BÀI TẬP LỚN MÔN TRÍ TUỆ NHÂN TẠO

Đề tài: Tìm hiểu thuật toán tìm kiếm có đối thủ minimax.

Áp dụng vào xây dựng game cờ tướng

Giáo viên hướng dẫn :TS. Nguyễn Quốc Tuấn

Sinh viên thực hiện:

* Nguyễn Xuân Thương Thùy : Tìm hình ảnh,trang trí bàn cờ,báo cáo
* Trần Hải Quân: Tìm hiểu các nước đi của quân cờ
* Phạm Xuân Quyết : (Tìm Hiểu Về AI, Thuật toán tìm kiếm minimax)
* Ma Quốc Chuyên : Tổng hợp lại,viết code cho game chạy

Phần I: Giới thiệu về game cờ tướng và lịch sử của game cờ tướng

1.1 Giới thiệu về game cờ tướng

Game cờ tướng đã xuất hiện từ rất lâu đời, được nhiều các thế hệ yêu thích bởi việc chơi cờ cũng như việc cần quân đánh trận. Hiện nay, Cờ Tướng trở thành một trò chơi trí tuệ mang tầm cỡ quốc tế. Mỗi ván chơi, người chơi đều phải trải qua ba giai đoạn: Khai cuộc, trung cuộc và tàn cuộc.   
Khai cuộc quyết định xem ván chơi có thuận lợi hay không? Trung cuộc thể hiện khả năng tư duy, chiến thuật trong từng nước đi. Tàn cuộc sẽ quyết định người chơi có nhanh chóng kết thúc ván cờ không?

Đây là trò chơi đối kháng giữa 2 người. Mỗi bên có 5 loại quân chính, bao gồm : Tướng, Xe, Pháo, Mã, Tượng, Sĩ. Mỗi loại quân có số lượng khác nhau, có một chức năng riêng và tiêu diệt quân đối thủ cũng khác nhau:  
  
- Tướng: là quân quan trọng nhất trên bàn cờ, mất tướng cũng đồng nghĩa với ván cờ kết thúc. Tướng có thể đi chéo, thẳng, có thể lùi hoặc tiến trong ô vuông cấm cung.  
  
- Sĩ: là quân bảo vệ cho tướng, chỉ được đi trong cấm cung, chỉ được đi chéo, sức chiến đấu thấp  
  
- Tượng: Cũng là một loại quân yếu chỉ có chức năng phòng thủ trong cấm cung. Quân này đi theo đường chéo của hình vuông gồm 2 ô cờ.  
  
- Mã: Loại quân khá cơ động có thể đi khắp bàn cờ, tuy nhiên đường đi của Mã có thể bị cản bởi quân khác. Mã có đường đi chéo 2 ô liền nhau.  
  
- Xe: Là quân cơ động nhất bàn cờ, Xe có thể đi ngang, đi dọc, tuy nhiên không thể đi chéo. Xe được coi là quân cờ mạnh nhất trong cờ tướng.  
  
- Pháo: Quân pháo cũng khá cơ động, có thể đi ngang dọc khắp bàn cờ, tuy nhiên để tiêu diệt được quân đối phương cần phải bay qua 1 quân cờ khác.  
Bên nào ăn được tướng của đối phương trước, bên đó sẽ giành chiến thắng.

1.2 Lịch sử game cờ tướng  
  
Đây loại cờ có từ khoảng thế kỷ 7. Cờ tướng được bắt nguồn từ Saturanga, một loại cờ cổ được phát minh ở Ấn Độ từ thế kỷ 5 đến thế kỷ 6 (trước cờ tướng khoảng 200 năm). Chính Saturanga được phát minh từ Ấn Độ, sau đó đi về phía tây, trở thành cờ vua và đi về phía Đông trở thành cờ tướng. Người Trung Quốc cũng đã thừa nhận điều này. Cờ  tướng cổ đại không có quân Pháo. Các nhà nghiên cứu đều thống nhất là quân Pháo được bổ sung từ thời nhà Đường (sau năm 618), là quân cờ ra đời muộn nhất trong bàn cờ tướng, bởi cho tới thời đó, con người mới tìm ra vũ khí pháo để sử dụng trong chiến tranh.



Tuy nhiên, người Trung Hoa đã cải tiến bàn cờ Saturanga như sau:  
Họ không dùng “ô”, không dùng hai màu để phân biệt ô, mà họ chuyển sang dùng “đường” để đặt quân và đi quân. Chỉ với động tác này, họ đã tăng thêm số điểm đi quân từ 64 của Saturanga lên 81.  
Đã là hai quốc gia đối kháng thì phải có biên giới rõ ràng, từ đó, họ đặt ra “hà”, tức là sông. Khi “hà” xuất hiện trên bàn cờ, 18 điểm đặt quân nữa được tăng thêm. Như vậy, bàn cờ tướng bây giờ đã là 90 điểm so với 64, đó là một sự mở rộng đáng kể. Tuy nhiên, diện tích chung của bàn cờ hầu như không tăng mấy (chỉ tăng thêm 8 ô) so với số điểm tăng lên tới 1 phần 3.  
Đã là quốc gia thì phải có cung cấm (宮) và không thể đi khắp bàn cờ như kiểu trò chơi Saturanga được. Thế là “Cửu cung” đã được tạo ra. Điều này thể hiện tư duy phương Đông hết sức rõ ràng.  
Bàn cờ  Saturanga có hình dáng quân cờ là những hình khối, nhưng cờ Tướng thì quân nào trông cũng giống quân nào, chỉ có mỗi tên là khác nhau, lại được viết bằng chữ Hán. Đây có thể là lý do khiến cờ tướng không được phổ biến bằng cờ vua, chỉ cần liếc qua là có thể nhận ra đâu là Vua, đâu là Hoàng hậu, kỵ sỹ, v.v. Tuy nhiên, đối với người Trung Hoa thì việc thuộc mặt cờnày là không có vấn đề gì khó khăn. Có lẽ việc cải tiến này cũng một phần là do điều kiện kinh tế bấy giờ chưa sản xuất được bộ cờ có hình khối phức tạp như cờ vua. Cờ  tướng không phải là một trò chơi sang trọng, muốn tạo ra một bàn cờ tướng cực kỳ đơn giản, chỉ cần lấy que vạch xuống nền đất cũng xong, còn cờ vua thì mất công hơn nhiều khi phải tạo ra các ô đen/trắng xen kẽ nhau.  
Gần đây ngày càng có nhiều ý kiến đề nghị cải cách hình dáng các quân cờ tướng và trên thực tế người ta đã đưa những phác thảo của những bộ quân mới bằng hình tượng thay cho chữ viết, nhất là khi cờ tướng được chơi ở những nước không sử dụng tiếng Trung Quốc.  
Với sự thay đổi bố cục bàn cờ, người Trung Hoa đã phải có những điều chỉnh để lấy lại sự cân bằng cho bàn cờ. Đó chính là những ngoại lệ mà người chơi phải tự nhớ. tướng được bắt nguồn từ Saturanga, một loại cờ cổ được phát minh ở Ấn Độ từ thế kỷ 5 đến thế kỷ 6 (trước cờ tướng khoảng 200 năm). Chính Saturanga được phát minh từ Ấn Độ, sau đó đi về phía tây, trở thành cờ vua và đi về phía Đông trở thành cờ tướng. Người Trung Quốc cũng đã thừa nhận điều này.



Đã là hai quốc gia đối kháng thì phải có biên giới rõ ràng, từ đó, họ đặt ra “hà”, tức là sông. Khi “hà” xuất hiện trên bàn cờ, 18 điểm đặt quân nữa được tăng thêm. Như vậy, bàn cờ tướng bây giờ đã là 90 điểm so với 64, đó là một sự mở rộng đáng kể. Tuy nhiên, diện tích chung của bàn cờ hầu như không tăng mấy (chỉ tăng thêm 8 ô) so với số điểm tăng lên tới 1 phần 3.Đã là quốc gia thì phải có cung cấm (宮) và không thể đi khắp bàn vờ như kiểu trò chơi Saturanga được. Thế là “Cửu cung” đã được tạo ra. Điều này thể hiện tư duy phương Đông hết sức rõ ràng.  
Bàn cờ Saturanga có hình dáng quân cờ là những hình khối, nhưng cờ Tướng thì quân nào trông cũng giống quân nào, chỉ có mỗi tên là khác nhau, lại được viết bằng chữ Hán. Đây có thể là lý do khiến cờ tướng không được phổ biến bằng cờ vua, chỉ cần liếc qua là có thể nhận ra đâu là Vua, đâu là Hoàng hậu, kỵ sỹ, v.v. Tuy nhiên, đối với người Trung Hoa thì việc thuộc mặt cờ này là không có vấn đề gì khó khăn. Có lẽ việc cải tiến này cũng một phần là do điều kiện kinh tế bấy giờ chưa sản xuất được bộ cờ có hình khối phức tạp như cờvua. Cờ tướng không phải là một trò chơi sang trọng, muốn tạo ra một bàn cờ tướng cực kỳ đơn giản, chỉ cần lấy que vạch xuống nền đất cũng xong, còn cờ vua thì mất công hơn nhiều khi phải tạo ra các ô đen/trắng xen kẽ nhau.  
Gần đây ngày càng có nhiều ý kiến đề nghị cải cách hình dáng các quân cờ tướng và trên thực tế người ta đã đưa những phác thảo của những bộ quân mới bằng hình tượng thay cho chữ viết, nhất là khi cờ tướng được chơi ở những nước không sử dụng tiếng Trung Quốc.  
Với sự thay đổi bố cục bàn cờ, người Trung Hoa đã phải có những điều chỉnh để lấy lại sự cân bằng cho bàn cờ. Đó chính là những ngoại lệ mà người chơi phải tự nhớ.

Phần 2: GIẢI THUẬT TÌM KIẾM MINIMAX

2.1 Giới thiệu thuật toán minimax

Trong 2 người chơi thì một người gọi là người chơi cực đại (Max) và đối thủ của họ là người chơi cực tiểu (Min).Cả 2 đấu thủ đều cố gắng đi những nước thế nào để điểm tuyệt đối của mình lớn hơn hay cao nhất có thể.Tức là người chơi Max sẽ tìm cách làm điểm của mình cao hơn và làm điểm của đối thủ bớt âm hơn (giảm về trị số) .Trong khi người chơi Min thì ngược lại, sẽ cố gắng làm cho điểm của mình âm hơn và làm cho điểm của đối thủ giảm.

Giải thuật tìm kiếm Min-Max được sử dụng để xác định tất cả những “diễn biến” tiếp theo của trò chơi cho đến tầng được yêu cầu.Điểm số ban đầu được gán cho lá, sau đó bằng cách lượng giá các nước đi, điểm số được gán cho các tầng ở trên qua giải thuật Min Max, thuật giải thực hiện một lát cắt cho trước và tính điểm trên đó.

Ý tưởng cơ bản của thuật giải Min-Max theo đệ quy

Nếu mức đang xét là người chơi cực tiểu thì áp dụng thuật toán Min-Max cho các con của nó.Lưu kết quả là giá trị nhỏ nhất.

Nếu mức đang xét là người chơi cực đại thì áp dụng thuật toán Min-Max cho các con của nó.Lưu kết quả là giá trị lớn nhất.

