

Programação em Lógica

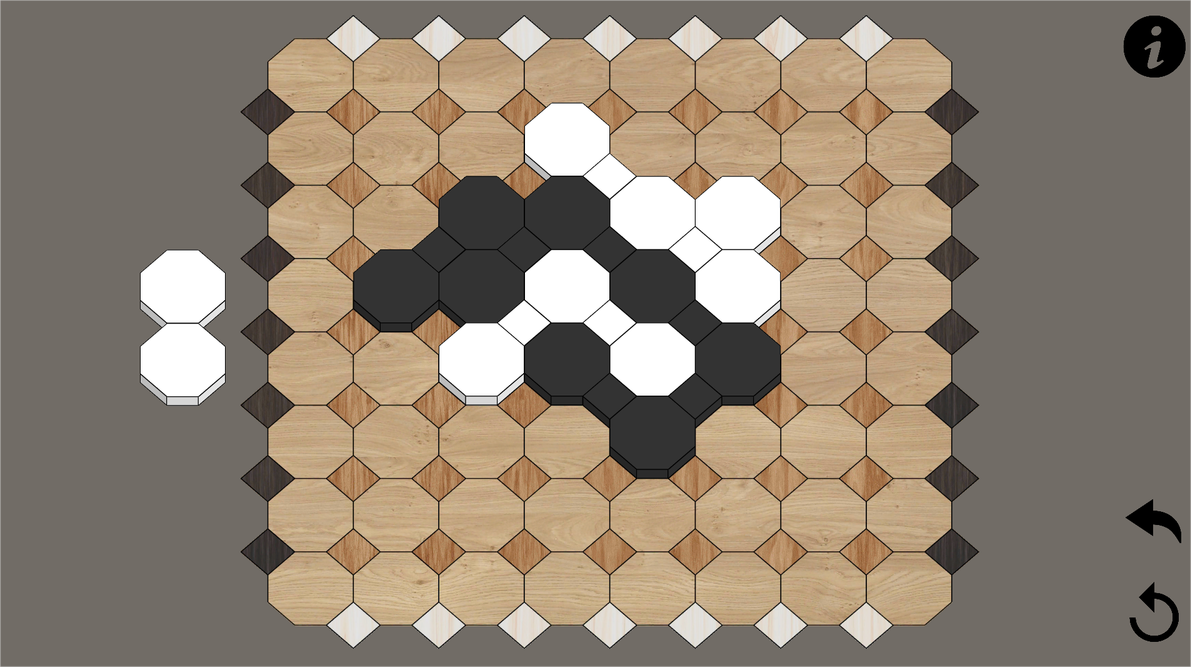
*Squex*

**Descrição:**

***Squex*** é um jogo criado em 2019, de conexão abstrata para 2 jogadores, semelhante ao ***Hex***.

O jogo é realizado num tabuleiro 8x8 de octógonos e quadrados.

O objetivo de cada jogador é conectar os lados correspondentes do tabuleiro com uma linha contínua de peças enquanto impede o seu oponente de fazer a sua conexão.

****

**Fig1. Tabuleiro de Squex nas jogadas iniciais**

**Regras:**

1. Na sua vez, cada jogador pode colocar uma peça octogonal em qualquer espaço octogonal do tabuleiro, desde que esteja livre.
2. Se uma peça é colocada na diagonal de uma peça da mesma cor, um quadrado dessa cor é colocado para conectá-las (é possível colocar até 4 quadrados colocando apenas uma peça).
3. Se uma peça for colocada na diagonal de uma peça da mesma cor e houver um quadrado da cor do oponente entre eles, o quadrado do oponente será substituído por um quadrado da cor do jogador. Os octógonos do oponente já não estão conectados pelo quadrado que lá estava. A isso chama-se “corte”.
4. Sempre que um jogador faz um "corte", o oponente pode jogar duas vezes seguidas (se um jogador fizer um "corte" na primeira jogada este perderá a sua segunda jogada e será a vez do oponente de jogar duas vezes consecutivas).
5. Um jogador só vence quando este conectar os seus dois lados do tabuleiro e for impossível para o seu oponente quebrar a ligação com um "corte" na jogada imediatamente seguinte.



