U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Boku

Relatório Final

Mestrado Integrado em Engenharia Informática e Computação

Programação Lógica 2017/2018

Faculdade de Engenharia da Universidade do Porto Rua Roberto Frias, sn,
4200-465 Porto, Portugal

Grupo : Boku_1

Joel Márcio Torres Carneiro - 201100775

Ângelo Daniel Pereira Mendes Moura - 201303828

15 de Outubro de 2017

12 de Novembro de 2017

Resumo

Neste relatório é apresentada uma possível implementação do jogo de tabuleiro Boku na linguagem de programação lógica, PROLOG. O desenvolvimento deste jogo surgiu no âmbito da Unidade Curricular de Programação em Lógica da Faculdade de Engenharia do Porto.

A implementação utiliza como base de visualização do jogo a consola, ou seja, o jogo é visualizado em modo de texto. Desta forma, foi possível desenvolver o jogo com mais foco na parte lógica do mesmo, como uma menor preocupação com a sua interface.

Como o jogo foi totalmente implementado, o utilizador pode escolher os diferentes modos de jogo e debater-se com este jogo de estratégia. Contudo, a implementação desenvolvida pode não ser a melhor, pelo menos em termos de inteligência do computador. Desta forma, a abordagem utilizada no modo computador poderia ser melhorada no sentido de esta ser mais inteligente, visto que apenas escolhe valores random para as posições do tabuleiro.

12 de Novembro de 2017

1. Introdução

O trabalho desenvolvido tem como objetivo a criação de um jogo usando a linguagem de programação Prolog. O jogo em questão trata-se do jogo de estratégia Boku.

Foi desenvolvida uma interface de texto para a visualização do jogo de modo a que o utilizador consiga perceber o que está a acontecer e tomar as suas decisões no que diz respeito às próximas jogadas.

Neste relatório será descrito o jogo e as sua regras assim como a lógica de jogo implementada. Posteriormente será especificado no relatório a representação do estado de jogo, a visualização do tabuleiro e os principais predicados utilizados. Para finalizar será detalhada a interface com o utilizador, serão retiradas as conclusões e ficará em anexo o código fonte desenvolvido.

2. Boku

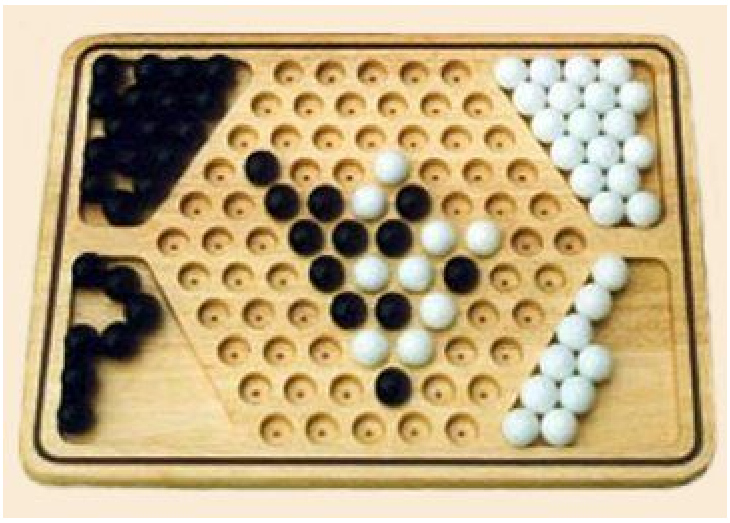


Figura 1 - Tabuleiro de jogo

Boku é um jogo de tabuleiro de estratégia que é jogado colocando pedras num tabuleiro hexagonal com 80 espaços. O objetivo do jogo é conseguir 5 pedras em linha. O jogo também foi vendido sob o nome Bollox, e mais tarde Bolix e ganhou um prêmio Mensa Select em 1999.

O jogo foi inventado por Rob Nelson, antigo lançador canhoto de Portland Mavericks e criador do bubblegum Big League Chew. A idéia para o jogo surgiu a Rob Nelson em 1991. Juntamente com um bom amigo e proprietário dos Spartans Malcolm Needs, eles desenvolveram e comercializaram o jogo. Por um tempo manteve a posição de ser o melhor jogo de estratégia de dois jogadores em Harrods e Hamleys. O jogo recebeu uma Mensa International Gold Star.

No início do jogo não há pedras no tabuleiro. Existem dois jogadores, um deles tem 36 pedras pretas e o outro jogador tem 36 pedras brancas.

2.1 Regras do jogo

O jogo duas duas regras principais:

- O jogo é ganho se o jogador colocar 5 peças da sua cor em linha, na horizontal ou numa das diagonais.

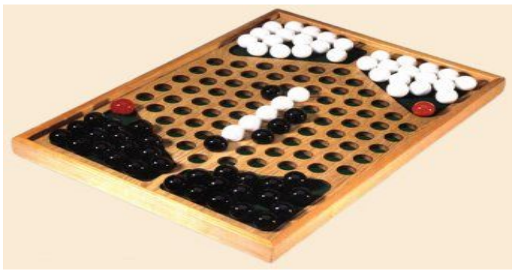


Figura 2 - 5 peças em linha (horizontal).

- Se um jogador armadilha duas das peças do seu oponente entre duas das suas peças, ele pode remover uma das peças do jogador oponente que foram "ensanduichadas". Para além disso, o adversário não pode colocar uma das suas peças no mesmo lugar na próxima jogada.

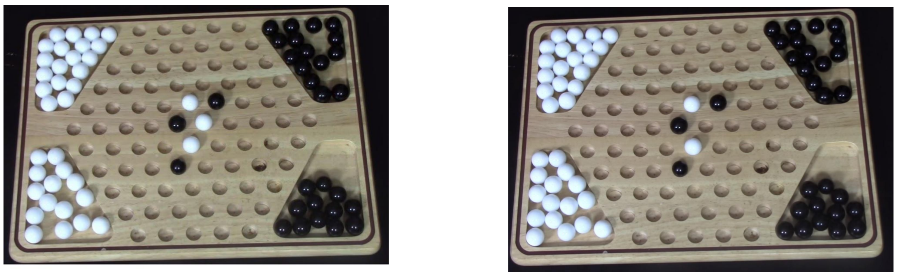


Figura 3 - Exemplo de peças "ensanduichadas" e captura de uma das peças.

- ❏ Este caso acontece apenas quando o jogador coloca a peça nas extremidades. Se o jogador colocar no interior as suas peças isto já não acontece. Por exemplo, daqui a umas jogadas o jogador colocar uma peça branca no sítio que ficou agora vazio. (De notar que não pode colocar lá a peça branca na jogada seguinte.)
- ❏ Envolvendo apenas uma peça adversária ou mais do que duas não dá direito a capturar nenhuma peça adversária. Além disso, as duas peças “ensanduichadas” devem estar numa linha horizontal ou diagonal direta.

3. Lógica de Jogo

O programa inicia num menu onde é possível escolher o tipo de jogadores envolvidos, sendo as escolhas possíveis: Player VS Pc, Pc VS Pc e Player VS Player. Depois da escolha feita o jogo em si começa.

O jogo é, em termos de código, um ciclo. Neste ciclo existe um momento de escolha de jogadas, quer por um humano quer pelo computador, a verificação se esta jogada é válida ou não, sendo que caso a jogada não seja válida o jogador perde o seu turno, de seguida é verificado se uma situação de captura existe e caso exista é pedido ao jogador que escolha a peça que deseja capturar e finalmente é verificado se o jogo chegou ao fim, caso isto tenha acontecido o programa acaba com a informação de quem o ganhou.

3.1 Representação do Estado do Jogo

Para representar o tabuleiro de jogo irá ser usada uma representação do tipo lista de listas, exemplificada em baixo. Os caracteres ‘ ’ representam um espaço vazio do tabuleiro e os caracteres ‘E’ são espaço vazio para manter todas as listas com o tamanho de 10 elementos.

```
emptyBoard([[' ',' ',' ',' ',' ','E','E','E','E','E'],
            [' ',' ',' ',' ',' ',' ','E','E','E','E'],
            [' ',' ',' ',' ',' ',' ',' ','E','E','E'],
            [' ',' ',' ',' ',' ',' ',' ',' ','E','E'],
            [' ',' ',' ',' ',' ',' ',' ',' ',' ','E'],
            [' ',' ',' ',' ',' ',' ',' ',' ',' ',' '],
```

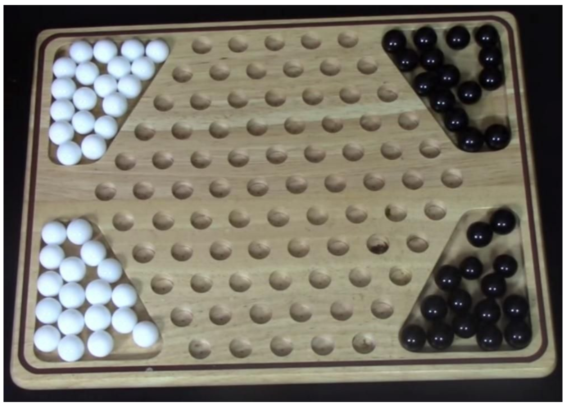


Figura 4 - Tabuleiro de jogo vazio.

```
        ['E',' ',' ',' ',' ',' ',' ',' ',' ',' '],
        ['E','E',' ',' ',' ',' ',' ',' ',' ',' '],
        ['E','E','E',' ',' ',' ',' ',' ',' ',' '],
        ['E','E','E','E',' ',' ',' ',' ',' ',' '],
        ['E','E','E','E','E',' ',' ',' ',' ',' ']
        ]).
```

De seguida são dados os exemplos de como é representada a estrutura do tabuleiro na lista de listas, de forma a representar a imagem 5 e imagem 6.

```
emptyBoard([[' ',' ',' ',' ',' ','E','E','E','E','E'],
        [' ',' ',' ',' ',' ',' ','E','E','E','E'],
        [' ',' ',' ',' ',' ',' ',' ','E','E','E'],
        [' ',' ',' ',' ',' ',' ',' ',' ','E','E'],
        [' ',' ',' ',' ','W','B',' ',' ',' ','E'],
        [' ',' ',' ',' ','B','W',' ',' ',' ',' '],
```

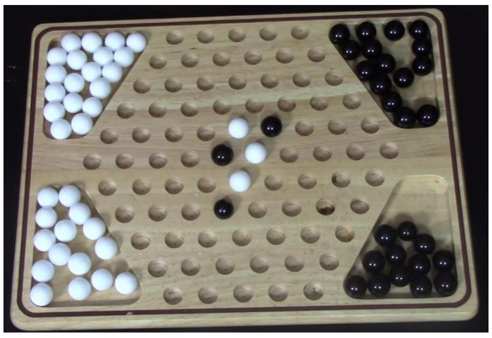


Figura 5 - Tabuleiro a meio do jogo.

```
        ['E',' ',' ',' ',' ','W',' ',' ',' ',' '],
        ['E','E',' ',' ',' ','B',' ',' ',' ',' '],
                ['E','E','E',' ',' ',' ',' ',' ',' ',' '],
                ['E','E','E','E',' ',' ',' ',' ',' ',' '],
                ['E','E','E','E','E',' ',' ',' ',' ',' '] ]).

emptyBoard([[' ',' ',' ',' ',' ','E','E','E','E','E'],
                [' ',' ',' ',' ',' ',' ','E','E','E','E'],
                [' ',' ',' ',' ',' ',' ',' ','E','E','E'],
                [' ',' ',' ',' ',' ',' ',' ',' ','E','E'],
                [' ',' ',' ',' ','W','B',' ',' ',' ','E'],

                [' ',' ',' ',' ','B',' ',' ',' ',' ',' '],
                ['E',' ',' ',' ',' ','W',' ',' ',' ',' '],
                ['E','E',' ',' ',' ','B',' ',' ',' ',' '],
                ['E','E','E',' ',' ',' ',' ',' ',' ',' '],
                ['E','E','E','E',' ',' ',' ',' ',' ',' '],
                ['E','E','E','E','E',' ',' ',' ',' ',' ']  ]).
```

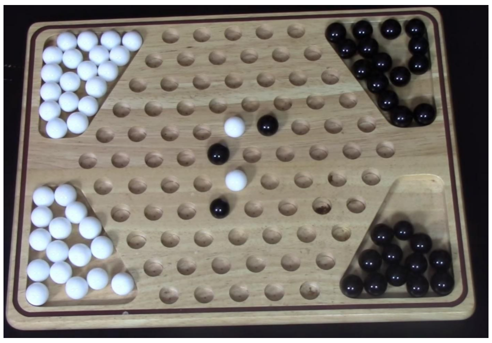
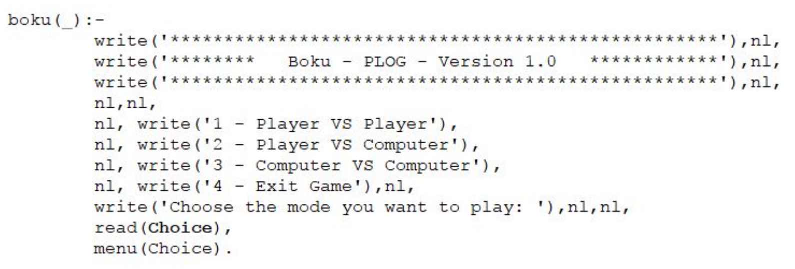


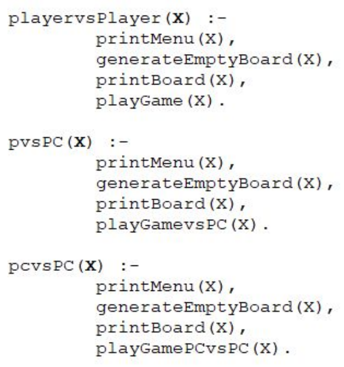
Figura 6 - Tabuleiro a meio do jogo.

3.2 Visualização do Tabuleiro

De forma a visualizar o tabuleiro nos diferentes estados de jogo são usados os predicados boku/1, playervsPlayer/1, pvsPC/1 e pcvsPC/1, que chamam os predicados generateEmptyBoard/1 e printBoard/1 de forma a que o tabuleiro seja visualizado em cada modo de jogo.



```
boku(_):-
	write('****************************************************'),nl,
	write('********   Boku - PLOG - Version 1.0   ************'),nl,
	write('****************************************************'),nl,
	nl,nl,
	nl, write('1 - Player VS Player'),
	nl, write('2 - Player VS Computer'),
	nl, write('3 - Computer VS Computer'),
	nl, write('4 - Exit Game'),nl,
	write('Choose the mode you want to play: '),nl,nl,
	read(Choice),
	menu(Choice).
```



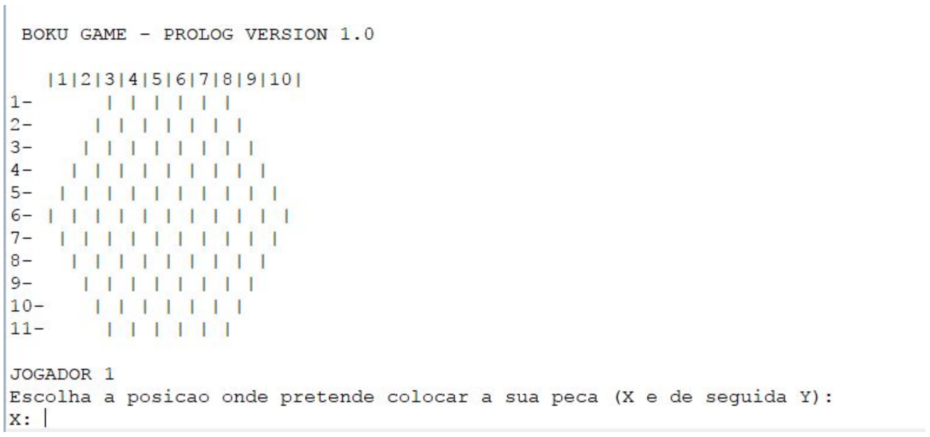
```
playervsPlayer(X) :-
	printMenu(X),
	generateEmptyBoard(X),
	printBoard(X),
	playGame(X).

pvsPC(X) :-
	printMenu(X),
	generateEmptyBoard(X),
	printBoard(X),
	playGamevsPC(X).

pcvsPC(X) :-
	printMenu(X),
	generateEmptyBoard(X),
	printBoard(X),
	playGamePCvsPC(X).
```

Figura 7 - Predicados boku/1, playervsPlayer/1, pvsPC/1 e pcvsPC/1.

Após a execução do programa em PROLOG deverá aparecer na consola o seguinte conteúdo:



```
 BOKU GAME - PROLOG VERSION 1.0

   |1|2|3|4|5|6|7|8|9|10|
1-       | | | | | |
2-      | | | | | | |
3-     | | | | | | | |
4-    | | | | | | | | |
5-   | | | | | | | | | |
6-  | | | | | | | | | | |
7-   | | | | | | | | | |
8-    | | | | | | | | |
9-     | | | | | | | |
10-     | | | | | | |
11-      | | | | | |

JOGADOR 1
Escolha a posicao onde pretende colocar a sua peca (X e de seguida Y):
X: |
```

Figura 7 - Predicados boku/1, playervsPlayer/1, pvsPC/1 e pcvsPC/1.

3.3 Jogadas válidas

Para verificar se o jogador pode realmente realizar a jogada que indicou será usado o predicado verificaCoordenadas/5. Onde Board1 será o tabuleiro actual, X e Y as coordenadas da peça, Player qual o jogador e Board2 o tabuleiro resultante após a jogada. Caso o jogador introduza uma jogada que não é válida ele perde o seu turno.

verifyCoordenates(Board1, Xpos, Ypos,Player, Board2):-

3.4 Execução de jogadas

Para efectuar a jogada que o jogador indicou, após verificação da mesma, será usado o predicado setPieceAt/5. Onde Board1 será o tabuleiro actual, X e Y as coordenadas da peça, Board2 o tabuleiro resultante e Piece a peça a colocar (W ou B).

setPieceAt(Board1, Xpos, Ypos, Board2, Piece) :-

Para saber qual a peça que está numa determinada coordenada será usado o predicado returnPiece/4. Onde Board1 será a o tabuleiro actual, X e Y as coordenadas da peça, Board2 o tabuleiro resultante e Piece será 'W' ou 'B'.

returnPieceAt(Board, X, Y, Piece) :-

3.5 Avaliação do Tabuleiro

Para verificar se uma jogada dá origem a uma situação de captura o predicado isCapturePlay\4 é invocado. Este predicado invoca dois predicados que verificam se a captura ocorreu na diagonal ou na horizontal, verifyCaptureDiagonals\4 e verifyCaptureHorizaontal\4 respectivamente. Estes predicados utilizam as coordenadas da última jogada para analisar as diagonais e a linha onde a peça foi colocada, de forma a verificar se duas peças da cor oposta à peça agora colocada estão agora rodeadas por esta e por outra peça da mesma cor. Nestes predicados Board é o tabuleiro atual, X e Y as coordenadas da peça colocada na última jogada e Board2 o tabuleiro resultante.

verifyCaptureDiagonals(Board,X,Y,Board2):-

verifyCaptureHorizontal(Board,X,Y,Board2):-

Quando uma situação de captura é reconhecida o predicado de changePieceAtCapture\8 é invocado. Neste predicado o jogador que realizou a captura introduz qual peça das duas rodeadas deseja capturar, e a escolhida é então retirada do tabuleiro. Aqui Board também representa o tabuleiro atual, X1 e Y1 são as coordenadas de uma das peças cuja captura é possível realizar, X2 e Y2 as coordenadas da outra peça, Board2 o tabuleiro resultante e Piece é a peça colocada na jogada que realizou a captura.

changePieceAtCapture(Board1,X1,Y1,X2,Y2,Board2, Piece) :-

Durante a verificação de uma linha ou de uma diagonal são invocados predicados auxiliares para facilitar a realização da tarefa. Nestes predicados Board representa o tabuleiro, X e Y representa a peça de momento a ser avaliada, PieceAnterior representa a peça estudada anteriormente, Board2 representa o tabuleiro resultante, Piece representa a peça que foi colocada na última jogada, X1, Y1, X2 e Y2 são as coordenadas das peças cuja captura é possível e Y representa a linha a ser avaliada de momento. A estrutura destes predicados é representada genericamente abaixo usando “N” e “K” como variáveis numéricas.

verifyCaptureHorizontalN(Board,X,Y,Board2):-

verifyCaptureHorizontalNauxK(Board,X,Y,PieceAnterior,Board2):-

verifyCaptureDiagonalN(Board,X,Y,Board2):-

verifyCaptureDiagonalNauxK(Board,X,Y,PieceAnterior,Board2):-

pieceAtcaptureAux(Board1,X1,Y1,X2,Y2,Board2, Piece):-

pieceAtcaptureHorizontalAux(Board1,X1,X2,Y,Board2, Piece):-

3.6 Final do Jogo

A verificação de situação de vitória é realizada pelo predicado isWinCondition\3. Este predicado invoca outros dois, verifyWinDiagonals\3 e verifyWinHorizontal\3 que verificam se a condição de vitória ocorreu na diagonal ou na horizontal, respectivamente, a partir da peça colocada na última jogada. Em todos estes predicados Board é o tabuleiro actual e X e Y são as coordenadas da última peça colocada.

isWinCondition(Board,X,Y) :-

verifyWinDiagonals(Board,X,Y):-

verifyWinHorizontal(Board, X,Y):-

Durante a verificação de uma linha ou de uma diagonal são invocados predicados auxiliares para facilitar a realização da tarefa. Aqui Board representa o tabuleiro, X e Y representa a peça de momento a ser avaliada e PieceAnterior representa a peça estudada anteriormente. A estrutura destes predicados é representada genericamente abaixo usando "N" e "K" como variáveis numéricas.

verifyWinHorizontalN(Board,X,Y):-

verifyWinHorizontalNauxK(Board, X, Y, PieceAnterior):-

verifyWinDiagonalN(Board, X, Y):-

verifyWinDiagonalNauxK(Board, X, Y, PieceAnterior):-

3.7 Jogada de computador

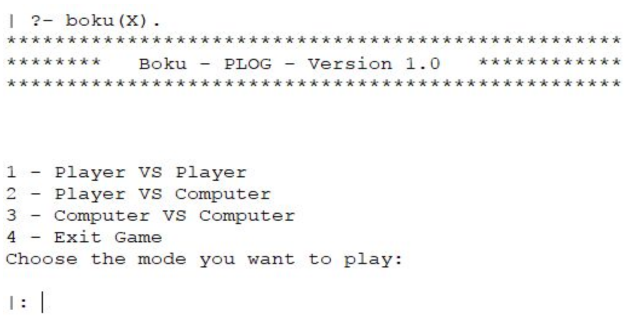
Devido às inúmeras possibilidades de situações de jogo e estratégias possíveis apenas foi desenvolvido um nível de dificuldade para o modo computador. Este modo usa o predicado random\3 para gerar dois valores dentro do tabuleiro, verifica se esta é uma posição válida e caso haja situação de captura escolhe também aleatoriamente a peça que captura.

