của ô bị ăn

Post: tất cả quân ở ô được chọn sẽ bị bỏ vào kho điểm của người chơi

- 1) **procedure** consumeCounterClockwise(*index*)
- 2) $player[flag] \leftarrow player[flag] + sqr[index]$
- 3) $sqr[index] \leftarrow 0$
- 4) **if** *index* = 5 **or** *index* = 11 **then**
- 5) **end procedure**
- 6) **else if** *index* = 10 **then**
- 7) **if** $sqr[11] = 0$ **and** $sqr[0] \neq 0$ **then**
- 8) consumeClockwise(0)
- 9) **end if**
- 10) **else if** $sqr[index + 1] = 0$
- 11) **and** $sqr[index + 2] \neq 0$ **then**
- 12) consumeClockwise(*index* + 2)
- 13) **end if**
- 14) **end procedure**

2.7. Giải thuật checkEndGame

algorithm checkEndGame

Pre: không có

Post: nếu hai ô quan đều trống, trả về true; ngược lại chuyển giá trị flag, trả về false

- 1) **procedure** checkEndGame()
- 2) **if** $sqr[0] = 0$ **and** $sqr[6] = 0$ **then**
- 3) **return** true
- 4) **else**
- 5) $flag \leftarrow 1 - flag$
- 6) **return** false
- 7) **end if**
- 8) **end procedure**

2.8. Giải thuật summarise

algorithm summarise

Pre: không có

Post: thông báo kết quả của ván cờ

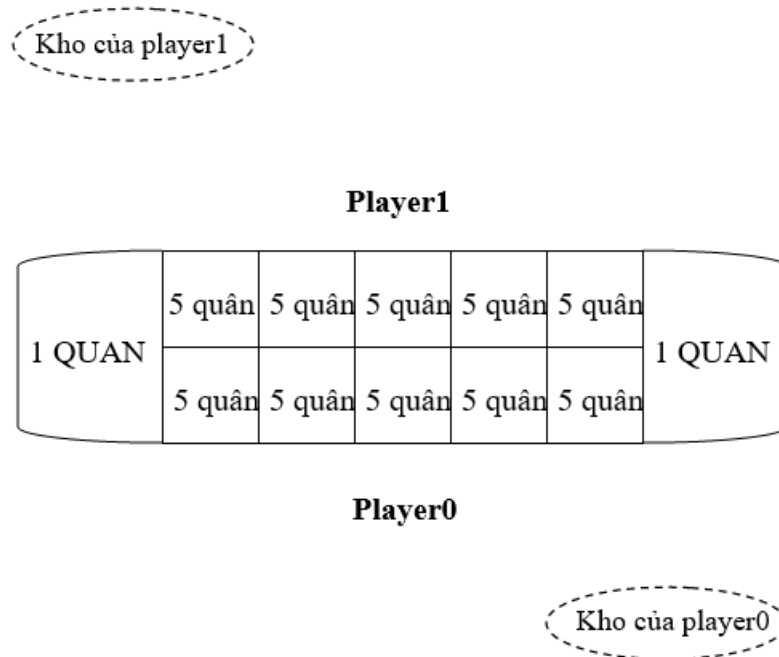
- 1) **procedure** summarise()
- 2) **for** $i \leftarrow 1$ **to** 5 **do**
- 3) $player[0] \leftarrow player[0] + sqr[i]$
- 4) **end for**
- 5) **for** $i \leftarrow 7$ **to** 11 **do**
- 6) $player[1] \leftarrow player[1] + sqr[i]$
- 7) **end for**
- 8) **if** $player[0] > player[1]$ **then**
- 9) **return** “Player0 chiến thắng”
- 10) **else if** $player[0] < player[1]$ **then**
- 11) **return** “Player1 chiến thắng”
- 12) **else**
- 13) **return** “Kết quả hòa”
- 14) **end if**
- 15) **end procedure**

CHƯƠNG 3.

