

Mini relatório 1 (Connect 4)

João Vivas nº 202108177;

Gelson Varela nº 202109347

Linguagem utilizada:

Decidimos fazer este trabalho em java. Nós escolhemos esta linguagem, pois no trabalho anterior usamos Python e agora decidimos usar algo mais complexo para variar. Também, porque Java possui muitas das qualidades que Python possui como poder criar vários objetos personalizados, mas com a vantagem de melhor performance devido a ser uma linguagem compilada.

Estrutura do tabuleiro do jogo

Ao contrário do trabalho anterior, em que o tabuleiro foi guardado numa lista de inteiros, o tabuleiro deste jogo vai ser guardado numa matriz de caracteres. Decidimos usar esta estrutura, para ser mais fácil para verificar se o jogo atingiu um fim, se um jogador ganhou ou se o tabuleiro ficou cheio, logo o jogo ficou empatado. Cada “local” onde pode ser guardada uma peça tem três estados ‘.’ caso não tenha nenhuma peça de nenhum jogador ‘O’ ou ‘X’ que representam as peças dos jogadores. A qual representam cada símbolo ao humano ou ao computador e a ordem de quem começa o jogo ainda está para ser definida no futuro.

Movimentos do jogo

Para implementarmos os movimentos do jogo nós criamos duas funções **isFull** que retorna um valor booleano. Esta função verifica se a peça no topo de cada coluna está ocupada ou não, caso todas as peças do topo de todas as colunas estiverem ocupadas, retorna *true*, senão retorna *false*. Outra função usada é a função **put** que simplesmente adiciona a peça à coluna que o utilizador quer caso, esta não esteja cheia. Enquanto o jogo estiver a decorrer os movimentos vão estar dentro de um ciclo onde em cada ciclo a função **isFull** é executada e tem que dar resultado falso.

Em cada ciclo a jogada é alterada entre humano e computador. Se for a vez do humano, o programa pede para inserir o número de uma coluna. Quando inserido a função **put** é executada onde verifica de baixo para cima o estado de cada local onde possa estar uma peça naquela coluna e verifica se está ocupado ou não. Assim que encontra um sítio vazio na coluna coloca lá a peça se chegar ao último sítio e estiver ocupado pede para se voltar a inserir outro número, para colocar a peça noutra coluna.