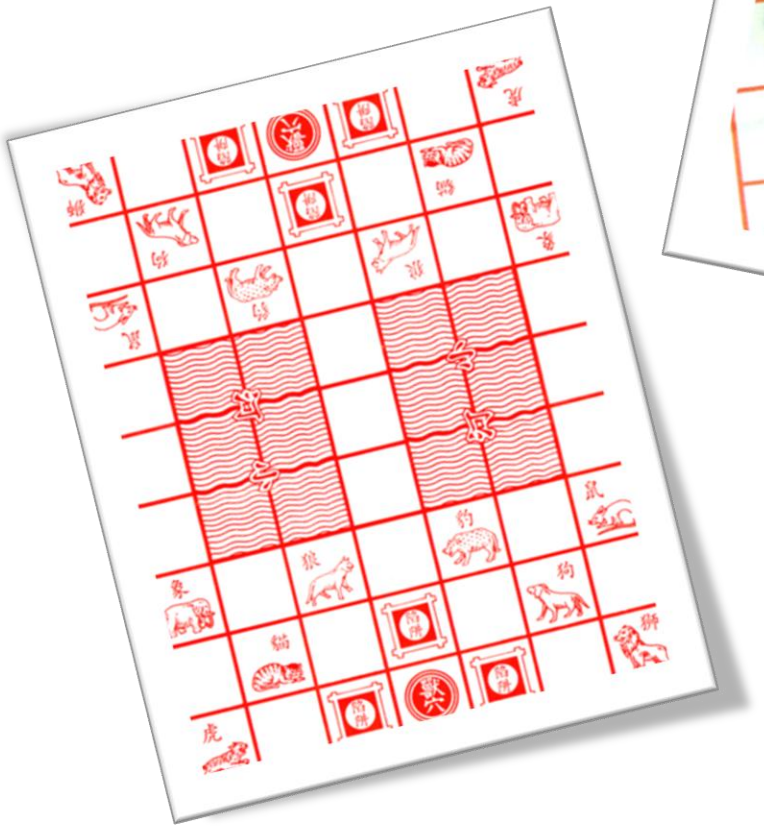
Jungle Chess

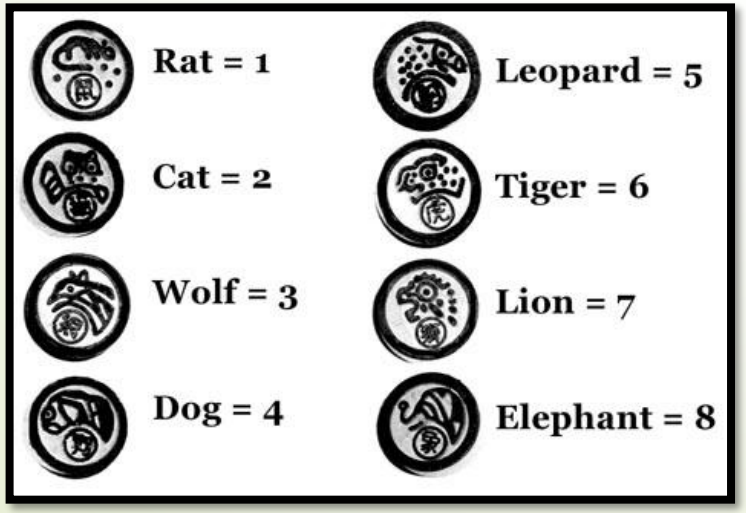
Trabalho Prático 1 - Jogo adversarial



Realizado por:
Ezequiel Paulo
Filipe Huang
Manuel Mota

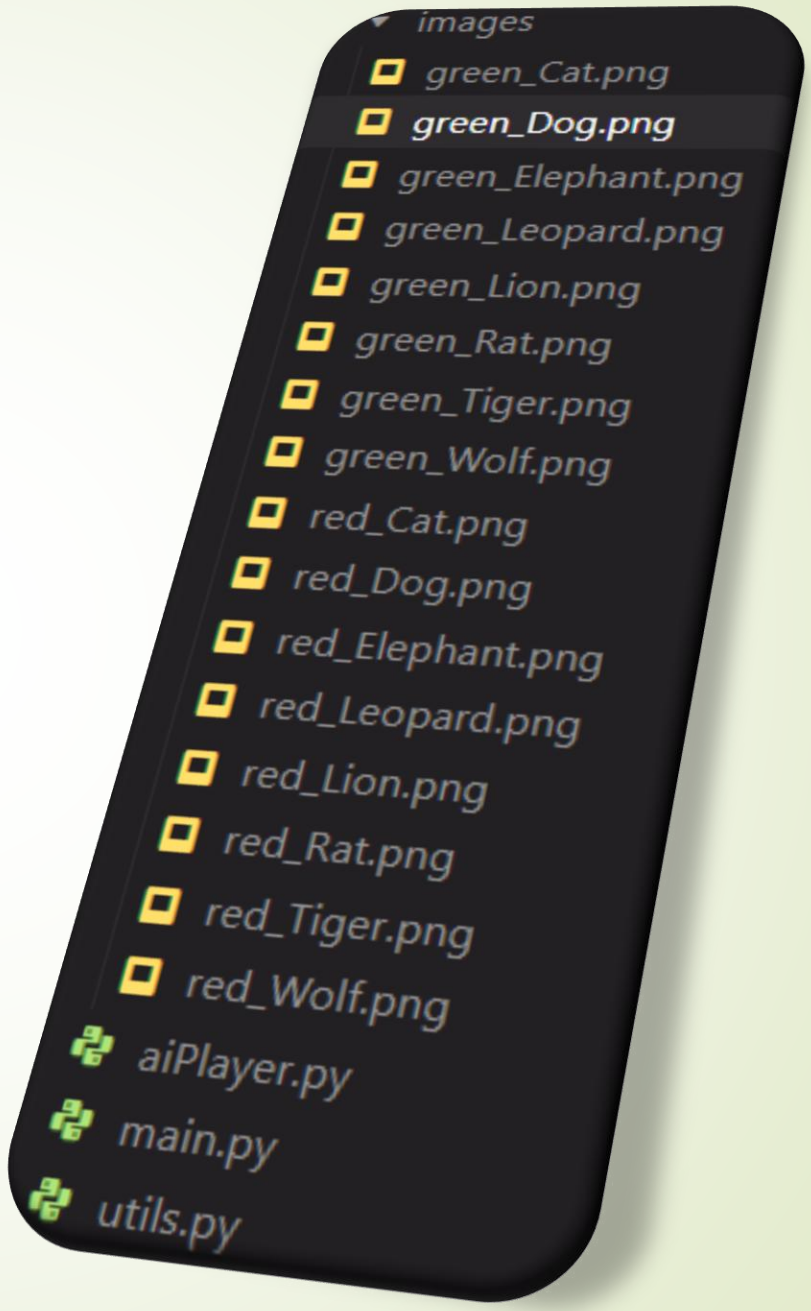
Introdução ao Jogo

- Jungle Chess, também conhecido por Dou Shou Qi, é um tabuleiro tradicional chinês de dois jogadores que consiste em:
  - 8 animais: Rato, Gato, Cão, Lobo, Leopardo, Tigre, Leão e Elefante
  - Com uma hierarquia crescente de captura dos animais, por ordem, com a exceção que o rato captura o elefante
  - Objetivo: invadir a “toca” (den) do adversário ou capturar todos os animais do mesmo
  - O tabuleiro é constituído por 3 terrenos além de casas normais: rios, armadilhas e tocas
  - Casos especiais: tigre e leão pode saltar por cima do rio e o rato pode nadar



Estrutura do Projeto

- Organização dos códigos:
  - main.py – Interface gráfica e a execução do jogo
  - utils.py – Constantes de funções auxiliares
  - aiPlayer.py – Implementações dos las
  - images – imagens utilizadas para as peças do jogo na interface gráfica



Formulação do Jogo

- Representação de estados:
  - Matriz bidimensional 9 por 7, com 16 peças (8 animais verdes e 8 animais vermelhos)
  - Dois jogadores: verde (baixo) e vermelho (cima)
  - Estado do jogo: a decorrer ou término com um vencedor
- Estado inicial:
  - Peças nas posições de acordo com as regras tradicionais
  - Terreno: zona dos rios, das armadilhas e das tocas
  - Turno inicial: começa o verde (jogador de baixo)
- Estado objetivo:
  - Mover a peça para a toca do oponente
  - Capturar todas as peças do oponente

Formulação do Jogo – Continuação

