MINESWEEPER

Play

Previous

Next

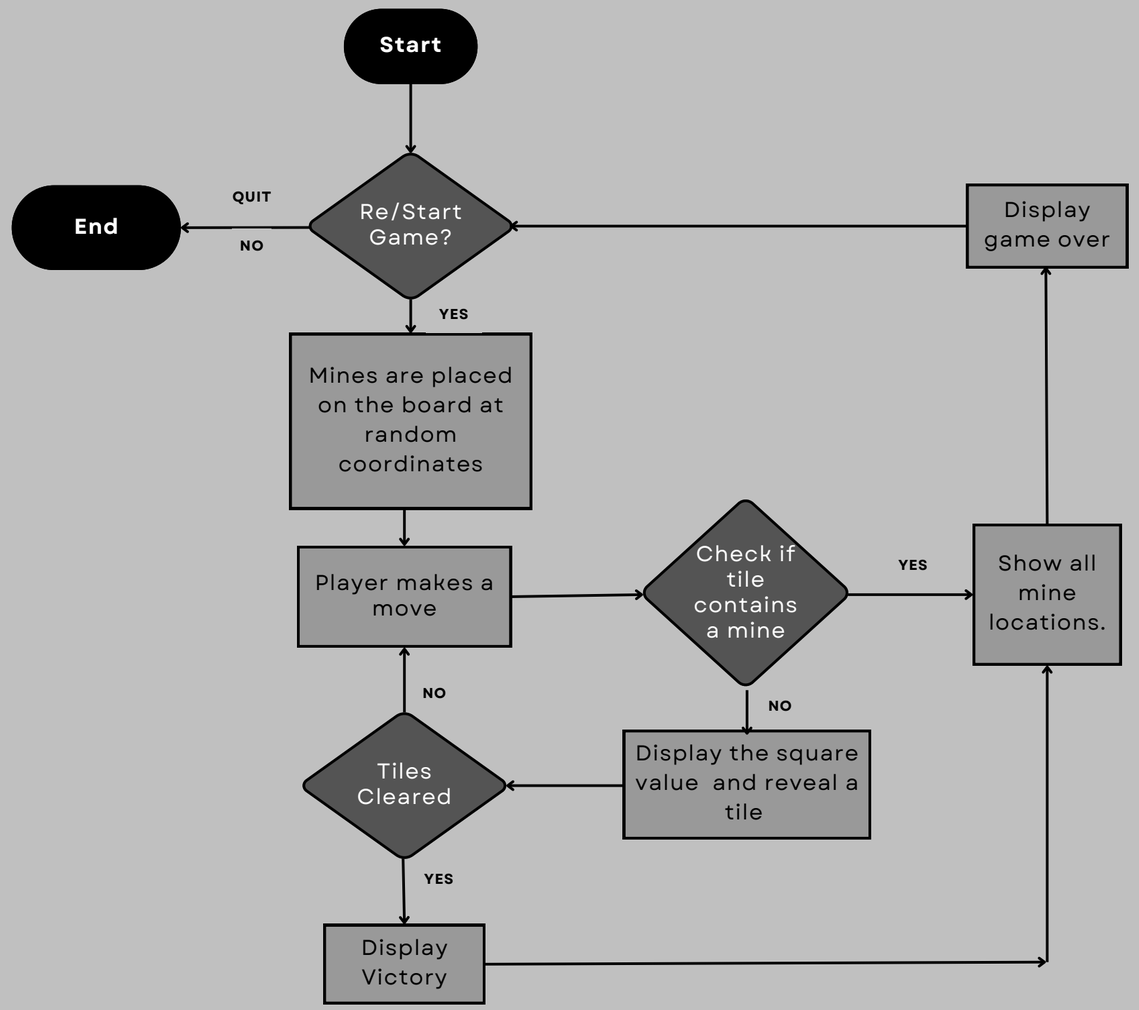
INTRODUCTION

In our Java programming project, we created a Minesweeper game with Java Swing, demonstrating our object-oriented programming skills. This classic puzzle game requires players to solve a grid of hidden mines using logic and deduction. We developed an interactive and visually appealing game, demonstrating our coding, problem-solving, and user interface design skills.

Previous

FLOWCHART

Next



Previous

Next

DESIGN

```
Run main | Debug main | Run | Debug
public static void main(String[] args) {
    // Start the Minesweeper game
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new Minesweeper();
        }
    });
}
}
```

```
if (dialogResult == JOptionPane.YES_OPTION) {
    frame.setSize(boardWidth, boardHeight + 50);
    frame.setLocationRelativeTo(c:null);
    frame.setResizable(resizable:false);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setLayout(new BorderLayout());
```

```
// Constructor
Minesweeper() {
    // Start the game
    System.out.println(x:"Starting the game!");

    // Show confirmation dialog before starting the game
    JFrame startFrame = new JFrame(title:"Start Game");
    startFrame.setSize(width:300, height:150);
    startFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    startFrame.setLocationRelativeTo(c:null);

    // Show confirmation dialog
    int dialogResult = JOptionPane.showConfirmDialog(frame, message:"
```

Previous

Next

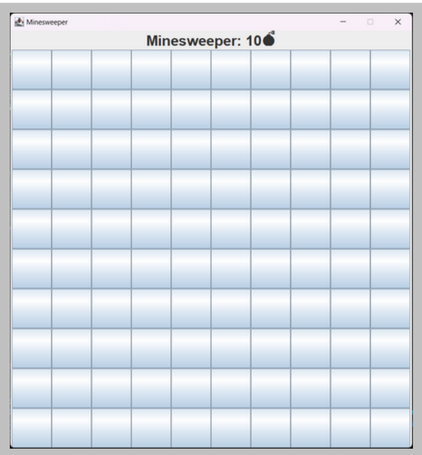
DESIGN

```
int tileSize = 70; // Each tile is going to be 70x70 pixels
int numRows = 10;
int numCols = numRows; // Setting it equal to numRows to create a
int boardWidth = numCols * tileSize; // It will multiply the numC
int boardHeight = numRows * tileSize; // It will multiply the num

JFrame frame = new JFrame(title:"Minesweeper"); // Creating the g
JLabel textLabel = new JLabel(); // label for text MINESWEEPER
JPanel textPanel = new JPanel(); // panel for text MINSWEEPER
JPanel boardPanel = new JPanel(); // panel for board

// Placing random mines
int mineCount = 10; // number of mines that will be placed random
```

```
textPanel.setLayout(new BorderLayout()); //is set up to
textPanel.add(textLabel); //adding the attributes of JLa
frame.add(textPanel, BorderLayout.NORTH); //adding JPane

boardPanel.setLayout(new GridLayout(numRows, numCols));

frame.add(boardPanel);
```

```
//Game Window
frame.setSize(boardWidth, boardHeight + 50); //specifying
frame.setLocationRelativeTo(c:null); //this will open the
frame.setResizable(resizable:false); //so the window fram
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //w
frame.setLayout(new BorderLayout());
```

```
//for text MINESWEEPER
textLabel.setFont(new Font(name:"Arial Unicode MS", Font.BOLD, size:25));
textLabel.setHorizontalAlignment(JLabel.CENTER); //centering the text
textLabel.setText("Minesweeper: " + Integer.toString(mineCount) + "💣");
textLabel.setOpaque(isOpaque:true); //makes the background of the label vi
```



Previous

Next

DESIGN

```java
//filling the grid with mine tiles using nested for loops
for(int row = 0; row < numRows; row++) {
    for(int col = 0; col < numCols; col++) {
        MineTiles tile = new MineTiles(row, col);
        board[row][col] = tile;
```

```java
tile.setFocusable(focusable:false);
tile.setMargin(new Insets(top:0, left:0, bottom:0, right:0));
tile.setFont(new Font(name:"Arial Unicode MS", Font.PLAIN, size:25 ));
```

```java
                    (bu
if (e.getButton() == MouseEvent.BUTTON1) {
    if(tile.getText() == "") {//this means we haven't opened it, by default all tiles dont have any
        if(mineList.contains(tile)) { //if the tile is empty, we want to check to see if its in the
            revealMines(); //if it's in mine list, it's essentially game over
        }
        else {
            checkMine(tile.row, tile.col); //check for mines
        }
    }
}                                                                                    //le
//right click (button3)
else if(e.getButton() == MouseEvent.BUTTON3) {
    if(tile.getText() == "" && tile.isEnabled()) { //we wont be able to place a flag on a clicked tile
        tile.setText(text:"🚩");
    }
    else if(tile.getText() == "🚩") { //to not click the tile with a flag
        tile.setText(text:"");
    }
}
```

```java
tile.addMouseListener(new MouseAdapter(){
    //overriding mouse pressed
    @Override
    public void mousePressed(MouseEvent e) {
        if(gameOver) {
            return;
        }
        MineTiles tile = (MineTiles)e.getSource();
```

