**Introduction**

**Implementation Strategies**

Algorithm implementation:

In this project, we have multiple choices for implementing the algorithm. Our team proposed two ways: (1) Implementing the algorithm in Java in back end. (2) Implementing the algorithm in JavaScript in front end.

From my point of view, there are advantages and disadvantages of both solutions.

For the first one, I am familiar with Java. It’s easy for me to implement the algorithm in Java. I don't need to learn anything more. However, If I did in this way, then I need to accept all the running environment parameters from front-end and return all running results to front-end. This would increase a lot of work.

For the second one, I am not proficient at JavaScript although I can write some simple JavaScript code. If I did in this way, I must pick up JavaScript in a short time at first. Apparently, the benefit of this solution is that I can use the environment parameters directly from the front end and return the results to GUI part rather than through Http request. This will save a lot of working in both my part and GUI part.

After comparing two solutions, I chose the second one. The other two teammates agreed my choice.

Another problem I met is using which algorithm to find the shortest path of two agents. This problem is different from the previous one as I can decide it by myself. At first, I used BFS to get the shortest path of two agents because the distance of each two adjacent space is the same. When I implement it, I found I need construct new objects to track the trace. It is difficult. Then I searched some other shortest path finding algorithms. I found A\* algorithm fit this project and there has a JavaScript package providing the function of finding the shortest path of two agents by given the environment and position of two agents. I was excited that I don't need to implement the function of finding shortest path by myself.

**Technologies and tools**

JavaScript

**Challenges**

1. Because we decide to use JavaScript to implement the whole project, all team members need to pick up it in a short time. I learned it by myself at first, if I met some problems I asked help from teammates. Fortunately, I can write code dependently after several days learning.
2. Another challenge we met is distribution of work during each sprint. At first, as Yifan is good at JavaScript, he wrote a lot of code. I and Chaohui did more work on documents part. After we got the feedback from the professor of the first demo, we adjusted the distribution of work. Every one of us need to write code as well as the document. Every one need to participate each part in the project. In addition, we need to guarantee the equality of work.
3. The format of configuration files is not given at first. At first, we set our own format of configuration files. But at last, professor gave his own format. We were forced to modify professor’s test configuration files manually or change the code of reading configuration file. Yifan didn’t want to rewrite the configuration file in our own format. He suggested me to rewrite the code of reading configuration. I was forced to do it.
4. The most challenging is the data structure of environment. As we designed the data structure at very beginning, we didn’t consider all cases. Therefore, our data structure has some drawbacks. We were forced to change a little bit of the data structure but it involves a lot of code. It took a lot of time to do that.

**Learning**

Previously, I know a little about Scrum method theoretically. As I didn’t experience how to apply Scrum in an actual project. It’s hard for me to figure out the differences of Sprint and Sprint tasks. Through this project, I can figure out them clearly and know what it is in the actual project.

The design of a project is very important. Previously, I know it is important but didn’t experience it because the requirements of most projects are not changed. In this project, I made a mistake at first which result in doing a lot of work at last to fix the drawback of the data structure. I realized that a tiny error at very beginning will become a big problem at last that need cost tons of time to fix it. Through this project, I realized the importance of the design in an actual project.

The management in a team is not easy as expected. When I did a project by myself, everything went well. However, in a team, it’s totally different. Teammates maybe misunderstand your means. They might forget what we discussed and decide a problem by themselves. Overall, in a team development process, there will be a lot of inconsistent stuff need to deal with.

In the technology aspect, I picked up JavaScript in a few days, know a lot of JavaScript related framework and tools such as webpack, node.js, D3.js and Express. I experienced the benefit of organizing the code in module.