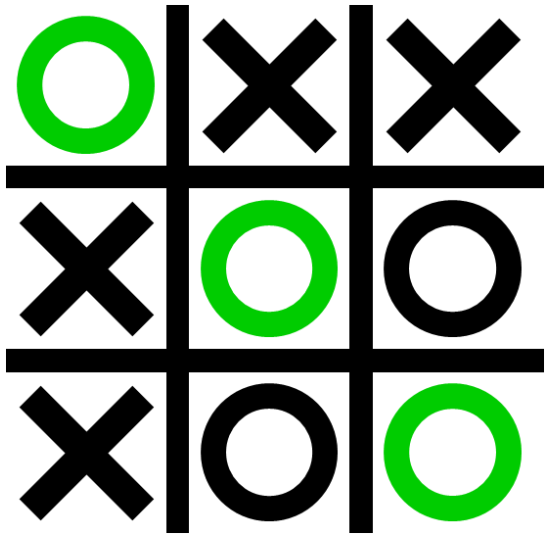
Mathematica Project——
Tic Tac Toe



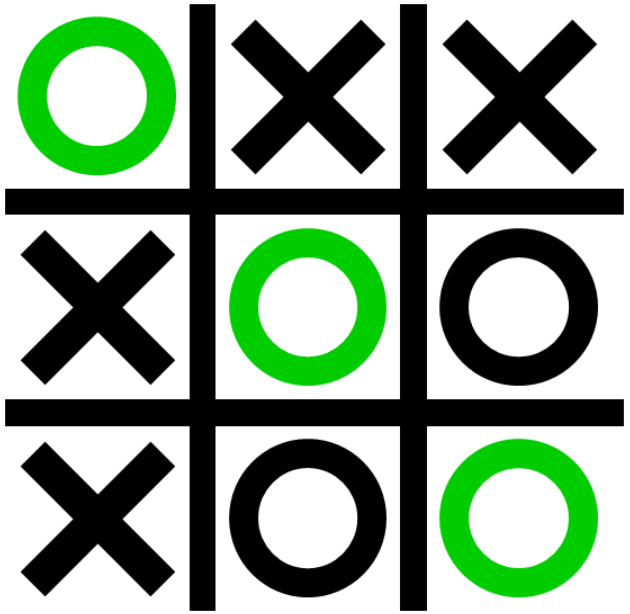
Qiao Ren

V2L5Q4

Introduction

- What is the problem: Rule and the Goal of the tic tac toe project
- How to solve the problem:
  - Initialization and assumption
  - Where to move：Minimax Algorithm
  - Who wins the game: Multiplication of Magic Square
- How to implement the idea:
  - code explanation

Rules and Goal



Board: 3*3

Two players:

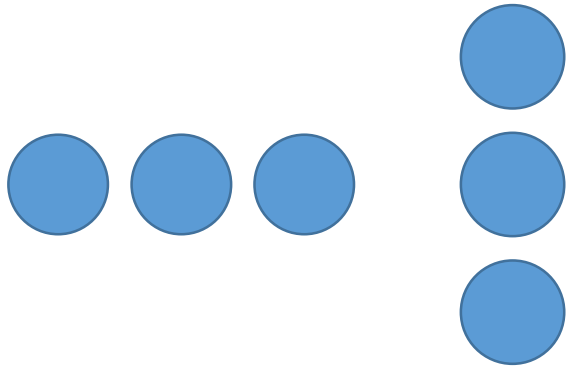
- Computer: X
- Human player: O

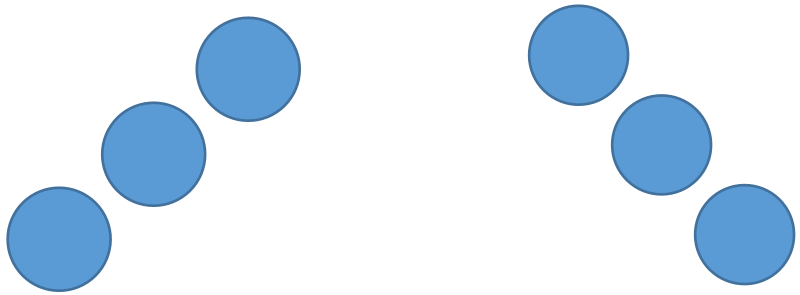
Success: three same things occur

- in a row
- or in a column
- or in a diagonal

Goal: computer will never loose the game:

- either win
- or draw





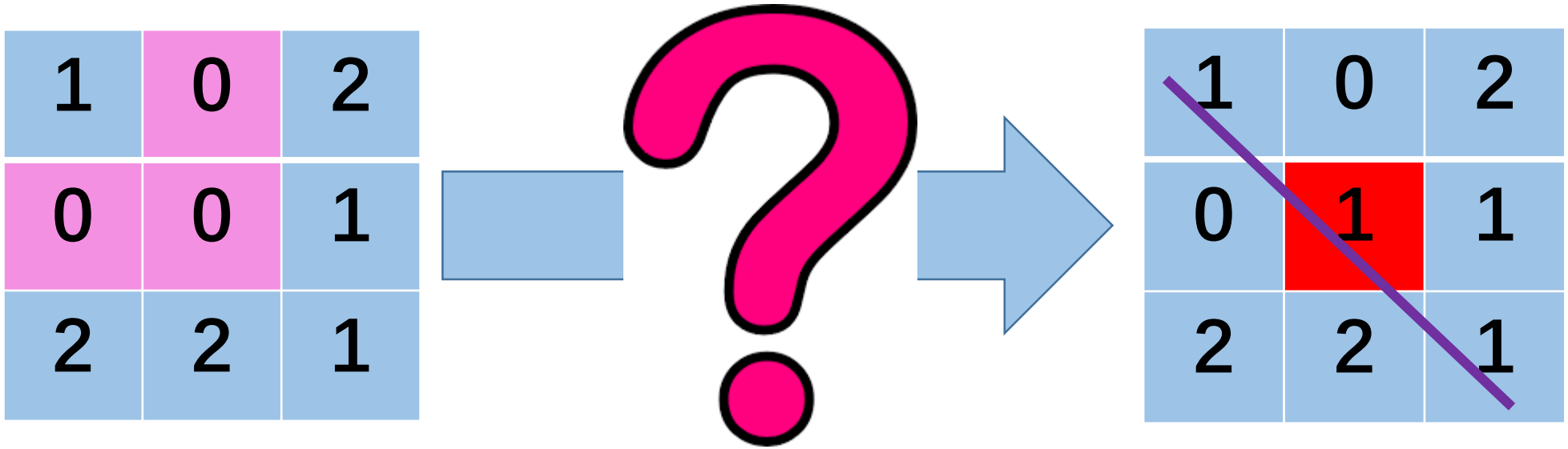
Assumptions and Initializations

i / j	1	2	3
1	(1,1) =0	(1,2) =0	(1,3) =0
2	(2,1) =0	(2,2) =0	(2,3) =0
3	(3,1) =0	(3,2) =0	(3,3) =0

- Coordinator of a cell: (i,j)
- Two players:
  - Computer: denoted by 1, draw a X
  - Human player: denoted by 2, draw a O
  - Empty cell: denoted by 0
- Order: Human player can choose to go first or second.
  - go first: {{0,0,0},{0,0,0},{0,0,0}}
  - go second: {{1,0,0},{0,0,0},{0,0,0}}
- Board is a matrix

As the computer player, which cell is the best position to move in?

