**CMPT 370 D1**

**DEPLOYMENT REPORT**

Dec 3, 2016

Contents

1 INTRODUCTION 2

1. **TEAM MEMBERS** 2

2 UNMET REQUIREMENTS 2

3 DESIGN CHANGES 3

4 REMAINING BUGS 3

# 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.

## **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

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.

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 to 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.

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 planned server as a class with two methods in it but now Server 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 that when there is something in that queue it will send to all connected clients. The Client class now also changed with new classes within it which is a connection to Server and each of the new class contains two threads a sending thread and a Receiving thread. We also now added that the Server knows that it is currently in Lobby that passes around a LobbyMessage which is a class that will be used by the view to show what is to be displayed.

# 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