

Modelling an Ant Colony

Functional Specification

Kevin Cleary - 16373026

James O'Neill - 16410652

Date: 22/11/2019

Table of Contents

1. Introduction
 - Overview
 - Business Context
 - Glossary
2. General Description
 - System Functions
 - User Characteristics and Objectives
 - Operational Scenarios
 - Constraints
3. Functional Requirements
 - 3.1 - Defining the Ant Movement Algorithm
 - 3.2 - Generating a Simulation Environment
 - 3.3 - Displaying Ant Movement
 - 3.4 - Gathering, Storing & Displaying Simulation Metrics
 - 3.5 - Allowing User to Generate Specified Environments & Ant Classes
4. System Architecture
 - 4.1 Graphical User Interface
 - 4.2 System Backend
5. High-Level Design
 - 5.1 Object Diagram
 - 5.2 Context Diagram
 - 5.3 Use Case Diagram
 - 5.1 Context Diagram
 - 5.2 Object Diagram
 - 5.3 Class Diagram
 - 5.4 Use Case Diagram

6. Preliminary Schedule
 7. Appendices
-

1. Introduction

1.1 Overview

This project is a study on Machine Learning and Genetic Algorithms through the implementation and simulation of a Virtual Ant Colony. We intend to train our virtual ant colony to solve fastest route problems in real time and apply their behaviour to emergency response vehicles facing the same problems. We are also going to implement various subspecies of ant in our simulations based on differing traits and allow users to alter these traits to suit their desired needs.

Ant Colony Optimisation Algorithms were first implemented by Marco Dorigo (IRIDIA, Brussels) in 1992. Dorigo used nodes and edges to represent his virtual ants and their environment, using his trained 'ants' to solve a classic Travelling Salesman problem. Ant Colony Optimisation (ACO) Algorithms are modelled on the behaviour of ants searching for food. In Nature, ant colonies work as a unit while foraging for food, with each individual ant leaving chemical trails known as pheromones, that can then be smelled by other ants on the route. The ants work together to find the stronger pheromone paths and over time find the quickest route to and from their desired food source. This project will involve studying Stigmergy in nature and how to apply it in the context of Machine Learning.

Our virtual ant colony will be displayed in a GUI environment for users to view their behaviour over time in finding their fastest routes. Important nodes in this environment will include the Ant Hill (Home), Food Sources (goals) and various obstacles which will cut off paths. Obstacles will appear by the user's hand or at random during any simulation, and the ants must adapt to find the shortest route to their goals based on these environmental changes.

As a further study into ant behaviour and biomimiry in modern design, we want to simulate different species of ant colony and have them interact within our GUI environment. We plan on doing this by having our basic ant class become an interface for our various subspecies and introduce different traits based on ant colonies in the wild e.g size, speed, repopulation rate etc. This will allow users to play around with different attributes of the ants to see what factors may improve their ant's optimal route, and will also allow the simulation of two rival colonies trying to access the same resource.

1.2 Business Context

Business Context 1: Emergency Response

We want to use our ant colony simulation to be a basis for Emergency Response strategies. In times of crisis, speed and safety are paramount, but in the wake of a natural disaster some important routes can be cut off. We want to use the behaviour of our trained ant colony to find the shortest route based on all available routes, as well as switching this route when obstacles appear to cut off any particular route.

This would need to be on the assumption that obstacles to routes are known in real time. Our project in its current form requires the known routes/obstacles and will provide the simulation and results containing the best route for the rescue/evacuation teams.

Business Context 2: Education

Although we are focusing on the context of Emergency Response, a lot of our work can be transferred into multiple other contexts. Our system will act as a resource for education around Genetic Algorithms and the theory behind our simulation. We are going to achieve this through the customisable elements within the Ant Colony and the Environment. Users will clearly be able to see what effect their changes have on any given simulation through the metrics our system will gather and display at the end of each simulation.

When dealing with this interaction-based learning feature of our project, we have to ensure that only the attributes changed by the user will differ between any two simulations. This way, the user can clearly see the impact of their changes on the simulation, unaffected by any other external factors.

Other potential business contexts for this project include autonomous cars, designing traffic systems etc.

