

Trabalho Programação II 2016/2017

Diogo Rafael nº37859 // Vasco Crespo nº 37913

19 de Junho de 2017



1.Introdução

Neste trabalho é pretendido criar um programa que pede ao utilizador um ficheiro de texto com um labirinto(sendo “_” os espaços em que se pode andar, “W” as paredes, “S” o principio e “E” o fim), é então criado esse labirinto num array de arrays e é pedido ao utilizador se quer entrar no modo play(onde pode escrever north, south, west ou east para movimentar o peão) ou se quer entrar no modo solve(onde o labirinto é resolvido automaticamente e o caminho é printado no fim).

Para tal, serão utilizados vários métodos e classes que serão descritos nas próximas páginas.

Funcionamento de métodos(Classse Main):

1.1- main:

Este método é o principal onde são chamados todos os outros métodos(quer seja diretamente ou através de outros métodos), primeiro é pedido como input o nome do ficheiro e depois é chamado **posS**(que por sua vez chama **Openandcreate** e **OpenandDraw**), é então printado o labirinto e pedido o modo desejado(play ou solve), caso seja escolhido **play**:

Enquanto o labirinto não estiver resolvido o programa pedirá uma direção ao utilizador e quando for resolvido irá printar uma mensagem a dar os parabéns ao utilizador e irá parar o programa.

E caso seja escolhido **solve**:

É chamado **findRoute** e é printada a rota final e o labirinto final(com X marcado por onde passou e O em sítios onde ficou preso e teve de voltar atrás)

1.2- OpenandCreate:

Este método vai ler as linhas do ficheiro de texto através de um ciclo **for** e definir um número máximo de linhas e colunas(e caso haja duas linhas com tamanhos diferentes retorna um erro) e retornar um labirinto **meize** com as dimensões corretas que será mais tarde preenchido.

1.3- OpenandDraw:

Este método chama **OpenandCreate** e vai preencher o **meize** anteriormente criado com os caracteres contidos no ficheiro txt.

1.4- posS:

Este método chama **OpenandDraw** e encontra a posição de S no labirinto e define um array com as posições encontradas nele.

1.5- player:

Este método chama **move** da classe **Maze** que consoante os inputs do utilizador vai mover o peão no labirinto, verificando sempre se o movimento é possível através do método **canMove**.

1.6- FindRoute:

Este método vai verificar uma rota possível para a resolução do labirinto através de verificações feitas no próprio método e de um método **canMovimentar** definido na classe **Maze**.

Funcionamento de métodos(Classe Maze, Route e Pawn):

2.1- construtor Maze:

Este construtor chama a classe Pawn para definir peões que se vão mexer no labirinto.

2.2- canMove:

Este método vai verificar a possibilidade de executar um movimento através de um switch, será apenas utilizado no modo **play**.

2.3- move:

Este método vai verificar se **canMove** é true e se for vai executar o movimento introduzido pelo utilizador caso **canMove** seja de facto true.

2.4- isSolvedBy:

Este método verifica se o labirinto está resolvido, comparando a posição do peão com E.

2.5- canMovimentar:

Este método faz uma verificação de se o peão se pode movimentar, desta vez sem inputs necessários.

3.1- move:

Este método adiciona todos os inputs introduzidos pelo utilizador até ao final do labirinto no modo play e depois printa-os por ordem.