Este algoritmo usa os predicados generateRandomMove\2, que retorna uma jogada válida na posição de coordenadas Xpos e Ypos, e variantes dos predicados de verificação de captura, cuja diferença vem do facto que não será lido nenhum input, mas será escolhido aleatoriamente uma peça para capturar.

generateRandomMove(Xpos,Ypos) :-

4. Interface

Para iniciar o jogo o utilizador deve invocar o predicado boku/1 atribuindo uma variável à sua escolha. Após a invocação do predicado é apresentado na consola o menu de início de jogo. Após a escolha do modo de jogo por parte do utilizador o jogo começa apresentando o tabuleiro de jogo na consola de forma a se poder jogar (figura 9).

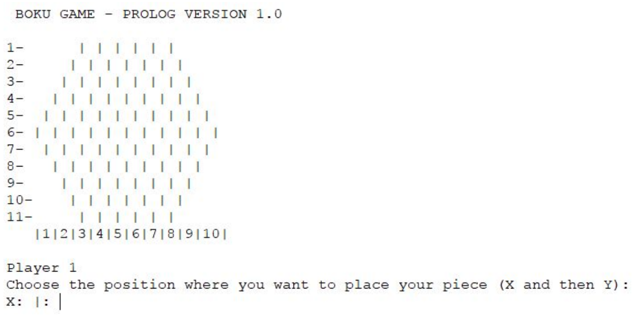


```
| ?- boku(X).
**************************************************
********    Boku - PLOG - Version 1.0    ************
**************************************************



1 - Player VS Player
2 - Player VS Computer
3 - Computer VS Computer
4 - Exit Game
Choose the mode you want to play:

|: 
```



```
 BOKU GAME - PROLOG VERSION 1.0

1-      | | | | | |
2-     | | | | | | |
3-    | | | | | | | |
4-   | | | | | | | | |
5-  | | | | | | | | | |
6- | | | | | | | | | | |
7-  | | | | | | | | | |
8-   | | | | | | | | |
9-    | | | | | | | |
10-    | | | | | | |
11-     | | | | | |
   |1|2|3|4|5|6|7|8|9|10|

Player 1
Choose the position where you want to place your piece (X and then Y):
X: |: 
```

Figura 9 - Menu Inicial do Jogo e Visualização de tabuleiro de jogo.

Conclusão

A realização do jogo Boku em Plog exigiu bastante tempo, não necessariamente devido à sua complexidade em termos de jogo, mas mais devido ao facto de a linguagem em si necessitar uma atenção invulgar perante as diversas situações e fases de desenvolvimento.

O código em si torna-se um pouco repetitivo, pois cada situação e possibilidade tinha que ser considerada e representada tornado o código bastante extenso. Devido a isto, a localização, resolução e por vezes mesmo reconhecimento de diversos bugs encontrados durante o processo de desenvolvimento foi um enorme desafio.

A implementação utilizada no modo computador poderia ser melhorada no sentido de esta ser mais inteligente, visto que apenas escolhe valores random para as posições do tabuleiro.

Mesmo com estas dificuldades o trabalho foi realizado a tempo e com confiança no resultado final e este simples mas muito interessante jogo foi chamado à nossa atenção.

Em suma, programar em Plog foi um desafio, mas estamos confiantes no resultado final e gostaríamos de convidar quem desejar a experimentar este belo jogo de estratégia.

Bibliografia

http://homepages.di.fc.ul.pt/~jpn/gv/boku.htm

https://en.wikipedia.org/wiki/B%C5%8Dku

http://boku.bandoodle.co.uk/rules.php

Anexo A - Boku.pl

```
/*******************************
   * para correr o jogo e' consultar este ficheiro e fazer boku(X). *
   *******************************/


:- include('print.pl').
:- include('board.pl').
:- include('capture.pl').
:- include('diagonals.pl').
:- include('diagonalsPC.pl').
:- include('diagonals_win.pl').
:- include('win_conditions.pl').

?- use_module(library(system)).
?- use_module(library(random)).

boku(_):-
        write('**************************************************'),nl,
        write('********   Boku - PLOG - Version 1.0   ************'),nl,
        write('**************************************************'),nl,
        nl,nl,
        nl, write('1 - Player VS Player'),
        nl, write('2 - Player VS Computer'),
        nl, write('3 - Computer VS Computer'),
        nl, write('4 - Exit Game'),nl,
        write('Choose the mode you want to play: '),nl,nl,
        read(Choice),
        menu(Choice).

menu(Choice):- Choice == 1,
	playervsPlayer(_).
menu(Choice):- Choice == 2,
	pvsPC(_).
menu(Choice):- Choice == 3,
	pcvsPC(_).
menu(Choice):- Choice == 4,
	exit(_).

playervsPlayer(X) :-
	printMenu(X),
	generateEmptyBoard(X),
	printBoard(X),
	playGame(X).

pvsPC(X) :-
	printMenu(X),
	generateEmptyBoard(X),
	printBoard(X),
	playGamevsPC(X).

pcvsPC(X) :-
	printMenu(X),
	generateEmptyBoard(X),
	printBoard(X),
	playGamePCvsPC(X).

exit(_) :- nl,nl,write('See you later!!!!'),nl,nl.

%se o player1 ganhar faz endgame, se nao ganhar verifica se o player dois ganha ou nao.
%se o player2 ganhar acaba, se nao ganhar chama de novo
playGame(X) :- p1(P1xpos, P1ypos),
	playerturn(X, X1, P1xpos, P1ypos, 'W'),
	printBoard(X1),
	(isWinCondition(X1,P1xpos,P1ypos),nl, endGame(_);
         p2(P2xpos, P2ypos),
         playerturn(X1, X2, P2xpos,P2ypos, 'B'),
         printBoard(X2),
         (isWinCondition(X2,P2xpos,P2ypos),nl,endGame(_);
          \+isWinCondition(X2,P2xpos,P2ypos), playGame(X2)) ).

%player vs pc
playGamevsPC(X) :- p1(P1xpos, P1ypos),
        playerturn(X, X1, P1xpos, P1ypos, 'W'),
        printBoard(X1),
        (isWinCondition(X1,P1xpos,P1ypos),nl, endGame(_);
         p2PC(P2xpos, P2ypos),
         pcTurn(X1, X2, P2xpos,P2ypos, 'B'),
         printBoard(X2),
         (isWinCondition(X2,P2xpos,P2ypos),nl,endGame(_);
          \+isWinCondition(X2,P2xpos,P2ypos), playGamevsPC(X2)) ).

%pc vs pc
playGamePCvsPC(X) :- p1PC(P1xpos, P1ypos),
        pcTurn(X, X1, P1xpos, P1ypos, 'W'),
        printBoard(X1),
        (isWinCondition(X1,P1xpos,P1ypos),nl, endGame(_);
         p2PC(P2xpos, P2ypos),
         pcTurn(X1, X2, P2xpos,P2ypos, 'B'),
         printBoard(X2),
         (isWinCondition(X2,P2xpos,P2ypos),nl,endGame(_);
          \+isWinCondition(X2,P2xpos,P2ypos), playGamePCvsPC(X2)) ).



p1(P1xpos, P1ypos):-  write('\nPlayer 1\n'),
        write('Choose the position where you want to place your piece (X and then Y):\n'),
        write('X: '),
        read(Xpos),
        P1xpos is Xpos,
        write('\n'),
        write('Y: '),
        read(Ypos),
        P1ypos is Ypos,
        write('\n').

p2(P2xpos, P2ypos):- write('\nPlayer 2\n'),
        write('Choose the position where you want to place your piece (X and then Y):\n'),
        write('X: '),
        read(Xpos),
        P2xpos is Xpos,
        write('\n'),
        write('Y: '),
        read(Ypos),
        P2ypos is Ypos,
        write('\n').



p1PC(P2xpos, P2ypos):- write('\nPlayer 1 - Computer\n'),
        write('Player 1 is choosing his move!\n'),
        sleep(2),
        generateRandomMove(P2xpos,P2ypos),nl,
        write('X: '),
        write(P2xpos),nl,
        write('y: '),
        write(P2ypos),nl,
        sleep(1),
        write('\n').

p2PC(P2xpos, P2ypos):- write('\nPlayer 2 - Computer\n'),
        write('Player 2 is choosing his move!\n'),
        sleep(2),
        generateRandomMove(P2xpos,P2ypos),nl,
        write('X: '),
        write(P2xpos),nl,
        write('y: '),
        write(P2ypos),nl,
        sleep(1),
        write('\n').

generateRandomMove(Xpos,Ypos) :-
        random(1, 11, Ypos),
        randomX(Ypos,Xpos).

randomX(Ypos,Xpos) :- (Ypos == 1,random(1, 5, Xpos));Ypos == 11,random(1, 5, Xpos).
randomX(Ypos,Xpos) :- (Ypos == 2,random(1, 6, Xpos));Ypos == 10,random(1, 6, Xpos).
randomX(Ypos,Xpos) :- (Ypos == 3,random(1, 7, Xpos));Ypos == 9,random(1, 7, Xpos).
randomX(Ypos,Xpos) :- (Ypos == 4,random(1, 8, Xpos));Ypos == 8,random(1, 8, Xpos).
randomX(Ypos,Xpos) :- (Ypos == 5,random(1, 9, Xpos));Ypos == 7,random(1, 9, Xpos).
randomX(Ypos,Xpos) :- Ypos == 6,random(1, 10, Xpos).

playerturn(Board1, Board3, Xpos, Ypos, Player) :-
        verifyCoordenates(Board1, Xpos, Ypos,Player, Board2),
        isCapturePlay(Board2, Xpos, Ypos, Board3).

pcTurn(Board1, Board3, Xpos, Ypos, Player) :-
        verifyCoordenates(Board1, Xpos, Ypos,Player, Board2),
        isCapturePlayPC(Board2, Xpos, Ypos, Board3).



endGame(_):-
        write('****************************************************'),nl,
        write('*************** Boku Version 1.0 *******************'),nl,
        write('****************************************************'),nl,
        nl,nl,nl.
```

12 de Novembro de 2017

Anexo B - Board.pl

```
/* Empty board generation */
generateEmptyBoard(X) :-
        emptyBoardAux(L1, 1),
        emptyBoardAux(L2, 2),
        emptyBoardAux(L3, 3),
        emptyBoardAux(L4, 4),
        emptyBoardAux(L5, 5),
        emptyBoardAux(L6, 6),
        emptyBoardAux(L7, 7),
        emptyBoardAux(L8, 8),
        emptyBoardAux(L9, 9),
        emptyBoardAux(L10, 10),
        emptyBoardAux(L11, 11),
        emptyBoardAux(L12, 12),
        append([], [L1], X1),
        append(X1, [L2], X2),
        append(X2, [L3], X3),
        append(X3, [L4], X4),
        append(X4, [L5], X5),
        append(X5, [L6], X6),
        append(X6, [L7], X7),
        append(X7, [L8], X8),
        append(X8, [L9], X9),
        append(X9, [L10], X10),
        append(X10, [L11], X11),
        append(X11, [L12], X).

emptyBoardAux(L, N) :-
        N == 1,
        L = [' ',' ',' ',' ',' ','E','E','E','E','E'].
emptyBoardAux(L, N) :-
        N == 2,
        L =[' ',' ',' ',' ',' ',' ','E','E','E','E'].
emptyBoardAux(L, N) :-
        N == 3,
        L =[' ',' ',' ',' ',' ',' ',' ','E','E','E'].
emptyBoardAux(L, N) :-
        N == 4,
        L =[' ',' ',' ',' ',' ',' ',' ',' ','E','E'].
emptyBoardAux(L, N) :-
        N == 5,
        L =[' ',' ',' ',' ',' ',' ',' ',' ',' ','E'].
emptyBoardAux(L, N) :-
        N == 6,
        L =[' ',' ',' ',' ',' ',' ',' ',' ',' ',' ','E'].
emptyBoardAux(L, N) :-
        N == 7,
        L =[' ',' ',' ',' ',' ',' ',' ',' ',' ','E'].
emptyBoardAux(L, N) :-
        N == 8,
        L =[' ',' ',' ',' ',' ',' ',' ',' ','E','E'].
emptyBoardAux(L, N) :-
        N == 9,
        L =[' ',' ',' ',' ',' ',' ',' ', 'E','E','E'].
emptyBoardAux(L, N) :-
        N == 10,
        L =[' ',' ',' ',' ',' ',' ','E','E','E','E'].
emptyBoardAux(L, N) :-
        N == 11,
        L =[' ',' ',' ',' ',' ','E','E','E','E','E'].
emptyBoardAux(L, N) :-
        N == 12,
        L =['1','2','3','4','5','6','7','8','9','10'].
```

```
/********************************** verificação das jogadas  *************************/


%Para verificar se o jogador pode realmente realizar a jogada e realiza se tal for possivel

verifyCoordenates(Board1, Xpos, Ypos, Player, Board2) :- returnPieceAt(Board1,Xpos, Ypos, Piece), %retorna qual a peça que está no tab

    Piece == ' ',

    write('\nPredicate verifyCoordenates/5 say that is an empty spot.'),

    write('\nYou can play to this position.'),

    write('\n'),

    write('\n'),

    setPieceAt(Board1, Xpos,Ypos, Board2, Player).




verifyCoordenates(Board1, Xpos, Ypos, Piece, Board2) :- returnPieceAt(Board1,Xpos, Ypos, Pieceat), %retorna qual a peça que está no tab

    Pieceat \= ' ',

    write('\nPredicate verifyCoordenates/5 say that there is a piece in that position: '),

    write(Pieceat),

    write('\n You can not play to this position and you lose your turn!!!!!'),

    write('\n'),

    write('\n'),

    setPieceAt(Board1, Xpos,Ypos, Board2, Piece).
/*******************************************************************************************/


/*******************************mudar peça no tabuleiro************************************/


%Para efectuar a jogada que o jogador indicou, após verificação da mesma
```

```
setPieceAt(Board1, Xpos, Ypos, Board2, Piece) :- changePiece(Board1, 1, Xpos, Ypos, Piece, Board2).


%recebe o tabuleiro de jogo e isola a coluna pretendida.
changePiece([B|Bs], N, X, Y, Piece, Board2) :-
	N == Y,
	  changeLinePiece(B, 1, X, Piece, BoardAux),% chama o changeline com a cabeça da lista que e' a coluna
selected
	append([BoardAux], Bs, Board2). % board 2 e' Bs (colunas para a frente) com a cabeça vazia.


changePiece([B|Bs], N, X, Y, Piece, Board2) :-
	N < Y,
	N2 is N + 1,
	 changePiece(Bs, N2, X, Y, Piece, BoardAux), %chamo com as restantes listas da lista (colunas) com o aux
vazio
	append([B], BoardAux, Board2). % guardo a lista nao alterada em Board2.


%percorre a linha e coloca a peça na posicao X tendo em conta a coluna (lista) escolhida em change piece
changeLinePiece([_|Ls], N, X, Piece, L2) :-
	N == X,
	append([Piece], Ls, L2). %coloca a peça na cabeça da lista, que corresponde à posição X pretendida


changeLinePiece([L|Ls], N, X, Piece, L2) :-
	N < X,
	N2 is N + 1,
	changeLinePiece(Ls, N2, X, Piece, Laux),%chama com os restantes elementos da linha
	append([L], Laux, L2). %guarda o elemento da posição n em L2. (guarda os que nao sao alterados)

/***************************************************************************************
*********/


/************************************saber peça no tabuleiro
************************************/
%Para saber qual a peça que está numa determinada coordenada
%codigo de outra pessoa. adaptar estes predicados. contudo parece que isto funciona assim
```

```
returnPieceAt(Board, X, Y, Piece) :- boardLine(Board, 1, Y, Line), % começa no 1 por causa da linha dos
numeros
        linePiece(Line, 1, X, Piece). %tem de ser com 1 para dar certo na linha. devido ao changelinepice ser 1
tambem


%recebe o tabuleira e isola a lista que é referente à coluna do tabuleiro
boardLine([B|_], N, Y, Line) :-
      N == Y,
      append([], B, Line). % se for a cabeça da lista de listas (primeira lista) guarda a lista em line
%seleciona a lista que queremos basicamente


boardLine([_|Bs], N, Y, Line) :-
      N < Y,
      N2 is N + 1,
      boardLine(Bs, N2, Y, Line). %percorre a lista de listas ate que a lista que queremos esteja À cabeça

%recebe a linha ja escolhida em boardline e retorna a peça que está na posição X.
linePiece([L|_], N, X, Piece) :-
      N == X,
      Piece = L. %se a posição x for a cabeça da lista a peça e' a cabeça da lista


linePiece([_|Ls], N, X, Piece) :-
      N < X,
      N2 is N + 1,
      linePiece(Ls, N2, X, Piece). %percorre a lista até que a posição X seja a cabeça da lista

/*****************************************************************************************
*************/
```

