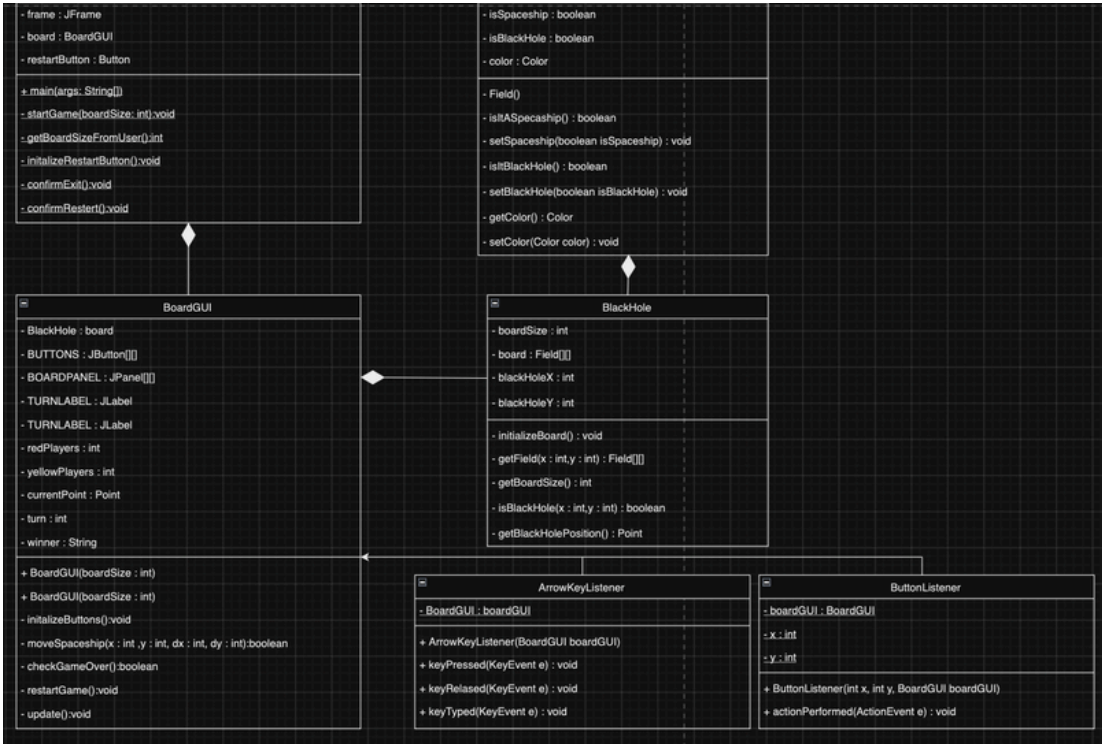
Name : Hajiyev Hajiaga **2 assignment 5th task** **7 October 2024**

Naptun code: C499UD

Task

Black hole is a two-player game, played on a board consists of n x n fields, which has a black hole at its center. Each player has n-1 spaceships, which are placed initially in the lower (upper) diagonal of the board (so the same colored spaceships placed on the same side). The players take turns moving one of their own spaceships. The black hole interferes with the navigation system of the spaceships, so they cannot move only one place, but they move until they reach the edge of the board, the black hole, or an other spaceship. A spaceship cannot jump over an other one. A player wins, if he manages to move half of his spaceships into the black hole. Implement this game, and let the board size be selectable (5x5, 7x7, 9x9). The game should recognize if it is ended, and it has to show in a message box which player won. After this, a new game should be started automatically.

UML Class Diagram



Description of major methods

The methods in BlackHole

1. BlackHole(int boardSize): Constructor that initializes the game board with a specified size, creates the board array, sets the black hole's position at the center, and initializes the board with spaceships and black hole.
2. initializeBoard(): Private method that sets up the board with default fields, places a black hole in the center, and positions spaceships for two players in predefined patterns.
3. getField(int x, int y): Returns the field located at the specified coordinates on the board.
4. getBoardSize(): Returns the size of the board.
5. isBlackHole(int x, int y): Checks if the specified coordinates correspond to the black hole's position.
6. getBlackHolePosition(): Returns the position of the black hole as a Point object.

The methods in BoardGUI

1. BoardGUI(int boardSize): Constructor that initializes the game board, GUI elements, and gameplay state with the specified board size.
2. getBoardPanel(): Returns a JPanel containing the game board and a label indicating whose turn it is.
3. initializeButtons(): Private method that creates and configures buttons representing the game board fields, assigning listeners for interaction.
4. getBoard(): Returns the current BlackHole board object.
5. getTurn(): Returns the current turn count.
6. incrementTurn(): Increments the turn counter by 1.
7. getCurrentPoint(): Returns the currently selected point on the board.
8. setCurrentPoint(Point currentPoint): Updates the currently selected point on the board.
9. moveSpaceship(int x, int y, int dx, int dy): Moves a spaceship from the given coordinates (x, y) in the specified direction (dx, dy) if valid, and handles interactions with black holes or other spaceships.
10. checkGameOver(): Checks if the game has ended by determining if a player has enough spaceships in the black hole.
11. restartGame(): Resets the game state, reinitializes the board, and updates the GUI for a new game.
12. update(): Updates the GUI to reflect the current game state, including the board's appearance, turn label, and checking for game over conditions.

