



Instituto Superior de Engenharia de Coimbra

Licenciatura Engenharia Informática

Programação Orientada a Objetos

Trabalho Prático 2022/2023

Meta 1

Miguel Semião 2012012267

Rafael Alves 2014013189

Índice

Introdução.....	3
Decisões tomadas	3
Classes usadas na meta 1	4
Descrição das classes	5
Classe Reserva:	5
Classe Interface:	6
Classe Animais:	7
Classe Alimentos	8
Relação das classes	9

Introdução

O trabalho prático proposto da disciplina de Programação Orientada a Objetos pretende criar um programa em C++ de um simulador de uma reserva natural povoada por diversos animais.

Para esta primeira meta do projeto foram pedidos os seguintes objetivos:

- Leitura do ficheiro de comandos e constantes;
- Construção da reserva e representação visual;
- Implementação da leitura e validação de todos os comandos;
- Implementar os comandos para: ver animais e alimentos;
- Definição do conceito de Animal e Alimento.

Para a realização do mesmo, foram interpretados diferentes conceitos e entidades que foram traduzidas em classes no projeto, cujo mesmos irão ser descritos na próxima secção.

Decisões tomadas

Na realização da meta 1 tendo em conta que começou com uma representação simplificada da reserva, tomamos as seguintes decisões:

Uma vez que a implementação dos comandos de criação de alimentos e animais ainda não eram exigidos para a realização desta meta decidimos criar vários elementos manualmente com valores fixos. (Ex: linha, coluna).

Optamos por tratar a reserva como um conceito e cada um dos elementos (animais e alimentos) sabem em que coordenadas se encontram.

Na representação da área visível (começa por defeito no canto superior esquerdo da reserva) é utilizado um *array* de 10x10, e preenchido por “espaços”, que por sua vez verifica a existência de alguns animais ou alimentos nas posições da área visível e correspondentes à sua posição na reserva.

Para o comando *slide* decidimos estabelecer um limite de movimento de 10 casas cada vez que o utilizador executa este comando.

Na leitura do ficheiro constantes.txt, visto os recursos não estarem incluídos nesta meta, implementamos a leitura do ficheiro e a separação das variáveis e o seu valor e apresentá-lo ao utilizador. Implementação esta que irá ser atualizada para a meta 2 introdução dos valores nos atributos de cada classe.

Classes usadas na meta 1

- Reserva
- Interface
- Animais
- Alimentos

Descrição das classes

Classe Reserva:

```
class Reserva {
    // Máxima dimensão reserva
    int linhas;
    int colunas;
    vector<Animais> animais;
    vector<Alimentos> alimentos;
    char areaVisivel [10][10]{};
    int xvisivel = 0, yvisivel= 0;
public:
    Reserva(int n_linhas, int n_colunas);
    vector<Animais> getanimais(){return animais;}
    int getlinhas() const{return linhas;}
    int getcolunas() const{return colunas;}
    int getxvisivel() const{return xvisivel;}
    int getyvisivel() const{return yvisivel;}
    void setarea(int x, int y){xvisivel=x;yvisivel=y;}
    void gerar();
    void atualizaReservaVisivel();
    void MostrarZonas();
    void verQuadrado(int l, int c);
};
```

A classe Reserva contém diversos métodos tais como: (*atualizaReservaVisivel*) atualiza o que está dentro da área visível, (*MostraZonas*) imprime a representação da área visível, (*verQuadrado*) informação da posição e das coisas que lá estão, (*gerar*) criação manual dos elementos.

Classe Interface:

```
class Interface {  
  
    public:  
        void ui();  
        void lerinput(istream& iss, Reserva * r);  
        void lerconstantes(istream &iss);  
};
```

A classe Interface executa os Comandos introduzidos pelo utilizador e leitura de ficheiros.

Classe Animais:

```
class Animais {  
    int id=1;  
    char especie;  
    int linha;  
    int coluna;  
    bool vivo=true;  
    int idade=10;  
    int saude=30;  
    int movimInst=1;  
    int aprox=4;  
    vector<Alimentos*> historico;  
public:  
    Animais(char especie_n, int linha_n, int coluna_n);  
    char getespecie() const{return especie;}  
    int getlinha() const{return linha;}  
    int getcoluna() const{return coluna;}  
    bool getvivo() const{return vivo;}  
    int getidade() const{return idade;}  
    int getsaude() const{return saude;}  
    int getmovimInst() const{return movimInst;}  
    int getaprox() const{return aprox;}  
    void imprimeInfoAnimais() const;  
};
```

A classe Animais guarda a informação dos animais. A função: (*imprimeInfoAnimais*) imprime a informação detalhada de cada animal.

Classe Alimentos

```
class Alimentos {  
    int id=1;  
    char tipo;  
    int linha;  
    int coluna;  
    bool consumido=false;  
    int tempo=10;  
    int valorNut=10;  
    int valorTox=0;  
    string cheiro="carne";  
public:  
    Alimentos(char tipo_n, int linha_n, int coluna_n);  
    int getid() const{return id;}  
    char gettipo() const{return tipo;}  
    int getlinha() const{return linha;}  
    int getcoluna() const{return coluna;}  
    bool getconsumido() const {return consumido;}  
    int gettempo() const {return tempo;}  
    int getvalornut() const {return valorNut;}  
    int getvalortox() const {return valorTox;}  
    string getcheiro() {return cheiro;}  
    void imprimeInfo();  
};
```

A classe Alimentos guarda a informação dos alimentos. A função: (*imprimeInfo*) imprime a informação detalhada de cada animal.

Relação das classes

- **Reserva**

Relaciona-se com a classe Animais e Alimentos tendo um vetor para cada um, para armazenar os elementos da respetiva classe.