Ant Simulation

How can the adaptability of ants be simulated?

School: BBZB Luzern

IDPA No: 1233

Supervisor: Andreas Bolfing

Authors: Cédric Gasser, BMLT19E

Dániel Polgár, BMLT19E

Vincent Hischier, BMLT19E

Dominik Costa, BMLT19E

School Year: 2021/22

Submission: 27.05.2022

Abstract

We humans do not understand nature in its whole complexity, but we are steadily discovering more patterns and reoccurring themes. A tiny proportion of the beautiful complexity of nature that we looked deeper into is how ants organize themselves.

Our goal was to try and simulate an ant colony and its gathering of food. We have asked ourselves: How can the adaptability of ants be simulated? Having a broad view over different types of simulations and their application would allow us to choose one that fits our use case. With the implementation of a simulation, we wanted to show the adaptability of ants in their environment and what this type of simulation is capable of displaying.

To simulate the individual ants acting in a colony we have decided upon implementing an agent-based simulation. With the simulation running we could demonstrate emergence and compare the performance of ants in our simulation initiated with different parameters. Each ant, implemented with basic rules, led to interesting complex behavior when observing the whole colony.

Preface

During our professional baccalaureate, we can choose an interesting scientific project as the IDPA (Interdisziplinäre Projektarbeit), which gave us a chance to prove our skills in different subjects in one project.

A common interest in simulations allowed us to narrow our scope down to a few options. Simulating ants occurred to be a reasonable mix between our interests and feasibility. Different experience levels with programming in general and specifically the used technologies result in a great learning experience for all of us. The project also permitted us to try out new technologies that we otherwise would never use.

A big thank you goes to our expert Mr. Andreas Bolfing, who was always able to advise us well. Through regular consultations with him, we were able to carry out the project completely and in time. We would also like to thank our training companies for the time they gave us to work on the project.

With the following documentation and the website accessible for everyone with a browser at https://ants.cedricgasser.com/, we would like to show our enthusiasm for the interdisciplinary subject. Our procedure, solution variants and difficulties with certain steps are described in detail below.

Table of Contents

ABSTRACT **2**

PREFACE **2**

TABLE OF CONTENTS **3**

1. ANT THEORY **4**

1.1. ANTS 4
1.2. FIRE ANT 5
1.3. CARPENTER ANT 6
1.4. BEHAVIORAL ADAPTABILITY 7
1.5. EVOLUTIONARY ADAPTABILITY 9

2. SIMULATION THEORY **10**

2.1. AGENT BASED SIMULATIONS 10
2.2. CELLULAR AUTOMATON (CONWAY'S GAME OF LIFE) 11
2.3. NEURAL AGENT EVOLUTION SIMULATIONS 11
2.4. TRAVELLING SALESMAN PROBLEM 12
2.5. RANDOMNESS IN SIMULATIONS 12

3. IMPLEMENTATION **13**

3.1. REQUIREMENTS 13
3.2. TECHNOLOGY STACK 16
3.3. DESIGN 17
3.4. ANT PICTURES 18
3.5. ANT BEHAVIOR 19
3.6. CODING STANDARDS 22
3.7. REQUIREMENTS EVALUATION 22
3.8. FEATURE STACK 23

4. FINDINGS **25**

4.1. USE OF ANT PATHS 25
4.2. OPTIMIZING OF PATHS 26

5. SUMMARY **27**

6. GLOSSARY **28**

7. BIBLIOGRAPHY **29**

8. TABLE OF TABLES **31**

9. TABLE OF FIGURES **31**

10. STATUTORY DECLARATION **32**

1. Ant Theory

Ants can be found everywhere around the world. Numerous different species have evolved over the years and adapted to their environment. To gain basic knowledge about ants and their behavior we have researched about them in general and specifically about 2 species.

1.1. Ants

Ants are insects and belong to the arthropods. There are more than ten billion ants (family Formicidae) worldwide which are divided into approximately ten thousand species (order Hymenoptera). This means that every hundredth animal is an ant.
Ants live in social, organized colonies. The colony may count from twelve up to several million ants. There are three types of ants in a colony: The Queen or respectively the queens, the males, and the workers. Queens and workers hatch out of the fertilized eggs, the males on the other hand hatch out of unfertilized eggs. The so-called "young queens" are the fertile queens. Both male ants and young queens grow wings to participate in the wedding flight. After being fertilized, the queens lose their wings again and fulfill their part by laying the eggs and thereby producing offspring. Every colony depends on one or multiple queens because they are the only ones who can produce offspring. The workers are also divided into different types: Soldiers who protect the colony and the nest, scouts who search and collect, foster-mother ants who take care of the offspring, and so on.
Ants settled almost all around the world and in every habitat. Only the continent Antarctica remains untouched by those small crawlers. Generally, ants live in an anthill on the ground or under a rock which they build with materials found in their region.
Ants mainly communicate through chemical, odorous substances, the so-called pheromones. They eject pheromones to form signposts. They can state different information with the pheromones, including a signal to a food source, an enemy, or the way back to the nest. Additionally, they can communicate through knocking with their head or legs or through stridulation. Stridulation is a form of communication, which belongs to the insects, where two body parts are rubbed against each other, and thereby a sound is created. This method however has a very limited range of a few centimeters. (Kiesewetter, 2020) (schule-bw.de, n.d.)

1.2. Fire ant

The fire ant shown in figure 1, is red and black colored and from 1.6 to about 5mm long. The species is originally found in the tropical regions of Central and South America. In 1920 a descendent was created: The red fire ant was imported into the USA and spread since then to a few other countries such as Australia, China, and Taiwan. This successful diffusion is not least because of their aggressive behavior against threatening animals, which also earned them their scientific name Solenopsis Invicta (the undefeated fire ant). Red fire ants build earthy anthills under which underground constructions are found. The hills can reach up to one meter in height as well as in width. They like warm and wet climates, which is why their nests are often found near rivers, lakes, or watered grasslands.

The red imported fire ant is especially aggressive against other ant species. Therefore, the ant has the potential to endanger the indigenous environment, because they often almost exterminate native ant species. But not only the fauna suffers under them, they also destroy the flora and no one can stop them because of their dominance over their competitors. Additionally, they have a penchant for electric devices, which makes them very annoying vermin in the garden.

If attacking, the ant bites the prey with the jaw and afterward injects poison with its sting located at the abdomen. The ant will launch several attacks within a short period. If confronted with only one ant, the attack is only dangerous for people with allergic reactions. It is something else entirely if a colony feels threatened, and hundreds of ants are attacking. The consequences of those encounters are severe burnings and possibly fatal shock reactions.

