RoboSport 370: Requirements Document

Group B1

Adam Ronellenfitsch, Dylan Prefontaine, Evan Snook

Matthew Frisky, and Wynston Ramsay

CMPT 370

Professor Chris Dutchyn

Tutorial Leader Jonathon Baxter

Table of Contents

Section 1: Game Summary

RobotSport370 is a turn-based strategy game in which the goal is to be the player with the last unit(s) standing. The game supports two, three or six players and is played on a different size of board depending on the number of players. A game can consist of all human players, all AI players or any combination of the two. If there are two players they must play on a size 5 board, if there are six players they must play on a size 7 board, and if there are three players they must choose between the two board sizes. Each player begins on a designated tile on the board with their three units, each with slightly different properties. A scout which can move 3 spaces, has an attack range of 2, health of 1 and deals 1 damage, a sniper which can move 2 spaces, has an attack range of 3, health of 2 and deals 2 damage, and a tank which can move 1 space, has an attack range of 1, health of 3 and deals 3 damage. Games are sorted into 3 different sections:

1. A play is where a player has a chance to move or shoot with one of his units, this unit is decided in order of highest movement first and lowest movement last.
2. A round is when every player has had a chance to use their unit.
3. A turn consists of when every player has now had a chance to play each of their units at least once.

Players can move and shoot in any order, but they cannot exceed their unit's maximum move distance in their turn but moving is not required and rotating a unit does not cost movement. The red player starts the game using their unit with the highest movement, they may move and shoot any units within its range, once red's turn is over, the player to reds left does the same and this continues until it is reds turn again. Red then continues the round by controlling his unit with the next highest movement. If a player's unit is dead, the player may still make a move with their highest moving unit since their last play. Once a player has no more units to move, the player is out of the game.

Section 2: Programming Task

RoboSport370 can be observed, played as a single player, or played with multiple people. Thus our game has an inherent need for bots with A.I. to fill up positions that are not occupied by an actual player. In cases where you wish to observe or have a match with 2,3, or 6 players and do not have enough humans to play, then A.I. can be a substitute. The game also has an element that is to hide parts of the board from each player. This is simple to do for an A.I. as they will only be able to see what their script allows them. This is different for human players as the interface will have to hide what they are not meant to see. Additionally, our game will not involve networking, and so if there is more than 1 human player, the screen must be 100% hidden from both players at the end of a turn. This is so that they may switch seats, then the player must press start turn so that neither can see information that should be hidden from them while the turn changes. Our software's architectural approach will be Model View Controller where the model updates the view, and the user sees the view and uses the controller to change the model. We chose this because it works well with interfaces and heavy user interaction as we want to be able to choose what inputs the user gives the model so that the view does not do anything unexpected. Other possible architectures would be Presentation-Abstraction-Control and Hierarchical Model-View-Controller. However, both of these would allow for our agents to interact with each other which is not something that we want; each should only manipulate their own data. This piece of software will be developed using a specific platform of other programs and software that meet the group’s needs. The hardware this game is designed for will be on the Tuxworld computers in the Spinks Laboratory of the University of Saskatchewan. The main software used will be: GitLab for version control, JSON for structuring data, Microsoft Word for creating documents with images, Discord for group communication, Gliffy for sequence diagrams, Sublime for text editing, and either Netbeans or Eclipse to serve as an IDE. Java, JavaSwing, and AspectJ will be the language used to develop most of the game. The only thing that Java will not be used for is artificial intelligence decision making, which will be done with Forth.

Section 3: Stakeholder Identification

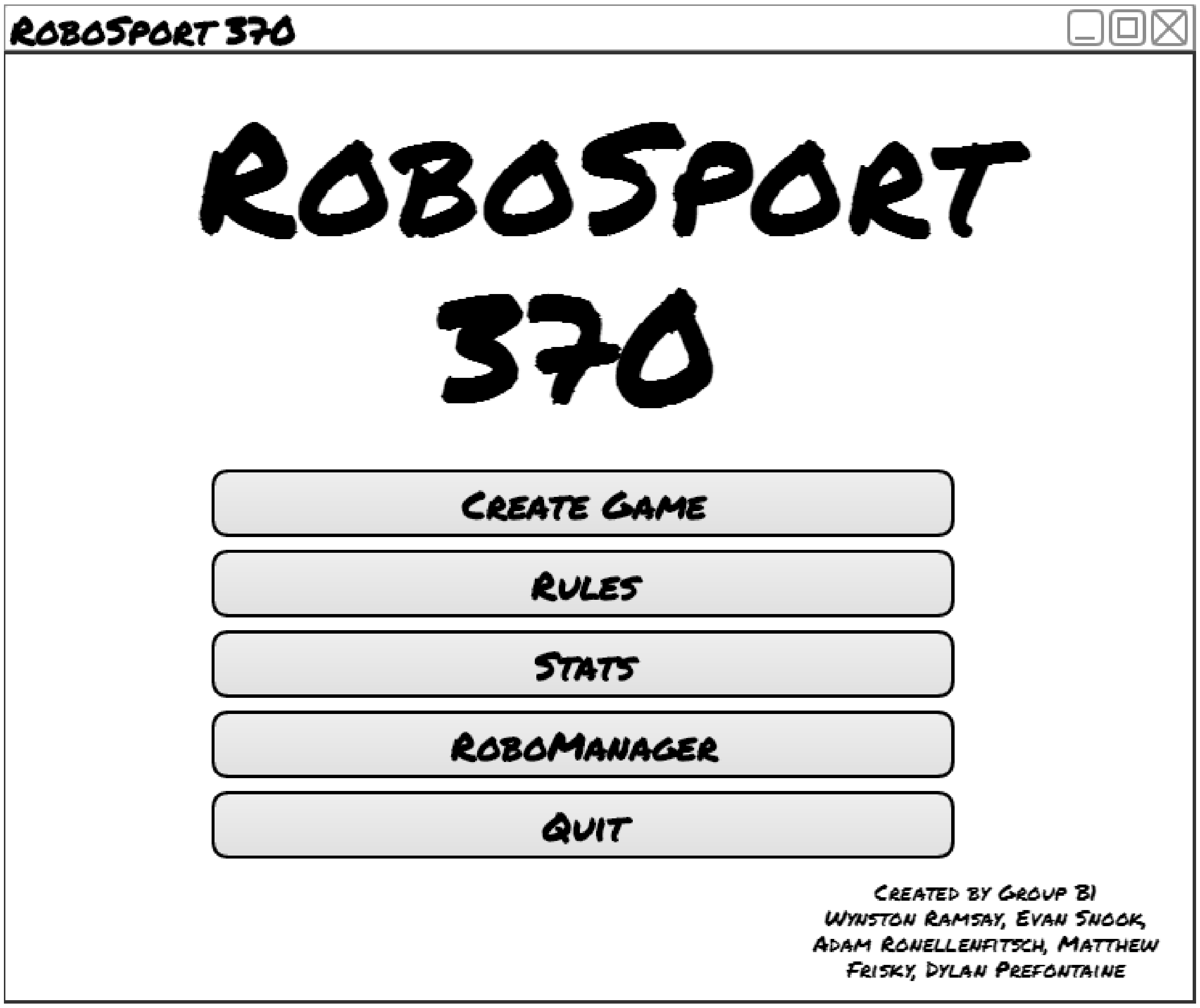
This piece of software is being designed and built for Professor Chris Dutchyn for CMPT 370. Our stakeholder is interested in our design process and documentation as it is a crucial part of software engineering to actually design your program before you begin to build it. During this process he will be looking at which features we choose to and not to add to the program as the project is tailored so that there are certain scenarios that arise that we must identify and deal with. Once we have successfully designed our program, we will then be graded for how well we structure our code and what tools we properly make use of. Lastly, all of this must be completed in a very strict time frame as the class only runs for one semester and thus we must make use of our time appropriately.

Section 4: User Interfaces

RoboSport 370 will have various screens to display to the user depending on the scenario.

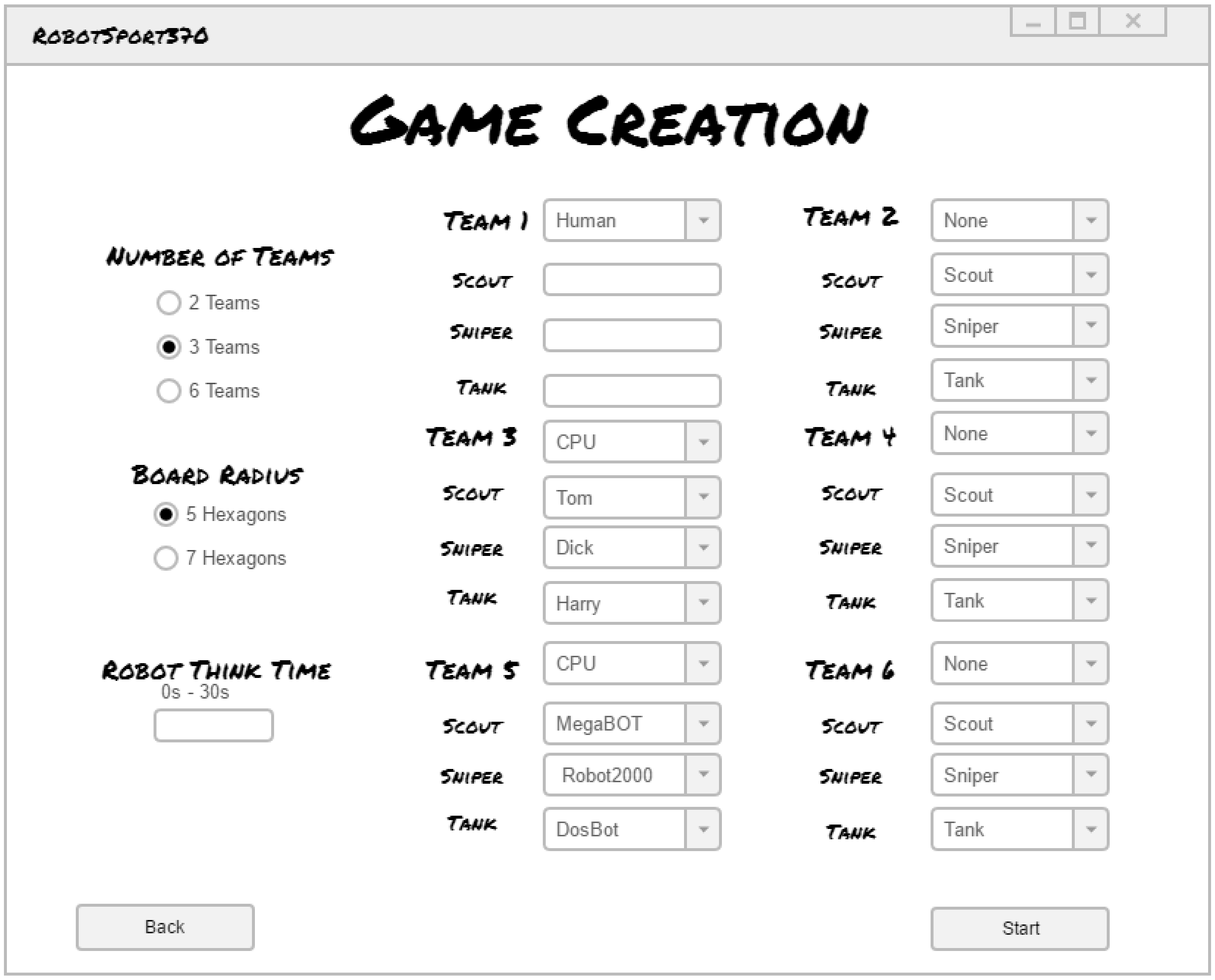
The program will start with the Main Menu (Figure 1) and can branch off to other screens such as Create Game (Figure 2), Rules (Figure 3), Stats (Figure 4), Edit-bots screen (Figure 5), In-game screen (Figure 6), an end game screen (Figure 7), and lastly a RoboManager screen (Figure 8).

