

Construction Document
CMPT 370

Group C4

Jack Huang

Brandon Jamieson

Ixabat Lahiji

Daniel Morris

Kevin Noonan

Changes from our design:

- **Model:**

For every class we decided to make every field variable private since there are some fields that will not be changed after the creation of each class. We do not want any of these to be accidentally changed. Since each field value is private we implemented the appropriate getter and setter methods.

Board:

Robot:

Since the range statistic is an integer we changed the datatype of range from Hex to an integer. We also changed the absDirection() method to being a field. It was not possible to calculate the absDirection so it will have to be stored in the robot instead. We decided to make every field private since there are some fields that will not be changed after the creation of the robot and so I do not want them to be accidentally changed.

RobotTeam:

We decided to implement a team number attribute. This would allow the creation of robots inside of this class to know what team number they are a part of.

Spectator:

Nothing further was changed.

Hex:

Nothing further was changed.

Interpreter:

- **View:**

- **Controller:**

Pair programming reports:

Daniel Morris pair programming session reports:

The pair programming session I had with Jack went very well. Jack and I had both been assigned to work on the model classes within our model view architecture. During the session Jack and I decided to work on the more difficult methods within the model such as searching for robots within a specified range of a location and moving robots between locations. At the start of the session it seemed as if we were both very involved in the problem at hand and at least in my mind we were both trying to be the pilot, instead of having one of us being the navigator. After a while I started to realize that when I am not the one at the computer I did not have to be totally involved in the task at hand. After realizing this I felt that when I was not the one sitting at the computer that I could think more lightly on the problem at hand and focus on the bigger picture. I definitely think this helped us become more efficient and ensure that the person at the computer was not too fatigued and could work very effectively.

At first I did not think the pair programming session with Kevin was going to work out very well but it turned out to be very successful. During the session Kevin and I decided to work on joining the model, view, and controller together. This turned out to be difficult for us since that there were some classes that either of us hadn't worked with yet since they were made by other team members. Although with that being said we were able to decipher what needed to be done to join the classes together and we started working on that. The main problem that we came across was that the panels in the view had initially been constructed with separate listeners for every button or combobox they had displayed. Due to the architectural choice made in the design document we were not able to make changes to the model within these listeners. The solution to this problem was to make the controller an action listener and whenever a view panel received input from the user the panel would trigger the listener within the controller with a unique `actionCommand`. The controller would then use this command to make the necessary changes to the model and update the view accordingly. Once we had decided on this solution it got the ball rolling for us and it became a very efficient pair programming session. We were able to take turns as the pilot / navigator to maintain our focus and went on to accomplish the task at hand.

Code review reports: