DATA STRUCTURES

Project 1 Report

BY TARIK ALRAYAN

2121221366

1) Node Design

In the first I was confusing about the pointers should I give to my node class but after I organize the project, I used a node class with only a Next and Up Pointer beside the value and column number.

2) Multi Linked List

what I called it Mlist class this creates a special list that has the all the methods we need to play our game and I use a pointer to head and a couple of methods to help adding or removing nodes from the list

3) Node add method

If we can use an array or create a null nodes the project will be more easy and relatable but we cant so I decide to add the nodes to the list side by side not considering the column sorting and I keep the column index to use it in next methods

```java
//this method adding the nodes to the list then print them
public void addNode(int column, T value) {

        if (CheckAddNode(value, column)) {
            printGrid();
            merge(column);

        }
        else {
            System.out.println( x: "error while adding the node");
        }
}

//this method check the adding operation and if evrey thing correct it will added the node to the list
private boolean CheckAddNode(T value, int column) {
    if (column > numCols) {
        System.out.println( x: " Error the column index you entered is undifiend");
    }
    int row = 0;
    TarikAlrayanNode temp = head;
    TarikAlrayanNode down = temp;
    TarikAlrayanNode prev = temp;
    while (temp != null) {
        if (temp.getColumnIndex() == column) {
            row++;
            prev = down;
            down = temp;
        }
        temp = temp.getNext();
    }

    if (row >= numRows) {
        System.out.println( x: "Eror While Adding Nodes To List !! Out Of Index The Game Is Over ;)");
        return false;
    } else {
        TarikAlrayanNode newNode = new TarikAlrayanNode(value, column);
        addLast(newNode);
        if (row != 0) {
            down.setUp( up: newNode);
        }

        return true;
    }
}
```

-So here I take the value and the column and send them to checkAddNode these methods check the conditions to add a node and check the rows if they are full its giving error else its going to add them to list and update the next and the up pointers

4) print the Grid method

Here I spent a lot of time thinking about printing the grid row by row from down to up because of our game requirements so I created a 2 for loop first one is for the row and second one is for column so I start from the last row that equal the top and I go around the list and check if the number of the repetition is equal to the number of the row that mean this column has an element in this row else its put an _ to present its empty

```
private void printGrid() {
    TarikAlrayanNode temp = head;
    TarikAlrayanNode prev = temp;
    //the game drop numbers start from down to up that why i neede to start from down to up in for loop
    //in this loop i go around the list and check if the current column  repetition number is the same o
    //if it equal im gonna save the value using prev then i print the values of it if it not equal print
    for (int i = numRows - 1; i >= 0; i--) {
        for (int j = 0; j < numCols; j++) {
            temp = head;
            int rep = -1;
            while (temp != null) {
                if (temp.getColumnIndex() == j) {
                    rep++;
                    if (rep == i) {
                        prev = temp;
                        break;
                    }

                }
                prev = temp;
                temp = temp.getNext();

            }
            if (rep == i) {
                if ((int) prev.getValue() > 99) {
                    System.out.print("|"+prev.getValue()+"|");
                } else if ((int) prev.getValue() > 9) {
                    System.out.print("|_"+prev.getValue()+"|");
                } else {
                    System.out.print("|__"+prev.getValue()+"|");
                }

            } else {
                System.out.print( s: "|___|");
            }
        }
        System.out.println( x: "");
    }
    System.out.println( x: "-----------------------------------------");
}
```

5) Get Grid String

It’s the same as the print grid method but here I used string builder after getting the permission to use it and I return a string contain the game grid to use it in Gui

