Data Structures and Algorithms

COMP 103 2019-20 Semester 2

Lecture 10a

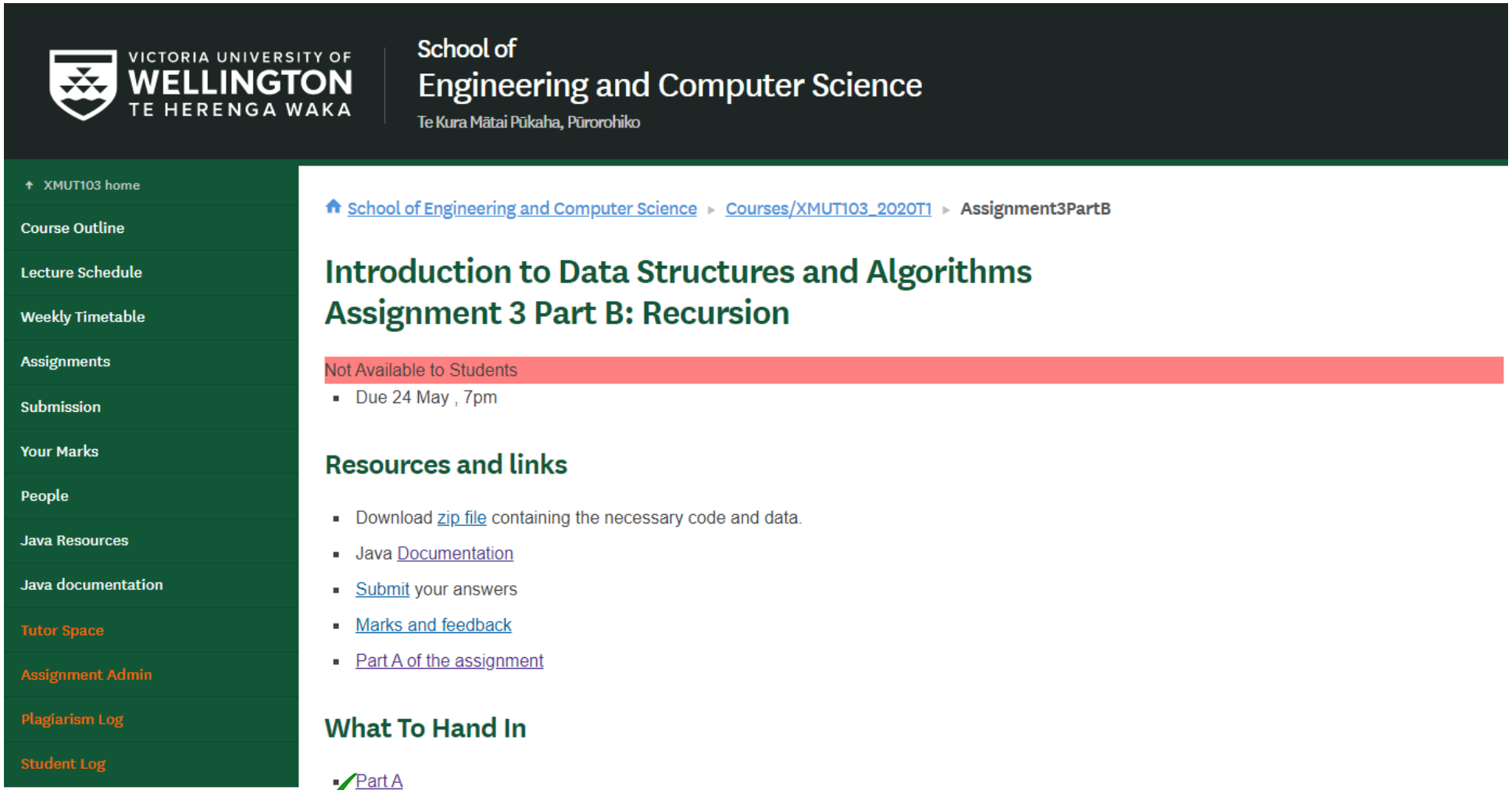
Dr. Kerese Manueli

kerese.manueli@ecs.vuw.ac.nz

Victoria University of Wellington

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB



VICTORIA UNIVERSITY OF WELLINGTON TE HERENGA WAKA

School of Engineering and Computer Science
Te Kura Mātai Pūkaha, Pūrorohiko

- XMUT103 home
- Course Outline
- Lecture Schedule
- Weekly Timetable
- Assignments
- Submission
- Your Marks
- People
- Java Resources
- Java documentation
- Tutor Space
- Assignment Admin
- Plagiarism Log
- Student Log

School of Engineering and Computer Science ▸ Courses/XMUT103_2020T1 ▸ Assignment3PartB

Introduction to Data Structures and Algorithms
Assignment 3 Part B: Recursion

Not Available to Students

- Due 24 May , 7pm

Resources and links

- Download zip file containing the necessary code and data.
- Java Documentation
- Submit your answers
- Marks and feedback
- Part A of the assignment

What To Hand In

- Part A
  - `HospitalERCore.java`
  - `Patient.java`
  - `HospitalERCompl.java` and `Department.java` (for the completion)
- Part B
  - `MineSweeper.java`

Assignment 3b

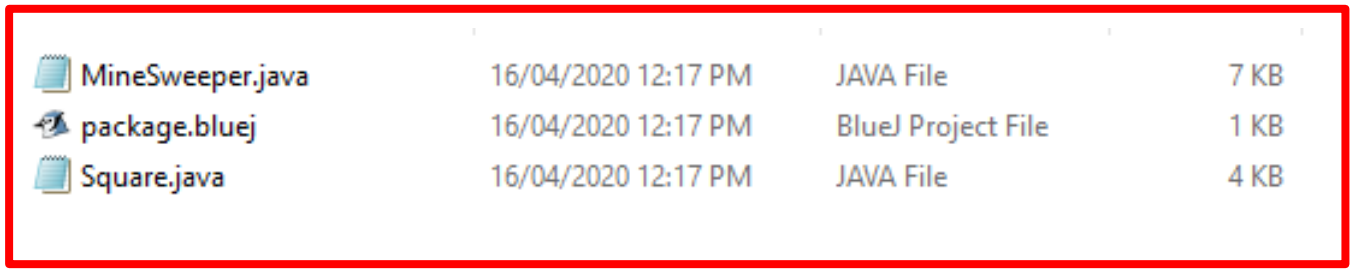
https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