Figure 1: Main Menu



The Main Menu is the first screen the user will see when they run RoboSport 370. It will be the main branch for which the user can branch off of. There are five options that are given: Create Game, Rules, Stats, RoboManager and Quit. There is nothing particularly special about any of these action; Create Game, Rules, Stats, and RoboManager brings up a new screen while Quit will close the program. Remember that the user will always have a way to come back to the main menu from any screen or state they are in. The five options the user has are all recorded with buttons. There is group identification in the bottom right-hand corner as well as a title to complete the main menu screen.

Figure 2: Create Game Screen



RobotSport370

GAME CREATION

Number of Teams: 2 Teams [✓ 3 Teams] 6 Teams

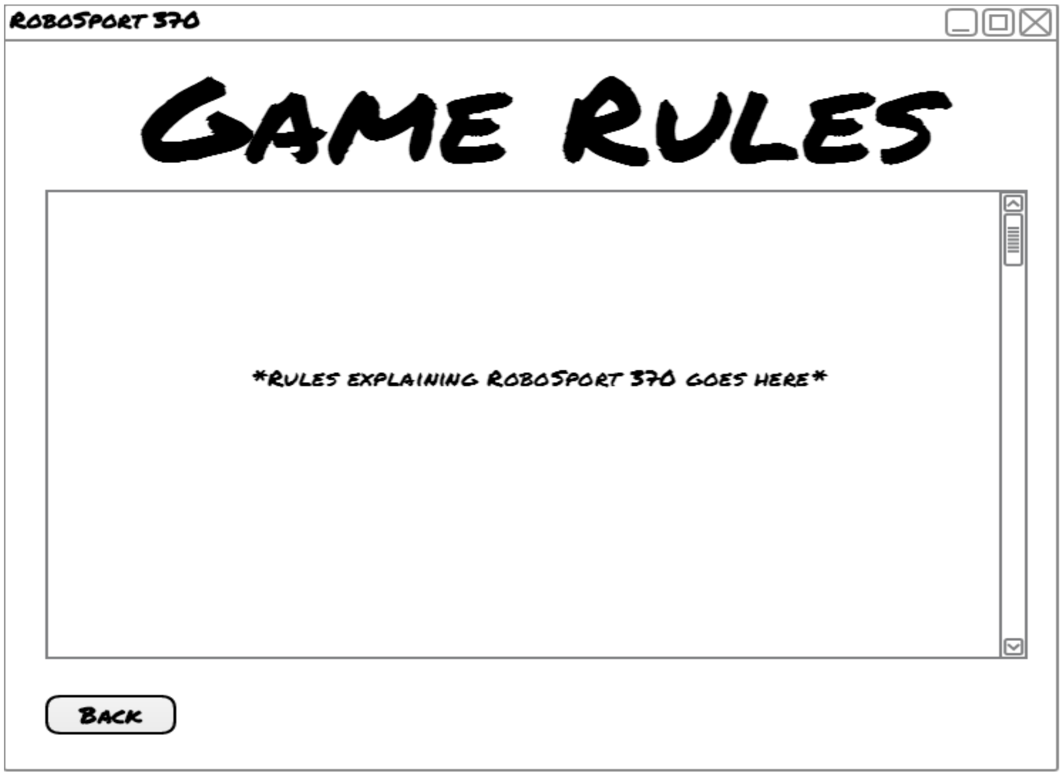
Board Radius: [✓ 5 Hexagons] 7 Hexagons

Robot Think Time 0s - 30s:

Team 1	Human	Team 2	None
Scout		Scout	Scout
Sniper		Sniper	Sniper
Tank		Tank	Tank
Team 3	CPU	Team 4	None
Scout	Tom	Scout	Scout
Sniper	Dick	Sniper	Sniper
Tank	Harry	Tank	Tank
Team 5	CPU	Team 6	None
Scout	MegaBOT	Scout	Scout
Sniper	Robot2000	Sniper	Sniper
Tank	DosBot	Tank	Tank

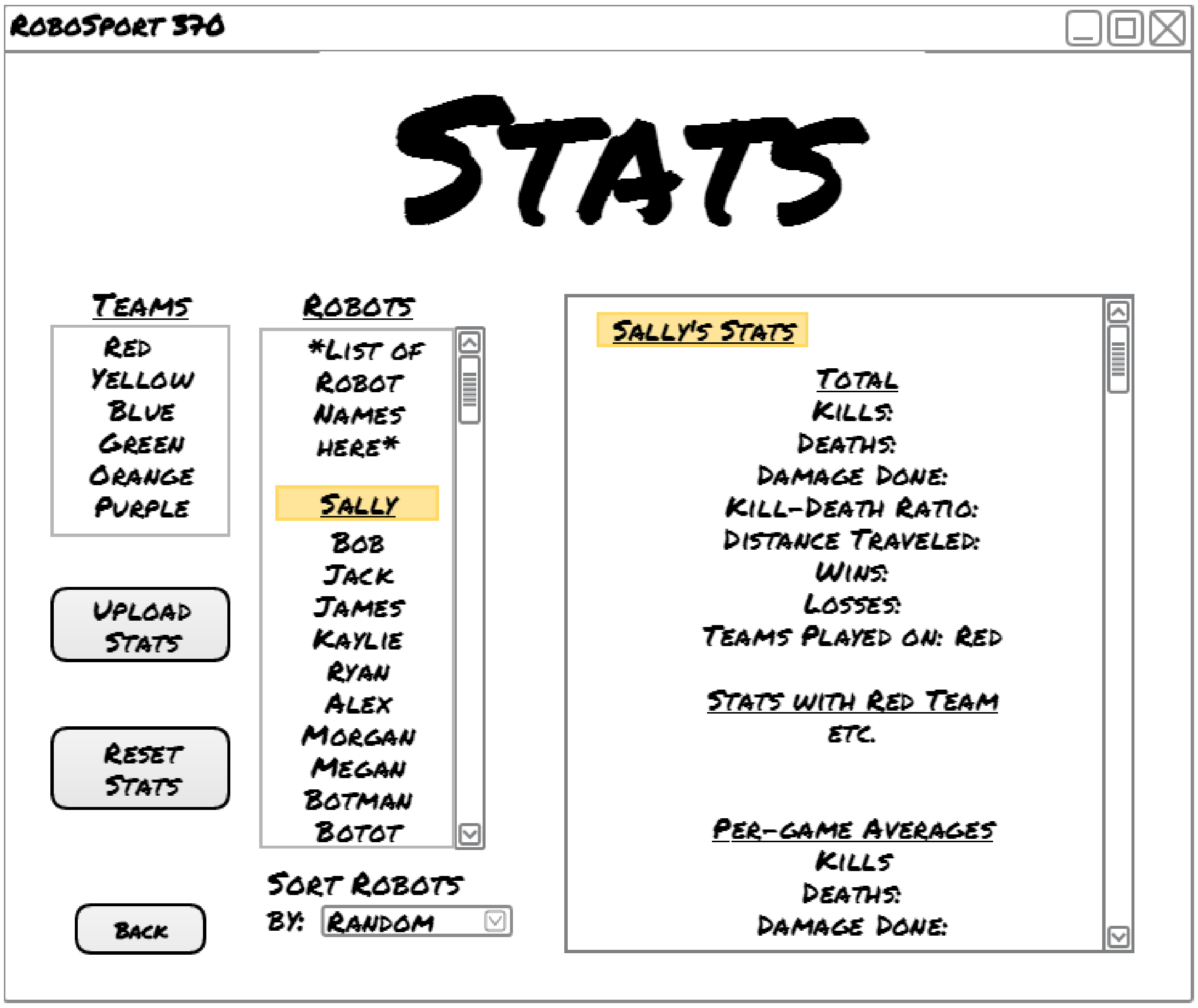
The Create Game screen can be brought up via the main menu (Figure 1), and the end-game menu (Figure 7). The user has the option to choose the number of teams playing; either 2, 3, or 6 teams can be selected as a choice using radio buttons. The number of teams selected effects how many teams can be edited and determines the size/s of board radius selected. If two teams are selected, then only the "Team 1" and "Team 2" options will be able to changed and a 5 hexagon radius will be auto-selected. If 3 teams are selected then only the "Team 1", "Team 2", and "Team 3" will be used and randomly assigned; a radius of 5 will be auto-selected but the user can choose either 5 or 7. Finally if the use chooses "6 teams" then all six times will be editable as our mock-up depicts it. There is a drop down list for each team where the user can choose if it's controlled by a human player or by A.I. If A.I. is selected for the team, then each unit (Scout, Sniper, Tank) can have its own A.I. chosen through a dropdown menu. No two robots can have the same A.I. In addition, an option to change the length of robot turns (0-30 seconds). To start the game, the operator will simply press the start button.

Figure 3: Game Rules Screen



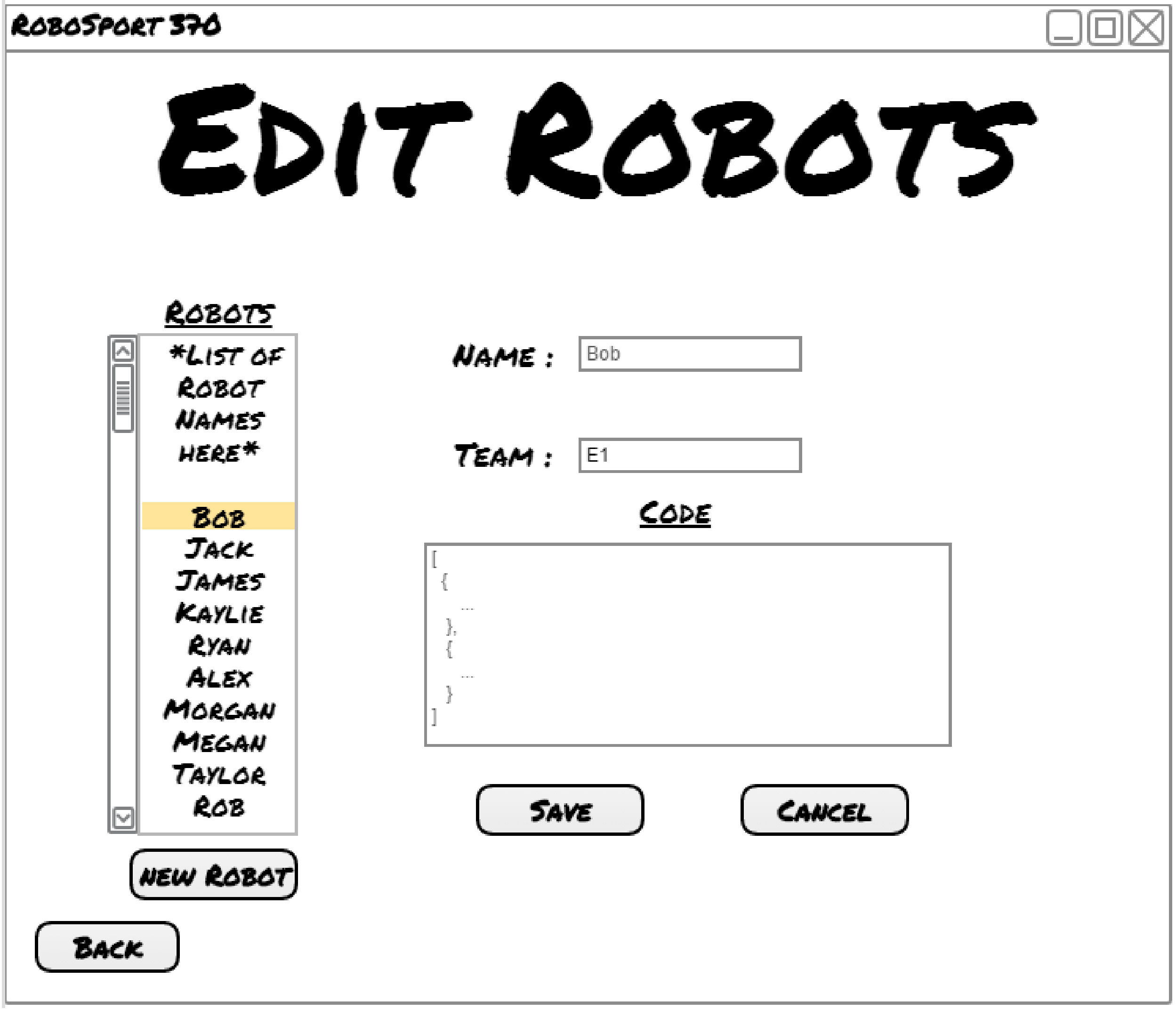
The Game Rules Screen has only one purpose; that is to display the rules of RoboSport 370 in a clean and organized looking scroll box. The user only has the options to scroll through the rules, or to press the back button and return to the main menu (figure 1) or the game screen (figure 6) depending on the previous state.

Figure 4: Stats Screen



The Stats screen can only be brought up from a button in the main menu (Figure 1). The goal of the screen is to display stats from all known robots in a meaningful matter. They will be able to upload each robot's data to the server (Moodle) as well as reset them completely. The list of robots is inside a scrolling text box to help contain them and will have a sorting option to easily find a specific bot. Stats will be displayed as well in a scrolling text box that distinctly shows which robots stats is being displayed. The user has the option to upload or reset the selected robot’s stats, or they can return back to the main menu (figure 1)

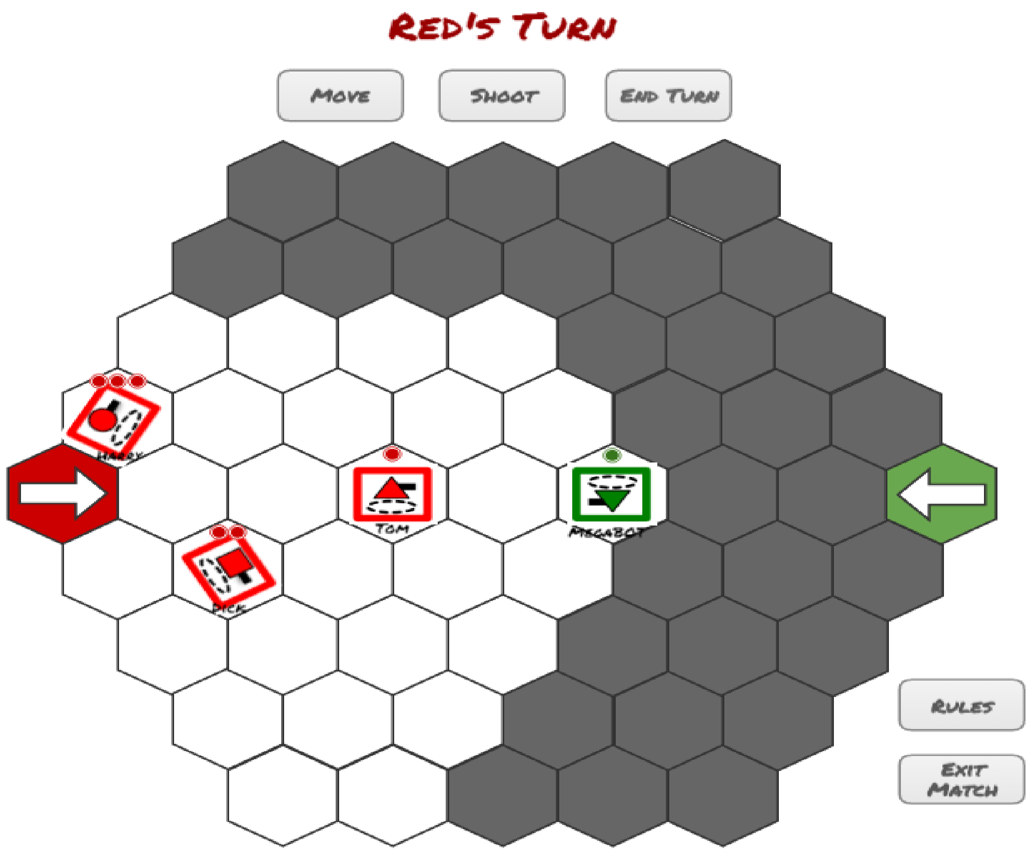
Figure 5: Edit-Bots Screen



```
[
 {
   ...
 },
 {
   ...
 }
]
```

The Edit Bots screen allows the operator to add or remove robots. The only way to enter this screen is through the RoboManger (figure 8) by pressing the "Edit" button. The operator can leave this screen at any time using the back button, returning them to the RoboManager (figure 8). The user can select a robot from the list to edit them; editing in this case includes changing its name, team name, or its Forth code. Likewise, they can create a new robot by hitting the “Create” button and set the three input fields to their likings. The three input fields are as said earlier, name, team, and code. When editing or create a new robot there is a save button or a cancel button to handle the data. Selecting a robot should autofill these fields to their current state and selecting a different robot while creating or editing a current one will “cancel” anything done to the existing or new robot.

Figure 6.1: Player Point of View

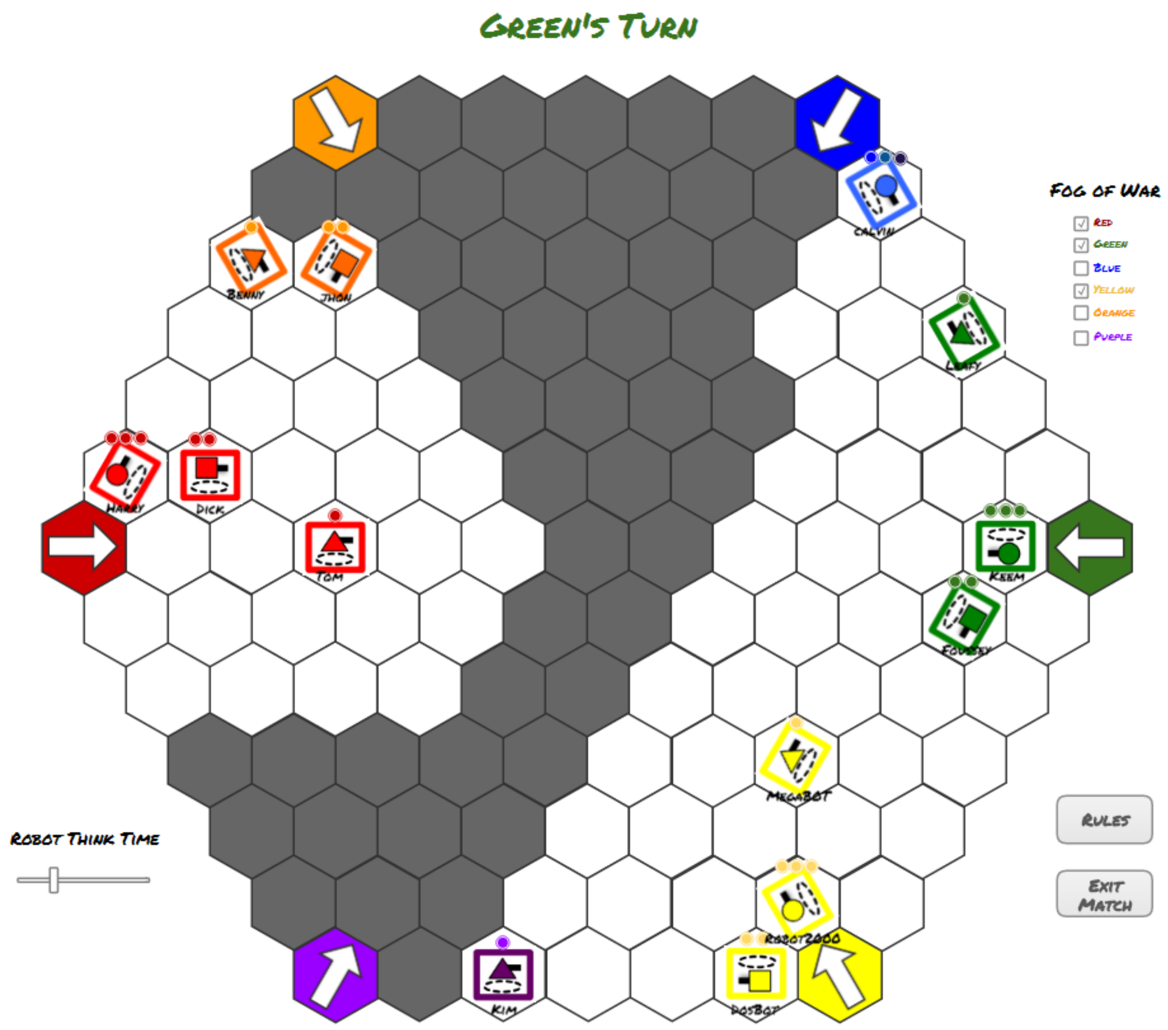


The player is shown the game board of size 5 or 7, depending the choice of the player or the number of teams in the game. White tiles represent spaces that the player's robots can see and grey tiles represent spaces that the player's robots cannot see. All enemies on grey tiles will be hidden from the player. As the player moves robots around the game board, tiles will switch from grey to white and vise versa. Revealing enemies as needed. All visible robots have 1, 2, or 3 dots above them, these dots represent the robot’s health. The dots are will turn black when depleted and remain coloured while available. Each robot also has a name tag, the same name that was chosen in the game creation screen. At the top of the screen there is a team turn indicator. This displays which team is currently being manipulated.

Players has five buttons at their disposal on the screen: a "Move" button, a "Shoot" button, an "End Turn" button, a "Rules" button, and an "Exit Match" button.

1. The "Move" button will tell the robot to move forward.
2. The "Shoot" button will tell the robot to shoot at the selected tile.
3. The "End Turn" button will end that players turn and move onto the next teams turn.
4. The "Rules" button will show the Rules Menu Screen (Figure: 3).
5. The "Exit Match" button will leave the match without finishing it, and display the Main Menu Screen (Figure: 1).

Figure 6.2: Observer Point of View



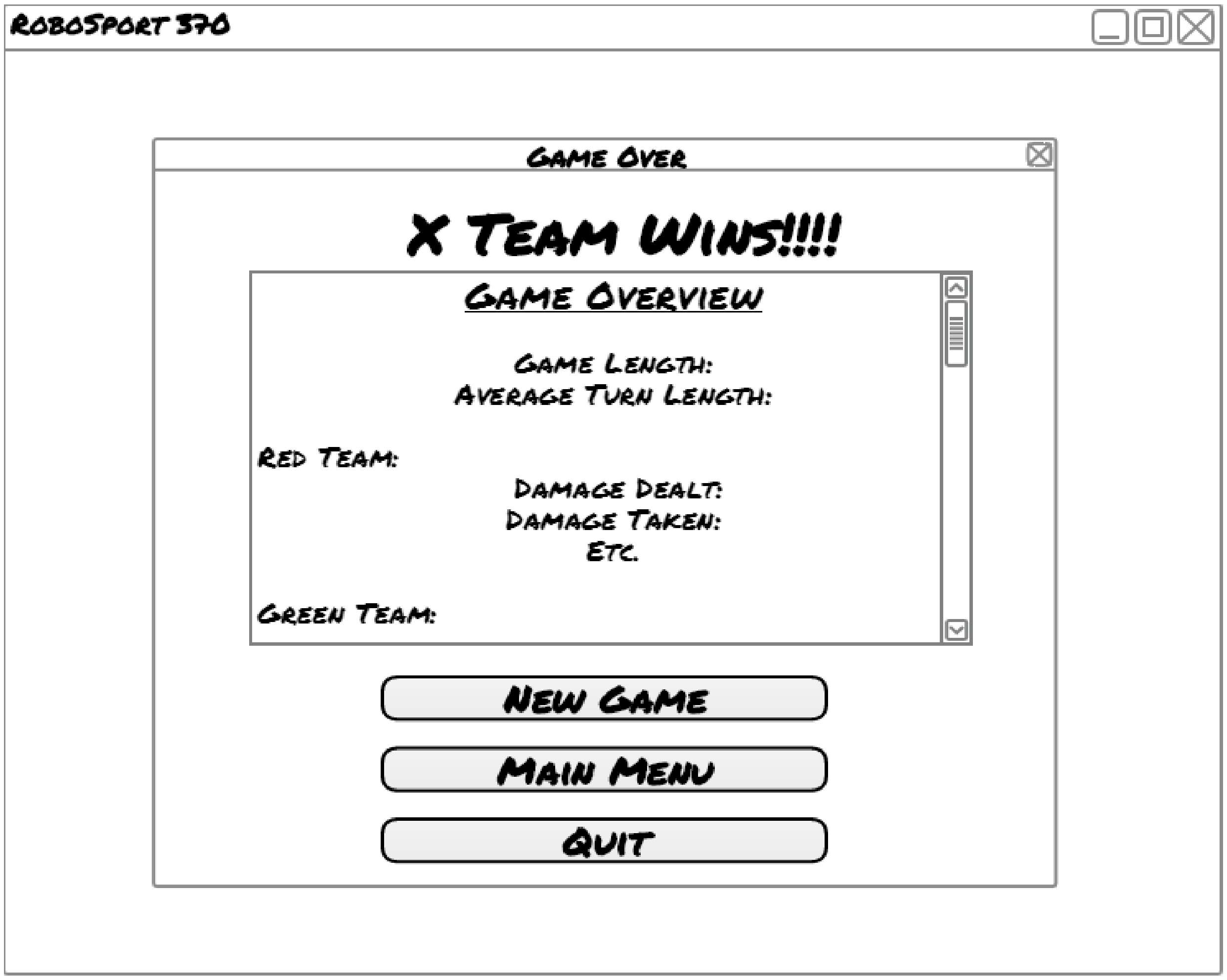
Fog of War

- ✓ Red
- ✓ Green
- ☐ Blue
- ✓ Yellow
- ☐ Orange
- ☐ Purple

The observer is shown the game board of size 5 or 7, depending the choice of the observer or the number of teams in the game. White tiles represent spaces that the A.I.'s robots can see and grey tiles represent spaces that the A.I.'s robots cannot see. All visible robots have 1, 2, or 3 dots above them, these dots represent the robot's health. The dots are will turn black when depleted and remain coloured while available. Each robot also has a name tag, the same name that was chosen in the game creation screen. At the top of the screen there is a team turn indicator. This displays which team is currently being manipulated. The observer has 6 "Fog of

