

PFL TRIKE

| Grupo | Nome | UP | Contribuição |
|---------|--|---------------|--------------|
| Trike 7 | Diogo Alexandre Figueiredo Gomes | [up201905991] | 50% |
| Trike 7 | Rafael Ferreira da Costa Silva Valqueresma | [up202104805] | 50% |

Instruções de Instalação do Jogo Trike

Para instalar o jogo Trike, siga as etapas abaixo:

1. Faça o download dos arquivos contidos no arquivo [PFL_TP1_T10_Trike_7.zip] e descompacte-os em um diretório de sua escolha.
 2. Navegue até a pasta "src" no diretório onde os arquivos foram descompactados.
 3. Abra o arquivo "trike.pl" usando o Sicstus Prolog 4.8. Isso pode ser feito através da linha de comando ou pela interface gráfica do Sicstus Prolog.
 4. Inicie o jogo executando o predicado "play/0" no Sicstus Prolog.
-

Descrição do Jogo

Trike é um jogo de estratégia abstrata e combinatória projetado para dois jogadores. A dinâmica do jogo gira em torno da criação de armadilhas, da desmontagem das armadilhas do adversário e da manobra de uma peça compartilhada, conhecida como peão neutro, com o objetivo de alcançar a vitória.

Regras Principais:

- **Tabuleiro:** O tabuleiro do jogo possui a forma de um triângulo equilátero.
- **Peças Utilizadas:** No Trike, são utilizados um peão neutro, bem como peças de cores preta e branca para representar os jogadores.
- **Fase Inicial:** No início da partida, o primeiro jogador posiciona uma peça em qualquer local do tabuleiro e coloca o peão neutro sobre ela. Neste momento, o segundo jogador tem a oportunidade única de trocar de lado (mudar de peça) em vez de efetuar um movimento convencional.

- **Movimento do Peão Neutro:** O peão neutro tem a capacidade de mover-se por qualquer número de posições vazias, seguindo uma trajetória em linha reta, em qualquer direção. No entanto, ele não pode mover-se através de posições ocupadas por outras peças.
- **Colocação de Peças:** Quando um jogador coloca uma peça em uma posição permitida, o peão neutro é transferido para essa peça, ocupando-a.
- **Término do Jogo:** O jogo encerra quando o peão neutro fica impossibilitado de realizar movimentos.
- **Contagem de Pontos:** Ao final da partida, cada jogador acumula um ponto por cada peça de sua cor que estiver adjacente ao peão neutro ou posicionada abaixo dele. O jogador com a maior pontuação é declarado o vencedor. Empates não são possíveis.

Nota Importante:

Essas são as regras básicas do Trike. Para informações detalhadas e estratégias avançadas, consulte o [website oficial do jogo](https://boardgamegeek.com/boardgame/307379/trike) (<https://boardgamegeek.com/boardgame/307379/trike>).

Estrutura do Jogo

O estado do jogo é representado por uma lista de três elementos: [Board, Player, NumberMove], onde o `Board` é o tabuleiro atual, `Player` é o jogador da próxima jogada e `NumberMove` é o número da jogada.

Visualização do Jogo

Antes de iniciar, o utilizador configura o jogo, escolhendo o modo e tamanho do tabuleiro. O jogador inicial é escolhido aleatoriamente. O tabuleiro é inicializado com `initial_state(+Size, -GameState)`.

Validação e Execução de Movimentos

O jogo funciona num ciclo contínuo até um jogador vencer. Os jogadores inserem coordenadas, que são validadas por `validate_move/2`. Um movimento é válido se as coordenadas estiverem vazias, numa direção reta em relação ao peão neutro e não estiverem obstruídas. O peão neutro é movido, e uma peça do jogador anterior é colocada na posição anterior do peão neutro.

Lista de Jogadas Válidas

O predicado `valid_moves(+GameState, +Player, -ListOfMoves)` guarda as coordenadas de movimentos válidos no estado atual do jogo usando `findall` e `validate_move`.

Fim do Jogo

`game_over(+GameState, -Winner)` verifica se o jogo terminou com o peão neutro preso ou com `find_who_wins/2` determina o vencedor e `show_winner/1` anuncia o vencedor e a pontuação.

Avaliação do Estado do Jogo

`value(+GameState, +Player, -Value)` avalia o tabuleiro com base no jogador, na posição do peão neutro e nas peças ao redor. As peças do jogador aumentam o `Value` em 1, e as do adversário diminuem em 1.

Jogadas do Computador

O computador pode jogar de forma aleatória (método `random`) ou com estratégia (método `greedy`). O método `greedy` usa `value/3` e `is_possible_win/6` para determinar a melhor jogada. Se não houver uma jogada vencedora, escolhe com base no `Value`, priorizando posições com maior `Value`. Se múltiplas posições tiverem o mesmo `Value`, escolhe uma aleatoriamente.

Conclusão

Durante o desenvolvimento do projeto, enfrentámos desafios significativos ao utilizar a linguagem Prolog num jogo relativamente complexo. O jogo em questão tem um tabuleiro triangular e regras que envolvem movimentos com peças hexagonais, o que exigiu um conhecimento profundo da mecânica do jogo.

Uma das principais dificuldades que enfrentámos foi a implementação do nível de movimento "greedy" para o computador. Dada a falta de familiaridade com o jogo, foi um desafio determinar as melhores estratégias para ganhar a partida.

No entanto, apesar das dificuldades encontradas, acreditamos que este projeto teve um papel fundamental no desenvolvimento das nossas competências de programação em Prolog. Além disso, trabalhar num grupo pequeno permitiu-nos assumir responsabilidades individuais de uma forma mais significativa.

Em suma, sentimos que este projeto representou uma oportunidade valiosa para aperfeiçoar as nossas competências técnicas, de colaboração e de resolução de problemas, enquanto explorávamos as complexidades da programação em Prolog.