Previous

Next

DESIGN

```java
//Placing random mines
int mineCount = 10; //number of mines that will be placed randomly

//using 2D array to store each mine tiles so we know where each tile is
MineTiles[][] board = new MineTiles[numRows][numCols];

//using ArrayList to create mine tiles
ArrayList<MineTiles> mineList;

//Creating Random object for the mines
Random random = new Random();
```

```java
//if the player clicked on a tile that's in the mine list, this method is for revealing the mines
void revealMines() {
    //iterate through the arraylist containing all the tiles with mines
    for(int i = 0; i < mineList.size(); i++) {
        MineTiles tile = mineList.get(i);
        tile.setText(text:"💣");
    }
```

```java
void setMines() {
    //creating arraylist for the mine list
    mineList = new ArrayList<MineTiles>();

    //placing random mines
    int mineLeft = mineCount;
    while(mineLeft > 0) {
        int row = random.nextInt(numRows); //rows 0-7 index
        int col = random.nextInt(numCols); //columns 0-7 index

        MineTiles tile = board[row][col]; //get the tile at row and column numbers

        if(!mineList.contains(tile)) { //checking if mineList already has this tile
            mineList.add(tile);
            mineLeft -= 1;
        }

    }
}
```

Previous

Next

DESIGN

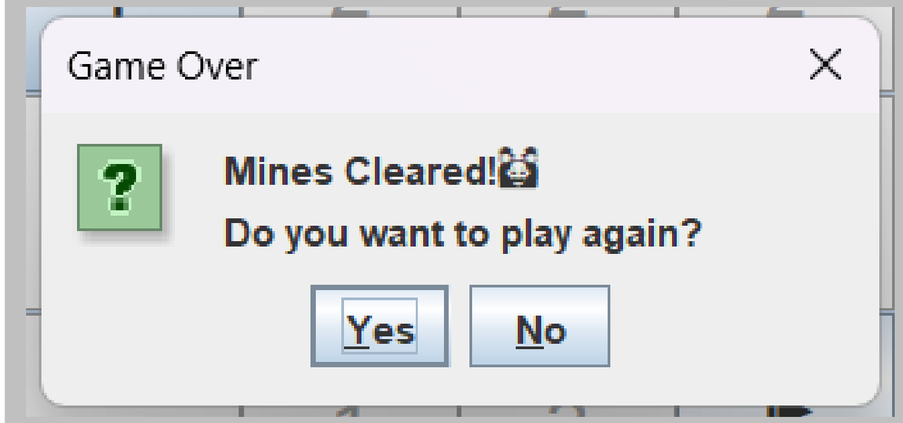
```java
//this method is to reset the game state
void resetGame() {
    gameOver = false; //setting the gameOver flag to false, meaning the game
    tilesClicked = 0; //Reset the count of tiles clicked to 0
    textLabel.setText("Minesweeper: " + Integer.toString(mineCount) + "💣");
```

```java
// Reset all tiles
for (int row = 0; row < numRows; row++) { //nested for loop through each row
    for (int col = 0; col < numCols; col++) { //loop through each column
        MineTiles tile = board[row][col]; //getting the tile at the current row and column
        tile.setEnabled(b:true); //enabling the tile allowing it to be clicked again
        tile.setText(text:""); //clearing any text on the tile
    }
}

// Reset mines
setMines(); //calling on the setMines method to place mines on the board again
```

```java
// method to show reset dialog
void showResetDialog(String message) {
    int result = JOptionPane.showOptionDialog(
        frame,
        message + "\nDo you want to play again?",
        title:"Game Over",
        JOptionPane.YES_NO_OPTION,
        JOptionPane.QUESTION_MESSAGE,
        icon:null,
        options:null,
        initialValue:null
    );

    if (result == JOptionPane.YES_OPTION) { //if the user selects YES
        resetGame(); //this will reset the game
    } else {
        frame.dispose(); //this will terminate the game
    }
}
```

```java
//losing condition
gameOver = true;
textLabel.setText(text:"Game Over!😭");
showResetDialog(message:"Game Over!😭");
```

```java
if(tilesClicked == numRows * numCols - mineList.size()) { //for winning case
    gameOver = true;
    textLabel.setText(text:"Mines Cleared!🙌");
    showResetDialog(message:"Mines Cleared!🙌");
}
```



Previous

Next

MANUAL

GAME RULES!