→Minimax algorithm



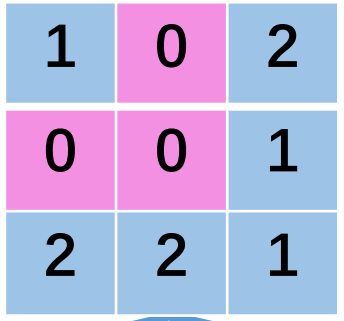
Minimax algorithm

- **Maximum: computer’ s turn:**
  best position for computer
  alpha
- **Minimum: computer’ s opponent (=human player)’ s turn**
  worst position for the computer
  beta

Minimax algorithm

Computer: 1
Human: 2
Empty: 0

a node



alpha=10

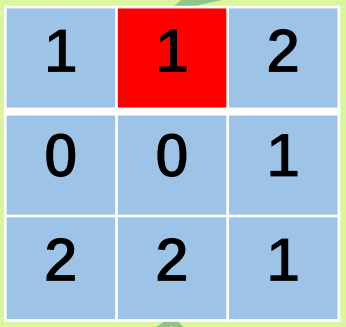
Computer: 1
Human: 2
Empty: 0

Children of a node

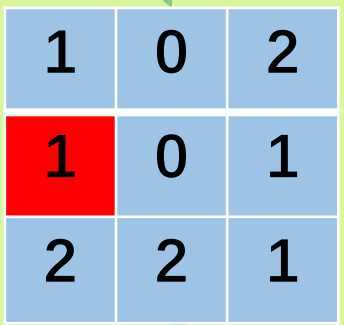
Max score of its children(-10,-10,10):10