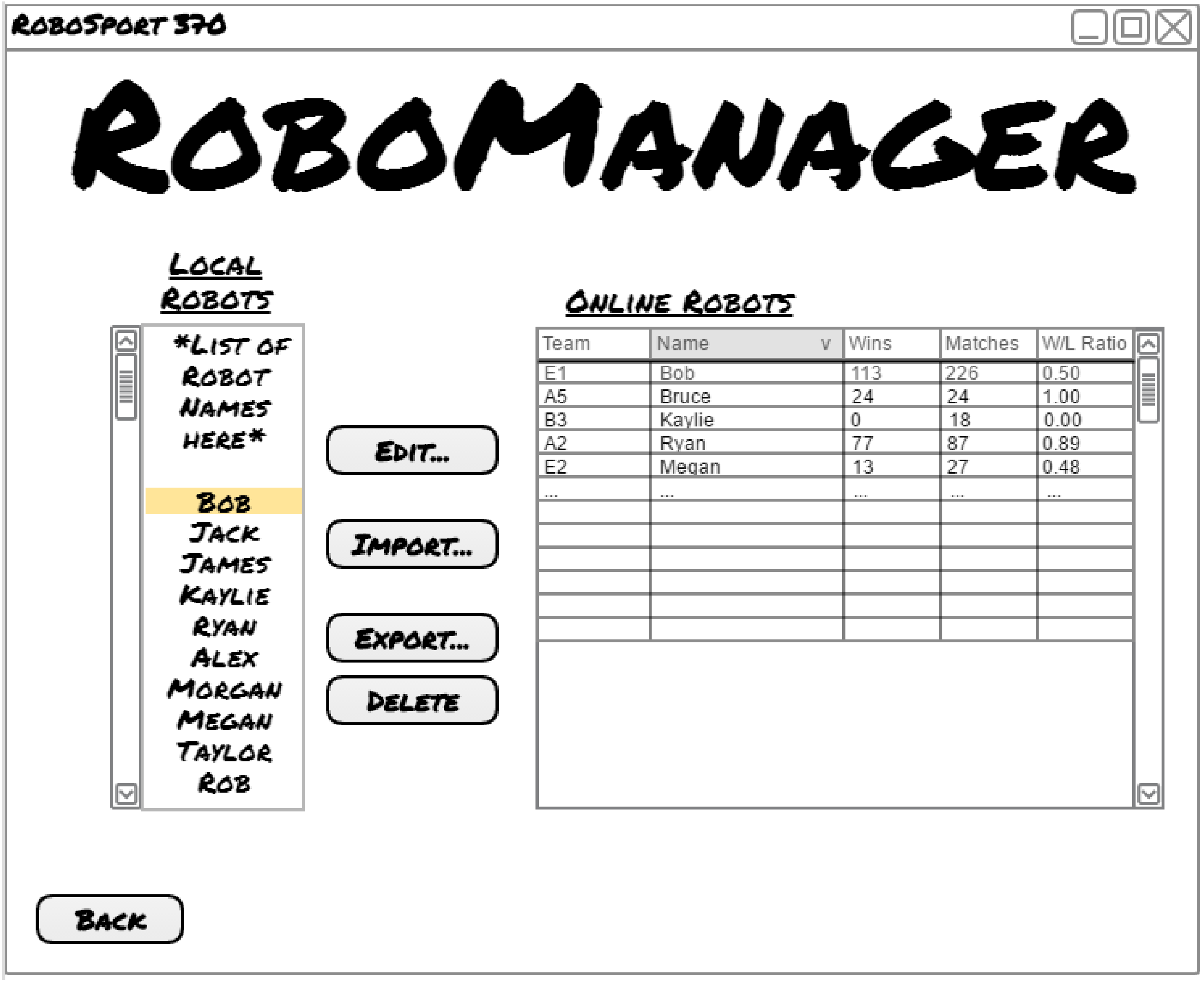
War" check boxes, a "Robot Turn Time" slider, and two buttons; a "Rules" button, and an "Exit Match" button. The "Fog of War" check boxes are colour coded and labeled for each team. When the "Red" box is checked you will be able to see all the tiles that the red team is able to see, the same goes for the rest of the checkboxes. The "Robot Turn Timer" slider controls how long the robots will take to complete their turn. If slider is all the way to the left, then the robots' turn will be complete almost instantly. The more the slider is to the right the longer the turns will take. The "Rules" button will show the Rules Menu Screen (Figure: 3). The "Exit Match" button will leave the match without finishing it, and display the Main Menu Screen (Figure: 1).

Figure 7: End-Game Screen:



The End-Game Screen is a pop-up screen that is displayed over Figure 6 (In-game screen). It occurs in the instance a game finishes and disappears when the user selects one of their three options. The screen is to display the winning team at the top as well as various stats for teams who participated in a scrolling text window. The user then has the option to create a new game which takes the user to Figure 2, return to the main menu which is Figure 1, or quit the program entirely. If the user tries to exit out the pop-up window, they will be taken to Figure 1, the main menu.

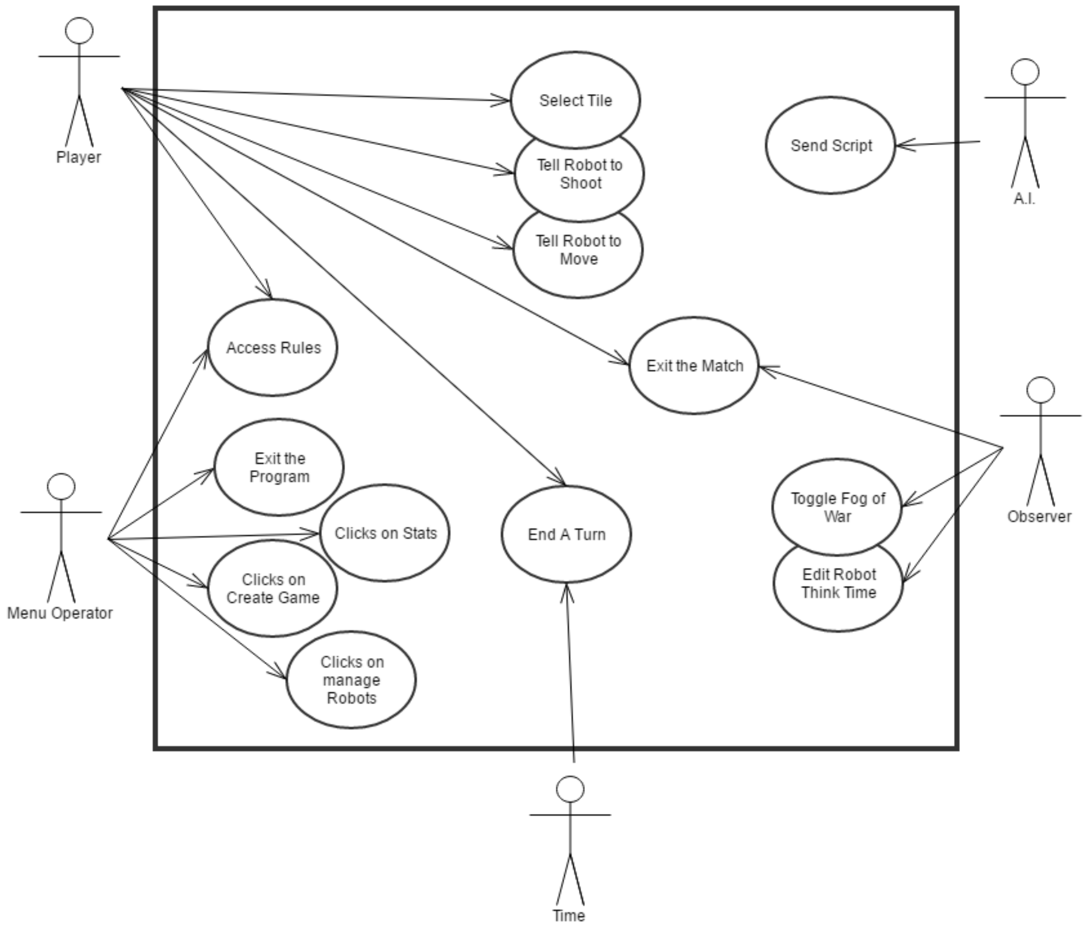
Figure 8: RoboManager



Team	Name v	Wins	Matches	W/L Ratio
E1	Bob	113	226	0.50
A5	Bruce	24	24	1.00
B3	Kaylie	0	18	0.00
A2	Ryan	77	87	0.89
E2	Megan	13	27	0.48
...	...	...	...	...

RoboManager can only be reached from the main menu. (Figure 1.) On the left-hand side it will display a list of all robots known to the local system, and on the right is a table query of robots stored online. In-between the list and table are buttons to interact with all the robots; edit takes the user to the Edit Robots screen, (Figure 5.) if there is a local robot selected then it will be selected in the new screen as well. An import button will take a selected robot from the online table and download to the local system. The export button will take a selected robot from the local list and upload it to the online table. Selecting a local robot and pressing the delete robot will remove that robot from the list. Lastly, the user can always use the back button to return to the main menu.

Figure 9: System Diagram



Section 5: Actors and Actions

Player is defined to be any human giving input to the game board or end game screen.

Player actions:

1. Select a tile

a) Pre-conditions: None.

b) Flow of Events: Click on a tile, the robot will turn toward that tile.

c) Post-conditions: The tile is selected and the unit is facing in the direction of the selected tile.

d) Error-conditions: None.

2. Command robot to shoot

a) Pre-conditions: A tile is selected and the selected tile is in the unit's range.

b) Flow of Events: Clicked the "Shoot" button and a shot is fired toward that tile.

c) Post-conditions: Damage is dealt to all units on the selected tile.

d) Error-conditions: None.

3) Command robot to move

a) Pre-conditions: The unit must have moves left and the tile that unit is trying to move to is on the board.

b) Flow of Events: Clicked the "Move" button and the robot moves forward 1 tile. Click different tiles to change direction of the robot.

c) Post-conditions: Unit moves 1 space in the direction that it is facing

d) Error-conditions: None

4) Access rules window.

a) Pre-conditions: None.

b) Flow of Events: Clicked on the “Rules” button. The rules screen is displayed.

c) Post-conditions: Displays a screen of the rules to the user.

d) Error-conditions: None.

