

Relatório Técnico:

Aluna: Maria Mariana Varela Cavalcanti Souto

Matrícula: 20250033894

Período: 2025.2

Nome do projeto: Jogo da velha com IA básica

1. Introdução e Contexto

1.1. Objetivo

O objetivo foi criar um jogo da velha para colocar em prática os conceitos que aprendi em aula, que foram: estruturas de repetição, condicionais, vetores e o uso de funções para organizar o código.

1.2. Problema que o Projeto Resolve

fazer um jogo da velha onde seja possível jogar contra o computador, criando uma IA simples para ser o oponente.

1.3. Justificativa da Escolha do Projeto

Escolhi o jogo da velha porque as regras são fáceis e todo mundo conhece, o que facilita na hora de programar a lógica. Também foi uma escolha mais prática para mim devido as minhas demandas pessoais de trabalho e estudo.

2. Análise Técnica

2.1. Metodologia

Para desenvolver o projeto, usei programação estruturada. A ideia foi quebrar o código em várias funções para uma organização melhor do código: uma função pra iniciar o tabuleiro, outra pra imprimir, outra pra checar o vencedor e assim por diante. As ferramentas que usei foram:

Linguagem: C

Compilador: GCC, para transformar o código em um programa executável.

Ambiente: Google Colab, por já ter um ambiente Linux com terminal pra compilar e rodar o código e pela praticidade de ser em nuvem e eu poder programar de qualquer lugar (tive essa necessidade já que nem sempre estava com o meu computador por perto)

Editor de Texto: Usei o próprio editor do Colab.

2.2. Aplicação dos Conceitos da U1

Estruturas Condicionais (if/else): Foram usadas para tomar as decisões do jogo, desde validar se a jogada do usuário era dentro ou fora do tabuleiro, até checar a cada rodada se alguém tinha vencido e verificar se deu empate quando o tabuleiro encheu.

Estruturas de Repetição (for/while): O for foi útil para "varrer" o tabuleiro, limpando no começo ou para imprimir na tela. O while foi o loop principal, que mantinha o jogo rodando enquanto não houvesse um vencedor ou empate. O do-while foi perfeito para a jogada do usuário, pois ele ficava insistindo até que uma jogada válida fosse digitada. Também usei na IA para ela sortear uma posição até encontrar uma casa vazia.

Vetores (arrays bidimensionais): O tabuleiro do jogo nada mais é do que um array bidimensional `char tabuleiro[3][3]`. Essa estrutura foi ideal porque permite acessar cada casa usando coordenadas de linha e coluna, o que simplificou a lógica de verificação das jogadas.

2.3. Organização das Funções

O código tem 7 funções:

`inicializarTabuleiro()`: Deixa o tabuleiro limpo para começar.

`imprimirTabuleiro()`: Mostra o tabuleiro atualizado na tela.

`verificarVencedor()`: checa se alguém ganhou.

`verificarEmpate()`: Verifica se o jogo empatou.

`jogadaHumano()`: armazena a jogada da pessoa.

`jogadaIA()`: Implementa a jogada aleatória do computador.

`main()`: É a função que chama todas as outras na ordem certa para fazer o jogo acontecer.

3. Implementação e Reflexão

3.1. Dificuldades Encontradas

Tive dificuldade de encontrar em que ambiente eu iria programar, já que eu precisava de um ambiente que me permitisse programar de diferentes dispositivos, já que nem sempre estou com o meu computador. Outro ponto que deu um pouco de trabalho foi a função `scanf`, que às vezes se comportava de forma estranha se digitasse algo diferente do esperado na entrada da posição das jogadas.

3.2. Soluções Implementadas

Para resolver o problema do ambiente de programação, usei o google colab, baixei o GCC e usei o comando `%%writefile` para criar o arquivo `.c` e o `!` para rodar os comandos de compilação e execução no terminal. Já a validação da jogada do usuário foi resolvida com um loop `do-while` e `uns ifs` para garantir que a posição era válida antes de continuar o jogo.

3.3. Organização do Código

Decidi separar tudo em funções, o que deixou o código bem mais limpo e fácil de entender o que cada parte fazia. O uso de variáveis globais para o tabuleiro e para os símbolos dos jogadores também simplificou o acesso a esses dados em todas as funções.

4. Conclusão

Desenvolver esse projeto foi importante para ver como variáveis, loops e condicionais podem criar um programa interativo. Aprendi bastante sobre a manipulação de arrays e a criar lógicas de verificação para o resultado do jogo.

4.1 Possíveis Melhorias

- Fazer com que a IA jogue com estratégia em vez de forma aleatória
- Criar uma interface, para deixar o jogo mais bonito do que só texto no console.
- Adicionar um modo para dois jogadores humanos.