

Relatório do trabalho “Fusion” para a disciplina de Programação 2



Trabalho realizado por:

Luís Paulo Ricardo nº37722

Miguel Portugal nº38128

Professores:

Salvador Abreu

Vasco Pedro

Ano Letivo 2017/2018

2º Semestre



Introdução

Este trabalho é constituído por 4 classes, uma que gere o jogo, Classe Fusion, uma que gere o quadro, Classe Board, uma que gere as jogadas, Classe Jogadas e uma que gere os inputs, Classe Inputs.

A Classe Fusion gere o jogo e chama todos os Métodos utilizados no jogo

Classe “Fusion”

Para esta classe, em primeiro fizemos um import das funcionalidades em geral do Java. De seguida, declaramos as variáveis que indicam o tamanho do quadro (boardsize) e o número de cores (ncores). Também invocamos o scanner.

De seguida criámos um método start para iniciar o programa. Criámos um booleano para usar como parâmetro no ciclo while, para que pudéssemos utilizar o try e o catch até que fosse introduzido um valor dentro dos parâmetros. Se o utilizador introduzir um valor fora do intervalo entre 6 e 50, inclusive, o programa vai pedir ao utilizador um novo valor. Caso o utilizador introduza um carácter diferente de um número, o catch é ativado, o programa dá erro e pede o valor outra vez. O mesmo pensamento se aplica ao número de cores.

A seguir, fizemos um método main. Este chama todas as outras classes e as funções necessárias para começar o jogo. Por fim realizámos um ciclo while para que o programa peça o input da jogada ao utilizador até deixar de haver jogadas possíveis. Quando o ciclo termina, chamamos o método pontuação

Classe Input

Nesta classe declaramos as variáveis linha e coluna e criamos o método pedir.

É pedida uma letra ao utilizador, ao que não importa se esta é minúscula ou maiúscula, com ajuda do código ascii e do código char(x), onde x é o número da letra no código ascii, introduzida pelo utilizador, podemos passar da letra que o utilizador que o utilizador pediu para um número, este será a linha e a coluna. Caso a letra não esteja no intervalo do tamanho escolhido para o tabuleiro, o programa pede outra letra até que o utilizador introduza uma letra correspondente ao intervalo.

Depois das verificações, guardamos o valor correspondente à letra escolhida, nas variáveis linha. Mesmo processo para a coluna

Classe Board

Nesta classe, após fazer o import do random, criámos um array bidimensional para fazer o quadro.

Criámos o método newBoard com o tamanho dado pelo input do utilizador. Com dois ciclos for, um para as linhas e outra para as colunas, preenchemos todas as posições do quadro com números aleatórios, sendo cada um destes corresponde a uma cor, gerados pelo random.

Seguidamente, criámos o método showBoard, que faz print das letras de acordo com o tamanho do quadro. De seguida imprimimos uma letra e todos os números dentro do array, por linha, ficando no final com todos os integrantes do array e por cada linha uma letra.

O método refresh percorre o quadro, coluna a coluna, começando pelo fim. Sempre que encontra uma posição com 0, copia o primeiro número não 0 dessa coluna para essa posição, repetindo este processo para todas as colunas, ficando com todos os 0 no topo do quadro. Depois deste acontecimento, este método vai ainda substituir todas as peças com número 0 por novas com número aleatórias diferentes de 0

Classe Jogada

O primeiro passo que nos fizemos nesta classe foi criar o método confirmPlay, o qual, através de quatro variáveis, sendo cada uma delas relativa a uma direção do quadro, verifica se a peça selecionada tem peças com o mesmo número nas posições adjacentes. Este método possui ainda uma variável relativa à pontuação, a qual, após confirmar que a jogada é possível, adiciona à pontuação total a pontuação parcial.

De seguida criámos o método possibleToMove, que, em caso de o utilizador ter jogadas possíveis, a função retorna True

Por fim, o método pontuação, que vai imprimir a pontuação obtida durante o jogo.

Conclusão

Com este trabalho colocamos em prática a matéria lecionada durante o semestre, o que nos permitiu desenvolver um código bem estruturado e organizado tal como nos foi pedido.