Nếu mức đang xét là lá (tầng cuối cùng của cây tìm kiếm), tình giá trị tĩnh của thế cờ hiện tại ứng với người chơi ở đó.Ghi nhớ kết quả

Mã 1

MinMax(x)

{ // x là nút muốn tính điểm

If x is a leaf

Return score of x;

Else

If x in a minNode

For allChildren of x : v1,…,vn

Return min {MinMax(v1),…,Min-Max(vn)}

Else

For allChildren of x : v1,…,vn

Return max {Min-Max(v1),…,Min-Max(vn)}

}

Tuy nhiên trên một cây có kích thước lớn thì ta không thể tìm hết tất cả các nút mà ta chỉ giới hạn trong một số tầng của cây và xem như đây là mô phỏng gần đúng của một cây Min-Max (chưa biết) bằng cách gán trọng số cho các lá của nó.Trọng số ở đây là trọng số không còn chính xác tuyệt đối mà là ước lượng.Trọng số nhận được theo cách này gọi là được tính toán với sự giúp đỡ của hàm lượng giá, hàm này được xây dựng vởi người dùng dựa trên sự hiểu biết và kinh nghiệm.

Mã 2

function MinMax (pos, depth): integer;

{

if depth = 0 then //Đạt đến giới hạn

MinMax = Eval (pos) //Tính giá trị thế cờ pos

else

{

Gen (pos); // Sinh ra mọi nước đi từ thế cờ pos

while còn lấy được một nước đi m do

{

pos = Tính thế cờ mới nhờ đi m;

value = MinMax (pos, depth-1); // Tính điểm của pos

}

}

}

Tham số depth – độ sâu tìm kiếm giúp ta biết phải tìm đến đâu, tham số pos cho biết thế cờ hiện tại để từ đó biết cách tính tiếp. Giá trị trả về của hàm chính là điểm của thế cờ pos. Hàm lượng giá Eval sẽ đánh giá được chất lượng của thế cờ pos hiện tại. Các thế cờ con pos' là các thế cờ được tạo ra từ pos bằng cách đi một nước đi hợp lệ x nào đó. Do đó ta phải có các lệnh thực hiện đi quân để đến các thế cờ mới. Để biết từ thế cờ pos có thể đi được những nước nào, ta dùng một thủ tục Gen có tham số là thế cờ cha pos. Thủ tục này sẽ cất các thế cờ con pos' đó vào bộ nhớ (dạng danh sách). Việc tiếp theo là ta lấy từng thế cờ đó ra và áp dụng tiếp thủ tục MinMax cho nó để tính điểm value của nó.

Mã 3

function MinMax (pos, depth): integer;

{

if depth = 0 then

MinMax = Eval (pos) // Tính giá trị thế cờ pos

else

{ best = -INFINITY;

Gen (pos); // Sinh ra mọi nước đi từ thế cờ pos

while còn lấy được một nước đi m

do

{

pos = Tính thế cờ mới nhờ đi m;

value = -Minimax (pos, depth - 1);

if value > best then best = value;

}

MinMax = best; //Trả về giá trị tốt nhất

}

}

Thông thường, bàn cờ được biểu diễn bằng các biến toàn cục. Do đó thay cho truyền tham số là một bàn cờ mới pos vào thủ thục MinMax thì người ta biến đổi luôn biến toàn cục này nhờ thực hiện nước đi "thử" (nước đi dẫn đến bàn cờ mới pos). Sau khi MinMax thực hiện việc tính toán dựa vào bàn cờ lưu ở biến toàn cục thì thuật toán sẽ dùng một số thủ tục để loại bỏ nước đi này. Như vậy MinMax bỏ các tham số pos như sau:

Mã 4

function MinMax (depth): integer;

{

if depth = 0 then MinMax = Eval // Tính thế cờ pos trong biến toàn cục

else

{

best = -INFINITY;

Gen; // Sinh ra mọi nước đi từ thế cờ pos

while còn lấy được một nước đi m do

{

thực hiện nước đi m;

value = -MinMax (depth - 1);

bỏ thực hiện nước đi m;

if value > best then best = value;

}

MinMax = best;

}

}

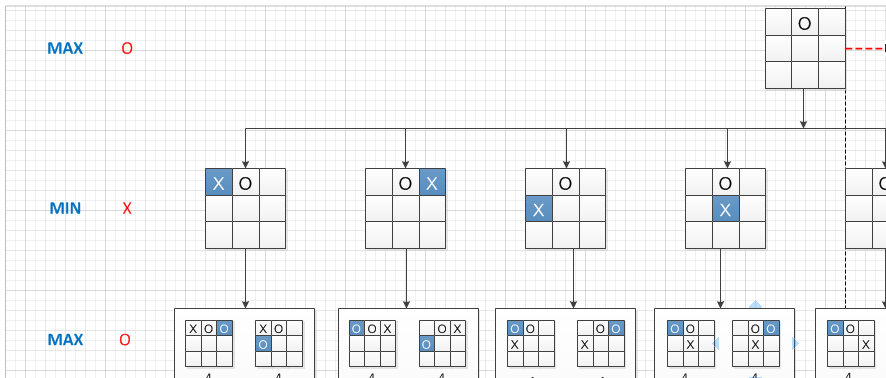
2.2.Đánh giá thuật toán :

Giả sử hệ số nhánh trung bình của cây là a , xét độ sâu b thì số nút ở đáy phải lượng giá là ab .Thực tế số nhánh khá lớn nên chỉ cần xét ở độ sâu nhỏ (cỡ nhỏ hơn 10) thì số nút cần xét cũng đã rất lớn.

|  |  |
| --- | --- |
| Depth | Node Count |
| 0 | 1 |
| 1 | 5 |
| 8 | 390625 |
| … |  |
| n | 5n |

2.3.Cây trò chơi

Các trạng thái bàn cờ khác nhau (hay còn gọi là một thế cờ, tình huống cờ) trong quátrình chơi có thể biểu diễn thành một cây tìm kiếm và ta sẽ tiến hành tìm kiếm trên cây để tìm được nước đi tốt nhất. Cây trò chơi có các nútcủa cây là các tình huống khác nhau của bàn cờ, các nhánh nối giữa các nút sẽ cho biết từmột tình huống bàn cờ chuyển sang tình huống khác thông qua chỉ một nước đi đơn nàođó. Dĩ nhiên, các nước đi này diễn ra theo cặp do hai đấu thủ lần lượt tiến hành. Độ sâucủa cây trò chơi ply là số tầng của cây (chính là độ sâu d của cây). Thuật ngữ “nước đi”trong sách được thống nhất chỉ bao gồm một lần đi của một đấu thủ hoặc một lần đi phảnứng lại của đối thủ bên kia. Chú ý nó khác với thói quen dùng trong thực tế một nước đi bao gồm lần đi của ta và một lần đi của đối thủ. Nói cách khác, nước đi ở đây thực chấtchỉ là "nửa nước" theo cách hiểu của làng cờ.

Ví dụ

2.4. Chiến lược tìm kiếm trong trò chơi

Một chiến lược thường được cả người lẫn máy dùng là phân tích thế cờ chỉ sau một số nước đi nào đó của cả hai bên. Sau khi "nhìn xa" xem bàn cờ có những khả năng biến đổinhư thế nào sau một số nước, ta sẽ đánh giá độ xấu tốt của các thế cờ nhận được. Tiếptheo, ta sẽ chọn nước đi sẽ dẫn tới một thế cờ tốt nhất trong số đó có cân nhắc đến cách đicủa cả hai bên. Với máy, thế cờ này được đánh giá là tốt hơn thế cờ kia nhờ so sánh điểmcủa thế đó do bộ lượng giá trả lại. Chúng ta chỉ có khả năng xét trước một số hữu hạn cácnước (ví dụ đại kiện tướng chơi cờ vua có thể xét trước 8-10 nước đi, người thường chỉ 2-4 nước đi). Rõ ràng là nếu xét càng sâu thì chơi càng giỏi. Nhưng không thể thực hiệnđiều này với độ sâu quá lớn được do số nút ở độ sâu đó có thể trở nên lớn khủng khiếp vàkhông đủ thời gian để phân tích. Nếu dừng ở một độ sâu hợp lý thì bộ phân tích có thểhoàn thành việc tính toán trong một thời gian hạn định.

2.5. Thủ tục Minimax

Giả sử chúng ta có một bộ phân tích thế cờ có thể áp dụng tất cả các luật, các phương pháp đánh cờ khác nhau vào từng thế cờ và chuyển đổi chúng thành một con số đại diện(cho điểm thế cờ). Mặt khác, ta giả sử con số đó là dương khi áp dụng cho thế cờ của mộtđấu thủ (được gọi là

người chơi cực đại

- maximizer), và là âm khi áp dụng cho đấu thủ bên kia (được gọi là

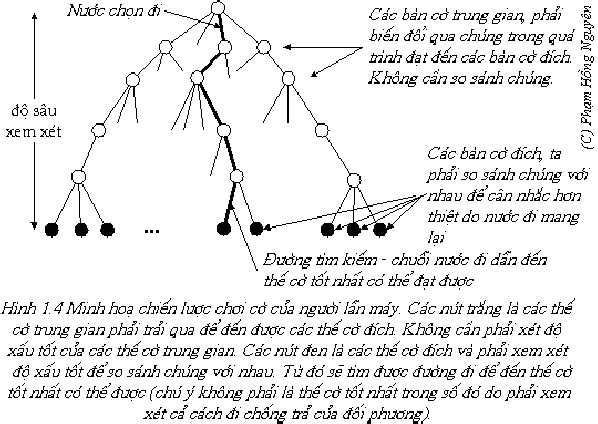
người chơi cực tiểu

- minimizer). Quá trình tính toán cho điểm thế

cờ được gọi là

lượng giá tĩnh

(static evaluation). Hàm thực hiện việc tính toán được gọilà một bộ lượng giá tĩnh, và giá trị nhận được gọi là điểm lượng giá tĩnh. Cả hai đấu thủđều cố gắng đi như thế nào đó để đạt được điểm tuyệt đối lớn nhất. Người chơi cực đại sẽtìm những nước đi dẫn đến điểm của mình trở nên lớn hơn (hay cao nhất có thể được)hay điểm của đối thủ bớt âm hơn (nhỏ hơn về giá trị tuyệt đối). Còn đấu thủ của anh ta,người chơi cực tiểu, lại ra sức phản kháng lại, để dẫn tới điểm âm của anh ta âm hơn hayđiểm dương của đối thủ nhỏ đi



Ví dụ một phần cây trò chơi trong hình 1.5



Người chơi cực đại hi vọng chọn nước đi bên phải để đạt được điểm 8. Thế nhưng nếu đinhư vậy thì khi đến lượt đi của người chơi cực tiểu, anh ta sẽ cố gắng không cho ngườichơi cực đại đạt được điểm này bằng cách chọn nước đi nhánh bên trái và như vậy, ngườichơi cực đại chỉ được có 1 điểm thay vì 8. Ngược lại, nếu người chơi cực đại chọn nướcđi bên trái, thì trong tình huống xấu nhất anh ta vẫn còn được 2 điểm, lớn hơn là chọnnước đi bên phải. Nói chung, người chơi cực đại sẽ phải tìm cách nhận ra các nước đi củađối phương tiếp theo làm cho điểm giảm xuống. Và tương tự như vậy, người chơi cựctiểu phải nhận biết được nước đi của người chơi cực đại cố gắng làm tăng điểm lên. Thủtục tìm nước đi tốt nhất trên cây trò chơi như trên được gọi là thủ tục Minimax do điểm ở mỗi nút có thể là điểm cực đại hoặc có thể là điểm cực tiểu và có thuật toán như sau:

.

2.6.Thuật toán Minimax

 Nếu như đạt đến giới hạn tìm kiếm (đến tầng dưới cùng của cây tìmkiếm), tính giá trị tĩnh của thế cờ hiện tại ứng với người chơi ở đó.Ghi nhớ kết quả

 Nếu như mức đang xét là của người chơi cực tiểu, áp dụng thủ tụcMinimax này cho các con của nó. Ghi nhớ kết quả nhỏ nhất

 Nếu như mức đang xét là của người chơi cực đại, áp dụng thủ tụcMinimax này cho các con của nó. Ghi nhớ kết quả lớn nhất.