1) Zip file → XMUT103-2020T1-Assig3PartB.zip 19/05/2020 8:08 AM Compressed (zipp... 5 KB

2) Unzipped folder → XMUT103-2020T1-Assig3PartB 19/05/2020 8:12 AM File folder

3) Folder inside the unzipped folder MineSweeper 16/04/2020 12:17 PM File folder

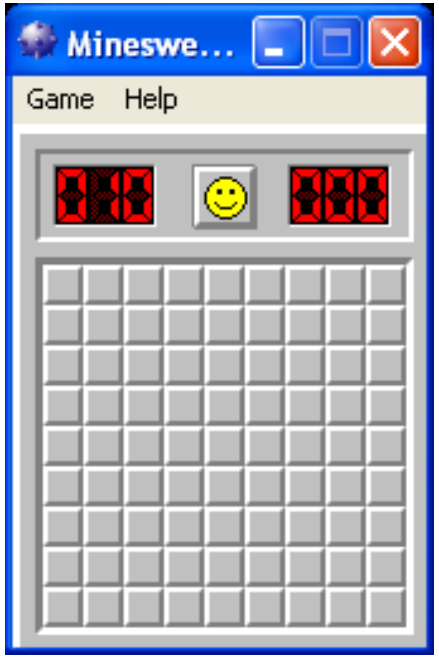
4) Files inside the **MineSweeper** folder



Minesweeper Game (Weight: 1/3)

Minesweeper is a very old video game in which the player tries to work out where the mines are hidden on a grid. Each square in the grid starts off as hidden (dark green). The player can mark any hidden square that they think contains a mine, and can expose any hidden square that they think is safe.

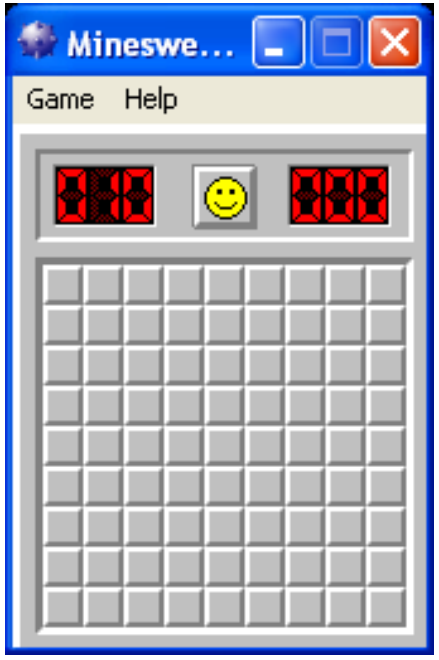
- If the player exposes a square with a mine, they lose and the game is over. In this case, all the squares are opened up so that the player can see where all the mines are.
- If the player exposes a safe square that is next to one or more mines, then the square displays a number saying how many of the eight surrounding squares contain a mine.
- If the player exposes a safe square that is not next to a mine, then it "spreads" to all the connected squares that are also safe, and exposes them also.



Minesweeper Game (Weight: 1/3)

Minesweeper is a very old video game in which the player tries to work out where the mines are hidden on a grid. Each square in the grid starts off as hidden (dark green). The player can mark any hidden square that they think contains a mine, and can expose any hidden square that they think is safe.

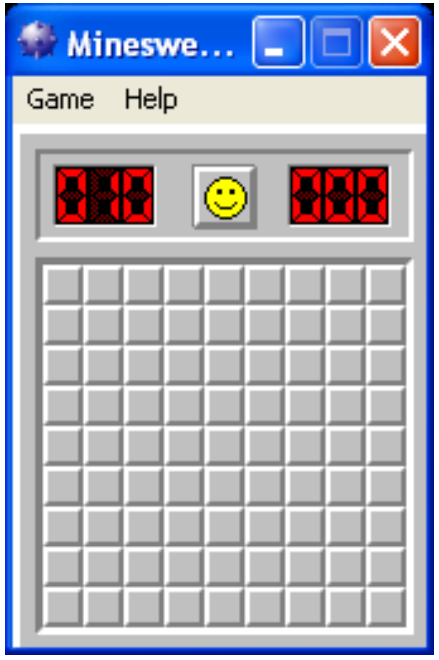
- If the player exposes a square with a mine, they lose and the game is over. In this case, all the squares are opened up so that the player can see where all the mines are.
- If the player exposes a safe square that is next to one or more mines, then the square displays a number saying how many of the eight surrounding squares contain a mine.
- If the player exposes a safe square that is not next to a mine, then it "spreads" to all the connected squares that are also safe, and exposes them also.



Minesweeper Game (Weight: 1/3)

Minesweeper is a very old video game in which the player tries to work out where the mines are hidden on a grid. Each square in the grid starts off as hidden (dark green). The player can mark any hidden square that they think contains a mine, and can expose any hidden square that they think is safe.

- If the player exposes a square with a mine, they lose and the game is over. In this case, all the squares are opened up so that the player can see where all the mines are.
- If the player exposes a safe square that is next to one or more mines, then the square displays a number saying how many of the eight surrounding squares contain a mine.
- If the player exposes a safe square that is not next to a mine, then it "spreads" to all the connected squares that are also safe, and exposes them also.



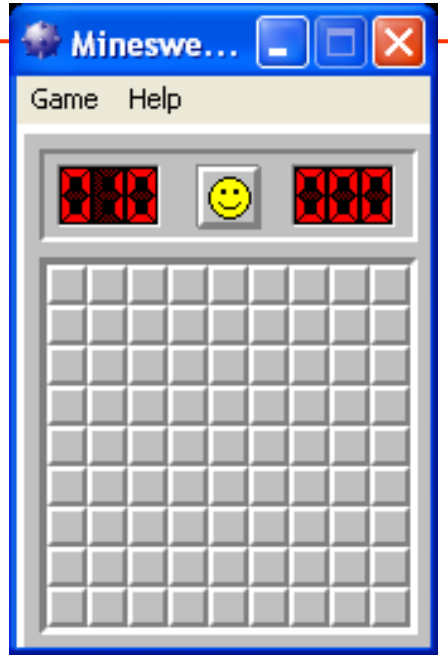
Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

Minesweeper Game (Weight: 1/3)

Minesweeper is a very old video game in which the player tries to work out where the mines are hidden on a grid. Each square in the grid starts off as hidden (dark green). The player can mark any hidden square that they think contains a mine, and can expose any hidden square that they think is safe.

