# ATTAX

- Objetivo: implementar o código do jogo ATTAX, e um algoritmo de pesquisa que garanta a vitória do computador em 3 diferentes tabuleiros;
- IDEs usadas: “Visual Studio Code”;

Linguagem usada: Python;

- Algoritmo de pesquisa: Minimax & alphabetacuts;

Trabalho prático 1

Apresentação final

Gonçalo Monteiro

Alexandre Carneiro

# Formulaçao do problema

O jogo:
(estado inicial)
num tabuleiro (eg. 5x5) dois jogadores iniciam a partida com duas peças de cores diferentes ( eg. vermelho vs azul) ;

em cada canto do tabuleiro; através do movimento das peças: mover uma casa ( deixando uma peça “filho” na posição anterior ou mover 2 casas (através de salto, sem deixar peça “filho”);

Quando uma peça entra em contacto com peças da cor oposta, estas ultimas são “infetadas” e mudam de cor/equipa.

(estado final)
O jogo termina quando não há mais jogadas possíveis ( casas vazias ) ou quando só existe uma cor no tabuleiro;
A equipa vencedora é aquela que tem mais peças da sua cor no tabuleiro;

# Operadores & Heuristica

- Posição inicial da bola: P[x][y];
- Azul= 1;
- Vermelho= 2;
- Bloco(cor preta)= valor 8;
- Casas vazias= 0;
- Casas estendiveis=3

| nome | Pré-condição | Efeito | Custo |
|---|---|---|---|
| CIMA | P[x][y+1]=0 | P[x][y+1]=1(v2);<br>Pfilho[x][Y]=1(v2); | 1 |
| BAIXO | P[x][y-1]=0 | P[x][y-1]=1(v2);<br>Pfilho[x][y]=1(v2); | 1 |
| DIREITA | P[x+1][y]=0 | P[x+1][y]=1(v2);<br>Pfilho[x][y]=1(v2); | 1 |
| ESQUERDA | P[x-1][y]=0 | P[x-1][y]=1(v2);<br>Pfilho[x][y]=1(v2); | 1 |
| DIAGONAL | P[x+-2][y+-2]=0 | P[x+-2][y+-2]=1(v2);<br>Pfilho[x][y]=0; | 1 |
| PULO<br>(-> , <- , ^ ,_) | P[x+-2]v[y+-2]=0 | P[x+-2]v<br>[y+-2]=1(v2);<br>Pfilho[x][y]=0; | 1 |

Heurística: o numero de peças ( o maior possível) do jogador que tem a vez;
função: valores _para_minimax;

Avaliaçao: a cada jogada, o numero de peças é contado de modo  a comparar e determinar quem esta mais perto da vitória( através da subtraçao do numero de peças de uma cor pela outra;

# READ ME

- Garantir que os três tabuleiros e o código de attax estão na mesma pasta;
- Abrir a shell e ir ao diretório onde estao localizados os 4 ficheiro;
- Abrir o terminal de python dentro da shell (escrever python);
- Escrever - exec(open("attax.py").read());
- Devera abrir uma janela com botões é so carregar no modo de jogo em seguida no tabuleiro e na situação de player vs pc também na dificuldade do pc;

# Resultados exprimentais

- Em todos os testes pc v hum a vitória foi da inteligência artificial ( minimax);
- Em jogos pc v pc a vitori é do player 2;

# CONCLUSÃO

- O que conseguimos:

-Implementamos a lógica do jogo no programa com sucesso, assim como 3 tabuleiros diferentes, o minimax com alfa beta cuts com 2 dificuldades ( fácil e difícil ), e 4 modos de jogo( pc v pc; pc v human, human v human);

- O que não conseguimos:

-o jogo em modo pc vs pc funciona sem ter de esperar com um “clic” pois se o fizesse sem esperar ele so me desenhava o tabuleiro no fim do jogo estar resolvido;

# Referencias bibliográficas e links como fonte de informação

- “tkinter — Python interface to Tcl/Tk — Python 3.10.4 documentation “

& “ https://anzeljg.github.io/rin2/book2/2405/docs/tkinter/index.html “ , para compreender melhor as funcionalidade do tkinter;

- Powerpoints teóricos da cadeira;

- Parte gráfica: Código de tic-tac-toe disponibilizado pelo professor para as noções básicas das capacidades gráficas do Python e da biblioteca tkinter;

- Artificial Intelligence ModernApproach 3rdEdition;