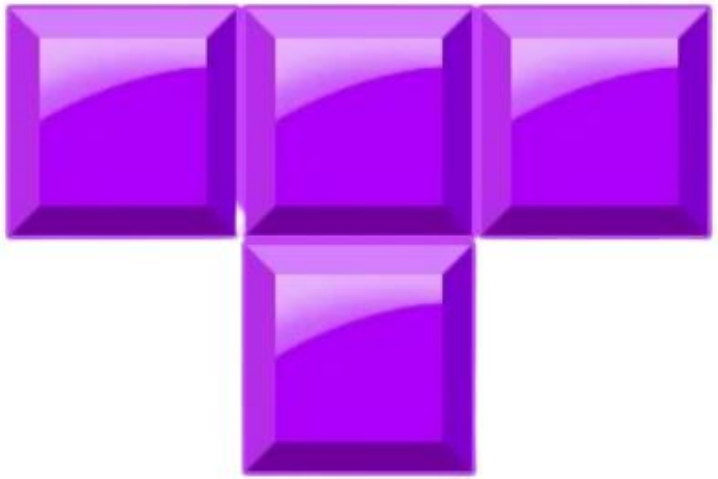
Desenvolvimento de um agente autónomo para o jogo Tetris

Relatório



Trabalho prático
Inteligência Artificial
2021/2022

Diogo Torrinhas 98440
João Torrinhas 98435

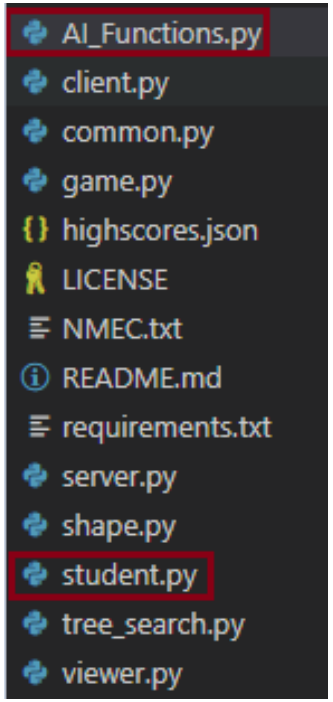
Organização do Código e Funcionamento do agente

- Student.py

  Comunicação com o servidor e inicialização do agente.

- AI_Functions.py

  Funções auxiliares e funções para o cálculo da melhor jogada de acordo com a heurística calculada.



O agente calcula todas as posições possíveis de uma peça e as suas rotações e, depois disso, irá atribuir um score, calculado através da heurística, a cada posição. A seguir, irá escolher a posição com o melhor score e irá mover a peça para essa posição e assim sucessivamente para as restantes peças.

