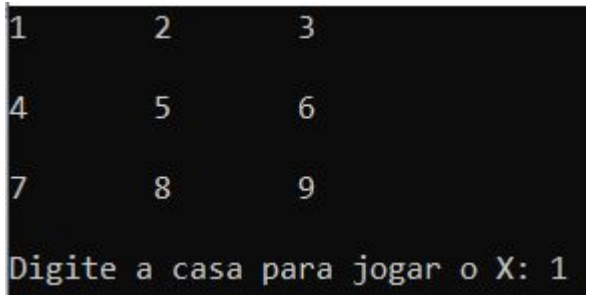
Projeto 1: Jogo da Velha

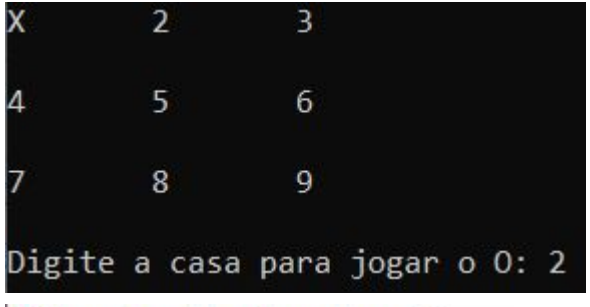
Introdução à Programação para Engenharias
SSC0304-101-2022

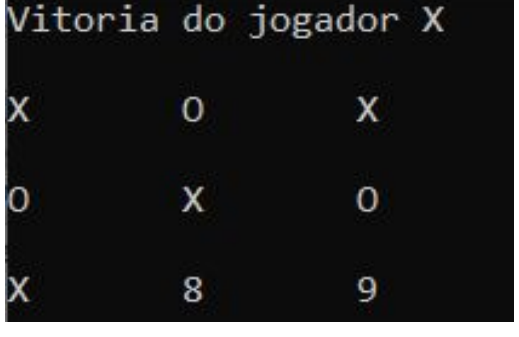
Professor: Pós Dr. Jó Ueyama

Flávia Rafaela Blandino - 13838878
Henri Miranda Magalhães - 13716600
João Henrique Aléssio - 13725928
Jonathan Silva dos Santos - 13679328

Como jogar nosso jogo da velha?







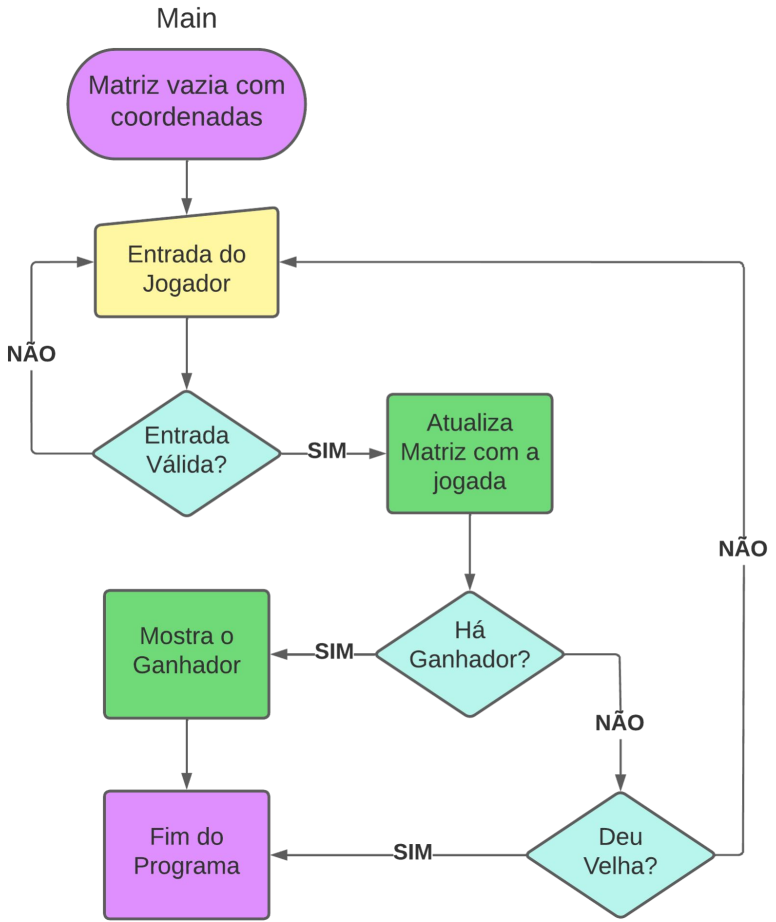
- Digitar o número correspondente à casa que o respectivo jogador deseja e apertar a tecla “Enter”;
- Verificar que sua jogada foi efetuada e a matriz atualizada;
- Digitar o número correspondente à casa que o próximo jogador deseja e apertar a tecla “Enter”;
- Continuar o Processo até que um dos jogadores vença.

