**Use Case: Start Game**

**Iteration:** 1st iteration

**Primary actor:** Player1, Player2

**Goal in context:** Begins a new Tic-Tac-Toe game session

**Preconditions:** The system is powered on and user has selected Tic-Tac-Toe as their game of choice

**Trigger:** Either Player1 or Player2 (depending on host) selects Tic-Tac-Toe as their game of choice with the “Tic-Tac-Toe” button)

**Scenario:**

1. Player selects Tic-Tac-Toe game option
2. The system will initialize a Tic-Tac-Toe session
3. The session will create any required information before setting up the board (visualization of the game).

**Post Conditions:** Game is setup and ready to print board (visual) to users

**Exceptions:**

* The button for Tic-Tac-Toe was unresponsive
* The system encounters an error while initializing
  + Errors
  + Missing information

**Priority: High -** the first initialization of the game and setups are important to the functionality of GUI and game logic.

**When Available:** Within 1 sprint (1st iteration)

**Frequency of Use:** Once per game session

**Channel to actor:** Interaction of a click of the Tic-Tac-Toe game option with their mouse

**Secondary actor:** N/A.

**Channel to Secondary Actors:** N/A

**Open issues:**

* The setup of the game (any information required) is finished before GUI implementation

**Use Case: Setup Board**

**Iteration:** 1st iteration

**Primary actor:** Player1, Player2

**Goal in context:** The game was set up and the board is ready to be displayed to the user. The game will call a function to print the board.

**Preconditions:** The game was properly setup as required with the GUI ready for response from printBoard function.

**Trigger:** The system has set up the game and will now call a function for printing the board.

**Scenario:**

1. Game information is setup and properly initialized
2. Game calls to a printBoard function to display to user

**Post Conditions:** board is set up (the visual) and ready for Player1 to make their move

**Exceptions:**

* The function call was unresponsive
* The system encounters an error while starting the visual
  + Possibly incorrect formatting

**Priority: High -** visual representation should be set up in-order for users to interact with the game.

**When Available:** Within 1 sprint (1st iteration)

**Frequency of Use:** Once per game session

**Channel to actor:** When they have selected Tic-Tac-Toe as their game of choice (is part of initialization)

**Secondary actor:** N/A.

**Channel to Secondary Actors:** N/A

**Open issues:**

* Board was incorrectly setup

**Use Case: Player Move on click**

**Iteration:** 1st iteration

**Primary actor:** Player1, Player2

**Goal in context:** Player1/player2 can make their move by clicking any of the tiles of tic-tac-toe to place their symbol (X or O) onto the board.

**Preconditions:** The game board was set up and is ready for user input.

**Trigger:** The user has clicked a tile to place their piece.

**Scenario:**

1. The board is ready for user to input their piece
2. User clicks the tile they desire to place their piece
3. The piece is placed in the desired spot

**Post Conditions:** the user has successfully placed their piece onto the board and will now prompt the other user to place their piece

**Exceptions:**

* The piece was not placed in the correct spot
  + The piece not placed at all
* The system encounters an error placing piece

**Priority: High -** Placing their pieces is the main functionality of the Tic-Tac-Toe game.

**When Available:** Within 1 sprint (1st iteration)

**Frequency of Use:** N times per player

**Channel to actor:** Interaction of a click on the tile they desire to place their symbol with their mouse

**Secondary actor:** N/A

**Channel to Secondary Actors:** N/A

**Open issues:** N/A

**Use Case: Update board**

**Iteration:** 1st iteration

**Primary actor:** Player1, Player2

**Goal in context:** After any user input, the board is updated and displayed to the user.

**Preconditions:** users have selected their move and the board needs to be updated.

**Trigger:** user has selected their move.

**Scenario:**

1. Player has made their move
2. GUI collects this information and passes to game logic
3. New updated game board is passed back to GUI to display

**Post Conditions:** The new updated board is returned to GUI to display to users

**Exceptions:**

* Board was not updated properly
* An error has occurred
  + No updates to board

**Priority: High -** In order to understand what is happening in the game, it is important to continually update the board after each player input.

**When Available:** Within 1 sprint (1st iteration)

**Frequency of Use:** N times per player

**Channel to actor:** player has clicked a tile to make their move and system calls to update the board

**Secondary actor:** N/A

**Channel to Secondary Actors:** N/A

**Open issues:** N/A

**Use Case: Check Winner**

**Iteration:** 2nd iteration

**Primary actor:** Player1, Player2

**Goal in context:** After each player move, the system should check if there is a winner every turn

**Preconditions:** the user has placed a symbol on the board and the board was updated properly

**Trigger:** the board was updated.

**Scenario:**

1. The board was updated based on player’s move
2. The system will check if a winner was detected

**Post Conditions:** system successfully checks if there was a winner detected

**Exceptions:**

* Board was updated incorrectly and checking winner returns incorrect information
* Conditions were never checked

**Priority: High -** determines the game conditions in order to win and end the game

**When Available:** Within 2 sprint (2st iteration)

**Frequency of Use:** N times per player

**Channel to actor:** once the board has been updated, system calls to check if there is a winner based on move

**Secondary actor:** N/A

**Channel to Secondary Actors:** N/A

**Open issues:**

* Game logic is incorrect

**Use Case: Check Tie**

**Iteration:** 2nd iteration

**Primary actor:** [fill here]

**Goal in context:** [fill here]

**Preconditions:** [fill here]

**Trigger:** [fill here]

**Scenario:** [fill here]

**Post Conditions:** [fill here]

**Exceptions:** [fill here]

**Priority:** [fill here]

**When Available:** [fill here]

**Frequency of Use:** [fill here]

**Channel to actor:** [fill here]

**Secondary actor:** [fill here]

**Channel to Secondary Actors:** [fill here]

**Open issues:** [fill here]

**Use Case: Announce Winner**

**Iteration:** 2nd iteration

**Primary actor:** Player1, Player2

**Goal in context:** A winner was found and a winner will be announced and displayed to GUI

**Preconditions:** a winner was found

**Trigger:** System has detected a winner

**Scenario:**

1. Winning conditions were found
2. A winner announcement is prompted to GUI

**Post Conditions:** A winner announcement is prompted to GUI

**Exceptions:**

* Announces incorrect prompt
  + Announces incorrect winner

**Priority: Low -** can be implemented at any stage of the program as long as winning conditions are correct

**When Available:** Within 2 sprint (2st iteration)

**Frequency of Use:** once per game

**Channel to actor:** winner was found and system announces winner to GUI

**Secondary actor:** N/A

**Channel to Secondary Actors:** N/A

**Open issues:** N/A

**Use Case: Announce No winner**

**Iteration:** 2nd iteration

**Primary actor:** Player1, Player2

**Goal in context:** No winner was found (board is full) and will be announced and displayed to GUI

**Preconditions:** no winner was found

**Trigger:** no winner was found

**Scenario:**

1. No winner was found
2. A no winner announcement is prompted to GUI

**Post Conditions:** A no winner announcement is prompted to GUI

**Exceptions:**

* Announces incorrect prompt

**Priority: Low -** can be implemented at any stage of the program as long as tie conditions are correct

**When Available:** Within 2 sprint (2st iteration)

**Frequency of Use:** once per game

**Channel to actor:** A tie game was found and system announces tie to GUI

**Secondary actor:** N/A

**Channel to Secondary Actors:** N/A

**Open issues:** N/A

**Use Case: Reset Board**

**Iteration:** [fill here]

**Primary actor:** [fill here]

**Goal in context:** [fill here]

**Preconditions:** [fill here]

**Trigger:** [fill here]

**Scenario:** [fill here]

**Post Conditions:** [fill here]

**Exceptions:** [fill here]

**Priority:** [fill here]

**When Available:** [fill here]

**Frequency of Use:** [fill here]

**Channel to actor:** [fill here]

**Secondary actor:** [fill here]

**Channel to Secondary Actors:** [fill here]

**Open issues:** [fill here]

**Use Case: End Game**

**Iteration:** [fill here]

**Primary actor:** [fill here]

**Goal in context:** [fill here]

**Preconditions:** [fill here]

**Trigger:** [fill here]

**Scenario:** [fill here]

**Post Conditions:** [fill here]

**Exceptions:** [fill here]

**Priority:** [fill here]

**When Available:** [fill here]

**Frequency of Use:** [fill here]

**Channel to actor:** [fill here]

**Secondary actor:** [fill here]

**Channel to Secondary Actors:** [fill here]

**Open issues:** [fill here]

**Use Case: Play Again Button**

**Iteration:** [fill here]

**Primary actor:** [fill here]

**Goal in context:** [fill here]

**Preconditions:** [fill here]

**Trigger:** [fill here]

**Scenario:** [fill here]

**Post Conditions:** [fill here]

**Exceptions:** [fill here]

**Priority:** [fill here]

**When Available:** [fill here]

**Frequency of Use:** [fill here]

**Channel to actor:** [fill here]

**Secondary actor:** [fill here]

**Channel to Secondary Actors:** [fill here]

**Open issues:** [fill here]

**Use Case: Exit**

**Iteration:** [fill here]

**Primary actor:** [fill here]

**Goal in context:** [fill here]

**Preconditions:** [fill here]

**Trigger:** [fill here]

**Scenario:** [fill here]

**Post Conditions:** [fill here]

**Exceptions:** [fill here]

**Priority:** [fill here]

**When Available:** [fill here]

**Frequency of Use:** [fill here]

**Channel to actor:** [fill here]

**Secondary actor:** [fill here]

**Channel to Secondary Actors:** [fill here]

**Open issues:** [fill here]