

Blackstone

Grupo: Blackstone_2

Membros

Gonalo Miguel Dias Ferros - up202207592 -> 50%

- Design do tabuleiro
- Menu
- Loops principais do jogo

Miguel Moita Caseira - up202207678 -> 50%

- Modos com computador
- Menu
- Variantes Medium Churn e High Churn

Instalao e Execuo

 necessrio ter apenas o SICStus instalado na mquina. Nas imagens deste ficheiro, foi utilizada a fonte *Cascadia Code Regular* para uma visualizao melhorada.

Passos

1. Abrir o SICStus.
2. Consultar `game.pl` no diretrio `src`:

```
?- consult('./game.pl').
```

3. Executar o predicado **play/0** sem argumentos para iniciar o jogo.

Descrio do Jogo

Blackstone  um jogo de estratgia e para **2 jogadores**, e foi desenvolvido por **Mark Steere** em maro de 2024 com a contribuio de Alex Erickson no seu design.

 jogado num **tabuleiro quadrado** de qualquer tamanho par (**6x6** ou maior). A borda do tabuleiro  povoada por **pedras vermelhas e azuis** em padres iniciais especficos, sendo controladas por jogadores diferentes.

O **objetivo**  **bloquear** ou **remover** as peas do oponente, at que no sobre nenhuma.

No **turno** de cada jogador, o mesmo deve escolher uma pedra da sua **cor** para se mover como uma **rainha no xadrez** (por qualquer nmero de casas ao longo de uma linha reta sem obstculos, em qualquer direo: horizontal, vertical e diagonal).

Aps a sua **jogada**, deve **colocar uma pedra preta** no local de onde a pedra que lhe pertencia se moveu. Esta pedra agora age como um **obstculo** para jogadas futuras.

Se, após colocar a pedra preta, **qualquer pedra vermelha ou azul estiver totalmente cercada por pedras de qualquer cor e não puder se mover**, essas pedras são **removidas do tabuleiro**.

Para chegar à **vitória**, um dos jogadores necessita de **eliminar todas as peças do oponente**, mesmo que nesse cenário acabe por **eliminar todas as suas peças restante restantes no tabuleiro**.

Lógica do Jogo

Ficheiro	Descrição
<code>board.pl</code>	Responsável por mostrar o tabuleiro
<code>game.pl</code>	Contém o predicado <code>play</code> e predicados para o loop e para as regras do jogo
<code>menu.pl</code>	Processa o input do utilizador no menu e guarda as opções selecionadas

Representação da Configuração

À medida que o utilizador navega no menu e seleciona as configurações do jogo, que são a variante do jogo, o modo, a dificuldade (para modos com computador), o tamanho do tabuleiro e a cor do primeiro jogador. Os argumentos correspondentes são passados a **create_game_config/5**, que guarda as informações, cria o estado inicial do jogo e passa-as a **initial_state/2**.

Interação com o Utilizador

Menu

Quando o utilizador acede ao menu, pode selecionar uma opção ao escrever o número correspondente à opção, tornando a escolha das configurações mais fácil e prática.

```
=====
      BLACKSTONE
=====
1. Play
2. Controls
3. How to Play
4. Exit
Choose an option (1-4): 1.

Select Game Variant:
1. Base Blackstone
2. Medium Churn Variant
3. High Churn Variant
4. Back
```

Menu Inicial

Execução de Movimentos

Relativamente aos movimentos, o jogador tem de escrever uma lista com as coordenadas de origem e destino da peça, como mostra a imagem:

	1	2	3	4	5	6
1		r		r		
2						b
3	b					
4						b
5	b					
6			r		r	

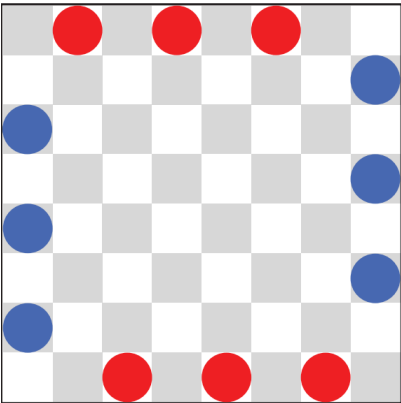
Player r, make your move!
Enter [FromRow, FromCol, ToRow, ToCol]: |: [1,2,2,2].

Movimento

Em ambas as situações, o utilizador dispõe de uma dica de ajuda caso tenha dificuldade em interagir com o jogo.

Representação Interna do Estado

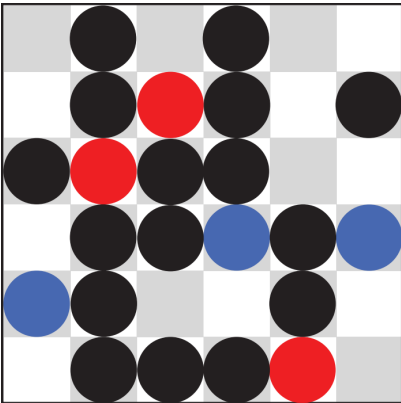
O estado do jogo guarda informações sobre as peças do tabuleiro, o próximo jogador a jogar e o tamanho do tabuleiro. A visualização do estado é tratada pelo predicado **display_game/5**.



Estado Inicial

	1	2	3	4	5	6
1		r		r		
2						b
3	b					
4						b
5	b					
6			r		r	

Estado Inicial Real



Estado Intermédio

	1	2	3	4	5	6
1		x	x	x		
2			r	x	b	x
3	x		x	r		
4			b	x		b
5	x			x		
6	b		x	r	r	

Estado Intermédio Real

Representação de Movimentos

Caso o jogador seja uma pessoa, o predicado **move/8** é chamado quando o jogador executa um movimento. O predicado recebe qual o jogador, posições de origem e destino da peça e o tabuleiro.

Caso o jogador seja um computador, é chamado o predicado **computer_move/5**, que recebe também a dificuldade do jogo, de modo a decidir se o movimento é aleatório ou com a utilização de um algoritmo greedy, chamando **random_computer_move/4** ou **greedy_computer_move/4**, respetivamente.

Fim do Jogo

Sempre que um jogador faz um movimento, são verificados os movimentos possíveis para cada jogador. Caso um jogador não tenha movimentos possíveis, é declarada a vitória do oponente. Caso nenhum dos jogadores tenha movimentos possíveis, trata-se de um empate.

Conclusões

No decorrer do projeto, pudemos explorar uma linguagem muito diferente das que estamos habituados, o que nos levou a pensar de uma maneira diferente.

As principais limitações do jogo estão relacionadas com predicados pouco otimizados, onde foi dada prioridade à funcionalidade.

Poderiam ser aplicadas algumas melhorias, nomeadamente a otimização de predicados já existentes.

Bibliografia

- [Regras do jogo](#)

Utilizámos o ChatGPT durante o desenvolvimento. A sua utilização foi principalmente para debugging e resolução de problemas no código.