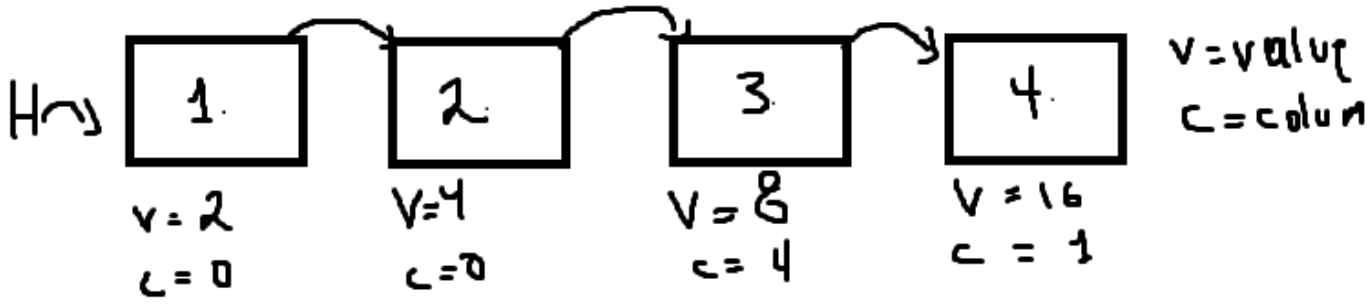
6) Merge Nodes

In this method I wanted to get the last two nodes in the wanted column and check if there equal it going to merge the last nod and delete it the previous node value will be value x2 and repeat this operation until no equal nodes found

7) the main class and GUI

I used an array after getting the permission to use it to keep the scenario data and add them one by one inside the GUI I have a button to add the next node and a text table to print the game grid string and in the console also all values and every change to the grid is printed