Computer' s turn:
find max

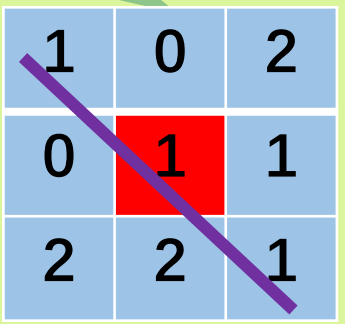
beta=-10



beta=-10



beta=10

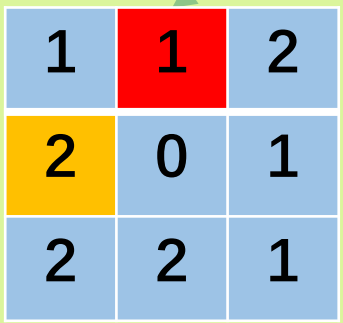


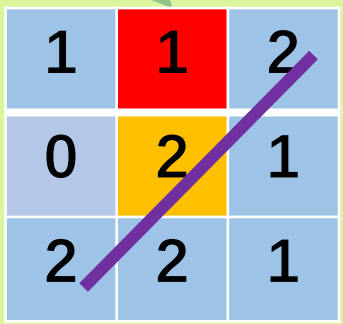
Length of children=0

Min score of its children (0,-10):-10

Min score of its children (0,-10):-10

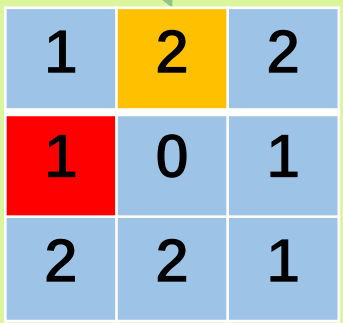
Human' s turn:
find min

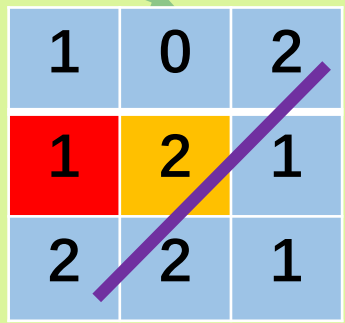




0

-10



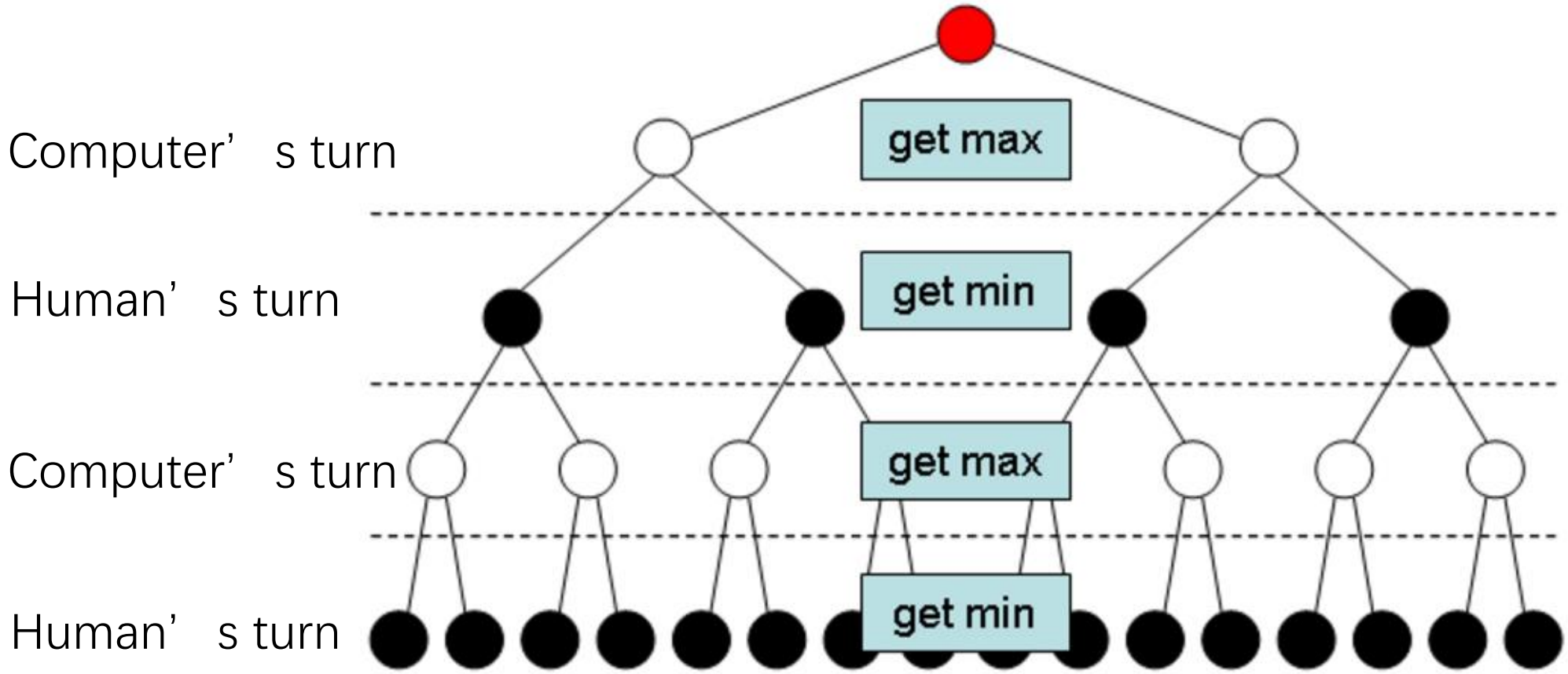


0

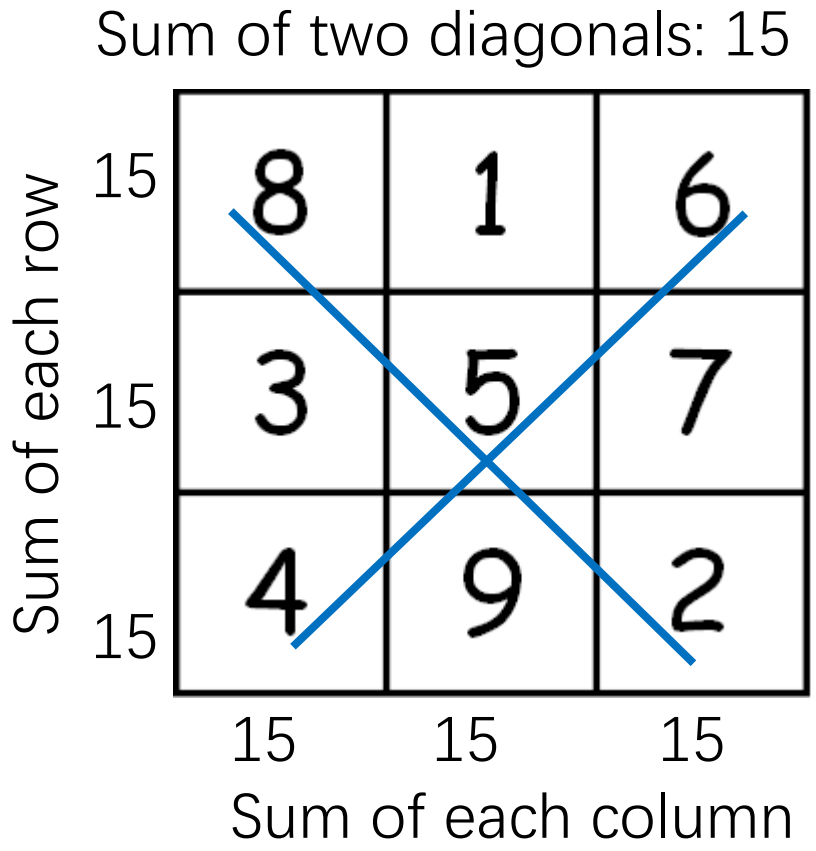
-10

Minimax algorithm

Node: board situation



How to know who wins or no body wins?
→multiplication with a magic square



How to know who wins or no body wins?
→multiplication with a magic square

Magic square

8	1	6
3	5	7
4	9	2

multiplies

Board matrix

1	1	1
2	0	2
1	2	0

Computer: 1
Human: 2
Empty: 0

=

Sum of two diagonals: 8,10

Sum of each row

15	8	1	6
20	6	0	14
22	4	18	0
	18	19	20

Sum of each column

score={{**15**,20,22},{18,19,20},8,10}

Computer: 1
Human: 2
Empty: 0

Magic square

8	1	6
3	5	7
4	9	2

multi plies

2	0	2
1	1	1
2	2	0

or

0	1	1
0	1	2
2	1	2

or

2	0	1
0	1	2
1	0	0

In the sum list, find 15
→player 1 wins
→ Return 1

{{28,**15**,26},{27,23,19},21,25}

{{7,19,21},{8,**15**,24},9,19}

{{22,19,4},{20,5,20},21,**15**}

1	0	1
2	2	2
1	2	1

or

0	2	1
0	2	2
1	2	1

or

