Lab Report of Shades

Autumn Wu

26 April 2015

1 Public Class *Shades()*

Class *Shades* extends *JFrame* and contains two classes *Board* and *TimerListener*. In the main method, I construct a timer and a menu. Four *JmenuItems* are added to my menu. They are *New Game, Pause, Continue* and *Exit.* Each one has a function as described by their names. Implementation of the functions is done by adding *actionListener* to each *JmenuItem.*

2 Class *Board()*

Class *Board* extends *JPanel* and implements *KeyListener.* It has several int variables: *color* – indicating the current color of the block, *x y* – indicating the position of the block, *Vely* – indicating the velocity of the block. It also has two boolean variables: ismerge and isdown.

There are two very significant matrices *map* and *Reccolor.* They keep track of the positions and colors of all the existing blocks, which will be heavily relied on later. Since the color variable is integer, I construct a Color array to be referred when painting the block.

Board also has a *newblock* method which generates a new block by giving the (x,y) position, score and color. Moreover, after generating the new block, this method also checks to see if the game is over by checking whether the newly constructed block can go down or not. If the new block can not go down right after its construction, then game is over. It would then initialize *map, Reccolor score*, which basically starts a new game.

blow is another very important method in *Board* class. It checks for a given block, whether it reaches the bottom; whether it goes beyond the the wall. It's then used in methods *left and right.* It's also used to decide when to fix the block.

merge and delline methods are simply. For *merge*, we loop through that specific column and then check two rows once to merge. If it satisfies the requirements, then the upper block disappears from map, and its color is set back to be -1. Then we increase the color of the bottom block by one and increase score by 20. For *delline*, we loop over all the rows, and then add color and map status of each column to see whether they meet the requirement of delete a whole row. If it does, then we change the value of the disappearing row in *map and Reccolor* to the row above it.

For *paintComponent* method, we loop over the whole map, paint the block when it's 1 in the map with the color recorded in *Reccolor.* After paint all these fix blocks, we paint the current block with current(x, y) and current color. And then paint the score.

The last thing is *keyPressed, keyReleased and keyTyped.*It's very intuitive since we already have *left, right, down* methods above. *keyReleased and keyTyped* are reserved for future use.

3 Class ***TimerListener()***

This class implements *ActionListener.* If the user pressed down key, *Vely* would become 8. If the block goes beyond the bound, it will turn it back. Also it controls the *delline and merge* methods, and after merging or deleting a row or the block is fixed, it calls the *newblock*method to generate a new block.