

PROGRAMAÇÃO ORIENTADA A OBJETOS

Licenciatura em Engenharia Informática

2022/2023

META 1

Dinis Meireles de Sousa Falcão (P6) / Tiago Alexandre Pais Dias (P2)

2020130403 / 2019126344

a2020130403@isec.pt / a2019126344@isec.pt

ÍNDICE

OBJETIVO	2
INTERFACE	3
ESTRUTURAS DE DADOS	4
CONCLUSÃO	6

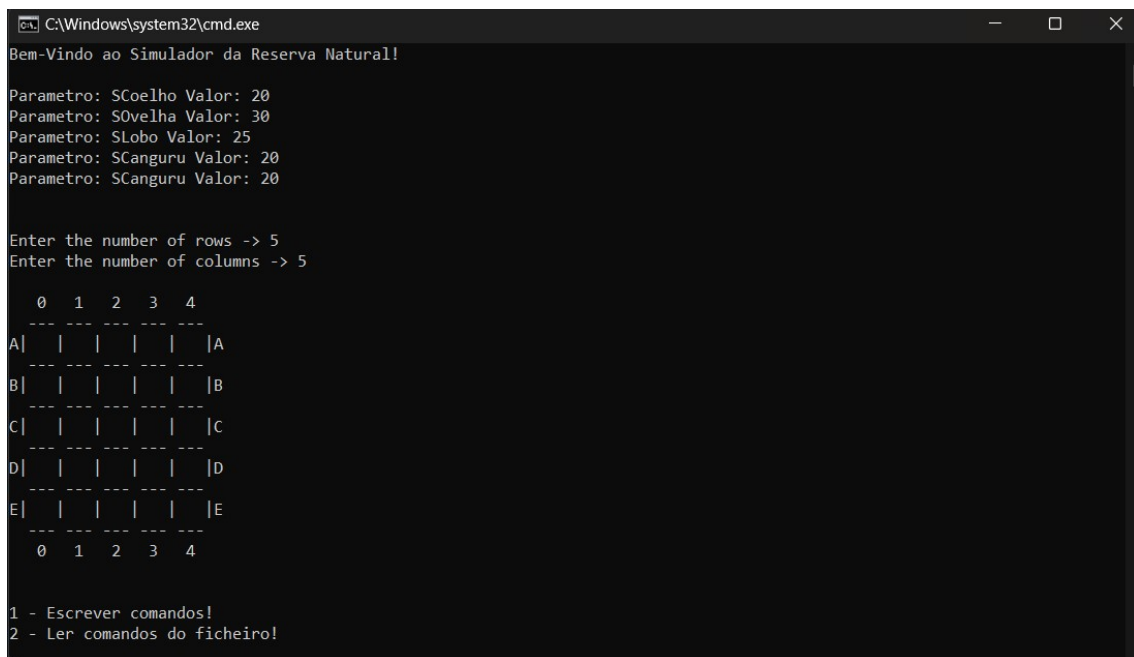
OBJETIVO

O objetivo deste trabalho prático consiste na implementação de um simulador de uma reserva natural desenvolvido em linguagem C++, povoada por diversos animais. Os animais têm comportamento autónomo e variado, deslocando-se pela reserva. O utilizador interage com o simulador, visualizando a reserva e o seu conteúdo, podendo especificar comandos que alteram o que acontece na reserva, afetando diversos aspetos da simulação e dos animais.

Foi fornecida uma biblioteca que permite posicionar o cursor em linha, coluna.

INTERFACE

Para a implementação da interface, inicialmente é apresentada uma mensagem de boas-vindas. De seguida, é feita a leitura e impressão dos valores do ficheiro “constantes.txt”. É apresentada uma tabela (grid), em que o utilizador toma a decisão do número de linhas e do número de colunas que são apresentadas no terminal. No fim desta, é feita a validação dos comandos através da colocação dos mesmos no terminal ou através da leitura dos valores do ficheiro “comandos.txt”.



```
C:\Windows\system32\cmd.exe
Bem-Vindo ao Simulador da Reserva Natural!

Parametro: SCoelho Valor: 20
Parametro: SOvelha Valor: 30
Parametro: SLoobo Valor: 25
Parametro: SCanguru Valor: 20
Parametro: SCanguru Valor: 20

Enter the number of rows -> 5
Enter the number of columns -> 5

  0  1  2  3  4
  ---
A|  |  |  |  |  |A
  ---
B|  |  |  |  |  |B
  ---
C|  |  |  |  |  |C
  ---
D|  |  |  |  |  |D
  ---
E|  |  |  |  |  |E
  ---
  0  1  2  3  4

1 - Escrever comandos!
2 - Ler comandos do ficheiro!
```

Figura 1 – Interface no terminal

ESTRUTURAS DE DADOS

Os ficheiros “animais.h”, “alimentos.h”, “interface.h” e “reserva.h” são responsáveis por armazenar as estruturas usadas para o funcionamento dos animais, dos alimentos, da interface e da reserva, respetivamente. Para esta meta foram implementadas as classes animais, alimentos, interface e reserva.

“animais.h”

Contém a class animais.

```
class animal{
    string especie;
    string nome;
    int id;
    int saude;
    int distpercepcao;
    int peso;
    int fome;
    int agressividade;
    int tempoVida;
    bool vida;
    int posX, posY;
public:
    animal(int id, string nome, string especie, int posX, int posY, bool vida, int tempoVida, int saude, int distpercepcao, int agressividade, int peso, int fome) :
        id(id), nome(nome), especie(especie), posX(posX), posY(posY), vida(vida), tempoVida(tempoVida), saude(saude), distpercepcao(distpercepcao), agressividade(agressividade), peso(peso), fome(fome) {}
    int getX() const;
    int getY() const;
};
```

Figura 2 – class animais

“alimentos.h”

Contém a class alimentos.

```
class alimentos{
    int duracao;
    int valorNutritivo;
    int toxicidade;
    bool consumido;
    vector<string> cheiros;
public:
    alimentos(int d, int vn, int t, bool c): duracao(d), valorNutritivo(vn), toxicidade(t), consumido(c){};
    alimentos(initializer_list<string> ch);
};
```

Figura 3 – class alimentos

“interface.h”

Contém a class interface.

```
class interface{  
  
public:  
    static string recebeComandos();  
    static int validaComandos(string recebe);  
    static int leFicheiro();  
    static int leFicheiroComandos();  
};
```

Figura 4 – class interface

“reserva.h”

Contém a class reserva.

```
class reserva{  
    int nl;  
    int nc;  
    vector<animal> vAnimais;  
    vector<alimentos> vAlimentos;  
  
public:  
    static void printGrid(int &nl, int &nc);  
    static int print();  
    void adicionaAnimais(animal a);  
    void adicionaAlimentos(alimentos as);  
};
```

Figura 5 – class reserva

CONCLUSÃO

No geral gostámos muito de implementar este trabalho prático, adquirimos vários conhecimentos que ainda não tínhamos, uma vez que é um tipo de programação com um “propósito diferente”, visto que se trata de um projeto por inteiro, e não de apenas um exercício. Cumprimos quase todos os requisitos dados pelo enunciado da meta 1, apesar de que com algumas dificuldades e cometendo algumas lacunas, mas achamos que realizámos um bom projeto.

FIM