**1. Square Click Handling (Gameplay Move Logic)**

**Function:** squareClick

**Purpose:** This function handles what happens when a player clicks on a grid square. It ensures that a player can only click on an empty square, then marks the square with the current player's symbol (either a "cross" or a "circle"). After marking the square, the function increments the move count and updates the game state.

**Algorithm:**

* Check if the square already has a "cross" or "circle" class.
* If not, add the class corresponding to the current player's symbol.
* Call incrementMove() to update the move counter, switch players, and check if the game has been won or drawn.

**2. Win Checking**

**Function:** checkForWin

**Purpose:** This function checks if the current move results in a win. A win occurs when any row, column, or diagonal contains three identical symbols (either all "crosses" or all "circles").

**Algorithm:**

* Define all possible winning combinations as arrays of squares.
* Iterate over each winning combination.
* For each combination, check if all squares contain the same symbol ("cross" or "circle").
* If a winning combination is found, update the winner's score, display the winning message, and trigger the playerWon() function.

**3. Tie Checking**

**Function:** checkForTie

**Purpose:** This function checks if the game has ended in a draw. A draw occurs when all squares are filled, and no player has won.

**Algorithm:**

* Collect all squares into an array.
* Check if every square is filled with either a "cross" or a "circle".
* If all squares are filled and no one has won (playerHasWon is false), update the information text to display "It's a Draw!" and call continueGame() to reset the game after a short delay.

Each of these algorithms plays a crucial role in managing the game's state and ensuring a smooth gameplay experience.