



Programação | Trabalho Prático

Ultimate Tic-Tac-Toe

Maria Beatriz Girão Abreu | 2020130937 | LEI-CE

domingo, 19 de junho de 2022

Índice

Introdução.....	3
Organização do Programa.....	4
Estruturas de Dados	5
Estrutura jogadas	5
Estrutura jogadores.....	5
Estruturas dinâmicas.....	6
Array dinâmico	6
Lista Ligada	6
Opções de Implementação	7
Regras.....	7
Jogada Inicial	7
Mini Tabuleiro Terminado.....	7
Gravação do Jogo no Ficheiro Binário.....	7

Introdução

Este trabalho foi realizado no âmbito da disciplina de Programação do curso de Engenharia Informática, do Instituto Superior de Engenharia de Coimbra.

O projeto visa a construção e organização de um jogo de tabuleiro, no caso, *Ultimate Tic-Tac-Toe*, através da linguagem C e utilizando os conteúdos abordados ao longo do semestre.

Este relatório tem como função explicar algumas funcionalidades do jogo, bem como o código implementado.

Organização do Programa

O código foi implementado no *CLion*, e está organizado em vários ficheiros:

- *menus.c/.h* – menu principal e menu que aparece no decorrer do jogo, disponibilizando opções para os jogadores;
- *tabuleiro.c/.h* – criação do tabuleiro e mini tabuleiros e demonstração dos mesmos na execução do programa;
- *jogo.c/.h* – está responsável pela gestão de todo o jogo, nomeadamente efetuar jogadas e respetivas verificações;
- *guardaJogadas.c/.h* – permite adicionar e aceder às jogadas efetuadas, através de uma lista ligada;
- *interromperJogo.c/.h* – como o nome indica, possibilita a interrupção do jogo e a sua retoma, posteriormente;
- *utils.c/.h* – funções que visam facilitar algumas implementações, por exemplo, gerar números aleatoriamente.

Estruturas de Dados

Durante a realização do trabalho, existiu a necessidade de criar algumas estruturas de dados, de modo a organizar algumas partes do mesmo. Assim, foram criadas duas estruturas.

Estrutura jogadas

Esta estrutura tem como objetivo guardar os dados relativos às jogadas que cada jogador faz durante o jogo, nomeadamente o mini tabuleiro em que joga e respetivas linha e coluna, o jogador, e o número da jogada.

A estrutura conta, então, com as respetivas variáveis, todas do tipo inteiro: *nMiniTab*, *linha*, *coluna*, *jogador* e *numJogada*.

Existe, ainda, um ponteiro para esta estrutura, pois vai ser usada na criação de uma lista ligada.

Estrutura jogadores

A estrutura jogadores guarda os dados dos jogadores, isto é, a identificação (jogador 1 ou 2) e a respetiva peça para colocar no tabuleiro (X ou O). Estes dados são representados pelas variáveis *numJogador* (inteiro) e *peca* (caractere), respetivamente.

Estruturas dinâmicas

Array Dinâmico

Para a criação do tabuleiro do jogo foi criado um *array* dinâmico único. Em cada posição, este *array* terá 9 mini tabuleiros que são, por sua vez, *arrays* dinâmicos bidimensionais.

Inicialmente é alocado espaço para os mini tabuleiros, de acordo com a quantidade existente (no caso são 9), e é feita uma verificação dessa alocação. Depois, para cada mini tabuleiro (*arrays* bidimensionais) é alocado espaço para as linhas. Faz-se, também, uma verificação, que caso corra mal, a tabela é libertada. Seguidamente, em cada linha do mini tabuleiro, foi feita uma alocação de memória, criando espaço para as colunas. Mais uma vez é feita uma verificação, e, se necessário o tabuleiro será libertado. Por fim, inicializa-se cada coluna do mini tabuleiro com um caractere.

Lista Ligada

Como referido na explicação da estrutura de dados “jogadas”, esta tem um ponteiro que aponta para o mesmo tipo de dados. A lista é constituída por nós, que ao guardarem a posição do próximo, permitem um acesso sequencial das jogadas que vão ser feitas.

Para a criação de cada nó da lista foi usada a função *adicionaJogada*, que os insere sempre na última posição.

Primeiro foram criados dois ponteiros, um para inserir um novo nó na lista e outro para auxiliar na manipulação da mesma. Depois é feita a alocação de memória e respetiva verificação.

Seguidamente cria um nó, guardando os dados fornecidos nos parâmetros da função nas respetivas variáveis da lista.

Para inserir o novo nó, primeiro verifica-se se a lista está vazia, e, se estiver, este é adicionado na primeira posição. Se não estiver vazia, a lista é percorrida através da variável auxiliar, ou seja, enquanto esta não encontrar o último nó, passa para o próximo. Quando encontrar, adiciona o novo nó criado anteriormente.

Opções de Implementação

Regras

De acordo com enunciado, algumas regras ficaram ao critério do aluno. Assim, foram criadas regras para a jogada inicial e para a jogada que calha num mini tabuleiro terminado.

Jogada Inicial

O jogador que faz a primeira jogada é escolhido aleatoriamente. A função *iniciaJogo* recebe num dos parâmetros a variável *jogadorAtual*, que é inicializada na *main()* aleatoriamente com 0 ou 1, correspondendo posteriormente ao jogador 1 ou ao jogador 2.

A escolha do mini tabuleiro é feita através da função *escolheMiniTab*, que recebe por parâmetro um ponteiro para a lista ligada das jogadas. Verifica se está vazia, e se estiver pede ao jogador que escolha um mini tabuleiro.

Mini Tabuleiro Terminado

Quando um mini tabuleiro for terminado, haverá uma verificação que impedirá que ocorra alguma jogada no mesmo. A jogada será então efetuada no mini tabuleiro do jogador anterior. Por exemplo, se o jogador 1 jogar no primeiro mini tabuleiro e a jogada que ele efetuar mandar o jogador 2 para um mini tabuleiro terminado, o jogador 2 terá de jogar, também, no primeiro mini tabuleiro.

Esta verificação é efetuada pela função *fazJogada* e o mini tabuleiro a jogar é retornado pela função *procuraJogadaAnterior*. Nesta função a lista vai ser percorrida pela variável atual até ser encontrado o último nó (a jogada atual).

Gravação do Jogo no Ficheiro Binário

A função *interrompeJogo* guarda os dados das jogadas todas efetuadas até ao momento. Para isso é criado um ficheiro, onde se escreve a informação através da função *fwrite*.

Na função *retomaJogo* é alocada memória para ler os dados da lista ligada para o ficheiro binário, através da função *fread*.

Quando o jogador retoma o jogo, é apresentado também o tabuleiro anterior. Conforme o enunciado, a função não lê o tabuleiro do ficheiro. Primeiro, é criado um tabuleiro vazio. Depois, através da função *setPos*, são colocadas as jogadas feitas no tabuleiro segundo o histórico de dados guardados no ficheiro.