- If the player exposes a square with a mine, they lose and the game is over. In this case, all the squares are opened up so that the player can see where all the mines are.
- If the player exposes a safe square that is next to one or more mines, then the square displays a number saying how many of the eight surrounding squares contain a mine.
- If the player exposes a safe square that is not next to a mine, then it "spreads" to all the connected squares that are also safe, and exposes them also.



Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

If you aren't familiar with the game, you can run the demo, or play the KDE version (KMines - in the Games menu, under Tactics and Strategy).

The `Minesweeper` class has code for most of the program, except for the key methods of

- `mark(row, col)`: The player has clicked on a square to mark it.
- `tryExpose(row, col)`: the player has clicked on a square to expose it
- `exposeSquareAt(row, col)`: expose the square at (row,col) and if it has no adjacent mines, spread the exposing to all connected safe squares. This method should be recursive, and needs to spread out horizontally, vertically, and diagonally.
- `hasWon()`: check whether the player has won the game yet (exposed all the squares without a mine).

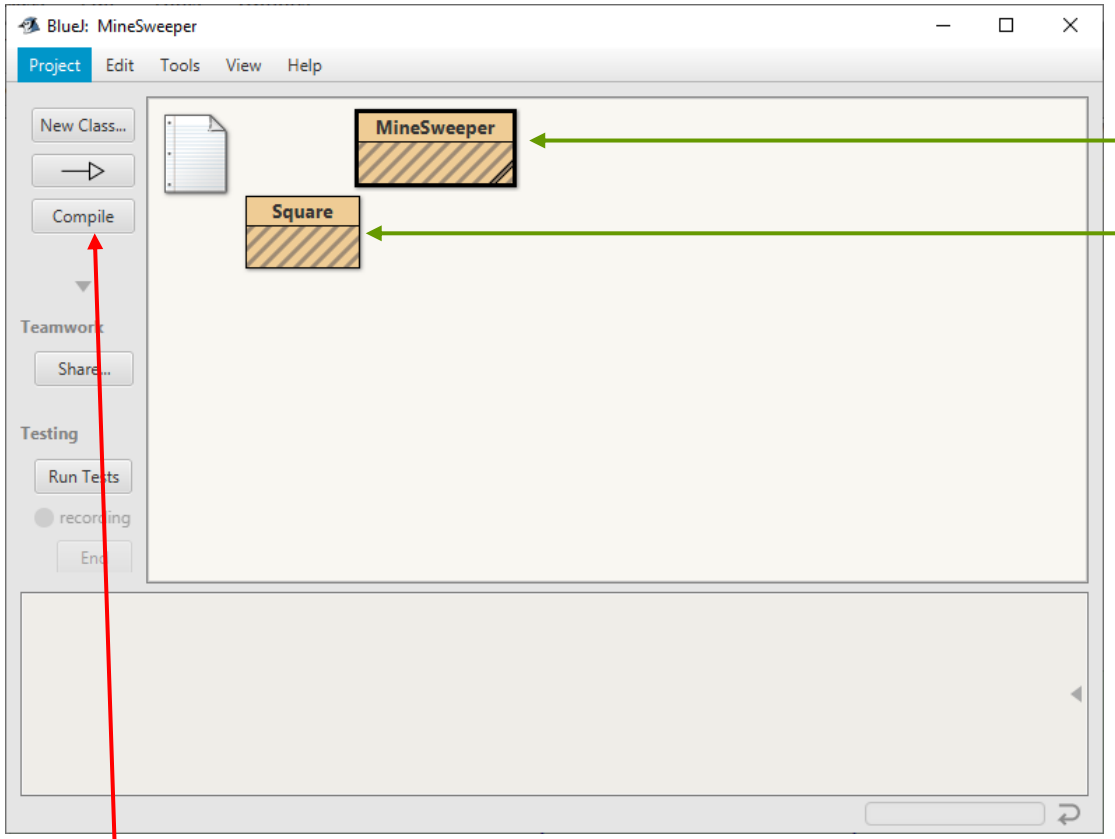
The `Square` class has code for individual squares; you need to use it, but you should not modify it.

Core and Completion:

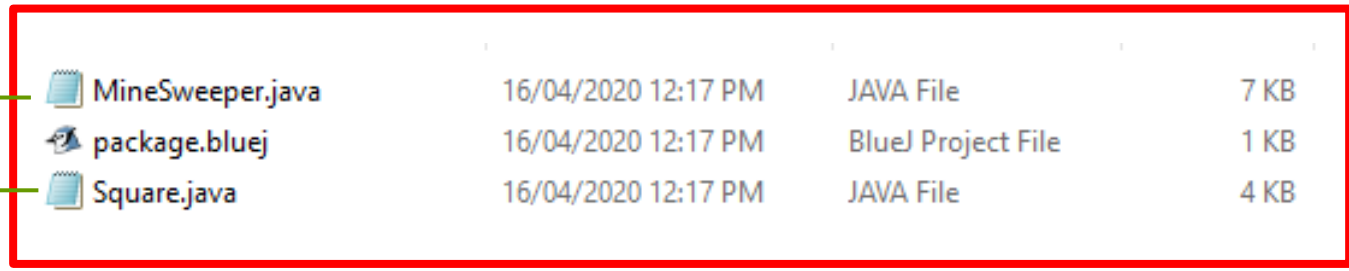
Complete the four incomplete methods in `Minesweeper` in order to make the game work.

