Project 9: XO Game

Concept / Idea:
Our app is a simple two-player XO game implemented in Jack. Players take turns marking 'X' or 'O' on a 3x3 grid by pressing keys 1-9. The game detects winning conditions and announces the winner or a tie when the board is full.

Motivation:
We chose to create an XO game because it's a classic game that reminds us of our childhood, and we thought it would be fun to play a childhood game on a computer we built.

Google Drive Link to Video

Names and Emails:
- Ohad Ben Amram, ohad.benamram@post.runi.ac.il
- Dvir Ben David, dvir.bendavid@post.runi.ac.il

Architecture:
- XO.jack: This class handles the game logic, including the board state, player turns, and win/tie conditions.
Fields:
1. **Array squares:**
   - Represents the 3x3 game board. Values: 0 (empty), 1 ('O'), 2 ('X').
2. **boolean playerO:**
   - Tracks the current player: true for 'O', false for 'X'.

Key Methods:
1. **constructor XO new():**
   - Initializes the game: creates the squares array, draws the board, and displays instructions.
2. **method void selection():**
   - Handles player input (keys 1-9) and validates moves.
     Calls nextTurn() to update the board and switch players.
3. **method void nextTurn(int location, int x, int y, boolean playerO):**
   - Places 'O' or 'X' on the board based on the current player.
     Calls drawO() or drawX() to render the move.
4. **method void drawBoard():**
   - Draws the 3x3 grid using Screen.drawLine().
5. **method void startingText():**
   - Displays game instructions and the initial board layout.
6. **method boolean isFull():**
   - Checks if the board is full (no empty squares).

7. **Drawing Methods:**
    - drawO(int x, int y): Draws an 'O' at the specified coordinates using Screen.drawCircle().
    - drawX(int x, int y): Draws an 'X' at the specified coordinates using Screen.drawLine().
8. **method boolean equalTriplet(int x, int y, int z):**
    - Checks if three values form a winning line. Announces the winner if a winning condition is met.
9. **method boolean win():**
    - Checks all possible winning conditions (rows, columns, diagonals). Returns true if a player wins.

How It Works:

- The game starts by initializing the board and displaying instructions.
- Players take turns pressing keys (1-9) to place 'X' or 'O' on the board.
- After each move, the game checks for a win or tie using win() and isFull().
- If a winning condition is met, the game announces the winner. If the board is full, it declares a tie.

- Main.jack: This class initializes the game and runs the main loop until a win or tie is detected.