Anexo C - capture.pl

```
/***************************************                                        captura
**************************************/

%Para verificar se a jogada é uma jogada em que pode ser feita uma captura
isCapturePlay(Board,X,Y,Board2) :- verifyCaptureDiagonals(Board, X,Y,Board1),
        Board \= Board1,
        Board1 = Board2.
%Para verificar se a jogada é uma jogada em que pode ser feita uma captura
isCapturePlay(Board,X,Y,BoardR) :- verifyCaptureDiagonals(Board, X,Y,Board1),
        Board == Board1,
        verifyCaptureHorizontal(Board,X,Y,BoardR).

/*****/
%%Para alterar a peça (para vazio) quando é feita a captura
changePieceAtCapture(Board1,X1,Y1,X2,Y2,Board2, Piece) :-
        pieceAtcaptureAux(Board1,X1,Y1,X2,Y2,Board2, Piece).

pieceAtcaptureAux(Board1,X1,Y1,X2,Y2,Board2, Piece):- Piece == 'W',
        write('\n[Player 1]\n'),
        write('\nCoordenates of the pieces that you can remove:\n'),
        write('X: '),write(X1),write(' - Y: '),write(Y1),nl,
        write('X: '),write(X2),write(' - Y: '),write(Y2),nl,
        write('\n[Player 1]\n'),
        write('Choose the piece of your opponent you want to remove\n'),
        write('X: '),
        read(Xpos),
        write('\n'),
        write('Y: '),
        read(Ypos),
        write('\n'),
        changePieceAtCaptureAux(Board1,X1,Y1,X2,Y2,Board2,Piece,Xpos,Ypos).

pieceAtcaptureAux(Board1,X1,Y1,X2,Y2,Board2, Piece) :- Piece == 'B',
        write('\n[Player 2]\n'),
        write('Coordenates of the pieces that you can remove:\n'),
        write('X: '),write(X1),write(' - Y: '),write(Y1),nl,
        write('X: '),write(X2),write(' - Y: '),write(Y2),nl,
        write('\n[Player 2]\n'),
        write('Choose the piece of your opponent you want to remove\n'),
        write('X: '),
        read(Xpos),
        write('\n'),
        write('Y: '),

        read(Ypos),

        write('\n'),

        changePieceAtCaptureAux(Board1,X1,Y1,X2,Y2,Board2,Piece,Xpos, Ypos).


changePieceAtCaptureAux(Board1,X1,Y1,_,_,Board2,              Piece,Xpos,Ypos)              :-
returnPieceAt(Board1,X1,Y1,Pieceat),

        X1 == Xpos,

        Y1 == Ypos,

        Piece \= Pieceat,

        setPieceAt(Board1,Xpos,Ypos,Board2,' ').


changePieceAtCaptureAux(Board1,_,_,X2,Y2,Board2,        Piece,Xpos,        Ypos)        :-
returnPieceAt(Board1,X2,Y2,Pieceat),

        X2 == Xpos,

        Y2 == Ypos,

        Piece \= Pieceat,

        setPieceAt(Board1, Xpos, Ypos, Board2, ' ').


changePieceAtCaptureAux(Board1,X1,_,_,_,Board2,_,Xpos,_) :-

        X1 \= Xpos,

        Board1 = Board2,
        write('Wrong coordenates. You lost your turn.  X1 diferente\n'),
        write('\n').

changePieceAtCaptureAux(Board1,_,_,X2,_,Board2,_,Xpos,_) :-
        X2 \= Xpos,
        Board1 = Board2,
        write('Wrong coordenate. You lost your turn.  X2 diferente\n'),
        write('\n').

changePieceAtCaptureAux(Board1,X1,Y1,_,_,Board2,_,Xpos,Ypos) :-
        X1 == Xpos,
        Y1 \= Ypos,
        Board1 = Board2,
        write('Wrong coordenate. You lost your turn.  y1 diferente\n'),
        write('\n').

changePieceAtCaptureAux(Board1,_,_,X2,Y2,Board2,_,Xpos,Ypos) :-
        X2 == Xpos,
        Y2 \= Ypos,
        Board1 = Board2,
        write('Wrong coordenate. You lost your turn.  y2 diferente\n'),
        write('\n').
```

```
/*******************/


%%Para alterar a peça (para vazio) quando é feita a captura na horizontal
changePieceAtCaptureHorizontal(Board1,X1,X2,Y,Board3, Piece) :-
        pieceAtcaptureHorizontalAux(Board1,X1,X2,Y,Board3, Piece).


pieceAtcaptureHorizontalAux(Board1,X1,X2,Y,Board3, Piece):- Piece == 'W',
        write('\n[Player 1]\n'),
        write('\nCoordenates to remove\n'),
        write(X1),write(Y),write('\n'),
        write(X2),write(Y),write('\n'),
        write('\n[Player 1]\n'),
        write('Choose the piece of your opponent you want to remove\n'),
        write('X: '),
        read(Xpos),
        write('\n'),
        write('Y: '),
        read(Ypos),
        write('\n'),
        changePieceAtCaptureHorizontalAux(Board1,X1,X2,Y,Board3,Piece,Xpos,Ypos).
```

```
pieceAtcaptureHorizontalAux(Board1,X1,X2,Y,Board3, Piece) :- Piece == 'B',

        write('\n[Player 2]\n'),

        write('Choose the piece of your opponent you want to remove\n'),

        write('X: '),

        read(Xpos),

        write('\n'),

        write('Y: '),

        read(Ypos),

        write('\n'),

        changePieceAtCaptureHorizontalAux(Board1,X1,X2,Y,Board3,Piece,Xpos, Ypos).


changePieceAtCaptureHorizontalAux(Board1,X1,_,Y,Board3,        Piece,Xpos,Ypos)        :-
returnPieceAt(Board1,X1,Y,Pieceat),

        X1 == Xpos,

        Y == Ypos,

        Piece \= Pieceat,

        setPieceAt(Board1,Xpos,Ypos,Board3,' ').



changePieceAtCaptureHorizontalAux(Board1,_,X2,Y,Board3,      Piece,Xpos,      Ypos)      :-
returnPieceAt(Board1,X2,Y,Pieceat),

        X2 == Xpos,
        Y == Ypos,
        Piece \= Pieceat,
        setPieceAt(Board1,Xpos,Ypos,Board3,' ').

changePieceAtCaptureHorizontalAux(Board1,X1,_,_,Board2,_,Xpos,_) :-
        X1 \= Xpos,
        Board1 = Board2,
        write('Wrong coordenates. You lost your turn.  X1 diferente\n'),
        write('\n').

changePieceAtCaptureHorizontalAux(Board1,_,X2,_,Board2,_,Xpos,_) :-
        X2 \= Xpos,
        Board1 = Board2,
        write('Wrong coordenate. You lost your turn.  X2 diferente\n'),
        write('\n').

changePieceAtCaptureHorizontalAux(Board1,X1,_,Y,Board2,_,Xpos,Ypos) :-
        X1 == Xpos,
        Y \= Ypos,
        Board1 = Board2,
        write('Wrong coordenate. You lost your turn.  y1 diferente\n'),
        write('\n').
```

```
changePieceAtCaptureHorizontalAux(Board1,_,X2,Y,Board2,_,Xpos,Ypos) :-
        X2 == Xpos,
        Y \= Ypos,
        Board1 = Board2,
        write('Wrong coordenate. You lost your turn.  y2 diferente\n'),
        write('\n').



/****** pc mode *******/
/**************************************                          captura
*************************************/
%Para verificar se a jogada é uma jogada em que pode ser feita uma captura
isCapturePlayPC(Board,X,Y,Board2) :- verifyCaptureDiagonalsPC(Board, X,Y,Board1),
        Board \= Board1,
        Board1 = Board2.
%Para verificar se a jogada é uma jogada em que pode ser feita uma captura
isCapturePlayPC(Board,X,Y,BoardR) :- verifyCaptureDiagonalsPC(Board, X,Y,Board1),
        Board == Board1,
        verifyCaptureHorizontalPC(Board,X,Y,BoardR).
```

```
/*****pc mode******/

changePieceAtCapturePC(Board1,X1,Y1,X2,Y2,Board2, Piece) :-
	pieceAtcapturePCAux(Board1,X1,Y1,X2,Y2,Board2, Piece).

pieceAtcapturePCAux(Board1,X1,Y1,X2,Y2,Board2, Piece):- Piece == 'W',
	write('\n[Player 1]\n'),
	write('\nCoordenates of the pieces that you can remove:(xy)\n'),
	write('X: '),write(X1),write(' - Y: '),write(Y1),nl,
	write('X: '),write(X2),write(' - Y: '),write(Y2),nl,
	sleep(2),
	write('Computer chose the part he wants to remove\n'),
	write('X: '),write(X1),write(' - Y: '),write(Y1),nl,
	write('\n'),
	setPieceAt(Board1,X1,Y1,Board2,' ').

pieceAtcapturePCAux(Board1,X1,Y1,X2,Y2,Board2, Piece) :- Piece == 'B',
	write('\n[Player 2]\n'),
	write('\nCoordenates of the pieces that you can remove:\n'),
	write('X: '),write(X1),write(' - Y: '),write(Y1),nl,
	write('X: '),write(X2),write(' - Y: '),write(Y2),nl,
	sleep(2),
        write('Computer chose the part he wants to remove\n'),

        write('X: '),write(X1),write(' - Y: '),write(Y1),nl,

        write('\n'),

        setPieceAt(Board1,X1,Y1,Board2,' ').


%%Para alterar a peça (para vazio) quando é feita a captura na horizontal

changePieceAtCaptureHorizontalPC(Board1,X1,X2,Y,Board3, Piece) :-

        pieceAtcaptureHorizontalPCAux(Board1,X1,X2,Y,Board3, Piece).


pieceAtcaptureHorizontalPCAux(Board1,X1,X2,Y,Board2, Piece):- Piece == 'W',

        write('\n[Player 1]\n'),

        write('\nCoordenates of the pieces that you can remove:\n'),

        write('X: '),write(X1),write(' - Y: '),write(Y),nl,

        write('X: '),write(X2),write(' - Y: '),write(Y),nl,

        sleep(2),

        write('Computer chose the piece he wants to remove\n'),

        write('X: '),write(X1),write(' - Y: '),write(Y),nl,

        write('\n'),

        setPieceAt(Board1,X1,Y,Board2,' ').


pieceAtcaptureHorizontalPCAux(Board1,X1,X2,Y,Board2, Piece) :- Piece == 'B',

        write('\n[Player 2]\n'),
	write('\nCoordenates of the pieces that you can remove:\n'),
	write('X: '),write(X1),write(' - Y: '),write(Y),nl,
	write('X: '),write(X2),write(' - Y: '),write(Y),nl,
	sleep(2),
	write('Computer chose the piece he wants to remove\n'),
	write('X: '),write(X1),write(' - Y: '),write(Y),nl,
	write('\n'),
	setPieceAt(Board1,X1,Y,Board2,' ').


/******************* parte de cima do tabuleiro, VERIFICA QUE DIAGONAIS USAR*********************/


%linha1 ate linha 3
verifyCaptureDiagonals(Board,X,Y,Board2):- Y < 4,
	verifyCaptureDiagonal1(Board,X,Y,Board1),
	Board \= Board1,
	Board1 = Board2.
%linha1 ate linha 3
verifyCaptureDiagonals(Board,X,Y,Board2):- Y < 4,
	verifyCaptureDiagonal1(Board,X,Y,Board1),
	Board == Board1,
        verifyCaptureDiagonal3(Board,X,Y,Board2).
%linha4
verifyCaptureDiagonals(Board,X,Y,Board2):- Y == 4,
        X == 1,
        verifyCaptureDiagonal2(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = Board2.
%linha4
verifyCaptureDiagonals(Board,X,Y,Board2):- Y == 4,
        X == 1,
        verifyCaptureDiagonal2(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal3(Board,X,Y,Board2).
%%linha4
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 4,
        X > 1,
        X < 4,
        verifyCaptureDiagonal1(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal2(Board,X,Y,Board2),
        Board \= Board2,
        Board2 = BoardR.
```

```
%%linha4
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 4,
    X > 1,
    X < 4,
    verifyCaptureDiagonal1(Board,X,Y,Board1),
    Board \= Board1,
    Board1 = BoardR.
%%linha4
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 4,
    X > 1,
    X < 4,
    verifyCaptureDiagonal1(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal2(Board,X,Y,Board2),
    Board == Board2,
    verifyCaptureDiagonal3(Board,X,Y,BoardR).
%%linha4
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 4,
    X > 3,
    X < 6,
    verifyCaptureDiagonal1(Board,X,Y,Board1),
	Board == Board1,
	verifyCaptureDiagonal2(Board,X,Y,Board2),
	Board == Board2,
	verifyCaptureDiagonal3(Board,X,Y,Board3),
	Board \= Board3,
	Board3 = BoardR.
%%linha4
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 4,
	X > 3,
	X < 6,
	verifyCaptureDiagonal1(Board,X,Y,Board1),
	Board == Board1,
	verifyCaptureDiagonal2(Board,X,Y,Board2),
	Board \= Board2,
	Board2 = BoardR.
%%linha4
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 4,
	X > 3,
	X < 6,
	verifyCaptureDiagonal1(Board,X,Y,Board1),
	Board \= Board1,
	Board1 = BoardR.
```

```
%%linha4
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 4,
	X > 3,
	X < 6,
	verifyCaptureDiagonal1(Board,X,Y,Board1),
	Board == Board1,
	verifyCaptureDiagonal2(Board,X,Y,Board2),
	Board == Board2,
	verifyCaptureDiagonal3(Board,X,Y,Board3),
	Board == Board3,
	verifyCaptureDiagonal4(Board,X,Y,BoardR).
%linha4
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 4,
	X > 5,
	X < 8,
	verifyCaptureDiagonal1(Board,X,Y,Board1),
	Board == Board1,
	verifyCaptureDiagonal2(Board,X,Y,Board2),
	Board \= Board2,
	Board2 = BoardR.
%%linha4
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 4,
    X > 5,
    X < 8,
    verifyCaptureDiagonal1(Board,X,Y,Board1),
    Board \= Board1,
    Board1 = BoardR.
%linha4
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 4,
    X > 5,
    X < 8,
    verifyCaptureDiagonal1(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal2(Board,X,Y,Board2),
    Board == Board2,
    verifyCaptureDiagonal4(Board,X,Y,BoardR).
%linha4
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 4,
    X == 8,
    verifyCaptureDiagonal1(Board,X,Y,Board1),
    Board \= Board1,
    Board1 = BoardR.
%linha4
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 4,
	X == 8,
	verifyCaptureDiagonal1(Board,X,Y,Board1),
	Board == Board1,
	verifyCaptureDiagonal4(Board,X,Y,BoardR).
%linha5
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 5,
	X < 3,
	verifyCaptureDiagonal2(Board,X,Y,Board1),
	Board \= Board1,
	Board1 = BoardR.
%linha5
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 5,
	X < 3,
	verifyCaptureDiagonal2(Board,X,Y,Board1),
	Board == Board1,
	verifyCaptureDiagonal3(Board,X,Y,BoardR).
%%linha5
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 5,
	X == 3,
	verifyCaptureDiagonal1(Board,X,Y,Board1),
	Board == Board1,
	verifyCaptureDiagonal2(Board,X,Y,Board2),
    Board \= Board2,

    Board2 = BoardR.

%%linha5

verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 5,

    X == 3,

    verifyCaptureDiagonal1(Board,X,Y,Board1),

    Board \= Board1,

    Board1 = BoardR.

%%linha5

verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 5,

    X == 3,

    verifyCaptureDiagonal1(Board,X,Y,Board1),

    Board == Board1,

    verifyCaptureDiagonal2(Board,X,Y,Board2),

    Board == Board2,

    verifyCaptureDiagonal3(Board,X,Y,BoardR).

%%linha5

verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 5,

    X > 3,

    X < 7,

    verifyCaptureDiagonal1(Board,X,Y,Board1),

    Board == Board1,
    verifyCaptureDiagonal2(Board,X,Y,Board2),
    Board == Board2,
    verifyCaptureDiagonal3(Board,X,Y,Board3),
    Board \= Board3,
    Board3 = BoardR.
%%linha5
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 5,
    X > 3,
    X < 7,
    verifyCaptureDiagonal1(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal2(Board,X,Y,Board2),
    Board \= Board2,
    Board2 = BoardR.
%%linha5
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 5,
    X > 3,
    X < 7,
    verifyCaptureDiagonal1(Board,X,Y,Board1),
    Board \= Board1,
    Board1 = BoardR.
%linha5
```

```
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 5,
    X > 3,
    X < 7,
    verifyCaptureDiagonal1(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal2(Board,X,Y,Board2),
    Board == Board2,
    verifyCaptureDiagonal3(Board,X,Y,Board3),
    Board == Board3,
    verifyCaptureDiagonal4(Board,X,Y,BoardR).
%linha5
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 5,
    X == 7,
    verifyCaptureDiagonal1(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal2(Board,X,Y,Board2),
    Board \= Board2,
    Board2 = BoardR.
%linha5
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 5,
    X == 7,
    verifyCaptureDiagonal1(Board,X,Y,Board1),
    Board \= Board1,
    Board1 = BoardR.
%linha5
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 5,
    X > 7,
    verifyCaptureDiagonal1(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal4(Board,X,Y,BoardR).
%linha5
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 5,
    X > 7,
    verifyCaptureDiagonal1(Board,X,Y,Board1),
    Board \= Board1,
    Board1 = BoardR.
%linha5
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 5,
    X == 7,
    verifyCaptureDiagonal1(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal2(Board,X,Y,Board2),
    Board == Board2,
    verifyCaptureDiagonal4(Board,X,Y,BoardR).
```

```
%%linha 6
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 6,
        X < 4,
        verifyCaptureDiagonal3(Board,X,Y,Board1),
        board \= 1,
        Board1 = BoardR.
%%linha 6
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 6,
        X < 4,
        verifyCaptureDiagonal3(Board,X,Y,Board1),
        board == Board1,
        verifyCaptureDiagonal2(Board,X,Y,BoardR).
%%linha 6
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 6,
        X > 3,
        X < 7,
        verifyCaptureDiagonal1(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%%linha 6
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 6,
        X > 3,
        X < 7,
        verifyCaptureDiagonal1(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal2(Board,X,Y,Board2),
        Board \= Board2,
        Board2 = BoardR.
%%linha 6
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 6,
        X > 3,
        X < 7,
        verifyCaptureDiagonal1(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal2(Board,X,Y,Board2),
        Board == Board2,
        verifyCaptureDiagonal3(Board,X,Y,Board3),
        Board \= Board3,
        Board3 = BoardR.
%%linha 6
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 6,
        X > 3,
        X < 7,
        verifyCaptureDiagonal1(Board,X,Y,Board1),
        Board == Board1,

        verifyCaptureDiagonal2(Board,X,Y,Board2),

        Board == Board2,

        verifyCaptureDiagonal3(Board,X,Y,Board3),

        Board == Board3,

        verifyCaptureDiagonal4(Board,X,Y,BoardR).

%%linha 6

verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 6,

        X > 6,

        verifyCaptureDiagonal1(Board,X,Y,Board1),

        Board \= Board1,

        Board1 = BoardR.

%%linha 6

verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 6,

        X > 6,

        verifyCaptureDiagonal1(Board,X,Y,Board1),

        Board == Board1,

        verifyCaptureDiagonal4(Board,X,Y,BoardR).


/***********************************************************/

/******************* parte de baixo do tabuleiro********************/
```

```
%linha11
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y > 8,
        Y < 12,
        X > 0,
        verifyCaptureDiagonal41(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%linha11
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y > 8,
        Y < 12,
        X > 0,
        verifyCaptureDiagonal41(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal21(Board,X,Y,BoardR).
%linha8
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 8,
        X == 1,
        verifyCaptureDiagonal21(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%linha8
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 8,
        X == 1,
        verifyCaptureDiagonal21(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal31(Board,X,Y,BoardR).
%linha8
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 8,
        X > 1,
        X < 4,
        verifyCaptureDiagonal41(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%linha8
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 8,
        X > 1,
        X < 4,
        verifyCaptureDiagonal41(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal21(Board,X,Y,Board2),
        Board \= Board2,
        Board1 = BoardR.
%linha8
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 8,
        X > 1,
        X < 4,
        verifyCaptureDiagonal41(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal21(Board,X,Y,Board2),
        Board == Board2,
        verifyCaptureDiagonal31(Board,X,Y,BoardR).
%linha8
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 8,
        X > 3,
        X < 6,
        verifyCaptureDiagonal11(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%linha8
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 8,
        X > 3,
        X < 6,
        verifyCaptureDiagonal11(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal21(Board,X,Y,Board2),
        Board \= Board2,
        Board2 = BoardR.
%linha8
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 8,
        X > 3,
        X < 6,
        verifyCaptureDiagonal11(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal21(Board,X,Y,Board2),
        Board == Board2,
        verifyCaptureDiagonal31(Board,X,Y,Board3),
        Board \= Board3,
        Board3 = BoardR.
%linha8
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 8,
        X > 3,
        X < 6,
        verifyCaptureDiagonal11(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal21(Board,X,Y,Board2),
        Board == Board2,
        verifyCaptureDiagonal31(Board,X,Y,Board3),
        Board == Board3,
	verifyCaptureDiagonal41(Board,X,Y,BoardR).
%linha8
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 8,
	X > 5,
	X < 8,
	verifyCaptureDiagonal11(Board,X,Y,Board1),
	Board \= Board1,
	Board1 = BoardR.
%linha8
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 8,
	X > 5,
	X < 8,
	verifyCaptureDiagonal11(Board,X,Y,Board1),
	Board == Board1,
	verifyCaptureDiagonal21(Board,X,Y,Board2),
	Board \= Board2,
	Board2 = BoardR.
%linha8
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 8,
	X > 5,
	X < 8,
	verifyCaptureDiagonal11(Board,X,Y,Board1),
    Board == Board1,

    verifyCaptureDiagonal21(Board,X,Y,Board2),

    Board == Board2,

    verifyCaptureDiagonal41(Board,X,Y,BoardR).

%linha8

verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 8,

    X == 8,

    verifyCaptureDiagonal11(Board,X,Y,Board1),

    Board \= Board1,

    Board1 = BoardR.

%linha8

verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 8,

    X == 8,

    verifyCaptureDiagonal11(Board,X,Y,Board1),

    Board == Board1,

    verifyCaptureDiagonal41(Board,X,Y,BoardR).

%linha7

verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 7,

    X < 3,

    verifyCaptureDiagonal21(Board,X,Y,Board1),

    Board \= Board1,

    Board1 = BoardR.
```

```
%linha7
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 7,
    X < 3,
    verifyCaptureDiagonal21(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal31(Board,X,Y,BoardR).
%linha7
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 7,
    X == 3,
    verifyCaptureDiagonal41(Board,X,Y,Board1),
    Board \= Board1,
    Board1 = BoardR.
%linha7
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 7,
    X == 3,
    verifyCaptureDiagonal41(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal21(Board,X,Y,Board2),
    Board \= Board2,
    Board2 = BoardR.
%linha7
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 7,
    X == 3,
    verifyCaptureDiagonal41(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal21(Board,X,Y,Board2),
    Board == Board2,
    verifyCaptureDiagonal31(Board,X,Y,BoardR).
%linha7
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 7,
    X > 3,
    X < 7,
    verifyCaptureDiagonal11(Board,X,Y,Board1),
    Board \= Board1,
    Board1 = BoardR.
%linha7
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 7,
    X > 3,
    X < 7,
    verifyCaptureDiagonal11(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal21(Board,X,Y,Board2),
    Board \= Board2,
    Board2 = BoardR.
```

```
%linha7
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 7,
    X > 3,
    X < 7,
    verifyCaptureDiagonal11(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal21(Board,X,Y,Board2),
    Board == Board2,
    verifyCaptureDiagonal31(Board,X,Y,Board3),
    Board \= Board3,
    Board3 = BoardR.
%linha7
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 7,
    X > 3,
    X < 7,
    verifyCaptureDiagonal11(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal21(Board,X,Y,Board2),
    Board == Board2,
    verifyCaptureDiagonal31(Board,X,Y,Board3),
    Board == Board3,
    verifyCaptureDiagonal41(Board,X,Y,BoardR).
```

```
%linha7
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 7,
    X == 7,
    verifyCaptureDiagonal11(Board,X,Y,Board1),
    Board \= Board1,
    Board1 = BoardR.
%linha7
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 7,
    X == 7,
    verifyCaptureDiagonal11(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal21(Board,X,Y,Board2),
    Board \= Board2,
    Board2 = BoardR.
%linha7
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 7,
    X == 7,
    verifyCaptureDiagonal11(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal21(Board,X,Y,Board2),
    Board == Board2,
    verifyCaptureDiagonal41(Board,X,Y,BoardR).
```

```
%linha7
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 7,
        X > 7,
        verifyCaptureDiagonal11(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%linha7
verifyCaptureDiagonals(Board,X,Y,BoardR):- Y == 7,
        X > 7,
        verifyCaptureDiagonal11(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal41(Board,X,Y,BoardR).


/*************************************************************************
***************/


/******************************          verifical          captura          na          horizontal
**********************/
%linha1
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 1,
        X == 3,
	append([],Board,Board2).
%linha1
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 1,
	X < 3,
	verifyCaptureHorizontal1(Board,X,Y,Board2).
%linha1
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 1,
	X > 3,
	verifyCaptureHorizontal2(Board,X,Y,Board2).
%linha2
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 2,
	X < 4,
	verifyCaptureHorizontal1(Board,X,Y,Board2).
%linha2
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 2,
	X > 3,
	verifyCaptureHorizontal2(Board,X,Y,Board2).
%linha3
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 3,
	X < 5,
	verifyCaptureHorizontal1(Board,X,Y,Board2).
%linha3
```

```
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 3,
    X > 3,
    verifyCaptureHorizontal2(Board,X,Y,Board2).
%linha4
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 4,
    X < 6,
    verifyCaptureHorizontal1(Board,X,Y,Board2).
%linha4
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 4,
    X > 3,
    verifyCaptureHorizontal2(Board,X,Y,Board2).
%linha5
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 5,
    X < 7,
    verifyCaptureHorizontal1(Board,X,Y,Board2).
%linha5
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 5,
    X > 3,
    verifyCaptureHorizontal2(Board,X,Y,Board2).
%linha6
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 6,
    X < 8,
	verifyCaptureHorizontal1(Board,X,Y,Board2).
%linha6
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 6,
	X > 3,
	verifyCaptureHorizontal2(Board,X,Y,Board2).
%linha7
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 7,
	X < 7,
	verifyCaptureHorizontal1(Board,X,Y,Board2).
%linha7
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 7,
	X > 3,
	verifyCaptureHorizontal2(Board,X,Y,Board2).
%linha8
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 8,
	X < 6,
	verifyCaptureHorizontal1(Board,X,Y,Board2).
%linha8
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 8,
	X > 3,
	verifyCaptureHorizontal2(Board,X,Y,Board2).
%linha9
```

```
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 9,
        X < 5,
        verifyCaptureHorizontal1(Board,X,Y,Board2).
%linha9
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 9,
        X > 3,
        verifyCaptureHorizontal2(Board,X,Y,Board2).
%linha10
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 10,
        X == 4,
        append([],Board,Board2).
%linha10
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 10,
        X < 4,
        verifyCaptureHorizontal1(Board,X,Y,Board2).
%linha10
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 10,
        X > 3,
        verifyCaptureHorizontal2(Board,X,Y,Board2).
%linha11
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 11,
        X < 3,
    verifyCaptureHorizontal1(Board,X,Y,Board2).
%linha11
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 11,
    X > 3,
    verifyCaptureHorizontal2(Board,X,Y,Board2).
%linha11
verifyCaptureHorizontal(Board,X,Y,Board2):- Y == 11,
    X == 3,
    append([],Board,Board2).
```