Como funciona nosso jogo da velha?

```c
int main (){                                                    //Inicia função main
    int casaX, casaO, j, somaColun, velha=0, somaLin;           //Declara variáveis inteiras
    do{
        system("cls");                                          //Limpa tela
        printarM();                                             //Imprime a matriz do jogo
        printf("Digite a casa para jogar o X: ");               //Pede para 'X' jogar

        scanf("%d", &casaX);                                    //Lê a variável e associa a X
        jogadaX(casaX, j);                                      //Entra na função X

        vitoriaColun(somaColun);                                //Entra na função de vitória de vertical
        vitoriaLin(somaLin);                                    //Entra na função de vitória de linha
        vitoriaDiag(somaColun);                                 //Entra na função de vitória de diagonal
        velha=velha+1;                                          //Conta o número de jogadas para conferir se "deu velha"

        if(v1==1 || v2==1 || velha==9){                         //Condiciona vitória de X, O ou velha
            break;                                              //Interrompe loop
        }

        system("cls");                                          //Limpa tela
        printarM();                                             //Imprime a matriz do jogo
        printf("Digite a casa para jogar o O: ");               //Pede para 'O' jogar

        scanf("%d", &casaO);                                    //Lê a variável e associa a X
        jogadaO(casaO, j);                                      //Entra na função O

        vitoriaColun(somaColun);                                //Entra na função de vitória de vertical e diagonal
        vitoriaLin(somaLin);                                    //Entra na função de vitória de linha
        vitoriaDiag(somaColun);                                 //Entra na função de vitória de diagonal
        velha=velha+1;                                          //Conta o número de jogadas para conferir se "deu velha"

        if(v1==1 || v2==1){                                     //Condiciona vitória de X ou O
            break;                                              //Interrompe loop
        }
    }while(v1<1 && v2<1);                                       //Confere se não houve vitória para continuar o loop
```

Como funciona nosso jogo da velha?

```
int main (){
    int casaX, casaO, j, somaColun, velha=0, somaLin;
    do{
        system("cls");
        printarM();
        printf("Digite a casa para jogar o X: ");

        scanf("%d", &casaX);
        jogadaX(casaX, j);

        vitoriaColun(somaColun);
        vitoriaLin(somaLin);
        vitoriaDiag(somaColun);
        velha=velha+1;

        if(v1==1 || v2==1 || velha==9){
            break;
        }

        system("cls");
        printarM();
        printf("Digite a casa para jogar o O: ");

        scanf("%d", &casaO);
        jogadaO(casaO, j);

        vitoriaColun(somaColun);
        vitoriaLin(somaLin);
        vitoriaDiag(somaColun);
        velha=velha+1;

            if(v1==1 || v2==1){
                break;
            }
        }while(v1<1 && v2<1);

        if(velha==9 && v1!=1 && v2!=1){
            system("CLS");
            printarM();
            printf("\nDeu velha!!!\n");
        }
    }
```

