



SOEN 6011- SOFTWARE ENGINEERING PROCESS

SUMMER 2016

ASSIGNMENT 2

Instructor: Nicolangelo Piccirilli

GROUP 9

PROJECT – TIC TAC TOE GAME

TEAM NAME: CHALLENGERS

TEAM LEADER: KUMARAN AYYAPAN RAVI

Team Members:

NAME	EMAIL ID	STUDENT ID
A R ASHIQUR RAHMAN	rahmanarashiqur@gmail.com	27420668
REENAL RANA	reenalrana9@gmail.com	27326688
KUMARAN AYYAPAN RAVI	kumaranayyappan10@gmail.com	27285388
MILIND RAVINDRA PUKALE	milind.ise@gmail.com	27652887
AAKASHDEEP SINGH JAHAGIRDAR	jakashdeep@gmail.com	27728131
JYOTSNA RANA	jyotsnarana91@gmail.com	40013246
ADITHYA SAJJANAM	adithya3838@gmail.com	27723032
ARUMUGAM SHANMUGAM	arumugam.dav@gmail.com	40010992

Table of Contents

Team Members:	1
Problem Statement	3
Background	3
Environment and system requirements:	3
Functional Requirements	4
Deliverable 1	4
Deliverable 2	4
Deliverable 3	5
Non Functional Requirements	5
Deliverable 1	5
Deliverable 2	5
Deliverable 3	6
Use Case Analysis	6
Use Case Diagram	6
Deliverable 1:	6
Deliverable 2:	7
Deliverable 3:	7
Use Case Scenarios	8
Deliverable 1:	8
Deliverable 2:	9
Deliverable 3:	11
Conclusion	13
References	13

Problem Statement

Vision: To develop an application to play Tic-Tac-Toe game. The application will work in desktop application as well as mobile application which is limited to android devices. The application is designed to be played against human player, through a suitable interface. The player should be able to choose X and O on the game board.

In the First deliverable The Graphical User Interface (GUI) of the board will be designed, where 3x3 board will be displayed and player will be able to enter X and O's.

In the Second Deliverable, the actual rule and gameplay of the Tic-Tac-Toe will be implemented such as the following: Two players will mark X and O's in 3x3 grid. The player who succeeds in getting X's or O's in three consecutive blocks across horizontally, vertically or diagonally will win. If none gets three X's or O's in a row the game ends in draw.

In the third, the human player will be able to play against the computer. The Gameplay should be implemented with different difficulty levels where the Computer will aim to beat human player in the advanced difficult level. In Worst case the game may end in draw.

We will be developing the game following the iterative method. For each iteration, there will be functional and non-functional requirements which will be our goal to reach.

Background

[1] Tic-tac-toe is a paper- and-pencil based game which is very simple and interesting. There will be two players 'X' and 'O' who takes their respective turns to fill a 3 X 3 grid. The player who marks the respective symbols in a vertical, horizontal and diagonal first, wins the game. We develop a 3 X 3 tic-tac-toe game using Java Programming Language. The application is designed in such a way that only two players can play the game at an instance. The system will update and check the winning conditions throughout the gameplay.

Environment and system requirements:

The operating environments and system capabilities in order to develop the application are as follows:

Items/Tools	Size/Quantity/Speed
Intel® Core™ i5 CPU	2.50 GHz
Memory (RAM)	4 GB
Hard Disk	512 GB
System Type	64-bit OS
Windows OS	7 or later
Eclipse IDE	Mars (4.5)
Android Studio	V2.1.0
Mobile device	Android V 4.4
Junit	4 or later

WBS Schedule Pro	5.1
MS Project	2013
MS Office	2013
ArgoUML	0.34

Functional Requirements

Some of the functional requirements which represents the functionality that the application must have is listed below. Each requirements aims to empower users and meet the stakeholder's needs:

Deliverable 1

A stand-alone Java application that is able to show the board and draw an “X” or an “O” where the user clicks.

1. The game must display a game board that contains 9 cells (3*3).
2. The game board must have the options to minimize/maximize the screen.
3. The game must have the option to close the game anytime the user want.
4. The game must have the options to reset the board by the player whenever he/she wants.
5. The game board must be responsive to mouse clicks and it must show “X” on the respective cell the user clicks, and the next click must draw “O” on the respective cell.
6. The game must have a menu option which shows the rules of the game.
7. Different colors for ‘X’ or ‘O’ can be chosen.

Deliverable 2

A java mobile application that works on Android devices for the full tic-tac-toe game. (2 human players)

1. The user must be able to go to the home screen when he/she presses home button in the mobile.
2. The player must be able to start a new game.
3. The game must have a database which contains the top 3 scores and show how many games the player has won.
4. The game must provide an indicator or checkbox to represent each player and show whose turn it is.
5. The game must allow the player to play a tournament of best of 3 and 5.
6. The game must be able to determine the results when there is a straight line or diagonal of “X” or “O”.
7. The game must be drawn if all the squares on the board are filled up but neither of the player (distinguished by “X” and “O”) manages to get three consecutive marks along a straight line or diagonally.
8. The game must display a message announcing the result of the game (which player won or drawn).

Deliverable 3

A two player computer version of the game against a computer player that uses a heuristic to attempt to beat the human player. Should work on desktop or android mobile.

1. The game must have different difficulty level (amateur, intermediate, pro).
2. The game will have back a background music when playing against computer.
3. The player will have the options to choose X or O before starting the game in “amateur” difficulty level and on “intermediate” difficulty level, the first turn is chosen randomly and on “pro” level, the computer will start the game.
4. The game must give a virtual gift to the winner of each game.
5. The game must have a timer which will automatically finish the game if the game is idle for certain amount of time and announce the other player who was not idle as the winner.

Non Functional Requirements

Some of the Non- functional requirements are listed below which describes how the system should works and what are all the quality attributes it should have.

Deliverable 1

1. Response Time: The game TIC TAC TOE should respond to the user within .5 seconds.
2. Extensibility: The game should be extensible so that it can be modified to add more functionality.
3. Usefulness: Game should be feasible to the end users.
4. Style: The game design should be pleasant and attractive.
5. Consistency: The blocks in board should be deactivated after a click so that player cannot override the previous decision.

Deliverable 2

1. Efficient: The game should perform its intended function or its processing time should be minimal.
2. Capacity: The number of players, records and data should be maintained.
3. Operability: The game should be operated in reliably efficient manner.
4. Recoverability: Mean time to recover (MTTR) for the game should be less.
5. Portability: The game should have the ability to be used in different versions of android systems.
6. Stability: Game should be stable, to extend where capabilities and characteristics should remain same.

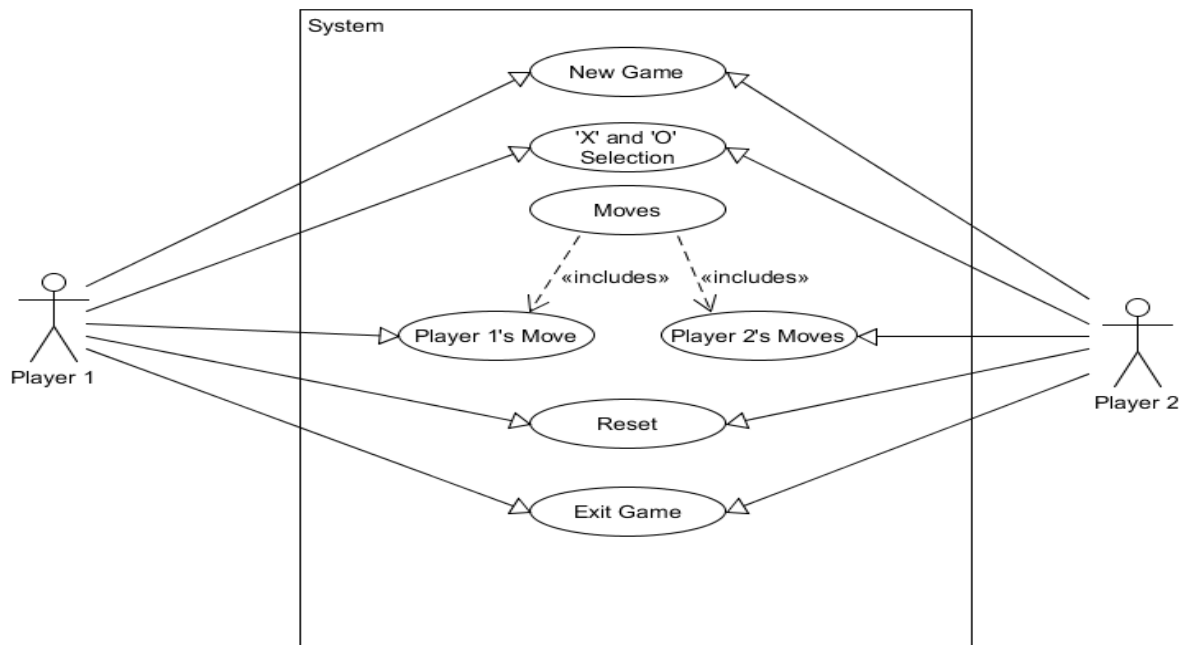
Deliverable 3

1. Implementation: The computer player must attempt to beat the human player using heuristic function.
2. Integrity: The game should maintain its integrity that is preserving all the data and structures in case of failure.
3. Usability: The computer player should start by choosing a position where it may win in the most number of possible ways.
4. Documentation: The code should be documented for better traceability and understanding..
5. Testability: The code should be testable.

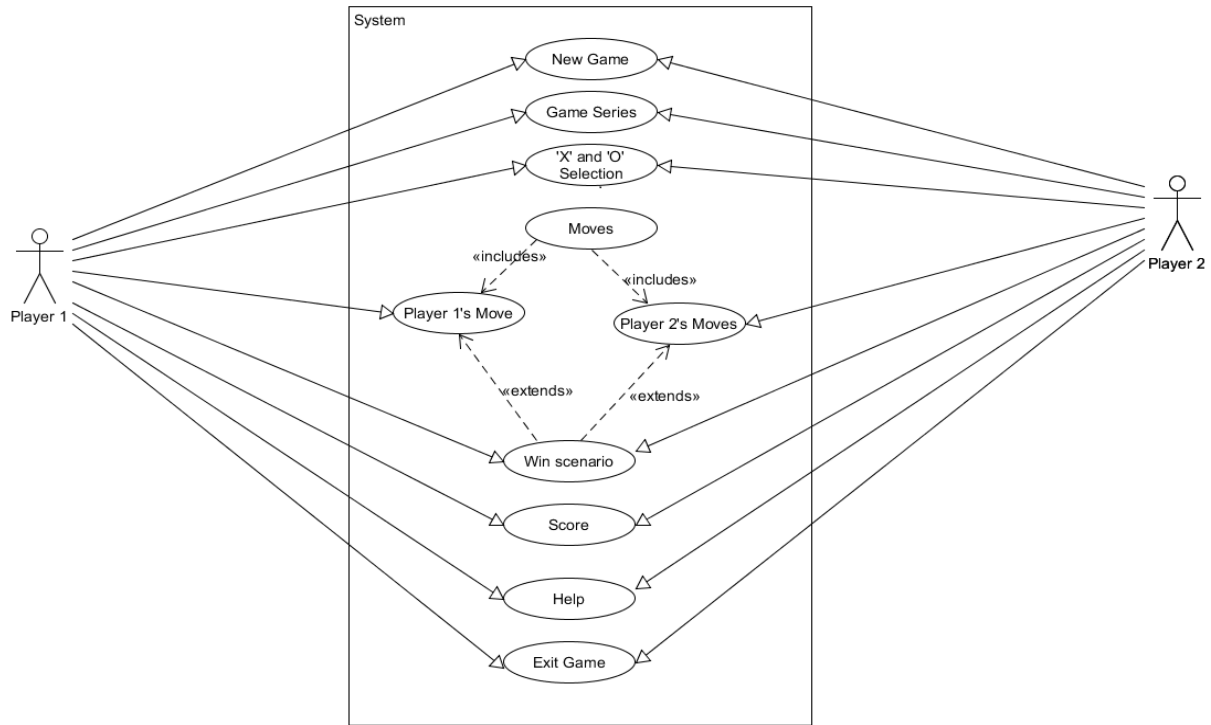
Use Case Analysis

Use Case Diagram

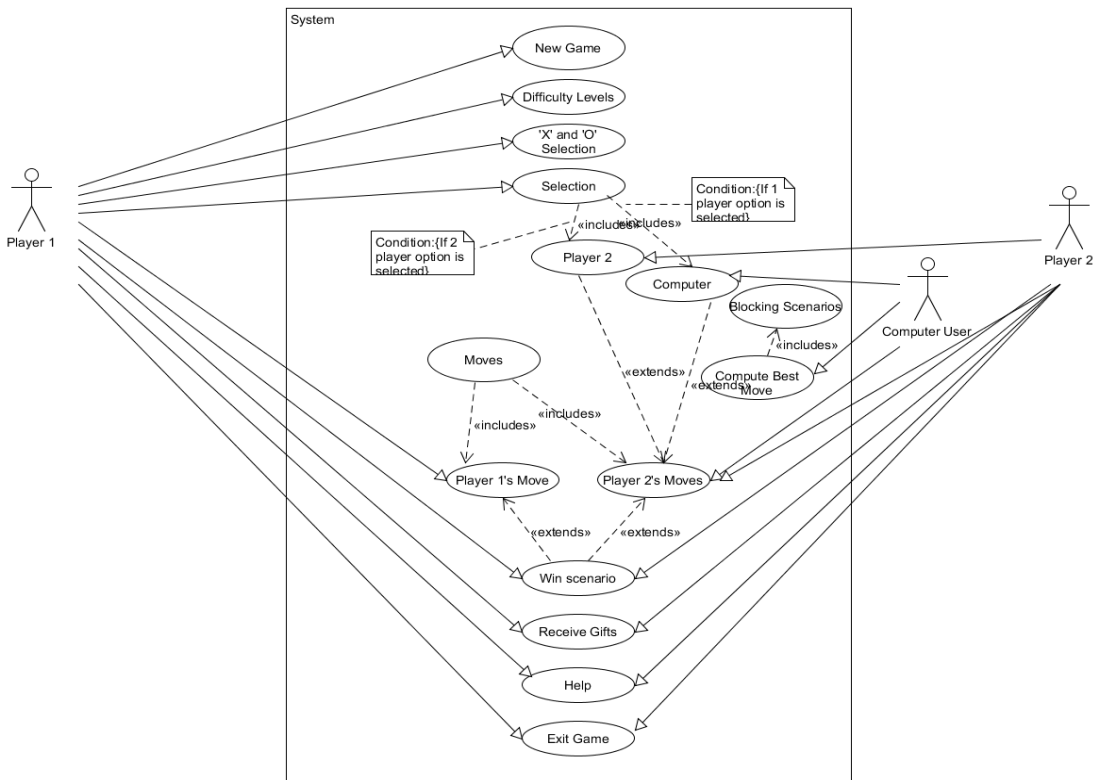
Deliverable 1:



Deliverable 2:



Deliverable 3:



Use Case Scenarios

Deliverable 1:

Scenario 1:

Use Case ID	Use Case 1
Use Case Name	Start Game
Description	This will start a new game for the 2 players with an option of X or O selection.
Primary Actor	Player 1
Secondary Actors	Player 2
Pre-Condition	The application is up and running
Post-Condition	The player will be able to play the game.
Trigger	Click on the Start Game.
Main Flow	<ol style="list-style-type: none"> 1. Player will start the application 2. Player selects either X or O. 3. Player 1 starts playing the game.
Extensions	N/A

Scenario 2:

Use Case ID	Use Case 2
Use Case Name	Reset
Description	The player's will be able to reset the game.
Primary Actor	Player 1 and 2
Secondary Actors	
Pre-Condition	The game is up and running.
Post-Condition	The board will be reset
Trigger	Player clicks the Reset button
Main Flow	<ol style="list-style-type: none"> 1. Player is still playing the game 2. Player select the reset button 3. The games board will be reset
Extensions	

Scenario 3:

Use Case ID	Use Case 3
Use Case Name	Moves
Description	After the game is ready to be played, the players can click on the 3*3 grid. The click can be either X or O depending on player number.
Primary Actor	Player 1/ 2
Secondary Actors	Player 1/ 2
Pre-Condition	1. The application is up and running.

	2. The block on which the player clicks, is not clicked by the opponent.
Post-Condition	Depending on different players, the clicks should show either X or O on the blocks in 3*3 grid.
Trigger	Player's click
Main Flow	<ol style="list-style-type: none"> 1. Player clicks on the block he desires. 2. The click is visible as either X or O depending on the player number
Extensions	<ol style="list-style-type: none"> 1. The player clicks on the block that is already clicked by other player gives no results.

Scenario 4:

Use Case ID	Use Case 4
Use Case Name	Exit Game
Description	Player can exit the game whenever the player wants.
Primary Actor	Player 1 or Player 2
Secondary Actors	
Pre-Condition	The game is active.
Post-Condition	Application is closed and the game is exited.
Trigger	Player click on the Exit Button
Main Flow	<ol style="list-style-type: none"> 1. The player clicks on exit game. 2. The game will be exited.
Extensions	

Deliverable 2:**Scenario 5:**

Use Case ID	Use Case 1
Use Case Name	Start the Game
Description	This will start a new game for the 2 players with an option of X or O selection.
Primary Actor	Player 1
Secondary Actors	Player 2
Pre-Condition	The application is up and running
Post-Condition	The player will be able to play the game.
Trigger	Click on the Start Game.
Main Flow	<ol style="list-style-type: none"> 1. Player will start the application. 2. The players will enter their name. 3. Player selects either X or O. 4. Either of the Player starts playing the

	game.
Extensions	

Scenario 6:

Use Case ID	Use Case 2
Use Case Name	Statistic
Description	The player should be able t
Primary Actor	Player 1 or 2
Secondary Actors	
Pre-Condition	The application is up and running
Post-Condition	The player will be able to see the top players
Trigger	Click on Statistics
Main Flow	<ol style="list-style-type: none"> 1. Player selects the statistics button 2. Application provides with the list of the top player 3. The user can than go back to its game
Extensions	N/A

Scenario 7:

Use Case ID	Use Case 3
Use Case Name	Moves
Description	After the game is ready to be played, the players can click on the 3*3 grid. The click can be either X or O depending on player number.
Primary Actor	Player 1/ 2
Secondary Actors	N/A
Pre-Condition	<ol style="list-style-type: none"> 1. The application is up and running. 2. The block on which the player clicks, is not clicked by the opponent
Post-Condition	<ol style="list-style-type: none"> 1. The game is played between two different players. 2. Either of the player will win.
Trigger	Player's clicks.
Main Flow	<ol style="list-style-type: none"> 1. Player start the application 2. Player 1 clicks on the block he desires. 3. Player 2 clicks on the empty block. 4. Turns go on until all three blocks in vertical or horizontal or cross line are filled up by either player 1 or 2, whosoever is first will be declared as winner.
Extensions	<ol style="list-style-type: none"> 1. The player clicks on the block that is

	already clicked by other player which gives no results.
--	---

Scenario 8:

Use Case ID	Use Case 4
Use Case Name	Show Symbol of Current Player
Description	Application displaying the symbol of the current player who needs to play the move indicating the players' turn.
Primary Actor	Player 1 or Player 2
Secondary Actors	Player 1 or Player 2
Pre-Condition	At least one click (move) has been made.
Post-Condition	Showing the symbol of the next player's turn
Trigger	Not Applicable, Implicit trigger event
Main Flow	<ol style="list-style-type: none"> 1. Player selects their move. 2. Display symbol of the current player to be played.
Extensions	N/A

Deliverable 3:**Scenario 9:**

Use Case ID	Use Case 1
Use Case Name	Start the Game
Description	User will be able to start the application and enter the user name and select the type of the player and play the game
Primary Actor	Player 1
Secondary Actors	Player 2 or computer
Pre-Condition	The application is up and running
Post-Condition	The player will be able to play the game.
Trigger	Player launches the application
Main Flow	<ol style="list-style-type: none"> 1. Player start the application 2. Enter the player names 3. Player selects either X or O. 4. Player select the type of the game play 5. Player starts the game
Extensions	<ol style="list-style-type: none"> a. Player selects the "one player" game 4.1.b. Player will select the difficulty level 4.2.a Player selects the "two player" game

	4.2.b Player 2 enters the name
--	--------------------------------

Scenario 10:

Use Case ID	Use Case 2
Use Case Name	Help menu
Description	Player can see the game rules from the help menu.
Primary Actor	Player 1 or Player 2
Secondary Actors	
Pre-Condition	The application is up and running
Post-Condition	The players are provided with the game rules.
Trigger	Player selects the help menu.
Main Flow	<ol style="list-style-type: none"> 1. Application is running. 2. User clicks the help menu 3. User is provided with the game rules.
Extensions	

Scenario 11:

Use Case ID	Use Case 3
Use Case Name	Moves
Description	After the game is ready to be played, the players can click on the 3*3 grid. The click can be either X or O depending on player's selection.
Primary Actor	Player 1
Secondary Actors	Computer User, Player 2
Pre-Condition	<ol style="list-style-type: none"> 1. The application is up and running. 2. The opponent player is selected. 3. The block on which the player clicks, is not clicked by the opponent
Post-Condition	<ol style="list-style-type: none"> 1. Depending on different players, the clicks should show either X or O on the blocks in 3*3 grid. 2. Either of the player will win.
Trigger	Player's clicks.
Main Flow	<ol style="list-style-type: none"> 1. Player start the application 2. Player clicks on the block he desires. 3. Opponent player clicks on the block other than the one clicked by the previous player. 4. Turns go on until all three blocks in

	vertical or horizontal or cross line are filled up by either player 1 or 2, whosoever is first will be declared as winner.
Extensions	1. The player clicks on the block that is already clicked by other player which gives no results.

Scenario 12:

Use Case ID	Use Case 4
Use Case Name	Receive Gifts
Description	Application grant gifts to the player after each successful win by any player.
Primary Actor	Player 1 or Player 2
Secondary Actors	
Pre-Condition	Player must successfully win the game against the computer or another player
Post-Condition	Computer will prompt the winning player with gift he won
Trigger	One of the player wins the game
Main Flow	<ol style="list-style-type: none"> 1. Game is active 2. Player defeats the another player or the computer 3. The winner will be greeted with the winning message. 4. A random gift will be provided to the winner.
Extensions	

Conclusion

Thus, the Requirements document helps development team to develop robust application system that meets the given requirements by the stakeholder. It helps them understand exactly what features must be included and what should not.

References

- [1] "Tic-tac-toe," Wikipedia, 2016. [Online].
- [2] N. Piccirilli, *Lecture Notes, Software Engineering Process-SOEN 6011*, 2016.