

# Reversi

Neste breve relatório, iremos discutir as estratégias que o grupo utilizou para implementar o jogo contra o computador no nosso programa.

Tal como foi sugerido no enunciado do projeto, o nosso bot possui 3 níveis de dificuldade:

- **Bot Básico (nível 1)**
- **Bot Médio (nível 2)**
- **Bot Avançado (nível 3)**

## 1. Bot Básico

Tal como o nome indica, o primeiro nível do bot é dos mais básicos possíveis. O grupo pretendia oferecer um bot que não causasse muitos problemas ao jogador, que simplesmente permitisse o funcionamento do modo automático e que oferecesse a resposta mais rápida possível, sem pensar nas jogadas seguintes. Desta forma, a função **jogaBot** irá receber o apontador para o estado atual do jogo e um array multidimensional, que contem as jogadas possíveis para um certo estado de jogo. Desta maneira, é nos possível selecionar a primeira jogada válida e assim, através das funções responsáveis por “comer” as peças do adversário, atualizar as peças do jogador e colocar as peça do jogador (‘X’ ou ‘O’) no tabuleiro. Uma vez que foi dado o apontador para esse estado, consegue-se que o estado seja modificado.

## 2. Bot Médio e Bot Avançado

Para a realização estes bots, o grupo seguiu as instruções fornecidas pelos professores e implementou o algoritmo do **MiniMax**.

Este algoritmo consiste em minimizar a maior perda possível, ou seja, fornecendo o estado atual do jogo e uma lista de jogadas válidas, usando a recursividade, é possível prever os estados futuros do jogo. Assim sempre que o adversário joga, o algoritmo escolhe a jogada que mais o beneficiará. Quando for a vez do jogador, de entre as jogadas que mais favorecem o adversário, será selecionada a de menor valor, para que assim, tal como foi dito anteriormente, se minimize a maior perda possível.

Assim, foram necessárias várias funções para realizar esse algoritmo:

- ✓ **maxIndice:** retorna o índice correspondente ao maior valor de um vetor;
- ✓ **max:** retorna o maior valor de um vetor;
- ✓ **min:** retorna o menor valor de um vetor;
- ✓ **avaliaEstado:** uma das principais funções do nosso algoritmo. Em vez de avaliar o estado pelo número de peças comidas, o grupo achou melhor atribuir valores a cada posição do tabuleiro. A valoração de cada posição foi feita com o pensamento de favorecer os cantos do tabuleiro e desfavorecer as posições anteriores a esses cantos, pois conquistando um canto, torna-se mais difícil para o adversário ganhar o jogo. Através deste método, é feita uma melhor gestão do jogo por parte do bot, uma vez que caso optasse pelo número de peças comidas, os cantos do tabuleiro nunca iriam ser uma prioridade.  
Assim, soma-se o valor das posições onde se encontram as peças do jogador, de modo a obter o resultado do tabuleiro de jogo.
- ✓ **simulaJogada:** de modo a conseguir saber os estados futuros, criou-se uma função que sabendo o estado atual do tabuleiro e as coordenadas de uma posição, simula a jogada para essa posição, criando um novo estado correspondente a essa jogada (de maneira a não alterar o estado atual do jogo).
- ✓ **recursiva:** primeiramente, a função recebe um valor correspondente à profundidade a que se pretende chegar. É criada uma lista com as posições válidas para um certo estado de jogo e fornecido o número de jogadas possíveis. Partindo desses dados, invoca-se a função **simulaJogada** e, recursivamente, faz-se a função **recursiva**, mas com (profundidade-1), para assim ter-se sempre os estados diferentes provenientes de uma jogada apenas, uma vez que os resultados são avaliados partindo da profundidade máxima para a profundidade mínima. Sempre que for a vez do adversário jogar maximiza-se essa jogada, e minimiza-se quando é o bot a jogar. Esta operação é feita através do resto da divisão do estado por 2, ou seja, sempre que nível é par, maximiza-se, sempre que nível for ímpar minimiza-se (pois quando é par é a jogada do bot e quando é ímpar é a jogada do adversário).
- ✓ **BOT:** faz com que sabendo os valores de cada jogada, se selecione o índice que contém o maior valor e efetue a jogada.

### **3. Diferenças entre Bot Médio e Bot Avançado**

De maneira a diferenciar os 2 bots, definiu-se que o Bot Médio iria percorrer 2 níveis de profundidade e o Bot Avançado iria percorrer 6 níveis de profundidade.

Após alguns testes, conclui-se que apesar de o Bot Avançado ter que analisar bastantes jogadas, o tempo que demorava a realizar a jogada não ultrapassava os 2 segundos, por isso o grupo achou melhor manter o nível de profundidade.