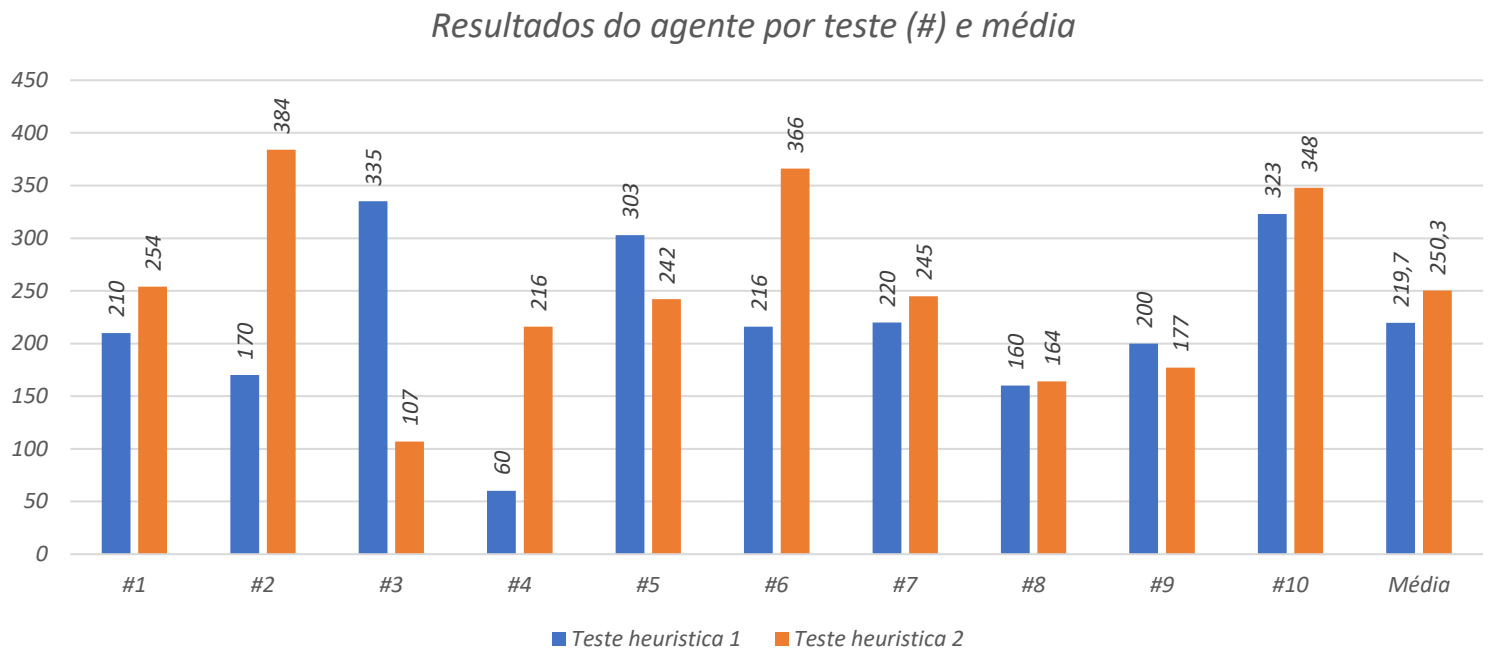
AI_Functions.py

- **Complete_lines**, calcula o número de linhas completas.
- **Calculate_holes**, calcula o número de buracos no tabuleiro.
- **Aggregate_height**, calcula a soma das alturas de todas as colunas.
- **Calculate_smotheness**, calcula as diferenças absolutas entre todas as duas colunas adjacentes.
- **Moveleft**, move a peça para a esquerda.
- **Moveright**, move a peça para a direita.
- **Lastposition**, calcula a posição final da peça na grelha.
- **Rotate_piece**, retorna todas as rotações de cada peça.
- **Out_of_board**, verifica se a peça está dentro dos limites do tabuleiro.
- **Next_actions**, retorna todas as posições possíveis da peça, bem como as suas keys.
- **Calculate_score_and_keys**, retorna as keys da posição que tem maior score.
- **Calculate_heuristic_and_score**, calcula o valor da heurística que irá ser usado para calcular a posição com melhor score.
- **Best_position**, calcula a melhor posição da peça tento em conta os diversos scores para cada uma das posições.
- **Collum_Heights**, calcula as alturas de todas as colunas.

Resultados Obtidos

Até agora, o resultado mais alto obtido foi de 428.

Na tabela seguinte são apresentados os resultados do agente em 10 jogadas diferentes:



No trabalho enviado, usámos os valores da heurística 2, pois os resultados são melhores.
Nota: Os valores da heurística estão presentes na função *Calculate_heuristics_and_Scores,* na classe *AI_Functions*.

Conclusão

- Otimizações que ficaram por implementar:
  - O agente não está a calcular as melhores posições para as próximas peças;
  - Os valores para o cálculo da heurística não são os mais rigorosos;
  - Tentámos implementar a pesquisa A* usando tree search mas, devido à falta de tempo, não conseguimos terminar a implementação.

Nota: Testámos o agente em Windows e verificámos que os resultados eram sempre à volta dos 220-230 e que, a partir desse valor, parece que o agente perde a conexão ao servidor, perdendo logo a seguir.

Apreciação final:
Consideramos que este projeto foi bastante interessante e que contribuiu positivamente para o nosso conhecimento na área da inteligência artificial.
De referir que da primeira entrega para esta, conseguimos uma grande evolução do nosso agente.