The methods in Field

1. Field(): Constructor that initializes the field with default values (isSpaceship as false, isBlackHole as false, and color as white).
2. isItASpaceship(): Returns true if the field contains a spaceship; otherwise, false.
3. setSpaceship(boolean isSpaceship): Sets whether the field contains a spaceship.
4. isItABlackhole(): Returns true if the field is a black hole; otherwise, false.
5. setBlackHole(boolean isBlackHole): Sets whether the field is a black hole. If true, the field's color is automatically set to black.
6. getColor(): Returns the current color of the field.
7. setColor(Color color): Updates the field's color.

The methods in Main

1. main(String[] args): Entry point for the application that sets up the main game frame, adds a confirmation dialog for exiting, initializes the restart button, and starts the game.
2. startGame(int boardSize): Initializes the game board with the specified size, updates the frame layout, and ensures the restart button is displayed persistently.
3. getBoardSizeFromUser(): Prompts the user to select the board size from available options and returns the selected size as an integer, defaulting to 5x5 if no choice is made.
4. initializeRestartButton(): Creates the restart button and assigns an action listener to handle game restarts if the button does not already exist.
5. confirmExit(): Displays a confirmation dialog when the user attempts to close the game and exits the application if the user confirms.
6. confirmRestart(): Displays a confirmation dialog when the restart button is clicked and restarts the game with a new board size if the user confirms.

The methods in ArrowKeyListener

1. ArrowKeyListener(BoardGUI boardGUI): Constructor that initializes the key listener with a reference to the main game board GUI class.
2. keyPressed(KeyEvent e): Handles key presses, checks the current spaceship position, calculates movement direction based on arrow key input, and calls moveSpaceship() to update the position if the move is valid. Updates the game state by incrementing the turn and refreshing the display.
3. keyReleased(KeyEvent e): Empty method, required by KeyListener, but not used in this implementation.
4. keyTyped(KeyEvent e): Empty method, required by KeyListener, but not used in this implementation.

The methods in ButtonListener

1. ButtonListener(int x, int y, BoardGUI boardGUI): Constructor that initializes the button listener with coordinates (x, y) and a reference to the main game board GUI class.
2. actionPerformed(ActionEvent e): Handles button click events, checks if the field at the specified coordinates contains a spaceship and if the turn condition is met (based on turn number and field color). If valid, sets the current point to the clicked field's coordinates and updates the board.

Connections between the events and event handlers.

ButtonListener (Button Click Handler)

- Event Trigger: User clicks on a game board tile (a button representing a field).
- Event Handler: ButtonListener class, specifically the actionPerformed() method.

ArrowKeyListener (Arrow Key Press Handler)

- Event Trigger: User presses an arrow key (Up, Down, Left, Right).
- Event Handler: ArrowKeyListener class, specifically the keyPressed() method.

Window Closing Event (Exit Confirmation)

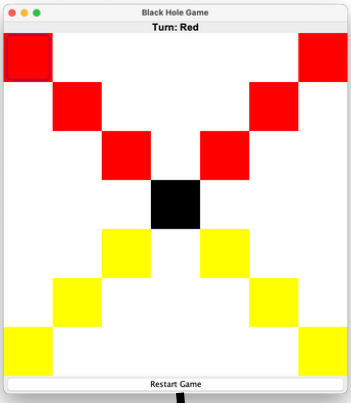
- Event Trigger: User attempts to close the game window.
- Event Handler: Window listener in Main class, specifically the windowClosing() method.

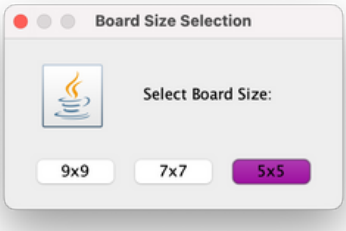
Restart Button (Game Restart)

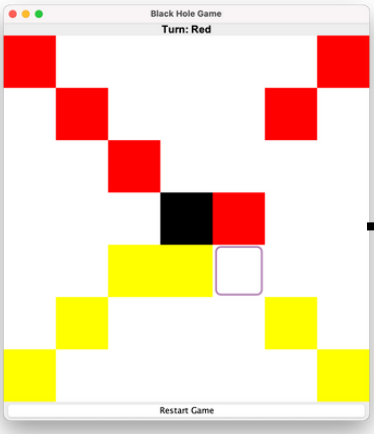
- Event Trigger: User clicks the "Restart Game" button.
- Event Handler: The ActionListener attached to the restartButton in the Main class.