Viết chương trình cho thuật toán Minimax

Bây giờ, ta thử dựa vào phát biểu trên để viết chương trình cho thuật toán này bằng ngônngữ tựa Pascal. Đây là một hàm có tên là Minimax và sẽ là loại đệ qui. Trước hết, để hàmnày biết đã đạt đến giới hạn tìm kiếm chưa, ta cần cung cấp cho nó một tham số về độ sâu tìm kiếm

depth

(để biết phải tìm đến đâu), đồng thời ta cũng phải cho biết thế cờ hiện tại

 pos

để nó từ đó nó biết cách tính tiếp. Giá trị trả về của hàm chính là điểm của thế cờ (bàn cờ)

 pos

Vậy hàm sẽ có khai báo dạng:

function

Minimax (pos, depth): integer;Mỗi khi Minimax được gọi, nó sẽ càng gần đến giới hạn tìm kiếm, do đó ta sẽ gọi hàmnày với độ sâu bằng độ sâu cũ trừ đi một. Đạt đến độ sâu giới hạn chính là khi

depth= 0.

Khi đạt độ sâu này ta sẽ gọi hàm lượng giá

 Eval

để đánh giá chất lượng của thế cờ

 pos

hiện tại (thực hiện điều một của thuật toán). Như vậy bước đầu hàm này có dạng sau:

function

Minimax (pos, depth): integer;

begin

if

depth = 0

then

{ Đã đạt đến giới hạn }

Minimax := Eval (pos)

{ Tính giá trị thế cờ pos }

 else begin

...Minimax (pos, depth - 1);

{ Gọi đệ qui với độ sâu giản dần}

...

End

;

end

;

Ở trên, Minimax được gọi với độ sâu giảm đi một. Đó là độ sâu của các thế cờ là con.Các thế cờ con

 Pos’

đó là các thế cờ được tạo ra từ

 pos

bằng cách đi một nước đi hợp lệm nào đó

Do đó ta phải có các lệnh thực hiện đi quân để đến các thế cờ mới. Để biết từ thếcờ

 pos

có thể đi được những nước nào, ta dùng một thủ tục

Gen

có tham số là thế cờ cha

 pos.

Thủ tục này sẽ cất các thế cờ con

 pos'

đó vào bộ nhớ (dạng danh sách). Việc tiếptheo là ta lấy từng thế cờ đó ra và áp dụng tiếp thủ tục Minimax cho nó để tính điểm value của nó.

\*Vậy hàm Minimax bây giờ có dạng:

function

Minimax (pos, depth): integer;

begin

if

depth = 0

then

 Minimax := Eval (pos)

{ Tính giá trị thế cờ pos }

else begin

 Gen (pos);

{ Sinh ra mọi nước đi từ thế cờ pos }

while

còn lấy được một nước đi m

dobegin

 pos := Tính thế cờ mới nhờ đi m;value := Minimax (pos, depth-1);

{ Tính điểm của pos }

...

end

;...

end

;

End

;

Theo phát biểu của thuật toán, ta thấy các điều 2 và 3 chỉ khác nhau ở cách chọn kết quả tốt nhất

best

phụ thuộc vào người chơi đang là người chơi cực đại hay cực tiểu. Cuốicùng thuật toán sẽ trả về điểm tốt nhất đạt được.

\*Vậy hàm này được phát triển tiếp thành:

function

Minimax (pos, depth): integer;

begin

if

depth = 0

then

 Minimax := Eval (pos)

{ Tính giá trị thế cờ pos }

else begin

Gen (pos);

{ Sinh ra mọi nước đi từ thế cờ pos }

while

còn lấy được một nước đi m

do

begin

pos := Tính thế cờ mới nhờ đi m;value := Minimax (pos, depth-1);

{ Tính điểm của pos }{ Chọn điểm tốt nhất tuỳ thuộc theo người chơi }

if

người chơi là người cực đại

thenbeginif

best < value

then

best := value;

end

else begin

if

best > value

then

best := value;

end

end

;

Minimax := best;

{ Trả về giá trị tốt nhất }

end

;

End

;

Thông thường để cho tiện (và cũng rất gần sự thực) ta coi cả hai người chơi (hai bên) cócùng cách đánh giá về một thế cờ. Có điều thế cờ này là tốt với một người thì phải đượcđánh giá là tồi với người kia và ngược lại. Trong máy tính cách thể hiện tốt nhất là ta chođiểm một thế cờ có thêm dấu âm dương: dấu âm dành cho người chơi cực đại và dấu âmcho người chơi cực tiểu. Với người chơi cực đại sẽ mong muốn điểm này càng dươngcàng tốt, còn người chơi cực tiểu lại mong muốn điểm này càng âm càng tốt. Do đó để dễxử lí ta sẽ tuỳ theo mức người chơi mà đổi dấu giá trị đánh giá thế cờ

 Pos.

Chú ý rằng,thay đổi độ sâu là chuyển sang đối phương nên phải đổi dấu. Chương trình thực hiện đổidấu như sau:value := -Minimax (pos, depth-1);

{ Tính điểm của pos }

 Cũng do dùng cùng hàm lượng giá nên khi đến lượt người chơi cực đại và cực tiểu cócùng cái nhìn như nhau về một thế cờ. Điều này dẫn đến có thể dùng cùng cách chọnnước đi tốt nhất cho họ (gộp được điều 2 và 3 lại với nhau được). Giá trị best cần đượckhởi đầu rất nhỏ để đảm bảo không vượt mọi giá trị value, tốt nhất là giá trị -vô cùng:

function

Minimax (pos, depth): integer;

begin

if

depth = 0

then

 Minimax := Eval (pos)

{ Tính giá trị thế cờ pos }

else begin

  best := -INFINITY;Gen (pos);

{ Sinh ra mọi nước đi từ thế cờ pos }

while

còn lấy được một nước đi m

do

begin

 pos := Tính thế cờ mới nhờ đi m;value := -Minimax (pos, depth - 1);

if

value > best

then

best := value;

end

;

Minimax := best;

end

;

end

;

Thông thường, bàn cờ được biểu diễn bằng các biến toàn cục. Do đó thay cho truyềntham số là một bàn cờ mới pos vào thủ thục Minimax thì người ta biến đổi luôn biến toàncục này nhờ thực hiện nước đi "thử" (nước đi dẫn đến bàn cờ mới

 pos).

Sau khi Minimaxthực hiện việc tính toán dựa vào bàn cờ lưu ở biến toàn cục thì thuật toán sẽ dùng một sốthủ tục để loại bỏ nước đi này. Như vậy Minimax bỏ các tham số

 pos

như sau:

function

Minimax (depth): integer;

begin

if

depth = 0

then

Minimax := Eval

{ Tính thế cờ pos trong biến toàn cục }

Nếu nước đi m;value := -Minimax (depth - 1);bỏ thực hiện nước đi m;

if

value > best

then

best := value;

end

;

Minimax := best;

end

;

End

;

Thuật toán Minimax với việc đảo dấu mỗi khi thay đổi độ sâu như trên đôi khi được gọi là thuật toán Negamax.

Đánh giá thuật toán Minimax

 Nếu hệ số nhánh trung bình của cây là b và ta thực hiện tìm kiếm đến độ sâu d

thì số nút phải lượng giá ở đáy cây như ta đã biết là bd

. Đây chính là số đo độ phức tạp của thuật toán. Nếu b = 40, d = 4

(các con số thường gặp trong trò chơi cờ) thì số nút phải lượng giá là 404 = 2560000 (trên 2 triệu rưỡi nút). Còn với b = 40,d  = 5 thì số nút phải lượng giá sẽ tăng 40 lần nữa thành 405 = 102400000 (trên 102 triệu nút).Lưu ý: toàn bộ ý tưởng của thuật toán này là dựa trên việc chuyển đổi mỗi thế cờ thànhmột con số để đánh giá. Rất tiếc là các con số này thường không tốt và không đủ để đánhgiá hết mọi điều. Mặt khác, thuật toán này có thể rất tốn kém (chạy chậm) do việc sinhcác nước đi và lượng giá rất tốn thời gian tính toán, do vậy độ sâu của cây trò chơi cũng bị hạn chế nhiều. Ta cần có thêm những cải tiến để cải thiện tình hình.

Phần III :Viết chương trình:

Ngôn ngữ: Html,css,js

Phần HTML:

index.html :

<!DOCTYPE HTML>

<html>

<head>

<meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>

<title>Cờ tướng AI</title>

<link href="css/chess.css" rel="stylesheet" type="text/css">

</head>

<body>

<div class="box" id="box">

</br>

    <div class="chess\_left">

        <canvas id="chess">Xin lỗi, trình duyệt của bạn không hỗ trợ HTML5, vui lòng nâng cấp trình duyệt của bạn lên IE9, Firefox hoặc Google Chrome!</canvas>

        <audio src="audio/click.wav" id="clickAudio" preload="auto"></audio>

        <!--<audio src="audio/check.wav" id="checkAudio" preload="auto"></audio>-->

        <audio src="audio/select.wav" id="selectAudio" preload="auto"></audio>

        <div>

            <div class="bn\_box" id="bnBox">

                <input type="button" name="offensivePlay" id="tyroPlay" value="Dễ" class="btn btn-warning"/>

                <input type="button" name="offensivePlay" id="superPlay" value="Khó" class="btn btn-warning"/>

                <input type="button" name="regret" id="regretBn" value="Lùi Bước Đánh" class="btn btn-warning"/>

                <input type="button" name="billBn" id="billBn" value="棋谱" class="bn\_box" class="btn btn-warning"/>

                <input type="button" name="stypeBn" id="stypeBn" value="+/-" class="btn btn-warning"/>

            </div>

        </div>

    </div>

    <div class="chess\_right" id="chessRight">

        <select name="billList" id="billList">

        </select>

        <ol id="billBox" class="bill\_box">

        </ol>

    </div>

    <div id="moveInfo" class="move\_info"></div>

</div>

<script type="text/javascript" src="js/common.js"></script>

<script type="text/javascript" src="js/play.js"></script>

<script type="text/javascript" src="js/AI.js"></script>

<script type="text/javascript" src="js/bill.js"></script>

<script type="text/javascript" src="js/gambit.js"></script>

</div>

</body>

</html>

Phần css trang trí màu :

chess.css :

body{  font-size: 12px; line-height: 150%; }

.box { margin:0 auto; width:460px; position: relative; }

.chess\_left {float:left; text-align:center }

.chess\_right {  float:left; display:none}

.move\_info {  float:left; margin-top:20px; color: #fff;}

.bill\_box{ height: 320px; width: 80px; overflow:auto;}

.bill\_box li { cursor:pointer; text-align:left}

.bill\_box li:hover { cursor:pointer; background: #C6A577; }

.bill\_box li:active { cursor:pointer; background: #fff; }

#billList{ margin-top:20px}

.bn\_box{display:none}

Các file ảnh trong mục img:

+Kiểu bàn cờ nhỏ:

A screenshot of a computer screen

Description automatically generated

+Kiểu bàn cờ to:

A screenshot of a computer screen

Description automatically generated

Ảnh các quân cờ và bàn cờ,ảnh nền :

A close up of a logo

Description automatically generatedA close up of a logo

Description automatically generatedA picture containing object, plate, cup

Description automatically generatedA close up of a logo

Description automatically generatedA close up of a logo

Description automatically generatedA close up of a logo

Description automatically generatedA picture containing object, clock

Description automatically generatedA close up of a door

Description automatically generatedDiagram, schematic

Description automatically generatedA picture containing object, gauge, device, clock