HƯỚNG DẪN CHƠI

3.1. Luật chơi

3.1.1. Bàn cờ ban đầu



Hình 3.1: Bàn cờ ban đầu

3.1.2. Di chuyển quân

Player1 sẽ luôn là người chơi trước, do đó hai người chơi tự thỏa thuận để xác định ai sẽ là Player1 trong ván cờ này.

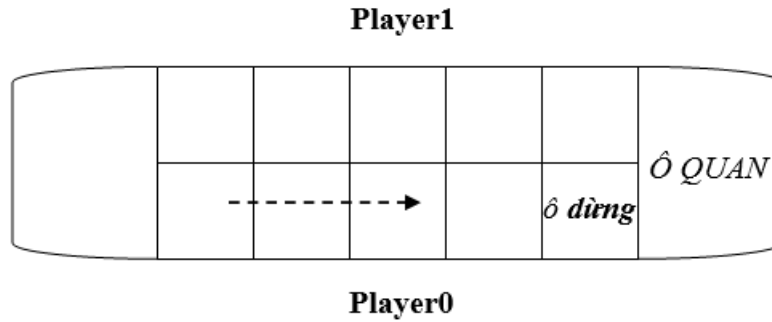
Ở mỗi lượt, người chơi bốc hết tất cả quân ở một ô bất kì phía bên mình để rải theo một trong hai hướng thuận chiều kim đồng hồ hoặc ngược chiều kim đồng hồ. Người chơi tiến hành rải bằng cách lần lượt thả 1 quân xuống các ô kế tiếp ô vừa bốc theo hướng đã chọn, cho đến khi hết quân để thả thì dừng lại. Lúc này người chơi sẽ gặp một trong ba trường hợp sau:

- TH1: Ô kế tiếp của ô vừa dừng là một ô quan.
- TH2: Ô kế tiếp của ô vừa dừng là một ô quân và hiện đang có quân.
- TH3: Ô kế tiếp của ô vừa dừng là một ô quân và ô này trống.

Câu lệnh thứ hai sẽ thiết lập thuộc tính của đồ thị p (được tạo bởi câu lệnh đầu tiên), màu của đường đồ thị sẽ chuyển sang màu đỏ.

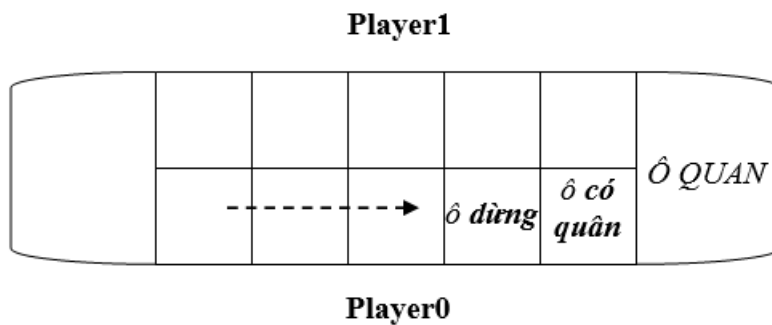
Với ba trường hợp trên, ta có ba cách giải quyết tương ứng:

- Đối với TH1: Người chơi bị mất lượt, quyền đi tiếp thuộc về đối phương.



Hình 3.2: Trường hợp mất lượt

- Đối với TH2: Người chơi tiếp tục bóc quân của ô kế tiếp của ô vừa dừng để tiếp tục thực hiện quá trình rải. Quá trình rải hoàn tất sẽ lại dẫn đến một trong ba trường hợp đã nêu.

