Amina Arooj

Uxm59h

Description

Hunting is a two-player strategy game played on an n-by-n size board(n can be 3 5 7), where one player takes on the role of the fugitive, and the other plays the role of the hunter. The fugitive starts in the center, while the hunter has four positions at the corners. Players alternate turns, moving their characters one step at a time if there is space, with the hunter aiming to surround the fugitive within a maximum of 4n steps. The game ends when the fugitive is captured or the maximum number of turns is reached, displaying a message box to announce the winner, and then the game restarts automatically.

Connection of event and event handlers

Here in our task, we will use 2 events that will be handled accordingly:

* ActionListener

Action performed will check if selected size is not null then it will initialize board according to that size which will make a table according to that with h and f at their designated places according to task.

* MouseListener

A mouse click will call handle cell click which will manage the movement of hunters and fugitive according to their turns and will check who wins.

Methods Description

1. HuntingGui(): Constructs a new HuntingGui instance and initializes the game window.
2. initializeBoard() :Initializes the game board with the selected size and places the characters.
3. placeCharacters(): Places the fugitive in the center and hunters in the corners of the board
4. handleCellClick(int x, int y): Handles the cell click event to move the fugitive or hunter.
5. moveFugitive(int x, int y) : Moves the fugitive to the specified cell if the move is valid.
6. moveHunter(int x, int y) : Moves a hunter to the specified cell if the move is valid.
7. moveCharacter(Player player, int x, int y) : Moves the specified player to the target cell.
8. isFugitiveSurrounded() : Checks if the fugitive is surrounded by hunters.
9. showWinner(String winner) : Displays the winner in a message box and restarts the game.
10. chooseHunter(List<Player> hunters, int targetX,int target):It will ask the user to choose between the available options for hunters.

Test

* [testInitializeBoard()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Verifies the initial positions of the fugitive and hunters on the board. Ensures the game board is set up correctly.
* [testSelectBoardSize()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Tests the functionality of selecting a different board size. Ensures the board size changes correctly when a new size is selected.
* [testMoveFugitive()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Tests the movement of the fugitive to a new cell. Ensures the fugitive moves correctly and the previous cell is cleared.
* [testMoveHunter()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Tests the movement of a hunter to a new cell. Ensures the hunter moves scorrectly and the previous cell is cleared.
* [testInvalidMoveFugitive()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Tests an invalid move attempt by the fugitive. Ensures the fugitive's position remains unchanged after an invalid move.
* [testInvalidMoveHunter()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Tests an invalid move attempt by a hunter. Ensures the hunter's position remains unchanged after an invalid move.
* [testFugitiveCannotMoveToOccupiedCell()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Tests that the fugitive cannot move to a cell occupied by a hunter. Ensures the fugitive's position remains unchanged.
* [testHunterCannotMoveToOccupiedCell()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Tests that a hunter cannot move to a cell occupied by the fugitive. Ensures the hunter's position remains unchanged.
* [testGameEndsWhenFugitiveIsSurrounded()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Tests the game-ending condition when the fugitive is surrounded by hunters. Ensures the game correctly identifies when the fugitive is surrounded.
* [testGameResetsAfterEnd()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Tests the game reset functionality after a game ends. Ensures the game board and move count are reset correctly.
* [testTurnAlternatesAfterFugitiveMove()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Tests the turn alternation after the fugitive moves. Ensures it becomes the hunter's turn after the fugitive moves.
* [testTurnAlternatesAfterHunterMove()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Tests the turn alternation after a hunter moves. Ensures it becomes the fugitive's turn after a hunter moves.
* [testFugitiveCannotMoveOutOfBounds()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Tests that the fugitive cannot move out of the board's bounds. Ensures the fugitive's position remains unchanged after an out-of-bounds move.
* [testInvalidInputFugitive()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Tests the handling of invalid input for the fugitive's move. Ensures the fugitive's position remains unchanged after invalid input.
* [testInvalidInputHunter()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Tests the handling of invalid input for a hunter's move. Ensures the hunter's position remains unchanged after invalid input.

Top of Form

Bottom of Form