Programação Funcional e em Lógica

Projeto 2 - Storm Clouds

Grupo: StormClouds_9

Membros:
Lucas Greco (50%)
João Pedro (50%)

Contribuição
Lucas Greco : Movimentação, Tabuleiro, peças, loops principais.
João Pedro : Bots, Apresentação do tabuleiro, melhorias visuais.

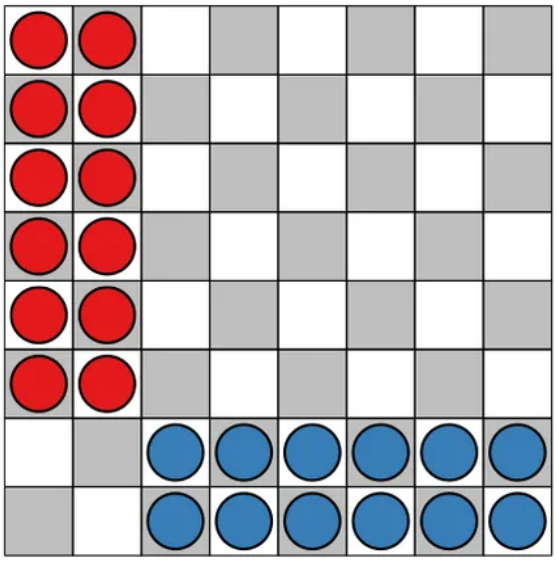
Execução
Compilar o arquivo game.pl e escrever play. (Abrirá o menu do jogo)

Descrição
- O tabuleiro é fixo sendo 8x8..
- São divididos em peças brancas e pretas. As brancas começam na vertical e as pretas na horizontal.

- O objetivo do jogo é capturar todas as peças do inimigo, não possui empates, quando não é possível fazer uma jogada válida a vez é passada para o próximo jogador, não sendo possível passar a vez de qualquer maneira.

- Posição inicial do tabuleiro, (nesse exemplo tirado do site oficial do jogo à direita, as peças brancas são vermelhas e as pretas são azuis):

```
    0   1   2   3   4   5   6   7
  +---+---+---+---+---+---+---+---+
0 | W | W |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
1 | W | W |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
2 | W | W |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
3 | W | W |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
4 | W | W |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
5 | W | W |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
6 |   |   | B | B | B | B | B | B |
  +---+---+---+---+---+---+---+---+
7 |   |   | B | B | B | B | B | B |
  +---+---+---+---+---+---+---+---+
```

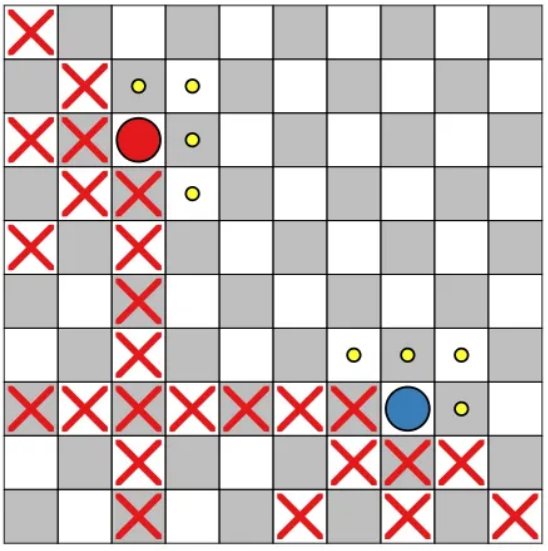


Movimento:

- Não capturante, só pode se mover se o espaço for livre (sem peça aliada ou inimiga), apenas uma casa por vez.
    - Branco:
        - Para cima.
        - Para diagonal de cima direita.
        - Para direita.
        - Para diagonal de baixo direita.
    - Preto:
        - Para diagonal de cima esquerda.
        - Para cima.
        - Para diagonal de cima direita.
        - Para direita.
- Capturante, pode se mover apenas para a primeira peça inimiga na direção escolhida.
    - Branco :
        - Para diagonal de cima esquerda.
        - Para esquerda.
        - Para diagonal de baixo esquerda.
        - Para baixo.
    - Preto :
        - Para esquerda.
        - Para diagonal de baixo esquerda.
        - Para baixo .
        - Para diagonal de baixo direita.

Movimento imagem:

- Descrição da imagem:
    - Peça Branca (Círculo vermelho).
    - Peça Preta (Círculo Azul).
    - Movimentos não capturantes (círculos pequenos amarelos)
    - Movimentos capturantes (“X”s vermelhos)



Considerações
Como a movimentação é complexa, no modo jogador x jogador e jogador x Bot mostramos as possíveis jogadas para facilitar.

Lógica do jogo
Dividimos o jogo em 3 loops principais que são escolhidos no menu principal do jogo:

O predicado **game_loop((Board, CurrentPlayer))** representa o ciclo principal do jogo no modo jogador x jogador.

- Board: Estado atual do tabuleiro.
- CurrentPlayer: Jogador atual (aquele que fará a jogada).
- Fluxo:
    - Verifica se o jogo acabou:
        - Se sim, termina o loop.
        - Se não, continua.
    - Exibe o estado atual do jogo e o jogador atual.
    - Jogador atual faz a jogada:
        - Escolhe a peça a mover e sua nova posição.
        - O tabuleiro é atualizado.
    - Troca para o próximo jogador.
    - Repete o loop até que o jogo termine.

O predicado **game_loop_against_bot((Board, CurrentPlayer), Difficulty)** define o ciclo principal de um jogo onde um jogador humano joga contra um bot. O loop alterna entre os dois jogadores, levando em consideração a dificuldade ao determinar as jogadas do bot.

- Board: Representa o estado atual do tabuleiro.
- CurrentPlayer: Jogador atual (pode ser player1 para humano ou player2 para o bot).
- Difficulty: Define o nível de dificuldade para o bot.
- Fluxo:
    - Fim do jogo:
        - Verifica se o jogo acabou. Se sim, o loop termina.
    - Exibição do estado atual:
        - Mostra o tabuleiro e o jogador atual.
    - Jogada:
        - Se for o bot:
            - Calcula e realiza uma jogada (ou passa o turno se não houver jogadas válidas, a variável difficulty auxilia em saber qual algoritmo usar para o bot (1 aleatório, 2 greedy)).
        - Se for o humano:
            - Solicita um movimento.
            - Valida e atualiza o estado do jogo.
    - Alterna o turno:
        - O próximo jogador faz sua jogada.

        - Repete o loop até que o jogo termine.

O predicado **game_loop_bot_against_bot((Board, CurrentPlayer), Difficulty1, Difficulty2)** define o ciclo principal de um modo de jogo bot contra bot. A dificuldade dos bots é escolhida separadamente.

- Board: Representa o estado atual do tabuleiro.
- CurrentPlayer: Jogador atual (pode ser player1 para Bot1 ou player2 para o Bot2).
- Difficulty: Define o nível de dificuldade para o bot1.
- Difficulty1: Define o nível de dificuldade para o bot1.
- Fluxo:
    - Verifica se o jogo acabou.
    - Exibe o estado atual do tabuleiro e o jogador ativo.
    - Atualiza a variável temporária Difficulty, com a dificuldade do Bot atual (Dfficulty1 ou Difficulty2).
    - O bot realiza um movimento válido:
        - Atualiza o tabuleiro.
        - Alterna o jogador.
    - Se o bot não puder realizar movimentos válidos:
        - Alterna o jogador sem modificar o tabuleiro.
    - Repete o loop até que o jogo termine.

Implementação da Movimentação

Para implementar a movimentação, criamos uma função que verifica se uma peça escolhida pode se mover para qualquer posição no tabuleiro. Utilizamos funções auxiliares que determinam se um movimento específico é válido para aquela peça. Todos os movimentos possíveis são armazenados em uma lista, que é apresentada ao jogador para que ele possa escolher qual movimento deseja realizar.

Baseado nessa lógica, também definimos o comportamento dos bots:

- **Bot Nível 1:** Escolhe um movimento aleatório da lista de movimentos válidos.
- **Bot Nível 2:** Prioriza movimentos que capturam peças inimigas, se houver algum disponível. Caso contrário, escolhe um movimento aleatório.

Interação com o usuário

Para começar o jogo o usuário deve digitar “ **play. “** , após isso todos os comandos serão feitos com **números**, como escolher a jogada, a dificuldade dos bots, o modo de jogo.

```
| ?- play.

-------------------------
 Welcome to Storm Clouds!
-------------------------
1. Play
2. Rules
3. Exit
|: 1.

Choose game mode:
1. Player vs Player
2. Player vs Bot
3. Bot vs Bot
|: 1.
```

```
    0   1   2   3   4   5   6   7
  +---+---+---+---+---+---+---+---+
0 | W | W |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
1 | W | W |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
2 | W | W |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
3 | W | W |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
4 | W | W |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
5 | W | W |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
6 |   |   | B | B | B | B | B | B |
  +---+---+---+---+---+---+---+---+
7 |   |   | B | B | B | B | B | B |
  +---+---+---+---+---+---+---+---+
```

```
Current Player: White

Calculating all valid moves...

Valid Moves:
0: From (0, 5) to (1, 6)
1: From (1, 0) to (2, 0)
2: From (1, 0) to (2, 1)
3: From (1, 1) to (2, 0)
4: From (1, 1) to (2, 1)
5: From (1, 1) to (2, 2)
6: From (1, 2) to (2, 1)
7: From (1, 2) to (2, 2)
8: From (1, 2) to (2, 3)
9: From (1, 3) to (2, 2)
10: From (1, 3) to (2, 3)
11: From (1, 3) to (2, 4)
12: From (1, 4) to (2, 3)
13: From (1, 4) to (2, 4)
14: From (1, 4) to (2, 5)
15: From (1, 5) to (2, 4)
16: From (1, 5) to (2, 5)

Choose a move number (0-16):
|: 1.

Move from 1-0 to 2-0

Confirm? 1 - Yes; 0 - No
|:
```

Conclusão

Primeiro desenvolvemos o modo jogador x jogador, escolhendo manualmente a jogada, depois para fazermos o bot desenvolvemos o predicado que verifica os movimentos possíveis, que futuramente implementamos no modo jogador x jogador, para facilitar a escolha da jogada. Como a movimentação é muito específica, várias funções auxiliares foram necessárias.

Bibliografia

Utilizamos as imagens e as regras : Storm Clouds
Auxílio de Ia : Github Copilot