Minesweeper is a game where you need to find all the hidden mines on a grid without setting them off. Use clues to find the mines.

Find all the mines without clicking on them.

Previous

Next

MANUAL

HOW TO PLAY?

The grid has tiles; some have mines.

Click a tile to see what's under it. You'll see a number (how many mines are next to it) or a mine (game over).

Use the numbers to figure out where the mines are.

Mark tiles you think have mines with a flag.

Previous

Next

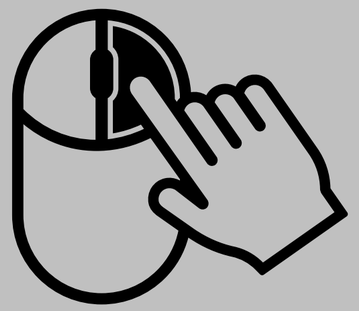
MANUAL

HOW TO PLAY?

LEFT CLICK

Left-click to reveal a tile.

RIGHT CLICK



Right-click to place or remove a flag on a tile.

Previous

Next

MANUAL

GAMEPLAY

The goal is to click through every single tile while avoiding the 10 mines.

The number shown inside the tiles indicates how many mines they are next to (the tiles that are one unit away from the number).

If you click on a tile that has no number, it will expand outwards and open up all the tiles until you get numbers.

Previous

Next

MANUAL

GAMEPLAY

To mark a tile you think is a mine, point and right-click on the tile and it will place a flag there.

These flags would help you eventually identify exactly where your problem areas were.

Placing a flag where you think a mine is placed won't open the tile and trigger the mine.

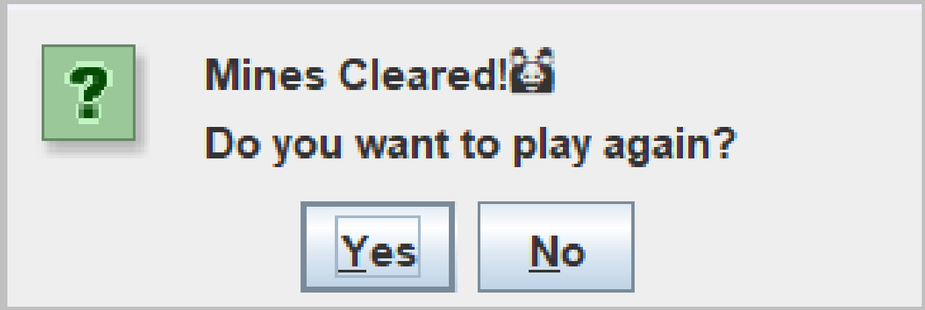
You can undo the flag by right clicking again.

Previous

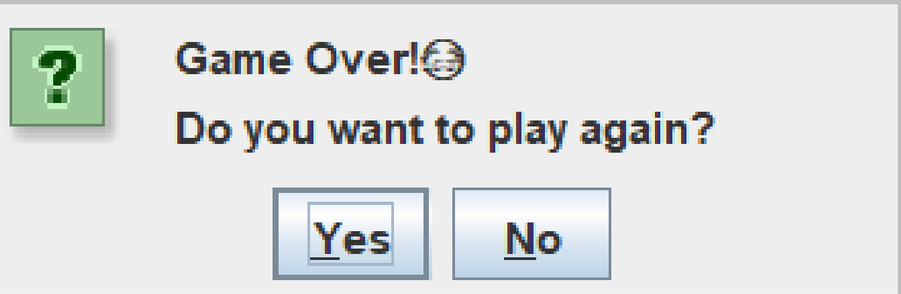
Next

MANUAL

SCORING



Win: You win the game by marking all the mines correctly and revealing all the other tiles.



Lose: You lose if you click on a mine.

THANK YOU!

You Win!