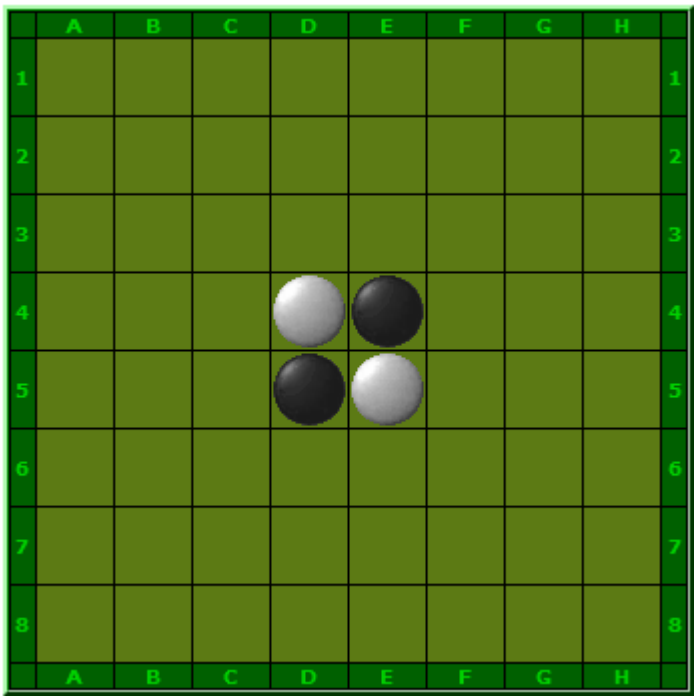
Riversi



Grupo 8 :

Paulo Oliveira A87550

Paulo Costa A87986

Rui Baptista A87989

Universidade do Minho
Escola de Ciências

Universidade do Minho

Modo Automático – Nível 1 (Básico)

Neste nível a estratégia utilizada foi fazer com que o bot executasse a sua jogada na primeira opção que fosse possível. Ao atuar deste modo, não escolherá a jogada por qualquer tipo de estratégia com intuito de vencer, mas sim apenas com o intuito de continuar o jogo, sendo ainda assim possível vir a vencer o adversário humano.

```
ESTADO jogadaBotBas (ESTADO e, int l, int c){

    int l2, c2;
    int r=0;

    for (l2 = 0; l2 < 8; l2++) {
        for (c2 = 0; c2 < 8; c2++) {
            if (valida (e,l2,c2) && r==0){
                e.grelha [l2] [c2] = e.peca;
                e= trocardepecascc (e, l2, c2);
                e= trocardepecascb (e, l2, c2);
                e= trocardepecasld (e, l2, c2);
                e= trocardepecasle (e, l2, c2);
                e= trocardepecasd1 (e, l2, c2);
                e= trocardepecasd2 (e, l2, c2);
                e= trocardepecasd3 (e, l2, c2);
                e= trocardepecasd4 (e, l2, c2);
                r++;
            }
        }
    }
    return e;
}
```

(1) Função utilizada para o bot básico

Modo Automático – Nível 2 (Médio)