1	0	2
0	2	1
2	0	1

In the sum list, find 30
→player 2 wins
→ Return -1

{{14,**30**,22},{18,28,20},18,20}

{{8,24,24},{4,**30**,22},12,20}

{{20,17,10},{16,10,21},20,**30**}

0	0	1
2	0	2
1	2	0

{{6,20,22},{10,18,20},0,10}

In the sum list,
neither 15 nor 30 exist
→ Return 0

Code explanation-part 1: List Children

- **listChildren[board_, player_] := listChildren[board, player] = ReplaceList[board, {a___, {x___, 0, y___}, b___} :> {a, {x, player, y}, b}]**

- Input: listChildren[{{1,0,0},{2,0,2},{0,0,0}},**2**]
- Output:

$$\{\{\{1,\mathbf{2},0\},\{2,0,2\},\{0,0,0\}\},\{\{1,0,\mathbf{2}\},\{2,0,2\},\{0,0,0\}\},\{\{1,0,0\},\{2,\mathbf{2},2\},\{0,0,0\}\},\{\{1,0,0\},\{2,0,2\},\{\mathbf{2},0,0\}\},\{\{1,0,0\},\{2,0,2\},\{0,\mathbf{2},0\}\},\{\{1,0,0\},\{2,0,2\},\{0,0,\mathbf{2}\}\}\}$$

Code explanation-part2: check who wins

```
score[board_] := With[{score = {
    Total /@ #,                                  (*sum of each rows*)
    Total /@ Transpose[#],                       (*sum of each columns*)
    Tr[#],                                       (*sum of diagonal*)
    Tr[Reverse@#]                                (* sum of the other diagonal *)
    } &[{{8, 1, 6}, {3, 5, 7}, {4, 9, 2}} board] },
  Which[  MemberQ[score, 15, 2], 1, MemberQ[score, 30, 2], -1,   True, ] ]
```

