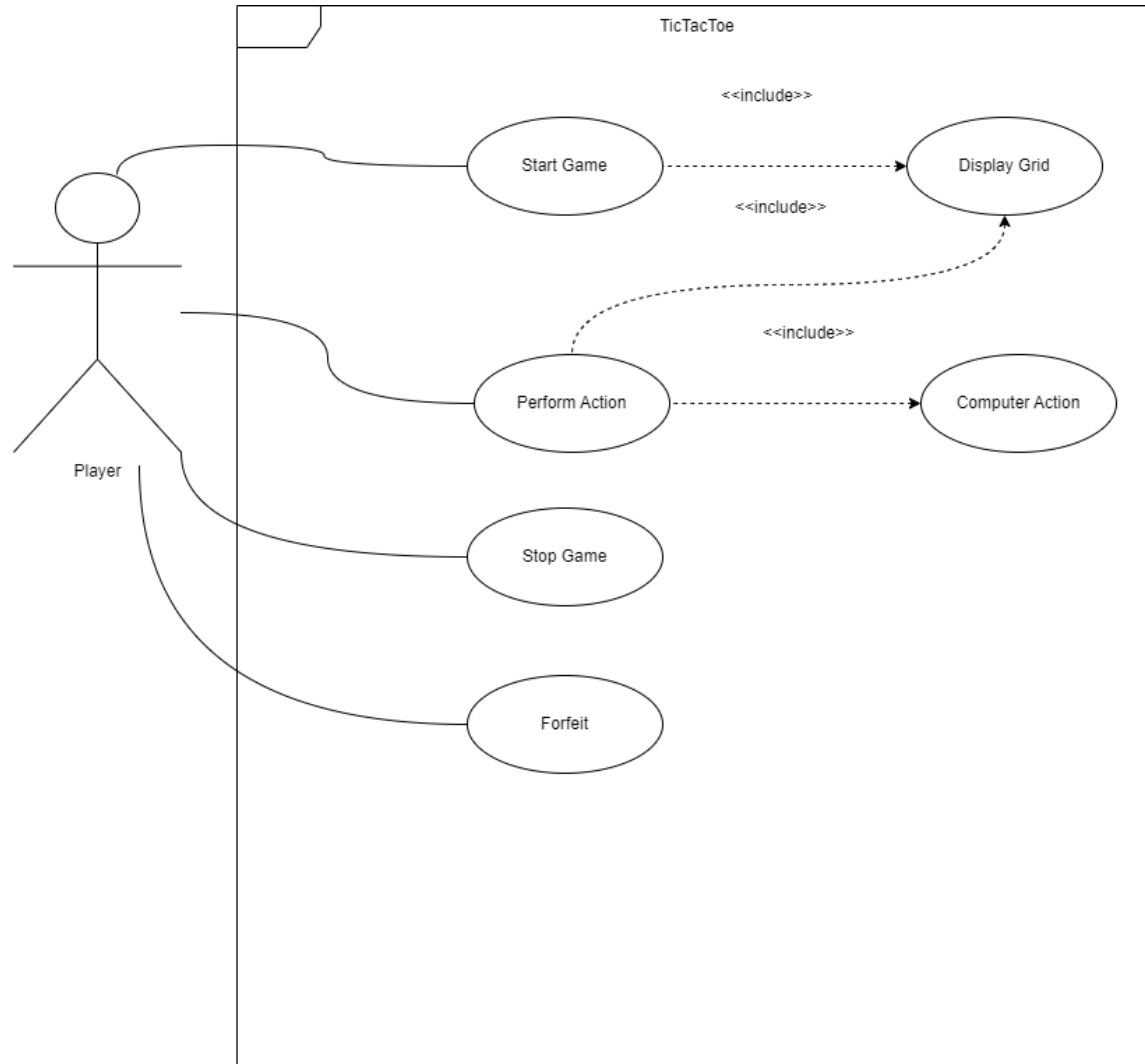


A background graphic featuring several stylized human figures (represented by grey spheres for heads and grey cylinders for bodies) connected by thin grey lines, suggesting a network or group structure. The figures are arranged in a way that some are in the foreground and others are in the background, creating a sense of depth.

Group Project – Part 3

LOGAN MILBRANDT, VIOREL ORTIZ, JOSÉ RIQUELME



Use Case Diagram

Actor Actions

System Responses

1. The player selects one of the two themes available.	2. Record the theme selected by the player.
3. The player specifies the number of minutes allotted to timer.	4. Record the minutes entered by the player.
5. The player specifies the number of seconds allotted to timer.	6. Record the seconds entered by the player.
7. The player specifies the number of columns of the game grid to be displayed.	8. Record the column number entered by the player.
7. The player specifies the number of rows of the game grid to be displayed.	9. Record the rows number entered by the player.
10. The player specifies the number of matches that lead to a win.	11. Record the number of symbol matches that lead to a win.
12. The player checks checkbox for a two-player game, or the game defaults to single player with a computer game.	13. Record the option entered by user.
14. The player specifies whether X, or O will be the first to make a move.	15. Record the option entered by user.
16. The player presses the start button.	17. A. Start the timer based on the configuration set by user.
	17. B. Display the game grid with specifications set by the user.

Use Case: Start Game

- **Main Flow**

Use Case: Perform Action

- Main Flow

Actor Actions	System Responses
1. The player presses one of the empty grids.	2a. The system records either "X" or "O" depending on the turn.
	2b. The system checks the Game Status to determine if a winning condition is met.
	2c. The system stops counting down the timer associated with symbol pressed.

- Alternative Flow

Actor Actions	System Responses
	1a. The system records either "X" or "O" depending on the turn.
	1b. The system checks the Game Status to determine if a winning condition is met.
	1c. The system stops counting down the timer associated with symbol pressed.

Use Case: Forfeit

- **Main Flow**

Actor Actions	System Responses
1. The player presses the forfeit button.	2.b. The system displays the game over dialog.

Use Case: Stop Game

- **Main Flow**

Actor Actions	System Responses
1. The player presses the "Back to Menu" option	2.a. The system resets the game status.
	2. b. The system displays the start menu.

- **Alternative Flow**

Actor Actions	System Responses
1. The player presses the "Close" option.	2. The system closes the game.

Use Case: Computer Action (inclusion)

- **Main Flow**

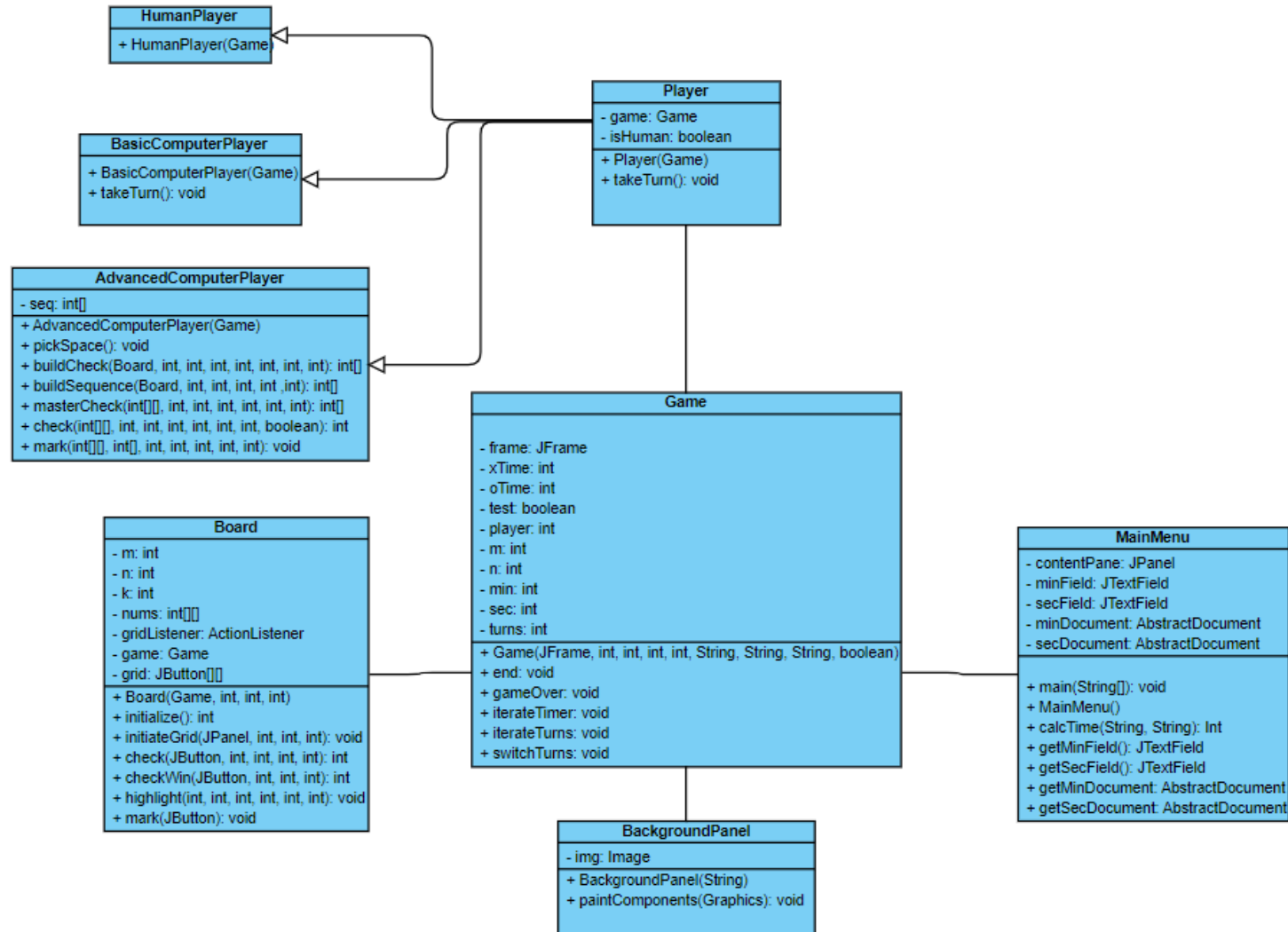
Actor Actions	System Responses
1. If the player has chosen single player game.	2. The system randomly generates places a mark on the game grid.