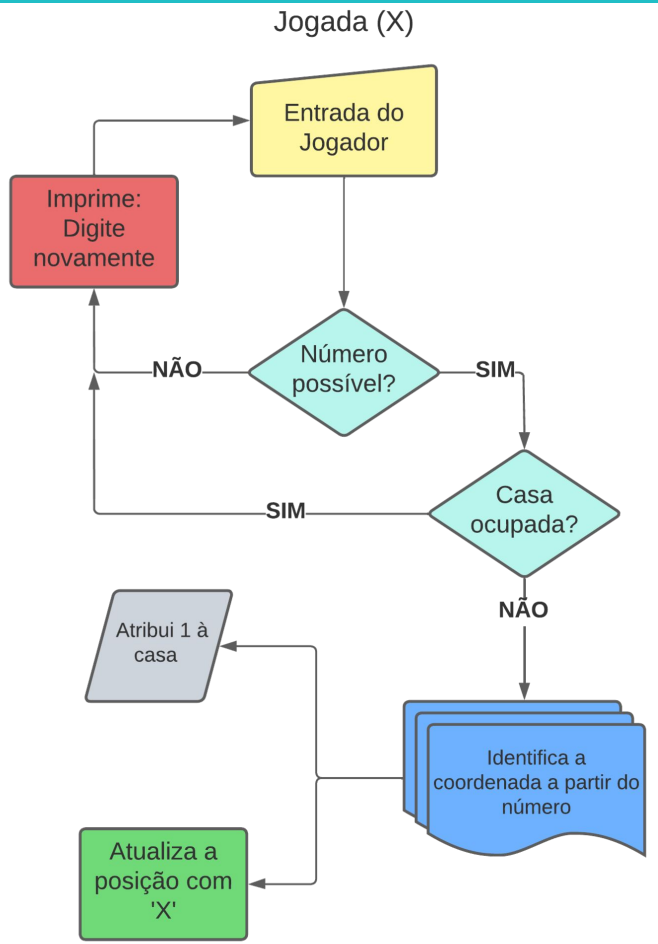


Função Jogada (X)

```c
void jogadaX (int casaX, int j) {                          //Função para jogada em X
    int x;
    if (casaX <= 0 || casaX > 9 || quantcasa[casaX]==casaX) {   //Condiciona existência e disponibilidade da casa
        do{
            printf ("Digite novamente: ");                 //Pede para digitar novamente
            scanf ("%d",&casaX);                           //(jogada inválida devido à posição inexistente ou casa ocupada)
        }while (casaX <= 0 || casaX > 9|| quantcasa[casaX]==casaX);   //Confere a disponibilidade da casa
    }
    quantcasa[casaX]=casaX;                                //Vetor que associa as casas digitadas a X, tornando-as ocupadas
    if (casaX <= 3){                                       //Condiciona número digitado X como parte da linha 1
        x = 3 - (4 - casaX);                               //Conversão para posição
        j = x;
        m[0][j] = 'X';                                     //Preenche posição com 'X'
        v[0][j] = 1;                                       //Ocupa casa
    }
    if (casaX > 3 && casaX <= 6){                          //Condiciona número digitado X como parte da linha 2
        x = 6 - (10 - casaX);                              //Conversão para posição
        j = x;
        m[1][j] = 'X';                                     //Preenche posição com 'X'
        v[1][j] = 1;                                       //Ocupa casa
    }
    if (casaX > 6) {                                       //Condiciona número digitado X como parte da linha 3
        x = 9 - (16 - casaX);                              //Conversão para posição
        j = x;
        m[2][j] = 'X';                                     //Preenche posição com 'X'
        v[2][j] = 1;                                       //Ocupa casa
    }
}
```

