Amin Ortiz
Ellis Levine
Marshall Lang

Use Case Analysis for Project 4

Introduction: The domain is a Tic-Tac-Toe game that we are preparing to learn more about software engineering principles, specifically those of domain analysis and use case analysis. The goal is to implement a simple Tic-Tac-Toe game between two players using a simple Graphical User Interface (GUI) that implements the basic rules of the game.

Glossary: the game is very simple, so no complex terms should be needed.

General knowledge about the domain: The basic rules for a game of tic-tac-toe are as follows:

- The game is played between two players: X and O.
- The game is played on a 3x3 board.
- The players take turns placing a single X or O (depending on whether they are player X or O, respectively).
- X always moves first.
- Whoever gets three X's or O's in a row wins.
- If all nine squares on the board are filled without either player managing to get three in a row, the game ends in a draw.

Customers and users: the only users will be the people who worked on the project (Amin Ortiz, Ellis Levine, Marshall) and the professor and grader.

The environment: the application will run on a personal computer.

Tasks and procedures currently performed: the application currently supports a basic implementation of the tic-tac-toe game for two human (or a human player and a computer player) on a m x n board. Interaction with the system is performed through a simple GUI.

Problem statement: to augment the tic-tac-toe system developed in version 3 with the following functionality:

1. Allow the user to change the look of the UI using themes.
2. Implement a harder version of the basic computer AI (hard AI).

Requirements:

- (Input) Before starting a match, the system will allow the user to choose whether they would like to play with a human player, with a computer opponent of basic skill, or with a hard computer opponent.
- (Input) After choosing what sort of opponent they would like to play against, the system will allow the user to determine the size of the board by entering two positive integers, m and n, that will determine the row size (m) and the column size (n) of the board, as well as allow the user to enter another positive integer (k) that will determine the number of pieces in a row necessary to obtain a victory.
- (Output) When a match starts, the system will display a board with m rows and n columns, as determined by the user's choice of m and n.
- (Output) The first player to get k pieces in a row, horizontally, vertically, or diagonally, will

be declared the winner.

- (Output) The user will be able to change the look and feel of the game by choosing from a set of themes that change the UI.

Use Case Model

Use case: Place a mark on the tic-tac-toe board.

Actors: Player

Goal: to allow one of the basic actions of the game of tic-tac-toe to be played.

Preconditions: a match of tic-tac-toe must have started and it must be the actor's turn.

Summary: When an actor wants to place a mark on the board, a match of tic-tac-toe must be in progress and it must be the actor's turn. They then select one of the empty cells on the board, after which their symbol (either X or O) is placed on the cell and the system moves the turn order forward.

Steps:

Actor Actions	*System Responses*
1. Select a cell on the board during their turn	2. Check if cell is empty
	3. If empty, place symbol on the cell
	4. Check if player has n in a row
	5. Transfer control to next player

Postconditions: the board remembers the actor's selection by placing a mark in the indicated cell. Determine if the move wins.

Use case: Time out an inactive actor.

Actors: player, ai player

Goal: Prevent a game from lasting forever by limiting the amount of time alloted to each actor's turn.

Preconditions: a match of tic-tac-toe must have started and the actor's current turn must have lasted longer than the amount of time alloted to their turn timer.

Summary: If a turn lasts more than the alloted time on the timer, the tic-tac-toe system ends the game and declares the winner to be the player whose turn did not time out.

Steps:

Actor Actions	*System Responses*
1. During their turn, does not select a valid move within the alloted time limit	2. Declares the player whose turn did not time out the winner and ends the match.

Postconditions: the match of tic-tac-toe ends and one player is declared the winner.

Use case: start a match

Actors: player

Goal: to start a match of tic-tac-toe

Preconditions: system must not be in the middle of a tic-tac-toe match.

