Table of Contents:

1. Executive Statement
2. Game Rules
3. Overview of the UI
4. Actors
5. System Diagram
6. Actions and Sub-actions
7. Primary Scenarios
8. Secondary Scenarios
9. The Team
10. Closing

**Executive Statement (WIP)**

The purpose of this project is to create a multiplayer board game in Java. The board game will require players to take turns moving and attack with a team of three robots. Once there are only one player’s robots remaining, they are declared the winner. The system must support one or two human players, as well as up to six AI-controlled players. The board itself is hex-based and must have functionality for choosing the number of players. As well, the user must also be able to select a board size of five or seven based on the number of players. If there are no human players in game, user must be able to spectate the AI’s gameplay.

**The user-interface**

The user-interface will be split into two parts, the menu and in-game. The menu must have functionality for starting a game with the user-chosen player number and board size. The menu must also have functionality for downloading and updating robots, as well as configuring robot teams. On top of this, the menu must also contain an option for exiting the application.

In-game, the UI must have an indicator of the current player’s turn. As well, it must have indicators for the current player’s team of robots and their individual statuses (health, range, movement). In-game, there must also be a button for attacking and moving. It should have an indicator telling the user they are playing, waiting, or spectating.

**Actors and their scenarios**

We have narrowed down the actors for our system to four: the players, AIs, spectators, and the robot librarian. We have identified thirteen primary scenarios and XXXXXXXX SECONDARY SCENARIOS FILL THIS IN LATER BASED ON THE REST OF THE GROUP

**Hardware/Software**

The system will be programmed entirely in Java, making use of the AspectJ extension. As per assignment requirements, the system must compile and run on the U of S’ *tuxworld*. The system will make use of the JSON file format for downloading and updating robots.

We are team C4, made up of Yu, Ixabat, Daniel, Kevin, and Brandon. Each member has their own unique skillset that makes us an overall well-rounded team. Decisions are decided in a democratic fashion with emphasis on member-input. Communication is frequent and meetings are organized multiple days a week to ensure members are up-to-date on current objectives. Our team is making use of Git as our form of version-control.

This requirement document marks the completion of the first phase in our software-engineering process, and is the first of four deliverables our team will be submitting.

­­ **Visual Requirements**

The visual requirements are composed of two main screens and one selection box on the menu screen. The screens are:

1. Menu Screen
2. Play and Options Selection Screen
3. In-game Screen

* Menu Screen

When the user launches the program the menu screen is presented. The menu screen is comprised of three options for the user to select at the top of the screen. A larger selection screen is below the options boxes where the user will select the game options when either “Play” or “Options” are pressed.

1. Play

The “Play” selection box will bring the user to the larger selection box. Inside this box the user has the options to select the number of players in the game, whether the player is an AI or a Human, the board size (between 5 and 7), the colour of the player’s team and the name of the player. Once the user has selected the options there is a “Begin” button at the bottom of the selection box where the user may start the game. Clicking the “Begin” button will bring the user to the in-game screen.

1. Options

The “Options” selection box will bring the user to the larger selection box. Inside this box the user has the options to change the sound and music volume of the game. Also inside the box is the Robot Library where the user can import and delete the team of robots they would like to use in-game. There are three buttons provided for the user and a (originally empty) list. The buttons are “Import from Server”, “Import from Local” and “Delete”. Clicking on the “Import from Server” button brings the user to the server page where they can select a robot team. Clicking on the “Import from Local” button will bring the user to their file directory where they can choose a robot team saved locally. Lastly the user has the option to click on already imported robot team in the scroll list and press the delete button to remove that team from the list.

1. Exit

The final selection box is the “Exit” box which closes the program.

* Play and Options Selection Screen

1. Play Selection Box Layout

At the top of the box will be a list with designated lines for the user to input the team names.

Directly to the left of the team’s name will be a small circle to designate the team colour of that team. To the right of the players name will be the option to declare the player’s team either a “Human” or “AI”. Below the list will be two images of a size five hexagonal board and a size seven hexagonal board. The boards will be highlighted based on the player size that the user chooses. Two players will highlight the size five board. Three players will highlight both boards and six players will highlight the size seven board. Below the two boards will be the “Begin” button to start the game.