The communication between the ants, as in most ant colonies, is primarily based on the pheromones. Additionally, they can communicate through stridulation.

A very rare attribute of the fire ant, which has been observed in Texas USA, is a super-colony: A few neighboring ant colonies merged into one big super-colony. While still having their separate anthills and queens, the colonies stay in close contact and organize themselves together. (biologie-seite.de, 2022) (Ludwig, 2019) (mefics.org, 2020)

Figure 1: Fire ant

1.3. Carpenter ant

There are three sub-species within the carpenter ant species: the Lasius emarginatus, which is black with a brown-yellow to brown-red tone while the smaller Lasius brunneus appearance differs with a yellow-brown coloration with a dark head. The Lasius fuliginosus is the third and the most remarkable of those three. It is from 4.5 to 6.5mm. But what catches the eye is its black and glittering body (as seen on figure 2).

Carpenter ants are found in Europe and Asia. They are especially widespread in central Europe. One colony can have up to two million ants and has always more than one Queen, except at the beginning of a new colony. Carpenter ants are a species, which sometimes makes use of parasitic behavior: A queen lays her eggs into the nest of another species. The host species will then take care of the queen and the brood until all ants of the host colony have died and the carpenter ant queen has her own workers. In large colonies, the nest is often divided into multiple nest parts. Those separate nests have one or multiple queens which belong to them. They do not have ant hills but build their comb-like nests inside of deadwood. This does not only include dead trees in the forest, but also construction wood, like for example a wooden stairway. The ants gnaw passageways into the wood and build their nest within those passageways with a substance, mostly made from wood and sugar. Empty corridors also get filled with said substance. This stimulates a certain mushroom which in return strengthens the nest with its roots.

Their nutrition consists mainly of a honey-like substance, which the ants collect from aphid colonies. If necessary, they supplement their nutrition with insects. (Garinger, 2019) (schaedlingskunde.de, n.d.) (Kress, 2021)

Figure 2: Carpenter ant

1.4. Behavioral adaptability

Within their behavior, many ant species develop special abilities to adapt and overcome certain situations. Ants are one of the best examples for the benefit of teamwork. Two examples for adaptive behaviors are described in the following chapters. Each example includes an evaluation mainly from a technical view.

1.4.1. Raft

According to the article of Jesslyn Shields from 2017, there is always a chance of flood, if the fire ant colony lives near to water or if it rains a lot. In case of flooding, they must leave their nest, because it gets destroyed. Through their survival skills, they are able to cope with this situation. The fire ants form a raft as shown in figure 3, with a tight bond of many ants. The entire colony swims on that raft. They make sure the queen and the larvae are safe. All of them are kept in a dry place covered by the workers. As soon as they find a place to climb out of the water, the ants will swarm out and search for a dry place to build their nest.

Figure 3: Fire Ant Raft (Ratchford, 2017)

To simulate a raft like this, a complex – possibly 3-dimensional - environment with a weather system would be needed. On top of that, the behavior to form a raft as a colony is immensely complex.

1.4.2. Food search

Debbie Hadley wrote an article in 2020 about the food scavenging of carpenter ants. In their colony, the insects use an interesting method to search for food. This task is always done by the workers. Their main goal is to obtain enough food for the colony, especially for the queen. After leaving the nest, each ant chooses a random path. To ensure the return, they create a trail of pheromone. The distance can be up to 90 meters with a safe comeback. After a food source is found, the carpenter ants make their way back to the nest. Thanks to the trail they made, it is not that difficult. On the way home, the production of the pheromone will not be stopped. This ensures that the path stays, and it helps other ants to find the food source. As a result, the ants will follow each other in a line. On figure 4 the trail is visible, which the carpenter ants created during the food transport to the nest.

Figure 4: Carpenter ant trail (Nanaimo Pest Control, 2016)

The food search is a more applicable example because it can be simulated with most species. The complexity of this behavior is still high, but doable and at the same time visually pleasing. There also is a huge variety of simulation parameters, (e.g.: food, ant count, distances) which leads to more experimental possibilities.

1.5. Evolutionary adaptability

"Ants first appeared on the earth between 140 to 168 million years ago, during the Jurassic period." (The Field Museum, n.d.) As nature changed, their numbers have increased because they could adapt. After the splitting of the ancient continent Pangaea into Eurasia and Gondwana, the climate has changed. Many animals adapted; others became extinct. Today, they are the most dominant insects in the world and evolved biologically in many directions, depending on their environment

Since in all ant species the queen lays the eggs, she plays a major role in the evolution, according to an article by Anna Thanukos (n.d.). The success of the evolution can be measured in the distribution of the genes. In colonies where there can be multiple queens, like the fire ants, there are two ways to evolve. Either the queen leaves the nest or she stays. By leaving, the queen must find a place for a new nest and give food to the first generation of workers. If she succeeds, the entire colony will carry her genes. If she fails, her genes get lost. The second option is safer for the queen and the transmission of the genes is almost guaranteed. But only a part of the colony will carry her genes, because of the other queen(s).

When talking about evolutionary adaptability, the adaption process happens on a larger time scale over multiple generations compared to ants adapting directly to a situation. In passing on genes, mutations occur, and the better genes have a higher probability of surviving. This is also called “survival of the fittest”. With ants, however, there is an interesting point to make: because a colony has only one or a few queens, all the next generation inherits genes from those queens.

Simulating multiple generations would require a tremendous amount of calculation power and would not have a visual effect as good as when just one generation is observed. Therefore, behavioral adaptability is a better use case for us. It allows us to implement the ants to just follow the rules and simulate the food search behavior.

2. Simulation theory

“A model is a simplified representation of a real-world situation” (Iilende, n. d.). A simulation is the application of such a model to imitate a system or procedure.

An example of a simulation could be the testing of a prototype aircraft wing in a wind tunnel. Hereby the wing would be the model which is applied to the simulation to simulate a real-world scenario where the wing glides through the air and generates lift for the aircraft. The critical aspects of simulating a behavior or system as accurately as possible are the quantity and quality of the available data and the correctness of the used model.

We have looked at several simulations to determine how we create our own ant simulation. Those simulations further discussed in this article all fall under the category of computer simulations. Those simulations model a real-life or hypothetical situation on a computer to easily observe the behavior after changes of parameters. For each simulation type, there is a description and an evaluation, to determine if it can be used in the ant simulation.

2.1. Agent based simulations

In the agent-based approach, a simulation is composed of multiple agents with autonomous behavior. Those agents together in a simulation, can form patterns in the whole system. Agent-based models are mostly used to get insight into complex, collective behavior, based on the simple rules implemented in each of the agents. Agent-based models, which also include the typical ant simulations, try to reach emergence. Emergence is also known in philosophy, systems theory, or art. It describes the evolution of a system composed of multiple parts, to form some complex behavior not seen in each individual piece. Emergence is also commonly expressed as “the whole is greater than the sum of its parts”.

