

### 3º Trabalho de Inteligência Artificial

1.a) A estrutura de dados escolhida para a representação dos estados do jogo foi uma lista de listas onde cada lista era uma linha do tabuleiro.

1.b)

```
terminal(G) :- linhas(G,_).
terminal(G) :- colunas(G,_).
terminal(G) :- diagonal(G,_).
terminal(G) :- cheio(G).

linhas([[X,X,X,_],_,_,_],X) :- X \= v.
linhas([[_,X,X,X],_,_,_],X) :- X \= v.
linhas([[_,_,X,X,X],_,_,_],X) :- X \= v.
linhas([[_,X,X,X,_],_,_,_],X) :- X \= v.
linhas([[_,_,X,X,X],_,_,_],X) :- X \= v.
linhas([[_,_,_,X,X,X],_,_,_],X) :- X \= v.
linhas([[_,_,X,X,X,_],_,_,_],X) :- X \= v.
linhas([[_,_,_,X,X,X],_,_,_],X) :- X \= v.
linhas([[_,_,_,_,X,X,X],_,_,_],X) :- X \= v.
linhas([[_,_,_,X,X,X,X],_,_,_],X) :- X \= v.
linhas([[_,_,_,_,_,X,X,X],_,_,_],X) :- X \= v.

colunas([[X,_,_,_,_],[X,_,_,_,_],[X,_,_,_,_],_,X) :- X \= v.
colunas([_,[X,_,_,_,_],[X,_,_,_,_],[X,_,_,_,_]],X) :- X \= v.
colunas([_,_,[X,_,_,_,_],[X,_,_,_,_],_,X) :- X \= v.
colunas([_,_,_,[X,_,_,_,_],[X,_,_,_,_]],X) :- X \= v.
colunas([_,_,_,_,[X,_,_,_,_],[X,_,_,_,_]],X) :- X \= v.
colunas([_,_,_,_,_,[X,_,_,_,_],[X,_,_,_,_]],X) :- X \= v.
colunas([_,_,_,_,_,_,[X,_,_,_,_],[X,_,_,_,_]],X) :- X \= v.
colunas([_,_,_,_,_,_,_,[X,_,_,_,_],[X,_,_,_,_]],X) :- X \= v.
colunas([_,_,_,_,_,_,_,_,[X,_,_,_,_],[X,_,_,_,_]],X) :- X \= v.
colunas([_,_,_,_,_,_,_,_,_,[X,_,_,_,_],[X,_,_,_,_]],X) :- X \= v.

diagonal([[X,_,_,_,_],[_,X,_,_,_],[_,_,X,_,_],[_,_,_,X,_,_]],X) :- X \= v.
diagonal([[_,X,_,_,_],[_,_,X,_,_],[_,_,_,X,_,_],[_,_,_,_,X,_,_]],X) :- X \= v.
diagonal([[_,_,X,_,_],[_,_,_,X,_,_],[_,_,_,_,X,_,_],[_,_,_,_,_,X,_,_]],X) :- X \= v.
diagonal([[_,_,_,X,_,_],[_,_,_,_,X,_,_],[_,_,_,_,_,X,_,_],[_,_,_,_,_,_,X,_,_]],X) :- X \= v.
diagonal([[_,_,_,_,X,_,_],[_,_,_,_,_,X,_,_],[_,_,_,_,_,_,X,_,_],[_,_,_,_,_,_,_,X,_,_]],X) :- X \= v.
diagonal([[_,_,_,_,_,X,_,_],[_,_,_,_,_,_,X,_,_],[_,_,_,_,_,_,_,X,_,_],[_,_,_,_,_,_,_,_,X,_,_]],X) :- X \= v.
```

1.c)

```
valor(G, 1) :- linhas(G,x).  
valor(G, 1) :- colunas(G,x).  
valor(G, 1) :- diagonal(G,x).  
valor(G, -1) :- linhas(G,o).  
valor(G, -1) :- colunas(G,o).  
valor(G, -1) :- diagonal(G,o).  
valor(_, 0).
```

1.d) Ficheiro minimax\_tres\_linha.pl enviado juntamente com o relatório

1.e) Ficheiro alfa\_beta\_linha.pl enviado juntamente com o relatório.

Relativamente à comparação com o algoritmo minimax podemos observar que em termos de tempo este algoritmo é muito mais rápido do que o minimax em situações que não a situação do tabuleiro estar vazio. Nestas outras situações o alfa-beta em termos de performance é melhor pois vê muitos menos estados do que o minimax e consequentemente em muito menos tempo mas se tivermos em conta a jogada ótima o minimax pode ter vantagem em relação ao alfa-beta pois vê mais estados.

1.f)Ficheiro inteligente.pl enviado juntamente com o relatório. A jogada inserida pelo jogador deve ser no formato “2,1.” por exemplo.

Obversação: Em anexo enviamos também uma pasta que contém os ficheiros anteriores mas desta vez para a versão de três linhas e três colunas visto que esta versão demora bastante tempo em alguns casos.