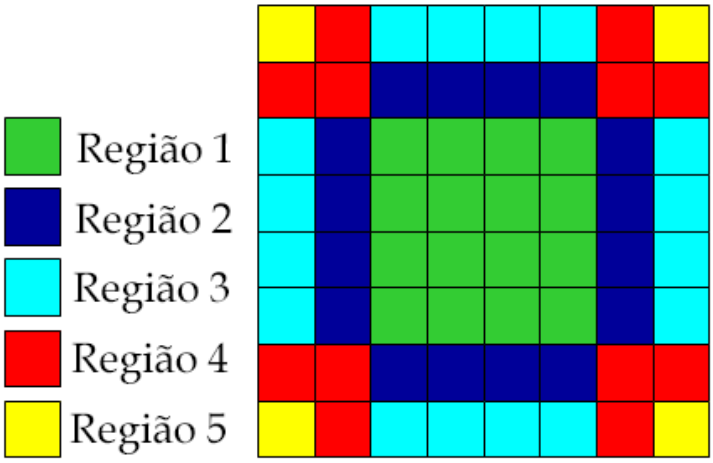
No nível intermédio optou-se por se fazer com que o bot opte pela jogada em que conquista mais peças, tendo assim sido criada uma função que compara todas as jogadas possíveis no momento, acabando por escolher a opção com mais peças para conquista. Desta forma, não joga de forma aleatória como no nível anterior, sendo assim a dificuldade superior ao mesmo.

```
ESTADO jogadaBotMed (ESTADO e, int l, int c){
    int lb, cb;          // linha bot & coluna bot
    int minbot=65;              // guardar a jogada que come menos peças
    int lbg, cbg;    // linha e coluna  do bot que está a ser guardada


    for (lb = 0; lb < 8; lb++) {
        for (cb = 0; cb < 8; cb++) {
            if (valida (e, lb, cb) == 1) {
                if (quantaspecas (e, lb , cb) < minbot){
                    minbot = quantaspecas (e, lb , cb);
                    lbg = lb;
                    cbg = cb;
                }
            }
        }
    }

    e.grelha [lbg] [cbg] = e.peca;
    e= trocardepecascc (e, lbg, cbg);
    e= trocardepecascb (e, lbg, cbg);
    e= trocardepecasld (e, lbg, cbg);
    e= trocardepecasle (e, lbg, cbg);
    e= trocardepecasd1 (e, lbg, cbg);
    e= trocardepecasd2 (e, lbg, cbg);
    e= trocardepecasd3 (e, lbg, cbg);
    e= trocardepecasd4 (e, lbg, cbg);

    //printa(e);
    return e;

}
```

(2) Função utilizada para o bot médio

Modo Automático – Nível 3 (Profissional)

No terceiro e último nível, o nível profissional, a estratégia abordada foi dividir o tabuleiro em 5 regiões distintas. Estas sendo a região 1, sendo que as jogadas nesta região são consideradas inúteis, visto que ocorrem na parte central do tabuleiro, sendo assim as peças nestas posições facilmente capturadas pelo adversário. A região 2, onde a jogada é bastante perigosa, dando ao adversário a oportunidade de obter os cantos do tabuleiro. A região 3, sendo esta região caracteristicamente perigosa, pois permite ao adversário conquistar posições importantes no tabuleiro, como as extremidades do mesmo. A região 4, sendo esta uma das regiões mais importantes do tabuleiro, visto que as jogadas nesta região permitem facilitar a conquista das laterais do tabuleiro, aumentando assim a probabilidade de vitória. E por fim, a região 5, a mais procurada e valiosa, pois permite ao jogador uma maior facilidade de obtenção dos cantos do tabuleiro, sendo estes as posições de maior importância no tabuleiro.



(3) Regiões do tabuleiro utilizadas

Para além da divisão do tabuleiro em cinco regiões, utilizou-se ainda uma maneira de "desempate", no caso, de existirem mais que uma jogada possível na mesma região. Quando então existir um caso destes, ainda se verá qual das jogadas em comparação irá trazer mais peças conquistadas em favor do bot. Assim, há uma jogada efetuada pelo bot, claramente mais forte em seu favor, comparativamente aos bots de dificuldade inferior, e daí este ser o método utilizado no bot de dificuldade profissional.

Observações

Para que o modo automático funcionasse de forma correta, foi necessário implementar algumas alterações no menu, face ao modo manual, atingindo assim todos os objetivos propostos. Um destes objetivos sendo que após a leitura do ficheiro, a primeira jogada após esta leitura teria de ser obrigatoriamente por parte do humano. Para que tal fosse possível foi implementado na função de leitura uma distinção entre os dois modos. Outro cenário relevante foi conseguir guardar o nível em que o modo automático estaria a atuar, continuando assim a jogar nesse mesmo nível após a leitura do ficheiro. Para além destes dois aspetos, criou-se uma função que não permitisse ao bot jogar quando houvesse um erro por parte do humano, como por exemplo uma jogada inválida. Para além disso, sempre que houvesse um jogo contra o bot, independentemente do nível, o primeiro jogador teria de ser obrigatoriamente o que estivesse a jogar com a peça X, tendo assim sido alterado o código de forma a tornar esse requerimento possível.

```
else if (v[0][0] == 'A') {

   e.modo= '1';

   if (v[0][2]=='X') e.peca = VALOR_X;
   else if (v[0][2]=='O') e.peca = VALOR_O;

   if (v[0][4]=='1') e.nivel = 1;
   else if (v[0][4]=='2') e.nivel = 2;
   else if (v[0][4]=='3') e.nivel = 3;
```

