
# SOEN 6011 SOFTWARE ENGINEERING PROCESS
## SUMMER 2016

## ASSIGNMENT 2

## Team 6

Submitted to:
Professor Nicolangelo Piccirilli

Prepared by:

| Team Member Names | Student ID |
|---|---|
| Vivek Khatri | 27292848 |
| Navneet Kaur | 27676638 |
| Sarvenaz Khaksarfard | 40002880 |
| Pardeep Kaur | 40014804 |
| Muhammad Raza Khan | 27735987 |
| Neha Kumari | 27382073 |
| Ankit Lathia | 27378327 |
| Mitesh Kaura | 27284756 |

# Contents

## Problem

We have to develop a game Tic Tac Toe on desktop as well as mobile platforms, which has three different modes. In the second deliverable we have to deliver a human vs human and human vs human tournament mode. We then have to submit the same game human vs computer but with three different levels of complexity from computers 'perspective; like easy, medium and difficult.

### Background Information

Tic Tac Toe is dual player game which can be played between two human players or between human and computer player. It is strategy game in which first player to make simultaneous three Noughts or crosses horizontally, vertically or diagonally wins the game.
We used Java, as programming language to build the game for Desktops and Android devices, for building the GUI of the game for Desktop version we used Java SWING. SWING is a widget toolkit which is an API for providing a graphical user interface for Java programs.
The first iteration would provide a GUI for the game with ability to click and display Noughts or Crosses and buttons for additional functionality of game, the second iteration would provide with a two player game where users can complete with each other in Android devices and the third deliverable tends to deliver a game which can be played against the computer at different levels and more user friendly functionalities.

# Deliverable 1

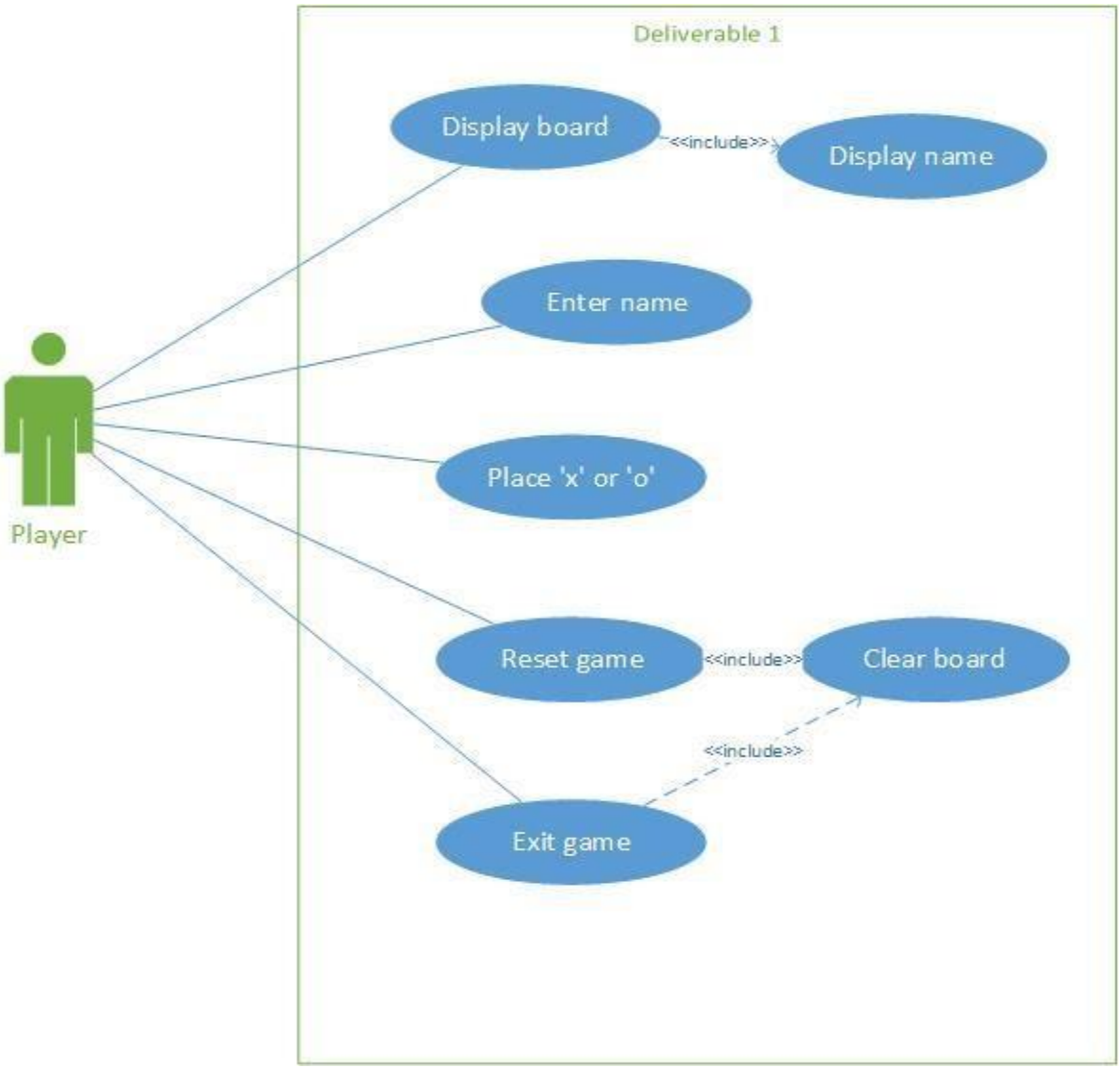
### Functional Requirements

- User can start a new game.
- System should display a 3*3 board to user.
- If user clicks on any square on a 3*3 board it should display "X" first and secondly "O" and then vice versa for further clicks.
- Users can enter their name and can reset the board.
- User can exit the game anytime.

### Non Functional Requirements

- **Learnability**: The system should be easy to learn for user as it will include help option.
- **Responsiveness:** The system should be ready to respond to a user's input no matter how frequently user clicks on square board.
- **Usability**: As it automatically displays the symbols by clicking on any box so it's easy to play.

- **Modifiability:** It should be easy to make any changes in the system without needing to be rebuilt.
- **Understandability:** User should easily comprehend what system does.


Deliverable 1
Display board
<<include>>
Display name
Enter name
Player
Place 'x' or 'o'
Reset game
<<include>>
Clear board
<<include>>
Exit game

## Use Case Scenarios

| | |
|---|---|
| Use Case ID | UC1 |
| Use Case Name | Display the board |
| Primary Actor | Tic Tac Toe Player 1 , Player 2 |
| Pre Conditions | Game should be downloaded in the system |
| Success Guarantee (Postconditions) | Game board is displayed to user |
| Main Success Scenario | 1. User starts the game<br>2. System displays the game board |
| Extensions (Alternative Scenario) | 2 (a). If there is any error while displaying the board, then system will display appropriate message to the player. |

| | |
|---|---|
| Use Case ID | UC2 |
| Use Case Name | Place “X” or “O” |
| Primary Actor | Tic Tac Toe Player 1 , Player 2 |
| Pre Conditions | 1) Game should be downloaded in the system<br>2) Space should be available on the board to click |
| Success Guarantee (Postconditions) | Noughts or crosses is displayed to user |
| Main Success Scenario | 1. User starts the game<br>2. Game board is displayed on the screen<br>3. User clicks on any of the 9 spaces on the board<br>4. User click will display either Noughts or crosses |
| Extensions (Alternative Scenario) | 3 (a). If there is no available space on the board, the board will reset. |

| | |
|---|---|
| Use Case ID | UC3 |
| Use Case Name | Reset the game |
| Primary Actor | Tic Tac Toe Player 1 , Player 2 |
| Pre Conditions | 1. Game should be downloaded in the system<br>2. Game board should be displayed to the user |
| Success Guarantee (Postconditions) | Clear game board should be displayed to the player |
| Main Success Scenario | 1. The user clicks reset button.<br>2. System clears the game board. |
| Extensions (Alternative Scenario) | 2 (a). If there is any error while reset, then system will display appropriate message to the player |

## Deliverable 2

### Functional Requirements

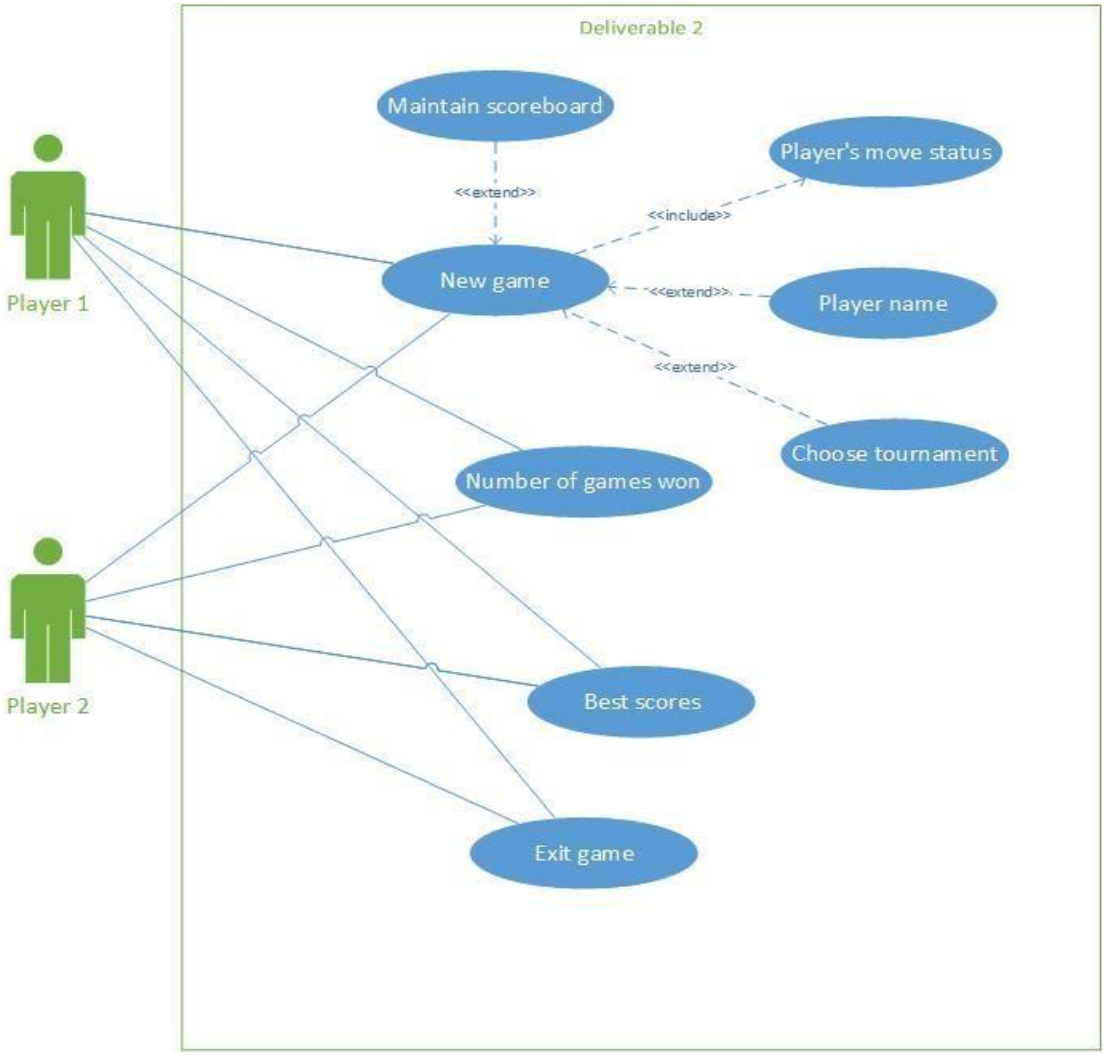
- User should be able to start a new game.
- The system should be able to maintain the scoreboard of the players playing the game
- The system should allow the user to play the game in tournament mode for best of 3 or best of 5 games.
- The system should show the status and the turn of each player for each move.
- The player should be able to enter their names, which would be displayed on the screen.
- As soon as first player is able to make a single line of 'X' horizontally, vertically or diagonally or second player is able to make a single line of 'O' horizontally, vertically or diagonally, a dialogue box should pop up which should display who is the winner of the game.

### Non Functional Requirements

- **Reliability –** The system should give proper responses as per the events and should not hang up or crash during the game.

- **Performance**: System should be fast enough to display the user selection and it should not involve the delay of more than 1.5 seconds.
- **Scalability**: Tic Tac Toe can support all the future advancements keeping all the previous functionality.
- **Testability**: As no major functionality is involved therefore, it is easy to test.
- **Portability**: System can run on different versions of the android.

## Use Case Diagram




## Use Case Scenarios

| | |
|---|---|
| Use Case ID | UC4 |
| Use Case Name | Play game |
| Primary Actor | Tic Tac Toe Player 1 , Player 2 |
| Pre Conditions | The game has to be installed in the Android mobile. |
| Success Guarantee (Postconditions) | The game should run successfully. |

| | |
|---|---|
| Main Success Scenario | 1. User starts the game<br>2. User clicks anywhere on the game board.<br>3. 'X' will be displaced for the first move and 'O' for the second player.<br>4. The moves will be displaced alternatively. |
| Extensions (Alternative Scenario) | 1 (a). If the system crash or freezes then appropriate error message should be displayed. |

| | |
|---|---|
| Use Case ID | UC5 |
| Use Case Name | Exit game |
| Primary Actor | Tic Tac Toe Player 1 , Player 2 |
| Pre Conditions | 1) Game should start successfully.<br>2) Game board should be displayed clearly |
| Success Guarantee (Postconditions) | Game board should exit successfully |
| Main Success Scenario | 1. Player starts the game<br>2. Player 1 and player 2 play the game against each other.<br>3. Player can exit the game at any point of time.<br>4. The player chooses to exit the game by clicking exit button.<br>5. System clears the game board. |
| Extensions (Alternative Scenario) | 3 (a). No response when clicked on the exit button<br>3 (b). Game freezes when clicked on the exit button |

| | |
|---|---|
| Use Case ID | UC6 |
| Use Case Name | Win game |
| Primary Actor | Tic Tac Toe Player 1, Player 2 |
| Pre Conditions | 1) Game should start successfully<br>2) Players start playing the game. |
| Success Guarantee (Postconditions) | Winning message is displayed on the screen. |
| Main Success Scenario | 1. User starts the game<br>2. User clicks anywhere on the game board.<br>3. 'X' will be displaced for the first move and 'O' for the second player.<br>4. The moves will be displaced alternatively.<br>5. One player successfully places his marks in horizontal, vertical or diagonal rows.<br>6. Then the player wins the game<br>7. The game ends and wining message is displayed. |
| Extensions (Alternative Scenario) | 7 (a). Winning message not displayed to the user after finishing the game |

| | |
|---|---|
| Use Case ID | UC7 |
| Use Case Name | New game |
| Primary Actor | Tic Tac Toe Player 1 , Player 2 |
| Pre Conditions | The game should be downloaded in the system. |
| Success Guarantee (Postconditions) | Game board should be displayed to the user. |
| Main Success Scenario | 1. Players start playing the game.<br>2. Game successfully completed.<br>3. Then the user can start a new game by clicking on the start button. |

| | |
|---|---|
| | 4. Game board is displayed. |
| Extensions (Alternative Scenario) | 3(a). User click on start button before the game ends. |

## Deliverable 3

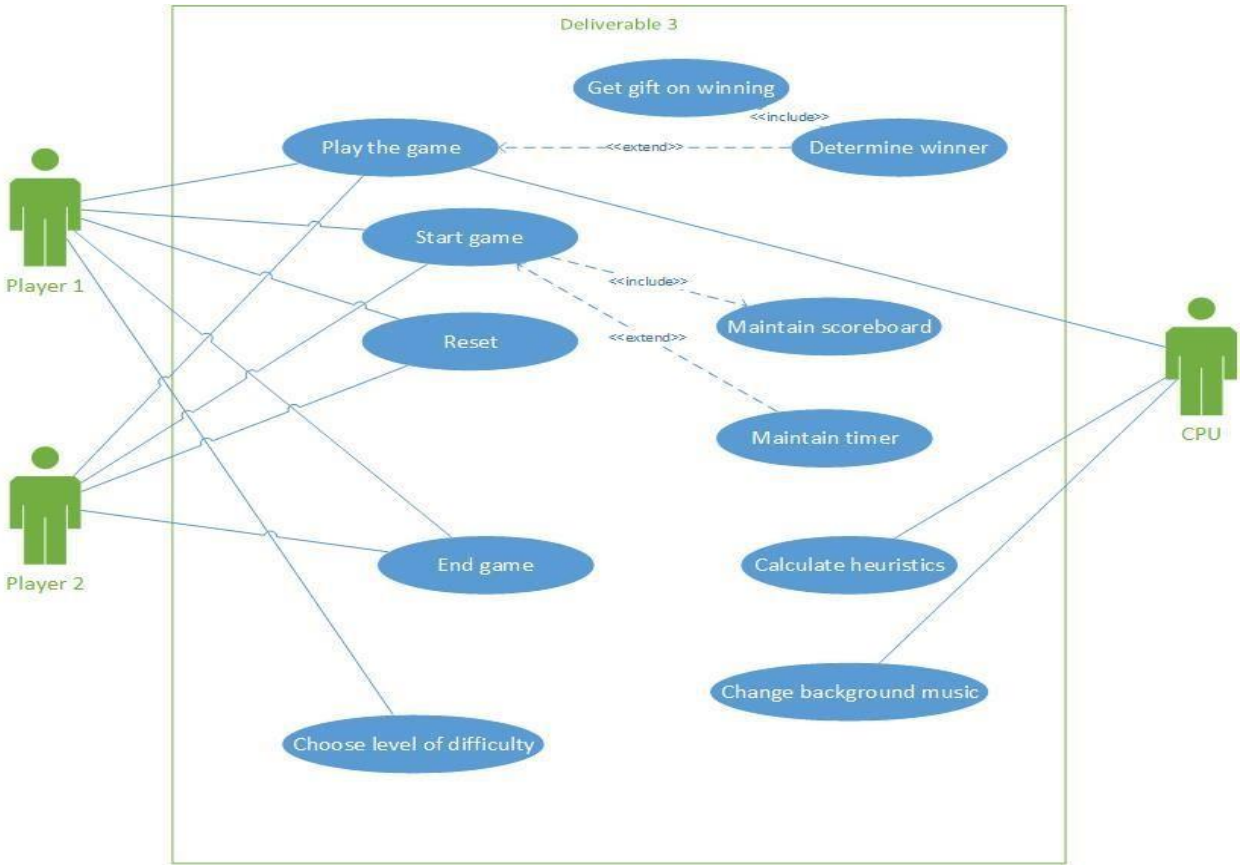
### Functional Requirements

- The CPU must be able to play the game against a human player.
- Game must have different levels of difficulty to play against the CPU.
- System maintains a scoreboard for each session.
- Background music for playing against the CPU should be different from the background music when playing against another player.

### Non Functional Requirements

1) **Performance**: System should be fast enough to display the user selection and it should not involve the delay of more than 2 seconds.
2) **Reliability**: System should deliver the correct results despite of certain amount of failures. System should work with a reliability of 99 percent, which means in a day player can fail to play no more than 1 %, or approximately 15 minutes.
3) **Scalability**: System should support the increase use and support the advancements.
4) **Usability**: User should be able to use the system without any difficulty. As it automatically displays the symbols by clicking on any box so it's easy to play.
5) **Maintainability:** This relates to the ease at which your system finds bugs and fixes them.

## Use Case Diagram




## Use Case Scenarios

| | |
|---|---|
| Use Case ID | UC8 |
| Use Case Name | Play against the CPU |
| Primary Actor | Tic Tac Toe Player |
| Secondary Actor | Game Board |
| Pre Conditions | The user should chose to play against CPU |
| Success Guarantee | The game should have a result |

| | |
|---|---|
| (Postconditions) | The game should have a result |
| Main Success Scenario | 1. The user chooses their level of difficulty against the CPU.<br>2. The players make their moves against the CPU.<br>2. The CPU counter attacks the player based on the difficulty.<br>4. The game/tournament ends with a definitive result. |
| Extensions (Alternative Scenario) | 2a. The player can reset or end the game at any move. |

| | |
|---|---|
| Use Case ID | UC9 |
| Use Case Name | Maintain a scoreboard |
| Primary Actor | Tic Tac Toe Player |
| Secondary Actor | Game Board |
| Pre Conditions | The user should successfully start the game |
| Success Guarantee (Postconditions) | The games should have a result |
| Main Success Scenario | 1. The user chooses to play games against human opponents/CPU.<br>2. The scorecard is update after each game successfully ends during each session of play.<br>3. The scorecard is reset once the user exits the game. |
| Extensions (Alternative Scenario) | 2a. The player can reset or end the game at any move. |

| | |
|---|---|
| Use Case ID | UC10 |
| Use Case Name | Maintain different levels of the game |
| Primary Actor | Tic Tac Toe Player 1 , Player 2 |
| Pre Conditions | The player should be able to select the level |
| Success Guarantee (Postconditions) | The new clear game board is displayed with the level selected. |

| | |
|---|---|
| | . |
| Main Success Scenario | 1. Players start the game.<br>2. Player select the level(Easy or difficult)<br>3. The clear game board is displayed to the user after selecting the level |
| Extensions (Alternative Scenario) | 2(a). Game is not reset after selecting the level from easy to difficult |