Assignment 3b

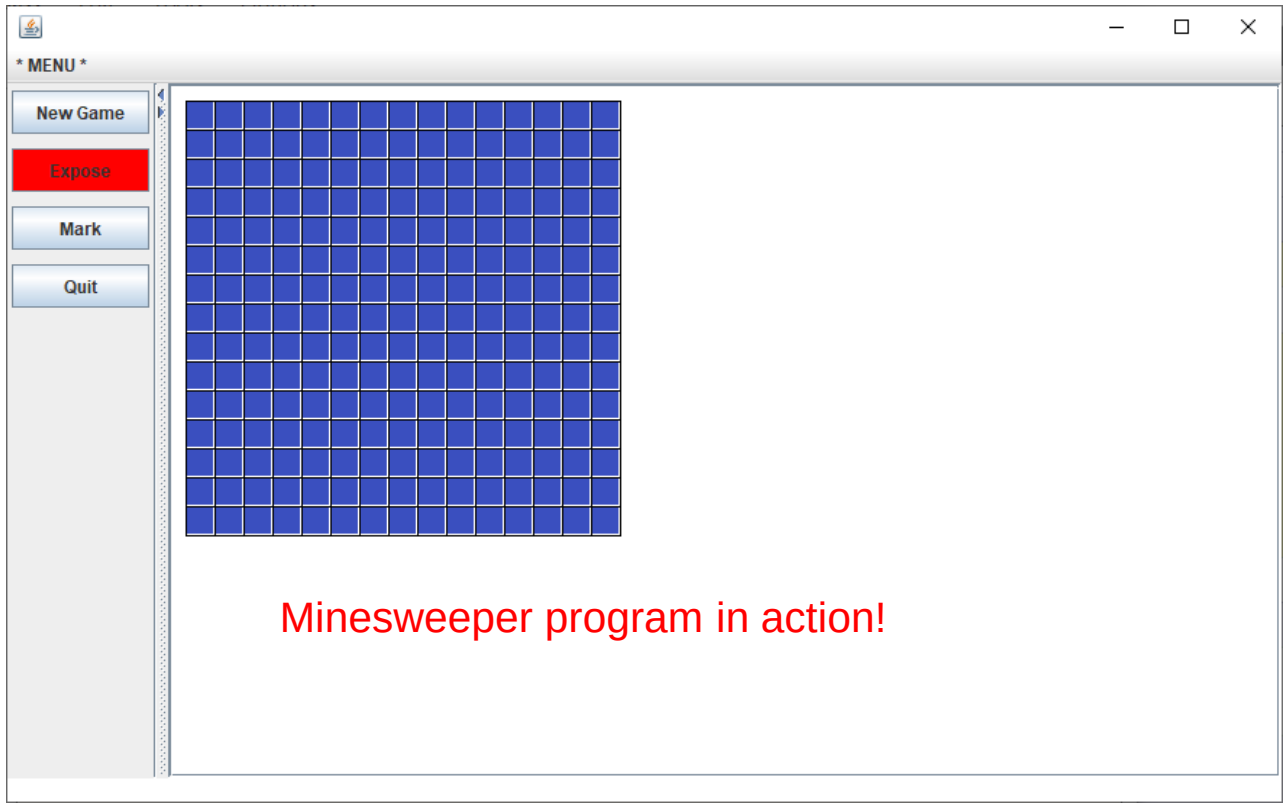
https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB



Files inside the **MineSweeper** folder



Compile and run the Minesweeper program

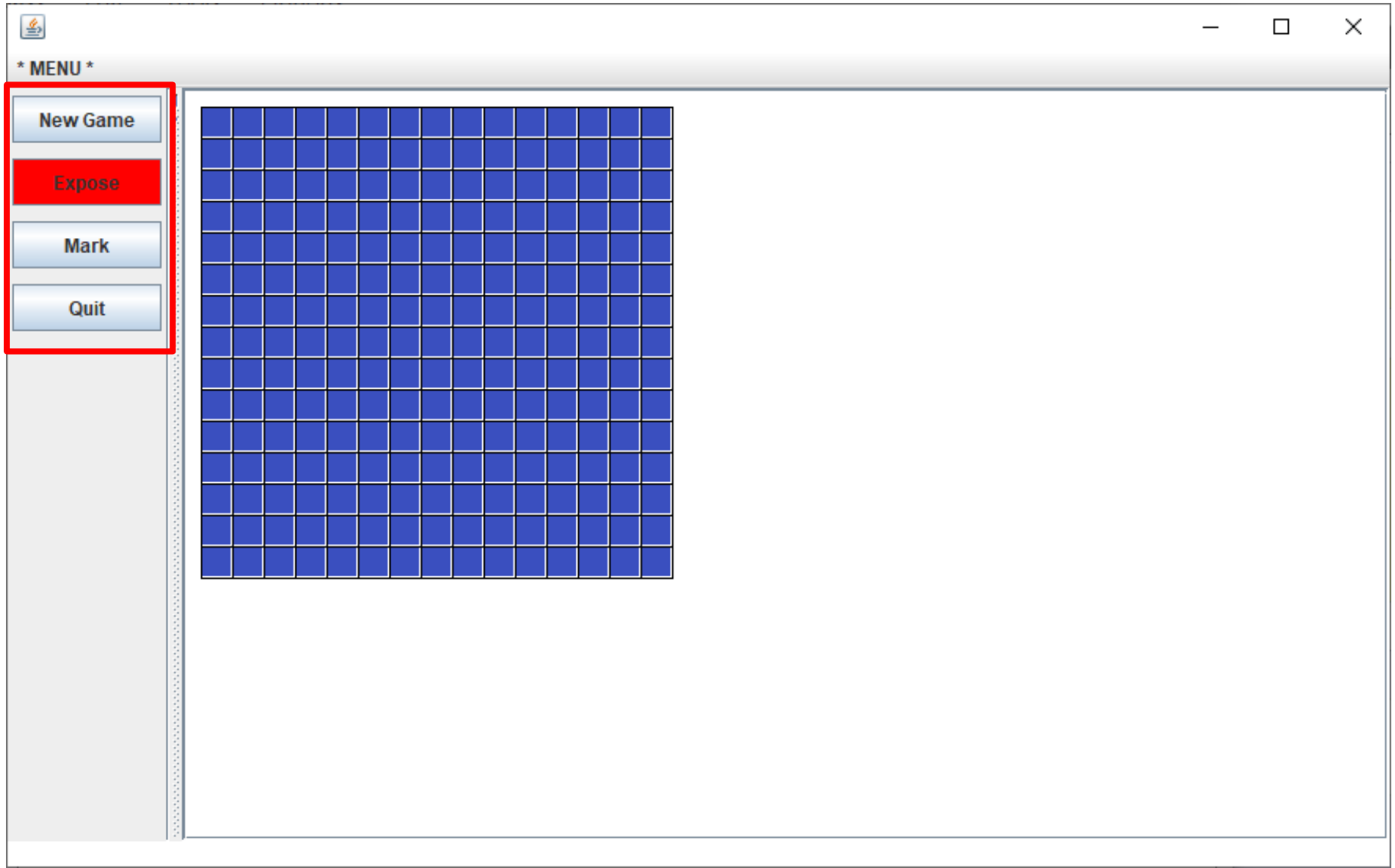


Minesweeper program in action!