5) Exit the match

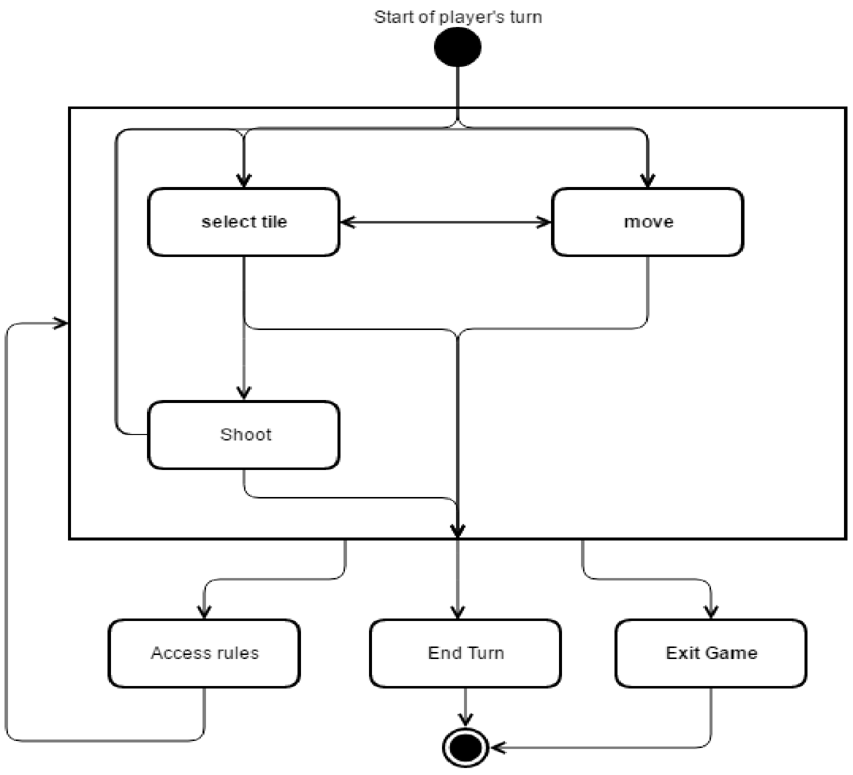
a) Pre-conditions: None.

b) Flow of Events: Clicked on the “Exit Match” button. The match is ended without finishing and no data is recorded.

c) Post-conditions: The match ends and the user returns to the main menu.

d) Error-conditions:

Figure 10. Overall flow of the player’s actions.



Menu Operator is defined to be any human giving input to the menu, rules, or stats screen.

Menu Operator Actions:

1) Create a game

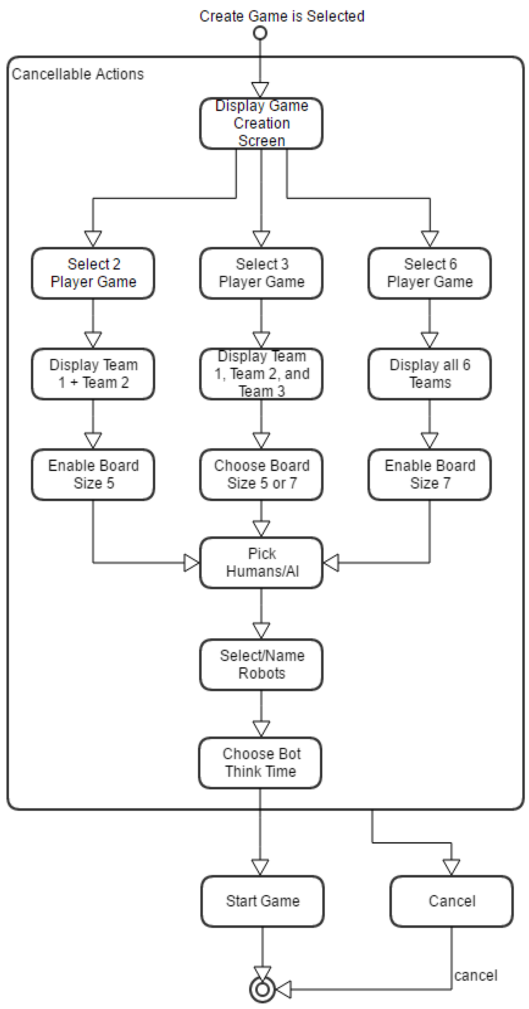
a) Pre-conditions: None.

b) Flow of Events: When create game is selected the game creation screen is displayed containing a choice for the number of players, a list of the teams with a choice to name/choose the robots for each team, as well as a choice for the think time for the A.I. The menu operator starts by selecting the number of players, and the extra teams are grayed out and cannot be used. The menu operator then chooses which teams will be computer controlled, and which teams will be humans. If the team is chosen to be a human player the menu operator can then input the names for each robot on the human's team. If the team is chosen to be a A.I. player, the menu operator can then select from a list of programed robots to be used for each robot (Scout, Sniper, Tank) on the A.I.'s team. The menu operator also has the choice to edit the CPU's think time, which determines how long an CPU will take to perform their turn. The menu operator then selects to either start the game or cancel all these actions and return to the main menu.

c) Post-conditions: A game is created with the options given by the menu operator.

d) Error conditions: None.

Figure 11: Game creation sequence diagram.



2) Access rules

a) Preconditions: None.

b) Flow of Events: As soon as the rules are selected the rules page will be displayed on the screen showing a list of all the rules for the game. The rules remain on the screen until the menu operator chooses to return to the main menu.

c) Post-conditions: The rules screen is displayed.

d) Error conditions: None.

3) View stats

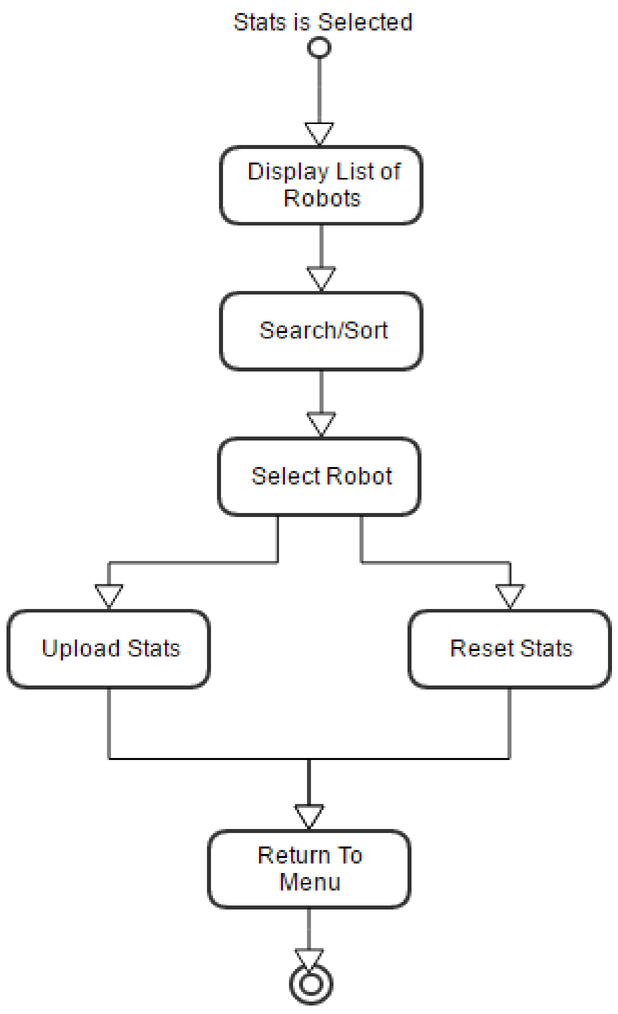
a) Pre-conditions: None.

b) Flow of Events: Menu Operator clicks on the “Stats” button from the Main Menu screen. (Figure 1.) The Stats screen is displayed. (Figure 4.) The menu operator can now sort and search stats of specific robots.

c) Post-conditions: The stats screen is displayed.

d) Error-conditions: None.

Figure 12: Stats screen sequence diagram



4) Exit the game

a) Pre-conditions: None.

b) Flow of Events: Clicking on “Exit Game” button will close the program.

c) Post-conditions: The program closes successfully.

d) Error-conditions: None.

5) Manage robots

a) Pre-conditions: The selected robot must exist.

b) Flow of Events: Clicking on “Manage Robots” button will display the Manage Robots screen. (Figure 8.) from there the menu operator can download, upload and delete robots. They can also click the “Edit Robots” button. This will display the Edit Robots screen. (Figure 5.)

c) Post-conditions: Changes were made to the selected robot.

d) Error-conditions: Syntax errors.