Description automatically generatedA close up of a logo

Description automatically generatedA picture containing logo

Description automatically generatedA picture containing object, food

Description automatically generatedA picture containing object, food

Description automatically generatedA picture containing object, drawing

Description automatically generatedA close up of a logo

Description automatically generatedA close up of a logo

Description automatically generated

Phần Javascript

Phần AI.js : Trí tuệ nhân tạo và thuật toán tìm kiếm

var AI = AI||{};

AI.historyTable =   {};     // Bảng lịch sử

// khởi tạo trí tuệ nhân tạo

AI.init = function(pace){

    var bill = AI.historyBill || com.gambit; // Mở thư viện

    if (bill.length){

        var len=pace.length;

        var arr=[];

        // Tìm kiếm trò chơi trước

        for (var i=0;i< bill.length;i++){

            if (bill[i].slice(0,len)==pace) {

            arr.push(bill[i]);

            }

        }

        if (arr.length){

            var inx=Math.floor( Math.random() \* arr.length );

            AI.historyBill = arr ;

            return arr[inx].slice(len,len+4).split("");

        }else{

            AI.historyBill = [] ;

        }

    }

     // Nếu không có trò chơi bên trong, trí tuệ nhân tạo bắt đầu hoạt động.

    var initTime = new Date().getTime();

    AI.treeDepth=play.depth;

    //AI.treeDepth=4;

    AI.number=0;

    AI.setHistoryTable.lenght = 0

    var val=AI.getAlphaBeta(-99999 ,99999, AI.treeDepth, com.arr2Clone(play.map),play.my);

    //var val = AI.iterativeSearch(com.arr2Clone(play.map),play.my)

    if (!val||val.value==-8888) {

        AI.treeDepth=2;

        val=AI.getAlphaBeta(-99999 ,99999, AI.treeDepth, com.arr2Clone(play.map),play.my);

    }

    //var val = AI.iterativeSearch(com.arr2Clone(play.map),play.my);

    if (val&&val.value!=-8888) {

        var man = play.mans[val.key];

        var nowTime= new Date().getTime();

        com.get("moveInfo").innerHTML='<h3>AI Kết quả tìm kiếm：</h3>Cách tốt nhất để di chuyển：'+

                                        com.createMove(com.arr2Clone(play.map),man.x,man.y,val.x,val.y)+

                                        '<br />Độ sâu tìm kiếm：'+AI.treeDepth+'<br />Tìm kiếm nhánh：'+

                                        AI.number+'个 <br />Đánh giá điểm thực thi tốt nhất：'+

                                        val.value+'Điển'+

                                        ' <br />Thời gian tìm kiếm：'+

                                        (nowTime-initTime)+'mili giây'

        return [man.x,man.y,val.x,val.y]

    }else {

        return false;

    }

}

// Phương pháp tìm kiếm sâu lặp đi lặp lại

AI.iterativeSearch = function (map, my){

    var timeOut=100;

    var initDepth = 1;

    var maxDepth = 8;

    AI.treeDepth=0;

    var initTime = new Date().getTime();

    var val = {};

    for (var i=initDepth; i<=maxDepth; i++){

        var nowTime= new Date().getTime();

        AI.treeDepth=i;

        AI.aotuDepth=i;

        var val = AI.getAlphaBeta(-99999, 99999, AI.treeDepth , map ,my)

        if (nowTime-initTime > timeOut){

            return val;

        }

    }

    return false;

}

// Lấy tất cả các tướng trên bảng

AI.getMapAllMan = function (map, my){

    var mans=[];

    for (var i=0; i<map.length; i++){

        for (var n=0; n<map[i].length; n++){

            var key = map[i][n];

            if (key && play.mans[key].my == my){

                play.mans[key].x = n;

                play.mans[key].y = i;

                mans.push(play.mans[key])

            }

        }

    }

    return mans;

}

// Nhận tất cả các di chuyển của quân cờ

AI.getMoves = function (map, my){

    var manArr = AI.getMapAllMan (map, my);

    var moves = [];

    var foul=play.isFoul;

    for (var i=0; i<manArr.length; i++){

        var man = manArr[i];

        var val=man.bl(map);

        for (var n=0; n<val.length; n++){

            var x=man.x;

            var y=man.y;

            var newX=val[n][0];

            var newY=val[n][1];

             //Nếu nó không phải là một bước đi sai

            if (foul[0]!=x || foul[1]!=y || foul[2]!=newX || foul[3]!=newY ){

                moves.push([x,y,newX,newY,man.key])

            }

        }

    }

    return moves;

}

// A: giá trị người chơi hiện tại / B: giá trị đối thủ / độ sâu: cấp độ

AI.getAlphaBeta = function (A, B, depth, map ,my) {

    if (depth == 0) {

        return {"value":AI.evaluate(map , my)}; // chức năng đánh giá tình huống;

　   }

　   var moves = AI.getMoves(map , my ); // Tạo tất cả các lần chạy;

　   // Điều này sẽ tăng hiệu quả sau khi sắp xếp

    for (var i=0; i < moves.length; i++) {

    // Đi theo con đường này;

        var move= moves[i];

        var key = move[4];

        var oldX= move[0];

        var oldY= move[1];

        var newX= move[2];

        var newY= move[3];

        var clearKey = map[ newY ][ newX ]||"";

        map[ newY ][ newX ] = key;

        delete map[ oldY ][ oldX ];

        play.mans[key].x = newX;

        play.mans[key].y = newY;

    　　if (clearKey=="j0"||clearKey=="J0") {// Bị ăn quân, hoàn tác vụ này;

            play.mans[key]  .x = oldX;

            play.mans[key]  .y = oldY;

            map[ oldY ][ oldX ] = key;

            delete map[ newY ][ newX ];

            if (clearKey){

                 map[ newY ][ newX ] = clearKey;

                // play.mans[ clearKey ].isShow = false;

            }

            return {"key":key,"x":newX,"y":newY,"value":8888};

            //return rootKey;

    　　}else {

    　　  var val = -AI.getAlphaBeta(-B, -A, depth - 1, map , -my).value;

            //val = val || val.value;

    // Hoàn tác di chuyển này

            play.mans[key]  .x = oldX;

            play.mans[key]  .y = oldY;

            map[ oldY ][ oldX ] = key;

            delete map[ newY ][ newX ];

            if (clearKey){

                 map[ newY ][ newX ] = clearKey;

                 //play.mans[ clearKey ].isShow = true;

            }

    　　  if (val >= B) {

        // Ghi lại di chuyển này vào bảng lịch sử;

                //AI.setHistoryTable(txtMap,AI.treeDepth-depth+1,B,my);

                return {"key":key,"x":newX,"y":newY,"value":B};

            }

            if (val > A) {

    　　　　    A = val;

                    // Đặt cách tốt nhất để di chuyển;

                if (AI.treeDepth == depth) var rootKey={"key":key,"x":newX,"y":newY,"value":A};

            }

        }

　   }

        // Ghi lại di chuyển này vào bảng lịch sử;

    //AI.setHistoryTable(txtMap,AI.treeDepth-depth+1,A,my);

    if (AI.treeDepth == depth) {// đã trở lại gốc

        if (!rootKey){

        // AI không có cách tốt nhất để đi, chỉ ra rằng AI đã chết, trả về sai

            return false;

        }else{

            // Đây là cách tốt nhất để đi;

            return rootKey;

        }

    }

　return {"key":key,"x":newX,"y":newY,"value":A};

}

// Phương thức giải thưởng được ghi lại trong bảng lịch sử.

AI.setHistoryTable = function (txtMap,depth,value,my){

    AI.setHistoryTable.lenght ++;

    AI.historyTable[txtMap] = {depth:depth,value:value}

}

// Đánh giá trò chơi để có sự khác biệt giữa các quân cờ

AI.evaluate = function (map,my){

    var val=0;

    for (var i=0; i<map.length; i++){

        for (var n=0; n<map[i].length; n++){

            var key = map[i][n];

            if (key){

                val += play.mans[key].value[i][n] \* play.mans[key].my;

            }

        }

    }

    AI.number++;

    return val\*my;

}

// Đánh giá trò chơi để có sự khác biệt giữa các quân cờ

AI.evaluate1 = function (map,my){

    var val=0;

    for (var i in play.mans){

        var man=play.mans[i];

        if (man.isShow){

            val += man.value[man.y][man.x] \* man.my;

        }

    }

    AI.number++;

    return val\*my;

}

Phần Bill.js: Viết trò chơi

var bill = bill || {};

// Khởi tạo

bill.init = function (){

    if (com.store){

        clearInterval(bill.timer);

        bill.setBillList(com.arr2Clone(com.initMap)); // Viết danh sách quân cờ

        play.isPlay=false;

        com.show();

    }else {

        bill.timer = setInterval("bill.init()",300);

    }

}

// Viết tất cả các bản ghi trò chơi vào danh sách trò chơi

bill.setBillList = function (map){

    var list=com.get("billList")

    for (var i=0; i < com.store.length ; i++){

        var option = document.createElement('option');

         option.text='棋谱'+(i+1);

         option.value=i;

        list.add(option , null);

    }

    list.addEventListener("change", function(e) {

        bill.setBox (com.store[this.value], map)

    })

    bill.setBox (com.store[0], map)

}

// Phân tích phổ trò chơi Viết

bill.setMove = function (bl,inx,map){

    var map = com.arr2Clone(map);

    for (var i=0; i<map.length; i++){

        for (var n=0; n<map[i].length; n++){

            var key = map[i][n];

            if (key){

                com.mans[key].x=n;

                com.mans[key].y=i;

                com.mans[key].isShow = true;

            }

        }

    }

    for (var i=0; i<= inx ; i++){

        var n = i\*4

        var y = bl[n+1]

        var newX = bl[n+2]

        var x = bl[n+0]

        var newY = bl[n+3]

        if (com.mans[map[newY][newX]]) {

            com.mans[map[newY][newX]].isShow = false;

        }

        com.mans[map[y][x]].x = newX;

        com.mans[map[y][x]].y = newY;

        if (i == inx) {

            com.showPane(x ,y,newX,newY);

        }

        map[newY][newX] = map[y][x];

        delete map[y][x];

    }

    return map;

}

// Viết trò chơi

bill.setBox = function (bl,initMap){

    var map = com.arr2Clone(initMap);

    var bl= bl.split("");

    var h='';

    for (var i=0; i< bl.length ; i+=4){

        h +='<li id="move\_'+(i/4)+'">';

        var x = bl[i+0];

        var y = bl[i+1];

        var newX = bl[i+2];

        var newY = bl[i+3];

        h += com.createMove(map,x,y,newX,newY);

        h +='</li>\n\r';

    }

    com.get("billBox").innerHTML = h;

    var doms=com.get("billBox").getElementsByTagName("li");

    for (var i=0; i<doms.length; i++){

            doms[i].addEventListener("click", function(e) {

            var inx = this.getAttribute("id").split("\_")[1];

            bill.setMove (bl , inx , initMap)

            com.show();

        })

    }

}

Phần Common.js: Luật,các nước đi của quân cờ

var com = com||{};

com.init = function (stype){

    com.nowStype= stype || com.getCookie("stype") ||"stype2";

    var stype = com.stype[com.nowStype];

    com.width           =   stype.width;        //Chiều rộng

    com.height          =   stype.height;       //Chiều cao

    com.spaceX          =   stype.spaceX;       //Khoảng điểm X

    com.spaceY          =   stype.spaceY;       //Khoảng điểm Y

    com.pointStartX     =   stype.pointStartX;  //Điểm đầu tiên là tọa độ X;

    com.pointStartY     =   stype.pointStartY;  //Điểm đầu tiên là tọa độ Y;

    com.page            =   stype.page;         //Danh mục hình ảnh

    com.get("box").style.width = com.width+130+"px";

    com.canvas          =   document.getElementById("chess"); //Canvas

    com.ct              =   com.canvas.getContext("2d") ;

    com.canvas.width    =   com.width;

    com.canvas.height   =   com.height;

    com.childList       =   com.childList||[];

    com.loadImages(com.page);       //Tải thư mục hình ảnh / hình ảnh

    //z(com.initMap.join())