1.3 Glossary

- **Travelling Salesman Problem:** An optimisation algorithm problem defined by mathematicians W.R. Hamilton and Thomas Kirkman, with a focus on finding the cheapest, shortest or fastest route around a graph of nodes.
 - **Pheromones:** a chemical substance produced and released into the environment by a mammal or insect, affecting the behaviour or physiology of others of its species.
 - **Stigmergy:** a mechanism of indirect coordination in which the trace left by an action in a medium stimulates subsequent actions
 - **GUI:** A graphical user interface, a computer environment that simplifies the user's interaction with the computer by offering visual elements to the user.
 - **Biomimicry:** the design and production of materials, structures and systems that are modelled on biological entities and processes
-

2. General Description

This project aims to develop a simulation that mimics the behaviour of ants traversing their environment to find food. In nature, ants are extremely proficient at finding the shortest path from their anthill to the source of food. Through use of chemical trails (pheromones) ants can successfully find the shortest path to and from a food source, and in turn notify the other ants of this short path using the chemical trails.

Our project will be broken down into various functional stages:

1. **Simulating Ant Movement**

The first stage will involve developing our Graphical User Interface to display the movement of our ants, generating the randomised environment that the ants will traverse, and developing the basic way the ants will move around that environment.

2. **Enhancing the Ants' Movement Algorithm**

The second stage will involve mirroring the behaviour of real-life ants, and making our ants as efficient at finding their way from anthill to food source as ants are in the real world.

3. **Incorporating Environmental Changes**

This is where we will simulate changes to the ants' environment. This will properly test the ants' ability to adapt to changing circumstances, and at this point, we will improve upon the ants' movement algorithm.

4. **Applying To Real-World Circumstances**

This will involve introducing our trained ants to real-life environments. By having the ants' "anthill" and their "food source" at different ends of a real world environment (ie. changing the environment to mimic Dublin's streets), we can find the shortest path to and from two points. Also, making use of the environmental changes, we can simulate streets being blocked, and a new route being calculated.

2.1 System Functions

- **Start Simulation**

Allow the user to start a simulation successfully. The graphical simulation will load up a

randomised environment, and the environment will be populated with the anthill and a number of food sources. Ants will spawn from this anthill and will begin their search for food.

This will be the basis for all other system functions, so this will be the function with the highest priority. However, while the other functions are being developed, this function will also be enhanced to suit the other functions as needs be, but basic functionality is needed before starting on any of the other system functions.

For us to consider this system function accomplished, the ants must spawn into the environment, have a movement algorithm that successfully finds the food, brings it back to the anthill, and leaves a trail that will aid other ants in finding the food source. The user must be able to view this simulation graphically after starting the program, and must be able to start and terminate the simulation at any point.

- **Edit environment**

Allow the user to edit the environment that the ants traverse. This will involve allowing the user to drop food sources, drop obstacles, and choose where the anthill will be placed. This will allow users to test our ant movement algorithm, and will allow users to model real life environments to find the shortest path.

For us to consider this system function accomplished, the user must be able to choose the location of the food sources, the location of the anthill, and the location of any obstacles. The user must be able to run a simulation with these new environmental changes with ease.

- **Edit ant class**

Allow the user to edit the ant characteristics. This will involve allowing them to edit the ant's speed, spawn rate, time-to-live, etc, allowing the user to impose extra constraints on the ants on their search for the food sources.

For us to consider this system function accomplished, the user must be able to successfully edit the ants characteristics and start the simulation with the new characteristics in effect.

- **View metrics**

Allow the user to see the statistics of previous simulations. The user will be able to see basic metrics, such as simulation duration, how many food sources collected, ant characteristics for that simulation, etc.

For us to consider this system function accomplished, the user must be able to view all statistics and metrics gathered from previous simulations.

2.2 User Characteristics and Objectives

The focus of this project is to apply our trained ants to solve route problems faced by Emergency Response teams, but we want to design our system to be applicable to many different user bases. As such, our aim is to create a system that is simple at first glance, and while focusing on our assigned problem we want to identify transferable aspects of our system for future growth.