Função Jogada (X)

```
void jogadaX (int casaX, int j) {
    int x;
    if (casaX <= 0 || casaX > 9 || quantcasa[casaX]==casaX) {
        do{
            printf ("Digite novamente: ");
            scanf ("%d",&casaX);
        }while (casaX <= 0 || casaX > 9|| quantcasa[casaX]==casaX);
    }
    quantcasa[casaX]=casaX;
    if (casaX <= 3){
        x = 3 - (4 - casaX);
        j = x;
        m[0][j] = 'X';
        v[0][j] = 1;
    }
    if (casaX > 3 && casaX <= 6){
        x = 6 - (10 - casaX);
        j = x;
        m[1][j] = 'X';
        v[1][j] = 1;
    }
    if (casaX > 6) {
        x = 9 - (16 - casaX);
        j = x;
        m[2][j] = 'X';
        v[2][j] = 1;
    }
}
```

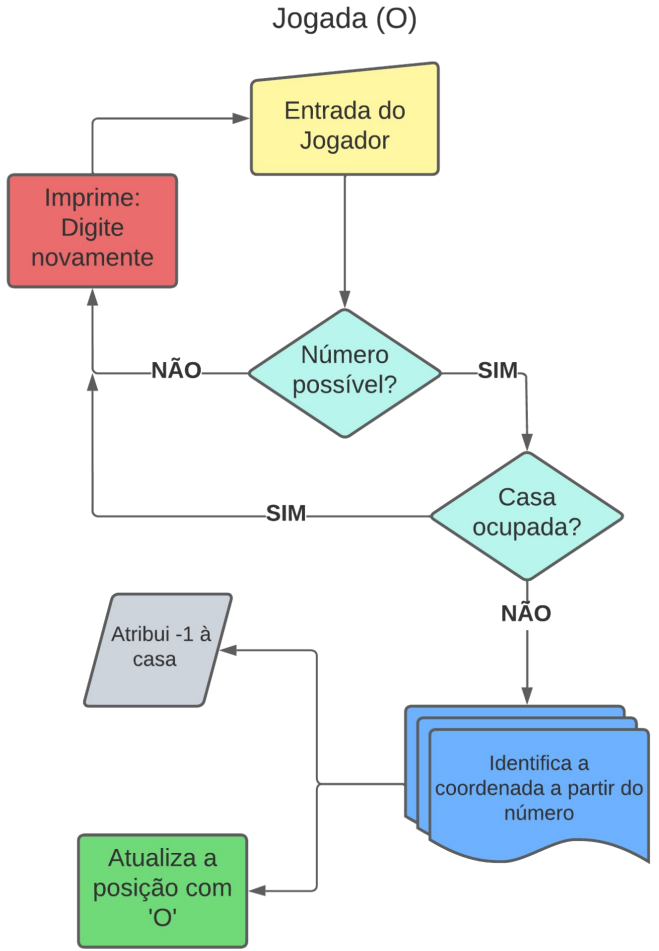


Função Jogada (O)

```c
void jogadaO (int casaO, int j) {                          //Função para jogada em O
    int y;
    if (casaO <= 0 || casaO > 9 || quantcasa[casaO]==casaO){   //Condiciona existência e disponibilidade da casa
        do{
            printf ("Digite novamente: ");                 //Pede para digitar novamente
            scanf("%d",&casaO);                            //(jogada inválida devido à posição inexistente ou casa ocupada)
        }while (casaO <= 0 || casaO > 9|| quantcasa[casaO]==casaO);   //Confere a disponibilidade da casa
    }
    quantcasa[casaO]=casaO;                                //Vetor que associa as casas digitadas a O, tornando-as ocupadas
    if (casaO <= 3){                                       //Condiciona número digitado O como parte da linha 1
        y = 3 - (4 - casaO);                               //Conversão para posição
        j = y;
        m[0][j] = 'O';                                     //Preenche posição com 'O'
        v[0][j] = -1;                                      //Ocupa casa
    }
    if (casaO > 3 && casaO <= 6){                          //Condiciona número digitado O como parte da linha 2
        y = 6 - (10 - casaO);                              //Conversão para posição
        j = y;
        m[1][j] = 'O';                                     //Preenche posição com 'O'
        v[1][j] = -1;                                      //Ocupa casa
    }
    if (casaO > 6){                                        //Condiciona número digitado O como parte da linha 3
        y = 9 - (16 - casaO);                              //Conversão para posição
        j = y;
        m[2][j] = 'O';                                     //Preenche posição com 'O'
        v[2][j] = -1;                                      //Ocupa casa
    }
}
```

