



Echek

Relatório Intercalar

Programação em Lógica

Turma 5 - Grupo **Echek_1**:

Carlos Jorge Direito Albuquerque - **up201706735**@fe.up.pt

Tito Alexandre Trindade Griné - **up201706732**@fe.up.pt

Índice

Descrição do Jogo	3
História	3
Regras	3
Modelação do Jogo	7
Representação Interna do Jogo	7
Visualização do Tabuleiro na Consola	9

Descrição do Jogo

História

Echek é um jogo criado por Léandre Proust, um autor de jogos de tabuleiro *Free-to-Play* natural de França. O criador conta já com 9 jogos originais, sendo Echek o primeiro e também o mais velho - foi criado em Fevereiro de 2019.

Segundo Léandre, “a ideia [do Echek] é reproduzir a experiência de um jogo de xadrez apenas com 12 cartaz e sem tabuleiro”.

Regras

Início: Cada jogador fica com as 6 peças de uma cor. Os Reis são colocados frente a frente na zona de jogo, ficando as restantes peças na reserva de cada jogador. Começa a jogar quem tiver as peças brancas.



Figura 1 - Disposição Original do Jogo

Objetivo: Movimentar as peças de modo a que o Rei do adversário fique rodeado, isto é, ter uma peça, do jogador ou do adversário, ou um limite nas quatro casas (Norte, Sul, Este e Oeste) adjacentes ao Rei.

Desenvolvimento: Os jogadores jogam à vez, podendo em cada jogada realizar uma de duas possíveis ações:

- ♔ Jogar uma peça: O jogador escolhe uma peça da sua reserva e coloca-a em jogo. A peça tem de ser posicionada por forma a ficar ligada por um canto ou por um lado a uma outra peça do jogador (mesma cor); não é possível jogar uma peça diretamente numa das 4 casas adjacentes ao Rei Inimigo.
- ♔ Mover uma peça: O jogador escolhe uma das suas peças em jogo e movimenta-a respeitando as regras do movimento de cada peça; após a jogada, todas as peças em

jogo devem estar conectadas umas às outras por um lado ou por um canto, caso contrário a jogada não é válida.

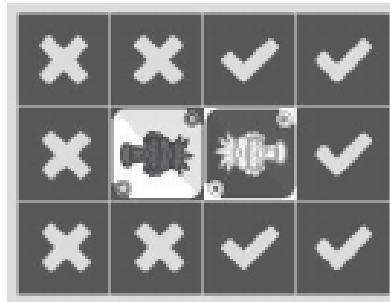


Figura 2 - Possíveis casas de colocação para a primeira peça do jogador branco

Não existe captura de peças, isto é, uma peça não pode ser movimentada para a casa de uma outra peça.

O espaço de jogo está limitado a 4 casas na horizontal e 4 na vertical, as peças não podem ser movimentadas nem jogados para lá desse limite. Os limites só existem quando as peças em jogo assim o forçam, ou seja, quando quatro casas numa direção contêm peças.