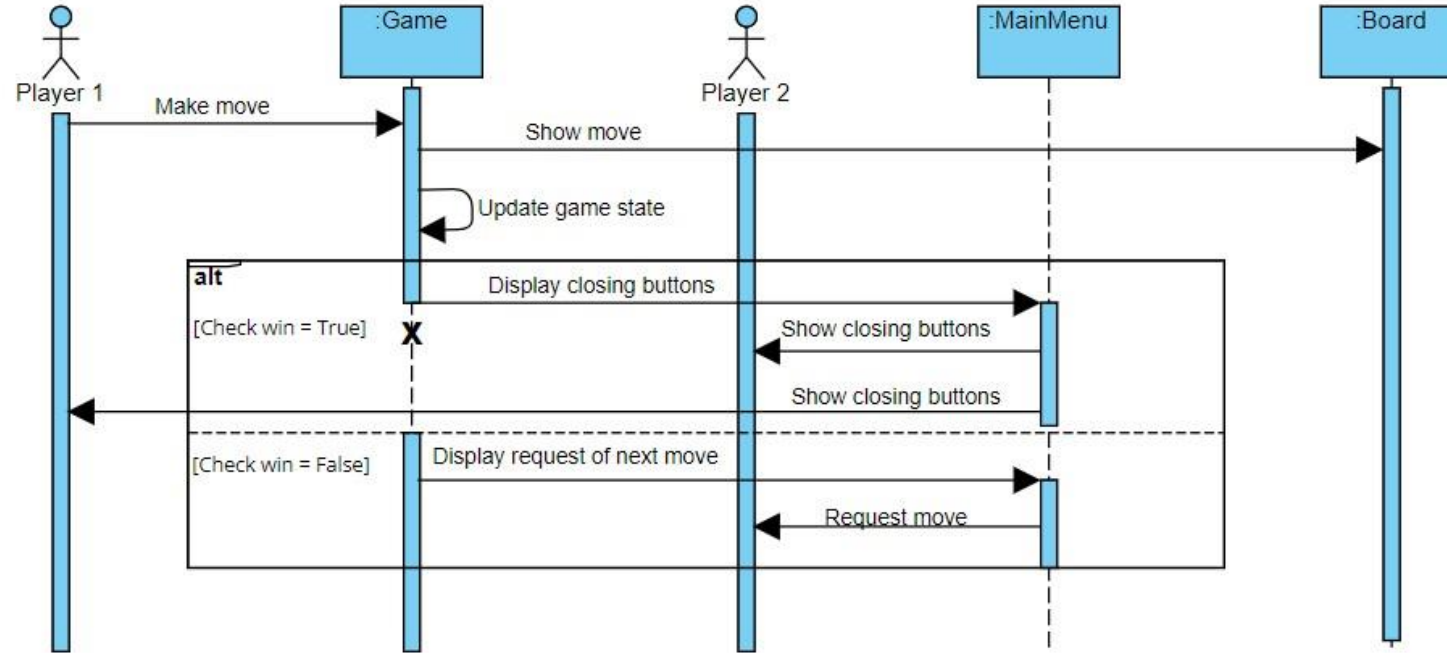
Actor Actions	System Responses
1. The user presses "Start Game".	1.a. The system generates a grid layout based on grid specifications entered by user.
	1.b. The system displays two countdown timers one for each player based on the specifications set by user.
	1.c. The system displays the turn of first player.
	1.d. The system displays the "Forfeit Button".

