# Introduction

We developed a tool which can help user to visualize multi-agent patrolling.

The user can do the following activities:

• Selecting algorithms

• Loading an environment file

• Viewing the whole block view

• Viewing a particular graph view

• Switching the between block view and graph view

• Run one step or multiple steps of an algorithm

• Viewing the detail about the agents in each node in graphical view

• Viewing the trace of an agent when running an algorithm

• Viewing the target list in graph view

• Save a run of an algorithm

• Filter historical runs of the algorithm

# Implementation

**User interface:**

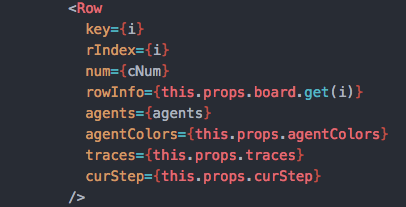
React.js is used in this project as a framework to build the user interface. It not only a JavaScript library, but also a thought of building user interface.

Componentization is one of the features of React.js and it heavily affected the way we develop the user interfaces. Writing React.js code is like using building blocks. We first use some basic components to build some higher-level components and then use those higher-level components to build some other components on top of them. The whole web application is built up in this way. This makes the UI structure clear and allows us to write reusable code. If a component is used in many parts of the code and we want to change its style or add some stuff in it, we only need to change one part of the code in where this component is built up. This saved us a lot of time and made it possible to spend some time on making the application looks better and easier to use.

Figure 3.1 – 3.3 shows an example of componentization used in this project.



3.1 Using some divs to build the “Row” component



3.2 Using the “Row” component to build the “Board” component (part of the code)



3.3 Using “Board” components to build the “Visualization” component (part of the code)

Another core feature or thought in React.js is “state”. We took this thought and implemented the user interface. A “state” is a storage of an attribute that will affect the appearance of user interface. In a traditional way, if a user changed some data after some operation and the data have some relations to the user interface (like the position of an agent), then there should be some code to update the user interface with these new data. It would be much complex if there are so much logic and elements on the user interface. While in the “state” way, these data are stored in “state”, what we only need to focus on is the relations between the “state” and the user interface. After these relations is built, the changes on a “state” will automatically reflects on user interface. In this project, “state” appears everywhere. Most updates on the user interface (like agents’ positions, visited spaces, target lists and the showing of graph views) are achieved by changing the “state” using the “setState” method.

For the graph view, we decided to use canvas to draw the graph. Compare to just using some basic HTML elements (like div) with CSS to draw a graph, canvas is much more flexible and powerful. Besides, there are a lot of excellent third-part libraries for constructing a canvas graph. D3 is selected as the library to develop the graph view, because some team members have already learned a little about it. A function called “graph” is implemented to build the graph. This function takes three parameters: region, traces and step. The “region” tells the structure of selected region, the “traces” tells the trace of each agent and the step tells the current step. When another region is selected or the “run step” button is clicked, this function will be called to build the new graph.

**Database:**

Mongo is used in this project to build the database. The reason why we decided to chosen it that is we used JS to wrote the frontend code and the structure of data is also designed by JS. So the JSON is the perfect data format for us to store the data, and we did not need to design the data structure in database. We just stored the data from the frontend, there is no data conversion process.

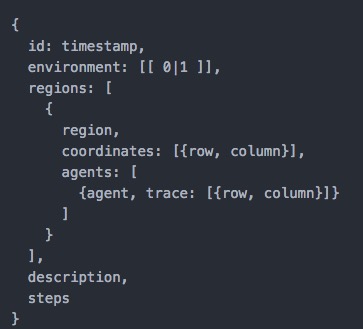


Figure The Data format that Stored in Mongo

The Figure shows what data we stored in our Mongo. The primary key is the timestamp which came from front end. The environment is an array which will indicate whether the space is an open space. The field “regions” is an array of objects, each object has the detailed data of a region. The field “description” will be added when the user finished an algorithm run. The field “steps” records the total steps of an algorithm run. In our database, we only had a table named “record” (We call the term “table” as “collection” in Mongo) to store those algorithm runs.

There are two main methods to operate those algorithm runs. One is the “save” method, the other is the “retrieve” method. The “retrieve” method will support the multiple conditions search. In our project, we used a simple way to realized the multiple conditions search. Firstly, we will check whether the user has entered the search conditions. If all the search fields all empty, just return all the historical runs in database. If the user enters some search conditions, the “retrieve” method will return those results which matched the filter conditions of the user. The basic idea of the multiple conditions search is using filter. The Figure is the filter we used in our project to get the matched runs. The parameter “envSize” corresponds to size of the environment. The parameter “start” and “end” correspond to a time period, and It can be accurate to minute. The parameter “description” corresponds to the description of an algorithm run. The parameter “regionNum” indicates the total number of region in a historical run. The parameter “steps” corresponds to the total run steps of an historical run.



