Jogo optado: Jogo da Velha 3x3

Nome dos integrantes:

Matheus Farias Porto Dos Anjos

RGM: 31259821

Ryan Do Nascimento Bezerra

RGM: 31268684

Fábio Harley Silva Filho

RGM: 30290759

Gabriel Henriques Cassiano

RGM: 32096143

Introdução:

O jogo da velha é um jogo para dois jogadores, a estrutura do jogo é apenas um tabuleiro 3x3 onde os jogadores marcam 'X' ou 'O' em cada campo do tabuleiro. Para vencer o jogo, um dos jogadores deve preencher uma linha, uma coluna ou uma diagonal inteira com 'X', ou 'O', o primeiro que fizer isso vence, e o jogo termina, caso o tabuleiro fique totalmente preenchido e ninguém ganhar, teremos um empate.

Descrição geral do jogo:

O jogo da velha desenvolvido possui um menu inicial onde o jogador pode escolher:

- **Jogar:**

  Abre mais duas opções de escolha, uma para o modo jogador vs jogador, e outra para o modo jogador vs computador;

- **Ver Ranking Geral:**

  Log de todas as partidas jogadas após a primeira execução do script;

- **Créditos:**

  Nome dos desenvolvedores do algoritmo;

- **Sair:**

  Opção de terminar a execução do script;

Para jogar, a cada jogada precisamos fornecer dois números que representam a posição do tabuleiro em que queremos jogar: a coluna e a linha, nessa ordem. As linhas, bem como as colunas, variam de 0 até 2 e somente os locais vazios podem ser escolhidos como jogada.

Demonstração do código:

1. Interação com o Jogador em jogo

```
do{
    printf("\n %s - Coloque a coluna: ", dados[i].jogador);
    retorno = scanf("%d",&c);
    printf(" %s - Coloque a linha: ", dados[i].jogador);
    retorno = scanf("%d",&l);
do{
    letra = getchar();
    printf("%c", letra);
    }while(letra != '\n');
}while(retorno == 0 || c < 0 || c >= 3 || l < 0 || l >= 3);



if((matriz[l][c] == 'O') || (matriz[l][c] == 'X')){
    printf("\nPosicao Ocupada");
    Sleep(1000);
    jogadas--;
    break;
}else if(jogadas %2 == 1){
    matriz[l][c] = 'X';
    sair = 1;
}else if(jogadas %2 == 0){
    matriz[l][c] = 'O';
    sair = 1;
}
```

Figura 1 - Entrada de Dados, Linha e Coluna

Tivemos que limpar o buffer do teclado para evitar o loop infinito no caso do jogador digitar sem querer alguma letra e isso pode ser resolvido com a função getchar(). O último dado no buffer do teclado sempre será um “\n”, que representa a tecla ENTER. Assim, basta ler caractere por caractere enquanto o caractere lido for diferente de “\n”, como apresentado no trecho de código a seguir, limpando o buffer e dando oportunidade para o usuário digitar novamente. O while de retorno == 0 foi modificado conforme os

limites impostos por nós, como visto acima, caso o jogador digite qualquer número diferente de 0, 1, 2, haverá o retorno de colocar a linha e a coluna novamente

Uma área de armazenamento temporário é chamada de **buffer**. Todos os dispositivos de entrada e saída padrão contêm um **buffer** de entrada e saída.

2. Dados de Vitórias em arquivo

```
void rank(){
    FILE *p;
    char str[80] = "test0.txt";
    char f;

    p = fopen(str,"a+");

    if(dados[0].vitorias >0 || dados[1].vitorias >0 || test == 1){
    fprintf(p,"%s ganhou %d vezes\n", dados[0].jogador, dados[0].vitorias);
    fprintf(p,"%s ganhou %d vezes\n", dados[1].jogador, dados[1].vitorias);
    fclose(p);
    }else{
    p = fopen(str,"a+");
    while (!feof(p)){
        fscanf(p,"%c",&f);
        printf("%c",f);
    }
    fclose(p);

}
}
```

Figura 2 – Área de Ficheiros

O "**void rank()**" foi definido como uma função que salva os dados de vitórias dos jogadores/computador na partida armazenados em uma struct, esses dados são salvos no arquivo "ranking" toda vez que uma nova partida é concluída, se encerrada pela opção "Sair (4)" (Partidas de uma execução anterior da atual também ficam salvas com o intuito de ser um ranking geral de todas as partidas já jogadas)