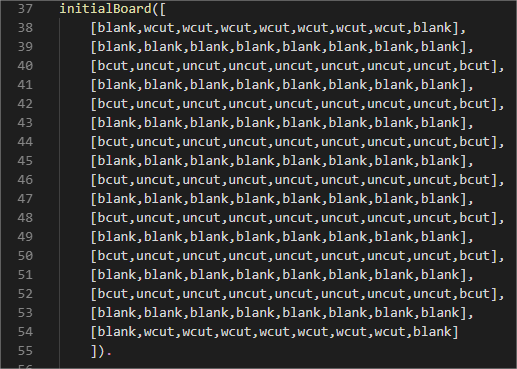
Fig2. Tabuleiro de Squex no final do jogo

(Preto ganha pois conecta de um lado ao outro sem ser possível ao branco fazer “cut”).

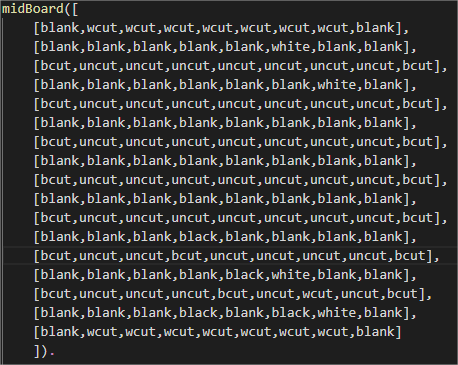
ABORDAGEM INICIAL

Representação interna do estado do jogo

Para representar o tabuleiro, decidimos criar uma lista de listas, em que as listas pares representam as linhas de quadrados (linhas de “cut”), e as listas ímpares representam linhas de peças (ou linhas de jogo).

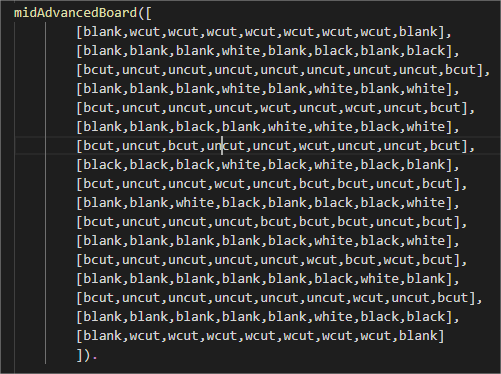


O nosso tabuleiro inicial é representado com todas as linhas pares vazias(“uncut”), exceto as das bordas. Todas as linhas ímpares estão também vazias inicialmente (“empty”).

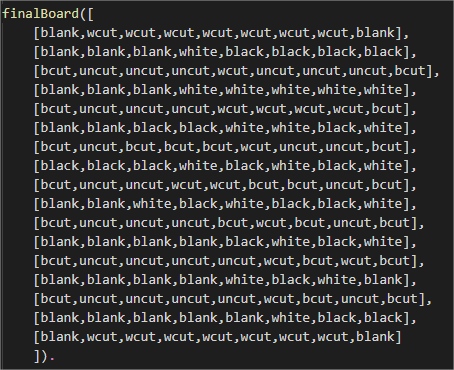


O tabuleiro intermédio inicial é representado como tendo já algumas peças colocadas e alguns “cut” realizados, mas ainda com ambos os jogadores longe de alcançar a vitória.

(wcut representa cut de peça branca, bcut representa cut de peça preta)

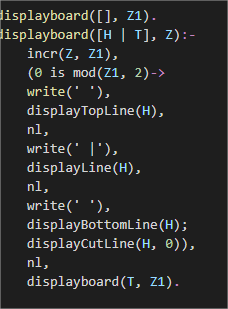
O tabuleiro intermédio avançado é representado como tendo já bastantes peças colocadas e “cut” realizados, com ambos os jogadores perto de ganhar.

(wcut representa cut de peça branca, bcut representa cut de peça preta)

O tabuleiro final é representado com um dos jogadores a ter um caminho que une os lados do tabuleiro, e sem que o adversário possa realizar um corte.