Anexo D - diagonals_win.pl

```
%/*********************************diagonal 1 -> esq peça em cima para baixo
*******************************************/

verifyWinDiagonal1(Board, X,Y):- returnPieceAt(Board, X, Y, Pieceat),
        Y1 is Y + 1,
        returnPieceAt(Board, X, Y1, PieceAtD1),
        PieceAtD1 == Pieceat,
        PieceAtD1 \= ' ',
        verifyWinDiagonal1aux(Board, X, Y1, Pieceat).

/**********/
verifyWinDiagonal1aux(Board, X, Y, PieceAnterior):-  returnPieceAt(Board, X, Y, Pieceat),
        PieceAnterior == Pieceat,
        Y1 is Y + 1,
        verifyWinDiagonal1aux2(Board, X, Y1, Pieceat).

/**********/
```

```
verifyWinDiagonal1aux2(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),
        PieceAnterior == Pieceat,
        Y1 is Y + 1,
        verifyWinDiagonal1aux3(Board, X, Y1, Pieceat).

/**********/
verifyWinDiagonal1aux3(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),
        PieceAnterior == Pieceat,
        Y1 is Y + 1,
        verifyWinDiagonal1aux4(Board, X, Y1, Pieceat).

/***********/
verifyWinDiagonal1aux4(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),
        PieceAnterior == Pieceat,
        Pieceat == 'W',
        write('\nPlayer 1 win the game!\n'),nl,nl.

verifyWinDiagonal1aux4(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),
        PieceAnterior == Pieceat,
        Pieceat == 'B',
        write('\nPlayer 2 win the game!\n'),nl,nl.
```

```
/*************************************************************************
**************************/

%/*********************************diagonal 2 -> direita baixo para cima
*******************************************/

verifyWinDiagonal2(Board, X,Y):- returnPieceAt(Board, X, Y, Pieceat),
	Y1 is Y - 1,
	returnPieceAt(Board, X, Y1, PieceAtD1),
	PieceAtD1 == Pieceat,
	PieceAtD1 \= ' ',
	verifyWinDiagonal2aux(Board, X, Y1, Pieceat).

/**********/
verifyWinDiagonal2aux(Board, X, Y, PieceAnterior):-  returnPieceAt(Board, X, Y, Pieceat),
	PieceAnterior == Pieceat,
	Y1 is Y - 1,
	verifyWinDiagonal2aux2(Board, X, Y1, Pieceat).

/**********/
```

```
verifyWinDiagonal2aux2(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),
    PieceAnterior == Pieceat,
    Y1 is Y - 1,
    verifyWinDiagonal2aux3(Board, X, Y1, Pieceat).
/**********/
verifyWinDiagonal2aux3(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),
    PieceAnterior == Pieceat,
    Y1 is Y - 1,
    verifyWinDiagonal2aux4(Board, X, Y1, Pieceat).


/***********/
verifyWinDiagonal2aux4(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),
    PieceAnterior == Pieceat,
    Pieceat == 'W',
    write('\nPlayer 1 win the game!\n'),nl,nl.


verifyWinDiagonal2aux4(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),
    PieceAnterior == Pieceat,
    Pieceat == 'B',
    write('\nPlayer 2 win the game!\n'),nl,nl.
```

```
/****************************************************************************
*************************/



/*********************************diagonal 3 -> direita cima para baixo
*******************************************/

verifyWinDiagonal3(Board, X,Y):- returnPieceAt(Board, X, Y, Pieceat),
    X1 is X + 1,
    Y1 is Y + 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1),
    PieceAtD1 == Pieceat,
    PieceAtD1 \= ' ',
    verifyWinDiagonal3aux(Board, X1, Y1, Pieceat).

/**********/
verifyWinDiagonal3aux(Board, X, Y, PieceAnterior):-  returnPieceAt(Board, X, Y, Pieceat),
    PieceAnterior == Pieceat,
    X1 is X + 1,
    Y1 is Y + 1,
    verifyWinDiagonal3aux2(Board, X1, Y1, Pieceat).
```

```
/**********/

verifyWinDiagonal3aux2(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),

        PieceAnterior == Pieceat,

        X1 is X + 1,

        Y1 is Y + 1,

        verifyWinDiagonal3aux3(Board, X1, Y1, Pieceat).


/**********/

verifyWinDiagonal3aux3(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),

        PieceAnterior == Pieceat,

        X1 is X + 1,

        Y1 is Y + 1,

        verifyWinDiagonal3aux4(Board, X1, Y1, Pieceat).


/***********/

verifyWinDiagonal3aux4(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),

        PieceAnterior == Pieceat,

        Pieceat == 'W',

        write('\nPlayer 1 win the game!\n'),nl,nl.


verifyWinDiagonal3aux4(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),

        PieceAnterior == Pieceat,
        Pieceat == 'B',

        write('\nPlayer 2 win the game!\n'),nl,nl.


/***************************************************************************
**************************/



/*********************************diagonal 4 -> esquerda baixo para cima
******************************************/

verifyWinDiagonal4(Board, X,Y):- returnPieceAt(Board, X, Y, Pieceat),

        X1 is X - 1,

        Y1 is Y - 1,

        returnPieceAt(Board, X1, Y1, PieceAtD1),

        PieceAtD1 == Pieceat,

        PieceAtD1 \= ' ',

        verifyWinDiagonal4aux(Board, X1, Y1, Pieceat).


/**********/

verifyWinDiagonal4aux(Board, X, Y, PieceAnterior):-  returnPieceAt(Board, X, Y, Pieceat),

        PieceAnterior == Pieceat,

        X1 is X - 1,

        Y1 is Y - 1,
        verifyWinDiagonal4aux2(Board, X1, Y1, Pieceat).


/**********/

verifyWinDiagonal4aux2(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),

        PieceAnterior == Pieceat,

        X1 is X - 1,

        Y1 is Y - 1,

        verifyWinDiagonal4aux3(Board, X1, Y1, Pieceat).


/**********/

verifyWinDiagonal4aux3(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),

        PieceAnterior == Pieceat,

        X1 is X - 1,

        Y1 is Y - 1,

        verifyWinDiagonal4aux4(Board, X1, Y1, Pieceat).


/***********/

verifyWinDiagonal4aux4(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),

        PieceAnterior == Pieceat,

        Pieceat == 'W',

        write('\nPlayer 1 win the game!\n'),nl,nl.
```

```
verifyWinDiagonal4aux4(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),
        PieceAnterior == Pieceat,
        Pieceat == 'B',
        write('\nPlayer 2 win the game!\n'),nl,nl.


/*******************************************************************************
***************************/
/*******************************************************************************
**************************/
/**************************
**************************************/
/************************** diagonais parte de baixo do tabuleiro
**********************************/
/**************************
**************************************/
/*******************************************************************************
****************************/
/*******************************************************************************
****************************/



%/*************************************diagonal 1 -> esq peça em cima para baixo
************************************************/
```

```
verifyWinDiagonal11(Board, X,Y):- returnPieceAt(Board, X, Y, Pieceat),
	X1 is X - 1,
	Y1 is Y + 1,
	returnPieceAt(Board, X1, Y1, PieceAtD1),
	PieceAtD1 == Pieceat,
	PieceAtD1 \= ' ',
	verifyWinDiagonal11aux(Board, X1, Y1, Pieceat).

/**********/
verifyWinDiagonal11aux(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y,
Pieceat),
	PieceAnterior == Pieceat,
	X1 is X - 1,
	Y1 is Y + 1,
	verifyWinDiagonal11aux2(Board, X1, Y1, Pieceat).

/**********/
verifyWinDiagonal11aux2(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y,
Pieceat),
	PieceAnterior == Pieceat,
	X1 is X - 1,
        Y1 is Y + 1,
        verifyWinDiagonal11aux3(Board, X1, Y1, Pieceat).


/*********/
verifyWinDiagonal11aux3(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),
        PieceAnterior == Pieceat,
        X1 is X - 1,
        Y1 is Y + 1,
        verifyWinDiagonal11aux4(Board, X1, Y1, Pieceat).


/**********/
verifyWinDiagonal11aux4(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),
        PieceAnterior == Pieceat,
        Pieceat == 'W',
        write('\nPlayer 1 win the game!\n'),nl,nl.


verifyWinDiagonal11aux4(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),
        PieceAnterior == Pieceat,
        Pieceat == 'B',
        write('\nPlayer 2 win the game!\n'),nl,nl.

/***************************************************************************
**************************/



%/*********************************diagonal 2 -> direita baixo para cima
*******************************************/

verifyWinDiagonal22(Board, X,Y):- returnPieceAt(Board, X, Y, Pieceat),
        X1 is X + 1,
        Y1 is Y - 1,
        returnPieceAt(Board, X1, Y1, PieceAtD1),
        PieceAtD1 == Pieceat,
        PieceAtD1 \= ' ',
        verifyWinDiagonal22aux(Board, X1, Y1, Pieceat).
/**********/
verifyWinDiagonal22aux(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y,
Pieceat),
        PieceAnterior == Pieceat,
        X1 is X + 1,
        Y1 is Y - 1,
        verifyWinDiagonal22aux2(Board, X1, Y1, Pieceat).


/**********/

verifyWinDiagonal22aux2(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y,
Pieceat),

        PieceAnterior == Pieceat,

        X1 is X + 1,

        Y1 is Y - 1,

        verifyWinDiagonal22aux3(Board, X1, Y1, Pieceat).



/**********/

verifyWinDiagonal22aux3(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y,
Pieceat),

        PieceAnterior == Pieceat,

        X1 is X + 1,

        Y1 is Y - 1,

        verifyWinDiagonal22aux4(Board, X1, Y1, Pieceat).


/***********/

verifyWinDiagonal22aux4(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y,
Pieceat),
        PieceAnterior == Pieceat,

        Pieceat == 'W',

        write('\nPlayer 1 win the game!\n'),nl,nl.


verifyWinDiagonal22aux4(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y,
Pieceat),

        PieceAnterior == Pieceat,

        Pieceat == 'B',

        write('\nPlayer 2 win the game!\n'),nl,nl.


/*************************************************************************************
***************************/


/*************************************************************************************
***************************/




/***********************************diagonal 3 -> direita cima para baixo
**************************************************/


verifyWinDiagonal33(Board, X,Y):- returnPieceAt(Board, X, Y, Pieceat),
        Y1 is Y - 1,
        returnPieceAt(Board, X, Y1, PieceAtD1),
        PieceAtD1 == Pieceat,
        PieceAtD1 \= ' ',
        verifyWinDiagonal33aux(Board, X, Y1, Pieceat).

/**********/
verifyWinDiagonal33aux(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),
        PieceAnterior == Pieceat,
        Y1 is Y - 1,
        verifyWinDiagonal33aux2(Board, X, Y1, Pieceat).

/**********/
verifyWinDiagonal33aux2(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),
        PieceAnterior == Pieceat,
        Y1 is Y - 1,
        verifyWinDiagonal33aux3(Board, X, Y1, Pieceat).

/**********/
```

```
verifyWinDiagonal33aux3(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),
        PieceAnterior == Pieceat,
        Y1 is Y - 1,
        verifyWinDiagonal33aux4(Board, X, Y1, Pieceat).

/***********/
verifyWinDiagonal33aux4(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),
        PieceAnterior == Pieceat,
        Pieceat == 'W',
        write('\nPlayer 1 win the game!\n'),nl,nl.

verifyWinDiagonal33aux4(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y, Pieceat),
        PieceAnterior == Pieceat,
        Pieceat == 'B',
        write('\nPlayer 2 win the game!\n'),nl,nl.

/***************************************************************************
**************************/
```

```
/*********************************diagonal 4 -> esquerda baixo para cima
*******************************************/

verifyWinDiagonal44(Board, X,Y):- returnPieceAt(Board, X, Y, Pieceat),
    Y1 is Y - 1,
    returnPieceAt(Board, X, Y1, PieceAtD1),
    PieceAtD1 == Pieceat,
    PieceAtD1 \= ' ',
    verifyWinDiagonal44aux(Board, X, Y1, Pieceat).

/**********/
verifyWinDiagonal44aux(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y,
Pieceat),
    PieceAnterior == Pieceat,
    Y1 is Y - 1,
    verifyWinDiagonal44aux2(Board, X, Y1, Pieceat).

/**********/
verifyWinDiagonal44aux2(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y,
Pieceat),
    PieceAnterior == Pieceat,
        Y1 is Y - 1,
        verifyWinDiagonal44aux3(Board, X, Y1, Pieceat).


/**********/
verifyWinDiagonal44aux3(Board,   X,   Y,   PieceAnterior):-   returnPieceAt(Board,   X,   Y,
Pieceat),
        PieceAnterior == Pieceat,
        Y1 is Y - 1,
        verifyWinDiagonal44aux4(Board, X, Y1, Pieceat).


/***********/
verifyWinDiagonal44aux4(Board,   X,   Y,   PieceAnterior):-   returnPieceAt(Board,   X,   Y,
Pieceat),
        PieceAnterior == Pieceat,
        Pieceat == 'W',
        write('\nPlayer 1 win the game!\n'),nl,nl.


verifyWinDiagonal44aux4(Board,   X,   Y,   PieceAnterior):-   returnPieceAt(Board,   X,   Y,
Pieceat),
        PieceAnterior == Pieceat,
        Pieceat == 'B',
        write('\nPlayer 2 win the game!\n'),nl,nl.
```

```
/*************************************************************************
***************/
/*************************************************************************
***************/
/********************************
****************************/
/********************************
****************************/
/******************************** verifical win na horizontal
****************************/
/********************************
****************************/
/********************************
****************************/
/*************************************************************************
***************/
/*************************************************************************
***************/

%horizontal esq to dir
verifyWinHorizontal1(Board, X,Y):- returnPieceAt(Board, X, Y, Pieceat),
    X1 is X + 1,
	returnPieceAt(Board, X1, Y, PieceAtD1),

	PieceAtD1 == Pieceat,

	PieceAtD1 \= ' ',

	verifyWinHorizontal1aux(Board, X1, Y, Pieceat).


/**********/

verifyWinHorizontal1aux(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y,
Pieceat),

	PieceAnterior == Pieceat,

	X1 is X + 1,

	verifyWinHorizontal1aux2(Board, X1, Y, Pieceat).


/**********/

verifyWinHorizontal1aux2(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y,
Pieceat),

	PieceAnterior == Pieceat,

	X1 is X + 1,

	verifyWinHorizontal1aux3(Board, X1, Y, Pieceat).


/**********/

verifyWinHorizontal1aux3(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y,
Pieceat),
        PieceAnterior == Pieceat,

        X1 is X + 1,

        verifyWinHorizontal1aux4(Board, X1, Y, Pieceat).


/***********/

verifyWinHorizontal1aux4(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y,
Pieceat),

        PieceAnterior == Pieceat,

        Pieceat == 'W',

        write('\nPlayer 1 win the game, left to right\n'),nl,nl.


verifyWinHorizontal1aux4(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y,
Pieceat),

        PieceAnterior == Pieceat,

        Pieceat == 'B',

        write('\nPlayer 2 win the game, left to right\n'),nl,nl.



/**********************************************************************
**************************/
```

```
/***************************** ************** ***********************/


%horizontal dir to esq

verifyWinHorizontal2(Board, X,Y):- returnPieceAt(Board, X, Y, Pieceat),

        X1 is X - 1,

        returnPieceAt(Board, X1, Y, PieceAtD1),

        PieceAtD1 == Pieceat,

        PieceAtD1 \= ' ',

        verifyWinHorizontal2aux(Board, X1, Y, Pieceat).


/**********/

verifyWinHorizontal2aux(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y,
Pieceat),

        PieceAnterior == Pieceat,

        X1 is X - 1,

        verifyWinHorizontal2aux2(Board, X1, Y, Pieceat).


/**********/

verifyWinHorizontal2aux2(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y,
Pieceat),

        PieceAnterior == Pieceat,

        X1 is X - 1,
        verifyWinHorizontal2aux3(Board, X1, Y, Pieceat).


/**********/

verifyWinHorizontal2aux3(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y,
Pieceat),

        PieceAnterior == Pieceat,

        X1 is X - 1,

        verifyWinHorizontal2aux4(Board, X1, Y, Pieceat).


/***********/

verifyWinHorizontal2aux4(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y,
Pieceat),

        PieceAnterior == Pieceat,

        Pieceat == 'W',

        write('\nPlayer 1 win the game, right to left\n'),nl,nl.


verifyWinHorizontal2aux4(Board, X, Y, PieceAnterior):- returnPieceAt(Board, X, Y,
Pieceat),

        PieceAnterior == Pieceat,

        Pieceat == 'B',

        write('\nPlayer 2 win the game, right to left\n'),nl,nl.
```

```
/**************************************************************************
***************************/
```

