Relatório

Jorge Ricardo Marques Duarte – a2021110042

Engenharia Informática

Trabalho prático

Cadeira de Programação

# Introdução

Este relatório veio no âmbito da cadeira de Programação contemplando a descrição genérica da organização do programa, a identificação do ambiente de desenvolvimento utilizado durante a implementação, a apresentação de estruturas dinâmicas implementadas e da organização dos ficheiros utilizados pelo programa, escolhas de opções tomadas em termos de implementação e especificação definida para as regras de jogo.

O programa foi implementado respeitando a norma C99, contendo uma interface simples e amigável, apresentado ao utilizador o contexto da jogada e tipos de interações que pode realizar.

# Ambiente de desenvolvimento

O programa foi desenvolvido no IDE CLion, software com suporte para as linguagens de programação C e C++, tendo vários tipos de funcionalidades como por exemplo código *boilerplate*, assistência de código, possibilidade de *refactor* de ocorrências especificas, boa documentação, entre outro tipo de funcionalidades.

Foi criado um repositório privado no Github, permitindo o armazenamento e progresso de desenvolvimento do jogo através de *commits* e *pushs para a branch* principal, demonstrando ser muito útil na parte de organização de gestão de código.

# Especificação de regras adicionais do jogo

A jogada inicial foi definida que é sempre dada pelo jogador 1 sendo que é requerido ao jogador qual mini tabuleiro deseja realizar a jogada e a sua posição.

No caso de um oponente ter efetuado uma jogada em uma posição (index) e o mini tabuleiro representativo do index estar finalizado, é permitido ao jogador escolher o tabuleiro e a posição que deseja colocar a sua peça.

Independentemente de vitória ou empate por parte dos jogadores no final do jogo, é sempre pedido aos mesmos o nome para o ficheiro que irá conter todas as jogadas realizadas durante o jogo.

# Organização do programa

## Estrutura de pasta e ficheiros

O programa foi dividido em vários ficheiros de modo a melhorar a organização de código. Dentro da pasta do programa foi criada uma subpasta game onde está presente os vários modos de jogo, *HumanVsBot* e *HumanVsHuman*. Além disto, existe ficheiros como o player.h e player.c que contem todas as ações que um jogador pode realizar, um ficheiro gameBoard.h e gameboard.c que contem todas as funções associadas aos tabuleiros mini tabuleiros e ao jogo e por último um ficheiro utils.h e utils.c, que contem algumas funções utilitárias utilizadas por outros ficheiros no decorrer do programa.

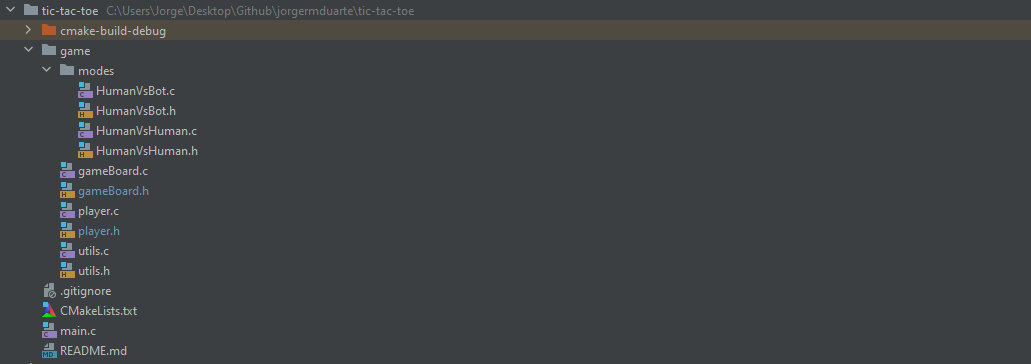


Figura - Estrutura de pastas e ficheiros da aplicação

# Estruturas da aplicação

A aplicação contém uma estrutura principal chamada *Game*. Nesta estrutura está presente o total de jogadas, um *array* contendo com todos os mini tabuleiros do jogo, ponteiros para os jogadores presentes durante o jogo, um ponteiro para o utilizador que está a efetuar uma jogada no momento, o tipo de modo de jogo, uma variável booleana representativa se o jogo já acabou, e uma lista ligada com todas as jogadas efetuadas durante o jogo.



Figura - Estrutura principal do programa

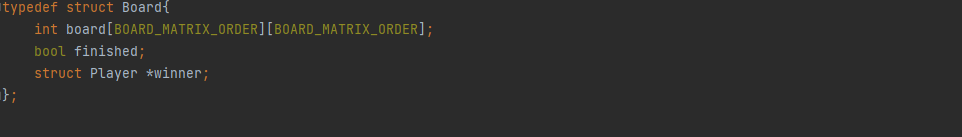
Como podemos observar na lista de mini tabuleiros, este tem uma estrutura especifica sendo o Board. Este Board contem as informações especificas do mini tabuleiro, como a verificação se este está finalizado, o ponteiro para o vencedor do mesmo e uma matriz bidimensional de valores inteiros, sendo que cada valor em si representa o determinado símbolo do utilizador.

Figura - Estrutura de um mini tabuleiro

A estrutura Player, contem o nome do utilizador, o total de vitorias, o total de derrotas, uma variável booleana representativa se o jogador é um Bot, ou seja, um computador e o seu respetivo símbolo associado. Este símbolo é um enumerador, sendo que 0 significa NULL, ou seja, é o predefinido quando o mesmo não tem nenhum atribuído, 1 sendo o “0” ou bola, 2 sendo o “X” e o 3 sendo “D” de empate.