Summary: when the user wants to start a match of tic-tac-toe, they select the kind of opponent that they would like to play against (human, basic computer, or hard computer). The system then queries the user for the size of the board (m, n) and the number of pieces in a row to determine a victory

Steps:

Actor Actions	*System Responses*
1. Choose opponent	2. Display 'board options' dialog
3. Specify board options (m, n, k)	
4. Confirm selection	5. Remove 'board options' dialog
	6. Initialize and display custom board

Postconditions: the game board is initialized according to the user's preferences and displayed to the user

Use case: start a match with invalid values

Related use cases:

Extension of:

start a match (extension point: step 3: Specify board options)

Steps:

1. Choose opponent	2. Display 'board options' dialog
3. Specify board options (m, n, k)	
4. Confirm Selection	5a. Indicate that values are invalid
5b. Specify valid board options	
5c. Confirm selection	6. Remove 'board options' dialog.
	7. Initialize and display custom board.

Use case: restart the board

Actors: player

Goal: allow another match of tic-tac-toe to begin

Preconditions: either a match must be in progress or have ended.

Summary: this option allows the user to reset the board at any point after a game has begun or ended.

Steps:

Actor Actions	*System Responses*
1. Selects the reset option	2. Discard the current board
	3. Render a new board with the same options as before
	4. Move to player one's turn

Use case: set theme

Actors: player

Goal: change the look of the game's UI

Preconditions: player must be in the menu screen

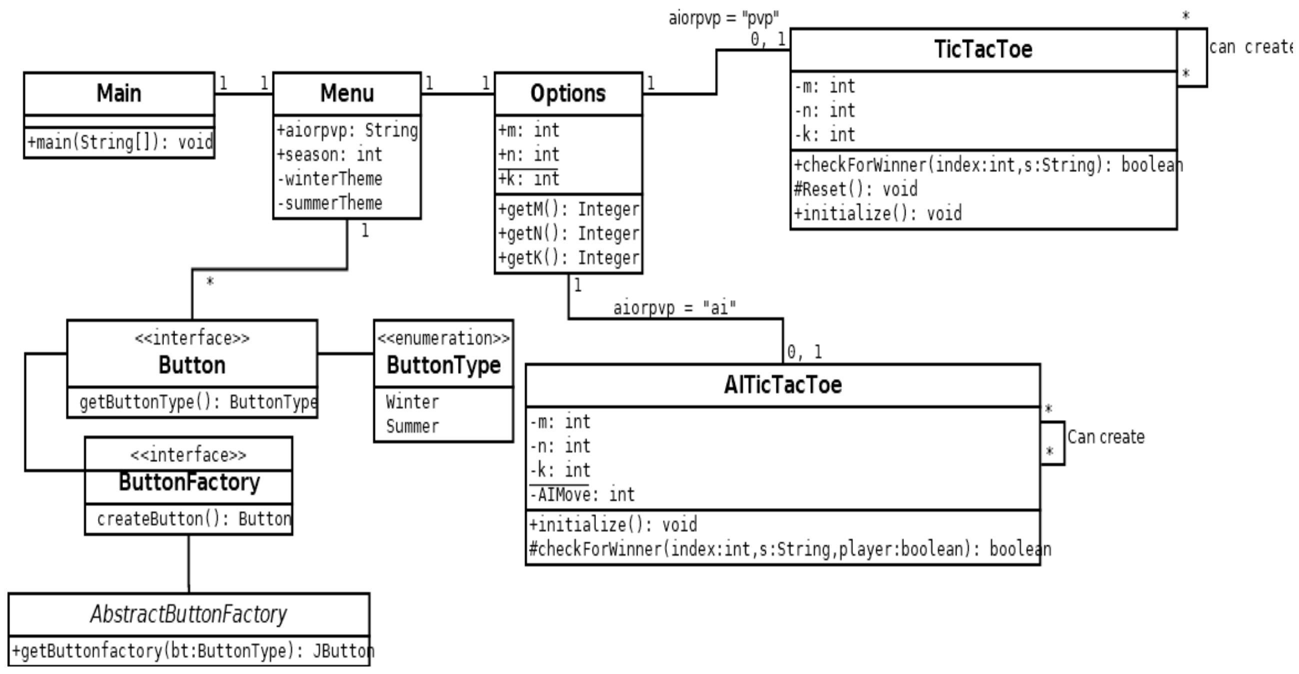
Summary: allows the player to change the look and feel of the game by setting different game themes that change the look and color of the board, buttons, and other graphics elements.