}

//Hiệu ứng

com.stype = {

    stype1:{

        width:325,      //Chiều rộng

        height:402,         //Chiều cao

        spaceX:35,      //Khoảng điểm X

        spaceY:36,      //Khoảng điểm Y

        pointStartX:5,      //Điểm đầu tiên là tọa độ X;

        pointStartY:19,     //Điểm đầu tiên là tọa độ Y;

        page:"stype\_1"  //Danh mục hình ảnh

    },

    stype2:{

        width:530,      //Chiều rộng

        height:567,         //Chiều cao

        spaceX:57,      //Khoảng điểm X

        spaceY:57,      //Khoảng điểm Y

        pointStartX:-2,     //Điểm đầu tiên là tọa độ X;

        pointStartY:0,      //Điểm đầu tiên là tọa độ Y;

        page:"stype\_2"  //Danh mục hình ảnh

    }

}

// Nhận ID

com.get = function (id){

    return document.getElementById(id)

}

window.onload = function(){

    com.bg=new com.class.Bg();

    com.dot = new com.class.Dot();

    com.pane=new com.class.Pane();

    com.pane.isShow=false;

    com.childList=[com.bg,com.dot,com.pane];

    com.mans     ={};       // Bộ sưu tập cờ tướng

    com.createMans(com.initMap)     // Tạo quân cờ

    com.bg.show();

    com.get("bnBox").style.display = "block";

    //play.init();

    com.get("billBn").addEventListener("click", function(e) {

        if (confirm("Bạn có kết thúc trò chơi và bắt đầu trò chơi?")){

            com.init();

            com.get("chessRight").style.display = "block";

            com.get("moveInfo").style.display = "none";

            bill.init();

        }

    })

    com.get("superPlay").addEventListener("click", function(e) {

        if (confirm("Xác nhận bắt đầu trò chơi mức Khó ?")){

            play.isPlay=true ;

            com.get("chessRight").style.display = "none";

            com.get("moveInfo").style.display = "block";

            com.get("moveInfo").innerHTML="";

            play.depth = 4;

            play.init();

        }

    })

    com.get("tyroPlay").addEventListener("click", function(e) {

        if (confirm("Xác nhận bắt đầu trò chơi mức Dễ？")){

            play.isPlay=true ;

            com.get("chessRight").style.display = "none";

            com.get("moveInfo").style.display = "block";

            com.get("moveInfo").innerHTML="";

            play.depth = 3;

            play.init();

        }

    })

    com.get("stypeBn").addEventListener("click", function(e) {

        var stype =com.nowStype;

        if (stype=="stype1") stype="stype2";

        else if (stype=="stype2") stype="stype1";

        com.init(stype);

        com.show();

        play.depth = 4;

        play.init();

        document.cookie="stype=" +stype;

        clearInterval(timer);

        var i=0;

        var timer = setInterval(function (){

            com.show();

            if (i++>=5) clearInterval(timer);

        },2000);

    })

    com.getData("js/gambit.all.js",

        function(data){

        com.gambit=data.split(" ");

        AI.historyBill = com.gambit;

    })

    com.getData("js/store.js",

        function(data){

        com.store=data.split(" ");

    })

}

// Tải hình ảnh

com.loadImages = function(stype){

// vẽ bàn cờ

    com.bgImg = new Image();

    com.bgImg.src  = "img/"+stype+"/bg.png";

    //điểm

    com.dotImg = new Image();

    com.dotImg.src  = "img/"+stype+"/dot.png";

    //cờ tướng

    for (var i in com.args){

        com[i] = {};

        com[i].img = new Image();

        com[i].img.src = "img/"+stype+"/"+ com.args[i].img +".png";

    }

    // Khung cờ

    com.paneImg = new Image();

    com.paneImg.src  = "img/"+stype+"/r\_box.png";

    document.getElementsByTagName("body")[0].style.background= "url(img/"+stype+"/bg.jpg)";

}

// Danh sách hiển thị

com.show = function (){

    com.ct.clearRect(0, 0, com.width, com.height);

    for (var i=0; i<com.childList.length ; i++){

        com.childList[i].show();

    }

}

// Hiển thị khung hộp cờ di chuyển

com.showPane  = function (x,y,newX,newY){

    com.pane.isShow=true;

    com.pane.x= x ;

    com.pane.y= y ;

    com.pane.newX= newX ;

    com.pane.newY= newY ;

}

// Tạo các quân cờ bên trong bản đồ

com.createMans = function(map){

    for (var i=0; i<map.length; i++){

        for (var n=0; n<map[i].length; n++){

            var key = map[i][n];

            if (key){

                com.mans[key]=new com.class.Man(key);

                com.mans[key].x=n;

                com.mans[key].y=i;

                com.childList.push(com.mans[key])

            }

        }

    }

}

//debug alert

com.alert = function (obj,f,n){

    if (typeof obj !== "object") {

        try{console.log(obj)} catch (e){}

        //return alert(obj);

    }

    var arr = [];

    for (var i in obj) arr.push(i+" = "+obj[i]);

    try{console.log(arr.join(n||"\n"))} catch (e){}

    //return alert(arr.join(n||"\n\r"));

}

//com.alert tốc ký, xem xét tên biến z là ít được sử dụng nhất

var z = com.alert;

// Lấy khoảng cách từ phía bên trái của trang

com.getDomXY = function (dom){

    var left = dom.offsetLeft;

    var top = dom.offsetTop;

    var current = dom.offsetParent;

    while (current !== null){

        left += current.offsetLeft;

        top += current.offsetTop;

        current = current.offsetParent;

    }

    return {x:left,y:top};

}

// Nhận cookie

com.getCookie = function(name){

    if (document.cookie.length>0){

        start=document.cookie.indexOf(name + "=")

        if (start!=-1){

            start=start + name.length+1

            end=document.cookie.indexOf(";",start)

        if (end==-1) end=document.cookie.length

            return unescape(document.cookie.substring(start,end))

        }

    }

    return false;

}

// bản sao mảng hai chiều

com.arr2Clone = function (arr){

    var newArr=[];

    for (var i=0; i<arr.length ; i++){

        newArr[i] = arr[i].slice();

    }

    return newArr;

}

// dữ liệu tải ajax

com.getData = function (url,fun){

    var XMLHttpRequestObject=false;

    if(window.XMLHttpRequest){

        XMLHttpRequestObject=new XMLHttpRequest();

    }else if(window.ActiveXObject){

    XMLHttpRequestObject=new ActiveXObject("Microsoft.XMLHTTP");

    }

    if(XMLHttpRequestObject){

        XMLHttpRequestObject.open("GET",url);

        XMLHttpRequestObject.setRequestHeader('Content-Type','application/x-www-form-urlencoded');

        XMLHttpRequestObject.onreadystatechange=function (){

            if(XMLHttpRequestObject.readyState==4 && XMLHttpRequestObject.status==200){

                fun (XMLHttpRequestObject.responseText)

                //return XMLHttpRequestObject.responseText;

            }

        }

    XMLHttpRequestObject.send(null);

    }

}

// Tạo tọa độ

com.createMove = function (map,x,y,newX,newY){

    var h="";

    var man = com.mans[map[y][x]];

    h+= man.text;

    map[newY][newX] = map[y][x];

    delete map[y][x];

    if (man.my===1){

        var mumTo=["一","二","三","四","五","六","七","八","九","十"];

        newX=8-newX;

        h+= mumTo[8-x];

        if (newY > y) {

            h+= "退";

            if (man.pater == "m" || man.pater == "s" || man.pater == "x"){

                h+= mumTo[newX];

            }else {

                h+= mumTo[newY - y -1];

            }

        }else if (newY < y) {

            h+= "进";

            if (man.pater == "m" || man.pater == "s" || man.pater == "x"){

                h+= mumTo[newX];

            }else {

                h+= mumTo[y - newY -1];

            }

        }else {

            h+= "平";

            h+= mumTo[newX];

        }

    }else{

        var mumTo=["１","２","３","４","５","６","７","８","９","10"];

        h+= mumTo[x];

        if (newY > y) {

            h+= "进";

            if (man.pater == "M" || man.pater == "S" || man.pater == "X"){

                h+= mumTo[newX];

            }else {

                h+= mumTo[newY - y-1];

            }

        }else if (newY < y) {

            h+= "退";

            if (man.pater == "M" || man.pater == "S" || man.pater == "X"){

                h+= mumTo[newX];

            }else {

                h+= mumTo[y - newY-1];

            }

        }else {

            h+= "平";

            h+= mumTo[newX];

        }

    }

    return h;

}

com.initMap = [

    ['C0','M0','X0','S0','J0','S1','X1','M1','C1'],

    [    ,    ,    ,    ,    ,    ,    ,    ,    ],

    [    ,'P0',    ,    ,    ,    ,    ,'P1',    ],

    ['Z0',    ,'Z1',    ,'Z2',    ,'Z3',    ,'Z4'],

    [    ,    ,    ,    ,    ,    ,    ,    ,    ],

    [    ,    ,    ,    ,    ,    ,    ,    ,    ],

    ['z0',    ,'z1',    ,'z2',    ,'z3',    ,'z4'],

    [    ,'p0',    ,    ,    ,    ,    ,'p1',    ],

    [    ,    ,    ,    ,    ,    ,    ,    ,    ],

    ['c0','m0','x0','s0','j0','s1','x1','m1','c1']

];

com.initMap1 = [

    [    ,    ,    ,, "J0"   ,    ,    ,    ,    ],

    [    ,    ,    ,    ,    ,    ,    ,    ,    ],

    [    ,    ,    ,    ,    ,"c0",    ,    ,    ],

    [    ,    ,    ,    ,    ,    ,    ,    ,    ],

    [    ,    ,    ,    ,    ,    ,    ,    ,    ],

    [    ,    ,    ,    ,    ,    ,    ,    ,    ],

    [    ,    ,    ,    ,    ,    ,    ,    ,    ],

    [    ,    ,    ,    ,    ,    ,    ,    ,    ],

    [    ,    ,    ,    ,"s0",    ,    ,"C0",    ],

    [    ,    ,    ,"s1",    ,"j0",    ,    ,    ]

];

com.initMap1 = [

    [    ,    ,    ,, "J0"   ,    ,    ,    ,    ],

    [    ,    ,    ,    ,    , ,    ,    ,    ],

    [    ,    ,    ,    ,    ,"z0",    ,    ,    ],

    [    ,    ,    ,    ,    ,    ,    ,    ,    ],

    [    ,    ,    ,    ,    ,    ,    ,    ,    ],

    [    ,    ,    ,    ,    ,    ,    ,    ,    ],

    [    ,    ,    ,    ,    ,    ,    ,    ,    ],

    [    ,    ,    ,    ,    ,    ,    ,    ,    ],

    [    ,    ,    ,    ,    ,    ,    ,    ,    ],

    [    ,    ,    , "j0"   ,,    ,    ,    ]

];

com.keys = {

    "c0":"c","c1":"c",

    "m0":"m","m1":"m",

    "x0":"x","x1":"x",

    "s0":"s","s1":"s",

    "j0":"j",

    "p0":"p","p1":"p",

    "z0":"z","z1":"z","z2":"z","z3":"z","z4":"z","z5":"z",

    "C0":"c","C1":"C",

    "M0":"M","M1":"M",

    "X0":"X","X1":"X",

    "S0":"S","S1":"S",

    "J0":"J",

    "P0":"P","P1":"P",

    "Z0":"Z","Z1":"Z","Z2":"Z","Z3":"Z","Z4":"Z","Z5":"Z",

}

// Cờ  có thể đi.