3. Jogador VS Computador

Um dos problemas que tivemos foi relacionado ao "BOT" que jogaria contra o usuário caso não houvesse segundo jogador. Inicialmente, pensamos em apenas fazer com que esse bot gerasse números aleatórios de 0 a 2, tanto para linha quanto para coluna, assim, definindo o local da jogada do computador. Segue a imagem:

```
while(sair == 0){
    if(bot ==1 && jogadas %2 == 0){

    srand (time(NULL)); //BOT
    l = rand() % 3 + 0;
    c = rand() % 3 + 0;

    bot_tentativa++;

    if(bot_tentativa > 5){
        for(l = 0; l < 3; l++){
            for(c = 0; c < 3; c++)
                if(matriz[l][c] == ' '){
                    matriz[l][c] = 'O';
                    l = 3;
                    c = 3;
                    bot_tentativa = 0;
                    sair = 1;
                }

        }

    }else if(matriz[l][c] == 'O' || matriz[l][c] == 'X'){
        jogadas--;
        break;
    }else if(jogadas %2 == 0){
        matriz[l][c] = 'O';
        sair = 1;
```

Figura 3 - Área do BOT

Para que isso fosse possível, utilizamos a função ***srand()*** para geração de números aleatórios, mas após alguns testes, notamos que, na verdade, gerar números aleatórios não é tão simples assim: o maior dos problemas era que a própria função ***srand()*** não gerava números aleatórios. Em todos os testes, os números gerados eram os mesmos, e no fim, o computador nunca jogava, pois o local da jogada já estava ocupado.

Mais tarde, descobrimos ser necessário definir uma **"semente"** para a função ***srand()***, o pontapé inicial para que a função começasse a gerar números **diferentes**. Precisávamos que essa semente mudasse a todo momento, para isso, usamos uma nova função: **time(NULL)** dentro da função **srand()**. Essa função retorna os segundos desde 1 de Janeiro de 1970 até o horário local da máquina que está rodando o código, dessa forma, a **semente** muda a cada segundo que é passado.

Mesmo depois disso, ainda existia um limitador que impedia que o código rodasse corretamente: a probabilidade que a função gerava números diferentes era bem baixa, acreditamos ser devido à pouca liberdade de escolha de números pela função, na qual definimos de 0 até 2. A solução implementada foi fazer com que, caso o computador tenha 6 ou mais tentativas de jogadas, e ainda assim, não foi possível encontrar um local

vazio, forçamos o BOT a percorrer toda a matriz da tabela 3x3 utilizando 2 “**for()”** até que ele encontre o primeiro espaço vazio no jogo da velha.