Steps:

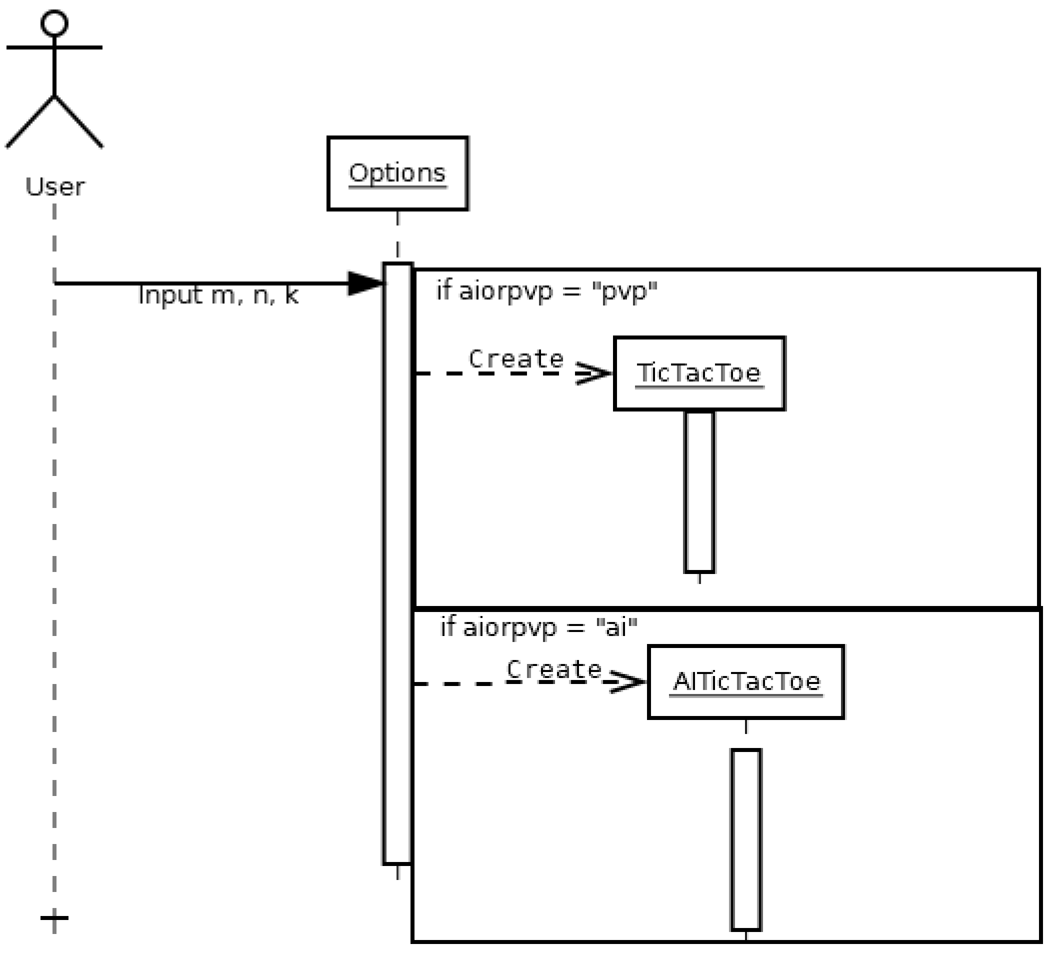
Actor Actions	*System Respones*
1. Select theme option	2.Render the 'theme options' dialog|
3. Select desired theme option	
4. Confirm selection	5. Render UI elements using selected theme.

Postconditions: the UI theme is changed to the user's selection. Until the theme is changed again, the system will render the UI elements using the selected theme.

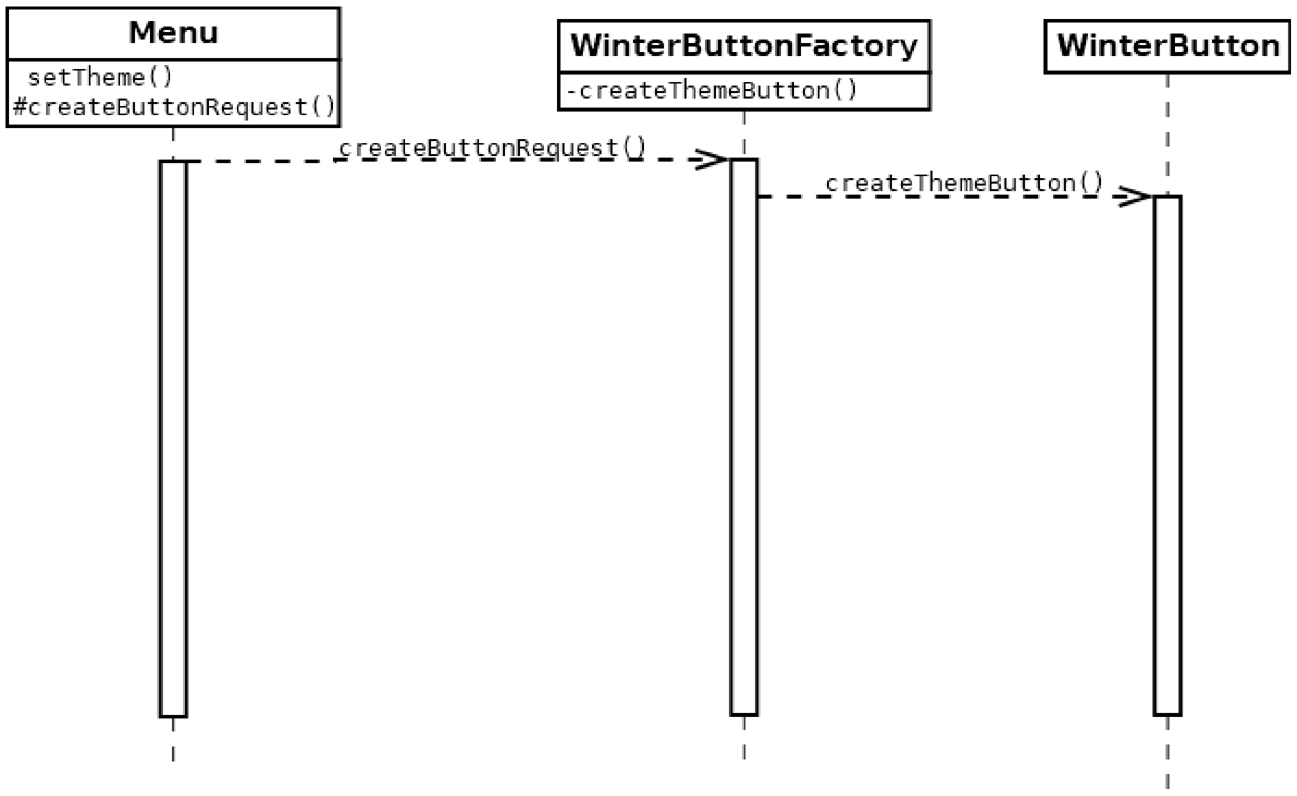
Updated Class Diagram



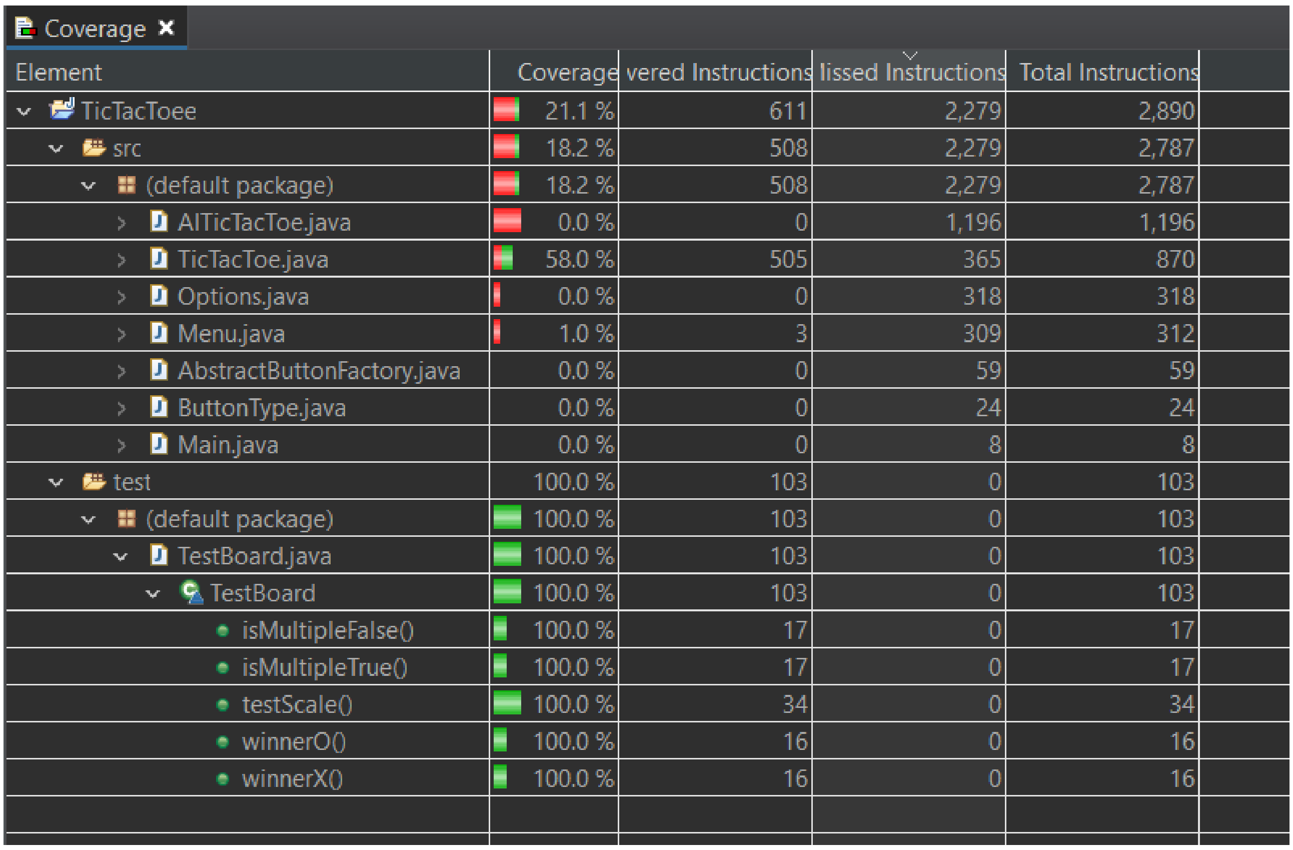
Board Creation Sequence Diagram



Create Winter Button Sequence Diagram



Unit Test Coverage Report



Coverage

Element	Coverage	vered Instructions	lissed Instructions	Total Instructions
TicTacToee	21.1 %	611	2,279	2,890
src	18.2 %	508	2,279	2,787
(default package)	18.2 %	508	2,279	2,787
AITicTacToe.java	0.0 %	0	1,196	1,196
TicTacToe.java	58.0 %	505	365	870
Options.java	0.0 %	0	318	318
Menu.java	1.0 %	3	309	312
AbstractButtonFactory.java	0.0 %	0	59	59
ButtonType.java	0.0 %	0	24	24
Main.java	0.0 %	0	8	8
test	100.0 %	103	0	103
(default package)	100.0 %	103	0	103
TestBoard.java	100.0 %	103	0	103
TestBoard	100.0 %	103	0	103
isMultipleFalse()	100.0 %	17	0	17
isMultipleTrue()	100.0 %	17	0	17
testScale()	100.0 %	34	0	34
winnerO()	100.0 %	16	0	16
winnerX()	100.0 %	16	0	16

Tic Tac Toe PA4 End-to-End Test Script

Test Case 1

Purpose:

1. Verify that the proper game mode and theme can be selected

Requirement Traceability:

1. Before starting a match, the system will allow the user to chose whether they would like to play with a human player or with a computer opponent of advanced skill.

Setup:

Obtain the most current version of TicTacToe- the PA4 Submission, and follow the instructions below:

Test Data:

Action	**Input**	**Expected Output**
Launch the application		The main menu with options appears. There are options labelled “pvp” and “ai”, and “winter” and “summer”
Verify PvP options may be selected	Click “PvP”	Another option screen appears, called “Game Options”, with text fields to enter m, n, and k
Verify AI options may be selected	Click “ai”	Another option screen appears, called “Game Options”, with text fields to enter m, n, and k
Verify the color themes work	Click “winter” and “summer” alternately	The buttons and game board change to suit the theme

Test Case 2

Purpose:
Verify that the game options menu for selecting m,n, and k accepts user input properly

Requirement Traceability:
2. Before starting a match, the system will allow the user to chose whether they would like to play with a human player or with a computer opponent of advanced skill.

Setup:
Obtain the most current version of TicTacToe- the PA4 Submission, and follow the instructions below:

Test Data:

Action	**Input**	**Expected Output**
Launch the application		The main menu with options appears. There are options labelled “pvp” and “ai”, and “winter” and “summer”
Select “ai” or “pvp”	Click the “ai” or “pvp” button	Verify that positive whole numbers may be entered in for m,n, and k
Check submit button	Click “submit”	The options disappear and a tic-tac-toe board appears

Test Case 3

Purpose:
Check that the game options affect the board properly

Requirement Traceability:
3. When a match starts, the system will display a board with m rows and n columns,
as determined by the user's choice of m and n.

