

INSTITUTO SUPERIOR DE ENGENHARIA DE COIMBRA

INSTITUTO POLITÉCNICO DE COIMBRA

Licenciatura em Engenharia Informática 2º Ano – 1º Semestre 2021/2022

Natural Reserve Simulator

Rafael Couto N° 2019142454

Rafaela Carvalho Nº 2019127935

COIMBRA

22 de novembro de 2021



Índice

ntrodução	3
Critérios de funcionamento	4
Classes	4
nterface	7
Decisões tomadas	8
anexos	9



Introdução

A elaboração deste trabalho prático visa consolidar conhecimentos em linguagem C++, explorados nas aulas teóricas e práticas, criando capacidade de desenvolvimento de aplicações nesta linguagem de programação.

Pretende-se criar um simulador, denominado *Natural Reserve Simulator*, destinado a gestão e desenvolvimento de uma reserva natural. Ao jogador será atribuída uma reserva que o mesmo deve gerir, provando-a e alimentando todos os seres vivos.

Deste modo, será implementada uma classe geral *Reserva* responsável por suportar todo o tipo de dados relativos a cada *célula* e à sua pormenorização.

Ora, cada *Reserva* tem *nLinhas* por *nColunas* e cada unidade será uma *cell* que é descrita por *animais* e *alimento*.

A interação com o jogo processa-se através de comando e ação por parte do jogador.



Critérios de funcionamento

O jogo *Natural Reserve Simulator* foi inicialmente desenvolvido na perspetiva de poupança de recursos e simplicidade de código, para tal, foi reduzido aos ficheiros e classes estritamente necessárias.

Classes

Foi criada uma classe *Cell* que representa cada unidade da *Reserva*. É uma classe descritora dos pormenores existentes em cada célula da reserva.

Foi criada uma classe *Alimento* que representa cada tipo de alimento que se encontra na *Reserva*. É uma classe que contém as características dos alimentos.

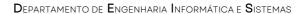


Foi criada uma classe *Animal* que representa cada tipo de animal que se encontra na *Reserva*. É uma classe que contém as características dos animais.

```
(class Animais {
  private:
    string const nome;
    string const nome;
    string const especie;
  int vida;
  int sauda;
  int sauda;
  int fone;
  float peso;
  public:
    Animais(string e,int v,float p,int s,int f=1):especie(e),vida(v),saude(s),fome(f),peso(p){}
  int getVida() const {return vida;}
  int getFame() const {return fome;}
  float getPeso() const {return peso;}
  string getMome() const{return nome;}
  string getMome() const{return nome;}
  void setVida(int vida);
  void setSaude(int saude);
  void setSaude(int saude);
  void setSaude(int saude);
  void setSaude(int fome);
  void setSeo(float peso);
  bool vivo();
  void diminuiFome();
  void diminuiF
```

Foi criada uma classe *Reserva* que é onde serão guardados os dados do jogo e é usado como intermediário no acesso à informação das *Cell* por parte da *Interface*.

```
class Reserva {
private:
   int NL,NC;
   int viewWindow = 5;
   int topLeftCornerX = 0, topLeftCornerY = 0;
   int id = 1;
   int simulatedTime{};
   vector<Reserva *> reservas; // Vector to store game states
   vector<Reserva *> reservas; // Vector to store the cells
public:
   Reserva(]=default; // Default constructor, applies predefined size
   Reserva(int Id=0):id(Id){}
   Reserva(Reserva *Reserva); // Copy constructor
   Reserva(int NL, int NC, string filename = ""); // Constructor with size and filename
   int getNL() const;
   int getNL() const;
   int getVolueWindow() const;
   int getTopLeftCornerX() const;
   void setTopLeftCornerX() const;
   void setTopLeftCornerX(int topLeftCornerX);
   int getTopLeftCornerY(int topLeftCornerY);
   int getTopLeftCornerY(int vopLeftCornerY);
   int getTopLeftCornerY(int vopLeftCornerY);
```





Foi criada um ficheiro header *utils* que contém todos os *includes* necessários para a execução do simulador.

#include <string>
#include <fstream>
#include <random>
#include <lostream>
#include <vector>
#include <vector>
#include <stream>
#include <ctime>
#include <ctidib>
#include <cstdio>
#include <ctdio>
#include <ctdio>
#include <ctdio>
#include <ctdio>
#include <inmanip>
@#include <unistd.h>



Interface

Criou-se uma classe *Interface*, onde é criado e tratado o ambiente do jogo e onde é tratado também da validação dos comandos que vão ser utilizados ao longo do simulador.





Decisões tomadas

Em relação à parte que visível no ecrã tomamos a decisão de mostrar um tamanho fixo, definido na classe *Reserva* com o nome *viewWindow*.

De modo que esta *viewWindow* seja independente de toda a visualização do tabuleiro de jogo foi criado uma variável *topLeftCorner* que permitirá fazer o cálculo de apresentação das células no intervalo desejado.

Tal como descrito no enunciado o número de linhas e colunas do tabuleiro de jogo é pedido ao utilizador em *runtime* no início da execução do programa.

Foi decidido que o jogo iniciaria sempre a sua área visual no canto superior esquerdo (0,0) e daí em diante é controlado através do utilizador.

Tomamos por iniciativa mais uma vez criar um comando clear, comando que irá ser destinado para limpar tudo o que já foi escrito no terminal para que se possa continuar a usar o simulador sem falta de informação por falta de espaço no terminal.



Programação Orientada a Objetos

2.ºA/1.ºS - ENG. INFORMÁTICA

Anexos

- Reserva.h
- Reserva.cpp
- Cell.h
- Cell.cpp
- Animais.h
- Animais.cpp
- Alimentos.h
- Alimentos.cpp
- Interface.h
- Interface.cpp
- Utils.h
- main.cpp
- constantes.txt
- commands.txt
- POO 2223 Relatório.pdf