Taking Emergency Response as our target userbase, the primary focus of our ant colony will be to find the fastest route to their food source while also adapting to environmental change as quickly as possible. We plan on determining the optimal route for emergency response vehicles looking for survivors in a natural disaster, by using our trained ant movement algorithm to determine the quickest paths from vehicles to survivors. The ant movement algorithm will determine the shortest route, even if some streets are blocked due to wreckage in the disaster.

We are not assuming any prior knowledge of genetic algorithms from our userbase, and so want our application to be both simple to use and also to offer descriptions of the theory behind our simulation. As well as a tool to be used by different services, we want our app to be as educational as possible. Our GUI will be designed in such a way that users can view our virtual colony at work in real time, as well as clearly being able to visually identify the different components within our simulated environment. We also plan to display informative metrics to all users during/after any given simulation for further clarity on the operation of our system. This will allow users to pinpoint which attributes have an impact on the colony in any particular environment.

2.3 Operational Scenarios

Scenario 1:

Goal in context: User starts up the system

Description: The user opens up the application and is met with a greeting menu displaying all available actions and basic information.

1. User runs the application
2. GUI starts running
3. GUI displays the system's home menu displaying options to Run the Simulation, Edit the Environment, Edit the Ant Colony and View Information surrounding the system.

4. Below the menu, the GUI will display information about the current state of the simulation; ant colony size, important colony attributes, simulation duration.

Scenario 2:

Goal in context: User runs a simulation

Description: The user selects the option to Start Simulation from the menu and an environment is displayed simulating the ant colony's behaviour

1. User clicks on the option 'Start Simulation' from the home menu
2. The GUI displays an environment including the ant nest, food source, and any extra attributes specified in the environment options
3. The ant colony simulation begins with the virtual ants spawning from the ant nest and running the ACO algorithm to find the optimal route to the food source.
4. If the user clicks the 'End Simulation' button at any time during the simulation, the simulation ends.
5. The simulation runs for the duration specified in the Environment Options menu.
6. Once the simulation ends, the GUI displays metrics giving info around the run.

Scenario 3:

Goal in context: User makes changes to the environment

Description: The user changes the values of the environment attributes in a menu affecting how the environment is displayed for the next simulation.

1. User clicks on the option 'Edit Environment' in the main menu
2. The user is taken to a new menu displaying advanced information about the Environment. These attributes include: Size of map, Obstacle frequency, simulation duration, Location of obstacles etc
3. The user is permitted to change the values beside any given attribute by entering a numeric value or by clicking on arrows within any option.
4. Once changes have been made, if the user clicks on the 'Save changes' button the new values are stored within the GUI when rendering the next test environment. The user is brought back to the main menu.
5. If the user clicks the back navigation button, they are brought back to the main menu with no unsaved changes applied.

Scenario 4:

Goal in context: User makes changes to the virtual ant colony

Description: The user changes the values of the ant colony attributes in a menu affecting the behaviour of the Ant and AntColonyOptimisation classes in operation in the next simulation.

1. User clicks on the option 'Edit Ant Colony' in the main menu
2. The user is taken to a new menu displaying advanced information about the Ant Colony.
3. The user is permitted to change the values beside any given attribute by entering a numeric value or by clicking on arrows within any option.
4. Once changes have been made, if the user clicks on the 'Save changes' button the new values are stored within the Ant and AntColonyOptimisation classes when rendering the next test environment. The user is brought back to the main menu.
5. If the user clicks the back navigation button, they are brought back to the main menu with no unsaved changes applied.

Scenario 5:

Goal in context: User views system information and theory

Description: The user views a menu surrounding the information and theory behind the system and its basis/applications

1. User clicks on the option 'View System Information' in the main menu
2. The user is taken to a text file detailing the operation of the system, the applications associated with our system and the theory behind our process and on genetic algorithms as a whole.
3. The user may search within this document using links and a table of contents.
4. If the user clicks the back navigation button, they are brought back to the main menu.

2.4 Constraints

Time Constraints: As we only have a limited amount of time to deliver this system, we are limiting our scope to have one main user base, although we plan to identify where we can expand our system to cater to a wider group.

New technologies to be learned: Some of the technologies we plan to use are ones we have never used before. For that reason, there will be some difficulty when it comes to implementing the project, as time and effort will have to be put aside to familiarise ourselves with the new technologies. For example, neither of us have ever developed anything to do with graphical animations, so this will be a new frontier for both project partners.