Figure. The Filter of the Result.

**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.

# Technology and tools

Npm. It is used in this project to install and manage JavaScript packages.

Webpack. It is a module bundler for JavaScript applications. We used this tool to achieve modular development using JavaScript.

Babel. It is a compiler. By using it, we can write JavaScript code with new features without caring about the compatibility in old browsers. It will compile the code to old version that supported by old browsers. It also compiles JSX which used in React.js to plain JavaScript.

Eslint. It is a tool for identifying and reporting on patterns found in ECMAScript/JavaScript code, with the goal of making code more consistent and avoiding bugs [1].

D3

VSCode

Node.js

Express

ECMAScript 2015

Git

Github

MongoDB

Linux

# Challenges

There were so many challenges for us to did this project. One aspect is for each team member, and this aspect is the challenges for each individual. Every team member had his own challenges on technology. The other aspect is for the whole team, and this aspect is focus on how could each member cooperated with other.

**The Challenges for Team**

The first challenge of our team is that only two team members are familiar with writing code by JavaScript. So we must train the other team member to master the language in a short time. It’s a big problem for us. Because if the team member did not master the language in a short time, we cannot divide the workload equally among all three of us at all times, there was no consistency of workload in the schedule.

The second challenge of our team is the communication between the team members. Sometimes, there were some conflicts on solving a problem. One team member insisted that his solution was perfect, actually the solution did have some problems. How to convince the team member and made his to change his solution was a challenge for the rest of us.

Another challenge for our team is that all of us are Chinese, and no one is the English native speaker. So sometimes, there are some misunderstanding for the description of this project. We may take more time on fixing the problem caused by misunderstanding.

**The Challenges for Yifan Gu(补充你的难题也是如何整合另外两个人的代码)**

Because this project is developed by three team members, the interactions between different modules developed by different people is a big problem. Therefore, at the very beginning, the team reached an agreement on the structures of data transferred between different modules. Besides, the team also discussed on how a module should be used in other modules and reached an agreement. After doing this, the three people just focus on developing their own modules and we did not encounter too much trouble when combining these modules together into one application.

**The Challenges for Chaohui Xu**

After lots of reaching on the Internet, our team decided to chosen D3 to develop our graph view. D3 is a JavaScript library for manipulating documents based on data [2]. It’s very convenient to do the DOM manipulation. I must know the basic properties of D3 in a short time, so that I can write the graph view code as soon as possible.

**The Challenges for Sheng Zhang**

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.

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.

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

Each team member learned a lot from this project. We think this course helped us learned a lot of skills on communication with others and some related technology.

**What Yifan Gu learned**

Before this project, I did not have much experience on a project which developed by more than two people. By developing this project, I realized that even three people are much different than two people. The difficulty of the management on project increased a lot. The communication between team members is much harder than just two people. The meetings gave us a good way to communicate with each other on the project and solve problems. Face to face communication is much more effective than communicating on Internet.

I learned Scrum a lot. Though developing this project may not be the same as developing a project in a real company, it still gave me a picture of how Scrum works and how it would affect the developing process. We need to care about sprint tasks, we need to care about both coding and testing. We also need to adapt to the changes on requirements.

I also learned many new technologies in the web world. All code is written in ECMAScript 2015 (also known as ECMAScript 6). There are many new features like arrow functions, let, const and promises. In the next few years, these features will be used more and more frequently in the area of web development. We also used React.js, Webpack, Babel, Node.js which also let me learned a lot.

**What Chaohui Xu learned**

At first, I had no idea what Scrum is and never worked within a team which has three team members. During the first sprint, actually we did not understand the meaning of Scrum, we just filled the date of the sprint task, and did not use it to track our schedule. The professor strongly suggested us to understand Scrum and use the date to track our tasks. After that, all of us searched related material about Scrum on the Internet, and had a clear basic understand of Scrum. It has an important role in our real development and it’s very common to use in a company. With this experience of using Scrum, now I know why it can improve the speed and the flexibility of a development. It has a “push” force to make every team member to finish the task as soon as possible.

During the implementation of graph view, I learned how to use D3 to draw some fancy diagrams. In addition, I think I can be more mature to use JavaScript to write code. I have a clear understand of how to write a whole project only use JavaScript, how to build the framework of the JavaScript project. By using Mongo, I have a more distinct understand of the methods in Mongo and the difference with SQL database.

During every meeting, every team member coordinated their own language to communicate with other. I thought I learned some skills on how to communicate with other. I can use a peace way to convince other. If you find someone is wrong on some aspects, we should approve them firstly, because they did their own job, then convince them with reasonable ground. I learned it from the lecture “people management”. I think the communication is very important for me to work with other.

During the demo, I learned I should speak loudly even maybe there are some problems in pronunciation.

**What Sheng Zhang learned**

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.

**Reference**

[1] http://eslint.org/

[2] https://d3js.org/