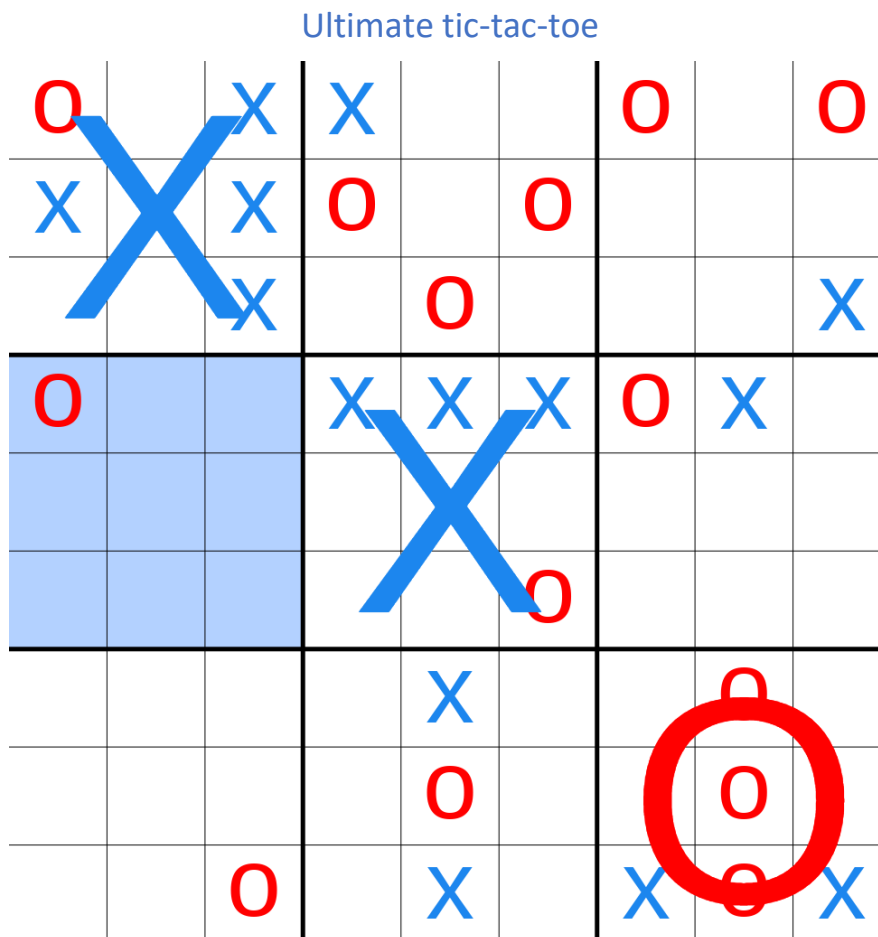


# RELATÓRIO DO TRABALHO PRÁTICO



## Índice

|                                     |   |
|-------------------------------------|---|
| Proposta de Trabalho .....          | 2 |
| Ferramentas usadas (IDE) .....      | 2 |
| Bibliotecas usadas .....            | 2 |
| Bibliotecas nativas C .....         | 2 |
| Biblioteca file.h.....              | 3 |
| Biblioteca game.h.....              | 3 |
| Biblioteca matdic.h e utils.h ..... | 3 |
| Explicação do código main.c .....   | 4 |

## Proposta de trabalho

Neste trabalho prático foi proposto fazer o jogo *Ultimate tic-tac-toe* em c. O jogo consiste num jogo do galo normal, mas com 9 jogos do galo em cada espaço num jogo do galo grande. A regra é o jogador joga no jogo do galo corresponde à localização da peça colocada pelo outro jogador no jogo do galo (ex.: No jogo do galo superior esquerdo jogou no centro, a próxima jogada será feita no jogo do galo do centro). Se o jogador ganhar um jogo do galo, esse jogo do galo é de e se fazer uma linha de jogos de galo ganhos, ele ganha o jogo.

## Ferramentas usadas

Este trabalho foi desenvolvido no IDE *Visual Studio Code* e uso de extensões para facilitar o uso (*Hex Editor* para ver o ficheiro .bin para verificar se tava a escrever as informações certas, *Code Runner* para compilar e correr o programar sem precisar de digitar os comandos necessários manualmente)

## Bibliotecas usadas

### Bibliotecas nativas do C

Foi utilizados as bibliotecas *stdio.h*, *stdlib.h*, *string.h* e *ctype.h*.

*Stdio.h* – biblioteca de input e output na consola

*Stdlib.h* – biblioteca para a função *malloc()* para alocar informação na memoria

*String.h* – biblioteca para manipular as strings com facilidade

*Ctype.h* – bliboteca usada para tirar o caps das strings

## **Biblioteca file.h**

Foi criada com o propósito de gerir a parte de ficheiros para escrever e ler.

saveFile() – função para escrever o jogo num ficheiro .bin

loadFile() – função de carrega a informação do ficheiro .bin para continuar o jogo

saveFileExists() – função para verificar se o ficheiro .bin exist

saveDelet() – função para apagar ficheiro .bin

## **Biblioteca game.h**

Foi criada com o propósito de gerir a parte do jogo.

addPlay() – adiciona a joga na lista de jogadas

fillItem() – preenche o novo item à lista de jogadas com as informações da jogada acabada de ser feita

showList() – função que mostra a listra de jogadas feitas

checkWin() – função que verifica se a matriz foi conquistada por um jogador

coorTrans() – função que transforma o valor de 1 a 9 em coordenadas (x,y)

nullListCheck() – função que verifica se a lista está vazia

startGamer() – função jogo

## **Biblioteca matdic.h e utils.h**

Biblioteca fornecida para fazer o trabalho

## Explicação do código do main.c

Começa por importar todas as bibliotecas citadas antes, depois declara as variáveis necessárias:

Selection – para selecionar a opção do menu

Mode – para definir qual modo de jogo (0 corresponde a Player vs Player e 1 corresponde a Player vs Machine)

Choose – para ficar dentro do loop caso não seja escolhido algo fora das hipóteses.

Save – para definir se é para ler o ficheiro .bin ou não

Perto do fim do código main é invocada a função startGamer() em que começa o jogo. Começa por declarar as variáveis necessária e por defini-las, initRandom() para a função que gera números funcionar, o ponteiro lista da struct de lista de jogadas NULL, selecionar o jogador aleatoriamente, criar a matriz corresponde ao tabuleiro global, uma array dinâmico de struct de mini tabuleiros com a matriz, a numeração e valor que corresponde ao jogador que ganhou naquele tabuleiro, criar uma matriz guia para ajudar nas jogadas caso de dúvida, e se for um jogo salvo vai abrir e importar os seus valores. Se for um jogo antigo, mostra o tabuleiro que param, se não, mostra o tabuleiro global para escolher em que tabuleiro começar o jogo. Depois pede um comando variando de um caracter até um número de 1 a 9. Quando é introduzido um número valida se é possível fazer tal jogada, e se é valido verifica se ganhou nesse tabuleiro caso o valor desse tabuleiro seja de ninguém. Depois mostra o próximo tabuleiro correspondente à posição da peça jogada. Se ganhar o tabuleiro mostra que ganhou esse tabuleiro verifica se ganhou no tabuleiro global. Se sim mostra que o jogador ganhou no jogo, se não continua o jogo até ou ganhar ou até sair, com ou sem save desse jogo.