In programming, the object-oriented paradigm is representatively a near-perfect fit for agent-based simulations because it allows to create blueprints for an agent with its behavior implemented in methods and member variables. It would also allow for the inheritance of behavior through polymorphism. The drawback of an object-oriented approach is performance related. Because of the instancing of the objects, a functional approach would generally be more efficient.

2.2. Cellular automaton (Conway’s Game of Life)

A cellular automaton is a simulation which consists of a grid with a finite number of states. Each generation, the cell updates its state according to some predefined rules. Those rules often define a new state for a cell depending on the neighbors of it. The neighbors of a cell can be the four cells that share the edges or eight cells including the ones that share corners with it. A popular cellular automaton is Conway’s game of life. Its rules are as follows:

“

- If the cell is alive, then it stays alive if it has either 2 or 3 live neighbors
- If the cell is dead, then it springs to life only in the case that it has 3 live neighbors

“ (Lipa, 2022)

The rules defined above cause the cells to die or reproduce by themselves. This enables the opportunity to simulate complex structures with easy rules. In fact, it allows for so much complexity, that Conway’s game of life is proven to be Turing complete. (Berto, 2012)

Although our simulation looks like a common cellular automaton, this is only because of the design as a grid with cells. Our ant simulation is not a cellular automaton because the cells do not change their state according to their neighbors, but rather the ants change their position according to the underlying information stored on the cells.

2.3. Neural agent evolution simulations

Since the beginning of programming, mankind tries to simulate the neural network of living beings. Such successful simulations may enable us to take a great leap in technology. It would enable us to automatically let machines teach themselves to adapt and overcome the limits set by humanity.

To showcase the basics of neuroevolution, Faustino Gomez, Juürgen Schmidhuber and Risto Miikkulainen wrote a Journal of Machine Learning in the year 2008, in which they wrote the following summary:

“The basic idea of Neuroevolution is to search the space of neural network policies directly using a genetic algorithm. In contrast to ontogenetic learning involving a single agent that learns incrementally (i.e., value-based RL), NE uses a population of solutions. The individual solutions are not modified during evaluation; instead, adaptation arises through repeatedly recombining the most fit individuals of the population in a kind of collective or phylogenetic learning. The population gradually improves as a whole until a sufficiently fit individual is found. In NE, neural network specifications are encoded in string representations or chromosomes. A chromosome can encode any relevant network parameter including synaptic weight values, number of processing units, connectivity (topology), learning rate, etc. These network genotypes are then evolved in a sequence of generations. Each generation each genotype is mapped to its network phenotype (i.e., the actual network), and then evaluated in the problem environment and awarded a fitness score that quantifies its performance in some desirable way.” (Faustino Gomez, 2008) inspired by (Yao, 1999)

It would take years of learning and practice, to simulate the fundamentals of neuroevolutionary. Due to this we did not implement such neural evolution and decided to hard code our ants in such a way, that they cannot evolve by themself and always follow the rules written by the programmers.

2.4. Travelling salesman problem

The traveling salesman problem simulation is well known for its simplicity which at the same time is extremely hard to optimize. The traveling salesman problem works with differently located checkpoints which could be interpreted as cities around the world to which the salesman has to travel to. To optimize the travel time the salesman searches for the best and shortest route through each city. However, it is very hard to find the perfect route, since the number of possible routes increases rapidly with each city/checkpoint. Such possibilities could easily reach numbers up to a few quintillions. As an example: If the simulation covers thirty cities, there are 30! possible routes, which is equal to **2.6525×10^{32}**. There is a way to calculate the perfect route with absolute certainty. When the number of checkpoints increases, it is no longer possible to calculate the perfect route with our computation power. It becomes more efficient to estimate the perfect route quite accurately with simulations like an ant simulation (Walker, 2018).

When looking at ant simulations, the situation is similar. Ants find the optimal path to their food sources. Now if ants would not go back to their nest and rather search for the optimal path to the next food source, we would have a way to approach the traveling salesman problem with an ant simulation.

2.5. Randomness in simulations

A deterministic simulation can be described in its outcome based on its initial state. There is no randomness involved, so the same initial state will always result in an identical course of the simulation. The course of a simulation will always be the same if no parameters are changed.

There are however problems, which are deterministic, that are solved with a simulation involving randomness. Those methods are called Monte Carlo methods and they use randomness in computational algorithms to solve problems which might be deterministic in principle. Those problems are often so complex that, with current calculation power, solutions can only be estimated with the help of simulations. In theory and with more calculation power, the solutions could be calculated. The Traveling Salesman problem is commonly solved with Monte Carlo methods.

Ant simulations are not deterministic because the movement of ants involves randomness. They use aspects of Monte Carlo methods, although they do not try to solve a problem which could be deterministic.

3. Implementation

The process and thoughts of implementing the ant simulation are described in the following chapters. The aim hereby is not to teach programming, but to explain the inner workings of the simulation and the process of developing the application.

The code can be found in the following GitHub Repository:
https://github.com/CediGasser/ant-simulation

As further changes could be made, a previous commit can be selected to access a version of the code before the submission date. The newest working versions are continuously deployed to: https://ants.cedricgasser.com/.

3.1. Requirements

The requirements of the project are defined in a user-story. This shows how the features should be implemented from the view of the user. A user-story has a title, a description, additions, an estimation, and acceptance criteria. The criteria show, which requirements must be implemented as a minimum to qualify the user-story as done.

Guiding questions for the description:

- Who?
- What?
- Why?

Estimation guide:

- Reflects the complexity of the story
- Is a Fibonacci Number

Description:

As a user, I want to organize a food scavenger ant simulation, in order to inspect their adaptability.

Acceptance criteria:

Table 1 defines what our simulation should look like and how the user experience should be:

Topic	**Given**	**When**	**Then**
Simulation	The user opens the web page	The simulation starts	The following items shall spawn on the field: One nest Min. five ants Min. five food sources Min. five obstacles
Simulation	The simulation starts	The objects spawn	Two items cannot be on the top of each other
Nest	The simulation started	The nest spawns	The nest is in the middle of the field
Ants	The simulation started	The ants spawn	The ants are next to the nest. The ants start to search for food
Ants	The simulation started	An ant finds food	The ant returns the food in multiple pieces to the nest The returning path of the ant will be marked
Ants	An ant began to return a food	Another ant finds the path of the first ant	The second ant helps to return the food
Obstacles	The ants are scavenging	An ant runs into an obstacle	The ant cannot cross the obstacle
UI Environment	The user opens the web page	The simulation started	The user has the following options to change in the simulation: Ant count Food count Obstacle count Ant type

Table 1: Acceptance criteria ant simulation

Addition:

All acceptance criteria can be visualized with a demo

Estimation: 13

Mockup:

To have a common understanding of what the features of the final product should look like, a design mockup for the website has been made. The following mockup should not dictate the implementation on a pixel-accurate level, but rather assist in creating and prioritizing implementation tasks. Figure 5 shows the mockup.

