**Laboratório de Introdução a Ciência da Computação**

**Jogo da Velha**

**1- Orientações sobre o código:**

Função Win\_condition:

A função **“win\_condition”** é do tipo bool, o que justifica a necessidade do **“#include<stdbool.h>”**. Essa função considera como parâmetro uma matriz 3x3 e retorna **TRUE** quando algum jogador vence a partida. Para isso, são realizados quatro testes sequenciais:

1) Algum jogador colocou três símbolos idênticos horizontalmente?

2) Algum jogador colocou três símbolos iguais verticalmente?

Logicamente, essas questões podem parecer simples, mas como a matriz é inicializada como vazia, então teoricamente a pergunta sempre retornaria **TRUE**. Nesse sentido, colocamos uma restrição para que nenhum dos testes considere o elemento “espaço” da matriz.

3) Algum jogador colocou três símbolos idênticos na primeira diagonal?

Para responder essa pergunta, primeiro devemos imaginar a matriz:

**| [a11] [a12] [a13] |**

**| [a21] [a22] [a23] |**

**| [a31] [a32] [a33] |**

Diferente dos outros dois testes, não é trivial escanear cada diagonal. Para isso, definimos uma string chamada **“Matrixline”**, da forma:

**Matrixline = a11 a12 a13 a21 a22 a23 a31 a32 a33.**

Ou seja, um vetor unidimensional que copiou a matriz original. Considerando que o primeiro elemento é o **“matrixline[0] = a11**”, o próximo elemento da diagonal seria o **“matrixline[4] = a22”** e por fim o **“matrixline[8] = a33”**.

4) Algum jogador colocou três símbolos idênticos na segunda diagonal?

De forma semelhante ao teste 3, usaremos a **“Matrixline”** para facilitar a programação. No entanto, o primeiro elemento observado deve ser o **“matrixline[2] = a13”**, seguido do **”matrixline[4] = a22”** e do **“matrixline[6] = a31”**. Caso nenhum dos testes seja satisfeito, a função retornará **FALSE**.

Testes condicionais (IF):

1) Verifica se os valores declarados são válidos, ou seja, verifica se o jogador escolheu uma linha e/ou coluna entre 1 e 3.

2)Verifica se os valores escolhidos pelo jogador correspondem a um “espaço” disponível ou ocupado.

OBS: Caso os valores sejam inválidos ou o “espaço” escolhido esteja “ocupado”, o programa imprime a mensagem de erro e solicita os valores novamente, sem prejuízo na sua execução.

**2- Orientações para a execução do programa:**

Dentro do **“main()”**, o programa inicializa a matriz que será usada como tabuleiro. Inicialmente, essa matriz está **“vazia”**,ou seja, os seus elementos são iguais a **“espaço”**.

Em seguida, o usuário precisa escolher uma das opções do menu inicial, sendo elas:

**1) Iniciar Partida:**No comando iniciar partida, um jogador comandará o “X” e o outro será o “O”. O primeiro jogador será o “X” e o segundo o “O”. Para “reconhecer” o turno de cada jogador, duas variáveis inteiras foram declaradas, **“turnoX”** e **“turnoO”**, sendo a primeiro inicializada com valor 1 e a segunda com valor 0. No final de cada turno, a variável referente ao turno será zerada, enquanto a outra receberá o valor 1.

**2) Ajuda:** No comando de ajuda, há orientações e curiosidades sobre o **“Jogo da Velha”**, de modo a aprofundar a experiência durante a sua execução.

**3) Encerrar o jogo:** De maneira intuitiva, esse comando encerra a execução do programa com **“return 0”** e com uma mensagem de despedida.

**Fluxograma**

Início

Escolher 1, 2 ou 3

3(Encerrar o jogo)

1(Iniciar uma partida)

2(Ajuda)

Imprime informações sobre o jogo da velha

Fim

Imprime “Adeus”

Imprime a matriz

Jogador X

Jogador O

Digite o valor da linha e da coluna desejadas

Alguém ganhou na horizontal, vertical, diagonal ou deu velha?

Não

Sim

Os valores são válidos?

Digite o valor da linha e da coluna desejadas

Sim

Não

Verificação

Trabalho de SSC0601 - Jogo da Velha.

Grupo: Pedro Henrique de Holanda Lousada, Roger Daiki Hata Sakai e Luiz Henrique Nogueira dos Santos.

Link para o git: <https://github.com/Saravoja/Jogo-Da-Velha> (O repositório foi configurado como público)

Link para o vídeo: <https://drive.google.com/drive/folders/1bzQIhDI06tO37E4s9XpmIY9feluXLC_2?usp=sharing>

A **autoavaliação** está registrada dentro do git, juntamente com a documentação de funcionamento do código, de qualquer forma, ela também será reproduzida aqui:

Pedro: 7/10

"Originalmente, o programa deveria conter não só uma opção de verificar o histórico de partidas, utilizando arquivo e ponteiros, como também contaria com uma espécie de "modo competitivo" onde os jogadores pontuariam a cada sequência numa matriz de tamanho variável. No entanto, optamos por um programa mais simples e estável, de fácil compreensão e de alta portabilidade. Dessa forma, nossa auto-avaliação, levando em conta a grandiosidade da ideia original, define o programa como "regular-bom", atendendo suficiente bem a demanda, embora não seja incrível em nenhum aspecto”.

Luiz: 8,5/10

"O programa funciona e cumpre o que é proposto. Apesar disso, é válido citar que ele pode ser melhorado e aperfeiçoado. Diante disso, considerando a estabilidade do conjunto da obra, ele é bom."

Roger: 8,5/10

"Primeiramente, o trabalho funcionou como esperado, não apresentando erros em seu funcionamento e sendo bem intuitivo para os usuários jogarem.

Porém, a ideia inicial era implementar um mecanismo de pontuação, o qual tornaria o jogo mais competitivo e completo, mas decidimos em não adicionar, por apresentar possíveis falhas em seu funcionamento e deixando o usuário frustrado enquanto joga."