12 de Novembro de 2017

Anexo E - diagonals.pl

```
/******************************************************************      Piece
************************/

/**********************************diagonal   1   ->   esq   cima   para   baixo   /
********************************************/

verifyCaptureDiagonal1(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),

        X1 is X,

        Y1 is Y + 1,

        returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */

        PieceAtD1 \= Pieceat,

        PieceAtD1 \= ' ',

        verifyCaptureDiagonal1aux(Board, X1, Y1, Pieceat, Board2).


verifyCaptureDiagonal1(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),

        X1 is X,

        Y1 is Y + 1,

        returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */

        PieceAtD1 == Pieceat,

        Board = Board2.


verifyCaptureDiagonal1(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, _),
	X1 is X,
	Y1 is Y + 1,
	returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
	PieceAtD1 == ' ',
	Board = Board2.

verifyCaptureDiagonal1aux(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board, X, Y, 
Pieceat),
	PieceAnterior \= Pieceat,
	Pieceat \= ' ',
	X1 is X,
	Y1 is Y + 1,
	verifyCaptureDiagonal1aux2(Board, X1, Y1, Pieceat, Board2). /* diagonal esq baixo */

verifyCaptureDiagonal1aux(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board, X, Y, 
Pieceat),
	PieceAnterior == Pieceat,
	Board = Board2.



verifyCaptureDiagonal1aux(Board,X,Y,_,Board2):-  returnPieceAt(Board, X, Y, Pieceat),
	Pieceat == ' ',
    Board = Board2.



verifyCaptureDiagonal1aux2(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    PieceAnterior == Pieceat,
    X1 is X,
    Y1 is Y + 1,
    verifyCaptureDiagonal1aux3(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal1aux2(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    PieceAnterior \= Pieceat,
    Board = Board2.

verifyCaptureDiagonal1aux2(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    Pieceat == ' ',
    Board = Board2.



verifyCaptureDiagonal1aux3(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    PieceAnterior \= Pieceat,

    Pieceat \= ' ',

    write('\nPosition of capture, diagonal left top to bottom\n'),

    X1 is X,

    Y1 is Y - 1,

    X2 is X1,

    Y2 is Y1 - 1,

    changePieceAtCapture(Board,X1,Y1,X2,Y2,Board2, Pieceat).


verifyCaptureDiagonal1aux3(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X, Y,
Pieceat),

    PieceAnterior == Pieceat,

    Board = Board2.


verifyCaptureDiagonal1aux3(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),

    Pieceat == ' ',

    Board = Board2.


/*************************************************************************
***************************/
```

```
/*********************************diagonal 2 -> direita baixo para cima /
***************************/
/**************************************************************
Piece ******************/
verifyCaptureDiagonal2(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        X1 is X,
        Y1 is Y - 1,
        returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
        PieceAtD1 \= Pieceat,
        PieceAtD1 \= ' ',
        verifyCaptureDiagonal2aux(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal2(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        X1 is X,
        Y1 is Y - 1,
        returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
        PieceAtD1 == Pieceat,
        Board = Board2.

verifyCaptureDiagonal2(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, _),
        X1 is X,
        Y1 is Y - 1,

        returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */

        PieceAtD1 == ' ',

        Board = Board2.


verifyCaptureDiagonal2aux(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board, X, Y,
Pieceat),

        PieceAnterior \= Pieceat,

        Pieceat \= ' ',

        X1 is X,

        Y1 is Y - 1,

        verifyCaptureDiagonal2aux2(Board, X1, Y1, Pieceat, Board2). /* diagonal esq baixo */


verifyCaptureDiagonal2aux(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board, X, Y,
Pieceat),

        PieceAnterior == Pieceat,

        Board = Board2.



verifyCaptureDiagonal2aux(Board,X,Y,_,Board2):-  returnPieceAt(Board, X, Y, Pieceat),

        Pieceat == ' ',

        Board = Board2.
```

```
verifyCaptureDiagonal2aux2(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X, Y,
Pieceat),
        PieceAnterior == Pieceat,
        X1 is X,
        Y1 is Y - 1,
        verifyCaptureDiagonal2aux3(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal2aux2(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X, Y,
Pieceat),
        PieceAnterior \= Pieceat,
        Board = Board2.

verifyCaptureDiagonal2aux2(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board = Board2.


verifyCaptureDiagonal2aux3(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X, Y,
Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',

        write('\nPosition of capture, right diagonal low up\n'),

        X1 is X,

        Y1 is Y + 1,

        X2 is X1,

        Y2 is Y1 + 1,

        changePieceAtCapture(Board,X1,Y1,X2,Y2,Board2, Pieceat).


verifyCaptureDiagonal2aux3(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X, Y,
Pieceat),

        PieceAnterior == Pieceat,

        Board = Board2.


verifyCaptureDiagonal2aux3(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),

        Pieceat == ' ',

        Board = Board2.


/*************************************************************************
**************************/
```

```
/*******************************************************************
Piece **************************/
/*********************************diagonal 3 -> direita peça cima para baixo \
*******************************************/


verifyCaptureDiagonal3(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
	X1 is X + 1,
	Y1 is Y + 1,
	returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
	PieceAtD1 \= Pieceat,
	PieceAtD1 \= ' ',
	verifyCaptureDiagonal3aux(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal3(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
	X1 is X + 1,
	Y1 is Y + 1,
	returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
	PieceAtD1 == Pieceat,
	Board = Board2.

verifyCaptureDiagonal3(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, _),
	X1 is X + 1,
        Y1 is Y + 1,

        returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */

        PieceAtD1 == ' ',

        Board = Board2.


verifyCaptureDiagonal3aux(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board, X, Y,
Pieceat),

        PieceAnterior \= Pieceat,

        Pieceat \= ' ',

        X1 is X + 1,

        Y1 is Y + 1,

        verifyCaptureDiagonal3aux2(Board, X1, Y1, Pieceat, Board2). /* diagonal esq baixo */


verifyCaptureDiagonal3aux(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board, X, Y,
Pieceat),

        PieceAnterior == Pieceat,

        Board = Board2.



verifyCaptureDiagonal3aux(Board,X,Y,_,Board2):-  returnPieceAt(Board, X, Y, Pieceat),

        Pieceat == ' ',

        Board = Board2.
```

```
verifyCaptureDiagonal3aux2(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X, Y, 
Pieceat),
        PieceAnterior == Pieceat,
        X1 is X + 1,
        Y1 is Y + 1,
        verifyCaptureDiagonal3aux3(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal3aux2(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X, Y, 
Pieceat),
        PieceAnterior \= Pieceat,
        Board = Board2.

verifyCaptureDiagonal3aux2(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board = Board2.


verifyCaptureDiagonal3aux3(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X, Y, 
Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',
        write('\nPosition of capture, diagonal right up down.\n'),
        X1 is X - 1,
        Y1 is Y - 1,
        X2 is X1 - 1,
        Y2 is Y1 - 1,
        changePieceAtCapture(Board,X1,Y1,X2,Y2,Board2, Pieceat).

verifyCaptureDiagonal3aux3(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X, Y,
Pieceat),
        PieceAnterior == Pieceat,
        Board = Board2.

verifyCaptureDiagonal3aux3(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board = Board2.

/************************************************************************
**************************/
```

```
/********************************diagonal 4 -> esquerda baixo para cima \
*******************************************/

/************************************************************************
Piece *****************/

verifyCaptureDiagonal4(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    X1 is X - 1,
    Y1 is Y - 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 \= Pieceat,
    PieceAtD1 \= ' ',
    verifyCaptureDiagonal4aux(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal4(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    X1 is X - 1,
    Y1 is Y - 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 == Pieceat,
    Board = Board2.

verifyCaptureDiagonal4(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, _),
    X1 is X - 1,
    Y1 is Y - 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 == ' ',
    Board = Board2.

verifyCaptureDiagonal4aux(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board, X, Y,
Pieceat),
    PieceAnterior \= Pieceat,
    Pieceat \= ' ',
    X1 is X - 1,
    Y1 is Y - 1,
    verifyCaptureDiagonal4aux2(Board, X1, Y1, Pieceat, Board2). /* diagonal esq baixo */

verifyCaptureDiagonal4aux(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board, X, Y,
Pieceat),
    PieceAnterior == Pieceat,
    Board = Board2.


verifyCaptureDiagonal4aux(Board,X,Y,_,Board2):-  returnPieceAt(Board, X, Y, Pieceat),
    Pieceat == ' ',
    Board = Board2.
```

```
verifyCaptureDiagonal4aux2(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X, Y,
Pieceat),
        PieceAnterior == Pieceat,
        X1 is X - 1,
        Y1 is Y - 1,
        verifyCaptureDiagonal4aux3(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal4aux2(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X, Y,
Pieceat),
        PieceAnterior \= Pieceat,
        Board = Board2.

verifyCaptureDiagonal4aux2(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board = Board2.


verifyCaptureDiagonal4aux3(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X, Y,
Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',
        write('\nPosition of capture, diagonal left low up\n'),
        X1 is X + 1,
        Y1 is Y + 1,
        X2 is X1 + 1,
        Y2 is Y1 + 1,
        changePieceAtCapture(Board,X1,Y1,X2,Y2,Board2, Pieceat).

verifyCaptureDiagonal4aux3(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X, Y, 
Pieceat),
        PieceAnterior == Pieceat,
        Board = Board2.

verifyCaptureDiagonal4aux3(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board = Board2.

/**************************************************************************
**************************/
```

```
/**********************************************************************************
**************************/

/**********************************************************************************
****************************/

/*****************************        diagonais        parte        de        baixo        do
tabuleiro************************************/

/**********************************************************************************
**************************/

/**********************************************************************************
****************************/



/***************************************diagonal    1    ->    esq    cima    para    baixo
**************************************************/

verifyCaptureDiagonal11(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),

	X1 is X - 1,

	Y1 is Y + 1,

	returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */

	PieceAtD1 \= Pieceat,

	PieceAtD1 \= ' ',

	verifyCaptureDiagonal11aux(Board, X1, Y1, Pieceat, Board2).


verifyCaptureDiagonal11(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        X1 is X - 1,

        Y1 is Y + 1,

        returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */

        PieceAtD1 == Pieceat,

        Board = Board2.


verifyCaptureDiagonal11(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, _),

        X1 is X - 1,

        Y1 is Y + 1,

        returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */

        PieceAtD1 == ' ',

        Board = Board2.


verifyCaptureDiagonal11aux(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,  X,
Y, Pieceat),

        PieceAnterior \= Pieceat,

        Pieceat \= ' ',

        X1 is X - 1,

        Y1 is Y + 1,

        verifyCaptureDiagonal11aux2(Board, X1, Y1, Pieceat, Board2). /* diagonal esq baixo */
```

```
verifyCaptureDiagonal11aux(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,  X,
Y, Pieceat),

        PieceAnterior == Pieceat,

        Board = Board2.



verifyCaptureDiagonal11aux(Board,X,Y,_,Board2):-  returnPieceAt(Board, X, Y, Pieceat),

        Pieceat == ' ',

        Board = Board2.



verifyCaptureDiagonal11aux2(Board,X,Y,PieceAnterior,Board2):-   returnPieceAt(Board,  X,
Y, Pieceat),

        PieceAnterior == Pieceat,

        X1 is X - 1,

        Y1 is Y + 1,

        verifyCaptureDiagonal11aux3(Board, X1, Y1, Pieceat, Board2).


verifyCaptureDiagonal11aux2(Board,X,Y,PieceAnterior,Board2):-   returnPieceAt(Board,  X,
Y, Pieceat),

        PieceAnterior \= Pieceat,

        Board = Board2.
```

```
verifyCaptureDiagonal11aux2(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board = Board2.



verifyCaptureDiagonal11aux3(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board,  X,
Y, Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',
        write('\nPosition of capture, diagonal left top to bottom. Bottom of the board\n'),
        X1 is X + 1,
        Y1 is Y - 1,
        X2 is X1 + 1,
        Y2 is Y1 - 1,
        changePieceAtCapture(Board,X1,Y1,X2,Y2,Board2, Pieceat).

verifyCaptureDiagonal11aux3(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board,  X,
Y, Pieceat),
        PieceAnterior == Pieceat,
        Board = Board2.
```

```
verifyCaptureDiagonal11aux3(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
	Pieceat == ' ',
	Board = Board2.

/*****************************************************************************
**************************/




/*********************************diagonal 2 -> direita baixo para cima
********************************************/

verifyCaptureDiagonal21(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
	X1 is X + 1,
	Y1 is Y - 1,
	returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
	PieceAtD1 \= Pieceat,
	PieceAtD1 \= ' ',
	verifyCaptureDiagonal21aux(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal21(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
	X1 is X + 1,
    Y1 is Y - 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 == Pieceat,
    Board = Board2.

verifyCaptureDiagonal21(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, _),
    X1 is X + 1,
    Y1 is Y - 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 == ' ',
    Board = Board2.

verifyCaptureDiagonal21aux(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,  X,
Y, Pieceat),
    PieceAnterior \= Pieceat,
    Pieceat \= ' ',
    X1 is X + 1,
    Y1 is Y - 1,
    verifyCaptureDiagonal21aux2(Board, X1, Y1, Pieceat, Board2). /* diagonal esq baixo */

verifyCaptureDiagonal21aux(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,  X,
Y, Pieceat),
    PieceAnterior == Pieceat,

    Board = Board2.



verifyCaptureDiagonal21aux(Board,X,Y,_,Board2):-  returnPieceAt(Board, X, Y, Pieceat),

    Pieceat == ' ',

    Board = Board2.



verifyCaptureDiagonal21aux2(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board,  X,
Y, Pieceat),

    PieceAnterior == Pieceat,

    X1 is X + 1,

    Y1 is Y - 1,

    verifyCaptureDiagonal21aux3(Board, X1, Y1, Pieceat, Board2).


verifyCaptureDiagonal21aux2(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board,  X,
Y, Pieceat),

    PieceAnterior \= Pieceat,

    Board = Board2.


verifyCaptureDiagonal21aux2(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    Pieceat == ' ',
    Board = Board2.



verifyCaptureDiagonal21aux3(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board,  X,
Y, Pieceat),
    PieceAnterior \= Pieceat,
    Pieceat \= ' ',
    write('\nPosition of capture, right diagonal low up. Bottom of the board.\n'),
    X1 is X - 1,
    Y1 is Y + 1,
    X2 is X1 - 1,
    Y2 is Y1 + 1,
    changePieceAtCapture(Board,X1,Y1,X2,Y2,Board2, Pieceat).

verifyCaptureDiagonal21aux3(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board,  X,
Y, Pieceat),
    PieceAnterior == Pieceat,
    Board = Board2.

verifyCaptureDiagonal21aux3(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    Pieceat == ' ',
    Board = Board2.

/**************************************************************************
***************************/




/**********************************diagonal   3   ->   direita   cima   para   baixo
********************************************/

verifyCaptureDiagonal31(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    X1 is X,
    Y1 is Y + 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 \= Pieceat,
    PieceAtD1 \= ' ',
    verifyCaptureDiagonal31aux(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal31(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    X1 is X,
    Y1 is Y + 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 == Pieceat,
    Board = Board2.

verifyCaptureDiagonal31(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, _),
    X1 is X,
    Y1 is Y + 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 == ' ',
    Board = Board2.

verifyCaptureDiagonal31aux(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,  X,
Y, Pieceat),
    PieceAnterior \= Pieceat,
    Pieceat \= ' ',
    X1 is X,
    Y1 is Y + 1,
    verifyCaptureDiagonal31aux2(Board, X1, Y1, Pieceat, Board2). /* diagonal esq baixo */

verifyCaptureDiagonal31aux(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,  X,
Y, Pieceat),
    PieceAnterior == Pieceat,
    Board = Board2.



verifyCaptureDiagonal31aux(Board,X,Y,_,Board2):-  returnPieceAt(Board, X, Y, Pieceat),

    Pieceat == ' ',

    Board = Board2.



verifyCaptureDiagonal31aux2(Board,X,Y,PieceAnterior,Board2):-   returnPieceAt(Board,  X,
Y, Pieceat),

    PieceAnterior == Pieceat,

    X1 is X,

    Y1 is Y + 1,

    verifyCaptureDiagonal31aux3(Board, X1, Y1, Pieceat, Board2).


verifyCaptureDiagonal31aux2(Board,X,Y,PieceAnterior,Board2):-   returnPieceAt(Board,  X,
Y, Pieceat),

    PieceAnterior \= Pieceat,

    Board = Board2.


verifyCaptureDiagonal31aux2(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),

    Pieceat == ' ',
    Board = Board2.


verifyCaptureDiagonal31aux3(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board,  X,
Y, Pieceat),
    PieceAnterior \= Pieceat,
    Pieceat \= ' ',
    write('\nPosition of capture, diagonal right up down. Bottom of the board.\n'),
    X1 is X,
    Y1 is Y - 1,
    X2 is X1,
    Y2 is Y1 - 1,
    changePieceAtCapture(Board,X1,Y1,X2,Y2,Board2, Pieceat).

verifyCaptureDiagonal31aux3(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board,  X,
Y, Pieceat),
    PieceAnterior == Pieceat,
    Board = Board2.

verifyCaptureDiagonal31aux3(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    Pieceat == ' ',
    Board = Board2.
```

```
/*************************************************************************
*************************/



/*********************************diagonal 4 -> esquerda baixo para cima
*******************************************/

verifyCaptureDiagonal41(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    X1 is X,
    Y1 is Y - 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 \= Pieceat,
    PieceAtD1 \= ' ',
    verifyCaptureDiagonal41aux(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal41(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    X1 is X,
    Y1 is Y - 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
```

```
        PieceAtD1 == Pieceat,
        Board = Board2.

verifyCaptureDiagonal41(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, _),
        X1 is X,
        Y1 is Y - 1,
        returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
        PieceAtD1 == ' ',
        Board = Board2.

verifyCaptureDiagonal41aux(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,  X,
Y, Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',
        X1 is X,
        Y1 is Y - 1,
        verifyCaptureDiagonal41aux2(Board, X1, Y1, Pieceat, Board2). /* diagonal esq baixo */

verifyCaptureDiagonal41aux(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,  X,
Y, Pieceat),
        PieceAnterior == Pieceat,
        Board = Board2.
```

```
verifyCaptureDiagonal41aux(Board,X,Y,_,Board2):-  returnPieceAt(Board, X, Y, Pieceat),
	Pieceat == ' ',
	Board = Board2.



verifyCaptureDiagonal41aux2(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board,  X, Y, Pieceat),
	PieceAnterior == Pieceat,
	X1 is X,
	Y1 is Y - 1,
	verifyCaptureDiagonal41aux3(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal41aux2(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board,  X, Y, Pieceat),
	PieceAnterior \= Pieceat,
	Board = Board2.

verifyCaptureDiagonal41aux2(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
	Pieceat == ' ',
	Board = Board2.
```

```
verifyCaptureDiagonal41aux3(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board,  X,
Y, Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',
        write('\nPosition of capture, diagonal left low up. Bottom of the board\n'),
        X1 is X,
        Y1 is Y + 1,
        X2 is X1,
        Y2 is Y1 + 1,
        changePieceAtCapture(Board,X1,Y1,X2,Y2,Board2, Pieceat).

verifyCaptureDiagonal41aux3(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board,  X,
Y, Pieceat),
        PieceAnterior == Pieceat,
        Board = Board2.

verifyCaptureDiagonal41aux3(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board = Board2.
```

```
/****************************************************************************
***************************/



/********************************** horizontal capture
*************************/

%horizontal esq to dir
verifyCaptureHorizontal1(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    X1 is X + 1,
    returnPieceAt(Board, X1, Y, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 \= Pieceat,
    PieceAtD1 \= ' ',
    verifyCaptureHorizontal1aux(Board, X1, Y, Pieceat, Board2).

verifyCaptureHorizontal1(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    X1 is X + 1,
    returnPieceAt(Board, X1, Y, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 == Pieceat,
    Board = Board2.
```

```
verifyCaptureHorizontal1(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, _),
        X1 is X + 1,
        returnPieceAt(Board, X1, Y, PieceAtD1), /* diagonal esq baixo */
        PieceAtD1 == ' ',
        Board = Board2.

/**********/
verifyCaptureHorizontal1aux(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,  X,
Y, Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',
        X1 is X + 1,
        verifyCaptureHorizontal1aux2(Board, X1, Y, Pieceat, Board2). /* diagonal esq baixo */

verifyCaptureHorizontal1aux(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,  X,
Y, Pieceat),
        PieceAnterior == Pieceat,
        Board = Board2.

verifyCaptureHorizontal1aux(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board = Board2.
```

```
/**********/

verifyCaptureHorizontal1aux2(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board,  X,
Y, Pieceat),

        PieceAnterior == Pieceat,

        X1 is X + 1,

        verifyCaptureHorizontal1aux3(Board, X1, Y, Pieceat, Board2).


verifyCaptureHorizontal1aux2(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board,  X,
Y, Pieceat),

        PieceAnterior \= Pieceat,

        Board = Board2.


verifyCaptureHorizontal1aux2(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),

        Pieceat == ' ',

        Board = Board2.


/***********/

verifyCaptureHorizontal1aux3(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board,  X,
Y, Pieceat),

        PieceAnterior \= Pieceat,

        Pieceat \= ' ',
	write('\nPosition of capture, horizontal left to right\n'),

	X1 is X - 1,

	X2 is X1 - 1,

	changePieceAtCaptureHorizontal(Board,X1,X2,Y,Board2, Pieceat).


verifyCaptureHorizontal1aux3(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board,  X,
Y, Pieceat),

	PieceAnterior == Pieceat,

	Board = Board2.


verifyCaptureHorizontal1aux3(Board, X, Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),

	Pieceat == ' ',

	Board = Board2.


/*************************************************************************
**************************/







/******************************* ************** ***********************/
```

```
%horixzontal dir to esq

verifyCaptureHorizontal2(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),

        X1 is X - 1,

        returnPieceAt(Board, X1, Y, PieceAtD1), /* diagonal esq baixo */

        PieceAtD1 \= Pieceat,

        PieceAtD1 \= ' ',

        verifyCaptureHorizontal2aux(Board, X1, Y, Pieceat,Board2).


verifyCaptureHorizontal2(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),

        X1 is X - 1,

        returnPieceAt(Board, X1, Y, PieceAtD1), /* diagonal esq baixo */

        PieceAtD1 == Pieceat,

        Board = Board2.


verifyCaptureHorizontal2(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, _),

        X1 is X - 1,

        returnPieceAt(Board, X1, Y, PieceAtD1), /* diagonal esq baixo */

        PieceAtD1 == ' ',

        Board = Board2.


/**********/
```

```
verifyCaptureHorizontal2aux(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,  X,
Y, Pieceat),

        PieceAnterior \= Pieceat,

        Pieceat \= ' ',

        X1 is X - 1,

        verifyCaptureHorizontal2aux2(Board, X1, Y, Pieceat,Board2). /* diagonal esq baixo */


verifyCaptureHorizontal2aux(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,  X,
Y, Pieceat),

        PieceAnterior == Pieceat,

        Board = Board2.


verifyCaptureHorizontal2aux(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),

        Pieceat == ' ',

        Board = Board2.


/**********/

verifyCaptureHorizontal2aux2(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board,  X,
Y, Pieceat),

        PieceAnterior == Pieceat,

        X1 is X - 1,

        verifyCaptureHorizontal2aux3(Board, X1, Y, Pieceat,Board2).
```

```
verifyCaptureHorizontal2aux2(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board,  X,
Y, Pieceat),
        PieceAnterior \= Pieceat,
        Board = Board2.

verifyCaptureHorizontal2aux2(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board = Board2.

/***********/
verifyCaptureHorizontal2aux3(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board,  X,
Y, Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',
        write('\nPosition of capture, horizontal right to left\n'),
        X1 is X + 1,
        X2 is X1 + 1,
        changePieceAtCaptureHorizontal(Board,X1,X2,Y,Board2, Pieceat).

verifyCaptureHorizontal2aux3(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board,  X,
Y, Pieceat),
        PieceAnterior == Pieceat,
        Board = Board2.

verifyCaptureHorizontal2aux3(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board = Board2.

/***************************************************************************
****************************/
```

