Second Big Homework

Programming I

December 7, 2024

This project focuses on creating a PathMaster3000 game in Java using GUI components. The game involves navigating from a start field to an end field on a grid, calculating scores based on the path taken, and incorporating advanced game features. The project will assess your skills in GUI design, event handling, and algorithm implementation.

1 Grid Creation and Gameplay (50 points)

- Build the Grid: Create an $n \times n$ grid of buttons, each labeled with a random number between 0 and 9, except:
  - One button labeled as the Start field.
  - One button labeled as the End field.
- Gameplay Rules
  - Users must navigate from the Start field to the End field by selecting field buttons.
  - Moves are restricted to neighboring fields (North, South, East, West).
  - A field cannot be revisited once selected.
  - The path must be visually marked as the user progresses.
- Score Calculation
  - The score is calculated as:

$$Score = \frac{\sum selected\ field\ values}{Total\ path\ length}$$

  - Display the score in the interface.

2 Menu Integration (10 points)

Add a Menu Window accessible from the main game interface. The menu should include:

- Restart Options
  - Change the grid size ($n \times n$).
  - Randomly or manually select the **Start** and **End** fields.
- Customize Appearance
  - Allow users to modify the game and path colors

Ensure that menu actions dynamically update the game state.

3 Save and Load Feature (15 points)

- Save Game

  Implement functionality to save the game state (grid, path, score, etc.) to a file.

- Load Game

  Allow users to load a saved game from the menu and continue playing.

4 Solution Statistics (5 points)

Add a statistics panel below the grid to display data such as:

- Path length (number of steps in the path).
- Sum of the selected field values.
- Score (calculated as per the formula above).
- A timer showing the elapsed time since the game started.

5 Optimization and Hints (20 points)

- Optimal Path Comparison

  At the end of the game, display how close the user's path is to the optimal solution as a percentage (e.g., "Your solution is 75% optimal").

- Hint Button

  Add a button to suggest the next best step for achieving a better score, following the game rules.

Note

- Make sure the basic elements of object-oriented programming are correctly used; static methods are forbidden, define individual program modules as stand-alone classes. (up to -5)
- Make sure you use the contracts correctly! (up to -5)
- Entire project should be submitted as a zip file.