Código Fonte:

```
1   #include <stdio.h>
2   #include <string.h>
3   #include <stdlib.h>
4   #include <windows.h>
5   #include <time.h>
6   #include <locale.h>
7
8   int **matriz;
9   int l, c, id;
10  int jogadas, empate, sair, rodando;
11  int escolha, esc, resposta, bot ,bot_tentativa;
12  char letra;
13  int retorno = -1;
14
15  struct t_dados{
16        char jogador[30];
17        int vitorias;
18  };
19        t_dados dados[2];
20  void rank(){
21        FILE *p;
22        char str[80] = "ranking";
23        char f;
```

```
24
25        p = fopen(str,"a+");
26
27        if(dados[0].vitorias >0 || dados[1].vitorias >0 || empate == 1){
28              fprintf(p,"%s   ganhou   %d   vezes\n",   dados[0].jogador,
29  dados[0].vitorias);
30              fprintf(p,"%s   ganhou   %d   vezes\n",   dados[1].jogador,
31  dados[1].vitorias);
32              fclose(p);
33        }else{
34             p = fopen(str,"a+");
35               while (!feof(p)){
36                     fscanf(p,"%c",&f);
37               printf("%c",f);
38               }
39                     fclose(p);
40        }
41  }
42  void tabela(){
43                     printf("\n");
44                     printf("\tJogo da velha\n");
45                     printf("\n\n\t 0   1   2\n\n");
46               for(l = 0; l < 3; l++){
47                   for(c = 0; c < 3; c++){
48                       if(c == 0)
49                           printf("\t");
50                           printf(" %c ", matriz[l][c]);
51                       if(c < 2)
52                           printf("|");
```

```
53                      if(c == 2)
54                          printf("  %d", l);
55                  }
56                  if(l < 2)
57                    printf("\n\t-----------");
58                    printf("\n");
59              }
60  }
61  void menu(){
62        do{
63              printf("\tJogo da Velha");
64              printf("\n1 - Jogar\n");
65              printf("2 - Ver Ranking Geral\n");
66              printf("3 - Créditos\n");
67              printf("4 - Sair\n");
68              printf("Digite o número da sua escolha: ");
69              retorno = scanf("%d" , &escolha);
70              do{
71                    letra = getchar();
72                    printf("%c", letra);
73                    system("cls");
74                    if (escolha == 1){
75                          system("cls");
76                          do{
77                                printf("1 - Jogador vs Computador\n");
78                                printf("2 - Jogador vs Jogador\n");
79                          printf("Digite o número da sua escolha: ");
80                          retorno = scanf("%d" , &bot);
```

```
81                      do{
82                            letra = getchar();
83                            printf("%c", letra);
84                            system("cls");
85                      }while(letra != '\n');
86                }while(retorno == 0 || bot != 1 && bot != 2);
87                      system("cls");
88                      if(bot == 1){
89                            fflush(stdin);
90                            printf("Nome de Usuario: ");
91                            gets(dados[0].jogador);
92                            strcpy(dados[1].jogador, "Computador");
93                            system("cls");
94                      }else
95                      if(bot == 2){
96                            fflush(stdin);
97                            printf("Nome do primeiro Usuario: ");
98                            gets(dados[0].jogador);
99                            fflush(stdin);
100                           printf("Nome do segundo Usuario: ");
101                           gets(dados[1].jogador);
102                           fflush(stdin);
103                           system("cls");
104                     }
105               }else
106               if (escolha == 2){
107                     system("cls");
108                     do{
```

```
109                               printf("Ranking Geral:\n");
110                               rank();
111                         printf("Digite (3) para retorna ou (4) para sair:
112   ");
113                         retorno = scanf("%d", &esc);
114                         do{
115                               letra = getchar();
116                               printf("%c", letra);
117                               system("cls");
118                         }while(letra != '\n');
119                   }while(retorno == 0 || esc != 3 && esc != 4);
120
121                         if(esc == 4){
122                               exit(0);
123                         }else
124                         if(esc == 3){
125                         system("cls");
126                         }
127
128                   }else
129                   if (escolha == 3){
130                               system("cls");
131                         do{
132                               printf("\tCréditos\n");
133                               printf("Ryan do Nascimento Bezerra\n");
134                               printf("Matheus    Farias    Porto    Dos
135   Anjos\n");
136                               printf("Fábio Harley Silva Filho\n");
137                               printf("Gabriel Henriques Cassiano\n");
```

```
138                         printf("Digite (3) para retorna ou (4) para sair:
139 ");
140                         retorno = scanf("%d", &esc);
141                         do{
142                               letra = getchar();
143                               printf("%c", letra);
144                               system("cls");
145                         }while(letra != '\n');
146                   }while(retorno == 0 || esc != 3 && esc != 4);
147                         if(esc == 4){
148                               exit(0);
149                         }
150                   }else
151                   if (escolha == 4){
152                         exit(0);
153                   }
154             }while(letra != '\n');
155       }while(retorno == 0 || escolha != 1);
156 }
157 void propriedade(){
158       matriz = (int **) malloc(sizeof(int*)*3);
159
160       for(l=0;l<3;l++){
161             matriz[l] = (int *) malloc(sizeof(int*)*3);
162             for(c=0;c<3;c++)
163                   matriz[l][c] = ' ';
164       }
165 }
```

```
166  void interacao(){
167
168         while(sair == 0){
169               if(bot ==1 && jogadas %2 == 0){
