

Relatório Projecto Final

Programação Orientada a Objectos

2017/2018

Trabalho realizado por: Samuel Camacho

Nº: 70285

Docente: Luís Nunes

Tabela 1 Preenchida:

Características:	Implementada na(s) classe(s)/método(s):
Encapsulamento	Todas as Classes têm os seus atributos a private, para q não possam ser acedidos diretamente fora da classe. Só podemos definir e obter valores dessas variáveis através dos métodos da classe.
Herança e Sobreposição de Métodos	Todos os objectos do jogo estendem a classe FireFightObject, ou seja, esta serve como classe Base. A class FireFightObject implementa a interface ImageTile. Os Objectos usam as funções implementadas na classe FireFightObject como o método setPosition(Point p); setName(String name); isBurnable(); isCatchable(), etc... A classe FireFightObject faz Override às Funções getName(); getLayer(); getPosition() implementadas na interface ImageTile. A classe FireSimulator implementa Observable.
Utilização de Listas ou Estruturas de Dados	No meu projecto são usadas duas Listas: A lista (objects) de Todos os Objectos que estendem a classe FireFightObject, excepto os objectos da class Fire; A segunda Lista(firePositions) contem os Objectos da classe Fire.
Leitura de Ficheiros	A Função que lê o ficheiro correspondente ao nível é a Função ReadConfig e está implementada na classe FireSimulator.

Excepções	Na classe FireSimulator, na função ReadConfig é lançada a excepção “FileNotFoundException e”.
JavaDoc	Os Comentários feitos através do JavaDoc estão implementados na classe FireFightObject e Pine.
JUnit	Os testes JUnit estão no pacote “pt.iul.poo.firefight.main” e nome é JUnitPine. Este teste é feito à classe Pine e tem os seguintes métodos a Override: @Before e @Test.

Relatório Sobre a Execução do Projecto em Geral:

Nesta secção irei falar em primeiro lugar sobre as dificuldades encontradas ao longo do trabalho, assim como as opções tomadas para as ultrapassar.

Creio que a primeira dificuldade encontrada, no meu caso, foi perceber a como iria acontecer a propagação do fogo pela floresta. Ao início pensei que um fogo é que se propagava, logo aconteceu um problema em que este só se propagava através da ultima posição para onde o fogo se tinha propagado anteriormente. Assim, decidi criar uma lista do tipo ArrayList<Fire> para guardar as minhas posições dos fogos. Desta maneira, e após criar as Funções onde é criado um ponto aleatório para ele se propagar, ver se nessa posição está um objecto ardível ou não, e se nessa posição não está nada a arder (para evitar redundâncias), então basta percorrer todos os fogos da lista e cada fogo ira criar um novo fogo na posição aleatória, se esta for válida. Assim consegui colocar todos os fogos a propagarem-se.

A segunda dificuldade foi fazer o bombeiro sair do Bulldozer através da tecla Enter. Para isso tive que criar uma variável booleana para que quando a tecla Enter é colocada e o valor booleano teve um certo valor ele não volta a entrar. Contudo, quando ele sai, tem que se actualizar essa variável para que quando o Bombeiro passa de novo na Posição do Bulldozer, poder” conduzi-lo” de novo.

A terceira maior dificuldade encontrada foi para o Avião descobrir a Coluna com maior número de Fogos. A dificuldade surgiu mais na questão de como é que iria guardar a cada iteração a coluna com mais fogos, no entanto esta dificuldade foi ultrapassada, criado duas variáveis adicionais chamadas de maxFireColumn e maxFire, em que ambas começam a zero e a cada iteração da variável currentColumns e da variável currentFires, se o valor destas variáveis for maior que maxFireColumn e maxFire, então, masFire será igual a currentFires e, maxFireColumn será igual a currentColumn.

Por fim, a maior dificuldade encontrada foi colocar a Imagem e Objecto Burnt depois de um objecto ardível estar a arder e o numero de ciclos em combustão desse objecto chegar a zero. Ao Início não estava a conseguir decrementar os ciclos dos objectos de eles estarem a arder. De uma maneira resumida, a solução passou por fazer uma função vazia na classe Base FireFightObject, em que depois é contruída nas classes dos Objectos ardíveis como Pine, Eucaliptus e Grass, o nome da função é StartCycle(). Esta função quando é chamada decrementa um valor ao ciclo do objecto, se o ciclo actual for diferente de zero. Esta função só é chamada se o objecto estiver a arder e se o seu Ciclo não estiver em zero.

Manual de Utilização do Jogo:

Manual para o Movimento dos Objectos Activos:

- O Bombeiro move-se através das setas no teclado do Portátil/ Computador Tradicional;
- À medida que o bombeiro se move, o fogo irá propagar-se pela floresta escolhendo um ponto aleatório;
- Para o Bombeiro apagar o fogo, basta “conduzi-lo” às posições dos fogos. Quando a próxima posição do mesmo estiver a arder, o Bombeiro apagar o fogo e, só depois de este estar extinto é que o Bombeiro irá avançar;
- Para Conduzir a Empilhadora, basta passar na Posição da mesma. Assim que se passa nesta posição, o Bombeiro entra na empilhadora e o utilizador assume o comando da mesma. A empilhadora transforma em terra todas as posições por onde passa, desde que nessa posição não exista fogo, de forma a fazer barreiras de terra antifogo;
- Para o Bombeiro sair da Empilhadora, basta carregar Enter e andar nas setas;
- Para chamar o Avião e pedir ajuda aérea, basta carregar na Tecla A, o avião irá começar na parte inferior do mapa, na coluna onde existirem um maior número de fogos;
- A cada movimento do Bombeiro, o Avião irá andar duas casas na vertical, apagando todos os fogos nas posições onde passa até que sai dos limites do mapa;
- Quando o Avião desaparece dos limites do mapa, basta carregar novamente a tecla A para que este volte a aparecer para ajudar no combate ao incêndio.

Declaração de Integridade e Autoria do Projecto Final

Eu, Samuel Coelho Camacho da Silva, N° 70285, matriculado no Instituto Superior ISCTE-IUL, no Curso de Informática e Gestão de Empresas, declaro que todo o conteúdo inserido no Projecto Final da Unidade Curricular Programação Orientada a Objectos foi da minha autoria e que, nenhum do código utilizado e apresentado foi plagiado a outra entidade ou pessoa.

O Aluno: *Samuel Camacho*

Data: 14 de Dezembro de 2017