**1. Name: Pressing Play**

**Identifier** - UC1

**Description -** Start game once the player of the game presses the ‘play’ button.

**Goal** - Initializing the game, allow simple option to allow for game to be played.

**Preconditions**

1. User loads the game and is on the main menu
2. Game ends (one or the other player beats the other or ends in a tie)

**Assumptions**

1. Human player as an actor

**Frequency** This use case will be used as many times as the player wishes to play tic-tac-toe.

**Basic Course**

1. Use case starts when user presses play.
2. Use case ends when the game begins and players are assigned and icon.

**Alternate Course A: User decides to exit game instead of pressing play**

**Condition:** User exits program by clicking ‘x’ in top right corner.

1. No alternate steps are taken, game closes.

**Exception Course A:**

**Condition:** N/A

1. No other planned features which would significantly alter flow of events.

**Post conditions**

1. User starts a game and will be assigned either ‘X’ or ‘O’.

**Actors**

1. Human player

**Included Use Cases** N/A

**Extended Use Case** N/A

**Notes**

No current issues with this feature of our program**.**

**2. Name: Making a Move**

**Identifier** - UC2

**Description** - Once the game is started and the user is assigned either ‘X’ or ‘O’ and the ‘person’ assigned marker ‘X’ is allowed to go first.. The user will then be asked to place with ‘X’ or ‘O’ on block of the tic tac toe board. This prompts the AI to make a corresponding move to answer this.

**Goal** - The goal is for the user (human actor) to finish his turn by completing a move on the board.

**Preconditions**

1. The game needs to first be started
2. Player needs to have been assigned their marker by the game.

**Assumptions**

1. That the space the user is moving into isn’t occupied.

**Frequency** from 5 to 8 times per game

**Basic Course**

1. Use case begins when user chooses a square
2. AI picks a square depending on what the human actor chooses to do
3. Use case ends when there is a decided outcome of who wins and loses

**Alternate Course A: N/A**

**Condition: N/A**

**Exception Course A:** The user attempts to move into an occupied square

**Condition:** User picks a location to place either ‘X’ or ‘O’ but the location is already occupied

1. A message is displayed saying that the square is occupied and that they should choose another square

**Exception Course B:**

1. The user cannot hope to win the game

**Condition:** AI detects user can’t win

1. Game goes to game ends case

**Exception Course C:**

1. The AI detects that the game must end in a tie

**Condition:** AI detects that neither the AI or human actor can win the game

1. Game goes to game ends case

**Post conditions**

1. User is given an option to play again or quit game

**Actors**

1. Human actor
2. AI

**Included Use Cases** N/A

**Extended Use Case** N/A

**Notes**

No current issues

**3. Name: Game ends (one or the other player beats the other or ends in a tie)**

**Identifier** - UC3

**Description** - User either wins, loses, or ties with the AI. User is presented with their

stats, a menu button that returns them to the main menu and a quit button that allows them to quit the game.

**Goal** - To decide if the current game is over, present the user with their overall/game stats, and move user to a case where to can choose to quit or play again.

**Preconditions**

1. The “Making Moves” case must come before this because moves have to be made in order for one of the conditions to be met.

**Assumptions**

1. We can assume that the current instance of the game has concluded.

**Frequency** This will happen as many times as the player decided to play the game. Unless they prematurely exit the application.

**Basic Course**

1. Use case begins when we exit the “making a move case”
2. Use case ends when we enter the “press play” and give the option to user to play or quit the game.

**Alternate Course A: N/A**

**Exception Course A: N/A**

**Post conditions**

1. User would be taken back to the main menu where they have the choice between the use cases for ‘pressing play’, ‘changing difficulty’, and ‘checking the rules’.
2. User quits the game and goes outside to experience life.

**Actors**

1. Human player

**Included Use Cases** N/A

**Extended Use Case** N/A

**Notes**

No current issues with this feature of our program**.**

**4. Name: Changing the Difficulty**

**Identifier** - UC4

**Description** - User clicks option button from main menu which takes them to a screen

where they can change AI difficulty.

**Goal** - To present the user with a list of difficulty options.

**Preconditions**

1. User is on the main menu after either having started the game for the first time, or after played through a game and deciding to return to menu instead of quitting.

**Assumptions**

1. We can assume that the user wants to change the difficulty.

**Frequency** This will happen as long as the user wants to keep playing the AI versus different difficulties.

**Basic Course**

1. Use case begins when user clicks on the options button from main menu.
2. User chooses a difficulty they would like the AI to be set at.
3. User presses back to return to the main menu.

**Alternate Course A:**

1. User presses back to return to main menu instead of choosing a difficulty.

**Exception Course A:**

1. User exits the game by closing the window.

**Post conditions**

1. User is back on the main menu.

**Actors**

1. Player

**5. Name: Checking the Rules**

**Identifier** - UC5

**Description** - User clicks ‘rules’ button from main menu which takes them to a screen

where they can view the Tic Tac Toe Rules.

**Goal** - To present the user with the rules for Tic Tac Toe

**Preconditions**

1. User is on the main menu after either having started the game for the first time, or after playing through a game and deciding to return to menu instead of quitting.

**Assumptions**

1. We can assume that the user is unfamiliar with the rules.

**Frequency** This will happen as long as the user wants to view the rules.

**Basic Course**

1. Use case begins when click on the ‘rules’ button
2. Use case ends when we click on the ‘back’ button

**Alternate Course A: N/A**

**Exception Course A: N/A**

**Post conditions**

1. User is back on main menu screen

**Actors**

1. Player

**6. Name: Undo a Move**

**Identifier** - UC6

**Description** - User presses CTRL-Z during a game to undo their last move and the

last AI move.

**Goal** - To allow the user to undo moves and retry different scenarios at their leisure.

**Preconditions**

1. User is in the middle of a game where they have moves that they can undo.

**Assumptions**

1. We will assume that the user wants to undo their move to try different moves for practicing or fun.

**Frequency** This will happen as long as there are moves to be undone and a game is in progress.

**Basic Course**

1. User makes a move.
2. User undoes move with CTRL-Z.

**Alternate Course A: N/A**

**Exception Course A: N/A**

**Post conditions**

1. User is in a previous boardstate after pressing CTRL-Z.

**Actors**

1. Player