

Report

In this short report, I shall be analysing the team's performance in the Ant Game project. I shall discuss our overall approach, and document any issues and resolutions to those issues.

Once the team allocations were released in February, communication between members was quickly established via social media. We all got in contact with each other and knew how to quickly contact one another both individually and as a group. An initial meeting was also organised to meet each other face-to-face ahead of the start of the seminars.

At this meeting, the tone of all future meetings was set. We quickly organised ourselves, agreeing that we were a democratic decentralised system (having equal footing and control in everything). We also began to discuss how the various deliverables were to be shared amongst each other. Communication and configuration management systems were set up using Slack (a chat client which has useful integrations for programmers) and GitHub. All team members were registered and had their local development environments (i.e. local Git repositories) setup by the end of the week.

This acted as a precursor to our approach as a whole to the project. Throughout the project we would set tasks to accomplish each week. We would go off on our own and do said task, and then meet together in a pre-arranged meeting to collate our thoughts and findings. We found this methodology worked very well, as it allowed individualism — it allowed each member to undertake each task (such as a PERT chart or a class diagram) as they saw fit. This made sure we had multiple 'versions' of each task, thus making it less likely for the final result to miss something or make an error with respect to the requirements. We would discuss our attempts as a group, and then combine and manipulate them all into an in-depth result of all our attempts.

This approach together with the excellent team dynamic ensured that the project went quite well. The final program is (almost) fully functional - it satisfies all user requirements:

- The program can check if an ant-brain supplied by a player is syntactically well-formed.
- The program can check if a given description of an ant world is syntactically well formed and meets the requirements for ant worlds used in tournaments.
- The program can visualise a given ant world.
- The program can generate random but well-formed ant worlds.
- The program allows two players to play: i.e. enables two players to upload their ant-brains

and choose an ant-world, and then runs the game in the ant world, taking statistics and determining the winner of the game.

- The program allows users to play tournaments, where an arbitrary number of players can upload ant-brains, who are all paired up to play against each other.

From release testing the ants appear to behave exactly as prescribed by their brains, and the final submission has all unit tests passing. This success is down to the careful planning and team organisation as described above. There is still some ‘polishing’ to be done with the GUI. For example, the visualised tournament simulation is not fully integrated with the core tournament simulation — even though all matches are simulated and viewable (along with their statistics), the winner is not determined or shown in the GUI program. However a non-GUI tournament simulation would be almost immediately available through a command-line program with just a few lines of code — the core functionality is there, but not linked with the GUI.

Due to our team structure and dynamic, we had had absolutely no issues or problems to document. The team has worked very well together, we all got on well, worked hard and did exactly what we were meant to do. We all knew that in order to complete this project we would all need to work in a consistent manner, and that resulted in a project with very few issues. There were of course minor problems caused by events outside our control such as train delays and illness. These were easily dealt with by looking at the recorded minutes (pushed to GitHub within a few hours after each meeting) to see what was missed.

As a whole, this project can be considered a success. Due to both the team work ethic and general cohesiveness, all required deliverables were completed, as well as the code behaving as it should.