List Diagram

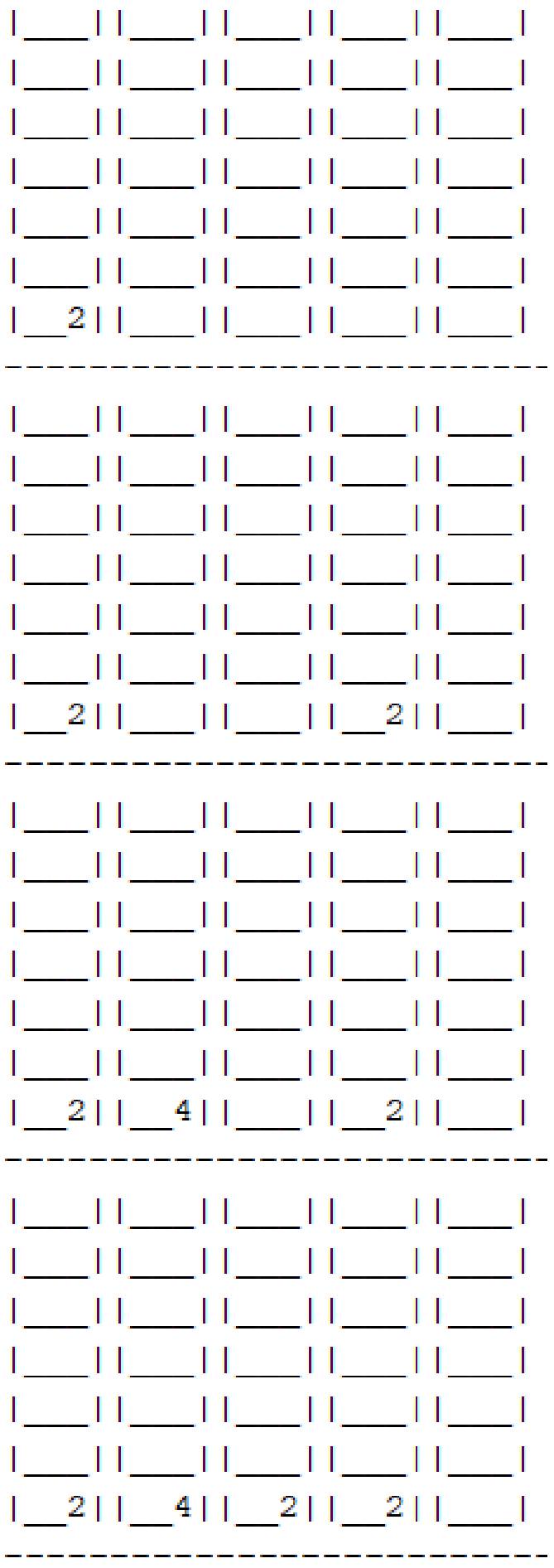


This how my list work I add the values next to each other

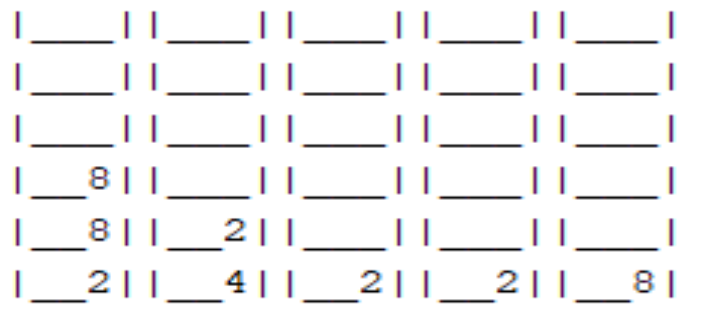
This is what the look like when its printing them or doing an operation on them like merge

My list and node design are simple because all the work and the sort is inside the methods I struggle with them but in the end I create a nice algorithm to deal with the list

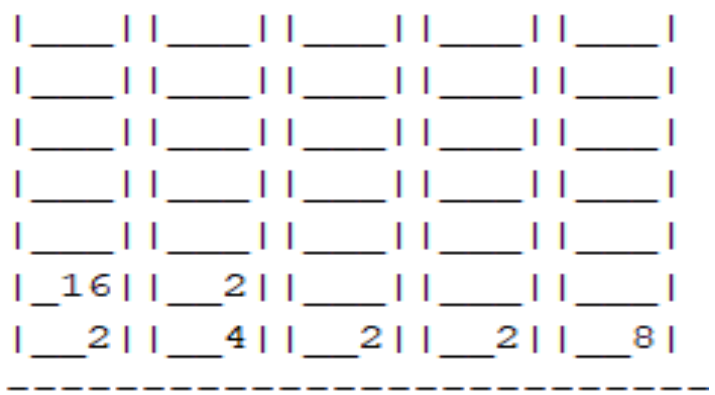
Output

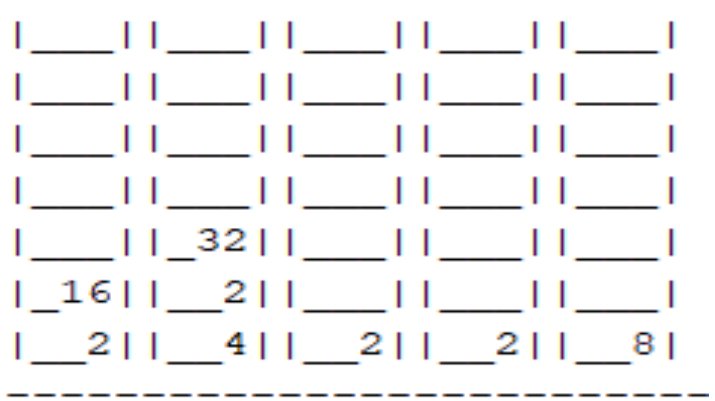


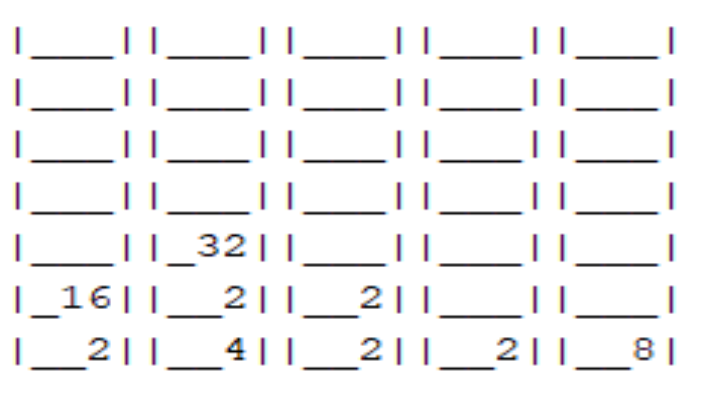
```
|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|__2||__4||__2||__2||__4|
----------------------------
|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|____||__2||____||____||____|
|__2||__4||__2||__2||__4|
----------------------------
|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|____||__2||____||____||__4|
|__2||__4||__2||__2||__4|
----------------------------
in column 4 merge 4->4
|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|____||__2||____||____||____|
|__2||__4||__2||__2||__8|
----------------------------
|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|__8||__2||____||____||____|
|__2||__4||__2||__2||__8|
----------------------------
```



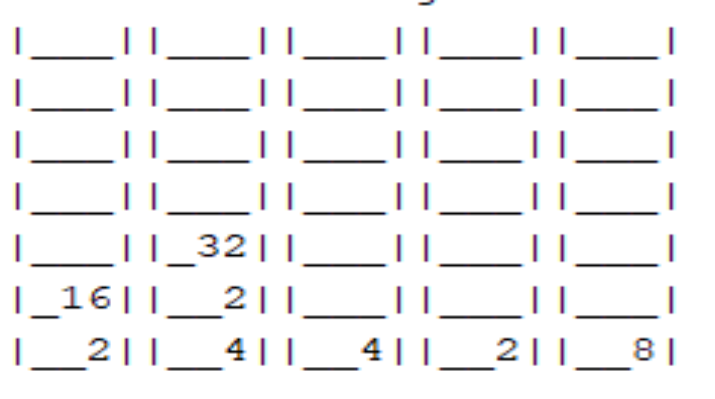
```
in column 0 merge 8->8
```







```
in column 2 merge 2->2
```



```
|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|____||_32||____||____||____|
|_16||__2||_64||____||____|
|__2||__4||__4||__2||__8|
--------------------------

|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|____||_32||____||____||____|
|_16||__2||_64||_16||____|
|__2||__4||__4||__2||__8|
--------------------------

|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|____||_64||____||____||____|
|____||_32||____||____||____|
|_16||__2||_64||_16||____|
|__2||__4||__4||__2||__8|
--------------------------

|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|____||_64||____||____||____|
|____||_32||_32||____||____|
|_16||__2||_64||_16||____|
|__2||__4||__4||__2||__8|
--------------------------

|____||____||____||____||____|
|____||____||____||____||____|
|____||____||____||____||____|
|____||_64||____||____||____|
|_16||_32||_32||____||____|
|_16||__2||_64||_16||____|
|__2||__4||__4||__2||__8|
--------------------------
```

```
in column 0 merge 16->16
|___||___||___||___||___|
|___||___||___||___||___|
|___||___||___||___||___|
|___||_64||___||___||___|
|___||_32||_32||___||___|
|_32||__2||_64||_16||___|
|__2||__4||__4||__2||__8|
----------------------------
|___||___||___||___||___|
|___||___||___||___||___|
|___||___||___||___||___|
|___||_64||___||___||___|
|___||_32||_32||___||___|
|_32||__2||_64||_16||_16|
|__2||__4||__4||__2||__8|
----------------------------
|___||___||___||___||___|
|___||___||___||___||___|
|___||___||___||___||___|
|___||_64||_32||___||___|
|___||_32||_32||___||___|
|_32||__2||_64||_16||_16|
|__2||__4||__4||__2||__8|
----------------------------
in column 2 merge 32->32
|___||___||___||___||___|
|___||___||___||___||___|
|___||___||___||___||___|
|___||_64||___||___||___|
|___||_32||_64||___||___|
|_32||__2||_64||_16||_16|
|__2||__4||__4||__2||__8|
----------------------------
in column 2 merge 64->64
|___||___||___||___||___|
|___||___||___||___||___|
|___||___||___||___||___|
|___||_64||___||___||___|
|___||_32||___||___||___|
|_32||__2||128||_16||_16|
|__2||__4||__4||__2||__8|
----------------------------
```

```
|___||___||___||___||___|
|___||___||___||___||___|
|___||_64||___||___||___|
|___||_64||___||___||___|
|___||_32||___||___||___|
|_32||__2||128||_16||_16|
|__2||__4||__4||__2||__8|
--------------------------
in column 1 merge 64->64
|___||___||___||___||___|
|___||___||___||___||___|
|___||___||___||___||___|
|___||128||___||___||___|
|___||_32||___||___||___|
|_32||__2||128||_16||_16|
|__2||__4||__4||__2||__8|
--------------------------
|___||___||___||___||___|
|___||___||___||___||___|
|___||___||___||___||___|
|___||128||___||___||___|
|___||_32||___||__8||___|
|_32||__2||128||_16||_16|
|__2||__4||__4||__2||__8|
--------------------------
|___||___||___||___||___|
|___||___||___||___||___|
|___||___||___||___||___|
|___||128||___||__4||___|
|___||_32||___||__8||___|
|_32||__2||128||_16||_16|
|__2||__4||__4||__2||__8|
--------------------------
|___||___||___||___||___|
|___||___||___||___||___|
|___||___||___||__2||___|
|___||128||___||__4||___|
|___||_32||___||__8||___|
|_32||__2||128||_16||_16|
|__2||__4||__4||__2||__8|
--------------------------
```

```
|___||___||___||___||___|
|___||___||___||___||___|
|___||___||___||__2||___|
|___||128||___||__4||___|
|___||_32||___||__8||___|
|_32||__2||128||_16||_16|
|__2||__4||__4||__2||__8|
----------------------------
|___||___||___||___||___|
|___||___||___||__2||___|
|___||___||___||__2||___|
|___||128||___||__4||___|
|___||_32||___||__8||___|
|_32||__2||128||_16||_16|
|__2||__4||__4||__2||__8|
----------------------------
in column 3 merge 2->2
|___||___||___||___||___|
|___||___||___||___||___|
|___||___||___||__4||___|
|___||128||___||__4||___|
|___||_32||___||__8||___|
|_32||__2||128||_16||_16|
|__2||__4||__4||__2||__8|
----------------------------
in column 3 merge 4->4
|___||___||___||___||___|
|___||___||___||___||___|
|___||___||___||___||___|
|___||128||___||__8||___|
|___||_32||___||__8||___|
|_32||__2||128||_16||_16|
|__2||__4||__4||__2||__8|
----------------------------
in column 3 merge 8->8
|___||___||___||___||___|
|___||___||___||___||___|
|___||___||___||___||___|
|___||128||___||___||___|
|___||_32||___||_16||___|
|_32||__2||128||_16||_16|
|__2||__4||__4||__2||__8|
----------------------------
```

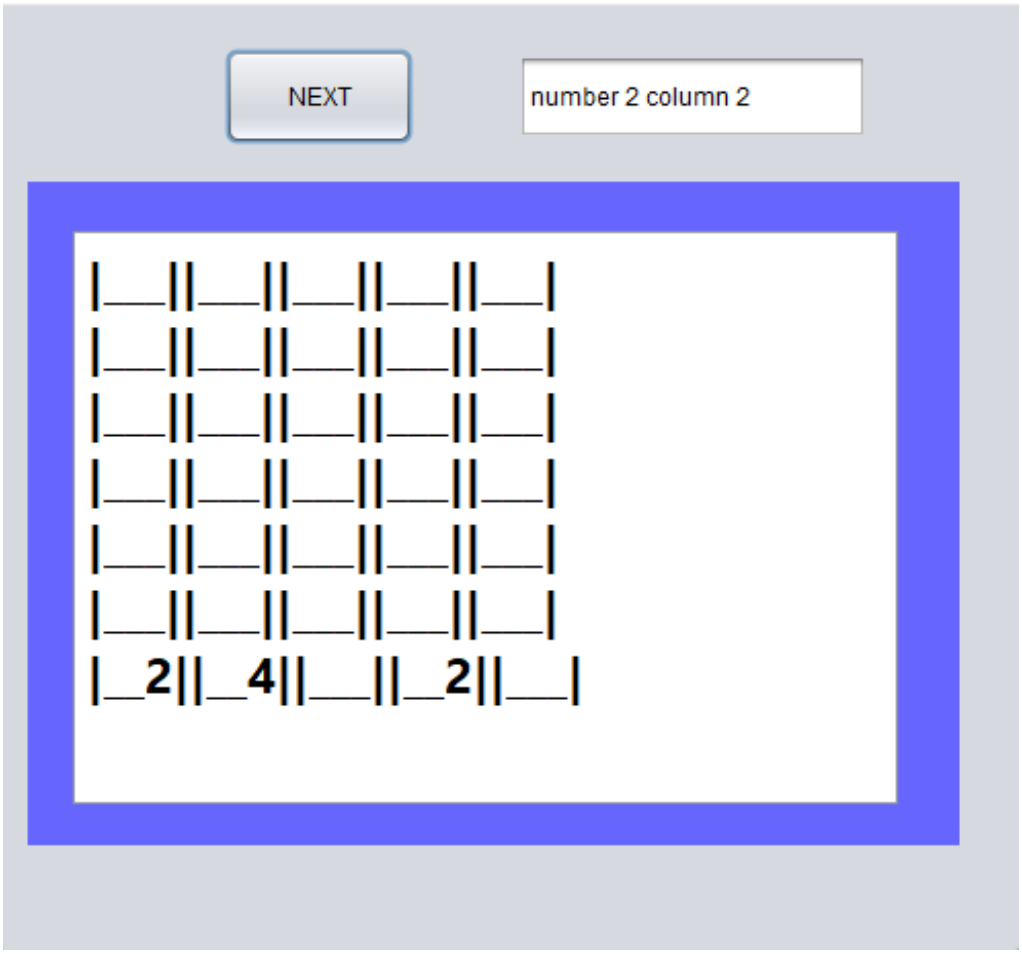
```
in column 3 merge 16->16
|___||___||___||___||___|
|___||___||___||___||___|
|___||___||___||___||___|
|___||128||___||___||___|
|___||_32||___||___||___|
|_32||__2||128||_32||_16|
|__2||__4||__4||__2||__8|
----------------------------
|___||___||___||___||___|
|___||___||___||___||___|
|___||__2||___||___||___|
|___||128||___||___||___|
|___||_32||___||___||___|
|_32||__2||128||_32||_16|
|__2||__4||__4||__2||__8|
----------------------------
|___||___||___||___||___|
|___||___||___||___||___|
|___||__2||___||___||___|
|___||128||___||___||___|
|___||_32||_64||___||___|
|_32||__2||128||_32||_16|
|__2||__4||__4||__2||__8|
----------------------------
|___||___||___||___||___|
|___||___||___||___||___|
|___||__2||___||___||___|
|___||128||_32||___||___|
|___||_32||_64||___||___|
|_32||__2||128||_32||_16|
|__2||__4||__4||__2||__8|
----------------------------
|___||___||___||___||___|
|___||___||___||___||___|
|___||__2||_16||___||___|
|___||128||_32||___||___|
|___||_32||_64||___||___|
|_32||__2||128||_32||_16|
|__2||__4||__4||__2||__8|
----------------------------
```

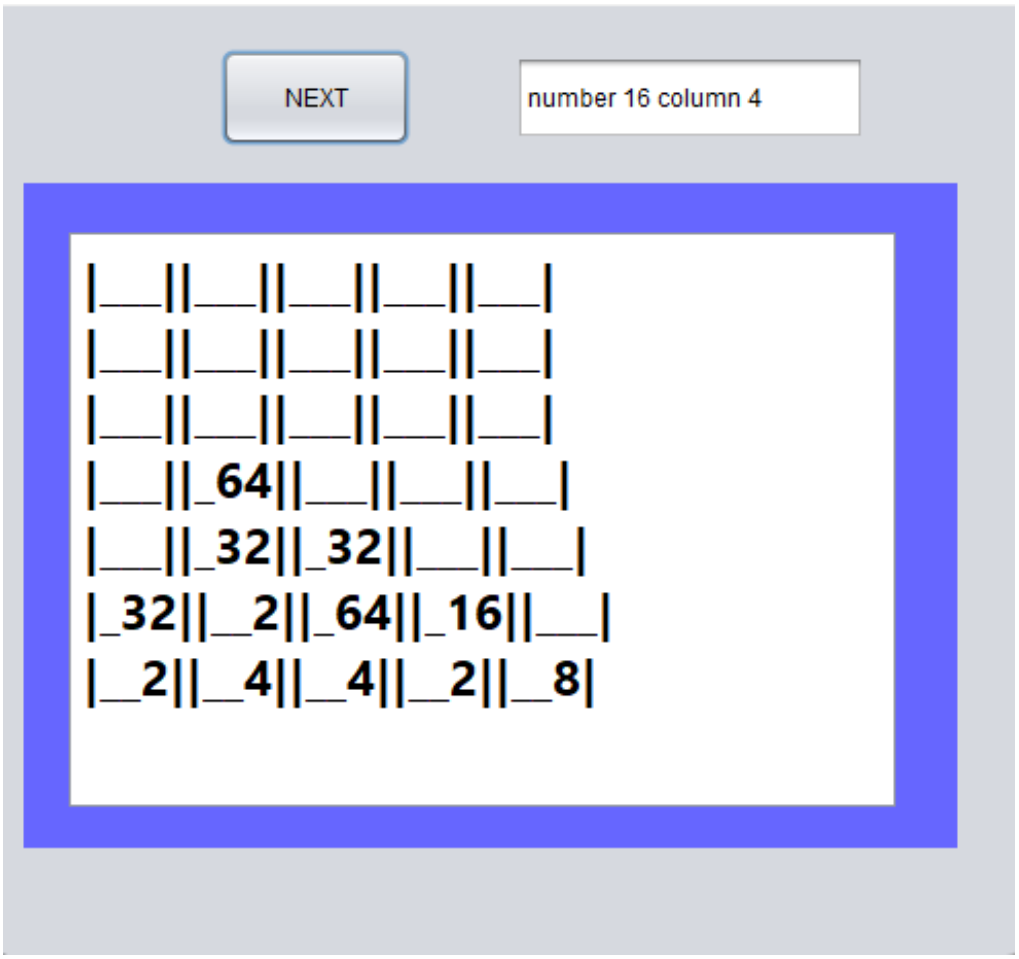
```
|___||___||___||___||___|
|___||___||__8||___||___|
|___||__2||_16||___||___|
|___||128||_32||___||___|
|___||_32||_64||___||___|
|_32||__2||128||_32||_16|
|__2||__4||__4||__2||__8|
--------------------------
|___||___||__8||___||___|
|___||___||__8||___||___|
|___||__2||_16||___||___|
|___||128||_32||___||___|
|___||_32||_64||___||___|
|_32||__2||128||_32||_16|
|__2||__4||__4||__2||__8|
--------------------------
in column 2 merge 8->8
|___||___||___||___||___|
|___||___||_16||___||___|
|___||__2||_16||___||___|
|___||128||_32||___||___|
|___||_32||_64||___||___|
|_32||__2||128||_32||_16|
|__2||__4||__4||__2||__8|
--------------------------
in column 2 merge 16->16
|___||___||___||___||___|
|___||___||___||___||___|
|___||__2||_32||___||___|
|___||128||_32||___||___|
|___||_32||_64||___||___|
|_32||__2||128||_32||_16|
|__2||__4||__4||__2||__8|
--------------------------
in column 2 merge 32->32
|___||___||___||___||___|
|___||___||___||___||___|
|___||__2||___||___||___|
|___||128||_64||___||___|
|___||_32||_64||___||___|
|_32||__2||128||_32||_16|
|__2||__4||__4||__2||__8|
--------------------------
```

