Tom Tianshu Lv

tl1443

Lab Report: Shades Game

Functions:

Left, right, quick down with keyboard

Merge

Eliminate one line

Pause, reset and showing score

Basic Structure:

The whole game is in an object called Shades extending JFrame. It contains two panels: menu panel on the top and board panel on the bottom.

Menu panel: contains instructions to the game in the form of blank bottom in order to makes the window look better.

Board panel: it extends a JPanel to add new blocks. It also implements KeyListener and Thread to make itself a separate thread with the main function and can listen to the behavior of itself. The Board panel has a blockArray to contain the object of blocks in its corresponding location. The original point is the left bottom. It also has a rowArray that has the number of blocks already existing on every row. Finally, it has a colorList containing all five colors.

Block: it’s an object that has JLabel as its instance field. The JLabel is a colorful rectangle with an empty string It has many methods to control the JLabel inside including placing it on the board, changing its location, changing its color and making it disappear.

Algorithm for all functions:

Basic movement: The infinite loop creates a new block in every round. It randomly gets an xAxis and gets an index in 0 to 3 randomly in the colorList. So the block has the position and color ready. Then in a loop that ends when the block reaches the end, which is the bottom or the top of another block, the block every time increase one in yAxis. Then I make the board thread sleep for 5 milliseconds to get the falling animation. When the block reaches the position, it is added to the corresponding position in blockArray and this row in the rowArray is added by 1. Then when checking the merge and elimination one line, it falls another block.

Merge algorithm:

First, check whether the xAxis the block just fell in has at least two blocks. If so, check the whether the up one and the block under has the same color. If they have the same color, I first change them to the next color level. And let the up block fall down using the same method as the basic movement. Then I delete the up block in blockArray and decrement rowArray. Then I follow the same checking process until getting out of loop.

Elimination one line:

First, I get the highest level where the four blocks are full. Then I loop from the bottom to full. Every time I check whether the four colors have the same color, and if so I enter the eliminating method. For every row, I update the color in the bottom with the color in its upper block from bottom to top. Then I delete the very top block in blockArray and decrement rowArray. Finally, I decrement full by 1 because all the upper blocks have fall by 1 level. If I do not do the elimination, I increase the level by 1.

Pause, reset:

Every time space is pressed, a Boolean variable isStart is changed into false, and the falling loop doesn’t increase y by 1 anymore. If it’s pressed again, the variable is set into true.

Reset is also done with a Boolean variable isReset. If it is true, it jumps out of the falling loop. Then it moves all the existing blocks to nowhere and clears the rowArray and blockArray. Then a new loop for a new block continues.

Left, right and down:

They are implemented by a keylistener as a board.

The down algorithm is interesting because it also controls a variable called isFast. If it is true, the thread sleep time goes to 1 millisecond. If not, it remains 5 milliseconds.

Challenges:

The program with JLabel representing a block is unique, so I mainly rely on myself to develop the whole project. Debugging is painful because there are little mistakes everywhere. But the setbounds method of JLabel is useful. I don’t need to repaint the whole board every time I update. I just have to update the falling block. But the block should have many method because it will contribute a lot in controlling JLabel, which makes Block class very heavy.

Overall, I have a strong feeling with this project because at the beginning I cannot even imagine I can finish this. I think it is beyond my ability. But after working on it night after night, I’m very satisfied with my current project.