Anexo F - diagonalsPC.pl

```
/*****************************************************************  Piece
************************/

/*********************************diagonal 1 -> esq cima para baixo /
******************************************/

verifyCaptureDiagonal1PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    X1 is X,
    Y1 is Y + 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 \= Pieceat,
    PieceAtD1 \= ' ',
    verifyCaptureDiagonal1PCaux(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal1PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    X1 is X,
    Y1 is Y + 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 == Pieceat,
    Board2 = Board.

verifyCaptureDiagonal1PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, _),
    X1 is X,
    Y1 is Y + 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 == ' ',
    Board2 = Board.

verifyCaptureDiagonal1PCaux(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board, X,
Y, Pieceat),
    PieceAnterior \= Pieceat,
    Pieceat \= ' ',
    X1 is X,
    Y1 is Y + 1,
     verifyCaptureDiagonal1PCaux2(Board, X1, Y1, Pieceat, Board2). /* diagonal esq baixo
*/

verifyCaptureDiagonal1PCaux(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board, X,
Y, Pieceat),
    PieceAnterior == Pieceat,
    Board2 = Board.


verifyCaptureDiagonal1PCaux(Board,X,Y,_,Board2):-  returnPieceAt(Board, X, Y, Pieceat),
    Pieceat == ' ',

    Board2 = Board.



verifyCaptureDiagonal1PCaux2(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X,
Y, Pieceat),

    PieceAnterior == Pieceat,

    X1 is X,

    Y1 is Y + 1,

    verifyCaptureDiagonal1PCaux3(Board, X1, Y1, Pieceat, Board2).


verifyCaptureDiagonal1PCaux2(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X,
Y, Pieceat),

    PieceAnterior \= Pieceat,

    Board2 = Board.


verifyCaptureDiagonal1PCaux2(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),

    Pieceat == ' ',

    Board2 = Board.
```

```
verifyCaptureDiagonal1PCaux3(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X,
Y, Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',
        write('\nPosition of capture, diagonal left top to bottom\n'),
        X1 is X,
        Y1 is Y - 1,
        X2 is X1,
        Y2 is Y1 - 1,
        changePieceAtCapturePC(Board,X1,Y1,X2,Y2,Board2, Pieceat).

verifyCaptureDiagonal1PCaux3(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X,
Y, Pieceat),
        PieceAnterior == Pieceat,
        Board2 = Board.

verifyCaptureDiagonal1PCaux3(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board2 = Board.

/************************************************************************
**************************/
```

```
/*********************************diagonal 2 -> direita baixo para cima /
**************************/

/*****************************************************************
Piece *****************/

verifyCaptureDiagonal2PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    X1 is X,
    Y1 is Y - 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 \= Pieceat,
    PieceAtD1 \= ' ',
    verifyCaptureDiagonal2PCaux(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal2PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    X1 is X,
    Y1 is Y - 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 == Pieceat,
    Board2 = Board.
```

```
verifyCaptureDiagonal2PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, _),
        X1 is X,
        Y1 is Y - 1,
        returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
        PieceAtD1 == ' ',
        Board2 = Board.

verifyCaptureDiagonal2PCaux(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board, X,
Y, Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',
        X1 is X,
        Y1 is Y - 1,
         verifyCaptureDiagonal2PCaux2(Board, X1, Y1, Pieceat, Board2). /* diagonal esq baixo
*/

verifyCaptureDiagonal2PCaux(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board, X,
Y, Pieceat),
        PieceAnterior == Pieceat,
        Board2 = Board.
```

```
verifyCaptureDiagonal2PCaux(Board,X,Y,_,Board2):-  returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board2 = Board.



verifyCaptureDiagonal2PCaux2(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X,
Y, Pieceat),
        PieceAnterior == Pieceat,
        X1 is X,
        Y1 is Y - 1,
        verifyCaptureDiagonal2PCaux3(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal2PCaux2(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X,
Y, Pieceat),
        PieceAnterior \= Pieceat,
        Board2 = Board.

verifyCaptureDiagonal2PCaux2(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board2 = Board.
```

```
verifyCaptureDiagonal2PCaux3(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X,
Y, Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',
        write('\nPosition of capture, right diagonal low up\n'),
        X1 is X,
        Y1 is Y + 1,
        X2 is X1,
        Y2 is Y1 + 1,
        changePieceAtCapturePC(Board,X1,Y1,X2,Y2,Board2, Pieceat).

verifyCaptureDiagonal2PCaux3(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X,
Y, Pieceat),
        PieceAnterior == Pieceat,
        Board2 = Board.

verifyCaptureDiagonal2PCaux3(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board2 = Board.

/************************************************************************
**************************/
```

```
/******************************************************************
Piece ************************/
/********************************diagonal 3 -> direita peça cima para baixo \
*******************************************/

verifyCaptureDiagonal3PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    X1 is X + 1,
    Y1 is Y + 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 \= Pieceat,
    PieceAtD1 \= ' ',
    verifyCaptureDiagonal3PCaux(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal3PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    X1 is X + 1,
    Y1 is Y + 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 == Pieceat,
    Board2 = Board.
```

```
verifyCaptureDiagonal3PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, _),
        X1 is X + 1,
        Y1 is Y + 1,
        returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
        PieceAtD1 == ' ',
        Board2 = Board.

verifyCaptureDiagonal3PCaux(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board, X,
Y, Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',
        X1 is X + 1,
        Y1 is Y + 1,
         verifyCaptureDiagonal3PCaux2(Board, X1, Y1, Pieceat, Board2). /* diagonal esq baixo
*/

verifyCaptureDiagonal3PCaux(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board, X,
Y, Pieceat),
        PieceAnterior == Pieceat,
        Board2 = Board.
```

```
verifyCaptureDiagonal3PCaux(Board,X,Y,_,Board2):-  returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board2 = Board.



verifyCaptureDiagonal3PCaux2(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X,
Y, Pieceat),
        PieceAnterior == Pieceat,
        X1 is X + 1,
        Y1 is Y + 1,
        verifyCaptureDiagonal3PCaux3(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal3PCaux2(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X,
Y, Pieceat),
        PieceAnterior \= Pieceat,
        Board2 = Board.

verifyCaptureDiagonal3PCaux2(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board2 = Board.
```

```
verifyCaptureDiagonal3PCaux3(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X,
Y, Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',
        write('\nPosition of capture, diagonal right up down.\n'),
        X1 is X - 1,
        Y1 is Y - 1,
        X2 is X1 - 1,
        Y2 is Y1 - 1,
        changePieceAtCapturePC(Board,X1,Y1,X2,Y2,Board2, Pieceat).

verifyCaptureDiagonal3PCaux3(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X,
Y, Pieceat),
        PieceAnterior == Pieceat,
        Board2 = Board.

verifyCaptureDiagonal3PCaux3(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board2 = Board.
```

```
/*************************************************************************
**************************/





/**********************************diagonal 4 -> esquerda baixo para cima   \
*******************************************/

/***********************************************************************
Piece ******************/

verifyCaptureDiagonal4PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
      X1 is X - 1,
      Y1 is Y - 1,
      returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
      PieceAtD1 \= Pieceat,
      PieceAtD1 \= ' ',
      verifyCaptureDiagonal4PCaux(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal4PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
      X1 is X - 1,
      Y1 is Y - 1,
      returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
```

```
        PieceAtD1 == Pieceat,
        Board2 = Board.

verifyCaptureDiagonal4PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, _),
        X1 is X - 1,
        Y1 is Y - 1,
        returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
        PieceAtD1 == ' ',
        Board2 = Board.

verifyCaptureDiagonal4PCaux(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board, X,
Y, Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',
        X1 is X - 1,
        Y1 is Y - 1,
         verifyCaptureDiagonal4PCaux2(Board, X1, Y1, Pieceat, Board2). /* diagonal esq baixo
*/

verifyCaptureDiagonal4PCaux(Board,X,Y,PieceAnterior,Board2):-  returnPieceAt(Board, X,
Y, Pieceat),
        PieceAnterior == Pieceat,
        Board2 = Board.



verifyCaptureDiagonal4PCaux(Board,X,Y,_,Board2):-  returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board2 = Board.



verifyCaptureDiagonal4PCaux2(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X,
Y, Pieceat),
        PieceAnterior == Pieceat,
        X1 is X - 1,
        Y1 is Y - 1,
        verifyCaptureDiagonal4PCaux3(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal4PCaux2(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X,
Y, Pieceat),
        PieceAnterior \= Pieceat,
        Board2 = Board.

verifyCaptureDiagonal4PCaux2(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board2 = Board.



verifyCaptureDiagonal4PCaux3(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X,
Y, Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',
        write('\nPosition of capture, diagonal left low up\n'),
        X1 is X + 1,
        Y1 is Y + 1,
        X2 is X1 + 1,
        Y2 is Y1 + 1,
        changePieceAtCapturePC(Board,X1,Y1,X2,Y2,Board2, Pieceat).

verifyCaptureDiagonal4PCaux3(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board, X,
Y, Pieceat),
        PieceAnterior == Pieceat,
        Board2 = Board.

verifyCaptureDiagonal4PCaux3(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board2 = Board.
```

```
/*************************************************************************
***************************/


/*************************************************************************
*************************/

/*************************************************************************
****************************/

/*************************** diagonais parte de baixo do tabuleiro**********************************/

/*************************************************************************
*************************/

/*************************************************************************
****************************/



/************************************diagonal 1 -> esq cima para baixo
***********************************************/

verifyCaptureDiagonal11PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),

    X1 is X - 1,

    Y1 is Y + 1,

    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
```

```
        PieceAtD1 \= Pieceat,

        PieceAtD1 \= ' ',

        verifyCaptureDiagonal11PCaux(Board, X1, Y1, Pieceat, Board2).


verifyCaptureDiagonal11PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),

        X1 is X - 1,

        Y1 is Y + 1,

        returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */

        PieceAtD1 == Pieceat,

        Board2 = Board.


verifyCaptureDiagonal11PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, _),

        X1 is X - 1,

        Y1 is Y + 1,

        returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */

        PieceAtD1 == ' ',

        Board2 = Board.


verifyCaptureDiagonal11PCaux(Board,X,Y,PieceAnterior,Board2):-      returnPieceAt(Board,
X, Y, Pieceat),

        PieceAnterior \= Pieceat,

        Pieceat \= ' ',
        X1 is X - 1,

        Y1 is Y + 1,

            verifyCaptureDiagonal11PCaux2(Board, X1, Y1, Pieceat, Board2). /* diagonal esq
baixo */


verifyCaptureDiagonal11PCaux(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,
X, Y, Pieceat),

        PieceAnterior == Pieceat,

        Board2 = Board.



verifyCaptureDiagonal11PCaux(Board,X,Y,_,Board2):-        returnPieceAt(Board,    X,    Y,
Pieceat),

        Pieceat == ' ',

        Board2 = Board.



verifyCaptureDiagonal11PCaux2(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,
X, Y, Pieceat),

        PieceAnterior == Pieceat,

        X1 is X - 1,

        Y1 is Y + 1,
        verifyCaptureDiagonal11PCaux3(Board, X1, Y1, Pieceat, Board2).


verifyCaptureDiagonal11PCaux2(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,
X, Y, Pieceat),
        PieceAnterior \= Pieceat,
        Board2 = Board.


verifyCaptureDiagonal11PCaux2(Board,X,Y,_,Board2):-    returnPieceAt(Board,    X,    Y,
Pieceat),
        Pieceat == ' ',
        Board2 = Board.



verifyCaptureDiagonal11PCaux3(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,
X, Y, Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',
        write('\nPosition of capture, diagonal left top to bottom. Bottom of the board\n'),
        X1 is X + 1,
        Y1 is Y - 1,
        X2 is X1 + 1,
        Y2 is Y1 - 1,
        changePieceAtCapturePC(Board,X1,Y1,X2,Y2,Board2, Pieceat).

verifyCaptureDiagonal11PCaux3(Board,X,Y,PieceAnterior,Board2):- returnPieceAt(Board,
X, Y, Pieceat),
        PieceAnterior == Pieceat,
        Board2 = Board.

verifyCaptureDiagonal11PCaux3(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y,
Pieceat),
        Pieceat == ' ',
        Board2 = Board.

/**************************************************************************
***************************/




/***********************************diagonal 2 -> direita baixo para cima
*********************************************/

verifyCaptureDiagonal21PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        X1 is X + 1,
        Y1 is Y - 1,
        returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
        PieceAtD1 \= Pieceat,
        PieceAtD1 \= ' ',
        verifyCaptureDiagonal21PCaux(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal21PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        X1 is X + 1,
        Y1 is Y - 1,
        returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
        PieceAtD1 == Pieceat,
        Board2 = Board.

verifyCaptureDiagonal21PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, _),
        X1 is X + 1,
        Y1 is Y - 1,
        returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
        PieceAtD1 == ' ',
        Board2 = Board.

verifyCaptureDiagonal21PCaux(Board,X,Y,PieceAnterior,Board2):-      returnPieceAt(Board,
X, Y, Pieceat),
        PieceAnterior \= Pieceat,

        Pieceat \= ' ',

        X1 is X + 1,

        Y1 is Y - 1,

            verifyCaptureDiagonal21PCaux2(Board, X1, Y1, Pieceat, Board2). /* diagonal esq
baixo */


verifyCaptureDiagonal21PCaux(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,
X, Y, Pieceat),

        PieceAnterior == Pieceat,

        Board2 = Board.



verifyCaptureDiagonal21PCaux(Board,X,Y,_,Board2):-    returnPieceAt(Board,    X,    Y,
Pieceat),

        Pieceat == ' ',

        Board2 = Board.



verifyCaptureDiagonal21PCaux2(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,
X, Y, Pieceat),

        PieceAnterior == Pieceat,
        X1 is X + 1,

        Y1 is Y - 1,

        verifyCaptureDiagonal21PCaux3(Board, X1, Y1, Pieceat, Board2).


verifyCaptureDiagonal21PCaux2(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,
X, Y, Pieceat),

        PieceAnterior \= Pieceat,

        Board2 = Board.


verifyCaptureDiagonal21PCaux2(Board,X,Y,_,Board2):-    returnPieceAt(Board,    X,    Y,
Pieceat),

        Pieceat == ' ',

        Board2 = Board.



verifyCaptureDiagonal21PCaux3(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,
X, Y, Pieceat),

        PieceAnterior \= Pieceat,

        Pieceat \= ' ',

        write('\nPosition of capture, right diagonal low up. Bottom of the board.\n'),

        X1 is X - 1,

        Y1 is Y + 1,
    X2 is X1 - 1,

    Y2 is Y1 + 1,

    changePieceAtCapturePC(Board,X1,Y1,X2,Y2,Board2, Pieceat).


verifyCaptureDiagonal21PCaux3(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,
X, Y, Pieceat),

    PieceAnterior == Pieceat,

    Board2 = Board.


verifyCaptureDiagonal21PCaux3(Board,X,Y,_,Board2):-    returnPieceAt(Board,    X,    Y,
Pieceat),

    Pieceat == ' ',

    Board2 = Board.


/****************************************************************************
***************************/






/***********************************diagonal  3  ->  direita  cima  para  baixo
**********************************************/
```

```
verifyCaptureDiagonal31PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    X1 is X,
    Y1 is Y + 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 \= Pieceat,
    PieceAtD1 \= ' ',
    verifyCaptureDiagonal31PCaux(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal31PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
    X1 is X,
    Y1 is Y + 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 == Pieceat,
    Board2 = Board.

verifyCaptureDiagonal31PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, _),
    X1 is X,
    Y1 is Y + 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 == ' ',
    Board2 = Board.
```

```
verifyCaptureDiagonal31PCaux(Board,X,Y,PieceAnterior,Board2):-     returnPieceAt(Board,
X, Y, Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',
        X1 is X,
        Y1 is Y + 1,
              verifyCaptureDiagonal31PCaux2(Board, X1, Y1, Pieceat, Board2). /* diagonal esq
baixo */


verifyCaptureDiagonal31PCaux(Board,X,Y,PieceAnterior,Board2):-     returnPieceAt(Board,
X, Y, Pieceat),
        PieceAnterior == Pieceat,
        Board2 = Board.



verifyCaptureDiagonal31PCaux(Board,X,Y,_,Board2):-     returnPieceAt(Board,     X,     Y,
Pieceat),
        Pieceat == ' ',
        Board2 = Board.
```

```
verifyCaptureDiagonal31PCaux2(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board, X, Y, Pieceat),
        PieceAnterior == Pieceat,
        X1 is X,
        Y1 is Y + 1,
        verifyCaptureDiagonal31PCaux3(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal31PCaux2(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board, X, Y, Pieceat),
        PieceAnterior \= Pieceat,
        Board2 = Board.

verifyCaptureDiagonal31PCaux2(Board,X,Y,_,Board2):-    returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board2 = Board.


verifyCaptureDiagonal31PCaux3(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board, X, Y, Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',
        write('\nPosition of capture, diagonal right up down. Bottom of the board.\n'),

        X1 is X,

        Y1 is Y - 1,

        X2 is X1,

        Y2 is Y1 - 1,

        changePieceAtCapturePC(Board,X1,Y1,X2,Y2,Board2, Pieceat).


    verifyCaptureDiagonal31PCaux3(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,
    X, Y, Pieceat),

        PieceAnterior == Pieceat,

        Board2 = Board.


    verifyCaptureDiagonal31PCaux3(Board,X,Y,_,Board2):-    returnPieceAt(Board,    X,    Y,
    Pieceat),

        Pieceat == ' ',

        Board2 = Board.


    /*************************************************************************
    ****************************/
```

```
/**********************************diagonal 4 -> esquerda baixo para cima
*******************************************/


verifyCaptureDiagonal41PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),

    X1 is X,

    Y1 is Y - 1,

    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */

    PieceAtD1 \= Pieceat,

    PieceAtD1 \= ' ',

    verifyCaptureDiagonal41PCaux(Board, X1, Y1, Pieceat, Board2).


verifyCaptureDiagonal41PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),

    X1 is X,

    Y1 is Y - 1,

    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */

    PieceAtD1 == Pieceat,

    Board2 = Board.


verifyCaptureDiagonal41PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, _),

    X1 is X,

    Y1 is Y - 1,
    returnPieceAt(Board, X1, Y1, PieceAtD1), /* diagonal esq baixo */
    PieceAtD1 == ' ',
    Board2 = Board.

verifyCaptureDiagonal41PCaux(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,
X, Y, Pieceat),
    PieceAnterior \= Pieceat,
    Pieceat \= ' ',
    X1 is X,
    Y1 is Y - 1,
        verifyCaptureDiagonal41PCaux2(Board, X1, Y1, Pieceat, Board2). /* diagonal esq
baixo */

verifyCaptureDiagonal41PCaux(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,
X, Y, Pieceat),
    PieceAnterior == Pieceat,
    Board2 = Board.



verifyCaptureDiagonal41PCaux(Board,X,Y,_,Board2):-    returnPieceAt(Board,   X,   Y,
Pieceat),
    Pieceat == ' ',
    Board2 = Board.


verifyCaptureDiagonal41PCaux2(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,
X, Y, Pieceat),
    PieceAnterior == Pieceat,
    X1 is X,
    Y1 is Y - 1,
    verifyCaptureDiagonal41PCaux3(Board, X1, Y1, Pieceat, Board2).

verifyCaptureDiagonal41PCaux2(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,
X, Y, Pieceat),
    PieceAnterior \= Pieceat,
    Board2 = Board.

verifyCaptureDiagonal41PCaux2(Board,X,Y,_,Board2):-    returnPieceAt(Board,    X,    Y,
Pieceat),
    Pieceat == ' ',
    Board2 = Board.
```

```
verifyCaptureDiagonal41PCaux3(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,
X, Y, Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',
        write('\nPosition of capture, diagonal left low up. Bottom of the board\n'),
        X1 is X,
        Y1 is Y + 1,
        X2 is X1,
        Y2 is Y1 + 1,
        changePieceAtCapturePC(Board,X1,Y1,X2,Y2,Board2, Pieceat).

verifyCaptureDiagonal41PCaux3(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,
X, Y, Pieceat),
        PieceAnterior == Pieceat,
        Board2 = Board.

verifyCaptureDiagonal41PCaux3(Board,X,Y,_,Board2):-    returnPieceAt(Board,    X,    Y,
Pieceat),
        Pieceat == ' ',
        Board2 = Board.
```

```
/*****************************************************************************************************/



/********************************* horizontal capture ************************/

%horizontal esq to dir
verifyCaptureHorizontal1PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
	X1 is X + 1,
	returnPieceAt(Board, X1, Y, PieceAtD1), /* diagonal esq baixo */
	PieceAtD1 \= Pieceat,
	PieceAtD1 \= ' ',
	verifyCaptureHorizontal1PCaux(Board, X1, Y, Pieceat, Board2).

verifyCaptureHorizontal1PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
	X1 is X + 1,
	returnPieceAt(Board, X1, Y, PieceAtD1), /* diagonal esq baixo */
	PieceAtD1 == Pieceat,
	Board2 = Board.
```

```
verifyCaptureHorizontal1PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, _),
        X1 is X + 1,
        returnPieceAt(Board, X1, Y, PieceAtD1), /* diagonal esq baixo */
        PieceAtD1 == ' ',
        Board2 = Board.

/**********/
verifyCaptureHorizontal1PCaux(Board,X,Y,PieceAnterior,Board2):-     returnPieceAt(Board,
X, Y, Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',
        X1 is X + 1,
         verifyCaptureHorizontal1PCaux2(Board, X1, Y, Pieceat, Board2). /* diagonal esq baixo
*/

verifyCaptureHorizontal1PCaux(Board,X,Y,PieceAnterior,Board2):-     returnPieceAt(Board,
X, Y, Pieceat),
        PieceAnterior == Pieceat,
        Board2 = Board.

verifyCaptureHorizontal1PCaux(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board2 = Board.

/**********/
verifyCaptureHorizontal1PCaux2(Board,X,Y,PieceAnterior,Board2):-   returnPieceAt(Board,
X, Y, Pieceat),
        PieceAnterior == Pieceat,
        X1 is X + 1,
        verifyCaptureHorizontal1PCaux3(Board, X1, Y, Pieceat, Board2).

verifyCaptureHorizontal1PCaux2(Board,X,Y,PieceAnterior,Board2):-   returnPieceAt(Board,
X, Y, Pieceat),
        PieceAnterior \= Pieceat,
        Board2 = Board.

verifyCaptureHorizontal1PCaux2(Board,X,Y,_,Board2):-   returnPieceAt(Board,   X,   Y,
Pieceat),
        Pieceat == ' ',
        Board2 = Board.

/***********/
verifyCaptureHorizontal1PCaux3(Board,X,Y,PieceAnterior,Board2):-   returnPieceAt(Board,
X, Y, Pieceat),
        PieceAnterior \= Pieceat,

        Pieceat \= ' ',

        write('\nPosition of capture, horizontal left to right\n'),

        X1 is X - 1,

        X2 is X1 - 1,

        changePieceAtCaptureHorizontalPC(Board,X1,X2,Y,Board2, Pieceat).


verifyCaptureHorizontal1PCaux3(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,
X, Y, Pieceat),

        PieceAnterior == Pieceat,

        Board2 = Board.


verifyCaptureHorizontal1PCaux3(Board,    X,    Y,Board2):-    returnPieceAt(Board,    X,    Y,
Pieceat),

        Pieceat == ' ',

        Board2 = Board.


/**************************************************************************
***************************/
```

```
/***************************** ************** ***********************/

%horixzontal dir to esq
verifyCaptureHorizontal2PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        X1 is X - 1,
        returnPieceAt(Board, X1, Y, PieceAtD1), /* diagonal esq baixo */
        PieceAtD1 \= Pieceat,
        PieceAtD1 \= ' ',
        verifyCaptureHorizontal2PCaux(Board, X1, Y, Pieceat,Board2).

verifyCaptureHorizontal2PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        X1 is X - 1,
        returnPieceAt(Board, X1, Y, PieceAtD1), /* diagonal esq baixo */
        PieceAtD1 == Pieceat,
        Board2 = Board.

verifyCaptureHorizontal2PC(Board,X,Y,Board2):- returnPieceAt(Board, X, Y, _),
        X1 is X - 1,
        returnPieceAt(Board, X1, Y, PieceAtD1), /* diagonal esq baixo */
        PieceAtD1 == ' ',
        Board2 = Board.
```

```
/**********/
verifyCaptureHorizontal2PCaux(Board,X,Y,PieceAnterior,Board2):-     returnPieceAt(Board,
X, Y, Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',
        X1 is X - 1,
          verifyCaptureHorizontal2PCaux2(Board, X1, Y, Pieceat,Board2). /* diagonal esq baixo
*/

verifyCaptureHorizontal2PCaux(Board,X,Y,PieceAnterior,Board2):-     returnPieceAt(Board,
X, Y, Pieceat),
        PieceAnterior == Pieceat,
        Board2 = Board.

verifyCaptureHorizontal2PCaux(Board,X,Y,_,Board2):- returnPieceAt(Board, X, Y, Pieceat),
        Pieceat == ' ',
        Board2 = Board.

/**********/
verifyCaptureHorizontal2PCaux2(Board,X,Y,PieceAnterior,Board2):-    returnPieceAt(Board,
X, Y, Pieceat),
        PieceAnterior == Pieceat,
        X1 is X - 1,
        verifyCaptureHorizontal2PCaux3(Board, X1, Y, Pieceat,Board2).

verifyCaptureHorizontal2PCaux2(Board,X,Y,PieceAnterior,Board2):-   returnPieceAt(Board,
X, Y, Pieceat),
        PieceAnterior \= Pieceat,
        Board2 = Board.

verifyCaptureHorizontal2PCaux2(Board,X,Y,_,Board2):-   returnPieceAt(Board,   X,   Y,
Pieceat),
        Pieceat == ' ',
        Board2 = Board.

/***********/
verifyCaptureHorizontal2PCaux3(Board,X,Y,PieceAnterior,Board2):-   returnPieceAt(Board,
X, Y, Pieceat),
        PieceAnterior \= Pieceat,
        Pieceat \= ' ',
        write('\nPosition of capture, horizontal right to left\n'),
        X1 is X + 1,
        X2 is X1 + 1,
        changePieceAtCaptureHorizontalPC(Board,X1,X2,Y,Board2, Pieceat).
```

```
verifyCaptureHorizontal2PCaux3(Board,X,Y,PieceAnterior,Board2):-   returnPieceAt(Board,
X, Y, Pieceat),
        PieceAnterior == Pieceat,
        Board2 = Board.

verifyCaptureHorizontal2PCaux3(Board,X,Y,_,Board2):-   returnPieceAt(Board,   X,   Y,
Pieceat),
        Pieceat == ' ',
        Board2 = Board.

/************************************************************************
**************************/

%linha1 ate linha 3
verifyCaptureDiagonal1PCPC(Board,X,Y,BoardR):- Y < 4,
        verifyCaptureDiagonal1(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%linha1 ate linha 3
verifyCaptureDiagonalsPC(Board,X,Y,Board2):- Y < 4,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal3PC(Board,X,Y,Board2).
%linha4
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 4,
        X == 1,
        verifyCaptureDiagonal2PC(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%linha4
verifyCaptureDiagonalsPC(Board,X,Y,Board2):- Y == 4,
        X == 1,
        verifyCaptureDiagonal2PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal3PC(Board,X,Y,Board2).
%%linha4
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 4,
        X > 1,
        X < 4,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal2PC(Board,X,Y,Board2),
        Board \= Board2,
        Board2 = BoardR.

%%linha4
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 4,
        X > 1,
        X < 4,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%%linha4
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 4,
        X > 1,
        X < 4,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal2PC(Board,X,Y,Board2),
        Board == Board2,
        verifyCaptureDiagonal3PC(Board,X,Y,BoardR).
%%linha4
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 4,
        X > 3,
        X < 6,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal2PC(Board,X,Y,Board2),
        Board == Board2,
        verifyCaptureDiagonal3PC(Board,X,Y,Board3),
        Board \= Board3,
        Board3 = BoardR.
%%linha4
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 4,
        X > 3,
        X < 6,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal2PC(Board,X,Y,Board2),
        Board \= Board2,
        Board2 = BoardR.
%%linha4
```

```
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 4,
        X > 3,
        X < 6,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%%linha4
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 4,
        X > 3,
        X < 6,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal2PC(Board,X,Y,Board2),
        Board == Board2,
        verifyCaptureDiagonal3PC(Board,X,Y,Board3),
        Board == Board3,
        verifyCaptureDiagonal4PC(Board,X,Y,BoardR).
%linha4
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 4,
        X > 5,
        X < 8,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal2PC(Board,X,Y,Board2),
        Board \= Board2,
        Board2 = BoardR.
%%linha4
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 4,
        X > 5,
        X < 8,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%linha4
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 4,
        X > 5,
        X < 8,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal2PC(Board,X,Y,Board2),
        Board == Board2,
        verifyCaptureDiagonal4PC(Board,X,Y,BoardR).
%linha4
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 4,
        X == 8,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%linha4
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 4,
        X == 8,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal4PC(Board,X,Y,BoardR).
%linha5
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 5,
        X < 3,
        verifyCaptureDiagonal2PC(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%linha5
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 5,
        X < 3,
        verifyCaptureDiagonal2PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal3PC(Board,X,Y,BoardR).
```

```
%%linha5
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 5,
    X == 3,
    verifyCaptureDiagonal1PC(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal2PC(Board,X,Y,Board2),
    Board \= Board2,
    Board2 = BoardR.
%%linha5
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 5,
    X == 3,
    verifyCaptureDiagonal1PC(Board,X,Y,Board1),
    Board \= Board1,
    Board1 = BoardR.
%%linha5
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 5,
    X == 3,
    verifyCaptureDiagonal1PC(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal2PC(Board,X,Y,Board2),
    Board == Board2,
    verifyCaptureDiagonal3PC(Board,X,Y,BoardR).
```

```
%%linha5
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 5,
        X > 3,
        X < 7,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal2PC(Board,X,Y,Board2),
        Board == Board2,
        verifyCaptureDiagonal3PC(Board,X,Y,Board3),
        Board \= Board3,
        Board3 = BoardR.
%%linha5
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 5,
        X > 3,
        X < 7,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal2PC(Board,X,Y,Board2),
        Board \= Board2,
        Board2 = BoardR.
%%linha5
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 5,
        X > 3,
        X < 7,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%linha5
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 5,
        X > 3,
        X < 7,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal2PC(Board,X,Y,Board2),
        Board == Board2,
        verifyCaptureDiagonal3PC(Board,X,Y,Board3),
        Board == Board3,
        verifyCaptureDiagonal4PC(Board,X,Y,BoardR).
%linha5
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 5,
        X == 7,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal2PC(Board,X,Y,Board2),
        Board \= Board2,
        Board2 = BoardR.
%linha5
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 5,
        X == 7,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%linha5
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 5,
        X > 7,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal4PC(Board,X,Y,BoardR).
%linha5
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 5,
        X > 7,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%linha5
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 5,
        X == 7,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal2PC(Board,X,Y,Board2),
        Board == Board2,
        verifyCaptureDiagonal4PC(Board,X,Y,BoardR).
%%linha 6
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 6,
        X < 4,
        verifyCaptureDiagonal3PC(Board,X,Y,Board1),
        board \= 1,
        Board1 = BoardR.
%%linha 6
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 6,
        X < 4,
        verifyCaptureDiagonal3PC(Board,X,Y,Board1),
        board == Board1,
        verifyCaptureDiagonal2PC(Board,X,Y,BoardR).
%%linha 6
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 6,
        X > 3,
        X < 7,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%%linha 6
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 6,
        X > 3,
        X < 7,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal2PC(Board,X,Y,Board2),
        Board \= Board2,
        Board2 = BoardR.
%%linha 6
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 6,
        X > 3,
        X < 7,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal2PC(Board,X,Y,Board2),
        Board == Board2,
        verifyCaptureDiagonal3PC(Board,X,Y,Board3),
        Board \= Board3,
        Board3 = BoardR.
%%linha 6
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 6,
        X > 3,
        X < 7,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal2PC(Board,X,Y,Board2),
        Board == Board2,
        verifyCaptureDiagonal3(Board,X,Y,Board3),
        Board == Board3,
        verifyCaptureDiagonal4PC(Board,X,Y,BoardR).
%%linha 6
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 6,
        X > 6,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%%linha 6
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 6,
        X > 6,
        verifyCaptureDiagonal1PC(Board,X,Y,Board1),
	Board == Board1,
	verifyCaptureDiagonal4PC(Board,X,Y,BoardR).

/**********************************************************/
/******************* parte de baixo do tabuleiro********************/

%linha11
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y > 8,
	Y < 12,
	X > 0,
	verifyCaptureDiagonal41PC(Board,X,Y,Board1),
	Board \= Board1,
	Board1 = BoardR.
%linha11
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y > 8,
	Y < 12,
	X > 0,
	verifyCaptureDiagonal41PC(Board,X,Y,Board1),
	Board == Board1,
	verifyCaptureDiagonal21PC(Board,X,Y,BoardR).
%linha8
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 8,
    X == 1,
    verifyCaptureDiagonal21PC(Board,X,Y,Board1),
    Board \= Board1,
    Board1 = BoardR.
%linha8
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 8,
    X == 1,
    verifyCaptureDiagonal21PC(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal31PC(Board,X,Y,BoardR).
%linha8
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 8,
    X > 1,
    X < 4,
    verifyCaptureDiagonal41PC(Board,X,Y,Board1),
    Board \= Board1,
    Board1 = BoardR.
%linha8
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 8,
    X > 1,
    X < 4,
    verifyCaptureDiagonal41PC(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal21PC(Board,X,Y,Board2),
    Board \= Board2,
    Board1 = BoardR.
%linha8
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 8,
    X > 1,
    X < 4,
    verifyCaptureDiagonal41PC(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal21PC(Board,X,Y,Board2),
    Board == Board2,
    verifyCaptureDiagonal31PC(Board,X,Y,BoardR).
%linha8
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 8,
    X > 3,
    X < 6,
    verifyCaptureDiagonal11PC(Board,X,Y,Board1),
    Board \= Board1,
    Board1 = BoardR.
%linha8
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 8,
    X > 3,
    X < 6,
    verifyCaptureDiagonal11PC(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal21PC(Board,X,Y,Board2),
    Board \= Board2,
    Board2 = BoardR.
%linha8
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 8,
    X > 3,
    X < 6,
    verifyCaptureDiagonal11PC(Board,X,Y,Board1),
    Board == Board1,
    verifyCaptureDiagonal21PC(Board,X,Y,Board2),
    Board == Board2,
    verifyCaptureDiagonal31PC(Board,X,Y,Board3),
    Board \= Board3,
    Board3 = BoardR.
%linha8
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 8,
    X > 3,
    X < 6,
	verifyCaptureDiagonal11PC(Board,X,Y,Board1),
	Board == Board1,
	verifyCaptureDiagonal21PC(Board,X,Y,Board2),
	Board == Board2,
	verifyCaptureDiagonal31PC(Board,X,Y,Board3),
	Board == Board3,
	verifyCaptureDiagonal41PC(Board,X,Y,BoardR).
%linha8
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 8,
	X > 5,
	X < 8,
	verifyCaptureDiagonal11PC(Board,X,Y,Board1),
	Board \= Board1,
	Board1 = BoardR.
%linha8
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 8,
	X > 5,
	X < 8,
	verifyCaptureDiagonal11PC(Board,X,Y,Board1),
	Board == Board1,
	verifyCaptureDiagonal21PC(Board,X,Y,Board2),
	Board \= Board2,
        Board2 = BoardR.
%linha8
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 8,
        X > 5,
        X < 8,
        verifyCaptureDiagonal11PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal21PC(Board,X,Y,Board2),
        Board == Board2,
        verifyCaptureDiagonal41PC(Board,X,Y,BoardR).
%linha8
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 8,
        X == 8,
        verifyCaptureDiagonal11PC(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%linha8
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 8,
        X == 8,
        verifyCaptureDiagonal11PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal41PC(Board,X,Y,BoardR).
```

```
%linha7
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 7,
        X < 3,
        verifyCaptureDiagonal21PC(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%linha7
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 7,
        X < 3,
        verifyCaptureDiagonal21PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal31PC(Board,X,Y,BoardR).
%linha7
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 7,
        X == 3,
        verifyCaptureDiagonal41PC(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%linha7
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 7,
        X == 3,
        verifyCaptureDiagonal41PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal21PC(Board,X,Y,Board2),
        Board \= Board2,
        Board2 = BoardR.
%linha7
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 7,
        X == 3,
        verifyCaptureDiagonal41PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal21PC(Board,X,Y,Board2),
        Board == Board2,
        verifyCaptureDiagonal31PC(Board,X,Y,BoardR).
%linha7
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 7,
        X > 3,
        X < 7,
        verifyCaptureDiagonal11PC(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%linha7
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 7,
        X > 3,
        X < 7,
        verifyCaptureDiagonal11PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal21PC(Board,X,Y,Board2),
        Board \= Board2,
        Board2 = BoardR.
%linha7
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 7,
        X > 3,
        X < 7,
        verifyCaptureDiagonal11PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal21PC(Board,X,Y,Board2),
        Board == Board2,
        verifyCaptureDiagonal31PC(Board,X,Y,Board3),
        Board \= Board3,
        Board3 = BoardR.
%linha7
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 7,
        X > 3,
        X < 7,
        verifyCaptureDiagonal11PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal21PC(Board,X,Y,Board2),
        Board == Board2,
        verifyCaptureDiagonal31PC(Board,X,Y,Board3),
        Board == Board3,
        verifyCaptureDiagonal41PC(Board,X,Y,BoardR).
%linha7
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 7,
        X == 7,
        verifyCaptureDiagonal11PC(Board,X,Y,Board1),
        Board \= Board1,
        Board1 = BoardR.
%linha7
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 7,
        X == 7,
        verifyCaptureDiagonal11PC(Board,X,Y,Board1),
        Board == Board1,
        verifyCaptureDiagonal21PC(Board,X,Y,Board2),
        Board \= Board2,
        Board2 = BoardR.
%linha7
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 7,
	X == 7,
	verifyCaptureDiagonal11PC(Board,X,Y,Board1),
	Board == Board1,
	verifyCaptureDiagonal21PC(Board,X,Y,Board2),
	Board == Board2,
	verifyCaptureDiagonal41PC(Board,X,Y,BoardR).
%linha7
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 7,
	X > 7,
	verifyCaptureDiagonal11PC(Board,X,Y,Board1),
	Board \= Board1,
	Board1 = BoardR.
%linha7
verifyCaptureDiagonalsPC(Board,X,Y,BoardR):- Y == 7,
	X > 7,
	verifyCaptureDiagonal11PC(Board,X,Y,Board1),
	Board == Board1,
	verifyCaptureDiagonal41PC(Board,X,Y,BoardR).


/*************************************************************************
****************/
```

```
/****************************** verifical captura na horizontal
************************/
%linha1
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 1,
    X == 3,
    append([],Board,Board2).
%linha1
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 1,
    X < 3,
    verifyCaptureHorizontal1PC(Board,X,Y,Board2).
%linha1
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 1,
    X > 3,
    verifyCaptureHorizontal2PC(Board,X,Y,Board2).
%linha2
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 2,
    X < 4,
    verifyCaptureHorizontal1PC(Board,X,Y,Board2).
%linha2
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 2,
    X > 3,
	verifyCaptureHorizontal2PC(Board,X,Y,Board2).
%linha3
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 3,
	X < 5,
	verifyCaptureHorizontal1PC(Board,X,Y,Board2).
%linha3
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 3,
	X > 3,
	verifyCaptureHorizontal2PC(Board,X,Y,Board2).
%linha4
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 4,
	X < 6,
	verifyCaptureHorizontal1PC(Board,X,Y,Board2).
%linha4
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 4,
	X > 3,
	verifyCaptureHorizontal2PC(Board,X,Y,Board2).
%linha5
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 5,
	X < 7,
	verifyCaptureHorizontal1PC(Board,X,Y,Board2).
%linha5
```

```
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 5,
        X > 3,
        verifyCaptureHorizontal2PC(Board,X,Y,Board2).
%linha6
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 6,
        X < 8,
        verifyCaptureHorizontal1PC(Board,X,Y,Board2).
%linha6
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 6,
        X > 3,
        verifyCaptureHorizontal2PC(Board,X,Y,Board2).
%linha7
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 7,
        X < 7,
        verifyCaptureHorizontal1PC(Board,X,Y,Board2).
%linha7
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 7,
        X > 3,
        verifyCaptureHorizontal2PC(Board,X,Y,Board2).
%linha8
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 8,
        X < 6,
	verifyCaptureHorizontal1PC(Board,X,Y,Board2).
%linha8
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 8,
	X > 3,
	verifyCaptureHorizontal2PC(Board,X,Y,Board2).
%linha9
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 9,
	X < 5,
	verifyCaptureHorizontal1PC(Board,X,Y,Board2).
%linha9
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 9,
	X > 3,
	verifyCaptureHorizontal2PC(Board,X,Y,Board2).
%linha10
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 10,
	X == 4,
	append([],Board,Board2).
%linha10
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 10,
	X < 4,
	verifyCaptureHorizontal1PC(Board,X,Y,Board2).
%linha10
```

```
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 10,
        X > 3,
        verifyCaptureHorizontal2PC(Board,X,Y,Board2).
%linha11
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 11,
        X < 3,
        verifyCaptureHorizontal1PC(Board,X,Y,Board2).
%linha11
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 11,
        X > 3,
        verifyCaptureHorizontal2PC(Board,X,Y,Board2).
%linha11
verifyCaptureHorizontalPC(Board,X,Y,Board2):- Y == 11,
        X == 3,
        append([],Board,Board2).

/*************************************************************/
```