Figure 13: Robot manager sequence diagram

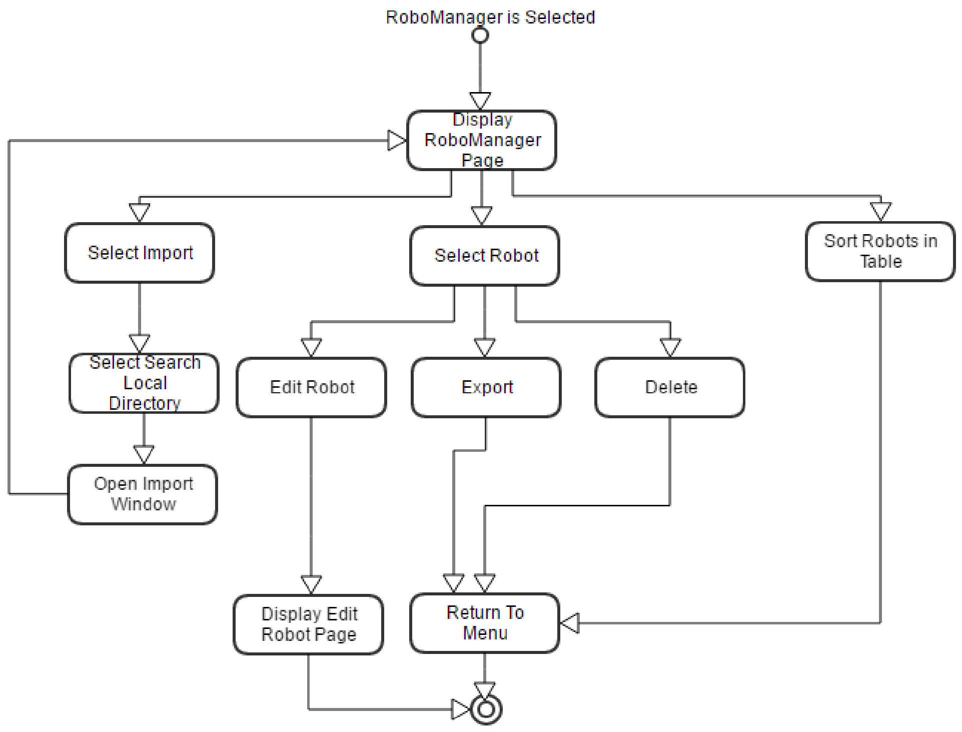
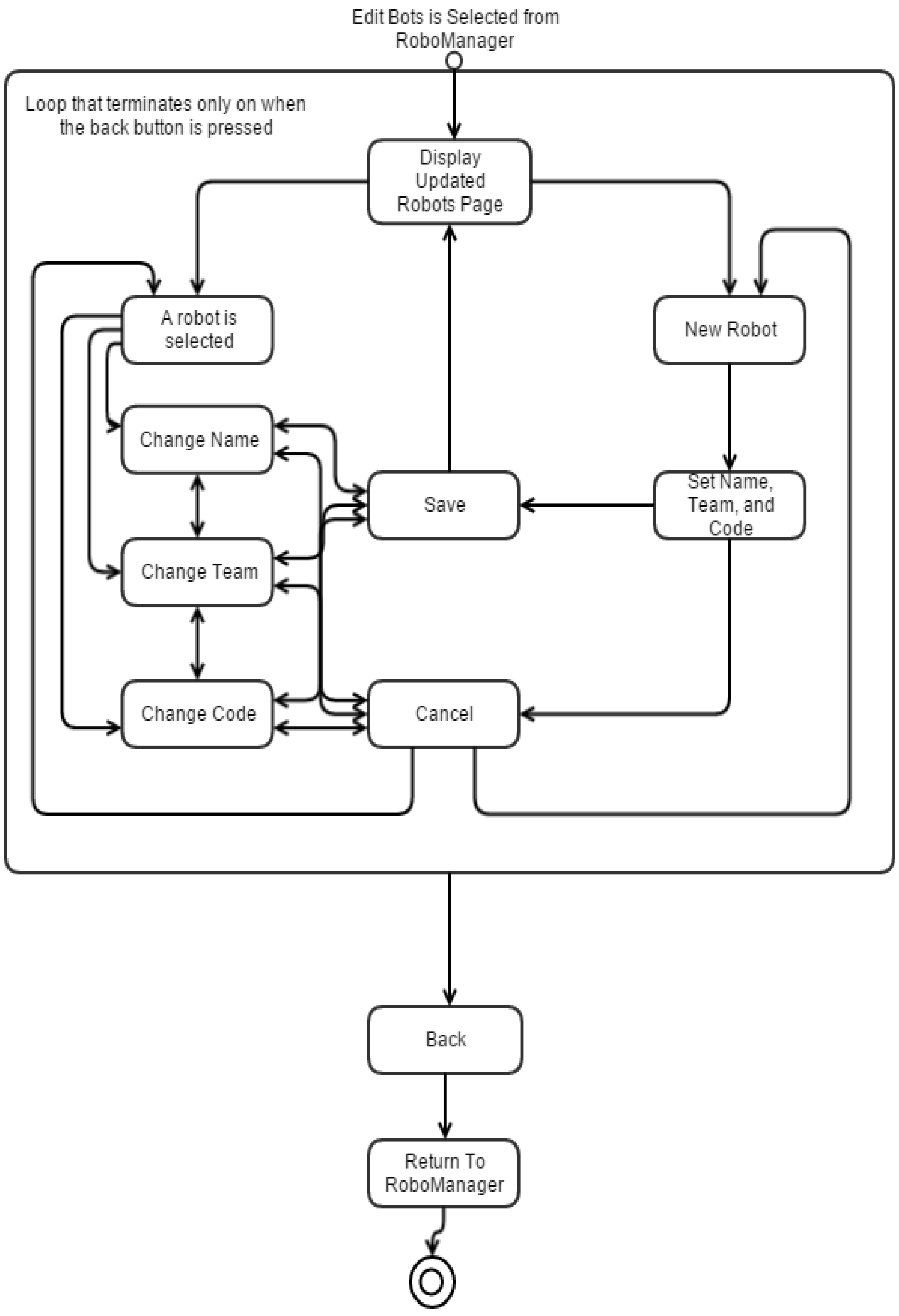


Figure 14: Edit Robots sequence diagram


Edit Bots is Selected from RoboManager
Loop that terminates only on when the back button is pressed
Display Updated Robots Page
A robot is selected
New Robot
Change Name
Save
Set Name, Team, and Code
Change Team
Change Code
Cancel
Back
Return To RoboManager

Observer is defined a human watching only AI's play the match and giving input to the game board or endgame screen.

Observer actions:

1) Toggle fog of war

a) Pre-conditions: None.

b) Flow of Events: The observer clicks on a “Fog of War” box and it will be toggled on or off. This will change the view of the board revealing or hiding tiles for each team.

c) Post-conditions: Fog of war is toggled on or off for the observer.

d) Error-conditions: None.

2) Exit the match

a) Pre-conditions: None.

b) Flow of Events: At any time the observer can click the “Exit Match” button and the match will be terminated without recording stats.

c) Post-conditions: Match ends and the observer returns to the menu.

d) Error-conditions: None.

3) Edit robot think time

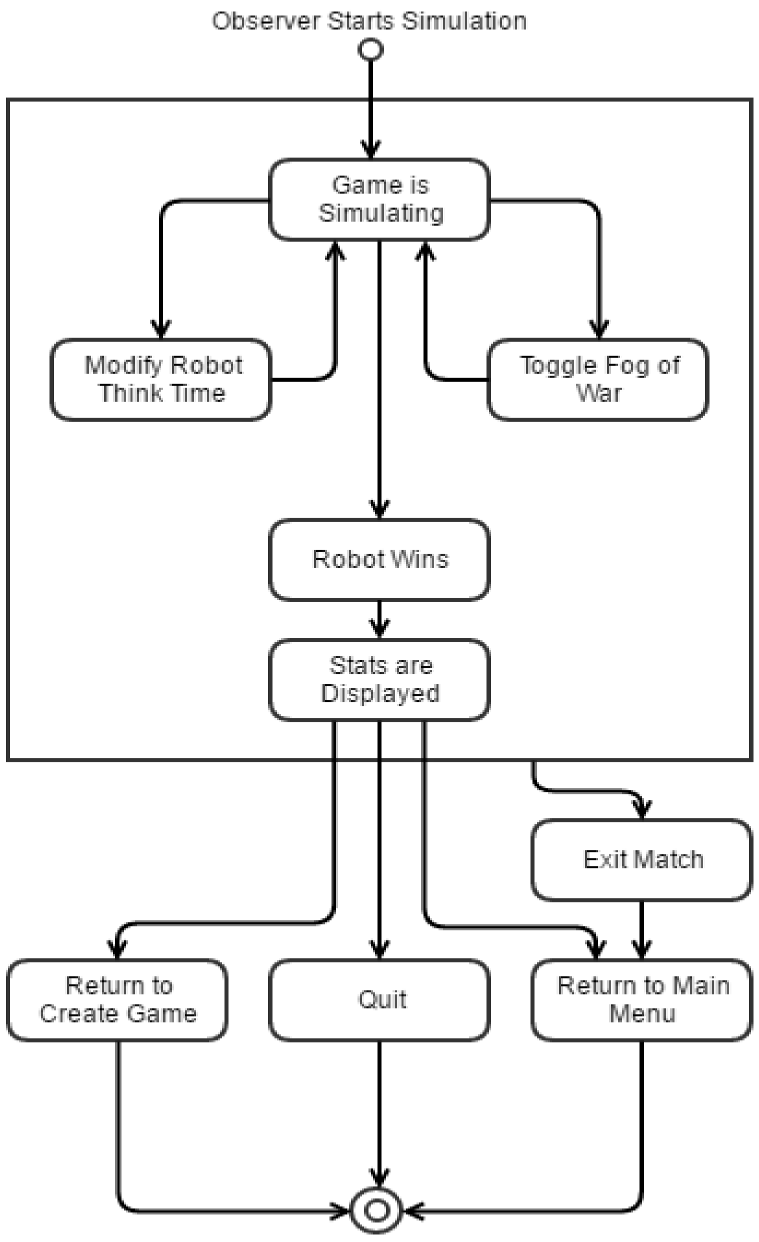
a) Pre-conditions: None.

b) Flow of Events: The observer can slide the “Robot Think Time” slider to change the time it takes the robot to perform its turn. Left is fast and right is slow.

c) Post-conditions: Robot think time changes.

d) Error-conditions: None.

Figure 15: Observer sequence diagram



Time is defined to be a non-human entity that keeps track of time for different parts of the game.

Time actions:

1) End turn

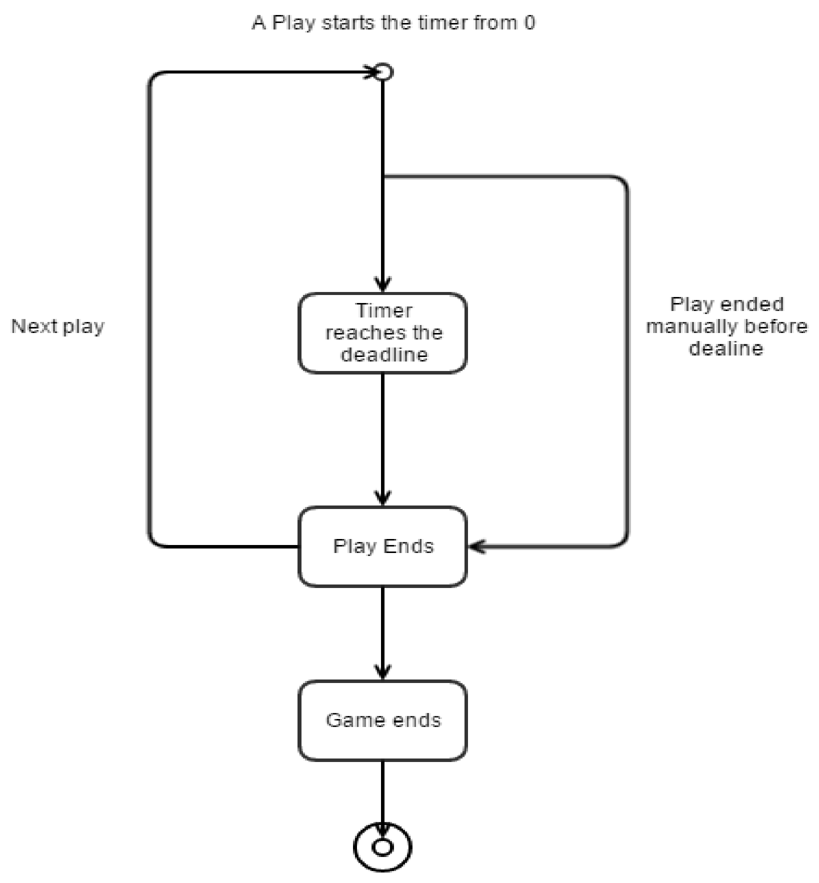
a) Pre-conditions: The game must be on a human players turn.

b) Flow of Events: On a human's play if 30 seconds pass then end the play ends.

c) Post-conditions: Timer ends the human's play.

d) Error-conditions: None.

