CMPT 370 D1

DEPLOYMENT REPORT

Dec 3, 2016

Contents

1 INTRODUCTION

This document is designed to give a report on the deployment status of the Robot Game system. It will include a description of unmet requirements, changes in the design of the system, and a description of remaining bugs at the time of release.

1. **TEAM MEMBERS**

Resource Name	**Role**
Dimaano, Nico	Developer / Tester
Neijmeijer, Niklaas	Developer/ Tester
Seidenthal, Kyle	Developer / Tester
Sterma, Brendon	Developer / Tester
Zang, Jiawei	Developer / Tester

2 UNMET REQUIREMENTS

2.1 Communication with the Robot Librarian

We were unable to secure enough time to begin working on communication between the Librarian and our system. This feature was cut because it was the least important feature that we could be spending our time implementing and was not critical to running the game. JSON files that are manually downloaded could still be read using the functions we created.

2.2 Reading and processing AI files

We began implementation on reading the Fourth code from the JSON file. We began implementation later than anticipated because we did not have access to a sample file in order to begin. As a result, the only method we were able to come up with in the timeframe was far too lengthy to implement given the remaining amount of time we were given and was cut in favour of other systems that were more critical and were closer to being finished.

2.3 Reading and writing statistics from JSON files

The code exists to read and write statistics from and to JSON files, but we did not have an opportunity to build a method to update the files and record statistics mid-game. Time was a factor and the function was not deemed important enough to prioritise.

3 DESIGN CHANGES

We had planned server as a class with a few methods in it but now the multiplayer networking piece contains many classes. The Server class contains multiple threads: The Server Thread that keeps the server alive and waits to on clients to connect, and a shotgun thread that works as a thread that waits on a queue. When there is something in the queue it will send to all connected clients. The Client class was added with a new class within it that represents a connection to Server. It also contains two threads: a sending thread and a Receiving thread. We also added a mechanism so that the Server knows that it is currently in Lobby, and passes around a LobbyMessage which is a class that will be used by the view to show what is to be displayed.

We also created new classes for the two tables, InspectTableViewController and HealthTableViewController, with their respective row objects, InspectRow and RobotHealth. These classes were made to reduce the clutter in the Game and GameView classes. We also removed all connections to the PostGameView, as we did not have time to code all the necessary parts of it. Finally, the attack, endTurn, forfeit and leaveGame methods were removed from the Player/Observer classes and are now done entirely in the controller.

4 REMAINING BUGS

Aside from the unimplemented classes and functionality listed above, we have encountered few remaining bugs.

- Entering an IP address that is not valid will crash the client.

- Will crash if two people with same username enter match.

- When a Player loses player connection the game crashes

Version History:

- **12/04/2016 – Initial Deployment Version – DEPLOY-V1.0**