Java Swing AI Project for MAC190 - Object Oriented Programming

Andi Toce

September 2, 2024

Objective

- Use AI toold 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

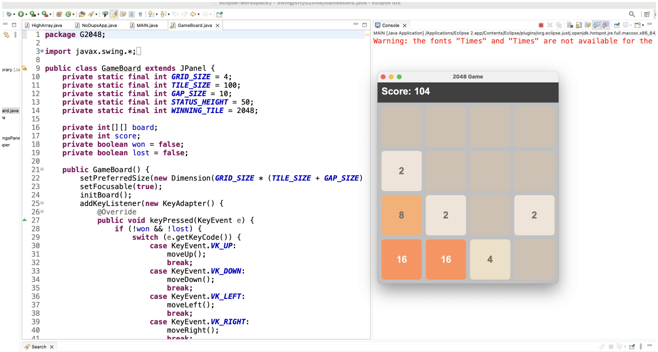
Your project will be evaluated based on:

- 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



```java
package G2048;

import javax.swing.*;

public class GameBoard extends JPanel {
    private static final int GRID_SIZE = 4;
    private static final int TILE_SIZE = 100;
    private static final int GAP_SIZE = 10;
    private static final int STATUS_HEIGHT = 50;
    private static final int WINNING_TILE = 2048;

    private int[][] board;
    private int score;
    private boolean won = false;
    private boolean lost = false;

    public GameBoard() {
        setPreferredSize(new Dimension(GRID_SIZE * (TILE_SIZE + GAP_SIZE)
        setFocusable(true);
        initBoard();
        addKeyListener(new KeyAdapter() {
            @Override
            public void keyPressed(KeyEvent e) {
                if (!won && !lost) {
                    switch (e.getKeyCode()) {
                        case KeyEvent.VK_UP:
                            moveUp();
                            break;
                        case KeyEvent.VK_DOWN:
                            moveDown();
                            break;
                        case KeyEvent.VK_LEFT:
                            moveLeft();
                            break;
                        case KeyEvent.VK_RIGHT:
                            moveRight();
                            break;
```

```
MAIN [Java Application] /Applications/Eclipse 2.app/Contents/Eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64
Warning: the fonts "Times" and "Times" are not available for the
```

2048 Game

Score: 104

2			
8	2		2
16	16	4	