

RoboSport

Brady Lang

Dylan Mcinnes

Parker Reese

Meagan Renneberg

Executive Summary:

This is a computer played board game called Robot Sport. Using simple point to shoot and move interactions it is the goal of each player to eliminate the enemy players from the board. We have identified sixteen actions as well as six actors that make up this system. Each action is described using Scenarios consisting of pre conditions, step-by-step events, post conditions, error conditions, sequence diagrams, and alternative cases. Each actor is described with an interface. Graphical user interfaces for each human to system interaction have been designed as well as described.

This requirements document completes the first phase of four in our software-engineering deliverables.

Figure 1

Game Rules:

The game rules are as follows. This game can be played with either two, three or six players. These players can consist of either entirely human players, entirely computer players or a mixture of computer and human players. There are two different sized boards that can be played on depending on how many players there are. If




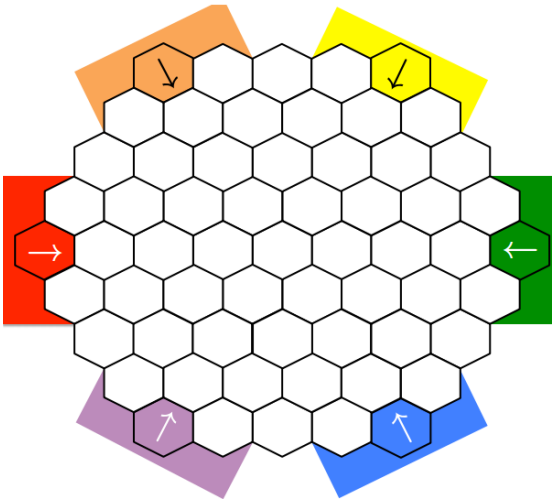
Scout	Sniper	Tank	
			
A:1	A:2	A:3	Attack
H:1	H:2	H:3	Health
M:3	M:2	M:1	Movement
R:2	R:3	R:1	Range

Figure 2

there are two or three players, then the board has five spaces per side (see figure 2). If there are six players, then the board has seven spaces per side. Each player is given a team of three robots: a scout, a sniper and a tank. Each robot has its own specific stats that can be seen in figure 1. The game starts with all robots starting on their teams colored arrow (see figure 2 for reference). Team colors are determined by how many players are in the game and the colors are randomly assigned. If there are two players then red and green are used, three players then red, yellow and blue are used, and if



there are six players all colors on the board are used. This game can be run on Unix, Windows or Linux and will utilize the tuxworld server for data and AI types. JSON is utilized for the storage of the robot library.

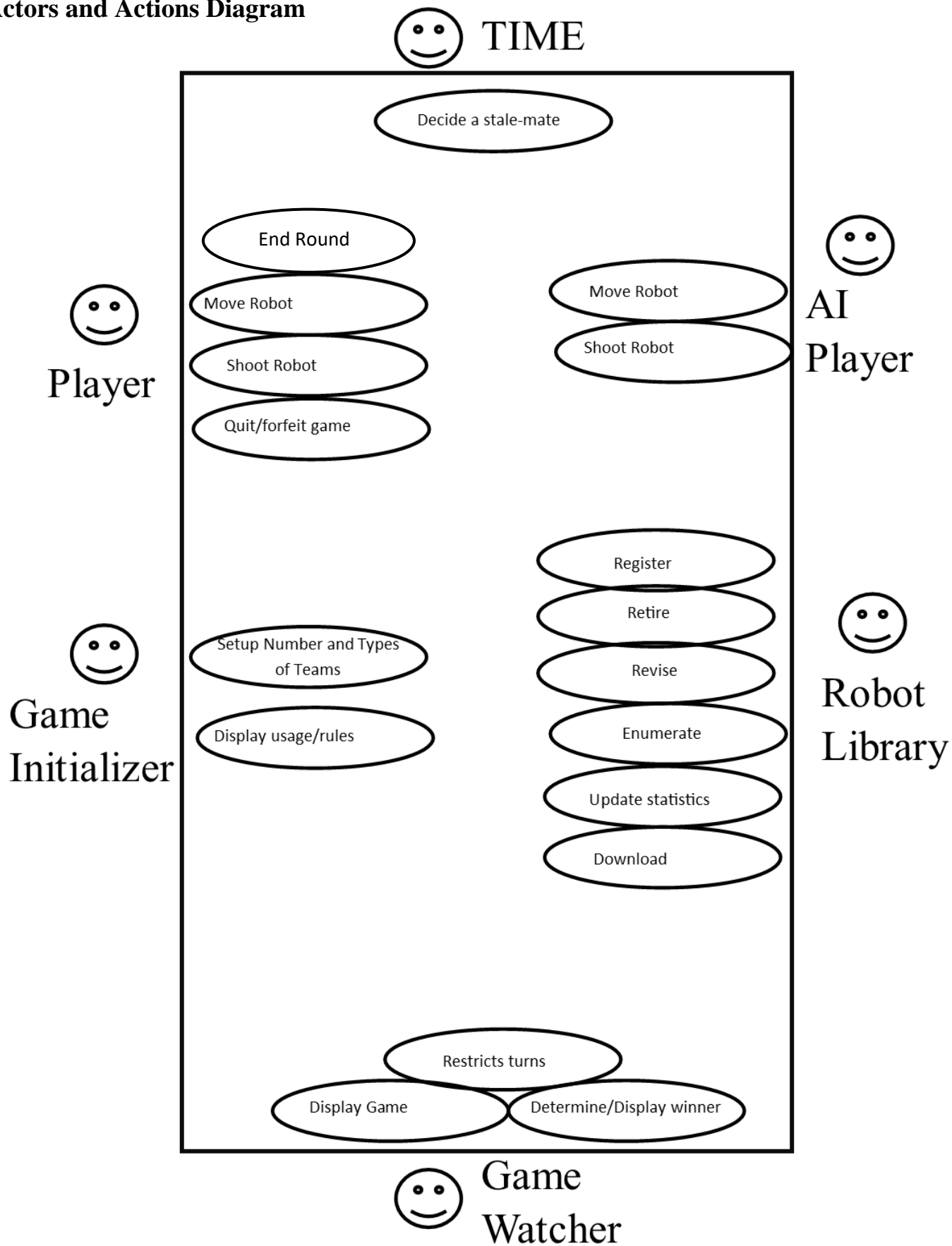
The game starts with team Red's scout robot and moves clockwise around the board to the next player. After every player has had a play with their scout, a new round is started with team Red's sniper and the next round with team Red's tank. Once all three rounds have been completed it is considered a turn. This order continues for the remaining rounds, if a player's scout has been destroyed when it's that player's play they just use the next robot in the rotation. So in this case they would use their sniper (assuming the sniper is not destroyed). If a player has no more robots they are eliminated and do not get a play, the game progresses to the next player clockwise.

At the beginning of a players turn they have the option to either move, shoot, do nothing, or quit. Players are able to move, shoot and then move again as long as they didn't use all their moves in their first movement (see figure 1 for reference). A robot can only shoot another space if that space is in range. If the space is in range and the player chooses to shoot it, then any robots on that space will take damage. Robots also have the option to shoot the space they are currently on, but be careful, robots can damage themselves. Each robot has different health and attack (see figure 1 for reference) so some robots are weaker than others. If a robot's health drops to zero or below, that robot is now dead and is no longer in play. The game ends when only one team has robots remaining.

#### System Diagram – Actors/Actions:

The system diagram showing the actors and actions of the system are as follows. See page 3.

## Actors and Actions Diagram



### Scenarios for All Actors, Action UML Diagrams and Actor Interfaces:

#### *Scenarios for Time actor:*

Time actor - Decide Stale-Mate:

-English description:

- In the event that only AI teams are active and 20 turns have passed without any robots being damaged, a draw will ensue. This will result in no wins, no losses, and no statistics updated.

-Pre Conditions:

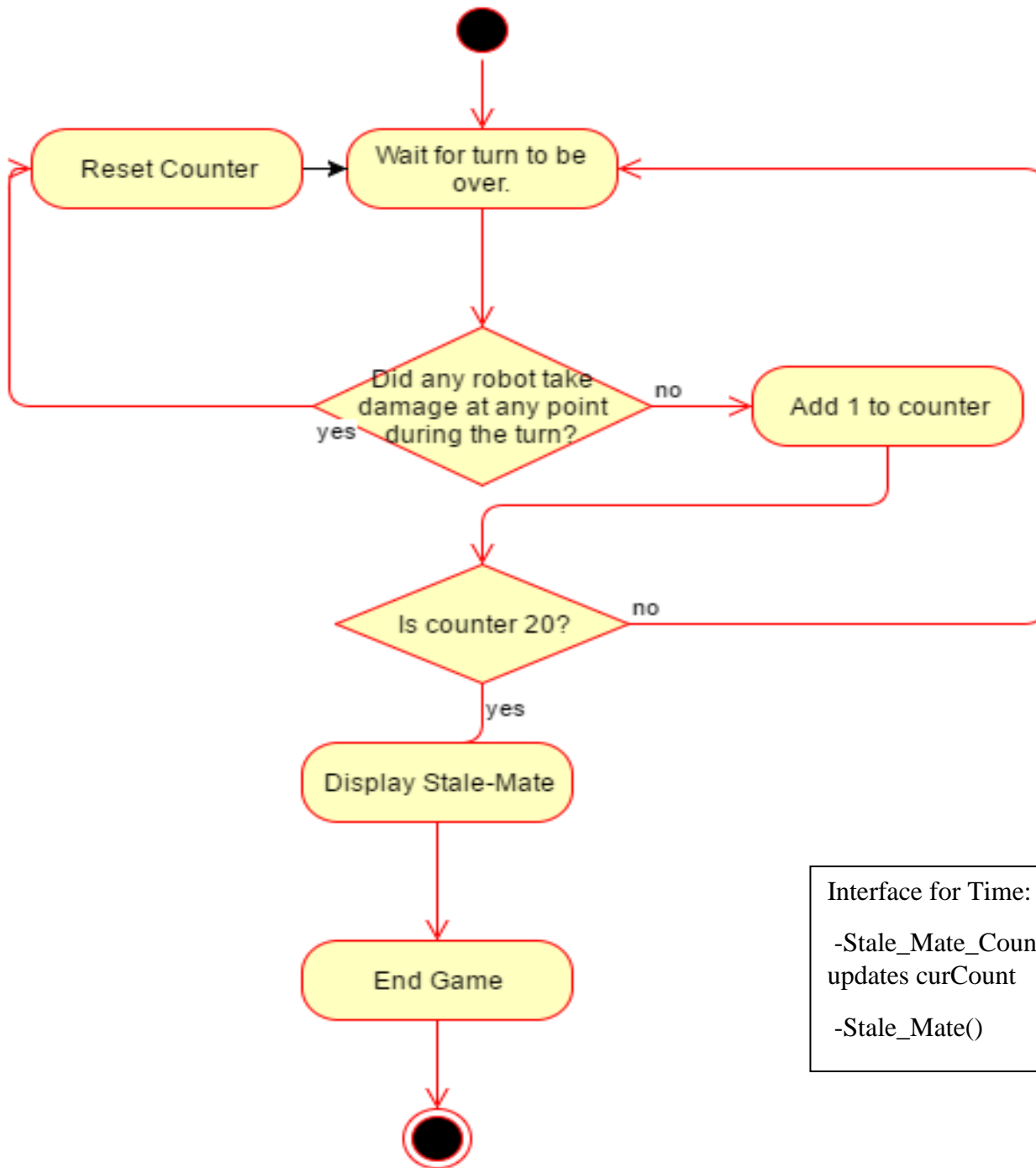
- Only AI Robots are present on the game board.
- 20 turns have passed without robots being damaged.