3. Functional Requirements

3.1 - Defining the Ant Movement Algorithm

Priority Urgency

1 2



Description

This will define the way in which the ants move around their environment, how they find food, and how they communicate with each other. This is the centrepiece for the entire project, and most other functional requirements will depend upon and make use of this function. Because of its importance to the project, it is likely that this function will require constant and thorough development across the entire project lifecycle. Optimisations and enhancements will be constantly and consistently made right up until project submission.

- **Criticality**

As mentioned in the description, this function is the centrepiece for the entire project, and as a result is essential to the system as a whole. Without this function, the project cannot expand in any meaningful direction, and any real world applications for the project are less prevalent.

- **Technical Issues**

A lot of research will go into this functional requirement, including research into biology and into computer science. We will need to study ants themselves in order to properly understand and emulate their movement and hive-like interaction with each other into our algorithm. Properly simulating a biological phenomenon will be a particular technical issue which we must overcome.

Along with researching ants in nature, we will also need to research more into computer science for search and pathfinding optimisations. Extensive research has gone into search and pathfinding, and it would be ignorant of us to disregard this research. Properly understanding these search and pathfinding optimisations, and then applying them correctly in our own project will be a significant technical challenge.

We think that there exists a combination of biological research and computer science research that will lead us to a fruitful simulation of ant movement.

- **Dependencies**

This function will depend upon "3.2 - Generating a Simulation Environment", as the ants need some space to actually traverse, need somewhere to spawn, and need some "food

source" to find a path to.

3.2 - Generating a Simulation Environment

Priority Urgency

2 1

-

Description

This function will make the environment in which the ants will move. It will determine the location of the anthill (where the ants will spawn from) and will determine the location of food sources. It will also include obstacles and impasses that the ants must be aware of and must move around in order to reach food sources.

Later iterations will involve mid-simulation blockages, requiring the ants to recalculate their path. These blockages must be generated randomly, and will be a part of this functional requirement.

- **Criticality**

This feature is another one of the most critical features in the system, and is depended upon by "3.1 - Defining the Ant Movement Algorithm". However, it is only a platform for our Ant Movement Algorithm to stand upon, so therefore it is defined as *more critical than all other features except for our Ant Movement Algorithm*.

- **Technical Issues**

The layout of the environment must be simulated in a specific way, one which is sufficiently random, but not so random that it becomes impassable. There must be a certain degree of fairness in the environment for the simulation to be realistic and useful. For example, there shouldn't be a case where the environment blocks all food sources from the anthill or vice versa. This could be combated by having a limited amount of blockages, so the environment doesn't become too cluttered.

On another note, the environment should be sufficiently challenging for the ant movement algorithm, and not have food sources in easy to access spots.

- **Dependencies**

As of writing this functional specification, there is no other feature in the system that this

function depends upon.

3.3 - Displaying Ant Movement

Priority Urgency

2 2

-

Description

This involves graphically displaying features "3.1 - Defining the Ant Movement Algorithm" and "3.2 - Generating a Simulation Environment" correctly to the user, so progress can be visually observed. This will be the main feedback for the user apart from the metrics gathered, and this visual feedback will be allow users to properly understand how our movement algorithm works.

- **Criticality**

As this is the main feedback to the user in our project, this feature is determined to be of medium to high importance, as it is the main output for the other features developed in this project. While functions 3.1 and 3.2 will be the ones developed most in this project, the importance of displaying these functions to the user should not be understated.

- **Technical Issues**

The main issues with this feature will be fully understanding how to display features 3.1 and 3.2 correctly. Displaying the position of each ant, along with it's updated position as it moves will be challenging. Each environmental feature (anthill, food sources and impasses) will need to be displayed correctly too.

We must learn the specific technologies needed to display these features, and this will be another technical issue/challenge.

- **Dependencies**

This feature directly depends on features 3.1 and 3.2, as these are the two features that this function will be graphically displaying.

3.4 - Gathering, Storing & Displaying Simulation Metrics

Priority Urgency

3 3

-

Description

During simulations, we plan to gather values of interest, such as current colony size, total ants spawned, total ants deceased, simulation runtime, food sources collected, and more. These metrics will not only help the user see how well our movement algorithm is working, but will also aid us in determining if we have improved our ant movement algorithm from one simulation to the next.