170                     srand (time(NULL));
171                     l = rand() % 3 + 0;
172                     c = rand() % 3 + 0;
173
174                     bot_tentativa++;
175
176               if(bot_tentativa > 5){
177                   for(l = 0; l < 3; l++){
178                      for(c = 0; c < 3; c++)
179                          if(matriz[l][c] == ' '){
180                              matriz[l][c] = 'O';
181                              l = 3;
182                              c = 3;
183                              bot_tentativa = 0;
184                              sair = 1;
185                           }
186                           }
187                     }else
188                     if(matriz[l][c] == 'O' || matriz[l][c] == 'X'){
189                           jogadas--;
190                           break;
191                     }else
192                     if(jogadas %2 == 0){
193                           matriz[l][c] = 'O';
```

```
194                         sair = 1;
195                   }
196             }else{
197                   do{
198                         printf("\n   %s   -   Coloque   a   coluna:   ",
199   dados[id].jogador);
200                         retorno = scanf("%d",&c);
201                         printf("   %s   -   Coloque   a   linha:   ",
202   dados[id].jogador);
203                         retorno = scanf("%d",&l);
204                         do{
205                             letra = getchar();
206                         printf("%c", letra);
207                   }while(letra != '\n');
208             }while(retorno == 0 || c < 0 || c >= 3 || l < 0 || l >= 3);
209
210
211                   if(matriz[l][c] == 'O' || matriz[l][c] == 'X'){
212                         printf("\n\tPosicao Ocupada");
213                         Sleep(1000);
214                         jogadas--;
215                         break;
216                   }else
217                   if(jogadas %2 == 1){
218                         matriz[l][c] = 'X';
219                         sair = 1;
220                   }else
221                   if(jogadas %2 == 0){
222                         matriz[l][c] = 'O';
```

```
                        sair = 1;
                    }
                }
        }
}
void VerificaVitoria(){

        if(matriz[0][0] == 'X' && matriz[1][1] == 'X' && matriz[2][2] ==
'X' ||
            matriz[0][2] == 'X' && matriz[1][1] == 'X' && matriz[2][0] ==
'X'){
            printf("\n\t%s (X), Venceu!", dados[0].jogador);
            dados[0].vitorias++;
            rodando=0;
        }else
        if(matriz[0][0] == 'X' && matriz[1][0] == 'X' && matriz[2][0] ==
'X' ||
                matriz[0][1] == 'X' && matriz[1][1] == 'X' && matriz[2][1]
== 'X' ||
                matriz[0][2] == 'X' && matriz[1][2] == 'X' && matriz[2][2]
== 'X'){
                printf("\n\t%s (X), Venceu!", dados[0].jogador);
                dados[0].vitorias++;
                rodando=0;
        }else
        if(matriz[0][0] == 'X' && matriz[0][1] == 'X' && matriz[0][2] ==
'X' ||
                matriz[1][0] == 'X' && matriz[1][1] == 'X' && matriz[1][2]
== 'X' ||
                matriz[2][0] == 'X' && matriz[2][1] == 'X' && matriz[2][2]
== 'X'){
```

```
254             printf("\n\t%s (X), Venceu!", dados[0].jogador);
255             dados[0].vitorias++;
256             rodando=0;
257       }else
258       if(matriz[0][0] == 'O' && matriz[1][1] == 'O' && matriz[2][2] ==
259   'O' ||
260             matriz[0][2] == 'O' && matriz[1][1] == 'O' && matriz[2][0]
261   == 'O'){
262             printf("\n\t%s (O), Venceu!", dados[1].jogador);
263             dados[1].vitorias++;
264             rodando=0;
265       }else
266       if(matriz[0][0] == 'O' && matriz[1][0] == 'O' && matriz[2][0] ==
267   'O'  ||
268             matriz[0][1] == 'O' && matriz[1][1] == 'O' && matriz[2][1]
269   == 'O' ||
270             matriz[0][2] == 'O' && matriz[1][2] == 'O' && matriz[2][2]
271   == 'O'){
272             printf("\n\t%s (O), Venceu!", dados[1].jogador);
273             dados[1].vitorias++;
274             rodando=0;
275       }else
276       if(matriz[0][0] == 'O' && matriz[0][1] == 'O' && matriz[0][2] ==
277   'O' ||
278             matriz[1][0] == 'O' && matriz[1][1] == 'O' && matriz[1][2]
279   == 'O' ||
280             matriz[2][0] == 'O' && matriz[2][1] == 'O' && matriz[2][2]
281   == 'O'){
282             printf("\n\t%s (O), Venceu!", dados[1].jogador);
283             dados[1].vitorias++;
284             rodando=0;
```

```
285         }else
286         if(jogadas == 9){
287               printf("\nEmpate");
288               rodando=0;
289               empate = 1;
290         }
291   }
292   int main(){
293
294         setlocale(LC_ALL, "Portuguese");
295         rodando = 1;
296         menu();
297
298         do{
299               jogadas = 0;
300               propriedade();
301
302               while(rodando=1){
303                     system("cls");
304                     tabela();
305                     sair = 0;
306
307                     VerificaVitoria();
308                     if(rodando == 0){
309                           jogadas--;
310                           break;
311                     }
312                     jogadas++;
```

```
313
314                     if(jogadas %2 == 1){
315                           id=0;
316                     }else
317                     if(jogadas %2 == 0){
318                           id=1;
319                     }
320                     interacao();
321               }
322               do{
323                     Sleep(1000);
324                     system("cls");
325               printf("\nDigite (3) para jogar novamente ou (4) para sair:
326   ");
327               retorno = scanf("%d", &esc);
328               do{
329                     letra = getchar();
330                     printf("%c", letra);
331                     system("cls");
332               }while(letra != '\n');
333         }while(retorno == 0 || esc != 3 && esc != 4);
334                     if(esc == 4){
335                           rank();
336                           exit(0);
337                     }
338         }while(esc == 3);
339
340   //Libera memória da Matriz
```

```
341         if (matriz!=NULL){
342               for (l=0 ; l<3 ; l++){
343                     if (matriz[l] != NULL)
344                           free(matriz[l]);
345               }
346                           free(matriz);
347         }
348   }
```