BATTLESHIP GAME USING JAVA

Huzaifa Jamil Osman 2412135042
Abrar Rahman 2411607042
Mobasshir Amin Efaz 2411351642

Rules for a BattleShip:

- Each player begins with a 10x10 grid and places 5 ships on their grid. All ships are 1 unit wide, and with lengths of 6, 5, 4, 3, and 2. These ships can be placed either horizontally or vertically.
- The game starts with the player and their opponent not knowing where the opponent's ships are.
- After the preparation stage of placing ships the game alternates between players allowing them to select one new position on the opponent's grid that has not yet been attacked.
- The positions are marked once they have been attacked, either with a blue marker indicating it was a miss, or a red marker indicating a ship was hit.
- Once a ship has been destroyed from all grid positions having been hit, the ship is revealed as destroyed to the other player.
- The game ends when all ships for either player's side have been destroyed. The winner is the player with ships still not destroyed.

AI logic:

The AI will choose random moves unless it has hit a ship that is not yet destroyed. It will attack random cells until the ship is destroyed.

Game Controls:

- At any time:
    - Escape to quit.
    - R to restart.
- During Placement Phase:
    - Click to place ship (only places the ship if it is a valid placement).
    - Z to rotate between vertical and horizontal for placement.
- During Attack Phase:

Click on the enemy’s grid in places that have not yet been marked to reveal the squares as hits or misses.

Structure of the Game:

The following briefly explains the classes and what each class does.

Representing the Grid:

- Marker: Represents the coloured red or blue markers that show up when attacking on the grid. All 10x10 of these are created at the start and drawn if they have been marked. They keep track of the ships and which coordinates have been marked by both the player and the AI. This data is represented visually, which the AI uses to calculate it’s next move
- Grid: The actual grid that contains a collection of markers, ships, and draws these to show where cells are. Includes appropriate methods to manage the state of the grid.
    - Ship: Defines how a ship will be represented. It keeps track of where it is located on the grid and when it should be drawn. The ship can also track whether it has been destroyed to notify other classes when asked.
    - Coordinate: Defines a representation of a Cartesian coordinate, where each square in the grid can be denoted in the form of (x, y).

- GridBox: Defines a representation of a quadrilateral, which is used in defining the grids, and checks the limits of the quadrilateral (whether it is inside or outside the board).
- AI: The class that defines how the AI plans its attacks. It is important to note that the current AI is extremely simple, only targeting coordinates that have not been hit randomly.
- MessageBoard: This represents the message board beside the two grids that prints the current state of the game, whether it is the ship-placing phase, the game phase, or the game-finished phase.

High level interface:

· Game2: Creates the JFrame to contain the JPanel which contains the game and manages collection of the keyboard input. It is the main class.

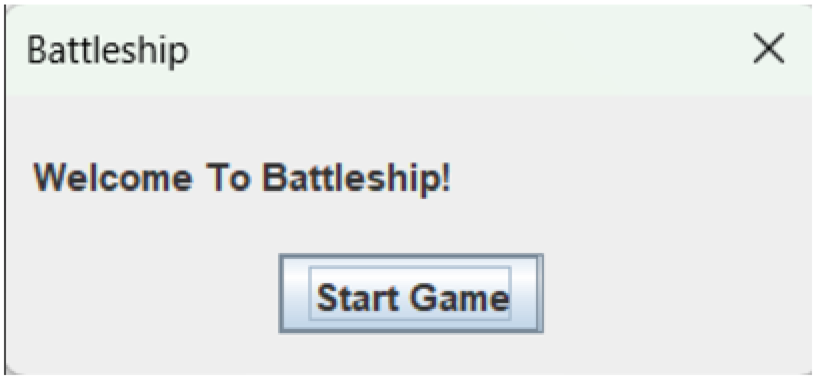
· ControlPanel: Initializes player and computer grids, integrates an AI opponent, and handles user input via mouse for ship placement and shooting, and keyboard for controls like toggling ship orientation, restarting, and exiting.

  Implements ship placement validation with real-time feedback (e.g., color changes for valid/invalid positions) and manages turn-based gameplay between the player and AI, checking for hits, misses, and destroyed ships.

