

Instituto Superior de Engenharia de Coimbra
Departamento de Engenharia Informática e de Sistemas



Licenciatura em Engenharia Informática

Trabalho Prático de Programação Orientada a Objectos

Relatório

(Meta 1)

Turma Prática nº5

Daniela Neves, 21220120
Nuno Rocha, 21240505

Coimbra, dezembro de 2017

Organização do código apresentado

1. Quais foram as classes consideradas na primeira versão da aplicação que foi testada?

Esta primeira versão do trabalho tem as seguintes classes com as seguintes funcionalidades:

- Formigas;
- Ninhos;
- Mundo;
- Comunidade;

Cada uma destas classes tem os seus métodos get's e set's implementados. Nenhuma delas ainda contém funcionalidades especiais, ainda.

Para além destas classes, temos também um ficheiro *Screen.h* e *Screen.cpp* que serve essencialmente para a parte da interface do jogo.

2. Quais os conceitos/classe que identificou ao ler o enunciado?

- Formigas;
- Comunidades;
- Ninhos;
- Mundo;
- Número de identificação da formiga;
- Visão da formiga;
- Movimento da formiga;
- Energia dos ninhos e das formigas;
- Número de identificação dos ninhos;

3. Relativamente a duas das principais classes da aplicação, identifique em que classes ou partes do programa são criados, armazenados e destruídos os seus objectos.

Formigas: os objectos desta classe são criados e armazenados na classe Ninhos;

Ninhos: os objectos desta classe são criados, armazenados e destruídos na classe Comunidade;

Comunidade: os objectos desta classe são criados, armazenados e destruídos na classe Mundo;

- 4. Indique um exemplo de uma responsabilidade atribuída a uma classe que esteja de acordo com a orientação dada acerca de Encapsulamento.**

A responsabilidade de “criar formigas e defendê-las” está atribuída à classe Ninhos, porque é constituída por um conjunto de formigas.

Já a responsabilidade de obterem energia (através das Migalhas) e moverem-se está atribuída à classe das formigas.

- 5. De entre as classes que fez, escolha duas e justifique por que considera que são classes com objetivo focado, coeso e sem dispersão.**

Classe Ninho: tem dados e responsabilidade de receber os coordenadas aonde irá criar o novo ninho.

Classe Formiga: está focada no tipo de formiga, na sua posição e a que ninho pertencem;

- 6. Relativamente à aplicação entregue, quais as classes que considera com responsabilidades de interface com o utilizador e quais as que representam a lógica?**

Responsabilidade de interface: Screen e Consola;

Responsabilidade da lógica da aplicação: Formigas, Comunidades, Ninhos e Mundo;

- 7. Identifique o primeiro objecto para além da camada de interacção com o utilizador que recebe e coordena uma funcionalidade de natureza lógica?**

As ordens vindas da camada de interacção com o utilizador são recebidas e processadas por um objecto da classe Mundo;

8. A classe que representa a envolvente de toda a lógica executa em pormenor muitas funcionalidades, ou delega noutras classes? Indique um exemplo em que esta classe delega uma funcionalidade noutra classe.

A classe Mundo representa a envolvente de toda a lógica. Para que sejam criados os ninhos e por sua vez as comunidades e as formigas a classe Mundo é que vai delegar a inserção no jogo de cada um dos objectos.

9. Apresente as principais classes da aplicação através da seguinte informação:

Classe: Mundo

Responsabilidade:

- ❖ Obter as coordenadas de um ninho, que por sua vez permite saber onde nascem as formigas;
- ❖ Representar os constituintes do nosso jogo;

Colaborações: Ninhos, Formigas e Comunidades;

Classe: Ninho

Responsabilidade:

- ❖ Obter a localização;
- ❖ Obter o número de identificação do ninho;
- ❖ Obter a energia inicial do ninho;
- ❖ Criar as formigas que irão gerar as comunidades;

Colaborações: Formigas e Comunidades;

Classe: Formiga

Responsabilidade:

- ❖ Obter a energia inicial da formiga;
- ❖ Obter a visão que cada tipo de formiga tem;
- ❖ Obter o raio de movimentação que cada formiga tem;
- ❖ Obter os diferentes comportamentos de cada formiga;

Colaborações: Formigas e Comunidades;

Classe: Screen

Responsabilidade:

- ❖ Mostrar as funcionalidades da aplicação;
- ❖ Obter do utilizador as opções pretendidas;
- ❖ Transmitir as ordens do utilizador ao Mundo;

Colaborações: Mundo;

Funcionalidades Implementadas

Componente do trabalho	Realizado	Realizado parcialmente	Não realizado
Leitura e validação de comandos	x		
Leitura de comandos em ficheiro	x		
Criação de mundo ninho e formigas		x	
Visualização do mundo e conteúdo		x	
Formiga exploradora (move)			x
Avanço de iteração			x
Listagens (listamundo, listaninho, listaposicao)			x