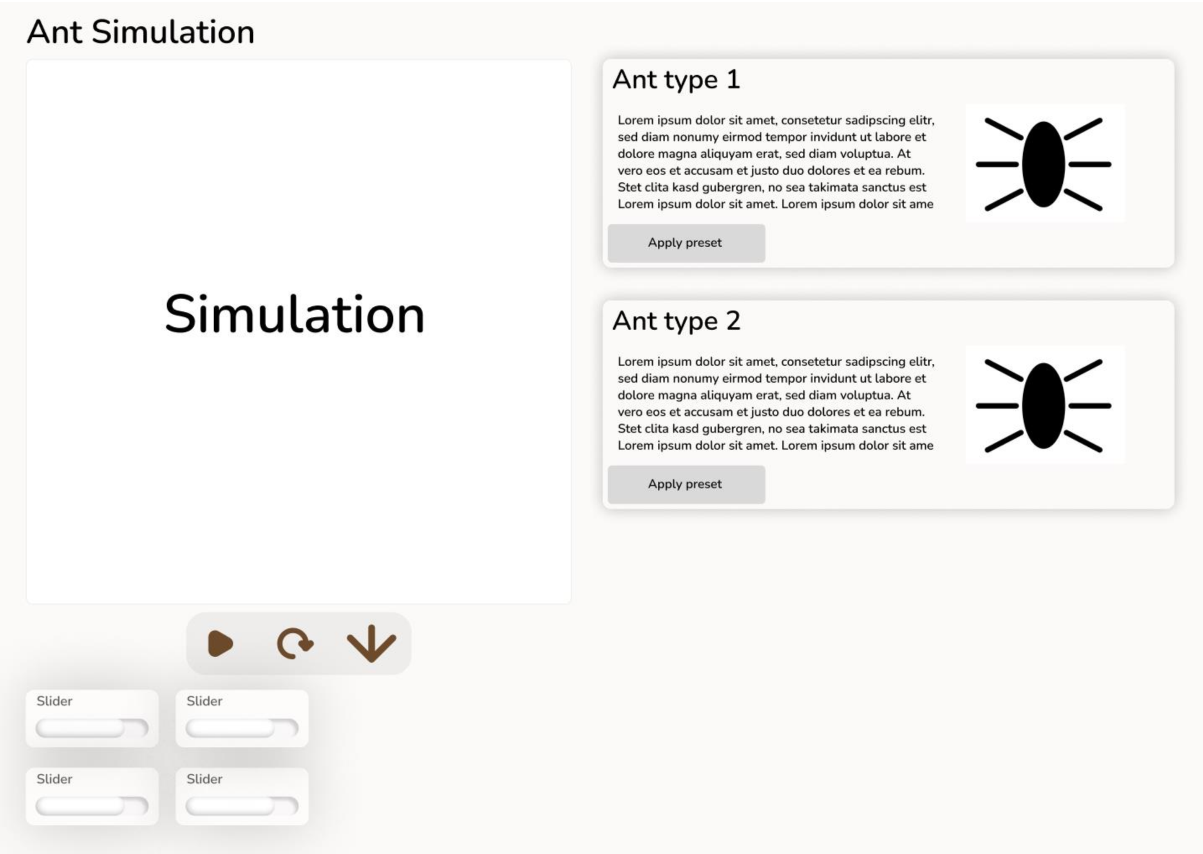


Figure 5: Mockup

3.2. Technology stack

In the initial phase of the project, important decisions have been made. Those include the frameworks and tools that we used.

To reduce the work with styles, to have a common design, and to have a basic protection against potential attacks, a UI framework had to be selected. Because of the improved page rendering and usability, the framework Svelte was chosen.

Basically all, directly in the browser executed code is written in JavaScript. Sometimes, it can be difficult to read JavaScript, that is why TypeScript is included. As the name says, it makes the difference between types easier. For example, a number can be simpler distinguished from a text. For the execution, TypeScript will be always converted to JavaScript.

To reach better performance, the rendering of images as a whole was out of question. Instead, the application should operate with pixels. P5.js is a package made for 2D rendering in the web and has Svelte support. A simulation that relies heavily on performance would best be implemented on a native layer like a desktop application. Although, because of accessibility and our commitment to provide our simulation as a website, we had to use the options we had in the web.

A version control tool helps the developers working in a team to access older versions of a project and it ensures ease of merging different versions when working in parallel. Nowadays, the best-known tool for version control is Git. With Git, the changes can be saved locally and to make them available for the developer team, they must be uploaded to a Git service provider, in our case GitHub.

We configured GitHub to automatically notify our server, where the application is hosted, when a new version is released. The server then builds a static version of the application which it hosts inside a Nginx Docker container. Nginx is used because it is a fast and reliable static web host. Everything runs inside of Docker containers for the ease of use and automation.

3.3. Design

All the colors in the simulation, are meant to be as realistic as possible. The following table shows the colors and what the colors try to represent. The color sets have two types for the two ant types included in the simulation.

Color	**Meaning**	**Simulation occurrence**
	grass	fire ant simulation
	meat	
	fire ants	
	nest (dirt)	
	stone	
	used path (darkest possible version of grass green)	
	parquet	carpenter ant simulation
	sugar	
	carpenter ants	
	nest (hole in the parquet)	
	wooden chunk	
	damaged parquet	
	pheromone	both

Table 2: Simulation colors

3.4. Ant pictures

In order to design the website more interesting, we added, among other things, one drawn picture of each simulated species (figure 1 and 2 / 6 and 7 / 14 and 15). The pictures were drawn on a tablet with a digital pen. The free digital drawing app ibisPaint X was used to do so. With the goal of a realistic picture, the first step was to sketch the skeleton of the ant, to establish the right proportions. The next step was to clean up the sketch, so that there were only clear lines left, precisely showing every detail of the ant. The third and most time-consuming step was the coloring: Trying to realistically and accurately draw every color while making smooth transitions between the different colors. It turned out, that combining dark, almost black, coloration with silver-white coloration (as seen on figure 7) was particularly difficult and time-consuming. Because of that, it almost took 12 hours to create the carpenter ant while in comparison the fire ant, with its brown-reddish coloration (on figure 6), only took four and a half hours.

