HACETTEPE

UNIVERSITY

BBM104

ASSIGNMENT 3

THE BEJEWELED GAME

Hüseyin Eren DOĞAN – 2210765009

In this assignment, we had to make a Bejeweled-like game. Firstly I need a game grid which is given in the “gameGrid.txt” and also commands that is given in a text file too. For this, I opened a class “Reader”. It has a method which returns a 2d array of ingredients of text files splitted by “ “.

Then created a class “Writer”. It has three methods. First one is for writing anything to a text file. It takes two parameters which is the file’s name and the string we’re wanting write to file.

The other methods is “drop” and “print_grid”. “drop” method changes the empty cells with its upper cell which is non-empty. “print_grid” prints the grid.

Then opened a class “Jewel”. It stores letter symbols of jewels in arrays, their scores in a HashMap. And also I create an empty ArrayList in this class to store the jewels which I matched for calculating the total score.

Then created “Diamond”, “Square”, “Triangle”, “Wildcard”, “Symbols” classes as subclasses of “Jewel”. All of them has methods named as the jewel’s name. They takes parameters which is game grid, vertical and horizontal position values, scores and jewel symbols.

In these classes, I assigned the directions which they searches for and then searched for these directions in each jewel. Then prints the grid with “print_grid” method and prints the score.

And then created the “Player” class which implements Comparable interface for adding new player to other players in “leaderboard.txt” and sorting them to find the new player’s ranking.

This class contains three methods and one constructor. All of them except “compareTo” takes player’s name and score as its parameters.

First method is “sorted_leaderboards”. It reads the “leaderboard.txt” and add the players in that text to an ArrayList. And adds the new player’s name and score to it, then sorts the ArrayList with their points and returns the sorted ArrayList.

The other method “ranking” searches our new player in sorted ArrayList and finds its ranking and difference to the other players which is in front or behind of him/her.

The main method is in the “Main” class and it reads the files with “text_reader” method in “Reader” class and calls the “game” method in “Game” class.

The “Game” class has a method that takes the game grid and commands as its parameters. And for each command, if it is a coordinate it selects that cell of grid and calls the method of its jewel.

Then when it reaches the end of the commands, it prints total score and name of the player and call the “ranking” method in “Player” class then adds the player’s name and score to the “leaderboard.txt” then prints “Good bye!” and ends the game.

And here you can see the UML class diagram of my code: