Software Requirements Specifications(SRS):

1. **Introduction**
2. **Functional Requirements**
3. **Non-Functional Requirements**
4. **Constraints**
5. **Assumptions and Dependencies**
6. **Verification and Validation**
7. **Use case Diagram**
8. **Sequence Diagram**

1. Introduction

- Purpose: The purpose of this tic-tac-toe game is to provide users with a digital version of the classic two-player board game.

- Scope: The game will include a graphical user interface (GUI) where two players can take turns marking spaces on a 3x3 grid to achieve a winning pattern.

2. Functional Requirements

- **FR1: Game Board**

- Description: Display a 3x3 grid where players can mark X or O in empty cells.

- Inputs: Player clicks on an empty cell to mark it with X or O

- Outputs: Updated display showing the current state of the game board.

- **FR2: Player Turns**

- Description: Allow two players to take turns playing the game.

- Inputs: Player actions (clicking on cells to place their marker).

- Outputs: Notification of whose turn it is (Player 1 or Player 2).

- **FR3: Winning Condition**

- Description: Determine when a player has achieved a winning pattern (three X's or three O's in a row, column, or diagonal).

- Inputs: Game board state after each player's turn.

- Outputs: Announcement of the winner or a draw when the game ends.

-**FR4: Restart Game**

- Description: Allow players to restart the game after a win, loss, or draw.

- Inputs: Player action (clicking a "Restart" button).

- Outputs: Reset the game board and allow players to start a new game.

3. Non-Functional Requirements

- **NFR1: User Interface**

- The game should have a user-friendly interface with clear labels and intuitive controls (clickable cells for marking X or O).

-**NFR2: Performance**

- The game should respond to player actions without noticeable delays (e.g., marking cells, updating game state).

-**NFR3: Compatibility**

- The game should run on multiple platforms (Windows, macOS, Linux) and be playable in popular web browsers.

4. Constraints

- The game must adhere to the rules of tic-tac-toe, including valid moves and winning conditions.

- Graphics and animations should be simple to maintain a lightweight and responsive gameplay experience.

5. Assumptions and Dependencies

- **Assumption**: Players have basic familiarity with tic-tac-toe rules and gameplay.

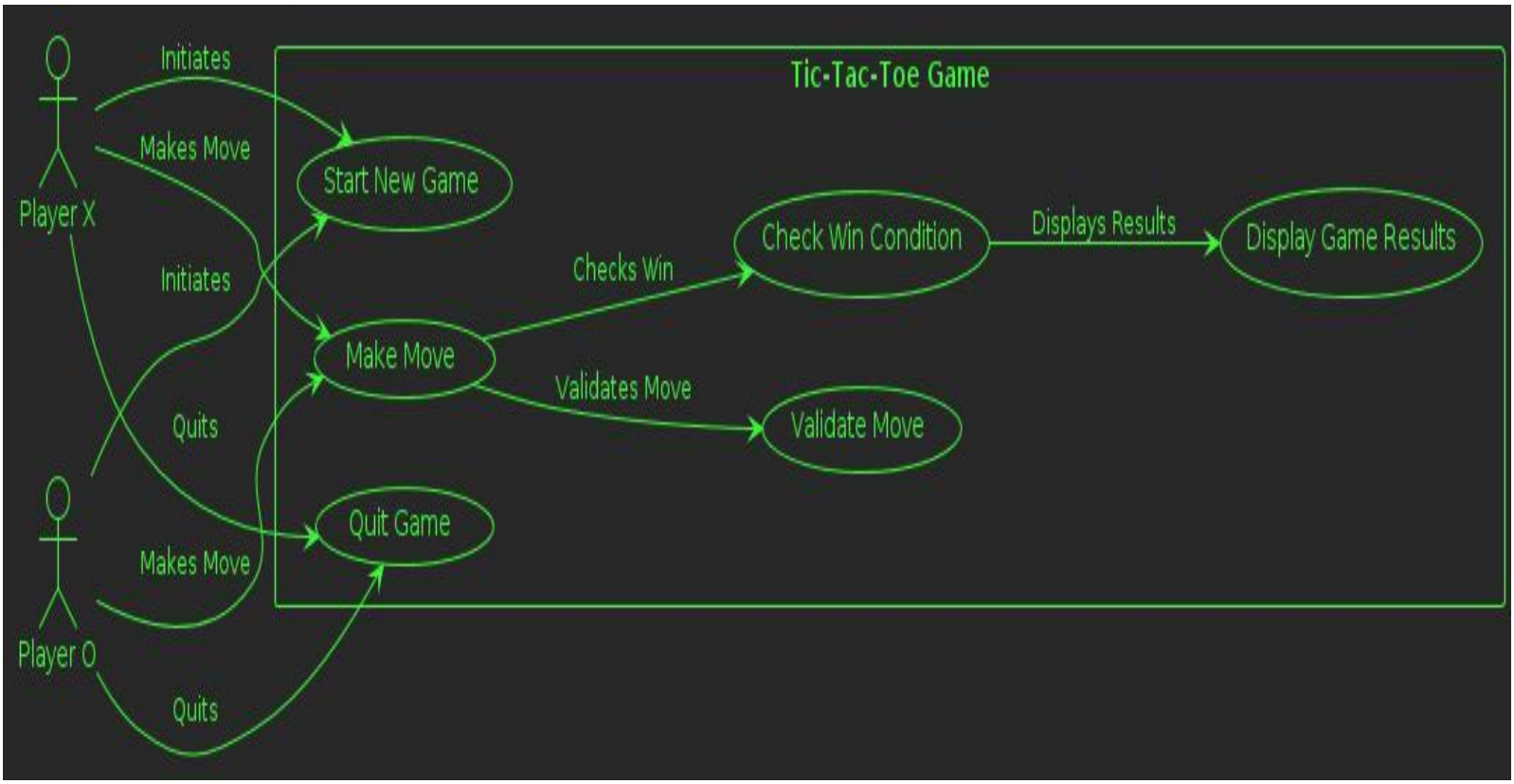
- **Dependency**: The game relies on stable rendering and input handling provided by the chosen programming environment (e.g., Unity, HTML5 canvas).

6. Verification and Validation

- **Verification**: Requirements will be verified through testing scenarios covering gameplay mechanics, winning conditions, and interface usability.

- **Validation**: The final game will be validated through playtesting to ensure it meets user expectations and performs correctly under different usage scenarios.

7. Use Case Diagram



9. **Sequence Diagram**