Figure 6: Fire ant

Figure 7: Carpenter ant

3.5. Ant behavior

The whole field is created out of cells. It is the same as a coordinate-system, it has a height and width of cells. The nest is always located in the middle of the field.

3.5.1. Legend

This legend is a guide to the upcoming figures in this chapter. The field colors are the same as in chapter 3.3 Design. In the illustrations, some new symbols can be found as shown in table 3.

Symbol	Meaning
	The previous or the next steps of an ant
	The storage of a value in the cell, the ant stands on

Table 3: Behavior legend

3.5.2. Scavenger mode

As soon as the simulation starts, all ants are in scavenger mode. At this state, they are looking for food by randomly stepping to a cell next to them. An ant has always eight potential cells to step on, if there is no obstacle nearby. These fields are shown on figure 8 below:

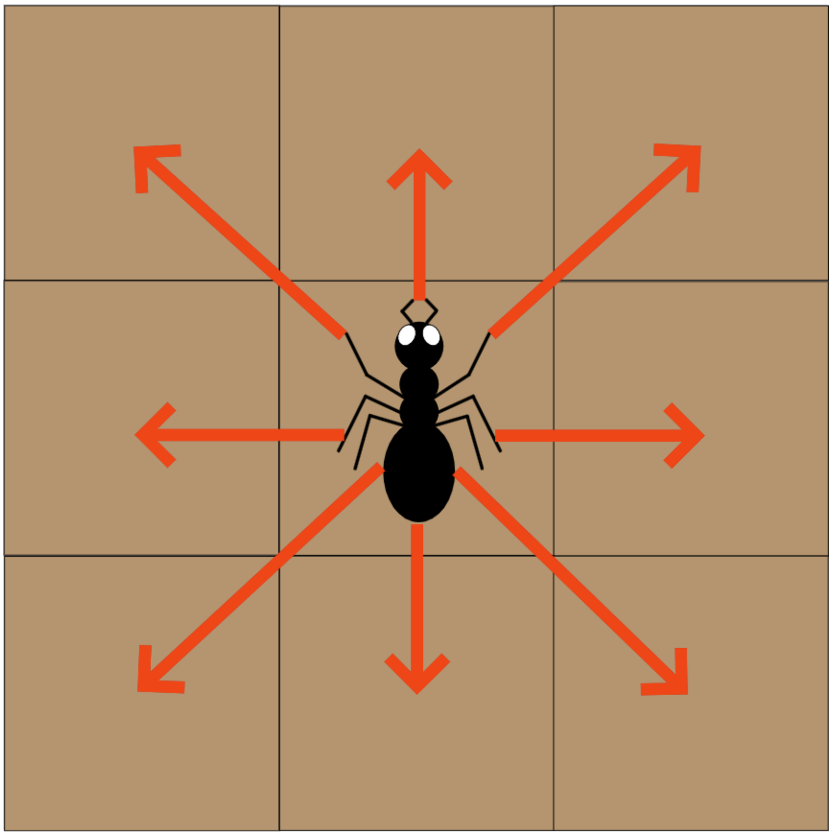


Figure 8: Ant directions

3.5.3. Ant-nest distance

As an ant changes its position, its distance to the nest will be updated. This number means the count of steps since it left the nest. Cells, which have been crossed, will save this number. If the ant is moving horizontally or vertically, the distance will be updated by one if diagonally, the increment will be two. The saved numbers can be overwritten, if another ant crosses the cell with a shorter distance than the ant before.

In the following illustration, figure 9 shows the path of an ant. Each affected field has a value saved in it.

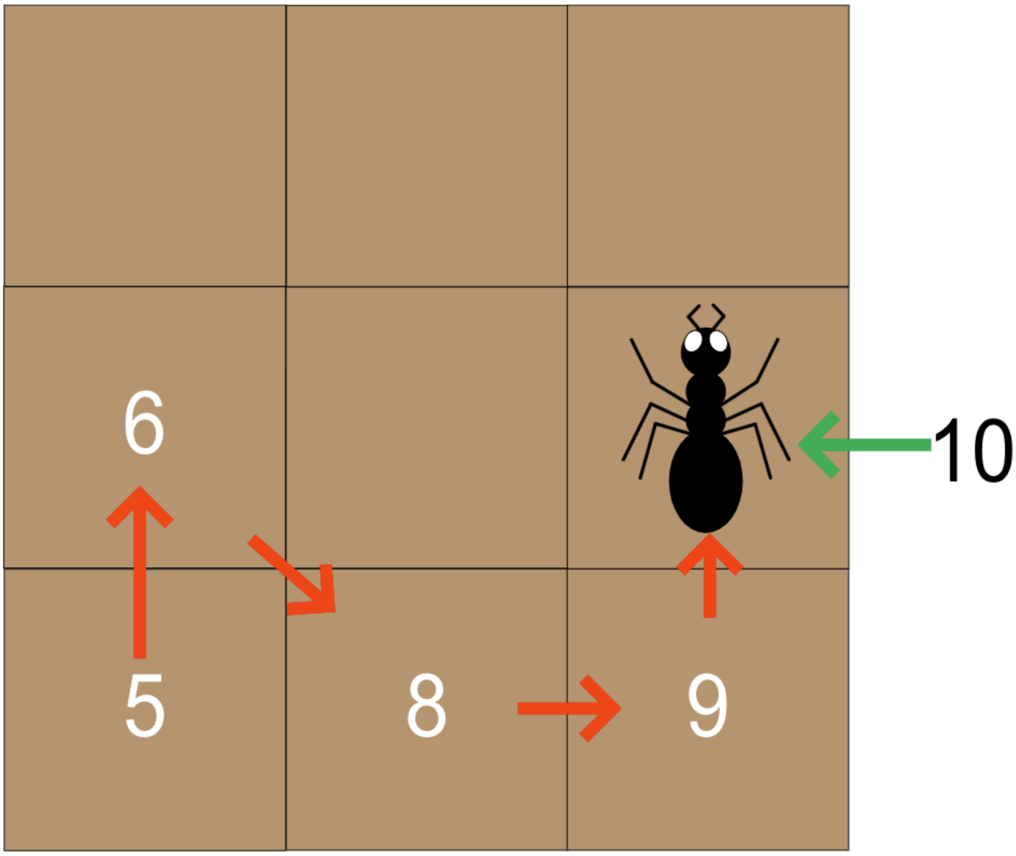


Figure 9: Scavenger ant distance

3.5.4. Delivery mode & pheromone trail

As soon as an ant reaches a cell with food on it, it turns into delivery mode. That means, that afterwards it always seeks the cell with the least nest distance value until it reaches the nest.