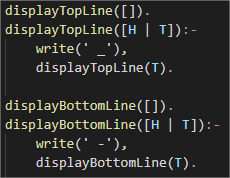
Visualização do tabuleiro

Para a visualização em modo de texto, criamos várias funções, que vamos passar a descrever, por níveis:



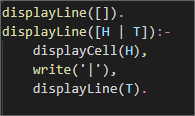
displayBoard()

Esta função recebe um board e um número de linha inicial(0). recursivamente, vai fazendo display a cada linha, sendo que, se a linha for par(linha de peças), imprime um top e um bottom, para facilitar a visualização, criando uma espécie de caixa para cada célula. Se a linha for ímpar, imprime a linha de cut, chamando a função respectiva com a cabeça da lista e o primeiro elemento (que será usado para “indentar” a linha, como veremos melhor mais à frente.

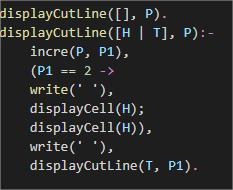


displayTopLine & displayBottomLine

Estas funções servem apenas para facilitar a perceção do tabuleiro ao jogador, uma vez que “dividem” as células com underscore e hífen, para representar o topo e fundo de uma célula, respetivamente.

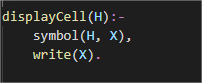
displayLine()

Esta função percorre a lista, chamando a função displayCell com o elemento da cabeça da lista, e imprimindo um caracter “|” para facilitar a visualização.

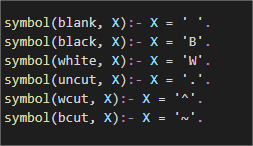


displayCutLine ()

Esta função serve para imprimir as linhas pares, ou seja, as linhas de corte. Fá-lo incrementando uma variável P, que quando for igual a 2, será imprimido um espaço antes da célula, facilitando a visualização e aproximando-a do jogo original, uma vez que neste os elementos da linha de corte situam-se entre as células das linhas de peça.



displayCell()

Esta função recebe um elemento e tradu-lo através de uma função, fazendo depois “write” do resultado da tradução.

symbol()

Esta função faz a tradução de cada elemento possível do tabuleiro para um símbolo que será visualizado, sendo que “blank” é vazio, “black” é preto, “white” é branco, “uncut” significa que não houve corte, “wcut” houve corte por peças brancas e “bcut” que houve corte por parte de peças pretas.

Exemplos de visualização de tabuleiros

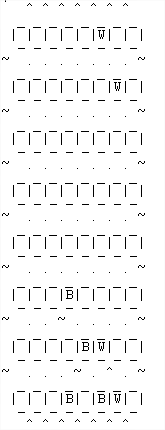
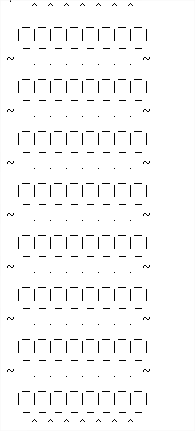


Fig.3 Tabuleiro Inicial Fig4. Tabuleiro intermédio inicial

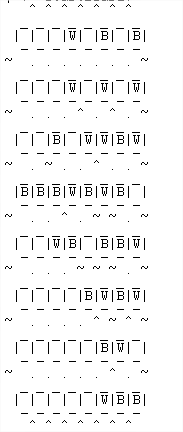
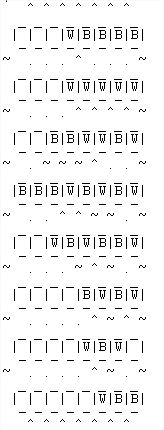


Fig 5. Tabuleiro avançado Fig6. Tabuleiro final

Bibliografia

As figuras 1 e 2 foram retiradas do seguinte URL:

<https://boardgamegeek.com/image/4726662/squex>

As restantes foram obtidas de código desenvolvido pelo grupo, e as figuras 3, 4, 5 e 6 são resultantes da execução desse mesmo código no software SICStus Prolog 4.5.1