Use Case: Display Grid (inclusion)

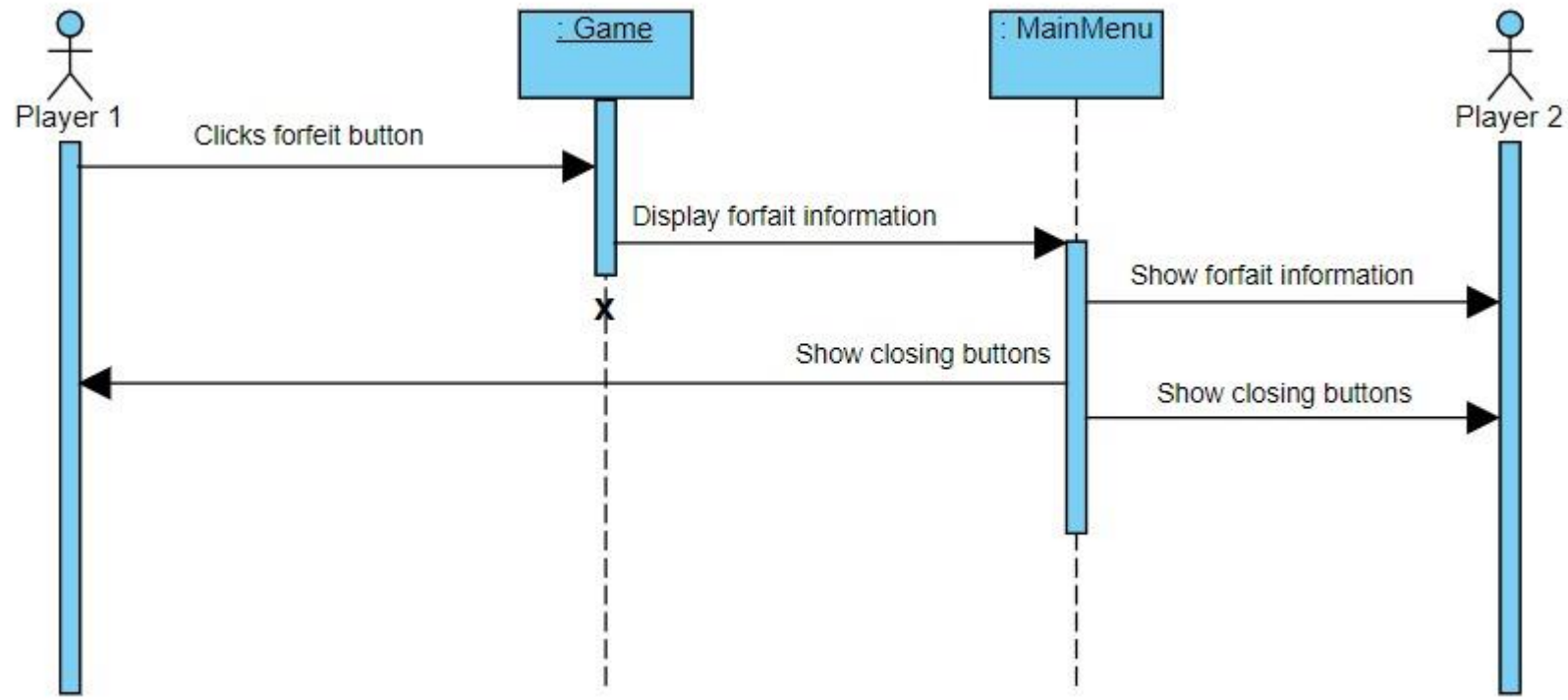
- Main Flow

UML Diagram





Sequence Diagram 1 – Player 1 movement



Sequence Diagram 2 – Player 1 forfeit


Commits log

Commits

 main ▾

 Commits on Dec 10, 2023

Add files via upload

 bluecrest22 committed 22 minutes ago

Images for themes

 LightningWizard committed 1 hour ago

Merge branch 'main' of <https://github.com/LightningWizard/CEN4010-Tea...> ...

 LightningWizard committed 1 hour ago

Restructures a good deal of the program, adds themes, and a WIP advanced ...

 LightningWizard committed 1 hour ago

Add files via upload

 bluecrest22 committed 2 hours ago

Test Case 1

Purpose:

1. Verify that the game correctly processes a Victory as an X Player as a 2 Player game.

Requirement Traceability:

Requirement number appears here.

Setup:

Open the Eclipse Application.

Create a new Java Project for Eclipse.

Create a new Java Class under Project named Game, copy and paste provided code for class.

Create a new Java Class under Project named Main Menu, and paste provide code for class.

Run Main Menu class.

Follow the directions below.

Test Data:

Action	Input	Expected Output
Launch the application		The application main GUI appears.
Select the minute box	Click the box under the minutes label. Enter 5 as a keyboard input.	The minutes box now reflects the user option.
Select the 2 Player Option to enable it.	Click the 2Players? Checkbox.	The checkbox next to the 2 Players? is now checked.
Start the game.	Click the START Button.	The application now displays the game GUI.
Mark X on the Game Grid.	Click on the leftmost uppermost square on the grid.	The game GUI now displays the X mark on the leftmost uppermost square on the game Grid.
Mark O on the Game Grid.	Click on the square to the right of the	The game GUI now displays the O mark to the right of the previously placed mark along with the previously marked squares.

Mark X on the Game Grid.	previously marked square.	The game GUI now displays the X mark below the previously marked X along with the previously marked squares.
Mark O on the Game Grid.	Click on the square below the previously marked X.	The game GUI now displays the O mark below the previously marked O along with the previously marked squares.
Mark X on the Game Grid.	Click on the square below the previously marked O.	The GUI now displays a pop denoting that Player X won, with two options Back to Game, and Close along with highlighting in black the sequence of marks that won the game.
Close the Game.	Click on the square below the previously marked X.	The GUI should now disappear as a result of pressing the Close Button.
	Click on the Close Button.	

Notes:

We don't check the Columns, Rows, or Win condition, we utilize the default.

Test Case 2

Purpose:

1. Verify that the game correctly processes a Victory as an O Player as a 2 Player game.

Requirement Traceability:

Requirement number appears here.

Setup:

Open the Eclipse Application.

Create a new Java Project for Eclipse.

Create a new Java Class under Project named Game, copy and paste provided code for class.

Create a new Java Class under Project named Main Menu, and paste provide code for class.

Run Main Menu class.

Follow the directions below.

Test Data:

Action	Input	Expected Output
Launch the application		The application main GUI appears.
Select the minute box	Click the box under the minutes label. Enter 5 as a keyboard input.	The minutes box now reflects the user option.
Select the 2 Player Option to enable it.	Click the 2Players? Checkbox.	The checkbox next to the 2 Players? is now checked.
Change the 1 st Player Mark to O.	Click the drop box and navigate to the O mark and click the mark.	The drop box should now display the mark O next to the label 1 st Player on the GUI.
Start the game.	Click the START Button.	The application now displays the game GUI.
Mark O on the Game Grid.		The game GUI now displays the O mark on the leftmost

Mark X on the Game Grid.	Click on the leftmost uppermost square on the grid.	uppermost square on the game Grid. The game GUI now displays the X mark to the right of the previously placed mark along with the previously marked squares.
Mark O on the Game Grid.	Click on the square to the right of the previously marked square.	The game GUI now displays the O mark below the previously marked O along with the previously marked squares.
Mark X on the Game Grid.	Click on the square below the previously marked O.	The game GUI now displays the X mark below the previously marked X along with the previously marked squares.
Mark O on the Game Grid.	Click on the square below the previously marked X.	The GUI now displays a pop denoting that Player O won, with two options Back to Game, and Close along with highlighting in black the sequence of O marks that won the game.
Close the Game.	Click on the square below the previously marked O.	The GUI should now disappear because of pressing the Close Button.
	Click on the Close Button.	

Notes:

We don't check the Columns, Rows, or Win condition, we utilize the default.

Test Case 3

Purpose:

1. Verify that the Back to Menu option functions returns you back to the Main Menu.

Requirement Traceability:

Requirement number appears here.

Setup:

Open the Eclipse Application.

Create a new Java Project for Eclipse.

Create a new Java Class under Project named Game, copy and paste provided code for class.

Create a new Java Class under Project named Main Menu, and paste provide code for class.

Run Main Menu class.

Follow the directions below.

Test Data:

Action	Input	Expected Output
Launch the application		The application main GUI appears.
Select the minute box	Click the box under the minutes label. Enter 5 as a keyboard input.	The minutes box now reflects the user option.
Select the 2 Player Option to enable it.	Click the 2Players? Checkbox.	The checkbox next to the 2 Players? is now checked.
Change the 1 st Player Mark to O.	Click the drop box and navigate to the O mark and click the mark.	The drop box should now display the mark O next to the label 1 st Player on the GUI.
Start the game.	Click the START Button.	The application now displays the game GUI.
Mark O on the Game Grid.		The game GUI now displays the O mark on the leftmost

Mark X on the Game Grid.	Click on the leftmost uppermost square on the grid.	uppermost square on the game Grid. The game GUI now displays the X mark to the right of the previously placed mark along with the previously marked squares.
Mark O on the Game Grid.	Click on the square to the right of the previously marked square.	The game GUI now displays the O mark below the previously marked O along with the previously marked squares.
Mark X on the Game Grid.	Click on the square below the previously marked O.	The game GUI now displays the X mark below the previously marked X along with the previously marked squares.
Mark O on the Game Grid.	Click on the square below the previously marked X.	The GUI now displays a pop denoting that Player O won, with two options Back to Menu, and Close along with highlighting in black the sequence of O marks that won the game.
Return to Main Menu.	Click on the square below the previously marked O.	The GUI should now display the main GUI with the options previously set before.
Close the Game.	Click on the Back to Menu option.	The GUI should now disappear.
	Click the close x button at the top right of the UI.	

Notes:

We don't check the Columns, Rows, or Win condition, we utilize the default.

Test Case 4

Purpose:

1. Verifying the timer counting down leads to a Game Over prompt.

Requirement Traceability:

Requirement number appears here.

Setup:

Open the Eclipse Application.

Create a new Java Project for Eclipse.

Create a new Java Class under Project named Game, copy and paste provided code for class.

Create a new Java Class under Project named Main Menu, and paste provide code for class.

Run Main Menu class.

Follow the directions below.

Test Data:

Action	Input	Expected Output
Launch the application.		The application main GUI appears.
Start the game.	Click the START button.	The Game Grid GUI should now display.
Let the game timer run out.		A Game Over prompt should now appear, with the options of Back to Menu, and Close.
Close the game.	Click the Close button.	The game GUI should now disappear.
.	.	.
.	.	.
.	.	.

Notes:

The Minute, Seconds, Columns, Rows, Win, 2 Players? 1st Player options are left as defaults.

Test Case 5

Purpose:

1. Verifying the Forfeit Option works as intended.

Requirement Traceability:

Requirement number appears here.

Setup:

Open the Eclipse Application.

Create a new Java Project for Eclipse.

Create a new Java Class under Project named Game, copy and paste provided code for class.

Create a new Java Class under Project named Main Menu, and paste provide code for class.

Run Main Menu class.

Follow the directions below.

Test Data:

Action	Input	Expected Output
Launch the application.		The application main GUI appears.
Set the Minutes Option to 5.	Click the textbox under the MINUTES label and enter the number 5 on the keyboard.	The textbox under the MINUTES should now display the inputted option.
Start the game.	Click the START button on the game GUI.	The application should now display the game GUI.
Forfeit the game.	Click the FORFEIT button on the game GUI.	The GUI should now display a popup with FORFEIT indicating Player O wins because the option was clicked during Player X's turn.
Close the game.	Click the Close button on the popup.	The popup box should also display Close, or Back to Game options.
.		The game should now close.
.		.

Notes:

The Seconds, Columns, Rows, Win, 2 Players? 1st Player options are left as defaults.

Test Case 6

Purpose:

1. Verify that the Player VS AI (Basic) is working as intended.

Requirement Traceability:

Requirement number appears here.

Setup:

Open the Eclipse Application.

Create a new Java Project for Eclipse.

Create a new Java Class under Project named Game, copy and paste provided code for class.

Create a new Java Class under Project named Main Menu, and paste provide code for class.

Run Main Menu class.

Follow the directions below.

Test Data:

Action	Input	Expected Output
Launch the application		The application main GUI appears.
Set the Minutes Option to 5.	Click the textbox under the MINUTES label and enter then umber 5 on the keyboard.	The textbox under the MINUTES should now display the input option.
Set the Player Option to Player VS AI (Basic)	Select the dropdown menu and click the Player VS AI(Basic) option.	The Main Menu UI should now display your selected option.
Start the game.	Click the START button on the game GUI.	The application should now display the game GUI.
Place an X mark on the Game Grid.	Click a square on the Game Grid.	The GUI should now display a mark placed upon the square click upon. Sequentially, the Computer will place an O mark on the Game Grid as well.
Repeat placing marks on the Game Grid till the game is finished.	Continue clicking marks on the Game Grid.	After placing marks, the game should either declare a victor, or a draw. Display a popup box with the options Close, and Back to Game.
Close the Game.		

Click the Close button on
the popup box.

The game should now close.

Notes:

The Seconds, Columns, Rows, Win, Go First, Theme Options are left as defaults.

Test Case 7

Purpose:

1. Verify that the Player VS AI (Advanced) is working as intended.

Requirement Traceability:

Requirement number appears here.

Setup:

Open the Eclipse Application.

Create a new Java Project for Eclipse.

Create a new Java Class under Project named Game, copy and paste provided code for class.

Create a new Java Class under Project named Main Menu, and paste provide code for class.

Run Main Menu class.

Follow the directions below.

Test Data:

Action	Input	Expected Output
Launch the application		The application main GUI appears.
Set the Minutes Option to 5.	Click the textbox under the MINUTES label and enter then umber 5 on the keyboard.	The textbox under the MINUTES should now display the input option.
Set the Player Option to Player VS AI (Advanced)	Select the dropdown menu and click the Player VS AI(Advanced) option.	The Main Menu UI should now display your selected option.
Start the game.	Click the START button on the game GUI.	The application should now display the game GUI.
Place an X mark on the Game Grid.	Click a square on the Game Grid.	The GUI should now display a mark placed upon the square click upon. Sequentially, the Computer will place an O mark on the Game Grid as well.
Repeat placing marks on the Game Grid till the game is finished.	Continue clicking marks on the Game Grid.	After placing marks, the game should either declare a victor, or a draw. Display a popup box with the options Close, and Back to Game.
Close the Game.		

Click the Close button on the
popup box.

The game should now close.

Notes:

The Seconds, Columns, Rows, Win, Go First, Theme Options are left as defaults.

Test Case 8

Purpose:

1. Verify that the Summer Theme is working as intended.

Requirement Traceability:

Requirement number appears here.

Setup:

Open the Eclipse Application.

Create a new Java Project for Eclipse.

Create a new Java Class under Project named Game, copy and paste provided code for class.

Create a new Java Class under Project named Main Menu, and paste provide code for class.

Run Main Menu class.

Follow the directions below.

Test Data:

Action	Input	Expected Output
Launch the application.		The application main GUI appears.
Set the Minutes Option to 5.	Click the textbox under the MINUTES label and enter the number 5 on the keyboard.	The textbox under the MINUTES should now display the input option.
Start the game.		The application should now display the game GUI with the Summer Theme.

Notes:

The Seconds, Columns, Rows, Win, Go First Options are left as defaults.

Test Case 9

Purpose:

1. Verify that the Winter Theme is working as intended.

Requirement Traceability:

Requirement number appears here.

Setup:

Open the Eclipse Application.

Create a new Java Project for Eclipse.

Create a new Java Class under Project named Game, copy and paste provided code for class.

Create a new Java Class under Project named Main Menu, and paste provide code for class.

Run Main Menu class.

Follow the directions below.

Test Data:

Action	Input	Expected Output
Launch the application.		The application main GUI appears.
Set the Minutes Option to 5.	Click the textbox under the MINUTES label and enter the number 5 on the keyboard.	The textbox under the MINUTES should now display the input option.
Set the Game's theme to Winter.	Select the THEME dropdown menu, and hover over the Winter Theme.	The selection will be displayed in the dropdown menu.
Start the game.		The application should now display the game GUI with the Summer Theme.

Notes:

The Seconds, Columns, Rows, Win, Go First Options are left as defaults.