Anexo G - print.pl

```
/***************************

* OUTPUT RELATED FUNCTIONS *

***************************/


printMenu(_):- write('\n BOKU GAME - PROLOG VERSION 1.0'),

       write('\n\n').




/*Printing an entire board */

printBoard([]).

printBoard([X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12]) :-

       printLine1(X1),

       printLine2(X2),

       printLine3(X3),

       printLine4(X4),

       printLine5(X5),

       printLine6(X6),

       printLine7(X7),

       printLine8(X8),

       printLine9(X9),
        printLine10(X10),

        printLine11(X11),

        printLine12(X12).


printLiteralLine([X|Xs]) :- write(X), printLiteralLine(Xs).




printLine1([X|Xs]) :- write('1-      '), printLine1aux([X|Xs]).

printLine1aux([]) :- write('|\n').

printLine1aux([X|Xs]) :- X \== 'E', write('|'), write(X), printLine1aux(Xs).

printLine1aux([X|Xs]) :- X == 'E', printLine1aux(Xs).


printLine2([X|Xs]) :- write('2-     '), printLine2aux([X|Xs]).

printLine2aux([]) :- write('|\n').

printLine2aux([X|Xs]) :- X \== 'E', write('|'), write(X), printLine2aux(Xs).

printLine2aux([X|Xs]) :- X == 'E', printLine2aux(Xs).


printLine3([X|Xs]) :- write('3-    '), printLine3aux([X|Xs]).

printLine3aux([]) :- write('|\n').

printLine3aux([X|Xs]) :- X \== 'E', write('|'), write(X), printLine3aux(Xs).

printLine3aux([X|Xs]) :- X == 'E', printLine3aux(Xs).
```

```
printLine4([X|Xs]) :- write('4-   '), printLine4aux([X|Xs]).

printLine4aux([]) :- write('|\n').

printLine4aux([X|Xs]) :- X \== 'E', write('|'), write(X), printLine4aux(Xs).

printLine4aux([X|Xs]) :- X == 'E', printLine4aux(Xs).


printLine5([X|Xs]) :- write('5-  '), printLine5aux([X|Xs]).

printLine5aux([]) :- write('|\n').

printLine5aux([X|Xs]) :- X \== 'E', write('|'), write(X), printLine5aux(Xs).

printLine5aux([X|Xs]) :- X == 'E', printLine5aux(Xs).


printLine6([X|Xs]) :- write('6- '), printLine6aux([X|Xs]).

printLine6aux([]) :- write('|\n').

printLine6aux([X|Xs]) :- X \== 'E', write('|'), write(X), printLine6aux(Xs).

printLine6aux([X|Xs]) :- X == 'E', printLine6aux(Xs).


printLine7([X|Xs]) :- write('7-  '), printLine7aux([X|Xs]).

printLine7aux([]) :- write('|\n').

printLine7aux([X|Xs]) :- X \== 'E', write('|'), write(X), printLine7aux(Xs).

printLine7aux([X|Xs]) :- X == 'E', printLine7aux(Xs).


printLine8([X|Xs]) :- write('8-   '), printLine8aux([X|Xs]).

printLine8aux([]) :- write('|\n').
```

```
printLine8aux([X|Xs]) :- X \== 'E', write('|'), write(X), printLine8aux(Xs).

printLine8aux([X|Xs]) :- X == 'E', printLine8aux(Xs).


printLine9([X|Xs]) :- write('9-    '), printLine9aux([X|Xs]).

printLine9aux([]) :- write('|\n').

printLine9aux([X|Xs]) :- X \== 'E', write('|'), write(X), printLine9aux(Xs).

printLine9aux([X|Xs]) :- X == 'E', printLine9aux(Xs).


printLine10([X|Xs]) :- write('10-    '), printLine10aux([X|Xs]).

printLine10aux([]) :- write('|\n').

printLine10aux([X|Xs]) :- X \== 'E', write('|'), write(X), printLine10aux(Xs).

printLine10aux([X|Xs]) :- X == 'E', printLine10aux(Xs).


printLine11([X|Xs]) :- write('11-     '), printLine11aux([X|Xs]).

printLine11aux([]) :- write('|\n').

printLine11aux([X|Xs]) :- X \== 'E', write('|'), write(X), printLine11aux(Xs).

printLine11aux([X|Xs]) :- X == 'E', printLine11aux(Xs).


%print dos numero em baixo

printLine12([X|Xs]) :- write('   '), printLine12aux([X|Xs]).

printLine12aux([]) :- write('|\n').

printLine12aux([X|Xs]) :- X \== 'E', write('|'), write(X), printLine12aux(Xs).
```

```
printLine12aux([X|Xs]) :- X == 'E', printLine12aux(Xs).
```