Função Jogada (O)

```
void jogadaO (int casaO, int j) {
    int y;
    if (casaO <= 0 || casaO > 9 || quantcasa[casaO]==casaO){
        do{
            printf ("Digite novamente: ");
            scanf("%d",&casaO);
        }while (casaO <= 0 || casaO > 9|| quantcasa[casaO]==casaO);
    }
    quantcasa[casaO]=casaO;
    if (casaO <= 3){
        y = 3 - (4 - casaO);
        j = y;
        m[0][j] = 'O';
        v[0][j] = -1;
    }
    if (casaO > 3 && casaO <= 6){
        y = 6 - (10 - casaO);
        j = y;
        m[1][j] = 'O';
        v[1][j] = -1;
    }
    if (casaO > 6){
        y = 9 - (16 - casaO);
        j = y;
        m[2][j] = 'O';
        v[2][j] = -1;
    }
```

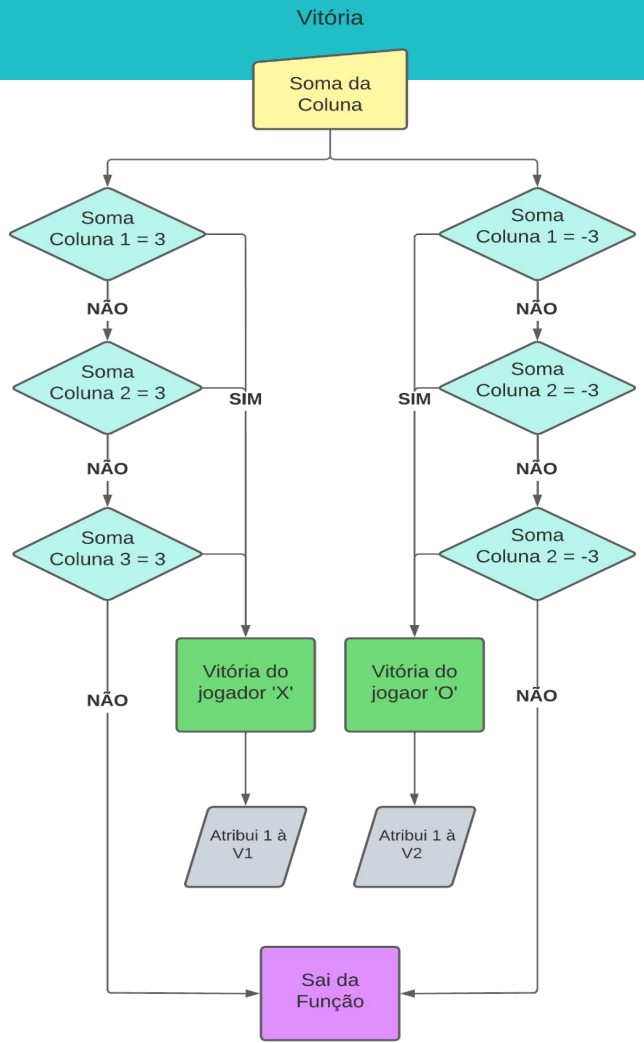


Função Vitória Coluna

```
void vitoriaColun(int somaColun){                                   //Função vitória vertical e diagonal
    somaColun=0;                                                    //Inicia soma dos valores da coluna em 0
    for(int i=0; i<3; i++){
        somaColun=somaColun+v[i][0];                                //Primeira coluna
            if(somaColun==3){                                       //Condiciona vitória de X pela primeira coluna
                system("CLS");                                      //Limpa Tela
                printf("Vitoria do jogador X\n\n");                 //Printa a vitória de X
                printarM();                                         //Imprime a matriz do jogo
                v1=1;                                               //Vitória de X
            }
            if(somaColun==-3){                                      //Condiciona vitória de O pela primeira coluna
                system("CLS");                                      //Limpa Tela
                printf("Vitoria do jogador O\n\n");                 //Printa a vitória de O
                printarM();                                         //Imprime a matriz do jogo