- Ações e operadores:
  - Movimentos normais: frente, trás, direito, esquerdo
  - Movimentos especiais: saltar o rio (tigre e leão) e entrar na água (rato)
  - Armadilhas: peças nas armadilhas podem ser capturadas por todos os animais
  - Captura das peças: de acordo com a hierarquia dos animais, e capturar o mesmo tipo
  - Exceção de captura: rato pode capturar o elefante, mas não o oposto

Algoritmos implementados

- Jogador random (RandomPlayer): escolhe o movimento aleatório de todas as jogadas legais e possíveis
- Jogador negamax (NegamaxPlayer): variante de Minimax, procurar sempre o movimento que dá máximo de pontuação e mínimo de adversário
- Jogador Alpha-Beta cut (AlphaBetaPlayer):
  - com base no algoritmo de negamax, cortar os ramos desnecessários, ou seja, não é importante verificar
  - Ordenação das pontuações dos movimentos para ser mais eficiente
  - Algoritmo de Iterative Deepening, facilitar as pesquisas
- Heurística:
  - Valor das peças baseado nos ranks
  - Proximidade da toca do adversário
  - Ameaças dos ambos às peças e à toca

Interface Gráfica

Alguns resultados

Estatísticas de casa IA em 10 tentativas :

Win/Loss (Av Time)	Random (red)	Negamax (red)	Alpha-Beta (red)
Random (green)	4/6 (32s)	0/10 (7s)	0/10 (4s)
Negamax (green)	10/0 (11s)	7/3 (37s)	5/5 (17s)
Alpha-Beta (green)	10/0 (2s)	4/6 (42s)	8/2 (12s)

- Jogo mais rápido: Negamax (green) vs. Alpha-Beta (red) – 2.37s
- Jogo mais lento: Negamax (green) vs. Negamax (red) – 1min 02

Bibliografia

- https://github.com/ManuelMota10/Jungle_GAME - GitHub do Projeto
- No fim do ReadMe tem todas as fontes que precisamos para o Projeto