Anexo H - win_conditions.pl

```
/************************************** win conditions **************************************/



%Para verificar se a jogada é uma jogada de win
isWinCondition(Board,X,Y) :- verifyWinDiagonals(Board, X,Y).
isWinCondition(Board,X,Y) :- verifyWinHorizontal(Board, X,Y).



/******************* parte de cima do tabuleiro*********************/
%linha1 ate linha 2
verifyWinDiagonals(Board,X,Y):- Y < 3,
	verifyWinDiagonal1(Board, X, Y).
%linha1 ate linha 2
verifyWinDiagonals(Board,X,Y):- Y < 3,
	verifyWinDiagonal3(Board, X, Y).
%linha3
verifyWinDiagonals(Board,X,Y):- Y == 3,
	X == 1,
	verifyWinDiagonal3(Board, X, Y).
```

```
%linha3
verifyWinDiagonals(Board,X,Y):- Y == 3,
        X > 1,
        X < 7,
        verifyWinDiagonal1(Board, X, Y).
%linha3
verifyWinDiagonals(Board,X,Y):- Y == 3,
        X > 1,
        X < 7,
        verifyWinDiagonal3(Board, X, Y).
%linha3
verifyWinDiagonals(Board,X,Y):- Y == 3,
        X == 7,
        verifyWinDiagonal1(Board, X, Y).
%linha4
verifyWinDiagonals(Board,X,Y):- Y == 4,
        X < 3,
        verifyWinDiagonal3(Board, X, Y).
%linha4
verifyWinDiagonals(Board,X,Y):- Y == 4,
        X > 2,
        X < 7,
    verifyWinDiagonal1(Board, X, Y).
%linha4
verifyWinDiagonals(Board,X,Y):- Y == 4,
    X > 2,
    X < 7,
    verifyWinDiagonal3(Board, X, Y).
%linha4
verifyWinDiagonals(Board,X,Y):- Y == 4,
    X > 6,
    verifyWinDiagonal1(Board, X, Y).
%linha5
verifyWinDiagonals(Board,X,Y):- Y == 5,
    X < 4,
    verifyWinDiagonal2(Board, X, Y).
%linha5
verifyWinDiagonals(Board,X,Y):- Y == 5,
    X < 4,
    verifyWinDiagonal3(Board, X, Y).
%linha5
verifyWinDiagonals(Board,X,Y):- Y == 5,
    X == 4,
    verifyWinDiagonal1(Board, X, Y).
```

```
%linha5
verifyWinDiagonals(Board,X,Y):- Y == 5,
        X == 4,
        verifyWinDiagonal2(Board, X, Y).
%linha5
verifyWinDiagonals(Board,X,Y):- Y == 5,
        X == 4,
        verifyWinDiagonal3(Board, X, Y).
%linha5
verifyWinDiagonals(Board,X,Y):- Y == 5,
        X == 5,
        verifyWinDiagonal1(Board, X, Y).
%linha5
verifyWinDiagonals(Board,X,Y):- Y == 5,
        X == 5,
        verifyWinDiagonal2(Board, X, Y).
%linha5
verifyWinDiagonals(Board,X,Y):- Y == 5,
        X == 5,
        verifyWinDiagonal3(Board, X, Y).
%linha5
verifyWinDiagonals(Board,X,Y):- Y == 5,
    X == 5,

    verifyWinDiagonal4(Board, X, Y).

%linha5

verifyWinDiagonals(Board,X,Y):- Y == 5,

    X == 6,

    verifyWinDiagonal1(Board, X, Y).

%linha5

verifyWinDiagonals(Board,X,Y):- Y == 5,

    X == 6,

    verifyWinDiagonal3(Board, X, Y).

%linha5

verifyWinDiagonals(Board,X,Y):- Y == 5,

    X == 6,

    verifyWinDiagonal4(Board, X, Y).

%linha5

verifyWinDiagonals(Board,X,Y):- Y == 5,

    X > 6,

    verifyWinDiagonal1(Board, X, Y).

%linha5

verifyWinDiagonals(Board,X,Y):- Y == 5,

    X > 6,

    verifyWinDiagonal4(Board, X, Y).
```

```
%linha6
verifyWinDiagonals(Board,X,Y):- Y == 6,
    X < 5,
    verifyWinDiagonal2(Board, X, Y).
%linha6
verifyWinDiagonals(Board,X,Y):- Y == 6,
    X < 5,
    verifyWinDiagonal3(Board, X, Y).
%linha6
verifyWinDiagonals(Board,X,Y):- Y == 6,
    X > 4,
    X < 7,
    verifyWinDiagonal1(Board, X, Y).
%linha6
verifyWinDiagonals(Board,X,Y):- Y == 6,
    X > 4,
    X < 7,
    verifyWinDiagonal2(Board, X, Y).
%linha6
verifyWinDiagonals(Board,X,Y):- Y == 6,
    X > 4,
    X < 7,
        verifyWinDiagonal3(Board, X, Y).
%linha6
verifyWinDiagonals(Board,X,Y):- Y == 6,
        X > 4,
        X < 7,
        verifyWinDiagonal4(Board, X, Y).
%linha6
verifyWinDiagonals(Board,X,Y):- Y == 6,
        X > 6,
        verifyWinDiagonal1(Board, X, Y).
%linha6
verifyWinDiagonals(Board,X,Y):- Y == 6,
        X > 6,
        verifyWinDiagonal4(Board, X, Y).


/************************************************************/
/******************* parte de baixo do tabuleiro*********************/


%linha10 e 11
verifyWinDiagonals(Board,X,Y):- Y > 9,
        Y < 12,
        X > 0,
	verifyWinDiagonal44(Board, X, Y).
%linha10 e 11
verifyWinDiagonals(Board,X,Y):- Y > 9,
	Y < 12,
	X > 0,
	verifyWinDiagonal22(Board, X, Y).
%linha8
verifyWinDiagonals(Board,X,Y):- Y == 8,
	X < 3,
	verifyWinDiagonal22(Board, X, Y).
%linha8
verifyWinDiagonals(Board,X,Y):- Y == 8,
	X > 2,
	X < 7,
	verifyWinDiagonal22(Board, X, Y).
%linha8
verifyWinDiagonals(Board,X,Y):- Y == 8,
	X > 2,
	X < 7,
	verifyWinDiagonal44(Board, X, Y).
%linha8
verifyWinDiagonals(Board,X,Y):- Y == 8,
        X > 6,
        verifyWinDiagonal44(Board, X, Y).
%linha7
verifyWinDiagonals(Board,X,Y):- Y == 7,
        X < 4,
        verifyWinDiagonal22(Board, X, Y).
%linha7
verifyWinDiagonals(Board,X,Y):- Y == 7,
        X < 4,
        verifyWinDiagonal33(Board, X, Y).
%linha7
verifyWinDiagonals(Board,X,Y):- Y == 7,
        X > 3,
        X < 6,
        verifyWinDiagonal11(Board, X, Y).
%linha7
verifyWinDiagonals(Board,X,Y):- Y == 7,
        X > 3,
        X < 6,
        verifyWinDiagonal22(Board, X, Y).
%linha7
verifyWinDiagonals(Board,X,Y):- Y == 7,
    X > 3,
    X < 6,
    verifyWinDiagonal33(Board, X, Y).
%linha7
verifyWinDiagonals(Board,X,Y):- Y == 7,
    X > 3,
    X < 6,
    verifyWinDiagonal44(Board, X, Y).
%linha7
verifyWinDiagonals(Board,X,Y):- Y == 7,
    X == 6,
    verifyWinDiagonal11(Board, X, Y).
%linha7
verifyWinDiagonals(Board,X,Y):- Y == 7,
    X == 6,
    verifyWinDiagonal22(Board, X, Y).
%linha7
verifyWinDiagonals(Board,X,Y):- Y == 7,
    X == 6,
    verifyWinDiagonal44(Board, X, Y).
%linha7
verifyWinDiagonals(Board,X,Y):- Y == 7,
	X > 6,

	verifyWinDiagonal11(Board, X, Y).

%linha7

verifyWinDiagonals(Board,X,Y):- Y == 7,

	X > 6,

	verifyWinDiagonal44(Board, X, Y).



/*********************************************************************
****************/


%linha1

verifyWinHorizontal(Board,X,Y):- Y == 1,

	X == 1,

	verifyWinHorizontal1(Board,X,Y).

%linha1

verifyWinHorizontal(Board,X,Y):- Y == 1,

	X == 5,

	verifyWinHorizontal2(Board,X,Y).

%linha2

verifyWinHorizontal(Board,X,Y):- Y == 2,

	X < 3,
	verifyWinHorizontal1(Board,X,Y).
%linha2
verifyWinHorizontal(Board,X,Y):- Y == 2,
	X > 4,
	verifyWinHorizontal2(Board,X,Y).
%linha3
verifyWinHorizontal(Board,X,Y):- Y == 3,
	X < 3,
	verifyWinHorizontal1(Board,X,Y).
%linha3
verifyWinHorizontal(Board,X,Y):- Y == 3,
	X > 4,
	verifyWinHorizontal2(Board,X,Y).
%linha4
verifyWinHorizontal(Board,X,Y):- Y == 4,
	X < 5,
	verifyWinHorizontal1(Board,X,Y).
%linha4
verifyWinHorizontal(Board,X,Y):- Y == 4,
	X > 4,
	verifyWinHorizontal2(Board,X,Y).
%linha5
```

```
verifyWinHorizontal(Board,X,Y):- Y == 5,
        X < 6,
        verifyWinHorizontal1(Board,X,Y).
%linha5
verifyWinHorizontal(Board,X,Y):- Y == 5,
        X > 4,
        verifyWinHorizontal2(Board,X,Y).
%linha6
verifyWinHorizontal(Board,X,Y):- Y == 6,
        X < 7,
        verifyWinHorizontal1(Board,X,Y).
%linha6
verifyWinHorizontal(Board,X,Y):- Y == 6,
        X > 4,
        verifyWinHorizontal2(Board,X,Y).
%linha7
verifyWinHorizontal(Board,X,Y):- Y == 7,
        X < 6,
        verifyWinHorizontal1(Board,X,Y).
%linha7
verifyWinHorizontal(Board,X,Y):- Y == 7,
        X > 4,
        verifyWinHorizontal2(Board,X,Y).
%linha8
verifyWinHorizontal(Board,X,Y):- Y == 8,
        X < 5,
        verifyWinHorizontal1(Board,X,Y).
%linha8
verifyWinHorizontal(Board,X,Y):- Y == 8,
        X > 4,
        verifyWinHorizontal2(Board,X,Y).
%linha9
verifyWinHorizontal(Board,X,Y):- Y == 9,
        X < 4,
        verifyWinHorizontal1(Board,X,Y).
%linha9
verifyWinHorizontal(Board,X,Y):- Y == 9,
        X > 4,
        verifyWinHorizontal2(Board,X,Y).
%linha10
verifyWinHorizontal(Board,X,Y):- Y == 10,
        X < 3,
        verifyWinHorizontal1(Board,X,Y).
%linha10
```

```
verifyWinHorizontal(Board,X,Y):- Y == 10,

        X > 4,

        verifyWinHorizontal2(Board,X,Y).

%linha11

verifyWinHorizontal(Board,X,Y):- Y == 11,

        X == 1,

        verifyWinHorizontal1(Board,X,Y).

%linha11

verifyWinHorizontal(Board,X,Y):- Y == 11,

        X == 5,

        verifyWinHorizontal2(Board,X,Y).
```