com.bylaw ={}

// xe

com.bylaw.c = function (x,y,map,my){

    var d=[];

    // trái tìm kiếm

    for (var i=x-1; i>= 0; i--){

        if (map[y][i]) {

            if (com.mans[map[y][i]].my!=my) d.push([i,y]);

            break

        }else{

            d.push([i,y])

        }

    }

    // Tìm kiếm bên phải

    for (var i=x+1; i <= 8; i++){

        if (map[y][i]) {

            if (com.mans[map[y][i]].my!=my) d.push([i,y]);

            break

        }else{

            d.push([i,y])

        }

    }

    // Tìm kiếm

    for (var i = y-1 ; i >= 0; i--){

        if (map[i][x]) {

            if (com.mans[map[i][x]].my!=my) d.push([x,i]);

            break

        }else{

            d.push([x,i])

        }

    }

    // Tìm kiếm dưới

    for (var i = y+1 ; i<= 9; i++){

        if (map[i][x]) {

            if (com.mans[map[i][x]].my!=my) d.push([x,i]);

            break

        }else{

            d.push([x,i])

        }

    }

    return d;

}

//Ngựa

com.bylaw.m = function (x,y,map,my){

    var d=[];

        //1Điểm

        if ( y-2>= 0 && x+1<= 8 && !play.map[y-1][x] &&(!com.mans[map[y-2][x+1]] || com.mans[map[y-2][x+1]].my!=my)) d.push([x+1,y-2]);

        //2Điểm

        if ( y-1>= 0 && x+2<= 8 && !play.map[y][x+1] &&(!com.mans[map[y-1][x+2]] || com.mans[map[y-1][x+2]].my!=my)) d.push([x+2,y-1]);

        //4Điểm

        if ( y+1<= 9 && x+2<= 8 && !play.map[y][x+1] &&(!com.mans[map[y+1][x+2]] || com.mans[map[y+1][x+2]].my!=my)) d.push([x+2,y+1]);

        //5Điểm

        if ( y+2<= 9 && x+1<= 8 && !play.map[y+1][x] &&(!com.mans[map[y+2][x+1]] || com.mans[map[y+2][x+1]].my!=my)) d.push([x+1,y+2]);

        //7Điểm

        if ( y+2<= 9 && x-1>= 0 && !play.map[y+1][x] &&(!com.mans[map[y+2][x-1]] || com.mans[map[y+2][x-1]].my!=my)) d.push([x-1,y+2]);

        //8Điểm

        if ( y+1<= 9 && x-2>= 0 && !play.map[y][x-1] &&(!com.mans[map[y+1][x-2]] || com.mans[map[y+1][x-2]].my!=my)) d.push([x-2,y+1]);

        //10Điểm

        if ( y-1>= 0 && x-2>= 0 && !play.map[y][x-1] &&(!com.mans[map[y-1][x-2]] || com.mans[map[y-1][x-2]].my!=my)) d.push([x-2,y-1]);

        //11Điểm

        if ( y-2>= 0 && x-1>= 0 && !play.map[y-1][x] &&(!com.mans[map[y-2][x-1]] || com.mans[map[y-2][x-1]].my!=my)) d.push([x-1,y-2]);

    return d;

}

//Tượng

com.bylaw.x = function (x,y,map,my){

    var d=[];

    if (my===1){ //Quân đỏ

        //4Nửa điểm

        if ( y+2<= 9 && x+2<= 8 && !play.map[y+1][x+1] && (!com.mans[map[y+2][x+2]] || com.mans[map[y+2][x+2]].my!=my)) d.push([x+2,y+2]);

        //7Nửa điểm

        if ( y+2<= 9 && x-2>= 0 && !play.map[y+1][x-1] && (!com.mans[map[y+2][x-2]] || com.mans[map[y+2][x-2]].my!=my)) d.push([x-2,y+2]);

        //1Nửa điểm

        if ( y-2>= 5 && x+2<= 8 && !play.map[y-1][x+1] && (!com.mans[map[y-2][x+2]] || com.mans[map[y-2][x+2]].my!=my)) d.push([x+2,y-2]);

        //10Nửa điểm

        if ( y-2>= 5 && x-2>= 0 && !play.map[y-1][x-1] && (!com.mans[map[y-2][x-2]] || com.mans[map[y-2][x-2]].my!=my)) d.push([x-2,y-2]);

    }else{

        //4Nửa điểm

        if ( y+2<= 4 && x+2<= 8 && !play.map[y+1][x+1] && (!com.mans[map[y+2][x+2]] || com.mans[map[y+2][x+2]].my!=my)) d.push([x+2,y+2]);

        //7Nửa điểm

        if ( y+2<= 4 && x-2>= 0 && !play.map[y+1][x-1] && (!com.mans[map[y+2][x-2]] || com.mans[map[y+2][x-2]].my!=my)) d.push([x-2,y+2]);

        //1Nửa điểm

        if ( y-2>= 0 && x+2<= 8 && !play.map[y-1][x+1] && (!com.mans[map[y-2][x+2]] || com.mans[map[y-2][x+2]].my!=my)) d.push([x+2,y-2]);

        //10Nửa điểm

        if ( y-2>= 0 && x-2>= 0 && !play.map[y-1][x-1] && (!com.mans[map[y-2][x-2]] || com.mans[map[y-2][x-2]].my!=my)) d.push([x-2,y-2]);

    }

    return d;

}

//sỹ

com.bylaw.s = function (x,y,map,my){

    var d=[];

    if (my===1){ //Quân đỏ

        //4Nửa điểm

        if ( y+1<= 9 && x+1<= 5 && (!com.mans[map[y+1][x+1]] || com.mans[map[y+1][x+1]].my!=my)) d.push([x+1,y+1]);

        //7点半

        if ( y+1<= 9 && x-1>= 3 && (!com.mans[map[y+1][x-1]] || com.mans[map[y+1][x-1]].my!=my)) d.push([x-1,y+1]);

        //1Nửa điểm

        if ( y-1>= 7 && x+1<= 5 && (!com.mans[map[y-1][x+1]] || com.mans[map[y-1][x+1]].my!=my)) d.push([x+1,y-1]);

        //10Nửa điểm

        if ( y-1>= 7 && x-1>= 3 && (!com.mans[map[y-1][x-1]] || com.mans[map[y-1][x-1]].my!=my)) d.push([x-1,y-1]);

    }else{

        //4Nửa điểm

        if ( y+1<= 2 && x+1<= 5 && (!com.mans[map[y+1][x+1]] || com.mans[map[y+1][x+1]].my!=my)) d.push([x+1,y+1]);

        //7Nửa điểm

        if ( y+1<= 2 && x-1>= 3 && (!com.mans[map[y+1][x-1]] || com.mans[map[y+1][x-1]].my!=my)) d.push([x-1,y+1]);

        //1Nửa điểm

        if ( y-1>= 0 && x+1<= 5 && (!com.mans[map[y-1][x+1]] || com.mans[map[y-1][x+1]].my!=my)) d.push([x+1,y-1]);

        //10Nửa điểm

        if ( y-1>= 0 && x-1>= 3 && (!com.mans[map[y-1][x-1]] || com.mans[map[y-1][x-1]].my!=my)) d.push([x-1,y-1]);

    }

    return d;

}

//Tướng

com.bylaw.j = function (x,y,map,my){

    var d=[];

    var isNull=(function (y1,y2){

        var y1=com.mans["j0"].y;

        var x1=com.mans["J0"].x;

        var y2=com.mans["J0"].y;

        for (var i=y1-1; i>y2; i--){

            if (map[i][x1]) return false;

        }

        return true;

    })();

    if (my===1){ //Quân đỏ

        //dưới

        if ( y+1<= 9  && (!com.mans[map[y+1][x]] || com.mans[map[y+1][x]].my!=my)) d.push([x,y+1]);

        //trên

        if ( y-1>= 7 && (!com.mans[map[y-1][x]] || com.mans[map[y-1][x]].my!=my)) d.push([x,y-1]);

         // Tình huống 2 binh doi đầu

        if ( com.mans["j0"].x == com.mans["J0"].x &&isNull) d.push([com.mans["J0"].x,com.mans["J0"].y]);

    }else{

        //dưới

        if ( y+1<= 2  && (!com.mans[map[y+1][x]] || com.mans[map[y+1][x]].my!=my)) d.push([x,y+1]);

        //trên

        if ( y-1>= 0 && (!com.mans[map[y-1][x]] || com.mans[map[y-1][x]].my!=my)) d.push([x,y-1]);

        // Tình huống 2 binh doi đầu

        if ( com.mans["j0"].x == com.mans["J0"].x &&isNull) d.push([com.mans["j0"].x,com.mans["j0"].y]);

    }

    //đúng

    if ( x+1<= 5  && (!com.mans[map[y][x+1]] || com.mans[map[y][x+1]].my!=my)) d.push([x+1,y]);

    // Còn lại

    if ( x-1>= 3 && (!com.mans[map[y][x-1]] || com.mans[map[y][x-1]].my!=my))d.push([x-1,y]);

    return d;

}

// pháo

com.bylaw.p = function (x,y,map,my){

    var d=[];

    // trái tìm kiếm

    var n=0;

    for (var i=x-1; i>= 0; i--){

        if (map[y][i]) {

            if (n==0){

                n++;

                continue;

            }else{

                if (com.mans[map[y][i]].my!=my) d.push([i,y]);

                break

            }

        }else{

            if(n==0) d.push([i,y])

        }

    }

    // Tìm kiếm bên phải

    var n=0;

    for (var i=x+1; i <= 8; i++){

        if (map[y][i]) {

            if (n==0){

                n++;

                continue;

            }else{

                if (com.mans[map[y][i]].my!=my) d.push([i,y]);

                break

            }

        }else{

            if(n==0) d.push([i,y])

        }

    }

    // Tìm kiếm

    var n=0;

    for (var i = y-1 ; i >= 0; i--){

        if (map[i][x]) {

            if (n==0){

                n++;

                continue;

            }else{

                if (com.mans[map[i][x]].my!=my) d.push([x,i]);

                break

            }

        }else{

            if(n==0) d.push([x,i])

        }

    }

    // Tìm kiếm dưới

    var n=0;

    for (var i = y+1 ; i<= 9; i++){

        if (map[i][x]) {

            if (n==0){

                n++;

                continue;

            }else{

                if (com.mans[map[i][x]].my!=my) d.push([x,i]);

                break

            }

        }else{

            if(n==0) d.push([x,i])

        }

    }

    return d;

}

//Tốt

com.bylaw.z = function (x,y,map,my){

    var d=[];

    if (my===1){ //quân đỏ

        //trên

        if ( y-1>= 0 && (!com.mans[map[y-1][x]] || com.mans[map[y-1][x]].my!=my)) d.push([x,y-1]);

        //phải

        if ( x+1<= 8 && y<=4  && (!com.mans[map[y][x+1]] || com.mans[map[y][x+1]].my!=my)) d.push([x+1,y]);

        //còn lại

        if ( x-1>= 0 && y<=4 && (!com.mans[map[y][x-1]] || com.mans[map[y][x-1]].my!=my))d.push([x-1,y]);

    }else{

        //dưới

        if ( y+1<= 9  && (!com.mans[map[y+1][x]] || com.mans[map[y+1][x]].my!=my)) d.push([x,y+1]);

        //phải

        if ( x+1<= 8 && y>=6  && (!com.mans[map[y][x+1]] || com.mans[map[y][x+1]].my!=my)) d.push([x+1,y]);