- Input: $\text{score}[\{\{1,1,1\},\{1,0,2\},\{0,0,0\}\}]$ Output: 1
- Input: $\text{score}[\{\{2,2,2\},\{1,0,2\},\{0,0,0\}\}]$ Output: -1
- Input: $\text{score}[\{\{1,2,2\},\{1,0,2\},\{0,0,0\}\}]$ Output: 0

Code explanation-part3

- **nextMove[oldBoard_] :=**

If[score[oldBoard] != 0 || ! MemberQ[oldBoard, 0, 2],

oldBoard,

Part[listChildren[oldBoard, 1], Last@Ordering [alphabeta[#, -10, 10, False] & /@ listChildren[oldBoard, 1]]]]

Some body wins

All the cells are occupied

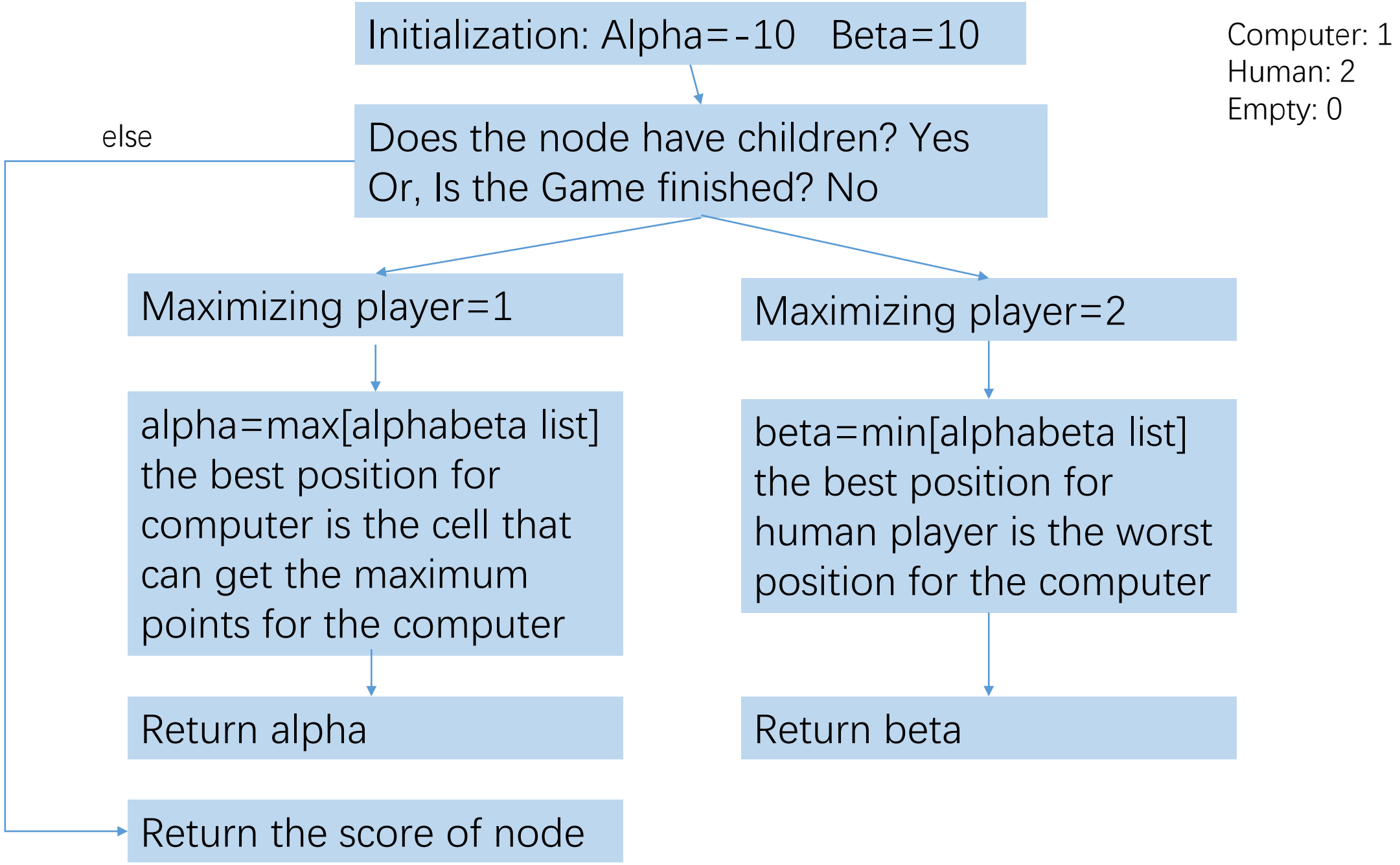
→ Game is over→return old board

No body wins and empty cells exist

→ Game is not finished→list all the children of computer player→move to the best position

Code explanation-part4: minimax algorithm

```
alphabeta[node_, a_, b_, maximizingPlayer_] :=
Module[  {alpha = a, beta = b, children},
  children = listChildren[node, If[maximizingPlayer, 1, 2]];