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

4
Buttons



Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

MineSweeper - MineSweeper

Class Edit Tools Options

MineSweeper

Compile Undo Cut Copy Paste Find... Close Source Code

```
1  // This program is copyright VUW.
2  // You are granted permission to use it to construct your answer to a XMUT103 assignment.
3  // You may not distribute it in any other way without permission.
4
5  /* Code for XMUT103 - 2020T1, Assignment 3
6   * Name:
7   * Username:
8   * ID:
9   */
10
11 import ecs100.*;
12 import java.awt.Color;
13 import javax.swing.JButton;
14
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

MineSweeper - MineSweeper

Class Edit Tools Options

MineSweeper

Compile Undo Cut Copy Paste Find... Close Source Code

```
15 /**
16  *  Simple 'Minesweeper' program.
17  *  There is a grid of squares, some of which contain a mine.
18  *  The user can click on a square to either expose it or to
19  *  mark/unmark it.
20  *
21  *  If the user exposes a square with a mine, they lose.
22  *  Otherwise, it is uncovered, and shows a number which represents the
23  *  number of mines in the eight squares surrounding that one.
24  *  If there are no mines adjacent to it, then all the unexposed squares
25  *  immediately adjacent to it are exposed (and and so on)
26  *
27  *  If the user marks a square, then they cannot expose the square,
28  *  (unless they unmark it first)
29  *  When all squares with mines are marked, and all the squares without
30  *  mines are exposed, the user has won.
31  */
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

MineSweeper - MineSweeper

Class Edit Tools Options

MineSweeper

Compile Undo Cut Copy Paste Find... Close Source Code

```
32 public class MineSweeper {
33
34     public static final int ROWS = 15;
35     public static final int COLS = 15;
36
37     public static final double LEFT = 10;
38     public static final double TOP = 10;
39     public static final double SQUARE_SIZE = 20;
40
41     // Fields
42     private boolean marking;
43
44     private Square[][] squares;
45
46     private JButton mrkButton;
47     private JButton expButton;
48     Color defaultColor;
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

MineSweeper - MineSweeper

Class Edit Tools Options

MineSweeper

Compile Undo Cut Copy Paste Find... Close Source Code

```
50  /**
51   * Construct a new MineSweeper object
52   * and set up the GUI
53   */
54  public static void main(String[] arguments){
55      MineSweeper ms = new MineSweeper();
56      ms.setupGUI();
57      ms.setMarking(false);
58      ms.makeGrid();
59  }
60
61  /** Set up the GUI: buttons and mouse to play the game */
62  public void setupGUI(){
63      UI.setMouseListener(this::doMouse);
64      UI.addButton("New Game", this::makeGrid);
65      this.expButton = UI.addButton("Expose", ()->setMarking(false));
66      this.mrkButton = UI.addButton("Mark", ()->setMarking(true));
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

MineSweeper - MineSweeper

Class Edit Tools Options

MineSweeper

Compile Undo Cut Copy Paste Find... Close Source Code

```
67
68      UI.addButton("Quit", UI::quit);
69      UI.setDivider(0.0);
70  }
71
72  /** Respond to mouse events */
73  public void doMouse(String action, double x, double y) {
74      if (action.equals("released")){
75          int row = (int)((y-TOP)/SQUARE_SIZE);
76          int col = (int)((x-LEFT)/SQUARE_SIZE);
77          if (row>=0 && row < ROWS && col >= 0 && col < COLS){
78              if (marking) { mark(row, col);}
79              else         { tryExpose(row, col); }
80          }
81      }
82  }
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

MineSweeper - MineSweeper

Class Edit Tools Options

MineSweeper

Compile Undo Cut Copy Paste Find... Close Source Code

```
83
84      /**
85       * Remember whether it is "Mark" or "Expose"
86       * Change the colour of the "Mark", "Expose" buttons
87       */
88      public void setMarking(boolean v){
89          marking=v;
90          if (marking) {
91              mrkButton.setBackground(Color.red);
92              expButton.setBackground(null);
93          }
94          else {
95              expButton.setBackground(Color.red);
96              mrkButton.setBackground(null);
97          }
98      }
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

MineSweeper - MineSweeper

Class Edit Tools Options

MineSweeper

Compile Undo Cut Copy Paste Find... Close Source Code

```
/**
 * The player has clicked on a square to expose it
 * - if it is already exposed or marked, do nothing.
 * - if it's a mine: lose (call drawLose())
 * - otherwise expose it (call exposeSquareAt)
 * then check to see if the player has won and call drawWon() if they have.
 * (This method is not recursive)
 */
public void tryExpose(int row, int col){
    /*# YOUR CODE HERE */

    if (hasWon()){
        drawWin();
    }
}
```

You need to complete this method (1)

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

MineSweeper - MineSweeper

Class Edit Tools Options

MineSweeper

Compile Undo Cut Copy Paste Find... Close Source Code

```
117
118  /**
119   *  Expose a square, and spread to its neighbours if safe to do so.
120   *  It is guaranteed that this square is safe to expose (ie, does not have a mine).
121   *  If it is already exposed, we are done.
122   *  Otherwise expose it, and redraw it.
123   *  If the number of adjacent mines of this square is 0, then
124   *     expose all its neighbours (which are safe to expose)
125   *     (and if they have no adjacent mine, expose their neighbours, and ....)
126   */
127  public void exposeSquareAt(int row, int col){
128      /*# YOUR CODE HERE */
129
130  }
```

You need to complete this method (2)

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

MineSweeper - MineSweeper

Class Edit Tools Options

MineSweeper

Compile Undo Cut Copy Paste Find... Close Source Code

```
131
132     /**
133      * Mark/unmark the square.
134      * If the square is exposed, don't do anything,
135      * If it is marked, unmark it.
136      * otherwise mark it and redraw.
137      * (Marking cannot make the player win or lose)
138      */
139     public void mark(int row, int col){
140         /*# YOUR CODE HERE */
141
142     }
```

You need to complete this method (3)

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

MineSweeper - MineSweeper

Class Edit Tools Options

MineSweeper

Compile Undo Cut Copy Paste Find... Close Source Code

```
143
144 /**
145  * Returns true if the player has won:
146  * If all the squares without a mine have been exposed, then the player has won.
147  */
148 public boolean hasWon(){
149     /*# YOUR CODE HERE */
150
151     return true;
152 }
```

You need to complete this method (4)