During the food delivery, the ants leave a pheromone trail on the cell they step onto. This works similar to the ant-nest distance. Instead of the ant-nest distance, they save the food-ant distance in the pheromone-colored cells. If an ant in scavenger mode is near a trail, it follows it to the food, then back to the nest.

Figure 10 illustrates an ant heading back to the nest and making a pheromone trail after finding food. After leaving the previous pheromone cell, the value seven will be saved in the actual, grey-colored cell. As next, the ant must take the field with a ten in it, because it is the lowest ant-nest distance value from the options.

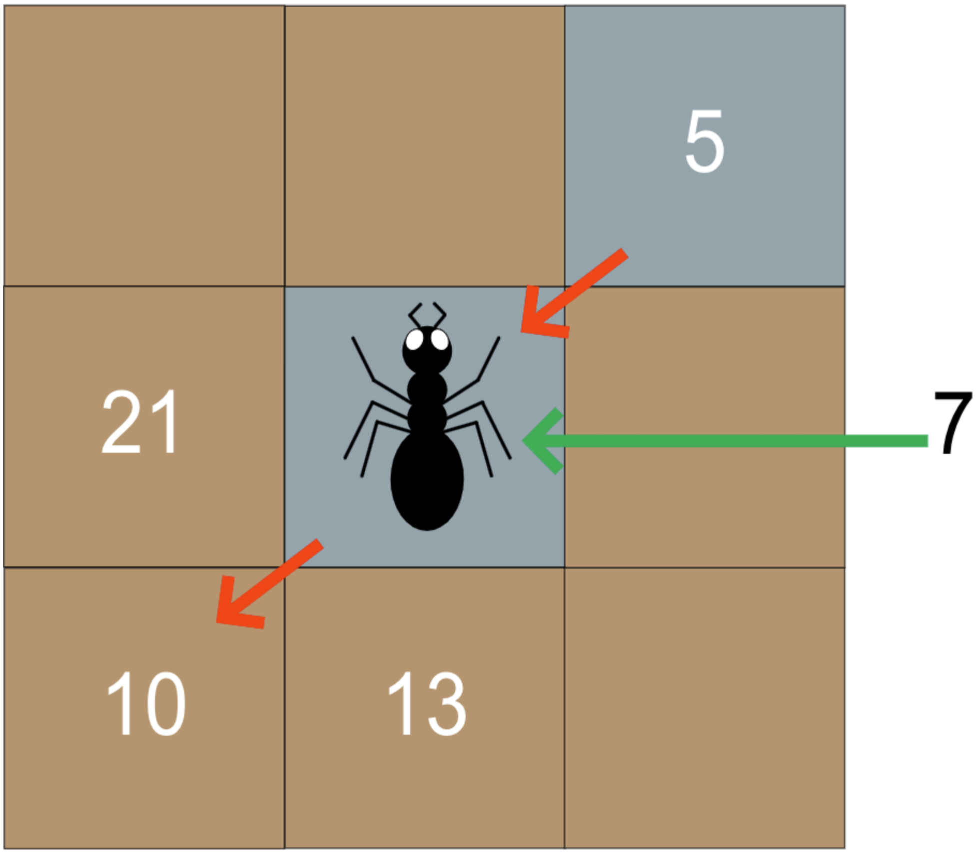


Figure 10: Deliverer ant distance

3.6. Coding standards

Since for the most part of the simulation development we used graphical elements, we realized that all entities (ants, cells, food, nest, and obstacles) of the simulation must be partially handled the same way. We linked them with inheritance, in other words we ensured, that some properties must be found in all of them, and they all must be updated and rendered.

We tried to use the Clean Code standards, since the code needs to be open for new changes and readable. To reach this, TypeScript helped a lot with the detailed definitions of functions and data types.

This project is a UI-based application; it would have been difficult to use units for tests. Only UI tests would have worked, and they were not of high importance because of the few svelte components. For that reason, we tested the application only manually.

3.7. Requirements evaluation

After finishing the implementation, the product had to be compared with the user-story, to ensure the minimal needed functionality of the application. The comparison of the requirements and the features of the app was successful, and all the requirements were fulfilled.

3.8. Feature stack

After finishing the user-story (the minimal requirements), we decided to make some extensions to the application. These were mostly small changes, so they did not get an additional user-story. In this chapter will be all the actual features presented.

First of all, the product supports all devices. Thanks to its responsive design the webapp can be run on phones too. It supports all the modern browsers.

As soon as the user visits the webpage, the simulation starts, and the ants begin to move. The only task of the simulation component is, to show the simulation. The nest and the ants start always in the middle of the field. The obstacles and food sources spawn randomly. The simulation has two possible color sets. The default set is presented on figure 11, but during the simulation, the colors can be changed to the set visible on figure 12.