If[Length@children == 0 || score@node != 0, Return@score@node];
  If[maximizingPlayer,
Do[alpha = Max[alpha, alphabeta[child, alpha, beta, False]];
        If[beta <= alpha, Break[]]; , {child, children}];    Return[alpha],
Do[beta = Min[beta, alphabeta[child, alpha, beta, True]];
        If[beta <= alpha, Break[]]; , {child, children}];   Return[beta]   ] ]
```

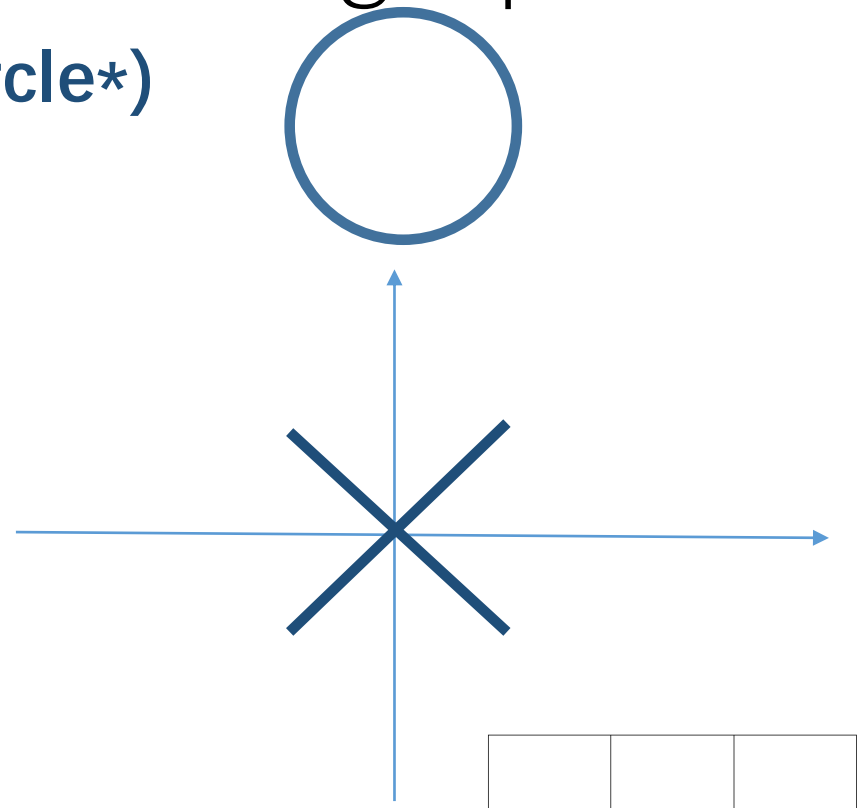

Initialization: Alpha=-10 Beta=10
Computer: 1
Human: 2
Empty: 0
else
Does the node have children? Yes
Or, Is the Game finished? No
Maximizing player=1
Maximizing player=2
alpha=max[alphabeta list] the best position for computer is the cell that can get the maximum points for the computer
beta=min[alphabeta list] the best position for human player is the worst position for the computer
Return alpha
Return beta
Return the score of node