- **Criticality**

This feature is of medium importance to the project, however it will be particularly useful to display exactly how much our algorithm has progressed over the course of the project.

- **Technical Issues**

The system must interface between both [Elasticsearch, Logstash and Kibana](#) in order to store and display the metrics gathered in the simulation. We must learn exactly how to interface with these outside applications, which will be this features most prominent technical issue/challenge.

- **Dependencies**

This feature depends on functions 3.1 and 3.2, and more loosely on function 3.3. Function 3.3 isn't directly needed, but metrics could be displayed to the user during the simulation, hence this functions loose dependency on function 3.3.

3.5 - Allowing User to Generate Specified Environments & Ant Classes

Priority Urgency

2 4

-

Description

This feature will allow the user to change the environment and ant classes in the simulation as they please.

For the environment, they will be able to define where food sources are, where the anthill will be, and any blockages that may be present in the simulation.

For the ant classes, the user will be able to define the speed and time-to-live of the ants in the simulation, along with other characteristics we may think are useful for the simulation. This will effectively give certain constraints to the ants, and will allow the user to test specific scenarios.

Also, this feature will allow the user to simulate environments in the real world, like a campus they wish to find the quickest path through. This ability to edit the environment and ants will make the project particularly applicable to everyday life.

- **Criticality**

This feature is essential when applying our algorithm to everyday life and travel. For this feature to be effective, it relies on previous features to be implemented first, as applying an algorithm to the user's specification is useless if the algorithm has not been optimised yet.

- **Technical Issues**

Defining exactly how the user will be able to change the simulation variables will be the main issue with this feature. We will need to be able to provide a graphical user interface for the variables in the simulation, and validation must also take place on the values entered by the user. Finally, we must also make sure that the variables applied by the user are correctly implemented.

- **Dependencies**

This feature depends on features 3.1 and 3.2, and feature 3.3 would not be 100% necessary, but would be incredibly helpful in order to visually see the changes made by this feature.

4. System Architecture

Due to the nature of this project, no outside services need to be communicated with, unless it is to post or store metrics. The entire project will mostly be a self contained unit.

We broke the project into two sections; the GUI and the backend.

4.1 Graphical User Interface

This component interacts with the user and the backend of the system, allowing the user to select simulation options, make changes, start the simulation and stop the simulation. It displays the main simulation to the user, and allows the user to see metrics on the current simulation and simulations in the past.

During the simulation, the GUI shows how the algorithm works through visual display. They can see all obstacles and food sources that the ants must travel to.

The makeup of the GUI can be seen below in the High-Level Design.

4.2 System Backend

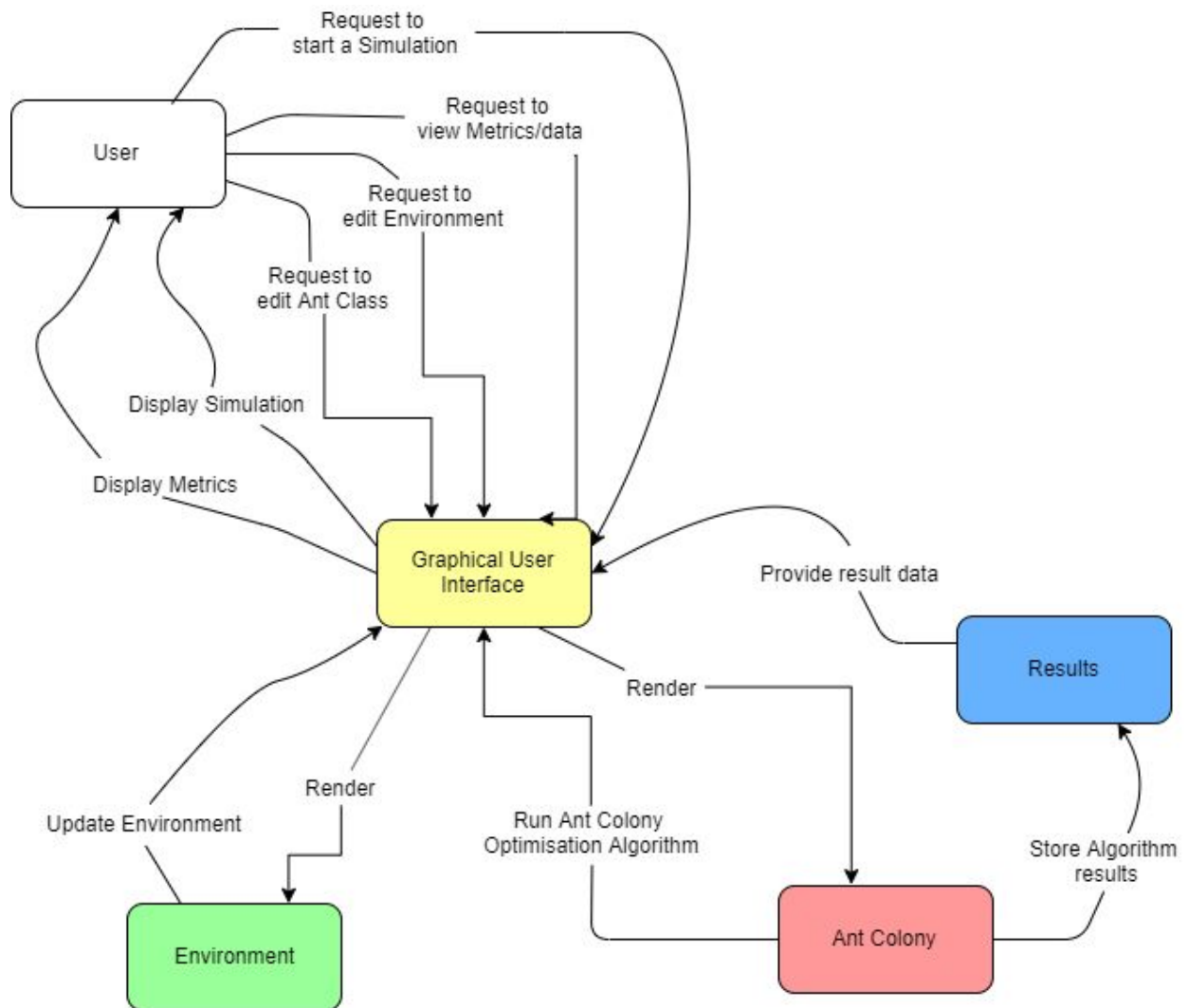
This component of the system is responsible for correctly defining the ants movements and sending them to the GUI where they can be rendered. The movement of the ants in the simulation will be determined by the movement algorithm defined in this component.

Taking the current position of the ants, the algorithm will determine the next best route for each ant, and will re-render the simulation, and send it to the GUI for the GUI to display the updated positions of the ants and food sources.