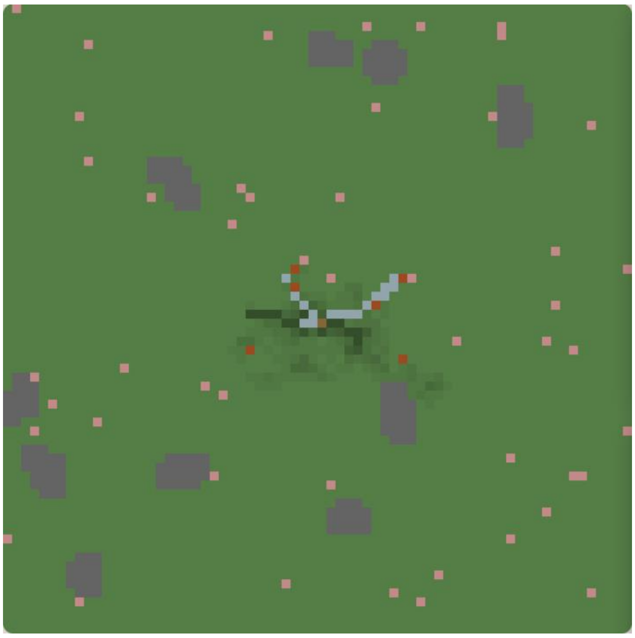


Figure 12: Fire ant color set

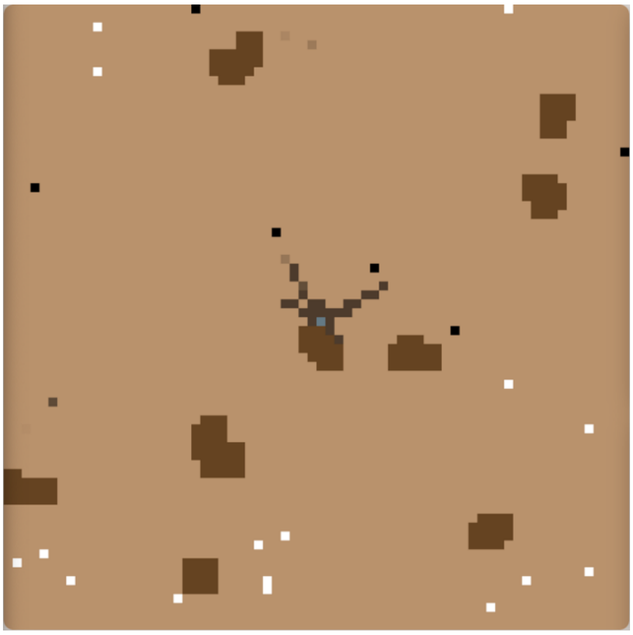


Figure 11: Carpenter ant color set

Below the simulation is a button bar and a fps displayer located as shown on figure 13. The "PAUSE" button stops the ants and changes into a "PLAY" button. By pressing it, the simulation will continue. If the "RESET" button gets pressed, the simulation will restart, and the obstacles and food sources will get a new location. The fps displayer shows the current fps value for the entire webpage.

Figure 13: Button bar and fps displayer

On the screen are the two ant types with each an image and two buttons displayed, as presented on figures 14 and 15 below. By pressing the “APPLY … ANTS” button, the color set for the actual ant will be activated and the simulation reloads. By clicking on “ABOUT”, the user comes to another page, where additional information about the ant can be read. The texts are shorter versions of chapter 1.2 “Fire ant” and chapter 1.3 “Carpenter ant”.

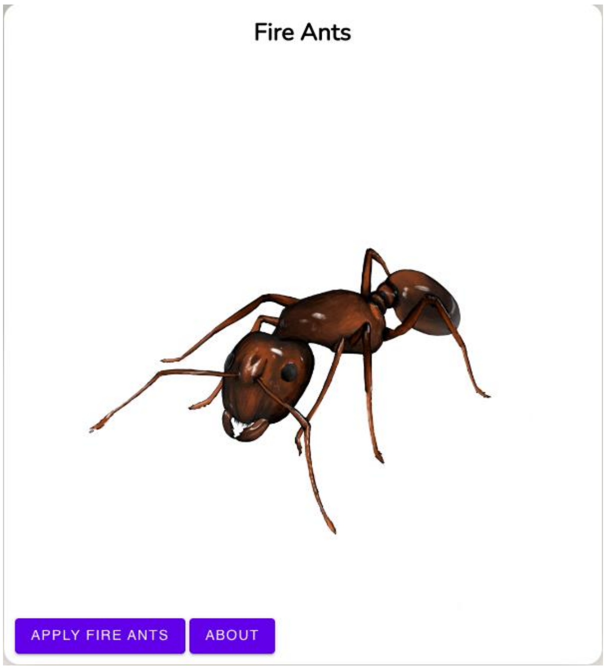


Figure 14: Fire ant activator

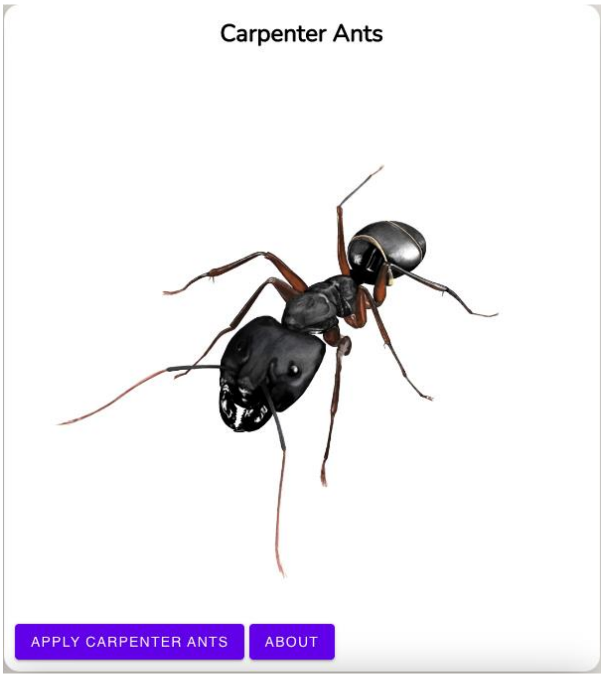


Figure 15: Carpenter ant activator

The best way to influence the simulation is through the sliders possible. With them, the parameters on figure 16 can be changed. By changing a slider, the simulation will restart with the new set value. Each slider has an individual range.

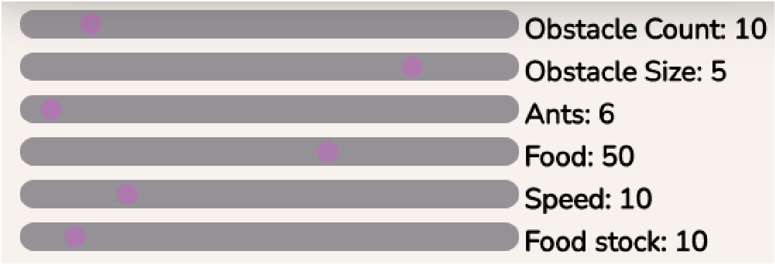


Figure 16: Parameter sliders

4. Findings

After finishing the implementation of the simulation, the app was ready to make findings.

To define our findings, we split them into two sections. One of them describes the path creation and how it gets optimized. The second one shows the adaptability of ants if we place more obstacles in their paths which holds them back from gathering food.

4.1. Use of ant paths

As we let the ants simulate with more food units but lesser quantity per source, the ants did not have time to create the most efficient paths, since they could not adapt from the other ants before them. As shown in figure 17, the ants created a lot of paths but did not have time to make them efficient. After we reversed the settings, the ants seemed to create quite structured paths which got optimized every time an ant went to the same food source. The ants however took longer to find the rare food which affected the efficiency in a negative way. A reoccurring theme in both variants was that the ants often lost the paths and abandoned them, since they could not find their way back. Just like in figure 18, the ants created the most efficient path possible and redid the same path repeatedly until they gathered all the food or found another food source.

By measuring the time until all the food has been collected, we have a performance indicator for the efficiency of the ants in food gathering. Following measurements show that fewer food sources with more stock tend to be more efficient for the ants. This statement however must be taken with caution because we suspect that those results could vary heavily depending on the way and speed the pheromone trails are built and disappear. The results are shown in table 4.