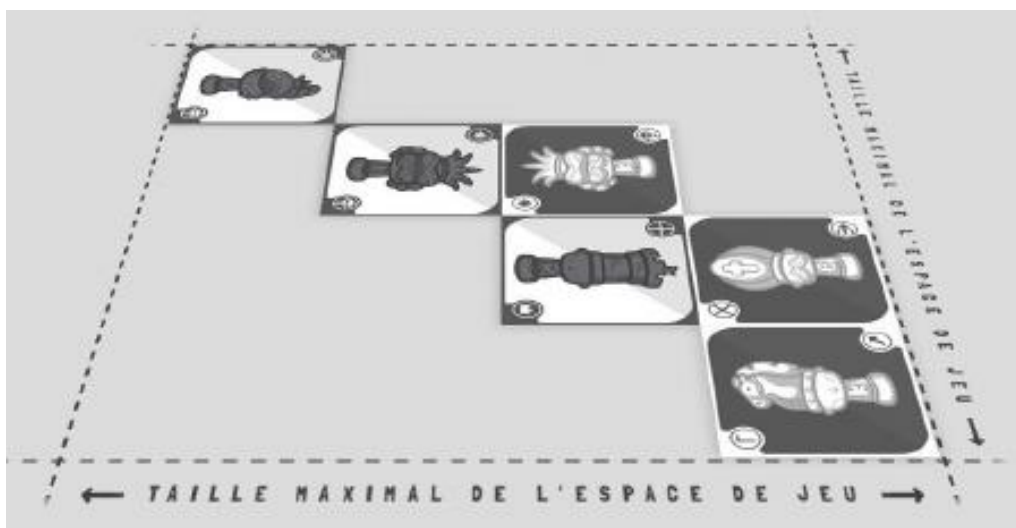


Figura 3 - Exemplo da criação de limites do tabuleiro

Mecânica das Peças: As peças usadas são as mesmas do xadrez, contudo, cada jogador apenas tem uma peça de cada tipo. Existem, portanto, as seguintes peças:

- ♔ Rei: Movimenta-se uma casa em linha reta ou na diagonal. Se for rodeado, o jogador perde o jogo.
- ♚ Rainha: Pode movimentar-se um número ilimitado de casas em linha reta ou na diagonal. Se for rodeada, é retirada do tabuleiro e não pode ser colocada em jogo outra vez.
- ♝ Bispo: Pode movimentar-se um número ilimitado de casa na diagonal. Quando é colocada em jogo, o jogador escolhe uma peça do tabuleiro (excluindo os reis) para ser colocada na reserva do jogador cuja peça pertence.
- ♖ Torre: Pode movimentar-se um número ilimitado de casas em linha reta. Apenas uma vez por jogo, se a peça estiver no tabuleiro, o jogador pode optar por não mover nenhuma peça, e trocar a Torre e o Rei de posição.
- ♗ Cavalo: Movimenta-se em L, ou seja, uma casa em linha reta e outra na diagonal. Pode passar por cima tanto de peças aliadas como inimigas.
- ♙ Peão: Movimenta-se uma casa em linha reta. Quando colocada em jogo, pode movimentar-se ainda na mesma jogada.

As peças que se movimentam um número ilimitado de casas (rainha, torre e bispo) não podem passar por casas contendo peças da cor oposta, mas é possível se as peças forem da mesma cor.

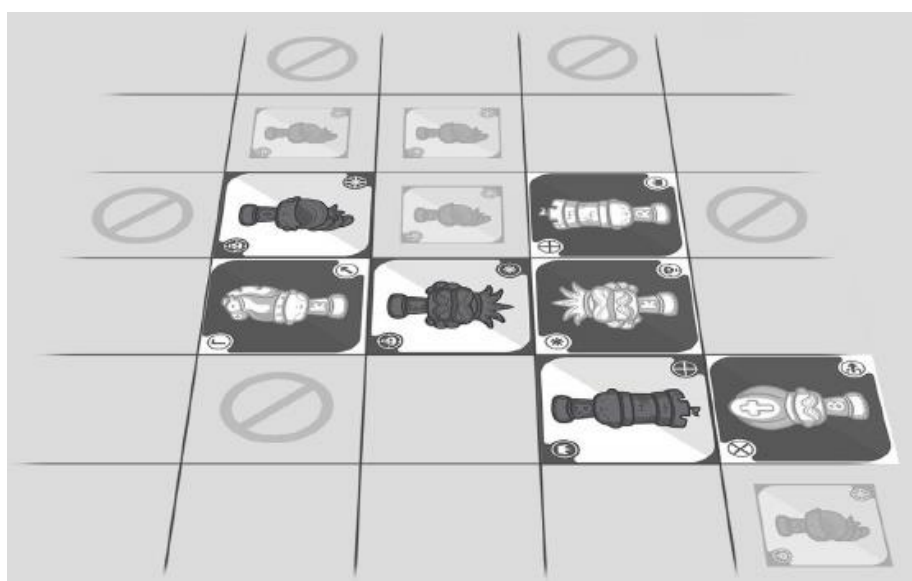


Figura 4 - Exemplo dos possíveis movimentos da rainha preta

Final: O jogo termina quando um dos jogadores cumpre o objetivo, isto é, rodeia o Rei do adversário. Não existe nenhuma outra condição de terminação do jogo, ou seja, até um dos Reis ser rodeado, é sempre possível realizar uma jogada.