                v2=1;                                               //Vitória de O
            }
    }
    somaColun=0;                                                    //Reseta soma dos valores da coluna
    for(int i=0; i<3; i++){
        somaColun=somaColun+v[i][1];                                //Segunda coluna
            if(somaColun==3){                                       //Condiciona vitória de X pela segunda coluna
                system("CLS");
                printf("Vitoria do jogador X\n\n");                   //Printa a vitória de X
                printarM();
                v1=1;                                               //Vitória de X
            }
            if(somaColun==-3){                                      //Condiciona vitória de O pela segunda coluna
                system("CLS");
                printf("Vitoria do jogador O\n\n");                   //Printa a vitória de O
                printarM();
                v2=1;                                               //Vitória de O
            }
    }
```

Função Vitória Coluna

```
void vitoriaColun(int somaColun){
    somaColun=0;
    for(int i=0; i<3; i++){
        somaColun=somaColun+v[i][0];
            if(somaColun==3){
                system("CLS");
                printf("Vitoria do jogador X\n\n");
                printarM();
                v1=1;
            }
            if(somaColun==-3){
                system("CLS");
                printf("Vitoria do jogador O\n\n");
                printarM();
                v2=1;
            }
    }
    somaColun=0;
    for(int i=0; i<3; i++){
        somaColun=somaColun+v[i][1];
            if(somaColun==3){
                system("CLS");
                printf("Vitoria do jogador X\n\n");
                printarM();
                v1=1;
            }
            if(somaColun==-3){
                system("CLS");
                printf("Vitoria do jogador O\n\n");
                printarM();
                v2=1;
            }
    }
```

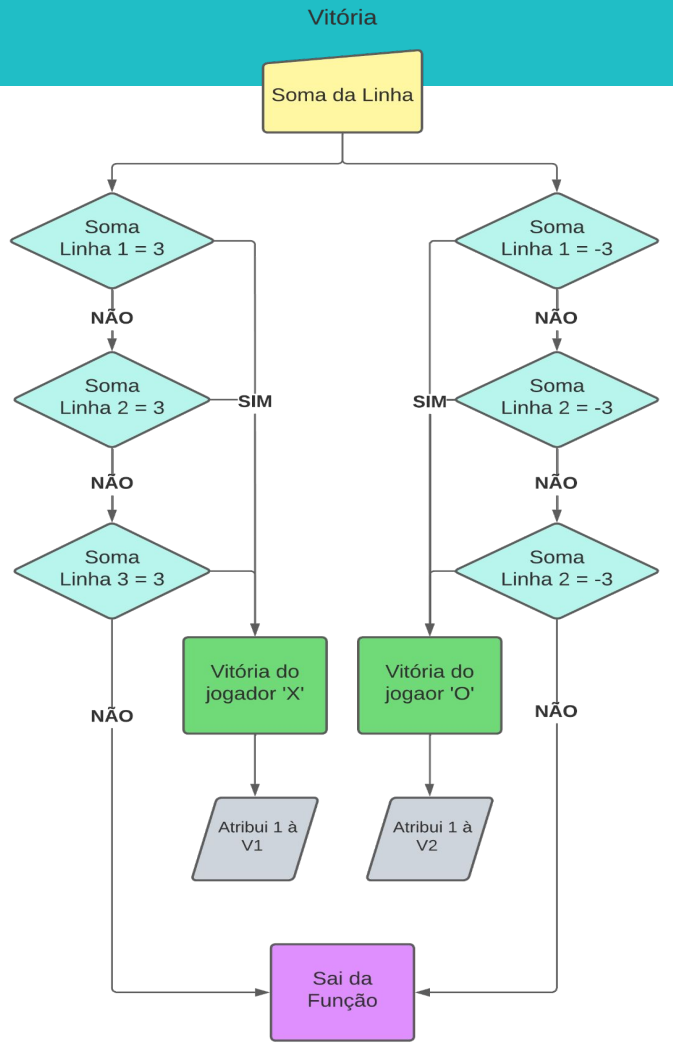


Função Vitória Linha

```
void vitoriaLin(int somaLin){                                   //Função vitória horizontal
    somaLin=0;                                                  //Inicia soma dos valores da linha em 0
    for(int j=0; j<3; j++){
        somaLin=somaLin+v[0][j];                                //Primeira linha
            if(somaLin==3){                                     //Condiciona vitória de X pela primeira linha
                system("CLS");
                printf("Vitoria do jogador X\n\n");              //Printa a vitória de X
                printarM();
                v1=1;                                           //Vitória de X
            }
            if(somaLin==-3){                                    //Condiciona vitória de O pela primeira linha
                system("CLS");
                printf("Vitoria do jogador O\n\n");              //Printa a vitória de O
                printarM();
                v2=1;                                           //Vitória de O
            }
    }
    somaLin=0;                                                  //Reseta soma dos valores para linha
    for(int j=0; j<3; j++){
        somaLin=somaLin+v[1][j];                                //Segunda linha
            if(somaLin==3){                                     //Condiciona vitória de X pela segunda linha
                system("CLS");
                printf("Vitoria do jogador X\n\n");              //Printa a vitória de X
                printarM();
                v1=1;                                           //Vitória de X
            }
            if(somaLin==-3){                                    //Condiciona vitória de O pela segunda linha
                system("CLS");
                printf("Vitoria do jogador O\n\n");              //Printa a vitória de O
                printarM();
                v2=1;                                           //Vitória de O
            }
    }
```

Função Vitória Linha

```
void vitoriaLin(int somaLin){
    somaLin=0;
    for(int j=0; j<3; j++){
        somaLin=somaLin+v[0][j];
            if(somaLin==3){
                system("CLS");
                printf("Vitoria do jogador X\n\n");
                printarM();
                v1=1;
            }
            if(somaLin==-3){
                system("CLS");
                printf("Vitoria do jogador O\n\n");
                printarM();
                v2=1;
            }
    }
    somaLin=0;
    for(int j=0; j<3; j++){
        somaLin=somaLin+v[1][j];
            if(somaLin==3){
                system("CLS");
                printf("Vitoria do jogador X\n\n");
                printarM();
                v1=1;
            }
            if(somaLin==-3){
                system("CLS");
                printf("Vitoria do jogador O\n\n");
                printarM();
                v2=1;
            }
    }
```