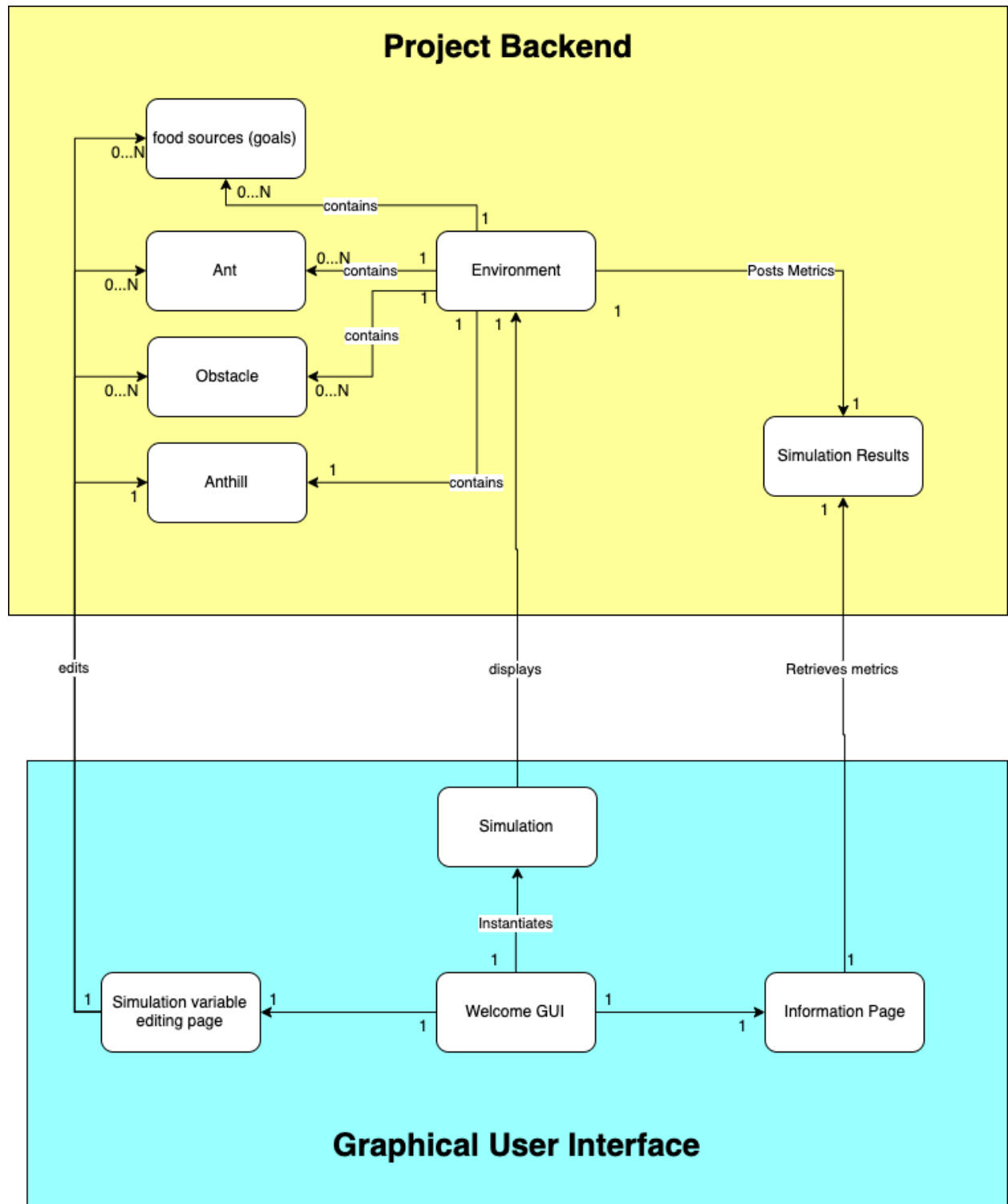
The makeup of the backend can be seen below in the High-Level Design.

5. High-Level Design

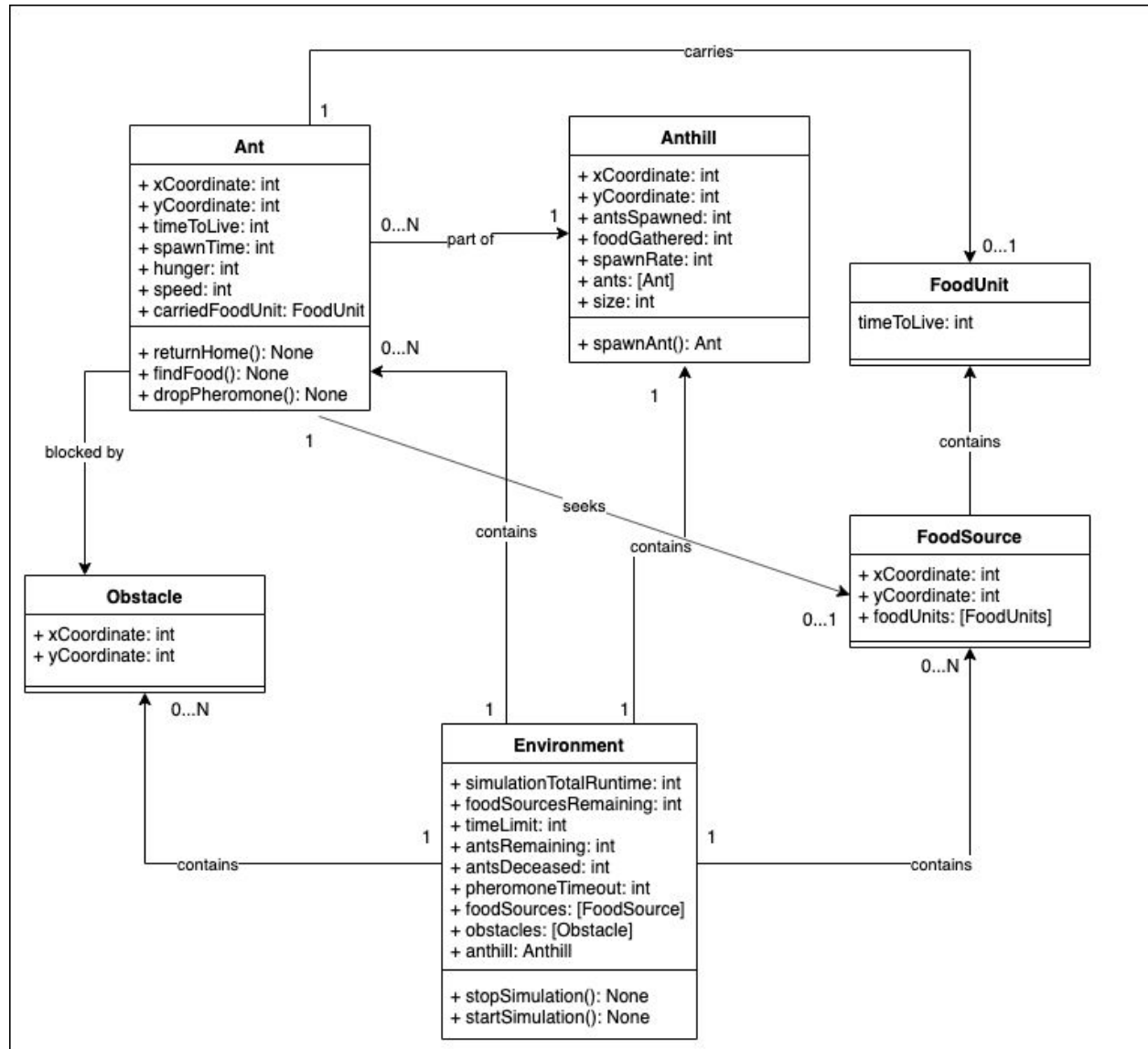
5.1 Context Diagram



5.2 Object Diagram



5.3 Class Diagram



5.4 Use Case Diagram



6. Preliminary Schedule

We have broken down the current projected workload for the project as in the table below. We plan to stick to this schedule with the use of a trello board to update on progress and to arrange weekly standup meetings in an agile fashion.

Task Name	Start	End	Duration (days)
Functional Specification	22/10/2019	22/11/2019	31
Dev environment setup	22/10/2019	05/11/2019	14
Learn tech	01/11/2019	31/11/2019	31
GUI Design	05/11/2019	31/11/2019	27
Algorithm Implementation	05/11/2019	31/11/2019	27
Simulation Implementation	01/12/2019	14/12/2019	14
Test + refine existing code	14/12/2019	17/12/2019	3
Study metrics for progress	18/12/2019	20/12/2019	2
Test + refine existing code	02/01/2020	05/01/2020	3
Study metrics for progress	05/01/2020	07/01/2020	2
Exams	08/01/2020	19/01/2020	--
Tech Spec + User Manual	11/01/2020	11/05/2020	150
Implement customisation feature: ants	20/01/2020	02/02/2020	14
Refine GUI simulation	03/02/2020	09/02/2020	7
Study metrics for progress	10/02/2020	16/02/2020	7
Test + refine existing code	17/02/2020	23/02/2020	7
User Testing	24/02/2020	08/03/2020	14
Implement customisation feature: environment	02/03/2020	22/03/2020	21
Study metrics for progress	23/03/2020	25/03/2020	2
Write up theory text file with research	23/03/2020	05/04/2020	14
User Testing	01/04/2020	27/04/2020	21
Refinements and Presentation prep	01/04/2020	11/05/2020	41

7. Appendices

- [Ant Colony Optimisation Algorithm, Indian Institue of Technology Bombay](#)
- [Java Swing](#)
- [ElasticSearch, for metrics gathering](#)