-Flow of Events:

- A draw is initiated.
- A message saying a draw has taken place will be shown.
- The game board view will exit.

Post Condition:

- No statistics have been changed.



Interface for Time:

-Stale\_Mate\_Count(curCount),  
updates curCount

-Stale\_Mate()

-The flow chart above is for the decide stale-mate action.

-The text box shows the interface for the actor Time.

## *Scenarios for Player*

### Player actor - Move Robot

#### English Description:

-For the player to move their robot it must be their teams turn to move and that specific robots turn to move and their robot must still be alive. If all preconditions are satisfied, then the robot moves to the clicked space.

#### -Preconditions:

- It is that teams turn to move
- It is that robots turn to move
- The robot is alive

#### -Flow of events:

- Player clicks on space they want to have their robot move to
- Robot moves to that space
- If player has not used all their movement points they may move their robot again

#### -Post conditions:

- The robot has moved to the chosen space

#### -Error condition

- If the player tries to move their robot either off the board or to a space that is out of their robot's range, an error message appears telling the player they must move their robot to a space that is in range on the board.

### Player actor - Shoot Robot

#### -English Description

-For the player to have their robot shoot another space it must be their teams turn to move and that specific robots turn to move, their robots must still be alive, and the space must be in range. If all preconditions are satisfied, then the player robot shoots the space and any robot on that space is damaged.

-Preconditions:

- It is that teams turn to move
- It is that robots turn to move
- The robot is alive

-Flow of events:

- Player clicks on space that their robot will shoot
- All robots on that space are damaged
- If player has not used all their movement points they may move their robot again

-Post condition

- All robots on the space shot are damaged

-Error condition

- The space the player clicks on is not in the robots range so an error message appears that tells the player they must click on a space in their robot's range

## Player actor - Quit/Forfeit Game

### Case (only two players)

-English description

-The player at any time can decided to quit or forfeit the game. The quitting player's robots are all destroyed and removed from the game. If there are only two players in the game, then the other player automatically wins.

-Precondition

- The player still has robots in the game
- There are only two players in the game

-Flow of events

- The player clicks a button to quit/forfeit the match and confirms their choice
- All their robots are destroyed and removed from play

-Post conditions

- The other player wins



### Alternative Case (more than two players):

#### -English description

-The player at any time can decide to quit or forfeit the game. The quitting player's robots are all destroyed and removed from the game. If there are more than two players in the game, then the game continues on without the quitting player.

#### -Preconditions

- The player still has robots in the game
- There are more than two players in the game

#### -Flow of events

- The player clicks a button to quit/forfeit the match and confirms their choice
- All their robots are destroyed and removed from play

#### -Post conditions

- The game continues on

### Player actor - End Round

#### -English description

- If it is the player's turn, their robots turn and their robot is alive. The player may end their turn by clicking the end turn button if all preconditions are met.

#### -Preconditions

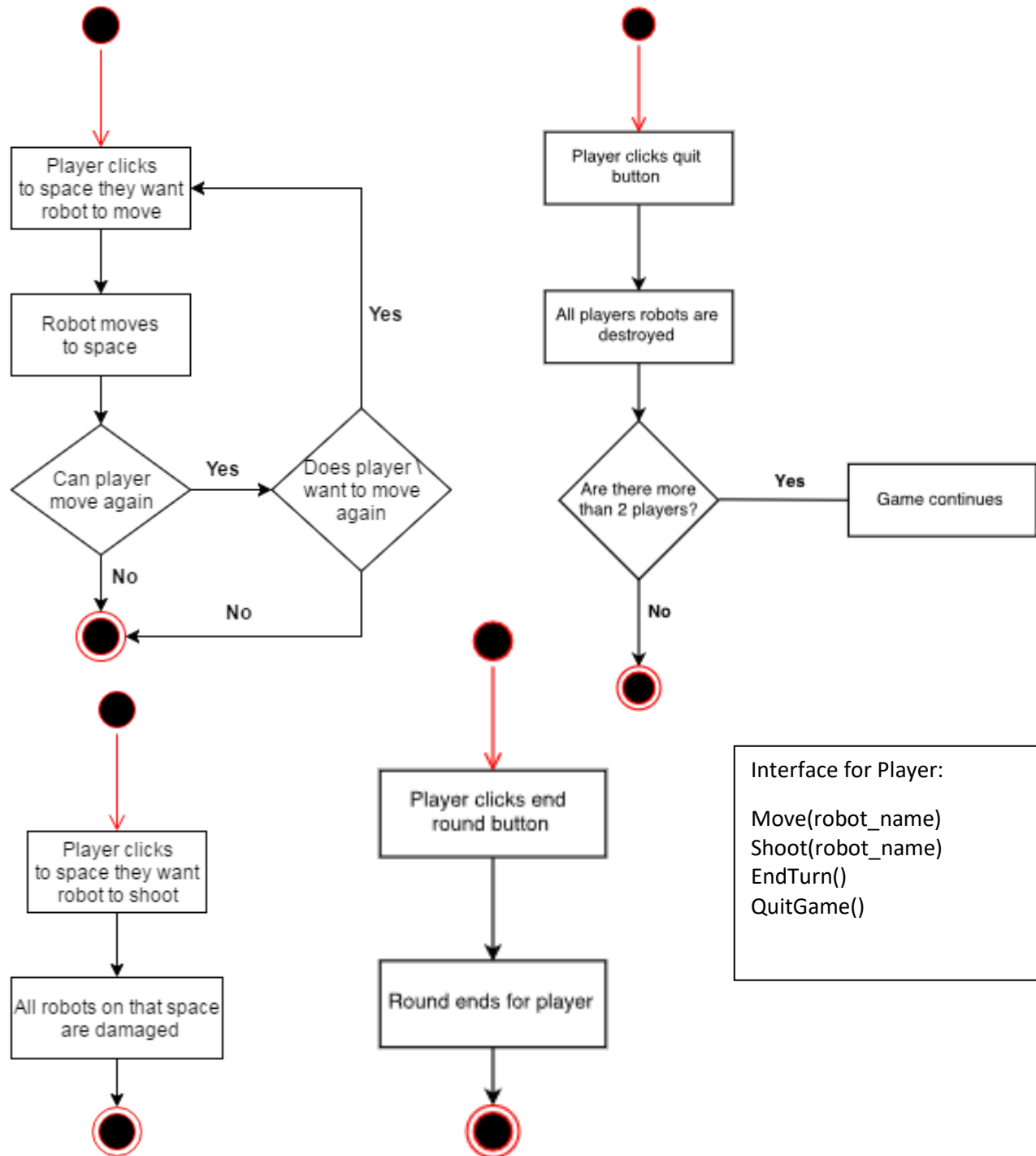
- It is that team's turn to move
- It is that robot's turn to move
- The robot is alive

#### -Flow of events:

- Player clicks "End Turn" button

#### -Post conditions

- The round progresses to the next player



- The top left flowchart is for the move action.
- The top right flowchart is for the quit action.
- The bottom left flowchart is for the shoot action.
- The bottom right flowchart is for the end turn action.
- The textbox shows the interface for the actor Player.

## *Scenarios for AI Player*

### AI Player actor - Move Robot

#### -English Description:

-For the player to move their robot it must be their teams turn to move and that specific robots turn to move and their robot must still be alive. If all preconditions are satisfied, then the robot moves to the clicked space.

#### -Preconditions:

- It is that teams turn to move
- It is that robots turn to move
- The robot is alive

#### -Flow of events:

- Player clicks on space they want to have their robot move to
- Robot moves to that space
- If player has not used all their movement points they may move their robot again

#### -Post conditions:

- The robot has moved to the chosen space

#### -Error condition

- If the player tries to move their robot either off the board or to a space that is out of their robot's range, an error message appears telling the player they must move their robot to a space that is in range on the board.

### AI Player actor - Shoot Robot

#### -English Description

-For the player to have their robot shoot another space it must be their teams turn to move and that specific robots turn to move, their robots must still be alive, and the space must be in range. If all preconditions are satisfied, then the player robot shoots the space and any robot on that space is damaged.

-Preconditions:

- It is that teams turn to move
- It is that robots turn to move
- The robot is alive

-Flow of events:

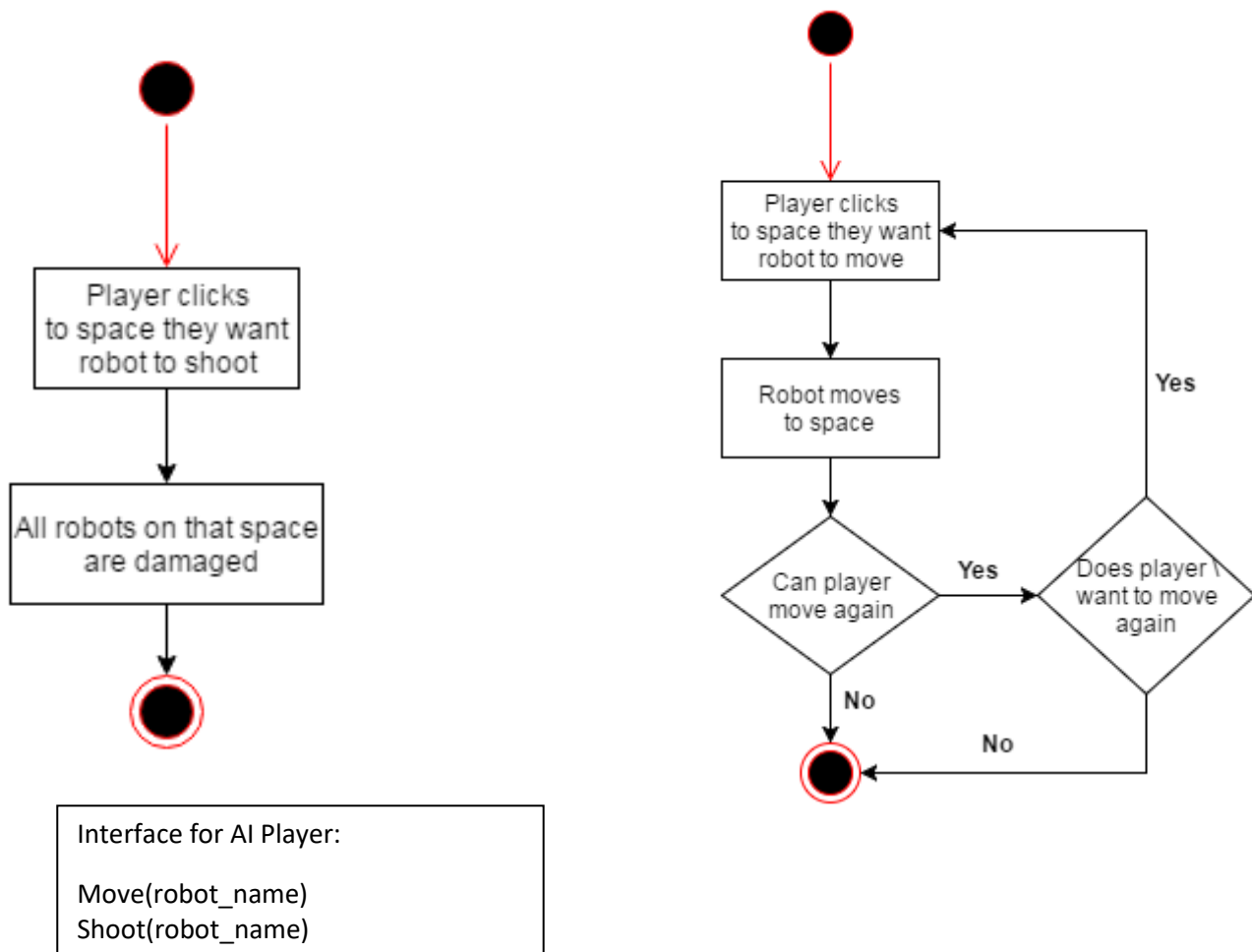
- Player clicks on space that their robot will shoot
- All robots on that space are damaged
- If player has not used all their movement points they may move their robot again

-Post condition

- All robots on the space shot are damaged

-Error condition

- The space the player clicks on is not in the robots range so an error message appears that tells the player they must click on a space in their robot's range



- The right flowchart is for the move action.
- The left flowchart is for the shoot action.
- The textbox shows the interface for the actor AI Player.

### *Scenarios for Game\_INITIALIZER*

#### Game\_INITIALIZER actor - Setup Number and Types of Teams:

##### -English Description:

-For the game initializer to setup the game it must first need to know the number of teams that are going to be playing in the game. This number includes both AI controlled teams as well as human controlled teams.

##### -Preconditions:

- “Start Game” has been clicked on the main menu

##### -Flow of Events:

- The radio box corresponding with the number of teams wanted has been clicked.
- The radio box corresponding to the amount of AI teams and Human teams has been clicked.
- Click the button “start” to finish selection.
- Choose team name for each human team from list.
- Choose team name for each AI team from list.

##### -Post conditions:

- Appropriate number of Teams, both Human and AI, are initialized.

##### -Error conditions:

- The team chosen does not have three robots.
- message displaying inability to choose team.

#### Game\_INITIALIZER - Display Rules:

##### -English Description:

-Should the button “Rules” be clicked on the Main Menu, a pop up box will be shown containing the rules of the game.

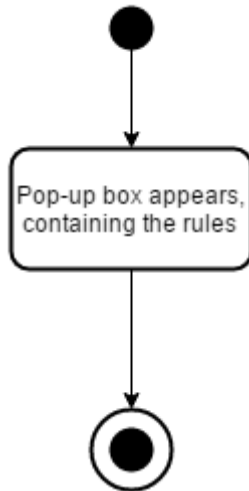
##### -Preconditions:

- “Rules” button has been clicked on the main menu

##### -Flow of events:

- pop-up box appears, containing the rules.
- pop-up box is exited.

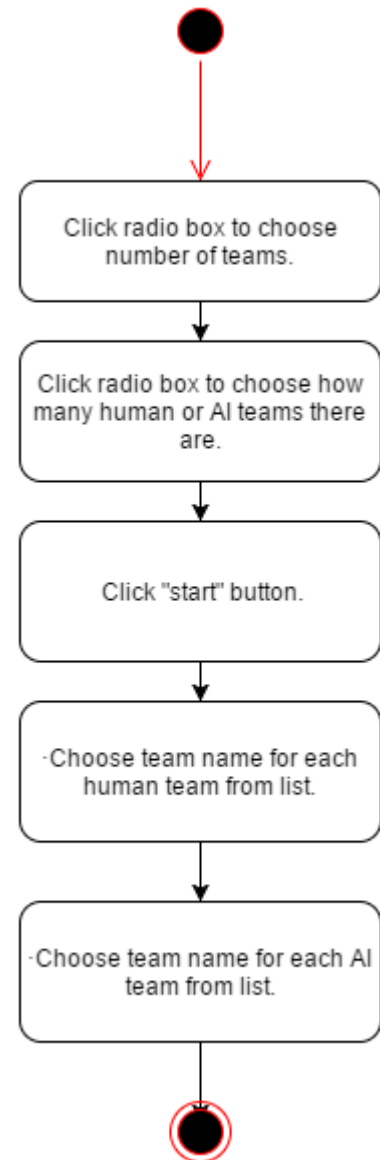
Rules button has been clicked.



Interface for Game\_INITIALIZER:

```

Setup(numPlayers, numHumans, numAI)
displayRules()
  
```



-The right flowchart is for the setup number and types of teams action.

-The right flowchart is for the display rules action

-The textbox shows the interface for the actor Game\_INITIALIZER.

### *Scenarios for Robot Library*

#### Robot Library actor - Register:

##### -English Description:

-Each robot used in the game must be registered to the robot library. Snipers, Tanks, and Scouts are chosen to register and must be given a name, if that name is not taken, it will be added to the library with blank statistics.

##### -Flow of Events:

- Click the button “Register Robot” on the Mange Robots GUI.
- Select the radio box “Scout”, “Sniper”, or “Tank” for which robot you intend to register.
- Type the name you would like to call the robot.
- Click “Ok”.

##### -Post conditions:

- A robot of that type and name has been added to the robot library with blank statistics.

##### -Error Conditions:

- The name entered is already in use:
- Display an error message stating the name is taken and to try a different one.

#### Robot Library - Retire:

##### -English Description:

-Select a robot from the list of robots in the library you wish to retire. This will remove the robot from the library so you are able to register a different robot with that name.

##### -Flow of Events:

- Click the button “Retire Robot” on the Mange Robots GUI.
- Given the list of robots that are registered, select the one you want to remove.
- confirm your choice.

##### -Post Conditions:

- The selected robot is removed from the library



### Robot Library - Revise:

#### -English Description:

-At any point in time the user is able to change the AI configuration of a robot in the robot library. They must select from a list which robot they want to revise and then select from a list which AI type they want to assign to that robot. Some examples of AI types include: "Aggressive, passive, sneaky, simple."

#### -Flow of Events:

- Click the button "Retire Robot" on the Mange Robots GUI.
- Click field under "Robot" to choose from a list of robots in the library you wish to revise.
- Click the field under "AI options" to choose from a list of AI types available.
- Click the button "Ok"

#### -Post Conditions:

- The robot's AI code has been updated to the chosen description.

### Robot Library - Enumerate:

#### -English Description:

-This will display all the robots in the robot library as well as their statistics. This list can be sorted by: Team, Name, Wins, Matches played, or Win/Loss ratio.

#### -Flow of Events:

- Click the button "Robot Statistics" on the Start Menu GUI.
- Sort the list in the manner you want by the given radio boxes.
- Scroll up and down on the given list of robots.
- Exit from the list.

### Robot Library - Update Statistics:

#### -English Description:

-At the end of every game, each robot that played will have their individual statistics changed to accommodate the latest game. It will update the robot's individual results (whether it died or survived), the team results (won or lost), damage inflicted and taken, distance travelled, and shots fired.

#### -Pre Conditions:

- A game has been completed and a winner determined.

-Flow of Events:

- Each robot name that played in the game will be sought out in the robot library.
- The running stats taken during the game will be added to the robot's stats in the library.
- Each team name that played will be sought out in the robot library.
- The running stats taken during the game will be added to the team's stats in the library.

-Post Conditions:

- The statistics for all the robots as well as the teams will be updated.

Robot Library - Download:

-English Description:

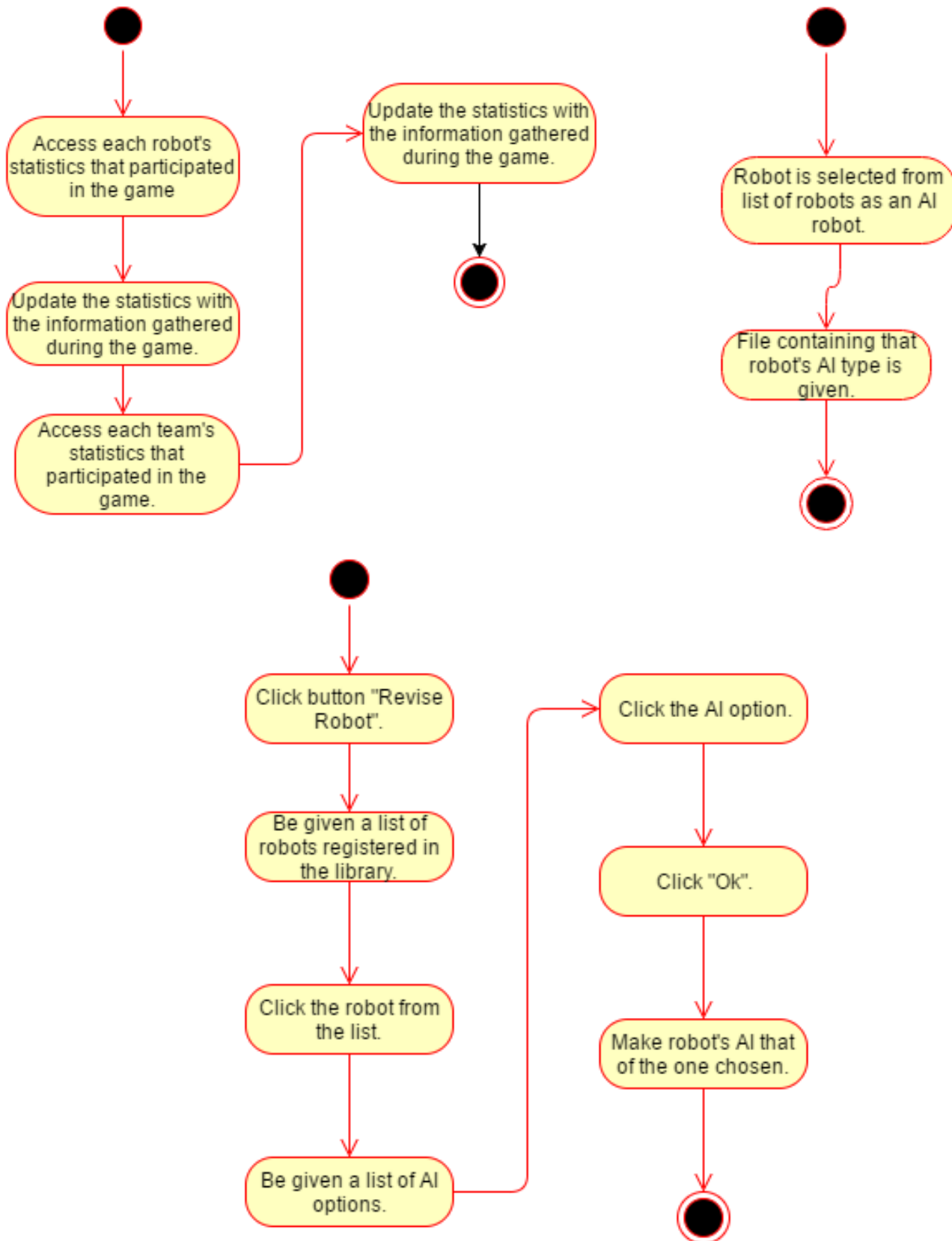
-When a robot is being used as an AI robot, it's record will be downloaded so our AI simulator can run it accordingly.

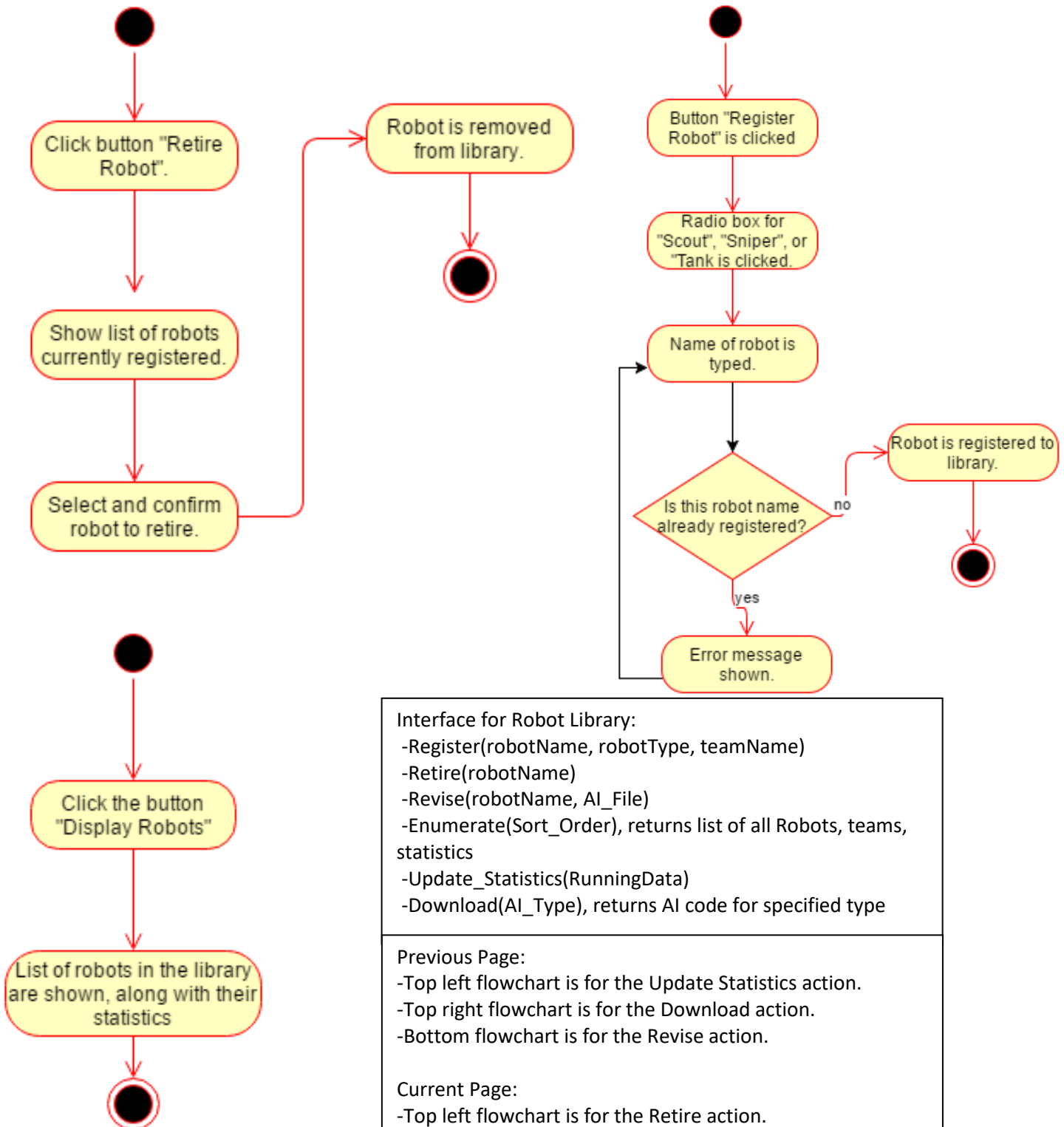
-Flow of Events:

- Robot is selected from list of robots as an AI.
- File containing the robot's AI type is given to the Game Initializer.

-Post Conditions:

- AI robot is moving, shooting, and turning, in accordance to it's AI type.





#### Interface for Robot Library:

- Register(robotName, robotType, teamName)
- Retire(robotName)
- Revise(robotName, AI\_File)
- Enumerate(Sort\_Order), returns list of all Robots, teams, statistics
- Update\_Statistics(RunningData)
- Download(AI\_Type), returns AI code for specified type

#### Previous Page:

- Top left flowchart is for the Update Statistics action.
- Top right flowchart is for the Download action.
- Bottom flowchart is for the Revise action.

#### Current Page:

- Top left flowchart is for the Retire action.
- Top right flowchart is for the Register action.
- Bottom left flowchart is for the Enumerate action.

In the textbox above, it shows the interface for the Robot Library actor.

### *Scenarios for Game Watcher*

#### Game Watcher - Display Game:

##### -English Description:

-The game board as well as team's robots are to be displayed for the players to interact with in accordance to the game rules

##### -Preconditions:

- The action "Setup Number and Types of Teams" has been completed.

##### -Flow of Events:

- The game board is shown.
- Any robots within the current robots range of sight is shown.

##### -Post conditions:

- Game is displayed

#### Robot Library - Determine Winner:

##### English Description:

-Should the game reach a point where only one team's robots are left standing, a winner will be declared and shown.

##### -Preconditions:

- A robot "died".

##### -Flow of Events

- Is there only one team's robots left?
- Display a winner dialog box showing which team has won.
- End the game.

##### -Post conditions:

- The game is back at the Main Menu.

##### -Alternative Paths:

- Before the 2nd step, there are more than one team's robot's left
- Do not display anything.

##### -Post Conditions:

- The game is continuing.

## Game Watcher actor - Restrict Turns:

### -English Description:

-During the game only one robot is allowed to play at a given time. As such, all other robots will be restricted in their ability to play.

### -Precondition:

- It is a team's turn to play.

### -Flow of Events:

- Depending on the round, the corresponding robot is used.
- No other team can move/shoot while it is a different teams turn to play
- Once the team has finished moving and/or shooting, the next team is given the capabilities to move.
- Every other team is once again restricted in their ability to move/shoot.

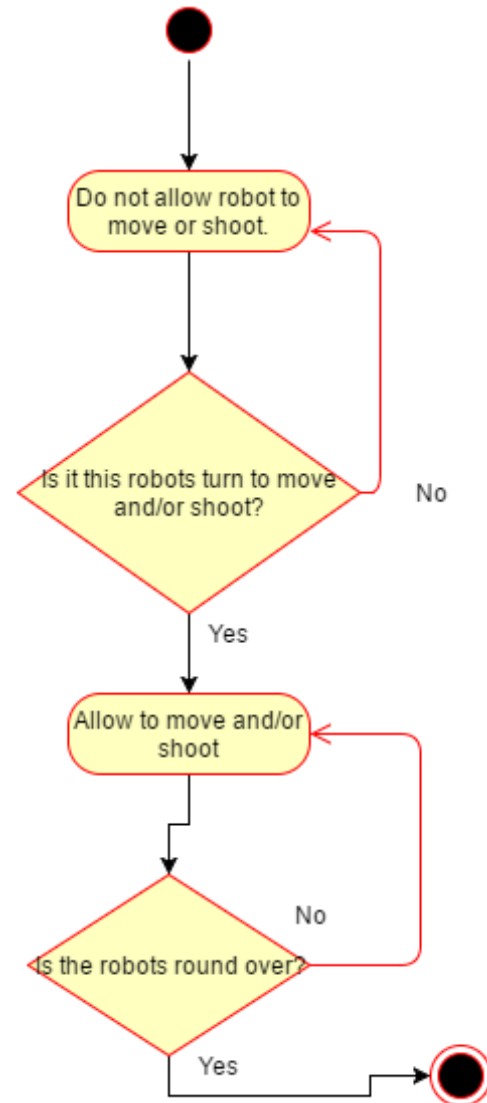
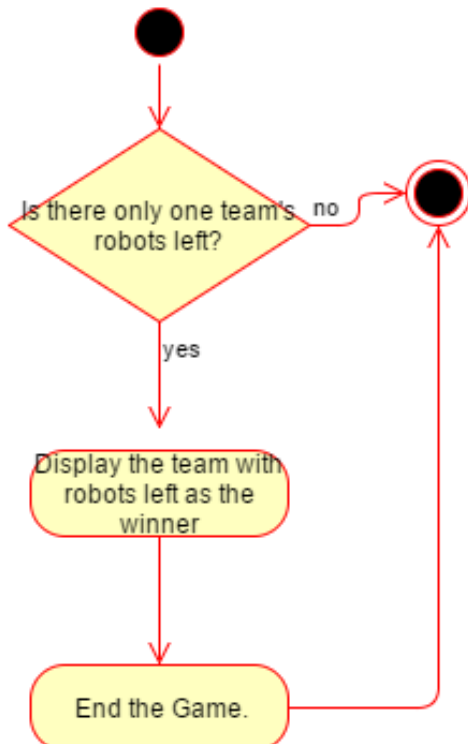
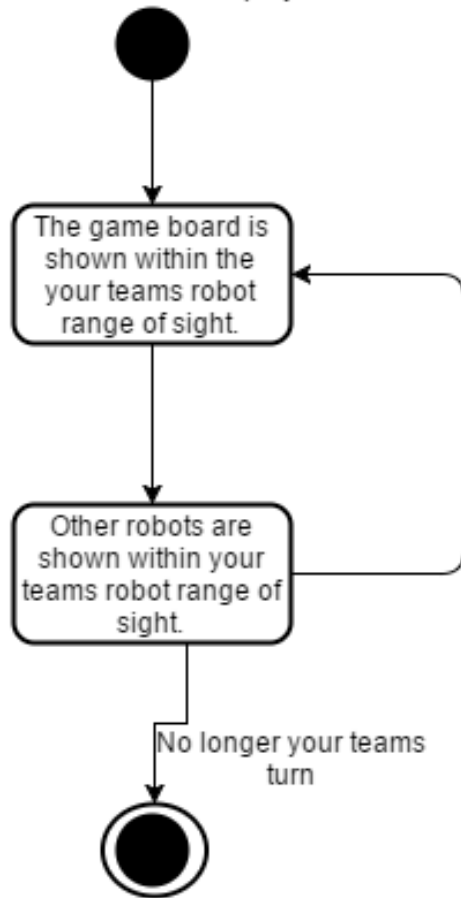
### -Post Conditions:

- No two robots are moving or able to move at the same time in any given round or turn.

### -Alternative paths:

- Before step 2, the corresponding robot to the round is dead
- The next robot in the cycle is given the ability to move/shoot instead
- Instead of step 3, the team decided to quit.
- All team's robots are destroyed.
- Game continues without team

Your teams turn to play



Interface for Game Watcher:

- Display\_Update()
- Restrict\_Robot(Robot\_Name)
- UnRestrict\_Robot(Robot\_Name)
- Check\_Winner(), return Team Name

-Top Left flowchart is for the Display Game action.

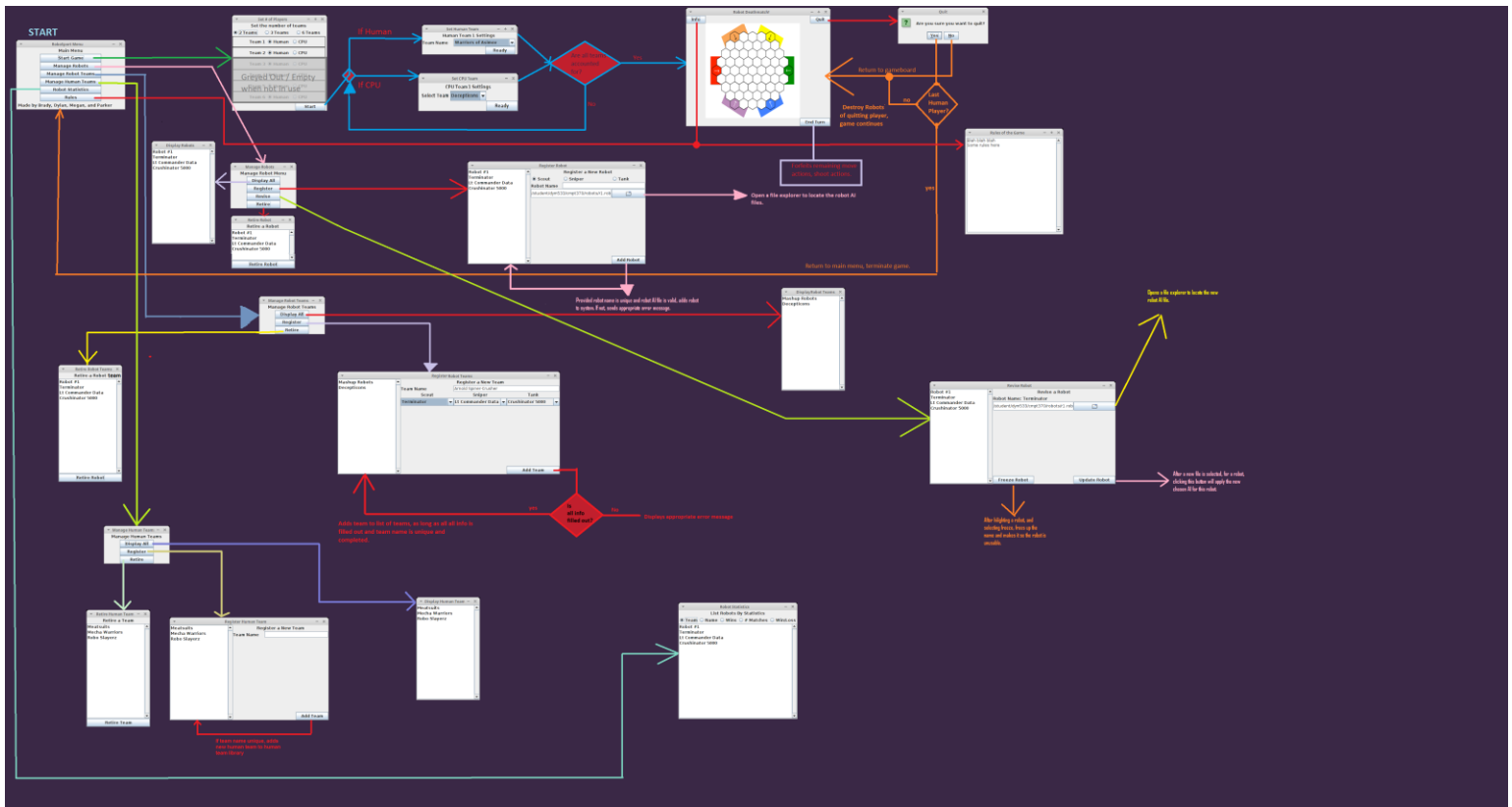
-Top right flowchart is for the Restrict Turns action.

-Bottom left flowchart is for the Determine/Display Winner action.

-The textbox directly above shows the interface for the Game Watcher actor.

## Graphical User Interfaces:

The following pages contain mock-ups of what the system is expected to look like, accompanied by corresponding descriptions.



For a better view of the GUI Interaction Map please see the accompanied “guiInteractionMap.png” file.

### Introduction Menu:

The Main Menu will be the first thing the user interacts with on program start up.

It will have buttons to Start Game, Manage Robots, Manage Robot Teams, Manage Human Teams, Robot Statistics, and the Rules.

<Nice to have - Not Mandatory> Credits will be listed along the bottom.

- The Start Game Button will bring the user to the Setup Game screen.
- The Manage Robots button will bring the user to Manage Robot Menu.
- The Manage Robot Teams button will bring the user to the Manage Robot Teams Menu.



