

Programação Orientada a Objetos

ISEC – Instituto Superior de Engenharia de Coimbra

**Trabalho prático I – Meta 1**

2021110042 - Jorge Ricardo Marques Duarte

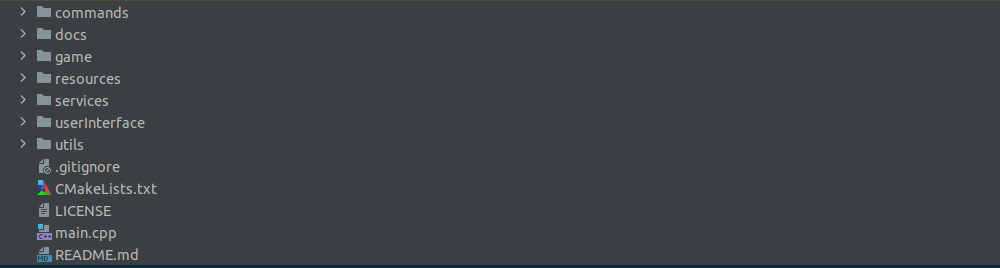
2021146383 - João Alexandre Caravela Marques

**Introdução**

Neste trabalho prático pretende-se aplicar os conhecimentos adquiridos durante as aulas teóricas e práticas da disciplina construindo em C++ um simulador de uma reserva natural povoada por diversos animais.

Este relatório vem com o intuito de demonstrar as diversas decisões tomadas ao longo do desenvolvimento, tanto a nível de estruturas usadas, assim como organização e estrutura da aplicação.

**Estrutura de pastas**



Na pasta *commands*, temos implementada toda a lógica relacionada a comandos e validações.

Na pasta *docs*, temos documentos e objetivos do projeto, estando estes seperados por metas.

Na pasta *game* é onde temos todos os modelos utilizados pela aplicação, desde tipos de comida e animais a classes utilizadas como estrutura base para a interface visual.

Na pasta *services* é onde contemos a base da nossa aplicação, sendo no ficheiro GameService.cpp que definimos qual o tamanho da nossa reserva natural, assim como a leitura do ficheiro de constantes e também a inicialização para a receção dos comandos por parte do utilizador.

Na pasta *userInterface*, está contida a lógica de apresentação visual do nosso jogo, isto é, desde a nível de informação sobre os animais, assim como a apresentação da reserva natural com todas as entidades existentes.

Por último, a pasta *utils* apenas contem funções utilitárias que podem ser reutilizadas por outras classes ou funções.

**Classes principais da aplicação**

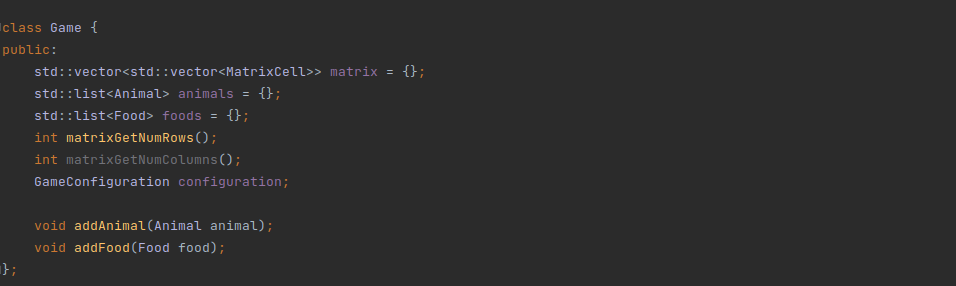
Como descrito na nossa estruturas de pastas, a nossa pasta principal é a services, contendo toda a lógica principal da aplicação. Nesta pasta contemos a classe GameService contendo classe Game e funções principais privadas para definição de critérios do jogo.

Observe a imagem seguinte referente a classe GameService:



A estrutura Game como podemos visualizar está presente dentro do nosso GameService. Esta estrutura contem todas as informações referentes a reserva natural, como a matriz sendo um vetor de vetores representando a nossa reserva natural, uma lista da classe Animal, outra lista da classe Food e uma classe referente a configurações do jogo.

Observe a imagem seguinte referente a classe Game:



Outra classe importante que contemos na nossa aplicação é a classe MatrixCell, que representa a posição na matriz assim como contem a lista de animais e comidas presentes na célula de jogo.

O objetivo desta listagem de animais e comidas é ser referente a lista de animais presente na lista de animais e comidas na classe Game, quando um animal se mover, assim tbm se moverá o animal na matriz, então o que pensamos em implementar foi retirar a entidade da MatrixCell atual e move-lo para outra MatrixCell referente a posição desejada.

Observe a imagem seguinte referente a classe MatrixCell:



Ainda dentro do Game, podemos observar a classe GameConfiguration, sendo a classe responsável por tudo que tenha a ver com configurações do nosso jogo. A sua estrutura contem o tamanho da nossa reserva natural, assim como uma posição de visualização do ecrã. Definimos como limite de área visivel 80% do tamanho do jogo quando criado numa variavel global constante. Nesta classe é possivel modificar a área de jogo visivel através da função moveScreenDisplayPosition.

Observe a imagem seguinte referente a classe GameConfiguration:

