



Course Name: Java 2

Course Code: CS209

Department: Computer Science

Exam Paper Setter: Stéphane Faroult

Part 2: Minesweeper Game (50 points)

第一条 Object design (15 points)

Widget to extend (5 points):

Any widget that you can click on is OK - Button or CheckBox probably the best choices .

Additional attributes: (5 points)

```
private boolean hasMine;    // 1.5 point
private boolean isClicked ; // 1.5 point
private boolean isFlagged;  // 0.5 point - likely to be forgotten
// Optional: count of mines around (can be counted dynamically)
// - 0.5 bonus even if not really useful
// For the location: either row and column on the board, or
// array of references to the surrounding cells - 2 points
private int    row;
private int    col;
```

Container: (5 points)

Best choice: GridPane (5 points)

Acceptable: multiple Hbox in VBox or multiple Hbox in VBox (3.5 points)

第一条 Game initialization (15 points)

Important points when grading:

- ☐ Function must be called on the first click (2 point if specified)
- ☐ There must be a boolean variable to check if initialization was performed, which must be set by the initialization method (3 points)
- ☐ No mine must be set on current position (6 points)
- ☐ Random computation of both row and column (2 points)
- ☐ 2 points for while loop to check if the right number of mines was set, 1 point if a for loop is used (if we try to set a mine on the current position, the mine count would be one less than expected)

Example code

I'm using class attributes rows, cols, mines, board and initialized. My board has a single

dimension, could be two dimensions.

```
private void initBoard(int excludeRow, int excludeCol) {
    Random rand = new Random();
    int    r;
    int    c;
    int    mineCount = 0;

    while (mineCount < mines) {
        r = rand.nextInt(rows);
        c = rand.nextInt(cols);
        if ((r != excludeRow) || (c != excludeCol)) {
            board[r * cols + c].plantMine();
            // Method that sets the mine attribute to true
            mineCount++;
        }
    }
    initialized = true;
}
```

□

第一条 Playing the game (20 points)

Important points when grading:

- Check if first click and call initialization if this is the case (5 points)
- Do nothing if cell was already clicked (3 points)
- Return with an "end of game" indication if mine hit (1 point)
- Reveal everything if mine hit (3 points)
- Reveal count (can count dynamically or display a stored count) if no mine (2 points)
- Recursive click of surrounding mines if no mine around (6 points)

2 points if loops on surrounding cells to show what they contain without recursion.

Example code

```
private boolean click() {
    boolean gameOver = false;
    if (!isClicked) {
        if (!initialized) {
            initBoard(this.row, this.col);
        }
        this.getStyleClass().add("clicked");
        isClicked = true;
        if (this.hasMine) {
            System.out.println("Boom");
            Image image = new Image("boom.png");
            ImageView iv = new ImageView();
```

```

        iv.setImage(image);
        setGraphic(iv);
        gameOver = true;
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                board[cols * i + j].reveal();
            }
        }
    } else {
        int minesAround = countSurroundingMines();
        if (minesAround > 0) {
            this.setText(Integer.toString(minesAround));
            this.getStyleClass().add("m" +
                                    Integer.toString(minesAround));
        } else {
            for (int i = -1; i <= 1; i++) {
                for (int j = -1; j <= 1; j++) {
                    if (((row + i) >= 0)
                        && ((row + i) < rows)
                        && ((col + j) >= 0)
                        && ((col + j) < cols)
                        && ((i != 0) || (j != 0))) {
                        gameOver = board[cols * (row + i)
                                       + (col + j)].click();
                    }
                }
            }
        }
    }
}
return gameOver;
}

```

Part 3: Data Handling (20 points)

1. Collection type (5 points)

- ☐ Two pieces of data to store (clue + gender) therefore begs for a Map
 - TreeMap or HashMap OK
- ☐ Array of two collections (one for male clues, other for female clues) not as good but could work 3 points.

2. Finding gender - ArrayList categories (15 points)

Important for grading:

- ☐ Loop on categories (3 points)

- ☐ Split category into words (5 points)
- ☐ Search each word as key of map (4 points) or try to get the first list where it is if using two lists (3 points)
- ☐ Returns immediately when found (3 points)
 - If use of counters 2 points
 - If looping on everything and remembering what was found 1 point
 - If looping n everything and overwriting what was found 0.5 for effort ...

```
String gender; // or char
for (String cat: categories) {
    //split cat into words
    String[] words = cat.split(' ');
    for (String w: words) {
        gender = clues.get(w); // if clues is a map
        if (gender != null) {
            // Two options:
            // a. Return immediately the gender found (Best)
            // b. Maintain one counter for each gender and count clues (OK)
        }
        // If clues is an array of two lists:
        // loop on two positions
        // if list contains w, then return immediately or count clues
    }
}
```

Bonus remark: give more importance to female than male clues. For instance, there is no special word for "female director". These are for example the categories for Ann Hui:

- ☐ [Asian Film Award winners](#)
- ☐ [1947 births](#)
- ☐ [Living people](#)
- ☐ [Alumni of the University of Hong Kong](#)
- ☐ [Chinese women film directors](#)
- ☐ [Fukuoka Asian Culture Prize winners](#)
- ☐ [Hong Kong film actresses](#)
- ☐ [Hong Kong film directors](#)
- ☐ [Hong Kong film producers](#)
- ☐ [Hong Kong people of Japanese descent](#)
- ☐ [Hong Kong screenwriters](#)
- ☐ [Hong Kong women writers](#)
- ☐ [Members of the Order of the British Empire](#)
- ☐ [People from Anshan](#)
- ☐ [Writers from Liaoning](#)
- ☐ [Actresses from Liaoning](#)

- ☐ [Film directors from Liaoning](#)
- ☐ [Asian film producers](#)
- ☐ [Alumni of the London Film School](#)
- ☐ [Chinese film directors](#)