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

MineSweeper - MineSweeper

Class Edit Tools Options

MineSweeper

Compile Undo Cut Copy Paste Find... Close Source Code

```
153
154     /**
155      * Construct a grid with random mines.
156      */
157     public void makeGrid(){
158         UI.clearGraphics();
159         this.squares = new Square[ROWS][COLS];
160         for (int row=0; row < ROWS; row++){
161             double y = TOP+row*SQUARE_SIZE;
162             for (int col=0; col<COLS; col++){
163                 double x =LEFT+col*SQUARE_SIZE;
164                 boolean isMine = Math.random()<0.1;     // approx 1 in 10 squares is a mine
165                 this.squares[row][col] = new Square(isMine);
166                 this.squares[row][col].draw(x, y, SQUARE_SIZE);
167             }
168         }
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

MineSweeper - MineSweeper

Class Edit Tools Options

MineSweeper

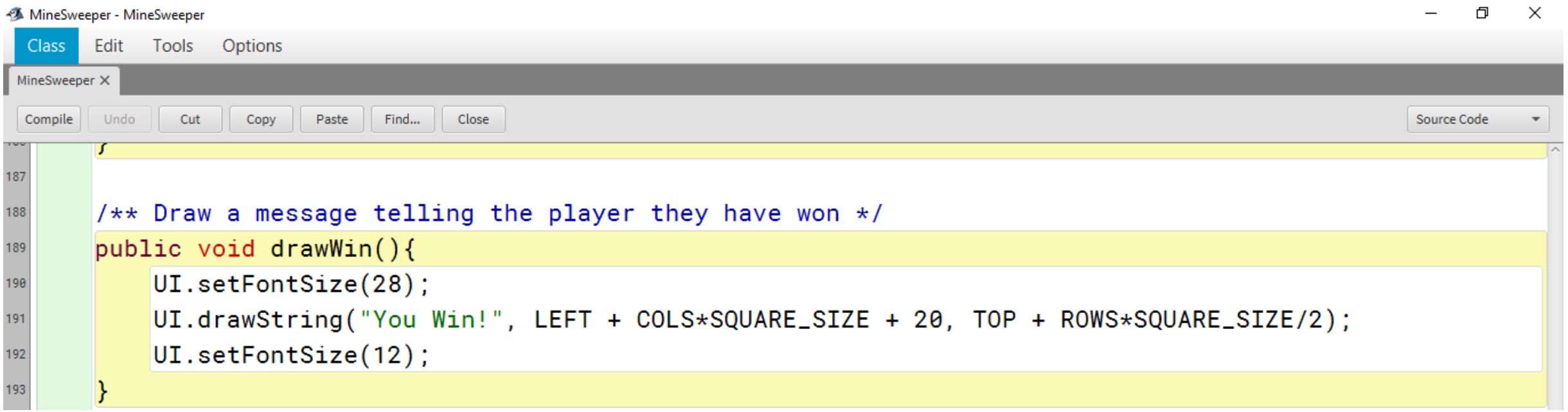
Compile Undo Cut Copy Paste Find... Close Source Code

```
}
// now compute the number of adjacent mines for each square
for (int row=0; row<ROWS; row++){
    for (int col=0; col<COLS; col++){
        int count = 0;
        //look at each square in the neighbourhood.
        for (int r=Math.max(row-1,0); r<Math.min(row+2, ROWS); r++){
            for (int c=Math.max(col-1,0); c<Math.min(col+2, COLS); c++){
                if (squares[r][c].hasMine())
                    count++;
            }
        }
        if (this.squares[row][col].hasMine())
            count--;  // we weren't suppose to count this square, just the adjacent ones.

        this.squares[row][col].setAdjacentMines(count);
    }
}
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB



Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

MineSweeper - MineSweeper

Class Edit Tools Options

MineSweeper

Compile Undo Cut Copy Paste Find... Close Source Code

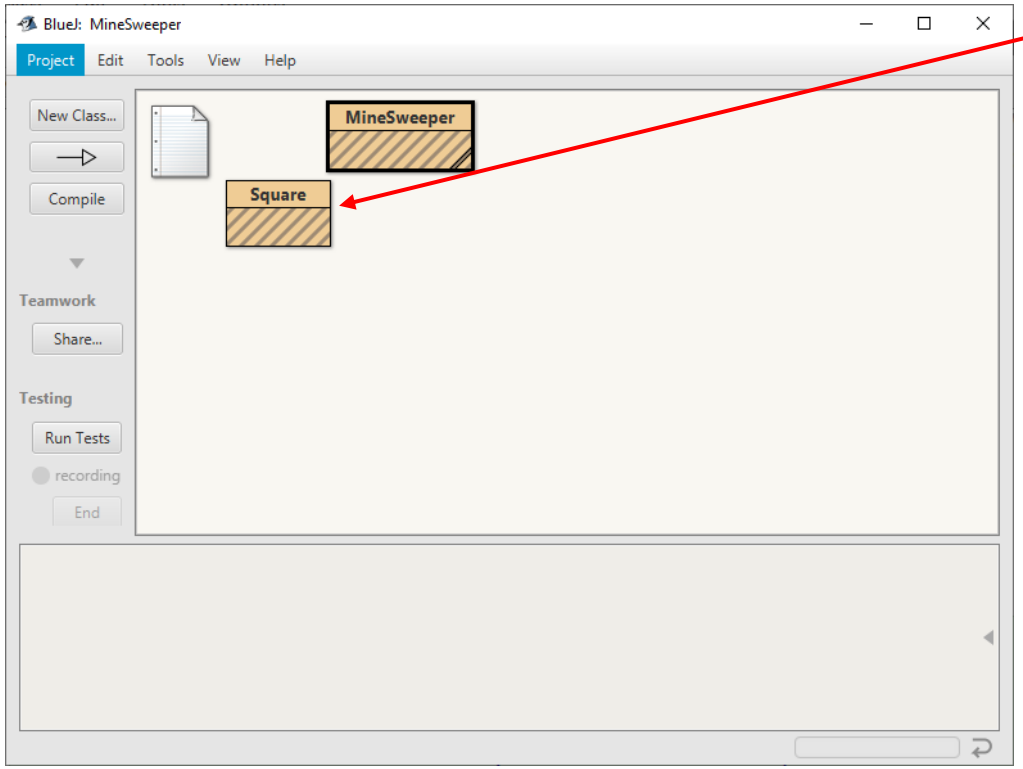
```
194
195     /**
196      * Draw a message telling the player they have lost
197      * and expose all the squares and redraw them
198      */
199     public void drawLose(){
200         for (int row=0; row<ROWS; row++){
201             for (int col=0; col<COLS; col++){
202                 squares[row][col].setExposed();
203                 squares[row][col].draw(LEFT+col*SQUARE_SIZE, TOP+row*SQUARE_SIZE, SQUARE_SIZE);
204             }
205         }
206         UI.setFontSize(28);
207         UI.drawString("You Lose!", LEFT + COLS*SQUARE_SIZE+20, TOP + ROWS*SQUARE_SIZE/2);
208         UI.setFontSize(12);
209     }
210
211 }
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

Files inside the **MineSweeper** folder

MineSweeper.java	16/04/2020 12:17 PM	JAVA File	7 KB
package.bluej	16/04/2020 12:17 PM	BlueJ Project File	1 KB
Square.java	16/04/2020 12:17 PM	JAVA File	4 KB



Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

Square - MineSweeper

Class Edit Tools Options

MineSweeper × Square ×

Compile Undo Cut Copy Paste Find... Close Source Code

```
1  // This program is copyright VUW.
2  // You are granted permission to use it to construct your answer to a XMUT103 assignment.
3  // You may not distribute it in any other way without permission.
4
5  /* Code for XMUT103 - 2020T1, Assignment 3
6   * Name:
7   * Username:
8   * ID:
9   */
10
11 import java.util.*;
12 import ecs100.*;
13 import java.awt.Color;
14 import java.io.*;
15
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

Square - MineSweeper

Class Edit Tools Options

MineSweeper × Square ×

Compile Undo Cut Copy Paste Find... Close Source Code

```
15
16 /** Represents information about one square in a MineSweeper map.
17  *   It records
18  *     its location (Row and Column)
19  *     whether it has a mine or not
20  *     how many squares around it have mines
21  *     whether it is marked or unmarked
22  *     whether it is hidden or exposed
23  *   Its constructor must specify the location and
24  *     whether it has a mine or not
25  *   It has methods to
26  *     draw itself, (showing its state appropriately) given origin of the map.
27  *     set the number of mines around it.
28  *     report whether it has a mine and whether it is exposed
29  *     change its state between marked and unmarked
30  *     change its state to exposed
31  */
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

Square - MineSweeper

Class Edit Tools Options

MineSweeper × Square ×

Compile Undo Cut Copy Paste Find... Close Source Code

```
*/
public class Square{
    // Fields
    private boolean mine;
    private int adjacentMines = 0;
    private boolean marked = false;
    private boolean exposed = false;

    public static final Color EXPOSED_COLOUR = new Color(196,208,231);
    public static final Color HIDDEN_COLOUR = new Color(58,79,191);

    // Constructors
    /** Construct a new Square object
     */
    public Square(boolean mine){
        this.mine = mine;
    }
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

Square - MineSweeper

Class Edit Tools Options

MineSweeper × Square ×

Compile Undo Cut Copy Paste Find... Close Source Code

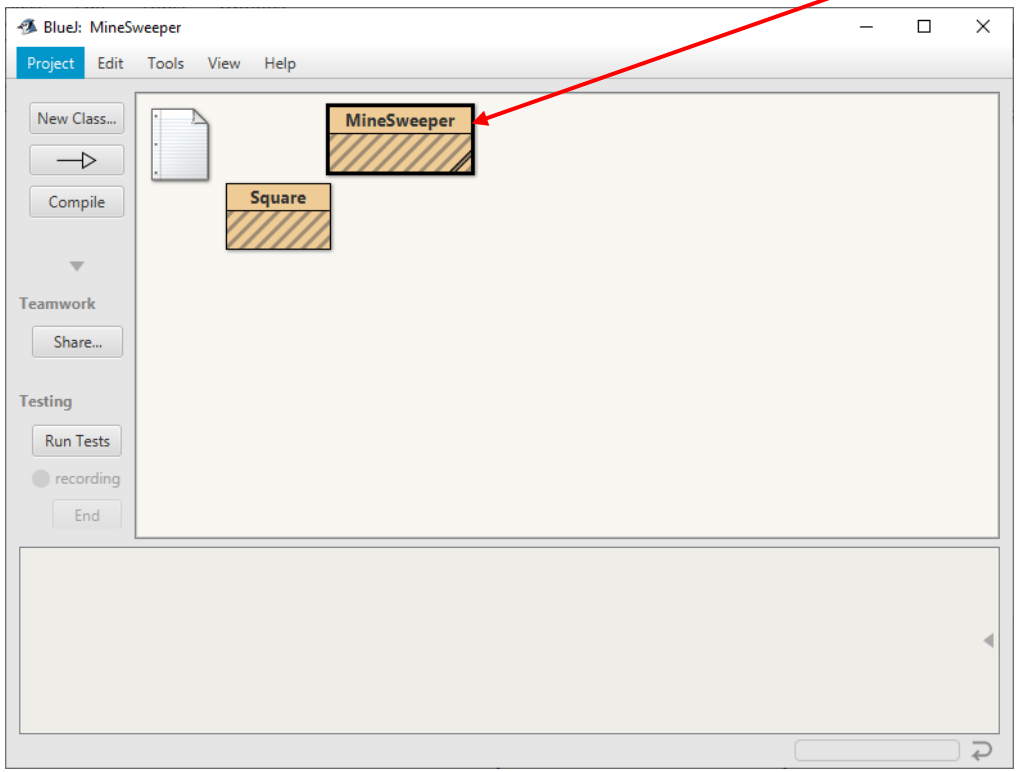
```
42  // Constructors
43  /** Construct a new Square object
44   */
45  public Square(boolean mine){
46      this.mine = mine;
47  }
48
49  // Methods
50  /** Get the number of mines adjacent to this square  */
51  public int getAdjacentMines(){
52      return adjacentMines;
53  }
54
55  /** Record the number of adjacent mines */
56  public void setAdjacentMines(int num){
57      adjacentMines = num;
58  }
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

Files inside the **MineSweeper** folder

MineSweeper.java	16/04/2020 12:17 PM	JAVA File	7 KB
package.bluej	16/04/2020 12:17 PM	BlueJ Project File	1 KB
Square.java	16/04/2020 12:17 PM	JAVA File	4 KB



Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

```
102     /**
103      * The player has clicked on a square to expose it
104      * - if it is already exposed or marked, do nothing.
105      * - if it's a mine: lose (call drawLose())
106      * - otherwise expose it (call exposeSquareAt)
107      * then check to see if the player has won and call drawWon() if they have.
108      * (This method is not recursive)
109      */
110     public void tryExpose(int row, int col){
111         /*# YOUR CODE HERE */
112
113         if (hasWon()){
114             drawWin();
115         }
116     }
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

```
102     /**
103      * The player has clicked on a square to expose it
104      * - if it is already exposed or marked, do nothing.
105      * - if it's a mine: lose (call drawLose())
106      * - otherwise expose it (call exposeSquareAt)
107      * then check to see if the player has won and call drawWon() if they have.
108      * (This method is not recursive)
109      */
110     public void tryExpose(int row, int col){
111         /*# YOUR CODE HERE */
112
113         if (hasWon()){
114             drawWin();
115         }
116     }
```

(1)

Type the following:
Square sq = squares[row][col];

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

```
102     /**
103      * The player has clicked on a square to expose it
104      * - if it is already exposed or marked, do nothing.
105      * - if it's a mine: lose (call drawLose())
106      * - otherwise expose it (call exposeSquareAt)
107      * then check to see if the player has won and call drawWon() if they have.
108      * (This method is not recursive)
109      */
110     public void tryExpose(int row, int col){
111         /*# YOUR CODE HERE */
112
113         if (hasWon()){
114             drawWin();
115         }
116     }
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

```
102  /**
103   * The player has clicked on a square to expose it
104   * - if it is already exposed or marked, do nothing.
105   * - if it's a mine: lose (call drawLose())
106   * - otherwise expose it (call exposeSquareAt)
107   * then check to see if the player has won and call drawWon() if they have.
108   * (This method is not recursive)
109   */
110  public void tryExpose(int row, int col){
111      /*# YOUR CODE HERE */
112
113      if (hasWon()){
114          drawWin();
115      }
116  }
```

```
if (sq.isExposed() || sq.isMarked()) {
    return;
}
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

```
/**
 * The player has clicked on a square to expose it
 * - if it is already exposed or marked, do nothing.
 * - if it's a mine: lose (call drawLose())
 * - otherwise expose it (call exposeSquareAt)
 * then check to see if the player has won and call drawWon() if they have.
 * (This method is not recursive)
 */
public void tryExpose(int row, int col){
    /*# YOUR CODE HERE */

    if (hasWon()){
        drawWin();
    }
}
```

(2)

```
if (sq.isExposed() || sq.isMarked()) {
    return;
}
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

```
102  /**
103   * The player has clicked on a square to expose it
104   * - if it is already exposed or marked, do nothing.
105   * - if it's a mine: lose (call drawLose())
106   * - otherwise expose it (call exposeSquareAt)
107   * then check to see if the player has won and call drawWon() if they have.
108   * (This method is not recursive)
109   */
110  public void tryExpose(int row, int col){
111      /*# YOUR CODE HERE */
112
113      if (hasWon()){
114          drawWin();
115      }
116  }
```

```
else if (sq.hasMine()) {
    drawLose();
    return;
}
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

```
102 /**
103  * The player has clicked on a square to expose it
104  * - if it is already exposed or marked, do nothing.
105  * - if it's a mine: lose (call drawLose())
106  * - otherwise expose it (call exposeSquareAt)
107  * then check to see if the player has won and call drawWon() if they have.
108  * (This method is not recursive)
109  */
110 public void tryExpose(int row, int col){
111     /*# YOUR CODE HERE */
112
113     if (hasWon()){
114         drawWin();
115     }
116 }
```

(3)

```
else if (sq.hasMine()) {
    drawLose();
    return;
}
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

```
102  /**
103   * The player has clicked on a square to expose it
104   * - if it is already exposed or marked, do nothing.
105   * - if it's a mine: lose (call drawLose())
106   * - otherwise expose it (call exposeSquareAt)
107   * then check to see if the player has won and call drawWon() if they have.
108   * (This method is not recursive)
109   */
110  public void tryExpose(int row, int col){
111      /*# YOUR CODE HERE */
112
113      if (hasWon()){
114          drawWin();
115      }
116  }
```

```
else {
    exposeSquareAt(row, col);
}
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

```
102 /**
103  * The player has clicked on a square to expose it
104  * - if it is already exposed or marked, do nothing.
105  * - if it's a mine: lose (call drawLose())
106  * - otherwise expose it (call exposeSquareAt)
107  * then check to see if the player has won and call drawWon() if they have.
108  * (This method is not recursive)
109  */
110 public void tryExpose(int row, int col){
111     /*# YOUR CODE HERE */
112
113     if (hasWon()){
114         drawWin();
115     }
116 }
```

(4)

```
else {
    exposeSquareAt(row, col);
}
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

MineSweeper - MineSweeper

Class Edit Tools Options

MineSweeper

Compile Undo Cut Copy Paste Find... Close Source Code

```
117
118     /**
119      *  Expose a square, and spread to its neighbours if safe to do so.
120      *  It is guaranteed that this square is safe to expose (ie, does not have a mine).
121      *  If it is already exposed, we are done.
122      *  Otherwise expose it, and redraw it.
123      *  If the number of adjacent mines of this square is 0, then
124      *     expose all its neighbours (which are safe to expose)
125      *     (and if they have no adjacent mine, expose their neighbours, and ....)
126      */
127     public void exposeSquareAt(int row, int col){
128         /*# YOUR CODE HERE */
129
130     }
```

```
/*# YOUR CODE HERE */
Square sq = squares[row][col];
if (sq.isExposed()) {return;}
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

MineSweeper - MineSweeper

Class Edit Tools Options

MineSweeper

Compile Undo Cut Copy Paste Find... Close Source Code

```
117
118     /**
119      *  Expose a square, and spread to its neighbours if safe to do so.
120      *  It is guaranteed that this square is safe to expose (ie, does not have a mine).
121      *  If it is already exposed, we are done.
122      *  Otherwise expose it, and redraw it.
123      *  If the number of adjacent mines of this square is 0, then
124      *     expose all its neighbours (which are safe to expose)
125      *     (and if they have no adjacent mine, expose their neighbours, and ....)
126      */
127     public void exposeSquareAt(int row, int col){
128         /*# YOUR CODE HERE */
129
130     }
```

```
sq.setExposed();
sq.draw(LEFT+col*SQUARE_SIZE, TOP+row*SQUARE_SIZE, SQUARE_SIZE);
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

```
122      *  Otherwise expose it, and redraw it.
123      *  If the number of adjacent mines of this square is 0, then
124      *     expose all its neighbours (which are safe to expose)
125      *     (and if they have no adjacent mine, expose their neighbours, and ....)
126      */
127     public void exposeSquareAt(int row, int col){
128         /*# YOUR CODE HERE */
129
130     }
```

```
if (sq.getAdjacentMines()==0) {
    for                                                    {
        for                                                {
            if
                exposeSquareAt(r, c);
        }
    }
}
```