Setup:
Obtain the most current version of TicTacToe- the PA4 Submission, and follow the instructions below:

Test Data:

Action	Input	Expected Output
Launch the application, and select game mode	Select “ai” or “pvp” from the main menu	The game options appear
Select whole numbers for m, n, and k	Choose a whole number for each variable and type it in to the text box	The text boxes accept inputs and display them
Submit options and create board	Click “submit”	A board appears of width m, height n, and in the appropriate theme with a chosen ai or human opponent

Test Case 4

Purpose:

Check that the game options affect the board properly and that TicTacToe plays as intended

Requirement Traceability:

4. The first player to get k pieces in a row, horizontally, vertically, or diagonally, will be declared the winner.

Setup:

Obtain the most current version of TicTacToe- the PA4 Submission, and follow the instructions below:

Test Data:

Action	Input	Expected Output
Launch the application, and select game mode and game options	Select “ai” or “pvp” from the main menu (optionally a theme), then choose dimensions for the board and a value to play to	A tictactoe board appears with all the chosen settings

Being playing tictactoe	Click a square each time it is your turn	A symbol corresponding to your team (O or X) fills the square
Get k-in-a-row and win	Click squares strategically with the goal of reaching k in a row before your opponent does	Upon reaching k in a row, the board greys out and the winner is announced