1. Options Selection Box Layout

At the top of the box will be two slider boxes for determining the volume level of the sound and music. Below these two sliders will be the Robot Library list. To the top-right of the Robot Library are three selection boxes. “Import from Server”, “Import from Local” and “Delete”.

* In-Game Screen

The in-game screen is composed of many different UI additions as well as the playing field they are listed here:

1. Hexagonal Playing Field

This is the main playing field located in the centre of the screen. If the user decided on a AI only game the entire field is visible to the user and they are able to see the decisions made by the AI in spectator mode. If the user is playing in the game though they can only see the part of the field that their pieces allow them to see the rest is covered in a fog of war. If a player dies then they go to spectator mode and can see the rest of the field. Each side of the field is separated into six different colours displaying the different spots where each team starts.

2. Announcer Box

The announcer box is used to display which team’s turn it is. This horizontal box is located at the top of the screen. This box displays the team’s name and is displayed in the colour of the team.

3. Display Log

The display log shows the health bar, the damage given of the players pieces that are shown on the board. If the user is playing then the display box will only show the information of the pieces that are outside of the fog of war. If the user is not playing and the only players are AIs then the display box will show the information of all the players pieces on the board separated between the teams. This box will be a vertical rectangular shaped box to the right of the hexagonal playing field.

4. Forfeit Box

If the user would like to end the game early as they see no chance of winning their is a forfeit box located to the top right of the screen. Pressing this box with more then one user left in the game will simply remove the user’s player from the game. If they are the last user left than the game is moved into spectator mode.

5. Exit Box

This selection box is located directly below the “Forfeit” selection box. Pressing this button quits the in-game screen and brings the user to the main menu screen.

6. Turn Box

* Move
* Shoot
* End Turn

This box is located below the playing field. Within the box are three buttons. “Move”, “Shoot” and “End Turn”. When either “Move” or “Shoot” is pressed the option is given to the user to click on a piece on the playing field and execute that given command. If “End Turn” is pressed it simply end’s the players turn and the round goes to the next player.

7. Piece Display Log

* Sniper
* Tank
* Scout

These display logs are located to the left of the hexagonal playing field. There is one log box for each piece. These boxes display the health of the piece, damage that piece has given to other players pieces, range of the piece and the movement. If that piece is lost then the log is blackened out showing that the piece is no longer available.

8. Spectator Mode Options

These options are displayed in the turn box and are only displayed when the user is in spectator mode. The buttons given to the player are “Pause/Play”, “Fast Forward”, and “End Game”. This gives options to the user to pause the AI’s play-through of the game. “Fast Forward” also has the option of times one the speed, times two the speed and times four the speed. End Game will bring the player to the victory box screen.

9. Victory Box

This is a display box that only pops up after the game has finished and the player has not exited to the main menu. The box covers most of the screen and is used to display the player that won and the robots stat’s for each individual team.

PUT SYSTEM DIAGRAM HERE

-

-

-

-

-

-

-

-

-

-

-

-

-

--

Actors

Menu Navigator:

Navigates the main menu and game creation, and would perform operations such as managing robots, game creation, and exiting the game.

Actions:

* createGame
  + Creates new game instance
    - Can change the number of players and modify the board size.
    - Chooses which robots are to be used in the match.
* exit
  + Quits application.

AI:

Computer player in the main game. Performs tasks on a set of robots in the game, such as move or shoot, as well as signal the end of their turn.

Actions:

* move
  + Moves robot forward.
* turn
  + Rotates current robot on the tile.
* shoot
  + Applies damage to all robots on a target tile.
* endRound
  + Finds next robot to control on same team.
* endTurn
  + Allows next player to control their robots.

Human Player:

Human player in the main game. Like the AI, the Human Players perform tasks on a set of robots in the game. The human player can also exit or forfeit the game, as well as move to spectator mode.