Figure 16: Timer sequence diagram


A Play starts the timer from 0
Next play
Timer reaches the deadline
Play ended manually before dealine
Play Ends
Game ends

A.I. is defined to be a non-human that is playing the game through an interpreter.

A.I. actions:

1) Send script

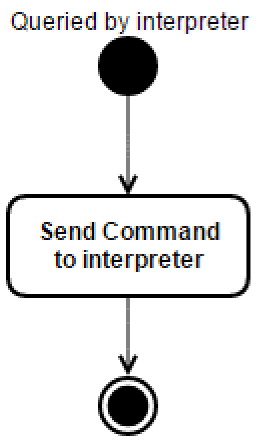
a) Pre-conditions: None.

b) Flow of Events: The A.I. when prompted will give an action to the robot.

c) Post-conditions: Robot receives and follows commands from A.I.

d) Error-conditions: Syntax errors.

Figure 17: A.I. sequence diagram



Section 6: Sub-Systems

Robot-Librarian

The Robot-Librarian is sub-system that can upload local statistics and robots to the online database. It will also be able to download robots from the online database to a local system. Robot-Librarian will query for robot teams by team, score and name to make it easy to search. It will show all robots stored in the database by specific fields. On top of all of this, it will be able to register a new robot, replace the code of an existing robot, and freeze a robot to reuse its name.

Interpreter

The interpreter essentially acts as a means of communication between the system and the A.I. The interpreter is notified from the system that a particular robot needs to make a play. The interpreter sends a command to the A.I. which asks it to make a few decisions; the interpreter wants to know where the robot will move in a turn, and where it will shoot. The A.I. will then make its decision and send its result back to the interpreter. The interpreter will receive a sequence of commands of moving or shooting once depending on what type of unit the robot is. The interpreter will then communicate with the system and command the robot from the information given by the A.I. The interpreter will wait forever until the system notifies that it needs a command for another robot play.

Section 7: Scenarios

Menu Operator

The menu operator actor has 6 actions, that is to select the rules, select the stats, create a game, manage the bots, or to exit the game.

Scenario 1: Rules Is Selected

1. The rules are displayed on the screen.
2. The menu operator reads the rules.
3. The menu operator returns to the Main Menu.

As soon as the rules are selected the rules page will be displayed on the screen showing a list of all the rules for the game. The rules remain on the screen until the menu operator chooses to return to the main menu.

Scenario 2: Game creation

1. The game creation screen is displayed.
2. The Menu Operator selects 2, 3, or 6 players for a game.
3. Board size 5 or 7 is displayed.
4. The menu operator chooses one human player and/or AI players.
5. The menu operator selects three robots for each AI team.
6. The menu operator gives his/her robots names.
7. The bot think time is selected.
8. The game is started.

See Figure 11

Scenario 3: The Menu Operator uploads and deletes the stats for a robot.

1. The list of robots is displayed.
2. The menu operator chooses to sort the robots A-Z.
3. The menu operator selects a robot.
4. The menu operator reads the stats for the robot in the box.
5. The menu operator uploads the stats for the selected robot.
6. The menu operator deletes the stats for the selected robot.
7. The menu operator returns to the menu.

When the stats are selected the menu creator is taken to the stats screen. On this screen a list of robots is displayed, along with an option to sort the robots, an option to upload the stats of a robot, an option to reset the stats of a robot, a window to display the stats of the robot, an option to return to the main menu, as well as an option to sort the robots by team. The menu operator can choose to sort the robots or leave them as they are initially sorted. The menu operator can then select a robot from the list which will automatically display the stats for the robot in the window. The menu operator can then choose to upload the starts, or reset the stats for the selected robot. When they are done viewing the stats for the robots the menu operator can return to the main menu.

See Figure 12

Scenario 4: Menu operator manages robots

1. RoboManager is selected from the Main Menu.
2. Operator selects a robot from the local or online robot list.
3. The operator can then import, export, or delete the selected robot depending on where the robot is located

See Figure 13

Scenario 5: Operator Exits the program

1. Operator selects “Quit” button, program then closes

Scenario 6: Editing/Creating local robots

1. Operator selects “RoboManager” button
2. Operator then selects “Edit Robots” button
3. The user can select a local robot from the given list or create new
4. They have the option to set its name, team, or code

See Figure 14

Player Scenarios

Scenario 1: Player selects a tile

1. Player clicks with their mouse a tile on the game board.
2. Robot turns in the direction of the tile

See Figure 10

Scenario 2: Player shoots a tile

1. Player selects a tile
2. Player presses the “shoot” button

See Figure 10

Scenario 3: Player moves to a tile

1. Player selects a tile
2. Player presses the “move” button

*See Figure 10

Scenario 4: Player ends a Play

1. Player presses the “End play” button

See Figure 10

Scenario 5: Player exits the match

1. User presses the “exit match” button
2. Sends user back to main menu

See Figure 10

Scenario 6: Player checks the game rules

1. Player presses the “Rules” button in game

2. The user is taken to Figure 3 (contains the game rules)
3. User presses the “back” button to return to the game

See Figures 3 and 10

<u>Observer Scenarios</u>

Scenario 1: Observer Exits a match

1. User presses the “exit match” button
2. Sends user back to main menu

See Figure 15

Scenario 2: Observer edits the robot think time

1. Observer selects the “Think time” slide
2. Moving the slider left increase robot think time
3. Moving the slider right decreases robot think time

See Figure 15

Scenario 3: Observer toggles the fog of war for teams

1. Observer selects one of the 6 Fog of War check boxes
2. A checked box means that the observer can see everything Team X sees

See Figure 15

<u>Time Scenarios</u>

Scenario 1: Time ends a Play

1. A play begins, resetting time
2. After 30s the Timer ends a play if not already ended manually

See Figure 16

A.I. Scenarios

Scenario 1: Send a script

1. Receivers prompt from interpreter to command a robot
2. Makes a decision on where to move the robot and where to shoot
3. Sends decision back to interpreter

See Figure 17

Section 8: Use-Cases

Menu Operator

1) Rules is Selected
   a) Read and scroll through rules
   b) Return to main menu
2) Stats is Selected
   a) Go back to main menu
   b) Display all robots
   c) Search / Sort
   d) Select a robot
      i. Upload/reset stats
3) Create
   a) 2 Players
      i) Display Team 1 and Team 2
      ii) Enable board size 5 only
   b) 3 Players
      i) Display Team 1, Team 2, and Team 3
      ii) Enable board size 5 and 7 option
   c) 6 Players
      i) Display all 6 teams
      ii) Enable board size 7 only

d) Set a team to AI controlled

    i) Set scout, tank, or sniper to an existing local robot

e) Set a team to human controlled
    i) Change name of scout, tank, or sniper

f) Edit bot think time

4) RoboManager

a) Export Robot
    i) Select local robot
    ii) Press "export"

b) Import Robot
    i) Select robot from Moodle server
    ii) Press "import"

d) Delete robot
    i) Select robot
    ii) Confirm deletion

e) Edit Robots
    i) Select existing robot
        i. Change name
        ii. Change team
        iii. Change code
        iv. Save
        v. Cancel
    ii) Create new robot
        i. Set Name
        ii. Set Team
        iii. Set Code
        iv. Save
        v. Cancel

5) Exit is Selected

a) Exit

<u>Observer</u>

1) Exit

2) Edit bot think time

        a) 0 auto completes plays

    3) Toggle fog of war
        a) Up to 6 check boxes
        b) Show all

A.I.

1) Get next command

Player

1) Select Tile
    a) Shoot
    b) move

2) Rules

3) End play

4) Exit Game

Time

1) End turn

Section 9: Executive Summary

RoboSport370 is a turn-based strategy game in which teams compete to be the first to destroy all tanks on enemy teams. The game can be played with any combination of human and AI players, with teams of either 2, 3 or 6 and played on a board of size 5 or 7. Each player begins on a tile pre-selected by the game, and has three units they can use which each have different properties. Players can move and shoot within the allowed ranges for each of their individual robots. The players are randomly assigned colors and red team is always the first to move and the player to the left goes next. Each player moves the tank with the highest movement on their team until all the players have gone, moving onto the next tank. A game is ended when only one team has tanks remaining.

The game will work single player or multi-player, but will not work over a network and so it must be played on one device and so our game will facilitate a switch between players so that they do not see anything that they are not supposed to see. We plan to use a model View Controller architecture to facilitate heavy user input, and our game will be designed for the computers in the Spinks lab at the University of Saskatchewan using Java/AspectJ, and forth for our AI as well as a Robot-librarian to manage our robots and an Interpreter for their forth code to Java. Our stakeholder is interested in our design process and documentation as it is a crucial part of software engineering to actually design your program before you begin to create.

The program will have various screens to display depending on the scenario, including the Main Menu, Create Game, Rules, Stats, Edit-Bots, In-Game, RoboManager, and end game screens. Once the game is started the player is shown a game board of the selected board size. The tiles on the game board are two different colors depending on what the player can see. White tiles represent the combined range of all the players' tanks, and the grayed out tiles represent the rest of the board which the player cannot see. If there is no human player in the game, then the player becomes the observer. The observer is shown a full game board and can select which team's perspective they see the game from.