

Java Swing AI Project for MAC190 - Object Oriented Programming

Andi Toce

September 2, 2024

Objective

- ▶ Use AI tool to create an interactive game using Java Swing graphics. This project will help reinforce students' Java programming skills, enhance understanding of UI development, and explore game design concepts.

Requirements

1. **Game Concept:**

- ▶ Choose a game concept and define mechanics, rules, and objectives.

2. **Java Swing Basics:**

- ▶ Use Swing components for the game window and UI elements.

3. **Graphics and Animation:**

- ▶ Create game graphics and (optional) animations.

4. **Technical Features:**

- ▶ Organize code, handle input, and include error handling.

5. **Presentation and Documentation:**

- ▶ Prepare a presentation and provide a README file.

Grading Criteria

Your project will be evaluated based on:

- ▶ Creativity and uniqueness of the game concept.
- ▶ Proper use of Java Swing components.
- ▶ Quality of graphics and animations.
- ▶ Code organization and documentation.
- ▶ Clarity and confidence during the presentation.

Note for Students on the Usage of AI Tools

While creating your interactive game using Java Swing graphics, use AI tools as a complementary resource. These tools can enhance your learning experience and streamline certain tasks. However, it's essential to strike a balance—AI should not replace your understanding of core concepts. Instead, view AI as a facilitator that assists with repetitive tasks, generates ideas, or provides insights. By actively participating in the development process alongside AI, you'll gain a deeper understanding of both game design and programming principles.

Example of AI Generated Project

The screenshot shows an IDE with two windows. The left window displays the Java source code for a class named `GameBoard`. The code includes package declarations, imports, and class-level variables for grid size, tile size, gap size, status height, and winning tile count. It also defines private attributes for the board, score, and game state. The `GameBoard` constructor sets the preferred size and initializes the board. A `KeyListener` is added to handle key events for moving the cursor and revealing tiles.

```
1 package G2848;  
2  
3 import javax.swing.*;  
4  
5  
6 public class GameBoard extends JPanel {  
7     private static final int GRID_SIZE = 4;  
8     private static final int TILE_SIZE = 100;  
9     private static final int GAP_SIZE = 10;  
10    private static final int STATUS_HEIGHT = 50;  
11    private static final int WINNING_TILE = 2848;  
12  
13    private int[][] board;  
14    private int score;  
15    private boolean won = false;  
16    private boolean lost = false;  
17  
18    public GameBoard() {  
19        setPreferredSize(new Dimension(GRID_SIZE * (TILE_SIZE + GAP_SIZE)  
20        setFocusable(true);  
21        initBoard();  
22        addKeyListener(new KeyAdapter() {  
23            @Override  
24            public void keyPressed(KeyEvent e) {  
25                if (!won && !lost) {  
26                    switch (e.getKeyCode()) {  
27                        case KeyEvent.VK_UP:  
28                            moveUp();  
29                            break;  
30                        case KeyEvent.VK_DOWN:  
31                            moveDown();  
32                            break;  
33                        case KeyEvent.VK_LEFT:  
34                            moveLeft();  
35                            break;  
36                        case KeyEvent.VK_RIGHT:  
37                            moveRight();  
38                            break;  
39                    }  
40                }  
41            }  
42        });  
43    }  
44
```

The right window shows the graphical user interface of the game. It features a score display at the top showing "Score: 104" and a 4x4 grid of tiles. The tiles are colored based on their values: 2, 8, 2, 2 in the first row; 16, 16, 4 in the second row. The tiles are arranged in a 4x4 grid with a gap between them.

2			
8	2		2
16	16	4	

Warning: the fonts "Times" and "Times" are not available for the