PROGRAMMING ASSIGNMENT 4

TAs [redacted]
Due Date : 17.12.2021 (23:59:59)

Identic Color Game

Background

In this assignment, you will get familiar with file operations, lists, higher order functions, exceptions, and design a relatively complex algorithm. You are expected to design a game called IdenticColorGame. This game is played on a rectangular board, typically initially filled with different colored balls as shown in Figure 1. By selecting a group of adjoining balls of the same color, a player may remove them from the screen. Balls that are no longer supported will fall down, and a column without any balls will be trimmed away by other columns always sliding to the left side. The goal of the game is to remove as many balls from the playing field as possible. How to play the game is explained in detail in the following sections.

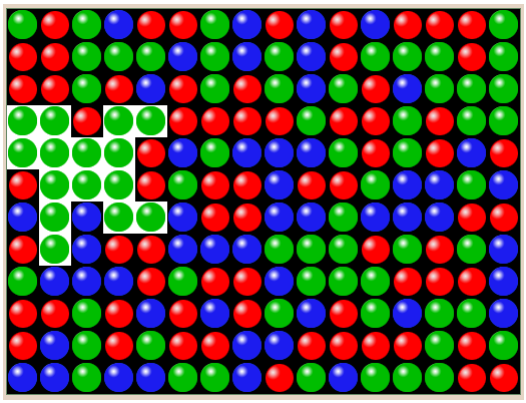

Figure 1: Example board of IdenticColorGame

Assignment

Your assignment is to develop a board game consisting of different colored balls. The board consists of several rows and columns where balls are distributed randomly among the cells of the board. In your assignment, you should read color of balls from an input file (input.txt), so that your program will work on different board sizes. The structure of the input file is as follow:

[Ball]***space***[Ball]***space***[Ball]***newline***
[Ball]***space***[Ball]***space***[Ball]***newline***
.....

Every cell has four neighbors left, right, above and below. This game is a collect game where of each turn you should collect two or more balls based on spatial relationship. That is, once

you pick a ball, all neighboring (including the cell you picked) that containing ball with the same color will disappear from the board. Note that the selected cell must include at least one neighboring ball with the same color. Figure 2 show that neighbor cells of selected cell in a sample board with balls of the same color.

Figure 2: Neighbor cells of selected cell in a sample board configuration

If a cell disappears in a column, the rest of that column moves down to fill the row blank cells, as shown Figure 3. Moreover, if a column disappears completely, all the cells which are at the right side of that blank column should move left to fill the empty space as shown in Figure 4.

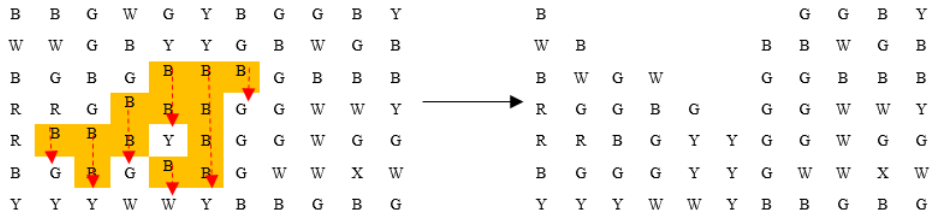

Figure 3: View of board after the neighboring balls of same color are deleted

Deleting empty cell is the most important part of your implementation. While you are designing your code, you have to be careful in finding the neighboring cells with the same ball, removing the corresponding cells and filling the blanks.

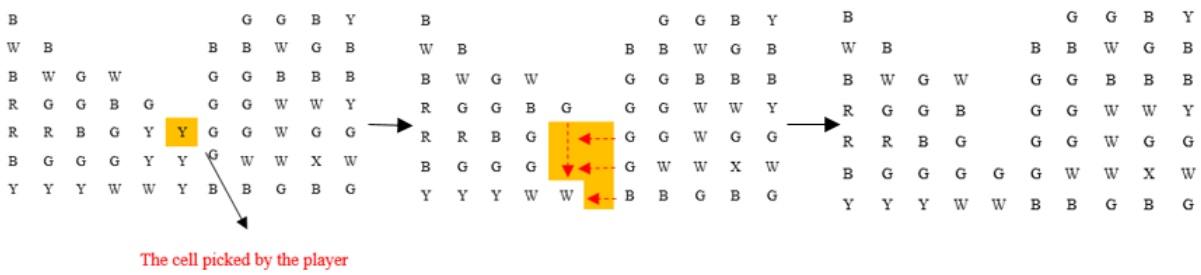

Figure 4: View of the board after a column disappears completely

In addition to the colored balls there may be a bomb on the board. If the player picks this bomb that shown by "X" character, all the balls in the row and column (including the bomb) where the bomb is located will be deleted regardless of their color as shown in Figure 5. While doing this process, the rules shown in Figures 3 and 4 should be considered.

Figure 5: View of the board after the explosion of the bomb

Game Play

In the beginning, you have to pose an input file which denotes the initial configuration of the board. Here, please note that file name should be given as an argument. After that, the board should be printed to screen. Also the current score (which is 0 at the beginning) should be printed to screen. The game begins by asking the user to select a cell the correspond row and column (There must be a space between the entered row and column).

-If the chosen cell is out of bounds, you have to print an informative error message to the screen ("Please enter valid size!").

-If the chosen cell has no neighbor with the same color, no change in the board is expected. In this case, the previous status of the board will be displayed on screen and the user will be asked for new coordinates.

Otherwise, the new state of the board should be printed together with the updated score. If there is no cell which has no neighbor with ball of the same color and also there is no bomb (x) in a cell, it means that the game is over. Realizing the end of the game is a bit tricky so you should be careful to implement this feature. **As a result of deleting balls of the same color, the view of the board changes. Therefore, please note that the coordinates of a ball in the board also change accordingly as shown in Figure 6**. An example gameplay is shown in Figure 7.

