REVERSI

No âmbito da cadeira de Laboratórios de informática II, foi realizado, ao longo do 2º semestre do 1º ano, uma versão virtual do famoso jogo de tabuleiro Reversi (ou Othello).

• Jogo em geral

Tal como o jogo de tabuleiro, e as versões digitais já existentes, o nosso jogo Reversi baseia-se num tabuleiro 8x8, semelhante ao de xadrez, no qual dois jogadores lutam para controlar a maioria do tabuleiro. Cada jogador começa com duas peças no centro do tabuleiro, e devem colocar peças entre as suas e as do adversário, para as "comer". O jogo acaba quando nenhum dos jogadores tiver mais jogadas válidas, ou, num caso mais concreto, quando o tabuleiro estiver completamente preenchido. O jogador que tiver mais peças suas no tabuleiro é o vencedor.

O nosso jogo, para além das funcionalidades básicas de jogar, tem a possibilidade de guardar e carregar jogos, de desfazer jogadas, de fornecer dicas ou de indicar onde o jogador pode jogar. É ainda possível jogar contra um jogador virtual, um "bot", sobre o qual vamos falar agora.

• Bot

Uma das opções oferecidas ao jogador é a possibilidade de jogar em modo automático, ou seja, contra um "bot", que pode adotar estratégias diferentes, dependendo da dificuldade do jogo. Definimos 3 modos diferentes para o bot:

1. Modo Fácil

Para este modo, a estratégia utilizada é a de selecionar, de todas as jogadas que o bot pode fazer, uma completamente aleatória. Assim, o bot pode assemelhar-se a alguém que não sabe jogar Reversi, e que coloca as peças à sorte. Dito isto, ainda é possível que o bot vença neste modo, se tiver muita sorte, mas o mais provável será o jogador ganhar, desde que tenha uma estratégia minimamente eficaz.

2. Modo Intermédio

Neste nível de dificuldade, o grupo adotou uma estratégia mais complexa. Primeiramente, é verificado se, na lista de posições onde o bot pode jogar, existe algum canto. Caso isto se verifique, essa posição é imediatamente escolhida para o bot jogar, devido ao seu valor estratégico, que na maioria das vezes pode levar o jogador à vitória.

Caso não existam cantos possíveis, o bot tenta jogar numa aresta, já que estas também são, de forma geral, posições boas para jogar. Dentro de todas as arestas disponíveis, é selecionada aquela que "come" um maior número de peças adversárias.

Se o bot não puder jogar num canto nem numa aresta, tenta selecionar, de todas as outras posições onde pode jogar, aquela que lhe permite "comer" mais peças do adversário.

Esta estratégia, ao contrário da anterior, já distingue entre posições boas e más onde jogar, apesar de apenas calcular o valor imediato de uma posição, por outras palavras, uma jogada pode parecer boa no momento, mas acabar por ser má mais à frente, algo que o bot não verifica neste modo. Para um jogador que esteja a aprender a jogar

Reversi, este bot será um adversário competente, mas para quem já domina o jogo, ser-lhe-á muito fácil derrotar este bot.

3. Modo Difícil

Para este modo, o grupo adotou uma estratégia mais elaborada e muito mais eficaz para a vitória do bot.

Através do algoritmo *minimax*, com otimização *alpha-beta pruning,* antes do bot efetuar uma jogada, ele tenta determinar todas as formas como o jogo pode decorrer nos próximos n turnos (o valor de n varia com base nas jogadas possíveis no momento, por razões de eficiência do algoritmo, mas nunca é menos que 5), e, a partir de todas essas possibilidades, tenta calcular aquela que mais beneficia o bot, assumindo sempre que o jogador jogará de forma a prejudicar o bot de forma máxima. Por outras palavras, o algoritmo atribui uma pontuação a cada um dos tabuleiros correspondentes às jogadas possíveis pelos n turnos. Depois, a partir das jogadas do adversário que dão menor valor ao tabuleiro (min), seleciona a maior dessas (max), daí o nome do algoritmo. A jogada que leva a esse resultado é a escolhida pelo bot para jogar.

Esta estratégia escolhe uma jogada com base nos “estragos” que o adversário pode fazer ao tabuleiro, tentando minimizá-los enquanto maximiza o valor do tabuleiro para o bot, e, ao contrário do modo intermédio, que apenas verificava qual a melhor jogada naquele turno, tenta calcular qual será a melhor jogada mais à frente.

• Validação de jogadas e dicas

Para validar uma jogada, isto é, para verificar se uma dada jogada pode ser efetuada, temos uma lista de jogadas possíveis para o jogador que irá jogar nesse turno, que é atualizada sempre que o tabuleiro é modificado. Quando um jogador tenta jogar numa determinada posição, o programa vê se essa posição se encontra na lista de jogadas possíveis, e em caso afirmativo realiza essa jogada. Para gerar essa lista, usamos uma função que verifica, em cada posição vazia, se existem peças do adversário adjacentes. Caso existam, verifica se é possível fazer uma linha reta que una essa posição a uma posição com uma peça do jogador, nunca passando por posições vazias. Esta reta pode ser horizontal, vertical ou diagonal. Esta lista é também usada para mostrar ao jogador onde pode jogar. Ao imprimir o tabuleiro, é verificado se cada posição pertence ou não a essa lista. Se pertencer, é imprimido um ‘.’ nessa posição.

O sistema de dicas funciona com base no bot. Uma dica não é mais do que uma jogada que um bot faria se estivesse a jogar em vez do jogador. Ao pedir uma dica, o jogo cria um bot temporário, ao qual é atribuída a peça do jogador, e é pedido a esse bot para escolher onde jogar. Depois, essa posição é imprimida no tabuleiro com um ‘?’, dando ao jogador uma ajuda sobre onde deve jogar. Este bot está na dificuldade 3, ou seja, usa o algoritmo minimax para decidir onde jogar.

Projeto realizado por:
Ana Filipa Pereira (a89589)
Carolina Santejo (a89500)
Sofia Santos (a89615)

MIEI – Universidade do Minho – 2019