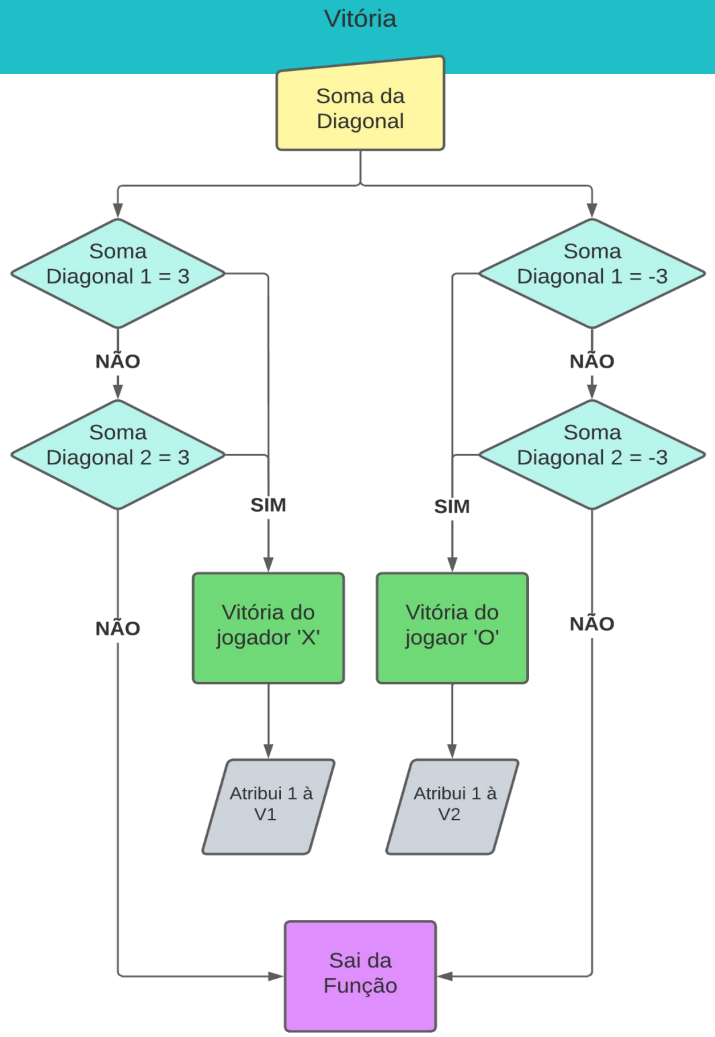


Função Vitória Diagonal

```
void vitoriaDiag(int somaDiag){
    somaDiag=0;                                          //Reseta soma dos valores para diagonal
    somaDiag = v[0][0] + v[1][1] + v[2][2];              //Soma posições que formam diagonal da esquerda para a direita
    if(somaDiag==3){                                     //Condiciona vitória de X pela primeira diagonal
        system("CLS");
        printf("Vitoria do jogador X\n\n");                //Printa a vitória de X
        printarM();
        v1=1;                                            //Vitória de X
    }
    if(somaDiag==-3){                                    //Condiciona vitória de O pela primeira diagonal
        system("CLS");
        printf("Vitoria do jogador O\n\n");                //Printa vitória de O
        printarM();
        v2=1;                                            //Vitória de O
    }
    somaDiag=0;                                          //Reseta soma dos valores para diagonal
    somaDiag=v[2][0]+v[1][1]+v[0][2];                    //Soma posições que formam diagonal da direita para a esquerda
    if(somaDiag==3){                                     //Condiciona vitória de X pela psegunda diagonal
        system("CLS");
        printf("Vitoria do jogador X\n\n");                //Printa a vitória de X
        printarM();
        v1=1;                                            //Vitória de X
    }
    if(somaDiag==-3){                                    //Condiciona vitória de O pela segunda diagonal
        system("CLS");
        printf("Vitoria do jogador O\n\n");                //Printa a vitória de O
        printarM();
        v2=1;                                            //Vitória de O
    }
}
```

Função Vitória Diagonal

```
void vitoriaDiag(int somaDiag){
    somaDiag=0;
    somaDiag = v[0][0] + v[1][1] + v[2][2];
    if(somaDiag==3){
        system("CLS");
        printf("Vitoria do jogador X\n\n");
        printarM();
        v1=1;
    }
    if(somaDiag==-3){
        system("CLS");
        printf("Vitoria do jogador O\n\n");
        printarM();
        v2=1;
    }
    somaDiag=0;
    somaDiag=v[2][0]+v[1][1]+v[0][2];
    if(somaDiag==3){
        system("CLS");
        printf("Vitoria do jogador X\n\n");
        printarM();
        v1=1;
    }
    if(somaDiag==-3){
        system("CLS");
        printf("Vitoria do jogador O\n\n");
        printarM();
        v2=1;
    }
}
```



Disponível em https://github.com/neginjhonx1/Projeto-Jogo-da-Velha

Obrigado pela atenção!!!