        // Còn lại

        if ( x-1>= 0 && y>=6 && (!com.mans[map[y][x-1]] || com.mans[map[y][x-1]].my!=my))d.push([x-1,y]);

    }

    return d;

}

com.value = {

    // giá trị xe

    c:[

        [206, 208, 207, 213, 214, 213, 207, 208, 206],

        [206, 212, 209, 216, 233, 216, 209, 212, 206],

        [206, 208, 207, 214, 216, 214, 207, 208, 206],

        [206, 213, 213, 216, 216, 216, 213, 213, 206],

        [208, 211, 211, 214, 215, 214, 211, 211, 208],

        [208, 212, 212, 214, 215, 214, 212, 212, 208],

        [204, 209, 204, 212, 214, 212, 204, 209, 204],

        [198, 208, 204, 212, 212, 212, 204, 208, 198],

        [200, 208, 206, 212, 200, 212, 206, 208, 200],

        [194, 206, 204, 212, 200, 212, 204, 206, 194]

    ],

    // giá trị ngựa

    m:[

        [90, 90, 90, 96, 90, 96, 90, 90, 90],

        [90, 96,103, 97, 94, 97,103, 96, 90],

        [92, 98, 99,103, 99,103, 99, 98, 92],

        [93,108,100,107,100,107,100,108, 93],

        [90,100, 99,103,104,103, 99,100, 90],

        [90, 98,101,102,103,102,101, 98, 90],

        [92, 94, 98, 95, 98, 95, 98, 94, 92],

        [93, 92, 94, 95, 92, 95, 94, 92, 93],

        [85, 90, 92, 93, 78, 93, 92, 90, 85],

        [88, 85, 90, 88, 90, 88, 90, 85, 88]

    ],

    //giá trị tượng

    x:[

        [0, 0,20, 0, 0, 0,20, 0, 0],

        [0, 0, 0, 0, 0, 0, 0, 0, 0],

        [0, 0, 0, 0,23, 0, 0, 0, 0],

        [0, 0, 0, 0, 0, 0, 0, 0, 0],

        [0, 0,20, 0, 0, 0,20, 0, 0],

        [0, 0,20, 0, 0, 0,20, 0, 0],

        [0, 0, 0, 0, 0, 0, 0, 0, 0],

        [18,0, 0, 0,23, 0, 0, 0,18],

        [0, 0, 0, 0, 0, 0, 0, 0, 0],

        [0, 0,20, 0, 0, 0,20, 0, 0]

    ],

    //giá trị sỹ

    s:[

        [0, 0, 0,20, 0,20, 0, 0, 0],

        [0, 0, 0, 0,23, 0, 0, 0, 0],

        [0, 0, 0,20, 0,20, 0, 0, 0],

        [0, 0, 0, 0, 0, 0, 0, 0, 0],

        [0, 0, 0, 0, 0, 0, 0, 0, 0],

        [0, 0, 0, 0, 0, 0, 0, 0, 0],

        [0, 0, 0, 0, 0, 0, 0, 0, 0],

        [0, 0, 0,20, 0,20, 0, 0, 0],

        [0, 0, 0, 0,23, 0, 0, 0, 0],

        [0, 0, 0,20, 0,20, 0, 0, 0]

    ],

    //giá trị tướng

    j:[

        [0, 0, 0, 8888, 8888, 8888, 0, 0, 0],

        [0, 0, 0, 8888, 8888, 8888, 0, 0, 0],

        [0, 0, 0, 8888, 8888, 8888, 0, 0, 0],

        [0, 0, 0, 0, 0, 0, 0, 0, 0],

        [0, 0, 0, 0, 0, 0, 0, 0, 0],

        [0, 0, 0, 0, 0, 0, 0, 0, 0],

        [0, 0, 0, 0, 0, 0, 0, 0, 0],

        [0, 0, 0, 8888, 8888, 8888, 0, 0, 0],

        [0, 0, 0, 8888, 8888, 8888, 0, 0, 0],

        [0, 0, 0, 8888, 8888, 8888, 0, 0, 0]

    ],

    // Giá trị pháo

    p:[

        [100, 100,  96, 91,  90, 91,  96, 100, 100],

        [ 98,  98,  96, 92,  89, 92,  96,  98,  98],

        [ 97,  97,  96, 91,  92, 91,  96,  97,  97],

        [ 96,  99,  99, 98, 100, 98,  99,  99,  96],

        [ 96,  96,  96, 96, 100, 96,  96,  96,  96],

        [ 95,  96,  99, 96, 100, 96,  99,  96,  95],

        [ 96,  96,  96, 96,  96, 96,  96,  96,  96],

        [ 97,  96, 100, 99, 101, 99, 100,  96,  97],

        [ 96,  97,  98, 98,  98, 98,  98,  97,  96],

        [ 96,  96,  97, 99,  99, 99,  97,  96,  96]

    ],

    //giá trị tốt

    z:[

        [ 9,  9,  9, 11, 13, 11,  9,  9,  9],

        [19, 24, 34, 42, 44, 42, 34, 24, 19],

        [19, 24, 32, 37, 37, 37, 32, 24, 19],

        [19, 23, 27, 29, 30, 29, 27, 23, 19],

        [14, 18, 20, 27, 29, 27, 20, 18, 14],

        [ 7,  0, 13,  0, 16,  0, 13,  0,  7],

        [ 7,  0,  7,  0, 15,  0,  7,  0,  7],

        [ 0,  0,  0,  0,  0,  0,  0,  0,  0],

        [ 0,  0,  0,  0,  0,  0,  0,  0,  0],

        [ 0,  0,  0,  0,  0,  0,  0,  0,  0]

    ]

}

// vết quân đen giá trị cho các vị trí đảo ngược quân màu đỏ

com.value.C = com.arr2Clone(com.value.c).reverse();

com.value.M = com.arr2Clone(com.value.m).reverse();

com.value.X = com.value.x;

com.value.S = com.value.s;

com.value.J = com.value.j;

com.value.P = com.arr2Clone(com.value.p).reverse();

com.value.Z = com.arr2Clone(com.value.z).reverse();

//tướng

com.args={

    // 红 Trung Quốc / địa chỉ hình ảnh / phe / trọng lượng

    'c':{text:"车", img:'r\_c', my:1 ,bl:"c", value:com.value.c},

    'm':{text:"马", img:'r\_m', my:1 ,bl:"m", value:com.value.m},

    'x':{text:"相", img:'r\_x', my:1 ,bl:"x", value:com.value.x},

    's':{text:"仕", img:'r\_s', my:1 ,bl:"s", value:com.value.s},

    'j':{text:"将", img:'r\_j', my:1 ,bl:"j", value:com.value.j},

    'p':{text:"炮", img:'r\_p', my:1 ,bl:"p", value:com.value.p},

    'z':{text:"兵", img:'r\_z', my:1 ,bl:"z", value:com.value.z},

    //quân xanh

    'C':{text:"車", img:'b\_c', my:-1 ,bl:"c", value:com.value.C},

    'M':{text:"馬", img:'b\_m', my:-1 ,bl:"m", value:com.value.M},

    'X':{text:"象", img:'b\_x', my:-1 ,bl:"x", value:com.value.X},

    'S':{text:"士", img:'b\_s', my:-1 ,bl:"s", value:com.value.S},

    'J':{text:"帅", img:'b\_j', my:-1 ,bl:"j", value:com.value.J},

    'P':{text:"炮", img:'b\_p', my:-1 ,bl:"p", value:com.value.P},

    'Z':{text:"卒", img:'b\_z', my:-1 ,bl:"z", value:com.value.Z}

};

com.class = com.class || {} //lớp

com.class.Man = function (key, x, y){

    this.pater = key.slice(0,1);

    var o=com.args[this.pater]

    this.x = x||0;

    this.y = y||0;

    this.key = key ;

    this.my = o.my;

    this.text = o.text;

    this.value = o.value;

    this.isShow = true;

    this.alpha = 1;

    this.ps = []; // Điểm

    this.show = function (){

        if (this.isShow) {

            com.ct.save();

            com.ct.globalAlpha = this.alpha;

            com.ct.drawImage(com[this.pater].img,com.spaceX \* this.x + com.pointStartX , com.spaceY \*  this.y +com.pointStartY);

            com.ct.restore();

        }

    }

    this.bl = function (map){

        var map = map || play.map

        return com.bylaw[o.bl](this.x,this.y,map,this.my)

    }

}

com.class.Bg = function (img, x, y){

    this.x = x||0;

    this.y = y||0;

    this.isShow = true;

    this.show = function (){

        if (this.isShow) com.ct.drawImage(com.bgImg, com.spaceX \* this.x,com.spaceY \*  this.y);

    }

}

com.class.Pane = function (img, x, y){

    this.x = x||0;

    this.y = y||0;

    this.newX = x||0;

    this.newY = y||0;

    this.isShow = true;

    this.show = function (){

        if (this.isShow) {

            com.ct.drawImage(com.paneImg, com.spaceX \* this.x + com.pointStartX , com.spaceY \*  this.y + com.pointStartY)

            com.ct.drawImage(com.paneImg, com.spaceX \* this.newX + com.pointStartX  , com.spaceY \*  this.newY + com.pointStartY)

        }

    }

}

com.class.Dot = function (img, x, y){

    this.x = x||0;

    this.y = y||0;

    this.isShow = true;

    this.dots=[]

    this.show = function (){

        for (var i=0; i<this.dots.length;i++){

            if (this.isShow) com.ct.drawImage(com.dotImg, com.spaceX \* this.dots[i][0]+10  + com.pointStartX ,com.spaceY \*  this.dots[i][1]+10 + com.pointStartY)

        }

    }

}

com.init();

Phần Play.js: Chạy game

var play = play||{};

play.init = function (){

    play.my             =   1;              // Phía người chơi

    play.map            =   com.arr2Clone (com.initMap);        // Khởi tạo bảng

    play.nowManKey      =   false;          // play sẽ vận hành ngay bây giờ

    play.pace           =   [];             // Ghi lại từng bước

    play.isPlay         =   true ;          // Bạn có thể chơi cờ tướng không?

    play.mans           =   com.mans;

    play.bylaw          =   com.bylaw;

    play.show           =   com.show;

    play.showPane       =   com.showPane;

    play.isOffensive    =   true;           // Cho dù đó là lần đầu tiên

    play.depth          =   play.depth || 3;                // Độ sâu tìm kiếm

    play.isFoul         =   false;  // Bạn có phạm lỗi

    com.pane.isShow     =    false;         // Ẩn hộp

    // Khởi tạo quân cờ

    for (var i=0; i<play.map.length; i++){

        for (var n=0; n<play.map[i].length; n++){

            var key = play.map[i][n];

            if (key){

                com.mans[key].x=n;

                com.mans[key].y=i;

                com.mans[key].isShow = true;

            }

        }

    }

    play.show();

    //Sự kiện nhấp chuột

    com.canvas.addEventListener("click",play.clickCanvas)

    com.get("regretBn").addEventListener("click", function(e) {

        play.regret();

    })

    /\*

    var initTime = new Date().getTime();

    for (var i=0; i<=100000; i++){

        var h=""

        var h=play.map.join();

        //for (var n in play.mans){

        //  if (play.mans[n].show) h+=play.mans[n].key+play.mans[n].x+play.mans[n].y

        //}

    }

    var nowTime= new Date().getTime();

    z([h,nowTime-initTime])

    \*/

}

// lùi

play.regret = function (){

    var map  = com.arr2Clone(com.initMap);

    // Khởi tạo tất cả các phần

    for (var i=0; i<map.length; i++){

        for (var n=0; n<map[i].length; n++){

            var key = map[i][n];

            if (key){

                com.mans[key].x=n;

                com.mans[key].y=i;

                com.mans[key].isShow = true;