Actions:

* move
  + Moves robot forward.
* turn
  + Rotates current robot on the tile.
* shoot
  + Applies damage to all robots on a target tile.
* endRound
  + Finds next robot to control on same team.
* endTurn
  + Allows next player to control their robots.
* forfeit
  + Removes the Human Player from the current match.
    - If no more Human Players are in play, the option for Spectator mode is given.
* exit
  + Quits application.

Spectator:

An observer to the main game. Cannot interact directly with any of the robots in the game. Has the ability to go to the end of the computer turns after they have completed it.

Actions:

* endRound
  + Finds next robot to control on same team.
* endTurn
  + Allows next player to control their robots.
* exit
  + Quits application.

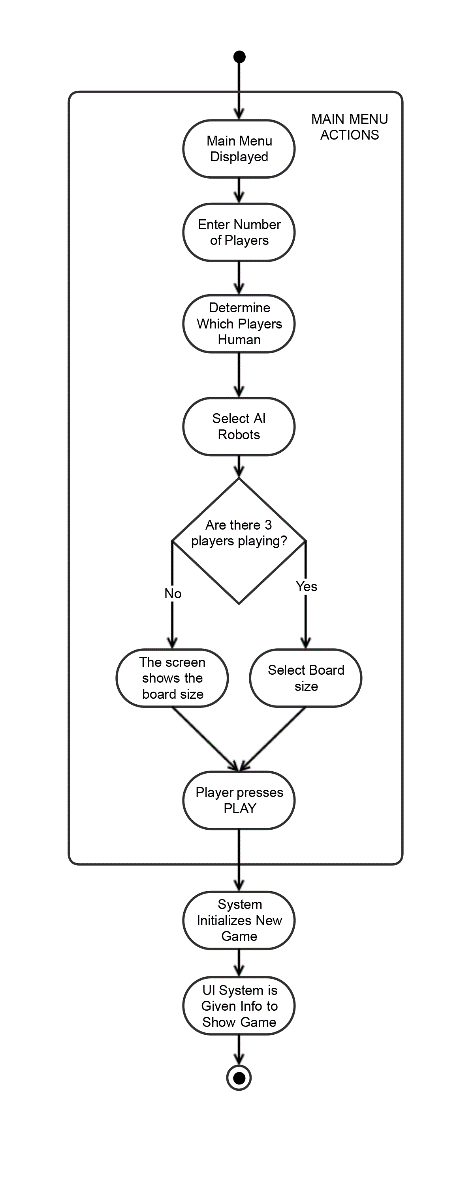
Robot Librarian:

Manages the robots in the system, as well as pulls data/updates for robots from a server.

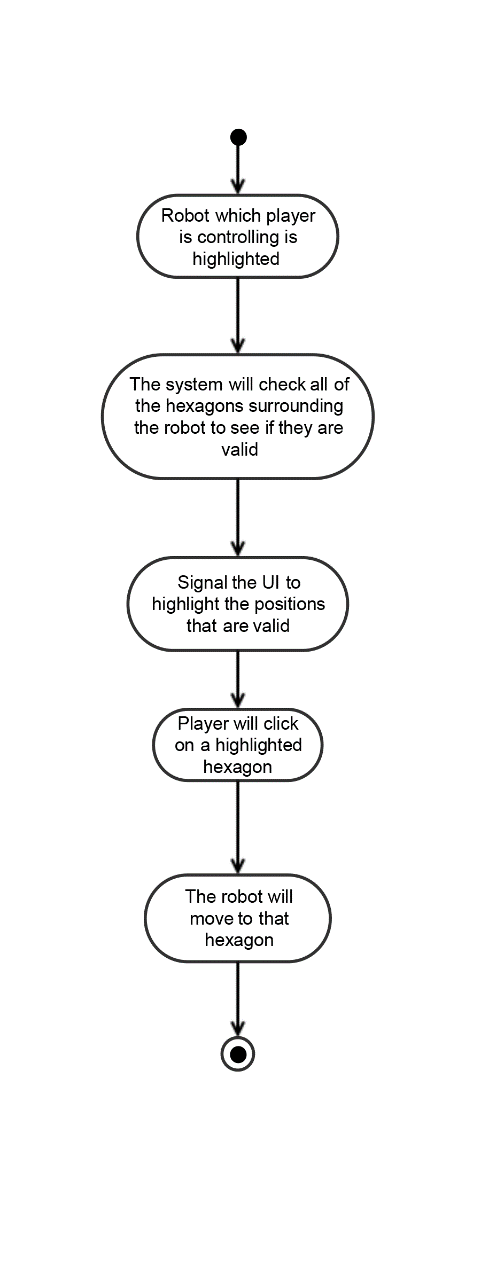
Actions:

* registeringRobots
  + Add new robot to application
* reviseRobot
  + Replace existing robot with updated code
* retireRobot
  + Freeze a robot so you can reuse name
* enumerateRobots
  + Display all robots stored
    - Sorted by values
    - Show stats and versions
* downloadRobots
  + Provide robot record so it can be run
* updateStats
  + Record details about robot result

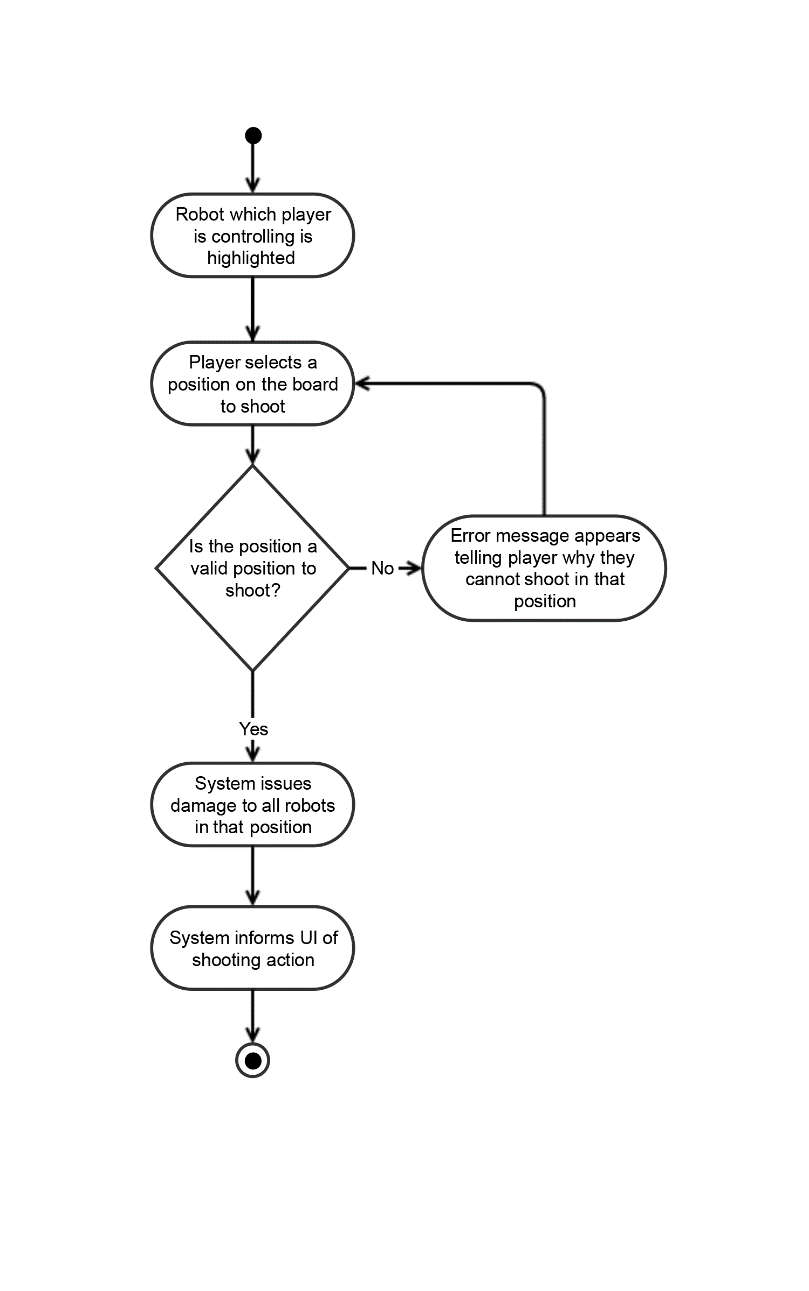
**Scenarios**

* Create Game Primary Scenario
* When a player presses “PLAY” from the main menu they are prompted with a display showing options that they could play the game with. These options are the number of human players, the number of computer players, the robots that the computer players are using, and an option to change the board size if there is exactly three players in the game. The player will then hit another “PLAY” button to continue to the in game view.

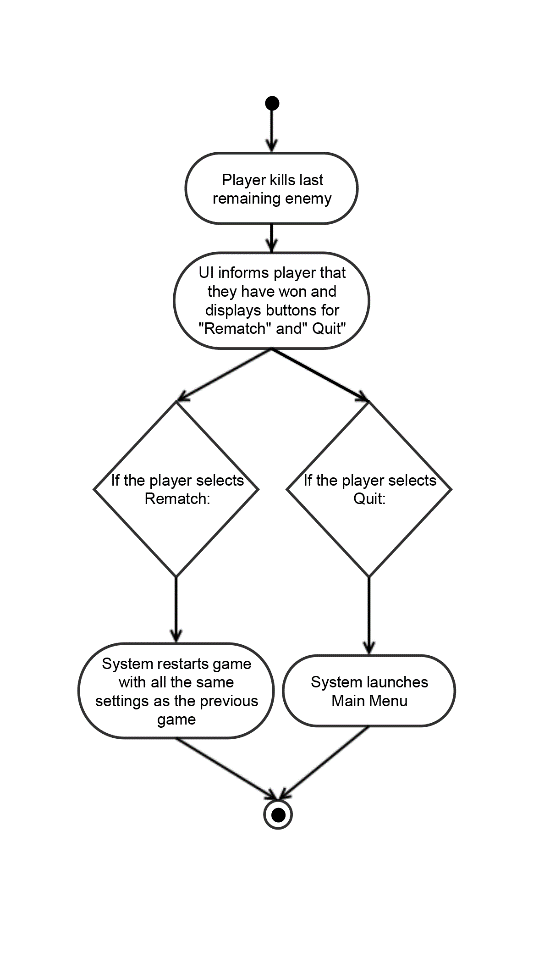
1. The use case begins when the player selects play from the main menu
2. The player enters the amount of players (2, 3, 6)
3. The player selects which players will be human and which will be computer
4. The player will select an AI for each of the three types of robot for every computer player
5. If the amount of players is 3 the player may select between a board size with sides of length 5 or length 7
6. The player presses play
7. The system initializes a new game with the specified parameters
8. The UI system is notified of the parameters and what will be displayed to the screen
9. The use case ends

* Move Robot Primary Scenario
* When it is a player’s turn and they wish to move a robot they can do so using the mouse. All of the spaces around the robot that it can possibly move to will be highlighted. Clicking on one of these locations will result in the robot moving one space into that location. If the robot has already moved as far as it can for that turn no spaces will be highlighted.

1. The use case begins when it is the player’s turn and they are in moving mode
2. The system will determine which of the six positions around the robot are valid locations to move to
3. The player will click on the game board on one of the hexagons adjacent to the robot that is highlighted
4. The system will determine the necessary number and direction of rotations the robot will need to make to face the selected hexagon
5. The system will perform the necessary rotations and move the robot forward into the selected position
6. The use case ends

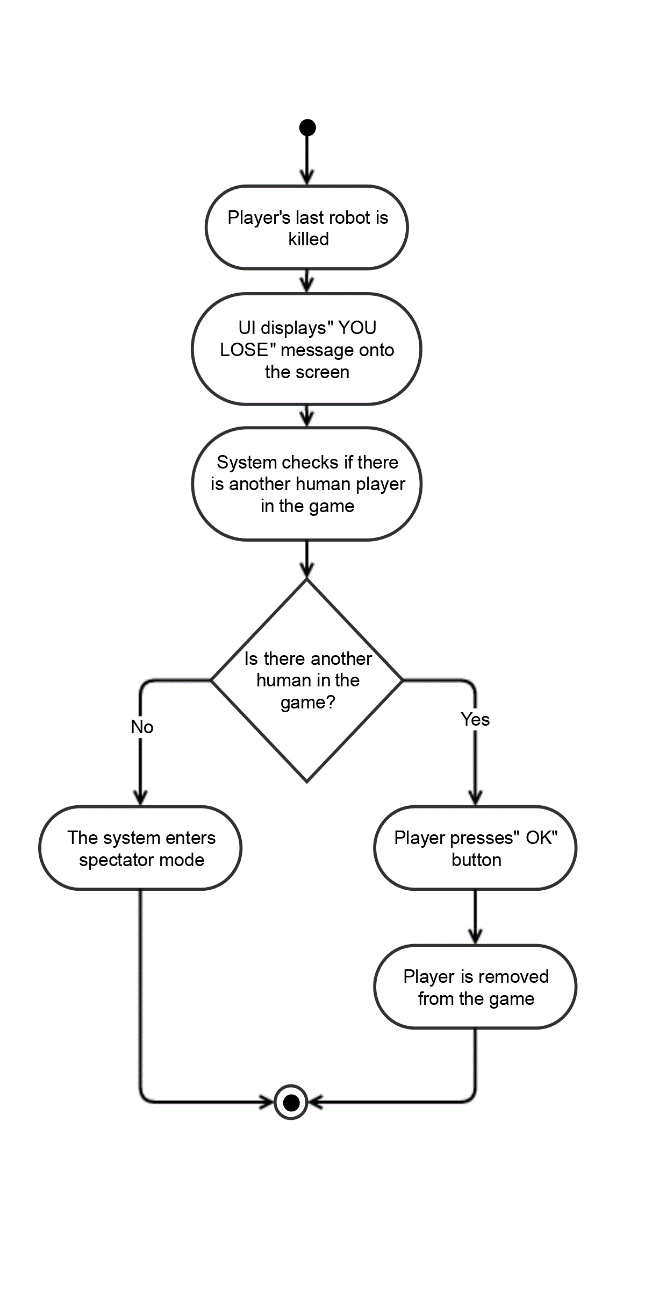
* Shoot Robot Primary Scenario
* When it is a player’s turn and they wish to shoot with a robot they can do so using the mouse. When the player is in shooting mode the player will be able to click on a board tile that is in range to fire upon that location. All robots in the location that is being fired upon will receive damage.

1. The use case begins when it is the players turn and they are in shooting mode
2. The player will select a position to shoot
3. The system will check that the board tile selected is within range and is a valid place to shoot.
4. If the board tile is selected is valid the system will issue damage to all robots on that board tile
5. The system will inform the UI of the shooting action
6. The use case ends



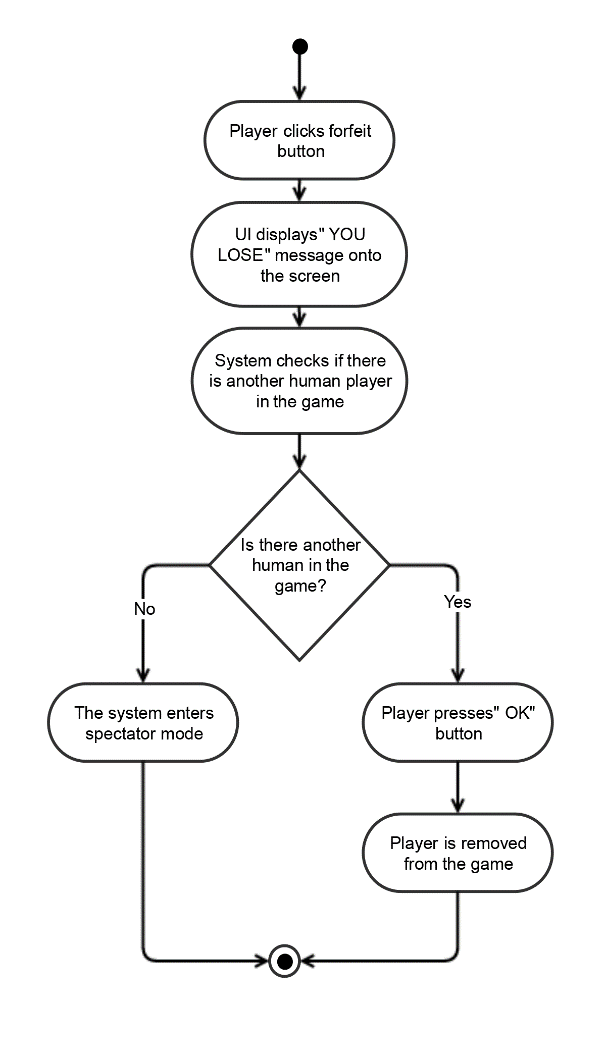
* Winning Primary Scenario
* If a player has killed the last robot of the only remaining team the player has won the game. A message will appear to that player that they have won and will prompt the player to have a rematch or to quit.

1. The use case begins when the player kills the last remaining robot of the only enemy team left alive
2. The system informs the UI of the victory
3. The UI shows on the screen the victory graphic and the two prompts for the player to select from, rematch and quit
4. If the player selects rematch the system will restart the game with all of the same settings as before
5. If the player selects quit the system will launch the main menu
6. The use case ends



* Losing Primary Scenario
* If a player’s last robot is killed they have lost the game. After a player’s last robot is killed when it is their turn again a message will show on the screen saying they have lost. If the player is the only human left in the game the game will enter into spectator mode.

1. The use case begins when the last robot of a player’s team is killed
2. The system informs the UI of the players state and the YOU LOSE message is shown
3. The system determines if there are any other human players in the game
4. If there are not any other human players in the game the game will enter spectator mode
5. If there is another human in the game the UI will simply display an OK button
6. The use case ends when the user selects an option

* Forfeiting Game Primary Scenario
* On any turn a player may choose to forfeit the game using the forfeit button in the top right corner of the screen. Upon doing so the player will be directed to the "YOU LOSE" screen which will be the exact same for either death or forfeit. If the player is the only human left in the game the game will enter into spectator mode.

1. Use case begins when the player selects forfeit when it is their turn
2. System changes all of the player's robots to 0 health.
3. The system informs the UI of the player’s state and the YOU LOSE message is shown
4. The system determines if there are any other human players in the game
5. If there are not any other human players in the game the game will enter spectator mode
6. If there is another human in the game the UI will simply display an OK button
7. The use case ends when the user selects an option

Spectator Mode

Once all of the human players in the game have forfeited or died the game will enter into spectator mode. In this mode the player(s) will not be playing but rather watching the remaining AI computer players make their moves. When in this mode the button options for “Move”, “Shoot”, and “End Turn” will be changed to “Pause/Play” , “Fast Forward”, “End Game” , “Fog of War”. During spectator mode the game will carry out based on time, regulating the speed of the computer player’s movements based on a timer. (Once every second the computer AI will make a move for example). This is why we have the need for speed controls in the spectator mode.

The fast forward control allows for quick viewing of the game and the pause button allows for the spectator time to view the board. The fog of war button is an on/off toggle for the fog of war view of the board. If the toggle is turned on then the spectator will only be able to see what the computer player can see of who’s turn it is. If the toggle is off, then the spectator can see the entire board regardless of which team’s turn it is. To exit spectator mode the spectator can press “End Game”. This button will skip all remaining turns of the computer players and show the Victory box displaying the stats of all the robots in the winning team.