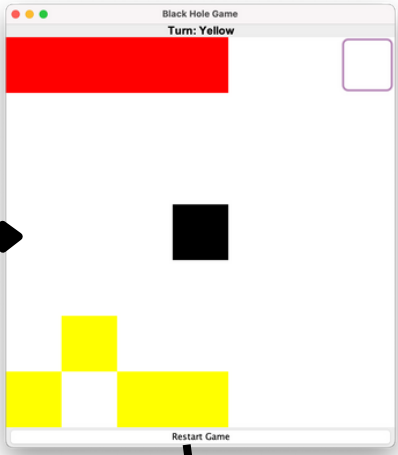
Board Size Selection (User Input for Game Setup)

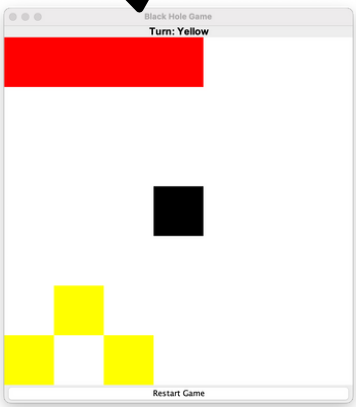
- Event Trigger: User selects a board size option from a dialog box.
- Event Handler: The getBoardSizeFromUser() method in the Main class.

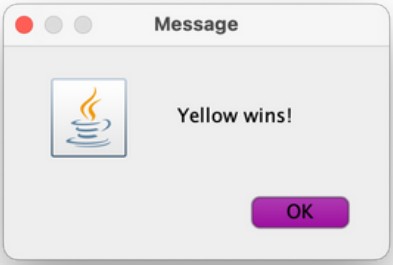

Black Hole Game
Turn: Red
Restart Game


Board Size Selection
Select Board Size:
9x9
7x7
5x5


Black Hole Game
Turn: Red
Restart Game


Black Hole Game
Turn: Yellow
Restart Game


Black Hole Game
Turn: Yellow
Restart Game


Message
Yellow wins!
OK

Tests

AS A player
I WANT TO Start game with 9x9 map
GIVEN given options
WHEN When I open game
THEN The game gets started with 9x9 map

AS A player
I WANT TO Start game with 7x7 map
GIVEN given options
WHEN When I open game
THEN The game gets started with 7x7 map

AS A player
I WANT TO Start game with 5x5 map
GIVEN given options
WHEN When I open game
THEN The game gets started with 5x5 map

AS A player
I WANT TO select spaceship to move
GIVEN the spaceships
WHEN when I click on that spaceship
THEN spaceships I clicked on gets selected

AS A player
I WANT TO move spaceship to right
GIVEN selected spaceship with no obstacle at right
WHEN I press at rigth arrow key
THEN spaceship moves until first obstacle at the right

AS A player
I WANT TO move spaceship to right
GIVEN selected spaceship with no obstacle at left
WHEN I press at rigth arrow key
THEN spaceship moves until first obstacle at the left

AS A player
I WANT TO move spaceship to right
GIVEN selected spaceship with no obstacle at up
WHEN I press at rigth arrow key
THEN spaceship moves until first obstacle at the up

AS A player
I WANT TO move spaceship to right
GIVEN selected spaceship with no obstacle at down
WHEN I press at rigth arrow key
THEN spaceship moves until first obstacle at the down

AS A player
I WANT TO close the game
GIVEN red close button on the left up corner
WHEN I press at close button
THEN confirmation tab opens up that asks if I want to exit

AS A player
I WANT TO not close the game
GIVEN confirmation tab
WHEN I press at no button
THEN it returns back to the game

AS A player
I WANT TO close the game
GIVEN confirmation tab
WHEN I press at yes button
THEN it closes the game

AS A player
I WANT TO restart game
GIVEN restartGame button at the bottom
WHEN I press at restartGame button
THEN confirmation tab pops out asking if I really want to restart

AS A player
I WANT TO not restart the game
GIVEN confirmation tab
WHEN I press at no button
THEN it returns back to the game

AS A player
I WANT TO restart the game
GIVEN confirmation tab
WHEN I press at yes button
THEN map size selection button pops up

AS A player
I WANT TO restart game with 9x9 map
GIVEN given options
WHEN When I open game
THEN The game gets started with 9x9 map

AS A player
I WANT TO restart game with 7x7 map
GIVEN given options
WHEN When I open game
THEN The game gets started with 7x7 map

AS A player
I WANT TO restart game with 5x5 map
GIVEN given options
WHEN When I open game
THEN The game gets started with 5x5 map

AS A player
I WANT TO move spaceship to obstacle(border, another spaceship)
GIVEN board and selected spaceship
WHEN click in that direction
THEN spcaeship remains still and move does not pass