(4) Leitura do modo de jogo e nível do ficheiro

```
case 'O':
    e = grelha_inicial(e, VALOR_O, '1');
    e.peca = inverso (e.peca);
    indicacao = 1;
    printf("A %s %s\n", peca, nivel);

    break;
```

(5) Caso em que o jogador escolhe a peça O e o bot tem de efetuar a primeira jogada

Validação de jogadas

A nossa estratégia para validar uma jogada introduzida pelo jogador, foi criar 8 funções distintas, mas ambas com o mesmo método, apenas com o intuito de abranger as 8 direções em que estas podem ser válidas. Isto é, funções que verificassem se a jogada introduzida estivesse precedida de peças inversas à sua, e quando estas não fossem do adversário, teria de haver uma peça sua, de forma, a que as suas peças "cercassem" as do adversário.

```
int validacc (ESTADO e, int x , int y){

    VALOR p,g;
    int i=0;
    p=e.peca;
    g=inverso(p);

    if(e.grelha [x-1] [y] == g){
        i= x -1;
        while (i>0 && e.grelha [i] [y] == g){
            i--;
        }
        if (e.grelha [i] [y] == p)
            return 1;
    }
    return 0;
}
```

(6) Uma das 8 funções que valida, ou não, a jogada

Após se validar uma jogada, as peças que foram cercadas, terão de ser "convertidas" para peças do jogador que efetuou a jogada. Para que isto acontecesse, criámos ainda 8 funções, também nas 8 direções possíveis.

```
ESTADO trocardepecascb (ESTADO e, int x, int y) {
    VALOR p, g;
    int i = 0;
    p = e.peca;
    g = inverso(p);

    if (validacb(e, x, y)) {
        i = x + 1;
        while (i < 7 && e.grelha[i][y] == g) {
            e.grelha[i][y] = e.peca;
            i++;
        }
    }
    return e;
}
```

(7) Uma das 8 funções que troca as peças conquistadas

Sugestão de jogada

Quando o jogador quisesse uma sugestão de jogada, opção (H) do interpretador, optámos por uma jogada semelhante à do bot de dificuldade básica, ou seja, a primeira das jogadas possíveis pelo jogador atual.

Isto, porque achamos que o jogador que pede uma sugestão de jogada, não possa ser "beneficiado" com uma jogada devidamente pensada e com bastante proveito para o seu jogo, como por exemplo, a utilizada pelo bot de dificuldade profissional, ou até mesmo, de dificuldade média.

```
case 'H':

    n = sscanf(linha, "%s", cmd);
    int l2, c2;
    int parar=0;
    int coord=0;

    printf ("Tabuleiro com a sugestao: \n");
    for (l2 = 0; l2 < 8; l2++) {
        for (c2 = 0; c2 < 8; c2++) {
            if (valida (e,l2,c2) && parar==0){
                printf ("? ");
                parar++;
            }
            else if (e.grelha [l2] [c2] == VALOR_X ) printf ("X ");
            else if (e.grelha [l2] [c2] == VALOR_O ) printf ("O ");
            else printf ("- ");
        }
        printf("\n");
    }

    printf ("Coordenadas da jogada sugestao: \n");              //2 CICLOS PARA DAR AS COORDENADAS DA JOGADA SUGESTÃO
    for (l2 = 0; l2 < 8; l2++) {
        for (c2 = 0; c2 < 8; c2++) {
            if (valida(e, l2, c2) && coord == 0) {
                printf("Linha: %d Coluna: %d \n", l2+1 , c2+1);  // +1 para assumir a posiçao corretamente
                coord++;
            }
        }
    }

    break;
```

(8) Função que sugere a jogada