            }

        }

    }

    var pace= play.pace;

    pace.pop();

    pace.pop();

    for (var i=0; i<pace.length; i++){

        var p= pace[i].split("")

        var x = parseInt(p[0], 10);

        var y = parseInt(p[1], 10);

        var newX = parseInt(p[2], 10);

        var newY = parseInt(p[3], 10);

        var key=map[y][x];

        //try{

        var cMan=map[newY][newX];

        if (cMan) com.mans[map[newY][newX]].isShow = false;

        com.mans[key].x = newX;

        com.mans[key].y = newY;

        map[newY][newX] = key;

        delete map[y][x];

        if (i==pace.length-1){

            com.showPane(newX ,newY,x,y)

        }

        //} catch (e){

        //  com.show()

        //  z([key,p,pace,map])

        //  }

    }

    play.map = map;

    play.my=1;

    play.isPlay=true;

    com.show();

}

// Bấm vào bảng sự kiện

play.clickCanvas = function (e){

    if (!play.isPlay) return false;

    var key = play.getClickMan(e);

    var point = play.getClickPoint(e);

    var x = point.x;

    var y = point.y;

    if (key){

        play.clickMan(key,x,y);

    }else {

        play.clickPoint(x,y);

    }

    play.isFoul = play.checkFoul();// Kiểm tra xem nó có lâu không

}

// Nhấp vào tướng, trong hai trường hợp, chọn hoặc ăn

play.clickMan = function (key,x,y){

    var man = com.mans[key];

    //an quan

    if (play.nowManKey&&play.nowManKey != key && man.my != com.mans[play.nowManKey ].my){

        //manCho các miếng được ăn

        if (play.indexOfPs(com.mans[play.nowManKey].ps,[x,y])){

            man.isShow = false;

            var pace=com.mans[play.nowManKey].x+""+com.mans[play.nowManKey].y

            //z(bill.createMove(play.map,man.x,man.y,x,y))

            delete play.map[com.mans[play.nowManKey].y][com.mans[play.nowManKey].x];

            play.map[y][x] = play.nowManKey;

            com.showPane(com.mans[play.nowManKey].x ,com.mans[play.nowManKey].y,x,y)

            com.mans[play.nowManKey].x = x;

            com.mans[play.nowManKey].y = y;

            com.mans[play.nowManKey].alpha = 1

            play.pace.push(pace+x+y);

            play.nowManKey = false;

            com.pane.isShow = false;

            com.dot.dots = [];

            com.show()

            com.get("clickAudio").play();

            setTimeout("play.AIPlay()",500);

            if (key == "j0") play.showWin (-1);

            if (key == "J0") play.showWin (1);

        }

    // kiểm tra tướng

    }else{

        if (man.my===1){

            if (com.mans[play.nowManKey]) com.mans[play.nowManKey].alpha = 1 ;

            man.alpha = 0.6;

            com.pane.isShow = false;

            play.nowManKey = key;

            com.mans[key].ps = com.mans[key].bl(); // Nhận tất cả các điểm bạn có thể

            com.dot.dots = com.mans[key].ps

            com.show();

            //com.get("selectAudio").start(0);

            com.get("selectAudio").play();

        }

    }

}

// Nhấp vào điểm

play.clickPoint = function (x,y){

    var key=play.nowManKey;

    var man=com.mans[key];

    if (play.nowManKey){

        if (play.indexOfPs(com.mans[key].ps,[x,y])){

            var pace=man.x+""+man.y

            //z(bill.createMove(play.map,man.x,man.y,x,y))

            delete play.map[man.y][man.x];

            play.map[y][x] = key;

            com.showPane(man.x ,man.y,x,y)

            man.x = x;

            man.y = y;

            man.alpha = 1;

            play.pace.push(pace+x+y);

            play.nowManKey = false;

            com.dot.dots = [];

            com.show();

            com.get("clickAudio").play();

            setTimeout("play.AIPlay()",500);

        }else{

            //alert("Không thể đi theo cách này!")

        }

    }

}

// Ai tự động di chuyển quân cờ

play.AIPlay = function (){

    //return

    play.my = -1 ;

    var pace=AI.init(play.pace.join(""))

    if (!pace) {

        play.showWin (1);

        return ;

    }

    play.pace.push(pace.join(""));

    var key=play.map[pace[1]][pace[0]]

        play.nowManKey = key;

    var key=play.map[pace[3]][pace[2]];

    if (key){

        play.AIclickMan(key,pace[2],pace[3]);

    }else {

        play.AIclickPoint(pace[2],pace[3]);

    }

    com.get("clickAudio").play();

}

// Kiểm tra nếu lâu sẽ lỗi

play.checkFoul = function(){

    var p=play.pace;

    var len=parseInt(p.length,10);

    if (len>11&&p[len-1] == p[len-5] &&p[len-5] == p[len-9]){

        return p[len-4].split("");

    }

    return false;

}

play.AIclickMan = function (key,x,y){

    var man = com.mans[key];

    // ăn quân

    man.isShow = false;

    delete play.map[com.mans[play.nowManKey].y][com.mans[play.nowManKey].x];

    play.map[y][x] = play.nowManKey;

    play.showPane(com.mans[play.nowManKey].x ,com.mans[play.nowManKey].y,x,y)

    com.mans[play.nowManKey].x = x;

    com.mans[play.nowManKey].y = y;

    play.nowManKey = false;

    com.show()

    if (key == "j0") play.showWin (-1);

    if (key == "J0") play.showWin (1);

}

play.AIclickPoint = function (x,y){

    var key=play.nowManKey;

    var man=com.mans[key];

    if (play.nowManKey){

        delete play.map[com.mans[play.nowManKey].y][com.mans[play.nowManKey].x];

        play.map[y][x] = key;

        com.showPane(man.x,man.y,x,y)

        man.x = x;

        man.y = y;

        play.nowManKey = false;

    }

    com.show();

}

play.indexOfPs = function (ps,xy){

    for (var i=0; i<ps.length; i++){

        if (ps[i][0]==xy[0]&&ps[i][1]==xy[1]) return true;

    }

    return false;

}

// Nhận điểm nhấp

play.getClickPoint = function (e){

    var domXY = com.getDomXY(com.canvas);

    var x=Math.round((e.pageX-domXY.x-com.pointStartX-20)/com.spaceX)

    var y=Math.round((e.pageY-domXY.y-com.pointStartY-20)/com.spaceY)

    return {"x":x,"y":y}

}

// Lấy tướng

play.getClickMan = function (e){

    var clickXY=play.getClickPoint(e);

    var x=clickXY.x;

    var y=clickXY.y;

    if (x < 0 || x>8 || y < 0 || y > 9) return false;

    return (play.map[y][x] && play.map[y][x]!="0") ? play.map[y][x] : false;

}

play.showWin = function (my){

    play.isPlay = false;

    if (my===1){

        alert("Xin chúc mừng, bạn đã thắng! ");

    }else{

        alert("Thật không may, bạn đã thua!");

    }

}

Ngoài ra còn file gambit.js và store.js là 2 file data bổ trợ cho AI

Phần III : DEMO

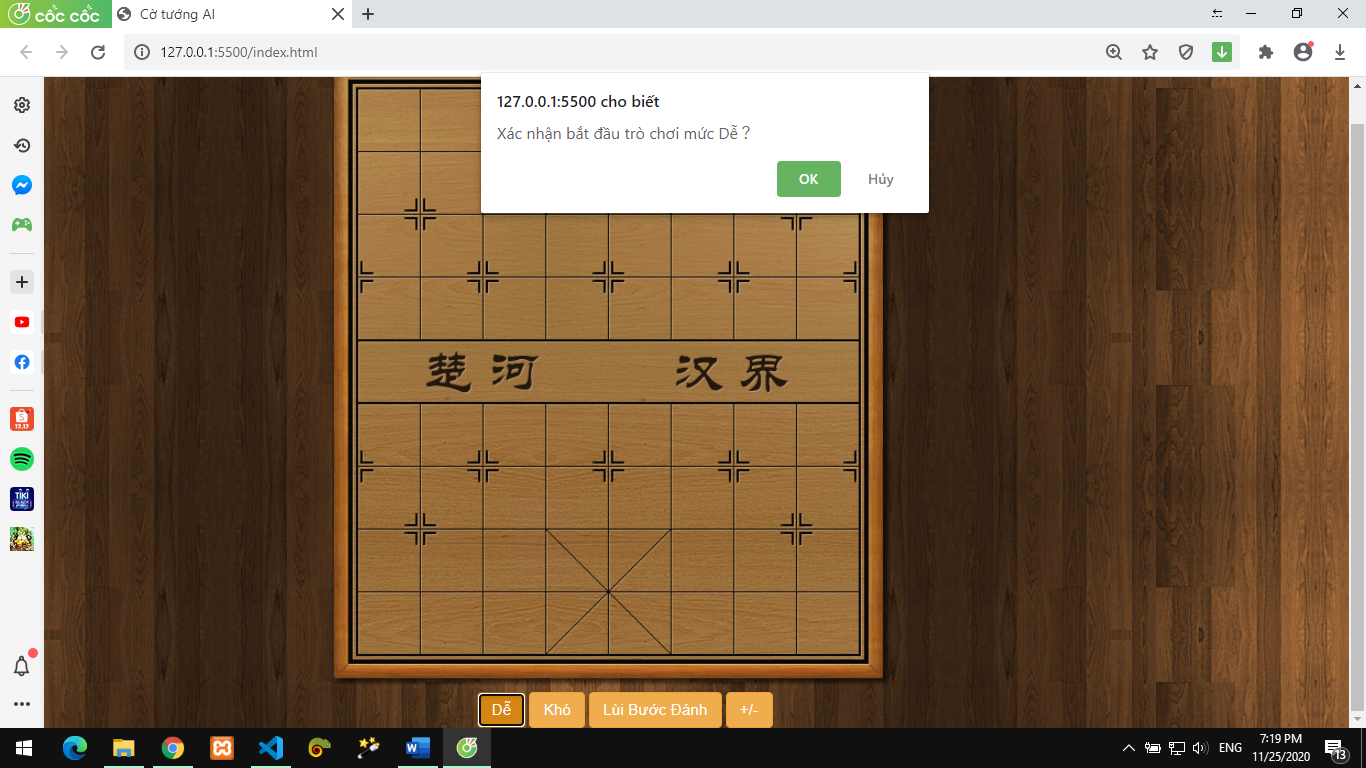
Chạy chương trình bằng cách khởi động file index.html:

+Giao diện khi chạy chương trình game cờ tướng :



+Chọn mức dễ hoặc khó để bắt đầu chơi,vd Dễ nhấn ok:

Mức dễ tìm theo độ sâu là :3 Mức khó có độ sâu: 4



+Khi ta chọn vào 1 quân cờ nó sẽ hiện các nước có thể đi, vd Pháo :



+Khi ta đi,Thuật toán AI tìm kiến sẽ tìm theo độ sâu để tìm được nước đi tốt nhất nó có thể đi:

+Vd khi ta dung pháo ăn tốt 5 của đối thủ:



+Thuật toán tìm ra nước đi tốt nhất là mã 7 tiến 5 để ăn quân pháo của ta:



+Show kết quả tìm kiếm ra màn hình:

Graphical user interface, website

Description automatically generated

+Khi chiếu tướng,tướng k còn nước thoát thì thắng và hiện thông báo thắng,đối với ta khi chơi mà thua thì ngược lại sẽ thông báo thua:

+Vd 2 quân xe ta đang chiếu tướng máy,tướng máy không còn đường thoát thì sẽ hiện thông báo ta chiến thắng.

Diagram

Description automatically generated

Link demo:

https://drive.google.com/drive/folders/1SKOY9bN\_1evF5jI62iJ6WY-rJhpQ1Y13?usp=sharing