I. Introduction

This project will be a simulator that will simulate battles between robots created by other players all on a virtual map with a set of rules that must be followed. The project will contain multiple parts that will all have different requirements and restrictions. The simulator created will be able to simulate battles with other players robots. The battle will then be displayed graphically on a hexagonal map which spectators will be able to watch and possibly manipulate.

The robots will be designed with a specific standard that will be decided in a standards meeting, this project will include designing a set of robots to participate in other teams simulations. The robots will be able to run on any simulation provided it adheres to the standards laid out in the standards meeting.

II. Background

In September, 2015, our group was commissioned by Professor Christopher Dutchyn to design a program, called RobotSport370, which would simulate battles between player's robots created on a virtual map. We were given certain requirements, important features, 'could-have's, and extras, which we will outline in our scope.

This project will require a design method to be followed to the successful completion of the project. We have begun this project to meet these requirements and practice this design process.

III. Scope

A. Must Have:

1. The software must be able to run on the Universities' Linux machines using TuxWorld. The software must be able to simulate matches between other players and be able to run simulations with robots designed by other teams.
2. A method of controlling the simulation will need to be implemented as well, such as a "pause,play, and rewind" button. The simulations will need to be run on a hexagonal map that will have predefined sizes of (5,7,9, or 11) hexes.
3. A User Interface will be implemented that will allow users to have some simple controls over the simulation and allow the user to see important information. Some sort of graphics will be required, at the very least ASCII graphics will be used, but if time permits more detailed graphics will be made.

B. Should Have:

1. The software should be able to have an option to run a simulation without displaying graphics, this would allow many simulations to be run in a short amount of time.

2. In the user interface having useful information about the simulation to show the spectators would be useful, players could see information such as the amount of time the simulation has been running or information about the robot's current status and team information.

C. Nice to Have

1. The software could have a debug/testing simulation in which certain restrictions would be lifted and edits would be able to be made real-time.

D. Would Like to Have

1. If time permits advanced graphics and sounds would be implemented.

IV. Limitations

A. Issues
B. Questions - Difference Between spectators and testers?
C. Number of teams on map?
D. Clarification on tournament mode?

V. Actors

A. External Robots
B. Spectators
C. Testers

VI. Use Cases

Text and then reference an attached diagrams

A. Spectators: Watching a match, Interact with match/GUI
B. As a spectator I want to watch a matchAs a robot I want to be loaded into the simulator
C. As a robot I want to be able to run my turn
D. As a Tester I want to be able to rewind
E. As a Tester I want to be able to step through instructions
F. As a Tester I want to simulate many matches simultaneously

VII. Activity Diagrams

A. Load robots into simulator

B. Select Map size
C. Select Number Of Teams
D. Select regular mode
E. Spectator watches match
F. Spectator will watch until match is over
G. Display results
H. Loop to beginning

A. Load robots into simulator
B. Select map size
C. Select number of teams
D. Select testing mode
E. Perform tests using testing gui (pause/play, rewind?)
F. Display Results
G. Return to beginning

VIII. Conclusion

Notes

Requirements 1. play,pause, rewind, skip
2. (5,7,9,11) size
3. UI
4. Graphics
5. Testing Interface

Diagrams 1. Use Cases/scenario
2. activity - fancy flow-chart
3. story board

Actors -Robots
-Spectators/UI Users

Use Cases -?

Standards meeting October 2nd

----

5 steps of design

1) Requirements -we are here
2) Architecture 1
3) Architecture 2
4) Coding

5) Documentation

Recipe cards to make class diagrams

-