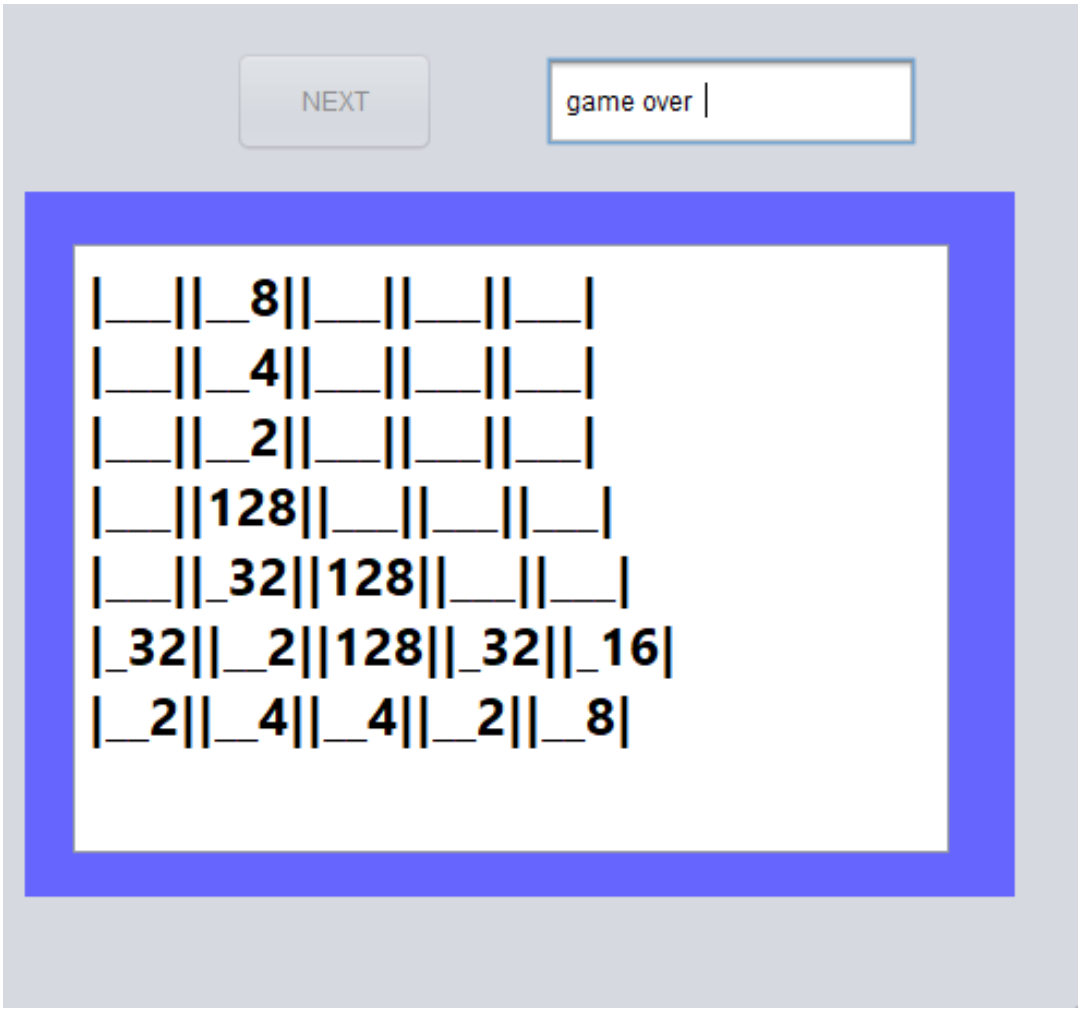
```
in column 2 merge 64->64
|___||___||___||___||___|
|___||___||___||___||___|
|___||__2||___||___||___|
|___||128||___||___||___|
|___||_32||128||___||___|
|_32||__2||128||_32||_16|
|__2||__4||__4||__2||__8|
----------------------------
|___||___||___||___||___|
|___||__4||___||___||___|
|___||__2||___||___||___|
|___||128||___||___||___|
|___||_32||128||___||___|
|_32||__2||128||_32||_16|
|__2||__4||__4||__2||__8|
----------------------------
|___||__8||___||___||___|
|___||__4||___||___||___|
|___||__2||___||___||___|
|___||128||___||___||___|
|___||_32||128||___||___|
|_32||__2||128||_32||_16|
|__2||__4||__4||__2||__8|
----------------------------
```

End

And the same for the GUI






NEXT
game over
__		_8		__		__		__
__		_4		__		__		__
__		_2		__		__		__
__		128		__		__		__
__		_32		128		__		__
_32		_2		128		_32		_16
_2		_4		_4		_2		_8