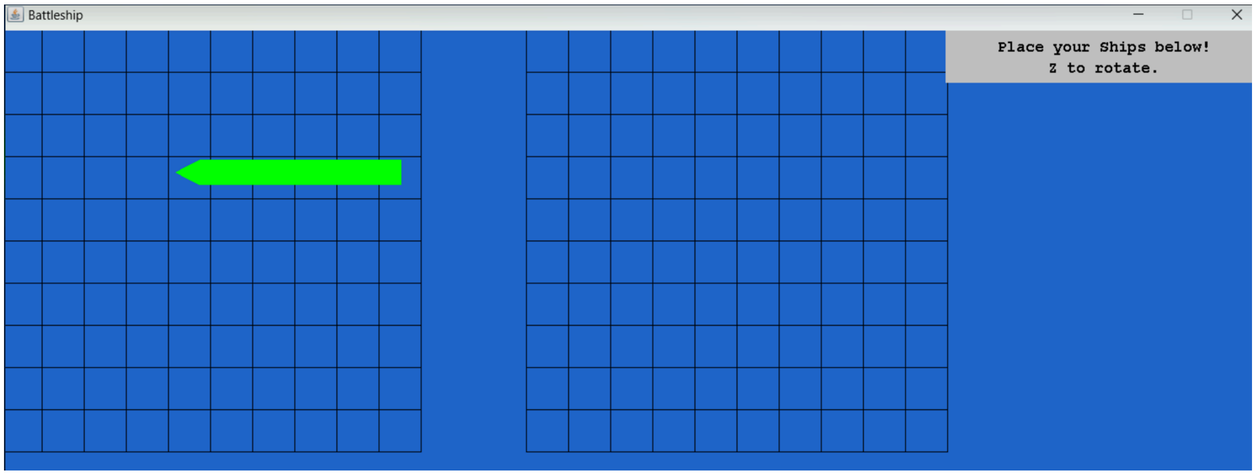
  Provides a restart mechanism to reset grids, ships, AI, and game state, and dynamically updates game visuals using an overridden “paint” method.

Explanation using Visual Representation:

The Start Menu - Consists of a button that starts when clicked, or user can press enter.

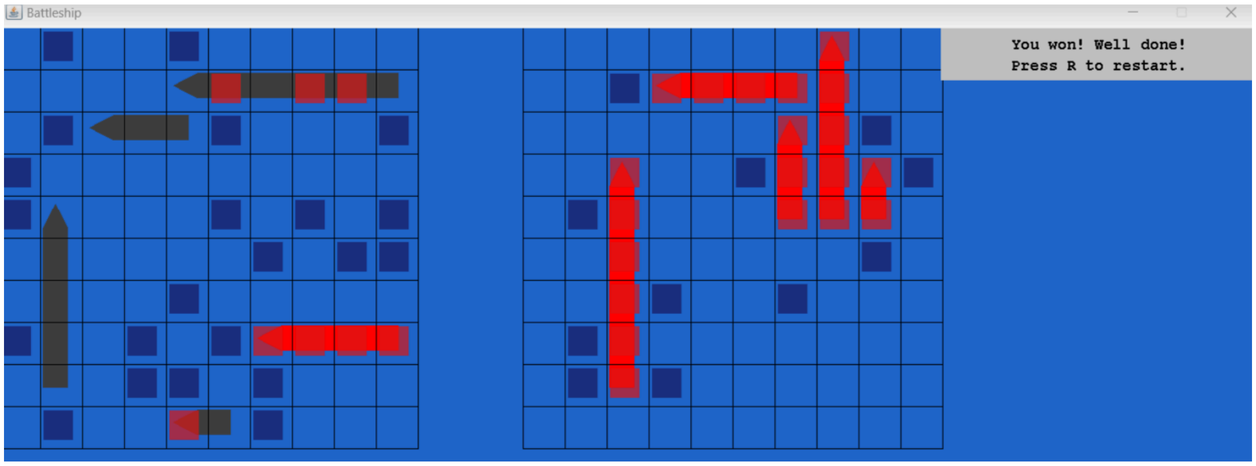


GridPanel - Consists of 2 grids, AI and Player. Ship is placed and can be rotated by pressing z.



Place your Ships below!
Z to rotate.

Game begins and coordinates are hit. If a ship is hit, the cell will turn red and if it misses, it will turn blue as shown. Once the game is finished, users can play again by pressing R or quit by pressing Esc.



You won! Well done!
Press R to restart.