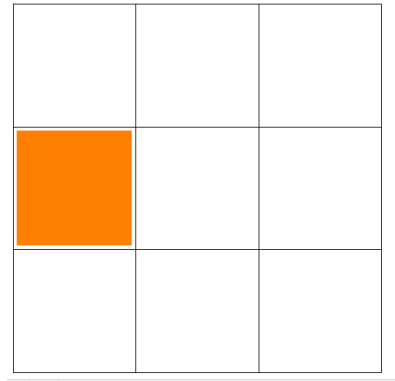
Code explanation-part5: draw graphs

```
circle = Graphics[Circle[]];(*draw a circle*)

cross = Graphics[{ (*draw a cross*)
    Thick,
    Line[{
     {{-1, 1}, {1, -1}},
     {{1, 1}, {-1, -1}}
     }] }];

empty = Graphics[]; (*an empty cell*)
hover = Graphics[{}, Background -> Orange];
```





Code explanation-part6: reaction to human player

button[{i_, j_}] := Button[(create a button: label “LineHand” , action)

Mouseover[empty,

(*when the mouse does not pass through a cell, the cell shows empty*)

MouseAppearance[hover, "LinkHand"]],

(*when the mouse passes through a cell, change color, shows LinkHand*)

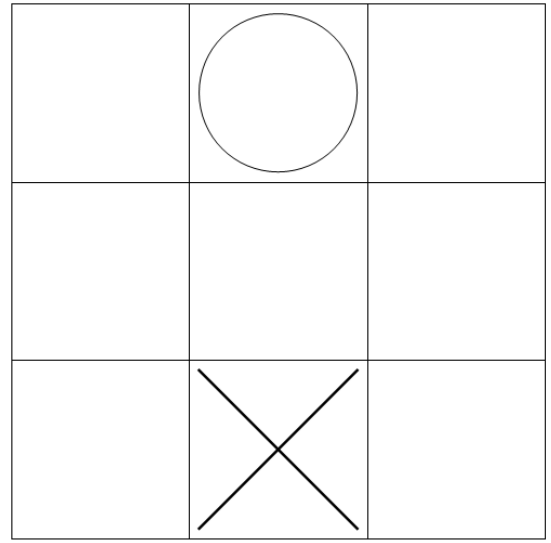
currentBoard = nextMove[ReplacePart[currentBoard, {i, j} -> 2]],
(*In current Board, cell(i,j)=2. *)

Appearance -> None]

Code explanation-part7

```
renderBoard[board_] := GraphicsGrid[Normal@SparseArray[
Join[
      # -> cross & /@ Position[#, 1], (*if the cell’ s value is 1, draw cross*)
      # -> circle & /@ Position[#, 2], (*if the cell’ s value is 2, draw circle*)
      {{i_, j_} /; Extract[#, {i, j}] == 0 :> button[{i, j}]} ], {3, 3}
     ] &[board], Frame -> All, ImageSize -> 400
  ]
```

- Input: renderBoard[{{0,2,0},{0,0,0},{0,1,0}}]
- Output:



Code explanation-part8: Initialization

```
Dynamic@Deploy@Overlay[{renderBoard[currentBoard],If[scor
e[currentBoard]!=0||!MemberQ[currentBoard,0,2],Pane[Column[
{Button["Go first",
currentBoard={{0,0,0},{0,0,0},{0,0,0}}],
Button["Go second",
currentBoard={{1,0,0},{0,0,0},{0,0,0}}]}],
ImageMargins->155],##&[]]},{1,2},
If[score[currentBoard]!=0||!MemberQ[currentBoard,0,2],2,1]]
```

Thank you for your attention