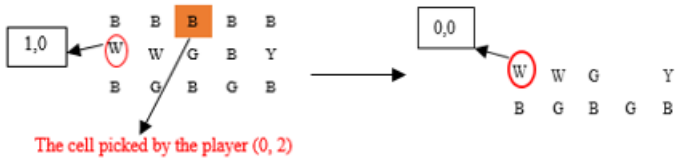

Figure 6: Changing coordinates of the ball

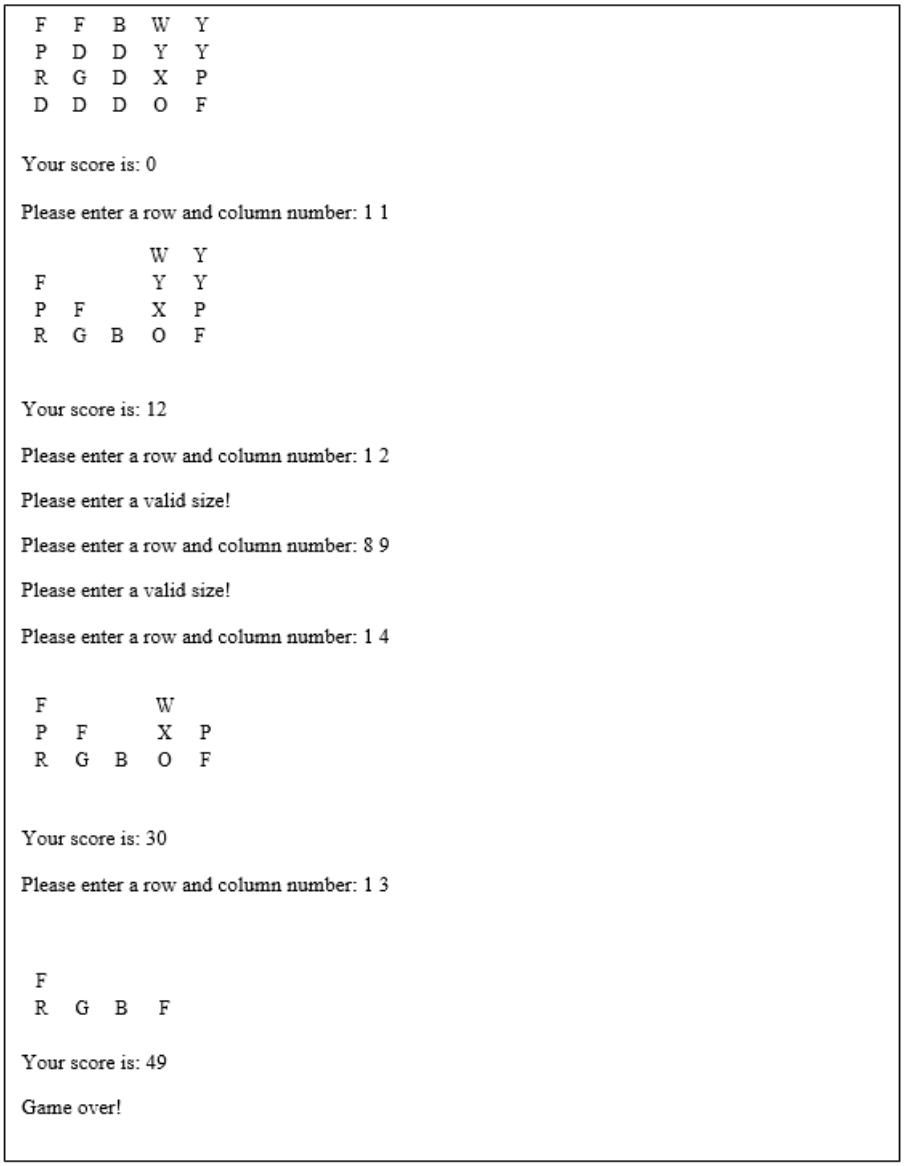

Figure 7: View of screen when source code is running

The Scoring

There can be 9 different colored balls and 1 bomb on the board. The weight of each of these different colored balls and the weight of the bomb are listed in the Table 1. Balls of each color and the bomb contribute to the score as much as their weight. For example, if 3 red (R) balls are removed from the board in one move, its contribution to the score will be 15 (number of balls disappeared * color weight (3*5=15)). Another example, if the user picks the cell with the bomb, they will contribute to the score equal to the total weight of the balls in that row and column as shown end of Figure 7.

Color	Abbreviation	Weight	Color	Abbreviation	Weight
Black	B	9	Pink	P	4
Gray	G	8	Orange	O	3
White	W	7	Dark blue	D	2
Yellow	Y	6	Fuchsia	F	1
Red	R	5	Bomb	X	0

Table 1: List of weights of colors

Grading Policy

Task	Point
Submit	1
Coding style	10
Coding standard	5
Output	84
Total	100

Important Notes

- Do not miss the submission deadline.
- Compile your code on dev.cs.hacettepe.edu.tr before submitting your work to make sure it compiles without any problems on our server.
- Save all your work until the assignment is graded.
- In the "Coding style" of the grading policy section, it will be taken into account whether unnecessary code fragments are used, the abstraction level is regular and your code is understandable.
- In the "Coding standard" of the grading policy section, it will be taken into account whether variable names or function names are used in the correct format.
- The assignment must be original, individual work. Duplicate or very similar assignments are both going to be considered as cheating. You can ask your questions via Piazza and you are supposed to be aware of everything discussed on Piazza. You cannot share algorithms or source code. All work must be individual! Assignments will be checked for similarity, and there will be serious consequences if plagiarism is detected.
- You have 3 days extensions for this assignment. 1-day late submission will be graded over 90 points. 2-day late submission will be graded over 80 points. 3-day late submission will be graded over 70 points
- You may assume that the input files will be given as command-line arguments in the following order: input.txt, so to execute your code on dev use the following command in your terminal:
  python3 assignment4.py input.txt

- You must submit your work with the file hierarchy as stated below:

  - $\langle studentid \rangle.zip$

    - assignment4.py