**CMPT 370 D1**

**CONSTRUCTION DOCUMENT**

Nov 26, 2016

**CODE REVIEW**

For our code review, we met to review the game class in our system. This is the largest piece of the system, and acts as the controller. We felt this was a good choice for review because it is the most complicated class, and it interfaces with almost all other classes in some way. Much of the game logic and communication takes place in the game class. Each feature interacts with it in a specific way, and understanding the layout and design of all of the functions within is crucial for further development.

We booked a room in spinks with a projector so we could display the code for everyone to see. We had the people who wrote the functions in the class describe an overview of what the function does, and then delve into more details. We asked a lot of questions, and made sure everyone was on the same page with each piece. We also discussed future improvements and plans for the class.

We mainly discussed the effectiveness and functionality of the code itself. We discussed proper commenting and documenting procedures as well. We had discussions about how the game will communicate with the server, RobotLibrarian, and the view. The turn ordering for robots and the system used to keep track of it was also discussed.

We benefited from this by having an all hands meeting where we had the opportunity to focus on the controller part of the game. We got the chance to see the current state of the game, and what needed to be fixed and added. We feel that this time was used well to improve the code for the game class. We also found and fixed many small bugs that will save headaches later on.

**PAIRED PROGRAMMING**

**NICO DIMAANO**

**NIKLASS NEIJMEIJER**

**KYLE SEIDENTHAL**

For my first paired programming session, I worked with Nico Dimaano. We decided to focus our session on the networking piece of the software. I started out as the driver. We quickly found some of the flaws with our design for the server class, and decided to break it up into two separate classes, the client and the server. We got quite a bit of work done. As we worked through how the system would work, he helped me navigate around many errors that may have caused headaches later. After about an hour, we switched places. I helped navigate through the code we had already written, and pointed out flaws in some of the logic we had come up with. Overall, we got a tremendous amount of work done. In the two hours we allocated for this session, we had completed most of the fundamental functionality of the server and client features. I would definitely do this again.

For my second session, I worked with Niklass Neijmijer. We decided to tackle to controller with our session. We made a lot of progress with the functionality. Because I had worked on the networking side, and he had done many of the view elements, we had a lot of good arguments about how to make the controller work well with both elements. This was very productive, and together we sidestepped a lot of headaches that would have been caused by different understandings of how some of the functions should be implemented. By combining our knowledge, I think we came up with the most effective solutions for the communications between the view, controller, and server. This was again a very successful session, and I would definitely do it again in the future.

**BRENDON STERMA**

**JIAWEI ZANG**

**CHANGES FROM DESIGN**

One major change we have made since our design phase was to the networking side of the system. We realized that what we had just would not work. We have decided to break the server class up into a server and a client. This allows us to run one server thread on a machine to host all clients connected to the match. Many new functions were created in the game to support this. We have also decided to make the client and server part of the controller, because they communicate directly with it to maintain the state of the game.