Figura - Estrutura de um jogador e o seu respetivo símbolo

Na estrutura Game, temos por sua vez então a estrutura UserPlay que é a lista ligada com todas as jogadas efetuadas durante o jogo. Esta lista ligada contém o número da jogada, o número do mini tabuleiro onde foi efetuada a jogada, o index representativo no mini tabuleiro, se a jogada fez com que o jogador ganhe o mini tabuleiro, se a jogada fez com que o tabuleiro ficasse finalizado, o ponteiro para o jogador que efetuou a jogada, o ponteiro para a próxima jogada e por último o ponteiro para a jogada anterior.



Figura - Estrutura da lista ligada com todas jogadas efetuadas durante o jogo

Um dos requisitos presentes para a realização deste trabalho prático foi a necessidade de guardar o estado atual do jogo após cada uma das jogadas, permitindo aos jogadores retomarem o jogo em qualquer momento. Com esta necessidade, optei por criar estruturas para este efeito muito parecidas com as estruturas atuais, mas agora sem ponteiros. Isto permitiu guardar todas as informações necessárias para que isto fosse possível.

Uma imagem com texto

Descrição gerada automaticamente

Figura - Estruturas utilizadas para guardar o estado do jogo

# Especificações do programa

## Interfaces Visuais e funcionalidades

A interação visual por defeito da aplicação é a apresentação do modo de jogo desejado.

Figura - Interface visual principal

Com a seleção de modo do jogo é nos requerido a introdução dos nomes dos jogadores, no caso do jogador contra computador, só é pedido o nome do jogador.

Uma imagem com texto

Descrição gerada automaticamenteApós a introdução dos nomes, é nos apresentado um estado global do jogo, os jogadores, todos os mini tabuleiros, qual o jogador que está a fazer a primeira jogada, e é perguntado em qual mini tabuleiro ele quer jogar sendo a primeira jogada.

Figura - Apresentação dos mini tabuleiros

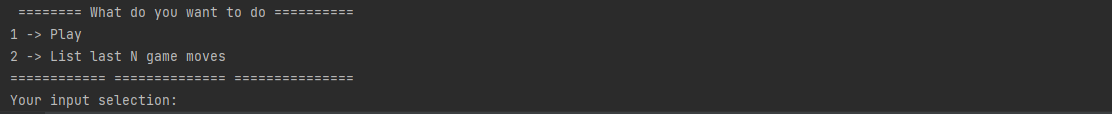
Após a seleção do primeiro mini tabuleiro o jogo entra num estado de loop até ser finalizado, dando a possibilidade ao jogador de jogar ou mostrar determinado número de jogadas efetuadas durante o jogo.

Figura - Possibilidade jogada ou apresentação de jogadas

Uma imagem com texto

Descrição gerada automaticamenteCaso o utilizador escolha jogar é requerido o número do index que este quer efetuar a jogada e apresentado o estado de jogo novamente, mas agora além de todos os mini tabuleiros, é nos apresentado também o estado global do jogo, representativo de qual jogador ganhou ou empatou determinado mini tabuleiro. Além disto, após cada jogada, esta é guardada numa lista ligada e por sua vez utilizada para guardar o estado de jogo num ficheiro binário.

Figura - Apresentação do estado global do jogo

Caso o utilizador escolha para demonstrar as respetivas jogadas, é pedido quantas jogadas este quer observar, e são apresentadas de forma descendente, ou seja, da mais recente para a mais antiga.

Figura - Lista de jogadas durante o jogo

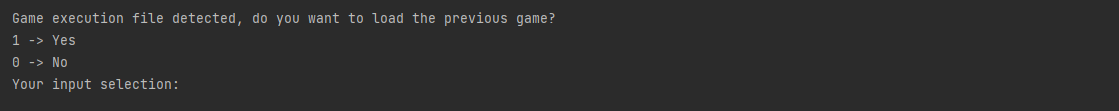
No caso em que algum dos jogadores tem de sair e se quiserem retornar o jogo se quiserem, é possível sendo que após cada jogada o estado é guardado. Isto permite aos jogadores voltarem ao estado de jogo onde tinham ficado.

Figura - Carregamento do estado de jogo

Uma imagem com texto

Descrição gerada automaticamenteNo caso de uma jogada em um index representativo de um mini tabuleiro que já foi finalizado, é perguntado ao jogador qual tabuleiro este gostaria de efetuar a sua jogada além da posição da mesma.

Figura - Jogada em um novo tabuleiro

Após a conclusão de vários mini tabuleiros, assim que o programa detete a vitória por parte de um jogador, é pedido na interface visual o nome do ficheiro onde desejam que todas as jogadas sejam guardadas. Estas jogadas contém as informações da lista ligada apresentando num ficheiro de texto, o número da jogada, por qual dos jogadores é que este Uma imagem com texto

Descrição gerada automaticamenteUma imagem com texto

Descrição gerada automaticamentefoi efetuada, o mini tabuleiro a posição no mesmo.

Figura - Apresentação de nome do ficheiro para conclusão final do jogo

Figura - Pedido de introdução de nome do ficheiro para conclusão do jogo

Uma imagem com texto

Descrição gerada automaticamente

Figura - Ficheiro de texto com todas as jogadas efetuadas

# Conclusão

Este trabalho foi interessante ao ponto de aplicar os conceitos aprendidos nas aulas teóricas, com apresentações práticas de exercícios ao longo do semestre. Na minha opinião o mais importante foi o conceito de ponteiros e aplicação das listas ligadas que nos levará para os próximos passos como Binary Trees, Graphs, Stacks, Queues entras outros tipos de estruturas de dados existentes e com diversos tipos de aplicações.