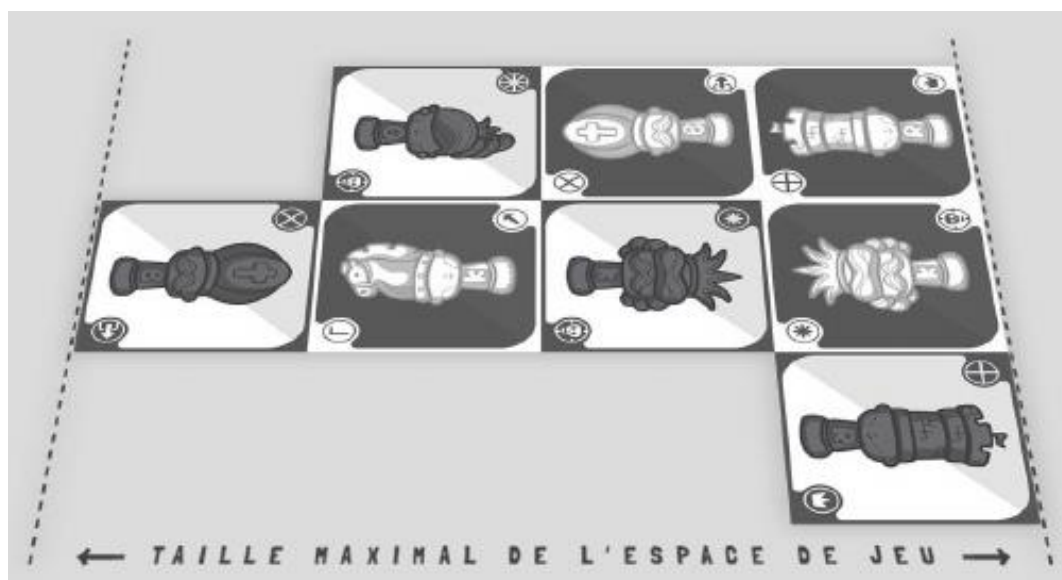


Figura 5 - Exemplo de uma derrota do jogador branco

Fontes:

<https://en.tipeee.com/leandreproust> (Site Oficial do Autor, consultado a 14/10)

<http://www.fabrikludik.fr/jeuxGratuit/echekRegle.pdf> (Livro de Regras do Jogo, consultado a 14/10)

Modelação do Jogo

Representação Interna do Jogo

Uma vez que o jogo difere um bocado dos jogos tradicionais de tabuleiro, visto que não tem um tabuleiro estático, mas sim um relativo, isto é, está dependente da posição das peças em jogo para definir os possíveis limites, a abordagem normal de mapear o tabuleiro numa lista de listas pareceu-nos inadequada ou pelo menos, pouco prática. O facto de ser um tabuleiro dinâmico não é trivial, e considerámos que forçar o tabuleiro a ser estático tiraria uma mecânica do jogo que acrescenta várias possibilidades para estratégias interessantes, o que torna, em último caso, o jogo mais divertido.

Assim, primeiro decidimos representar internamente o jogo através de um conjunto de predicados que ditam as peças e respetivas posições que se encontram em jogo. O jogador vê apenas uma grelha 6x6, que auxilia apenas para poder indicar movimentos. Na realidade, internamente, pretendemos que o jogo desenrole até uma das peças sair do quadrado 4x4 no interior do tabuleiro, situação em que as peças são ajustadas por forma a recolocá-las de novo no quadrado central (fazendo *shift* de todas as peças uma casa na direção necessária). Caso a disposição do jogo obrigue a limites, dado o reajuste que é feito, o limite estará sempre no quadrado 4x4 interior. Isto assegura também que nunca há um movimento válido fora do quadrado 6x6, já que as peças têm de estar sempre ligadas umas às outras.

```
% Pieces
piece(king, black, '♔').
piece(king, white, '♚').
piece(queen, black, '♕').
piece(queen, white, '♛').
piece(bishop, black, '♗').
piece(bishop, white, '♝').
piece(rook, black, '♖').
piece(rook, white, '♜').
piece(knight, black, '♞').
piece(knight, white, '♟').
piece(pawn, black, '♟').
piece(pawn, white, '♙').
```

Figura 6 - Representação das Peças

```
cell(3, 2, white, king).
cell(2, 2, black, king).
```

Figura 7 - Representação do Tabuleiro Inicial
(as casas em branco não têm predicado)

```
cell(1, 2, white, king).
cell(3, 3, black, king).
cell(3, 4, white, pawn).
cell(3, 2, black, pawn).
cell(2, 1, white, knight).
cell(4, 2, black, knight).
cell(2, 3, white, queen).
cell(4, 3, white, bishop).
cell(3, 1, black, bishop).
cell(1, 3, white, rook).
cell(4, 4, black, rook).
```

Figura 8 - Representação do tabuleiro a meio de um jogo

```
cell(1, 2, white, king).
cell(3, 3, black, king).
cell(3, 4, white, pawn).
cell(2, 2, black, pawn).
cell(2, 1, white, knight).
cell(4, 2, black, knight).
cell(2, 3, white, queen).
cell(1, 3, black, queen).
cell(3, 2, white, bishop).
cell(3, 1, black, bishop).
cell(1, 4, white, rook).
cell(4, 4, black, rook).
```

Figura 9 - Representação do tabuleiro no final do jogo (Rei preto rodeado)

```
display_board :-
    nl, draw_space, draw_title, nl,
    draw_space, draw_top_boundary,
    draw_space, draw_column_coordinates,
    \+display_board([6, 6], 0),
    draw_space, draw_bottom_boundary, nl.

display_board([R, C], X) :-
    X < R,
    draw_space, draw_separator(C),
    draw_space, draw_line_coordinate(X),
    \+display_row([X, C], 0), nl,
    N is X + 1, display_board([R, C], N).

display_row([R, C], X) :-
    X < C, display_piece(R, X),
    N is X + 1, display_row([R, C], N).

display_piece(R, C) :-
    cell(C, R, Color, Piece),
    piece(Piece, Color, Char),
    opposite(Color, Opposite),
    draw_piece(Char, Color, Opposite).

display_piece(R, C) :-
    \+cell(C, R, Color, Piece),
    draw_blank.
```

Figura 10 - Funções principais responsáveis pelo display do tabuleiro. Para visualizar é preciso chamar '?- display_board.'

Nota: Outras funções também são utilizadas para o display do tabuleiro, só para manter a aparência consistente, pelo que envia-mos esse código em anexo

Visualização do Tabuleiro na Consola

	♠ ECHEK ♠					
	0	1	2	3	4	5
0						
1						
2			♠	♠		
3						
4						
5						

Figura 11 - Configuração Inicial

	♠ ECHEK ♠					
	0	1	2	3	4	5
0						
1				♠		
2			♠	♠		
3			♠	♠	♠	
4						
5						

Figura 12 - Configuração Intermédia 1

	♠ ECHEK ♠					
	0	1	2	3	4	5
0						
1			♠	♠		
2		♠	♠	♠	♠	
3		♠	♠	♠		
4		♠		♠	♠	
5						

Figura 13 - Configuração Intermédia 2

	♠ ECHEK ♠					
	0	1	2	3	4	5
0						
1			♠	♠		
2		♠		♠	♠	
3		♠	♠	♠	♠	
4				♠	♠	
5						

Figura 14 - Configuração Final
(Rei Preto Rodeado)

Nota: De modo a ser possível visualizar as cores e os caracteres de Unicode utilizados, recorreu-se a [SWI](#) em vez de o Sictus como sistema de desenvolvimento para Prolog.