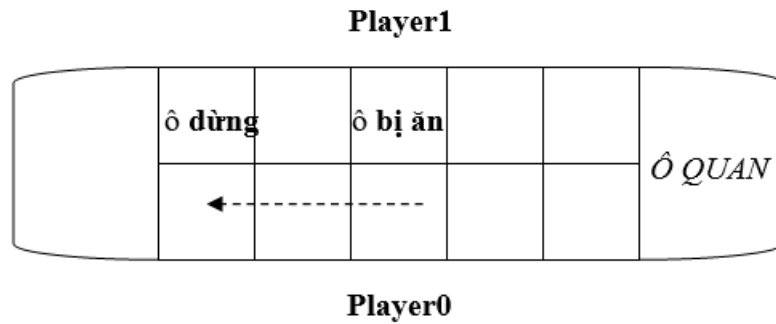


Hình 3.3: Trường hợp đi tiếp

- Đối với TH3: Đây là trường hợp có khả năng được ăn quân, gồm 2 trường hợp nhỏ hơn:
 - a. Nếu ô kế tiếp ô trống cũng là một ô trống thì người chơi cũng mất lượt.
 - b. Nếu ô kế tiếp ô trống là một ô không trống, thì ô này sẽ trở thành ô bị ăn.

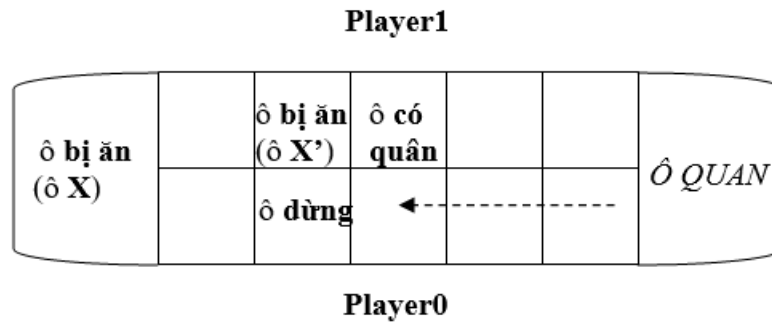
3.1.3. Ăn quân

Khi trường hợp 3b xảy ra, toàn bộ quân và quân trong ô bị ăn sẽ bị bắt đem bỏ vào kho của người chơi.



Hình 3.4: Trường hợp ăn quân

Mở rộng hơn, nếu ô kế tiếp của ô vừa bị ăn (ô **X**) là một ô trống, và ô kế tiếp của ô trống đó là một ô không trống (ô **X'**), thì ô **X'** cũng trở thành ô bị ăn. Do đó, ô bị ăn có thể là một hoặc nhiều ô cách nhau bởi một ô trống. Trường hợp này gọi là ăn liên hoàn.



Hình 3.5: Trường hợp ăn liên hoàn

3.1.4. Rải quân đầu lượt

Rải quân đầu lượt là việc làm của người chơi khi tới lượt chơi của mình, tất cả các ô quân ở phía người chơi đó đều là ô trống. Khi đó, người chơi phải trích ra 5 quân từ kho của mình và cho 1 quân vào mỗi ô quân của mình. Trong trường hợp kho của người chơi không có đủ 5 quân, người chơi vẫn có thể trích 5 quân, tuy nhiên điểm trong kho sẽ bị âm.

3.1.5. Kết thúc ván cờ

Khi cả hai ô quan đều trống, hai người chơi thu quân còn sót lại trên các ô ở phía mình về kho. Tiếp theo, cả hai tiến hành tổng kết cuộc chơi bằng cách tính điểm từ số quan quân trong kho của mình. Mỗi quân có giá trị 1 điểm, mỗi quan có giá trị 10 điểm. Bên nào đạt được nhiều điểm hơn, bên đó là bên thắng cuộc.

3.2. Các thao tác chơi

3.2.1. Chọn ô để bốc quân

Khi tới lượt chơi của mình, người chơi nhập vào chỉ số của ô mà người chơi muốn bốc quân. Chỉ số đó phải tuân theo quy ước của chương trình; nếu chọn ô không hợp lệ, người sẽ được yêu cầu nhập lại.

3.2.2. Chọn hướng để rải quân

Sau khi chương trình yêu cầu người chơi chọn hướng, nhập số 0 để chọn hướng rải theo chiều kim đồng hồ, nhập số 1 để chọn hướng rải ngược chiều kim đồng hồ. Sau đó, màn hình sẽ hiển thị thông báo xác nhận người chơi đã chọn ô nào và hướng đi nào.