- The Manage Human Teams button will bring the user to the Manage Human Teams Menu.
- The Robot Statistics button will take you to the Robot Sort menu.
- The Rules button will bring the user to the Rules display.

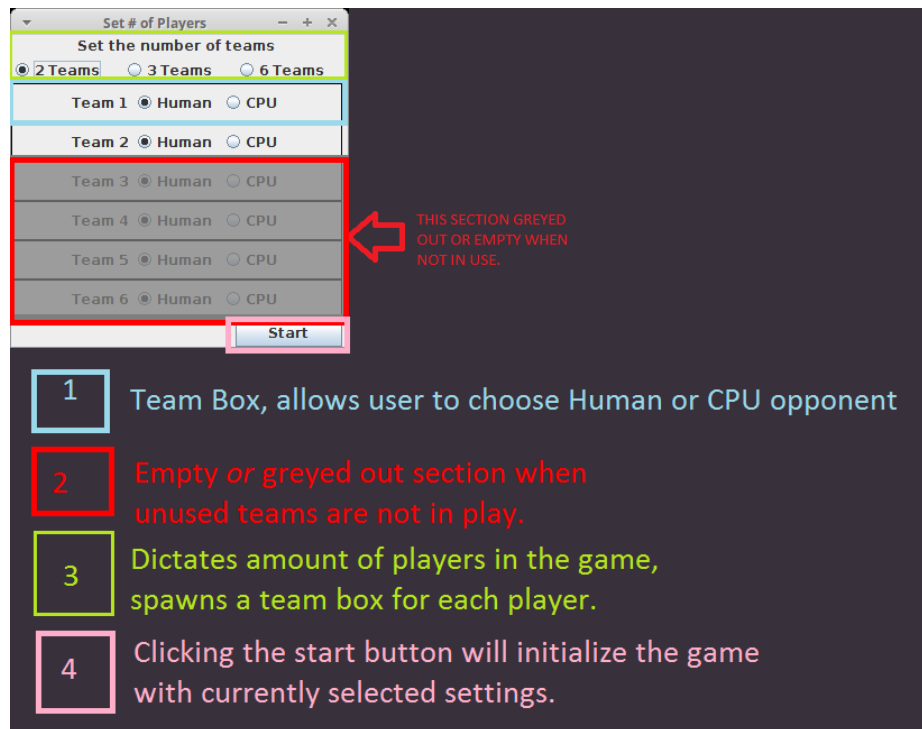


### *Setup Number of Players:*

Once a game has been selected to start, the set number of player's screen will pop up. The "Set number of Players" screen will allow the user to choose how many teams they'll play within the game, and which teams will be controlled by a human or CPU.

- The playable options are selected via radio buttons. The options are 2, 3 or 6 teams.
- Below this, an appropriate number of boxes depicting the teams are spawned.
- If '2 Teams' radio button is selected, two team boxes will appear.
- If '3 Teams' radio button is selected, three team boxes will appear.
- If '6 Teams' radio button is checked, six team boxes will appear.
- Inside each team box will be two radio buttons, allowing the user to select HUMAN or CPU player.
- Lastly, a button at the bottom right will start the game, taking the input from the radio buttons into the game master initializer.

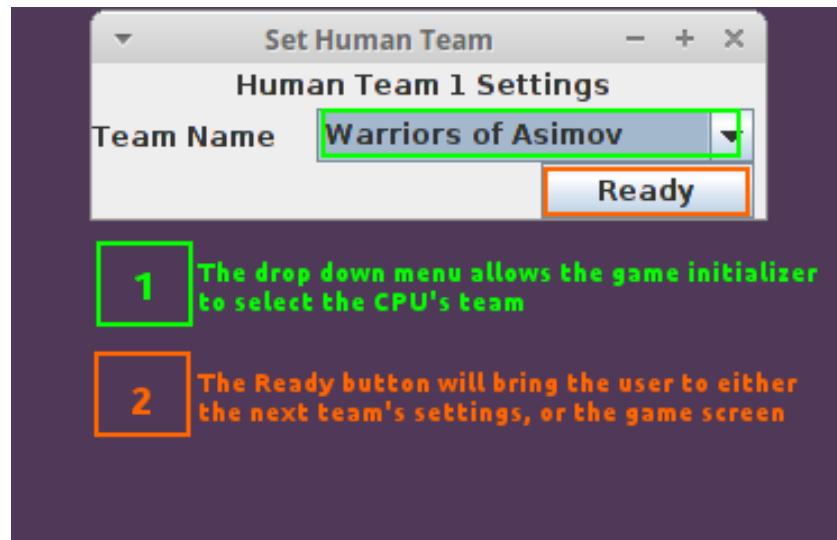
Start button will spawn the robot selection GUI box and human selection GUI box for each respectively chosen team.



### *Set Up Human Team Menu:*

The setup Human Team will be the screen the game initializer will be shown when a game is started that contains a human team. It will have a drop down menu for team selection, and a button to confirm the selection.

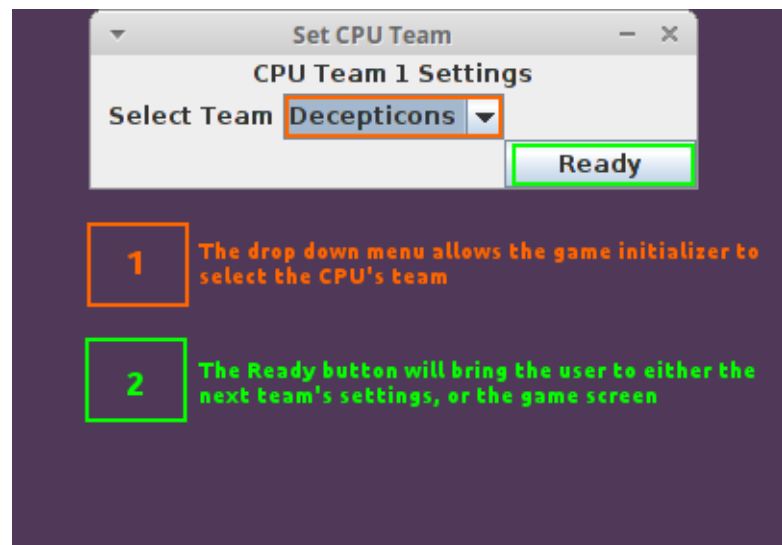
- The drop down menu allows the game initializer to select the CPU's team.
- The Ready button will bring the user to either the next team's settings, or the game screen.



### *Set Up CPU Team Menu:*

The setup CPU Team will be the screen the game initializer will be shown when a game is started that contains a CPU team. It will have a drop down menu for team selection, and a button to confirm the selection.

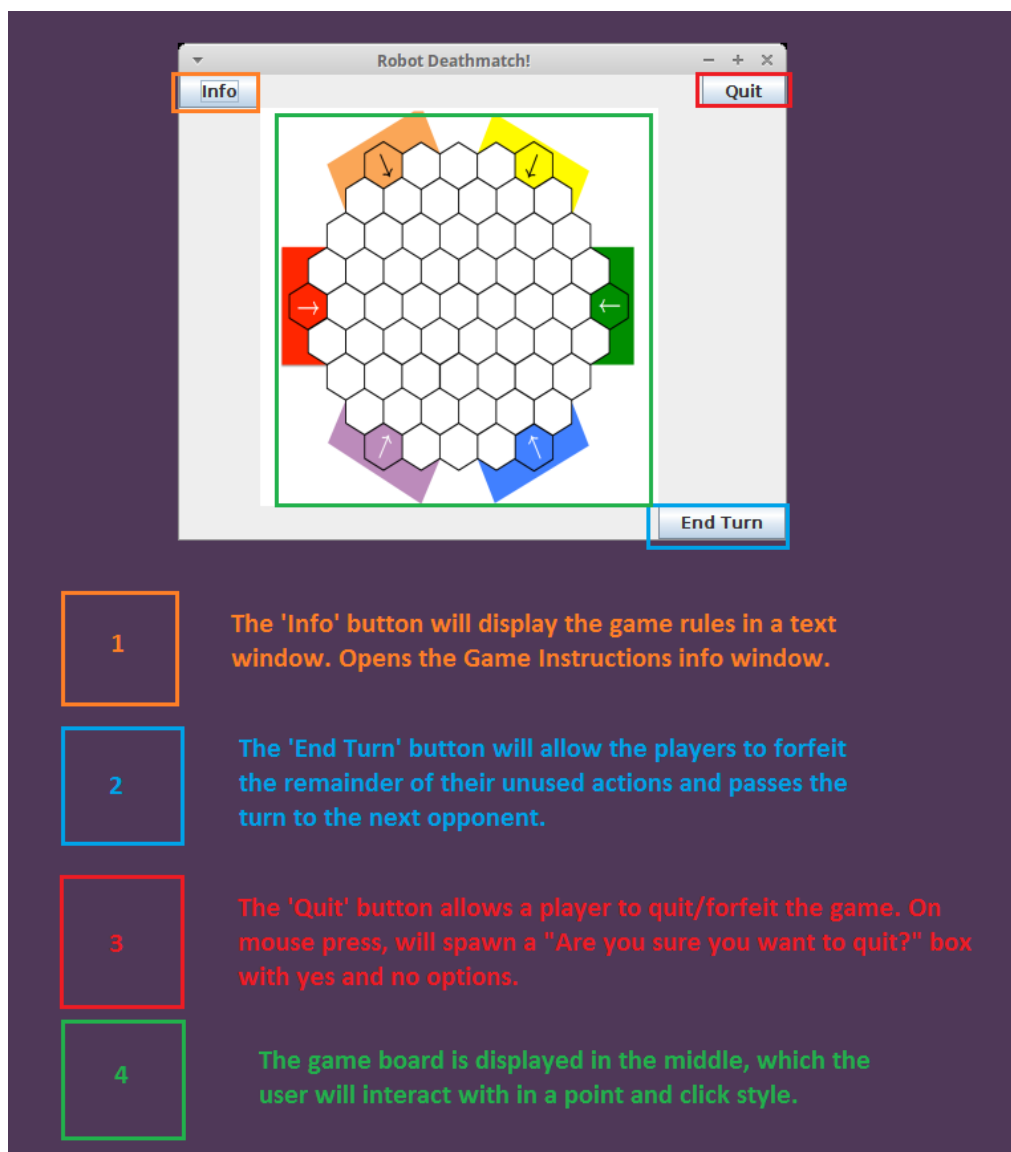
- The drop down menu allows the game initializer to select the CPU's team.
- The Ready button will bring the user to either the next team's settings, or the game screen.



### *The Game Board Screen:*

The game board screen features info, end turn and quit buttons. It also displays the game board during play.

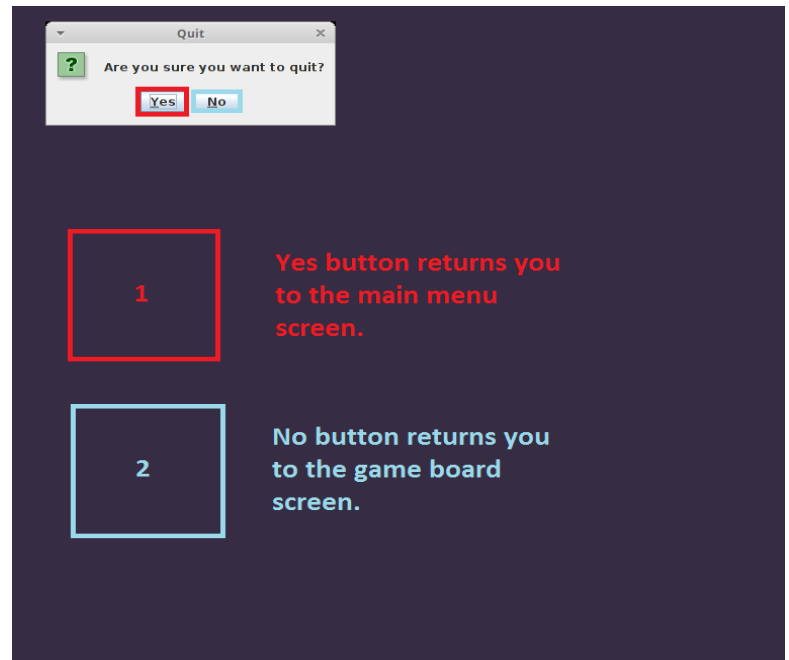
- The info button will display the game rules in a text window. Opens the Game Instructions info window.
- The quit button allows a player to quit/forfeit the game. On mouse press, will spawn a "Are you sure you want to quit?" box with yes and no options.
- The end turn button will allow the players to forfeit the remainder of their unused actions and passes the turn to the next opponent.
- The game board is displayed in the middle, which the user will interact with in a point and click style.



### *'Are you sure you want to quit?' Menu:*

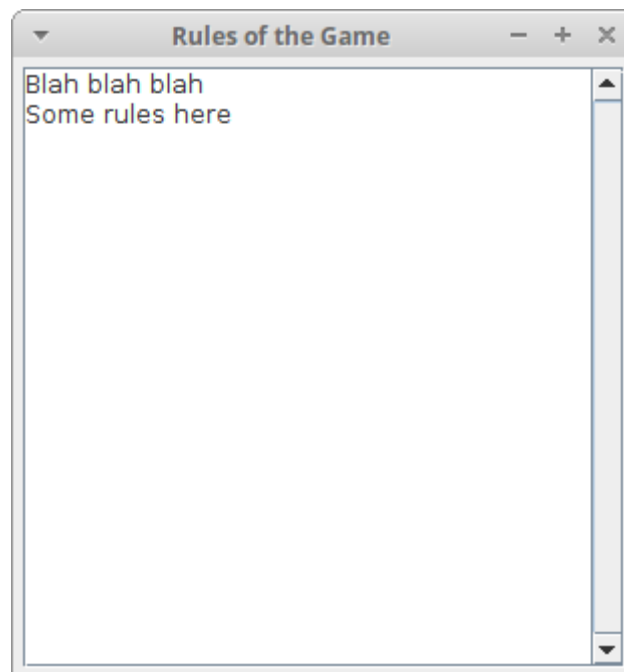
A confirmation menu with yes and no buttons asking a user if they're sure they want to exit game.

- If the yes button is selected, the user is returned to the main menu.
- If the no button is selected, the user is returned to the game screen.



### *Info Box - Game Instructions:*

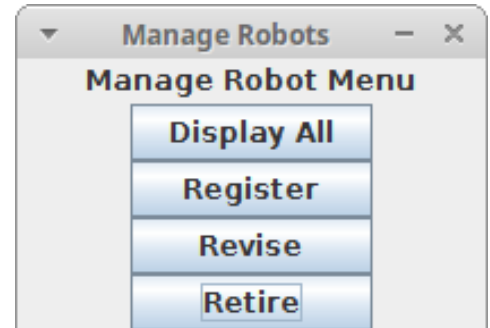
Displays a popup textbox of all rules in the game in a text format. Scrolls if the file is long enough.



### *Manage Robot Menu:*

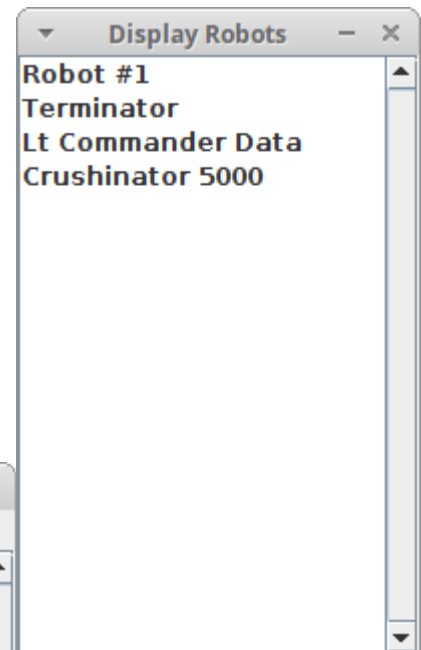
The manage robot menu lets the user choose to register a new robot, revise a robot or retire a robot. It can also display all current robots in the system.

- The Display All Button displays all robots currently in the system. Opens the 'Display All' robots window.
- Register takes the user to the screen to register new robots in the system.
- Revise takes the user to the screen to freeze or update existing robots in the system.
- Retire allows a user to retire the robot, permanently deleting it off the system.



### *'Display Robots' window:*

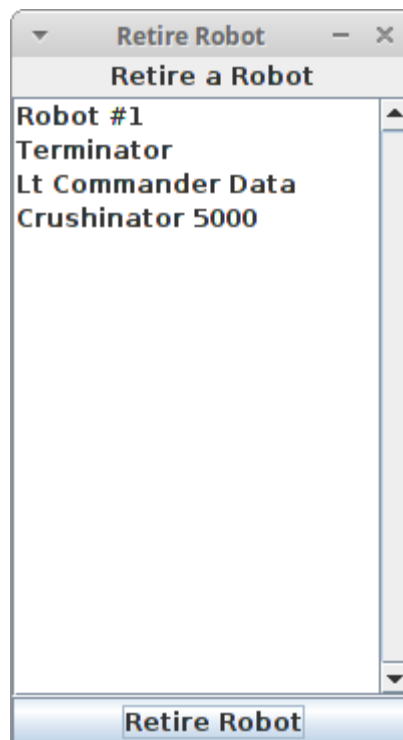
Displays a popup textbox of all robots within the system.



### *Retire Robot System:*

Displays a list of all current robots in the system, and allows the user to delete them.

- Shows an on-click highlight list with all robots in the system.
- The Retire Robot Button deletes the highlighted robot off the system.

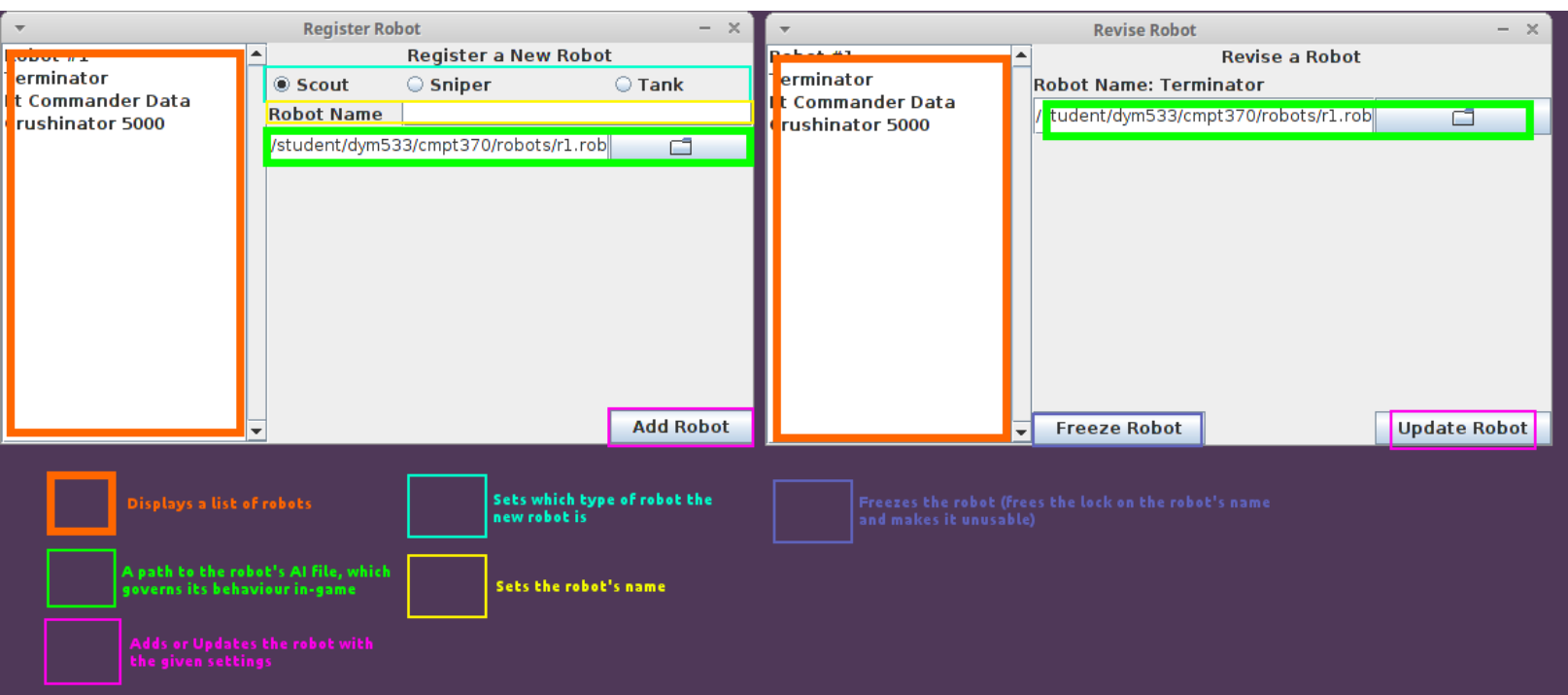


### Register Robot:

Allows the user to add new robots into the system. User gets to choose which type of robot, as well as it's name.

### Revise Robot:

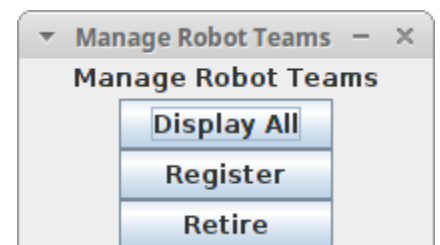
Allows the user to edit a robot that is already in the system. The user can change the AI type of a robot, or “freeze” a robot. Freezing a robot will allow its name to be used again and is no longer a playable robot.



### Manage Robot Team Menu:

The 'Manage Robot Teams Menu' lets the user choose to register a new team or retire a team. It can also display all current teams in the system.

- 'Display All' Button displays all robot teams in current system. Opens the 'Display All' teams window.
- 'Register' takes the user to a screen to register new robot teams in the system.
- 'Retire' takes the user to a screen to retire a robot team that exists in the system.



### *Register Robot Team Menu:*

This view will allow the user to register a new team of robots. There will be a display of the existing teams, a place to enter a new team name, three select menus for robots, and an add team button.

- The Add Team button will exit this part of the GUI, adding the robots to a team.

The screenshot shows a window titled "Register Robot Team" with a standard macOS-style title bar (minimize, maximize, close buttons). Inside the window, the title "Register a New Team" is centered at the top. On the left side, there is a scrollable list box containing the text "Mashup Robots" and "Decepticons". To the right of this list box is a form area. At the top of the form area is a text input field for "Team Name". Below this, there are three columns of robot types: "Scout", "Sniper", and "Tank". Under each column is a dropdown menu, each currently displaying "Robot #1". At the bottom right of the window is a blue button labeled "Add Team".

### *Retire Robot Team System:*

Displays a list of all current robot teams in the system, and allows the user to delete them.

- Shows an on-click highlight list with all teams in the system.
- The 'Retire team' Button deletes the highlighted team off the system.

The screenshot shows a window titled "Retire Robot Team" with a standard macOS-style title bar. Inside the window, the title "Retire a Team" is centered at the top. Below the title is a scrollable list box containing the text "Mashup Robots" and "Decepticons". At the bottom of the window is a blue button labeled "Retire Team".



### *Manage Human Team Menu:*

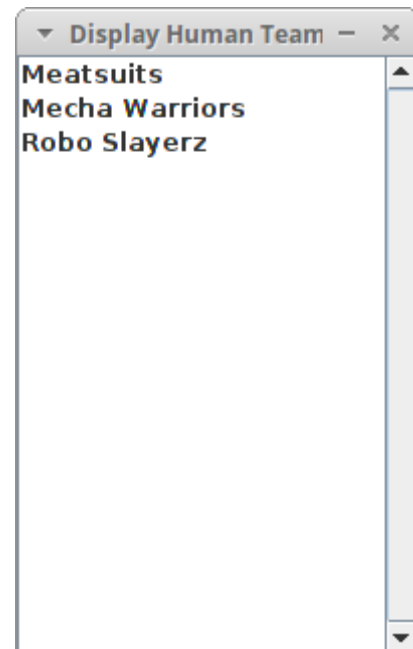
The 'Manage Human Teams Menu' lets the user choose to register a new human team, or retire a human team. It can also display all current human teams in the system.

- 'Display All' Button displays all human teams in current system. Opens the 'Display All' teams window.
- 'Register' takes the user to a screen to register new human teams in the system.
- 'Retire' takes the user to a screen to retire a human team that exists in the system, permanently deleting it off the system.



### *'Display All' Human Teams window:*

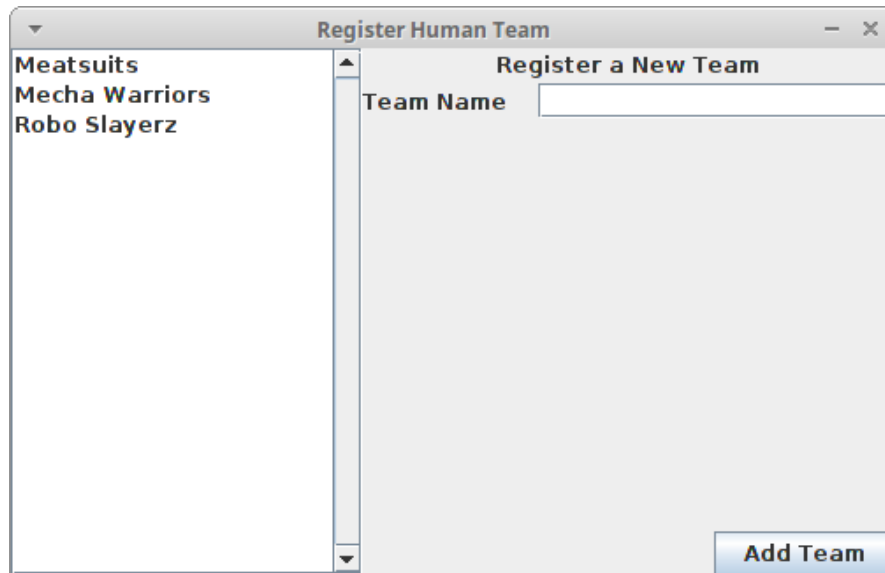
Displays a popup textbox of all robots within the system.  
Lists human teams by name.



### *Register Human Team Menu:*

This view will allow the user to register a new human team. There will be a display of the existing teams, a place to enter a new team name, and an add team button.

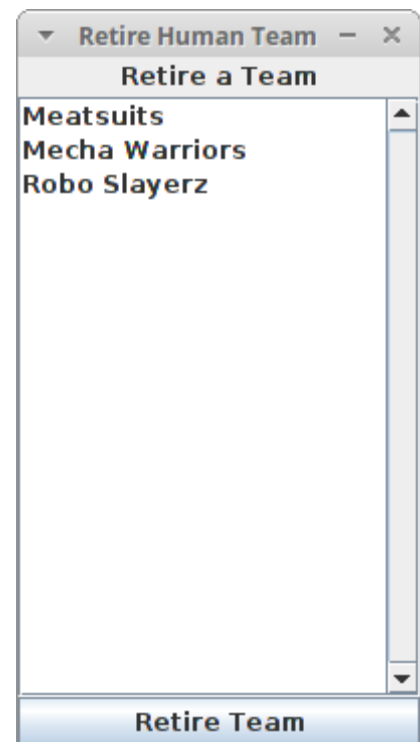
- The Add Team button will exit this part of the GUI, adding a new human team.



### *Retire Human Team System:*

Displays a list of all current human teams in the system, and allows the user to delete them.

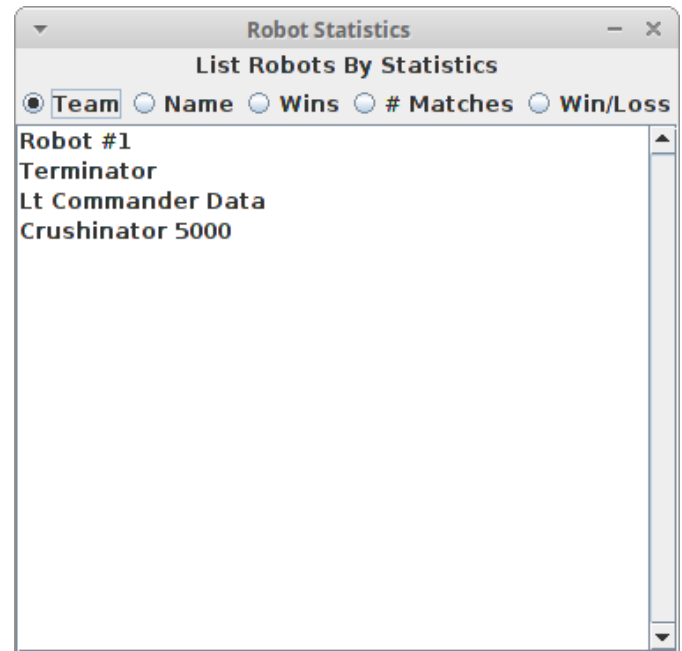
- Shows an on-click highlight list with all teams in the system.
- The 'Retire Team' Button deletes the highlighted team off the system.



*Sort Robot Menu:*

This view will show all the robots sorted by various metrics. It will have radio buttons to select the statistic to sort by, and a list of the robots.

- The various radio buttons will select the sort used to sort the robots.
- The list will show all the robots sorted based on some metric.



Created tag on GIT labeled "REQUIREMENTS\_COMPLETE" to mark the completion of this requirements document.