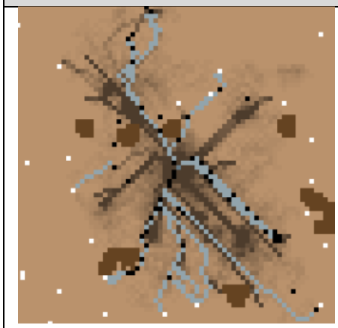
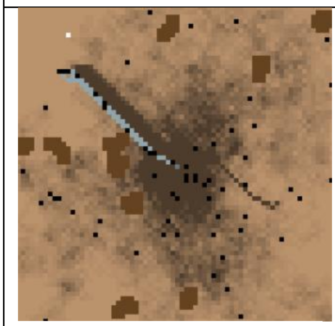
Figure	Parameters	Average time of gathering all food sources out of ten runs
 *Figure 17: More food, lesser food stock*	Obstacle Count: 10 Obstacle size: 5 Ants: 70 Food: 50 Speed: 30 Food stock: 10	59 seconds
 *Figure 18: Less food, more food stock*	Obstacle Count: 10 Obstacle size: 5 Ants: 70 Food: 5 Speed: 30 Food stock: 100	57 seconds

Table 4: Ant path findings

4.2. Optimizing of paths

By using the “Obstacle Count” and the “Obstacle Size” slider, we had the opportunity to put obstacles in the way of the ants. Because of this, the ants had to find their way around the obstacles to find new food sources and bring them back to their nest. Just like in chapter “4.1. Use of ant paths” the ants needed several attempts to create the most optimized path from the food source to the nest. Such optimizations included completely different paths around the obstacle which sometimes even led to multiple paths to the same food source. As shown in figure 19 the ants found their way around the obstacles and brought back the food with the most optimized path.

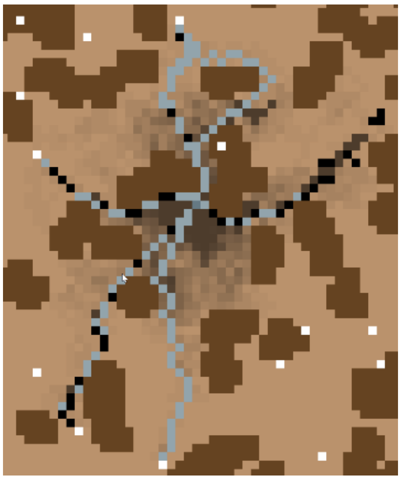


Figure 19: Adaptability of ants

5. Summary

The application we made shows the answer to our thesis “How can the adaptability of ants be simulated?”. The usage of this simulation is, to imitate the adaptability of the ants as they optimize their paths between the food and the nest. The final product of this project appeals to people with interests in ants and their behavior or simulations and how they are programmed. It turns out that the algorithms used to simulate ant behavior can also be used to solve a wide variety of optimization problems including traffic or network routing as well as scheduling problems amongst others.

It is to say, that our ant simulation is a heavily simplified simulation of the social behaviors of real ants. The reason for that is the amount of work that would be necessary to simulate such reality, which is immense and almost impossible in such a short time without deep understanding in a broad class of subjects. We did however manage to simulate the basic adaptability of ants while searching for food using their pheromone trails to navigate. Our application also shows the improvements of the paths each time a new ant walks from the nest to the food source. This demonstrates the efficiency of ants in our simulation.

Our Simulation still could have a few improvements to make it more realistic and more informative. The following paragraphs show several improvements or ideas which did not fit into the project scope.

We wanted to add a feature called “Dying and rebirth of ants”. By that, we mean an implementation where ants die if they do not find something to eat in time. But if they do, they could reproduce and grow in population. When adding regrowing food sources, this could simulate an equilibrium where the simulation becomes stable around a balance between population size and food availability.

We would have also liked to implement predators, like spiders or other colonies to simulate the behaviors between different conflicting species. Such addition to the simulation would bring way more signs of life and would salient the reality of nature.

We thought of an addition of a new painting tool which would have brought the opportunity to draw the obstacles by hand during a running simulation. Such a tool would have created a more interactive experience for the simulation and the users would have the option to influence the ants to create or adapt their paths.

To add more realism to our simulation, more different parameters, and a slight change in the logic of the ants would be needed. Additional parameters could be for example the lifetime of pheromone trails. Then, the parameters could be set according to the ant types or manually.

To enable the species to really behave differently compared to others, additional behaviors, maybe dynamically assignable, would have to be implemented. This could include a larger sensory distance for ants or a way to systematically search the field instead of utilizing completely random movements.

6. Glossary

Since we used a lot of technical words, we list them in table 5 together with a small explanation.

Expression	Meaning
Clean Code	A list of practices which make the code developer friendly.
Deterministic	The final state can be calculated out of the initial parameters.
Emergence	Complex behavior as a whole results from simple rules applied to parts.
Object-oriented programming (OOP)	A programming paradigm where things with their behavior and properties are created in code.
Pheromone	A chemical, which is used by the ants to signal other ants. Different pheromones can signal different messages.
Stylesheet	The styles of the webpage are written in stylesheets.
Unit tests	A coded test case by the developer, which asserts the functionality of features without running the application.

Table 5: Glossary

7. Bibliography

Berto, F. (2012, 03 26). *Plato Stanford*. Retrieved from Cellular Automata: https://plato.stanford.edu/entries/cellular-automata/

biologie-seite.de. (2022). *biologie-seite.de*. Retrieved from Rote Feuerameise: https://www.biologie-seite.de/Biologie/Rote_Feuerameise

Cassidy, J. (2019, 9 24). *kqed.org*. Retrieved from Kidnapper Ants Steal Other Ants' Babies — And Brainwash Them: https://www.kqed.org/science/1947369/kidnapper-ants-steal-other-ants-babies-and-brainwash-them

Faustino Gomez, e. a. (2008). *people.idsia.ch.* Retrieved from gomez08a.pdf: https://attentionagent.github.io/

Garinger. (2019, 03 01). *umweltbundesamt.de*. Retrieved from Holzameise: https://www.umweltbundesamt.de/holzameise#vorsorge-und-chemischen-und-alternative-bekampfungsmassnahmen

Hadley, D. (2020, 8 27). *thoughtco.com*. Retrieved from Carpenter Ants, Genus Camponotus: https://www.thoughtco.com/carpenter-ants-genus-camponotus-1968094#:~:text=Carpenter%20ants%20undergo%20complete%20metamorphosis,females%20establish%20a%20new%20colony.

Iilende. (n. d.). *coursehero.* Retrieved from Thus-a-model-is-a-simplified-representation-of-a-real-world-situation-tha: https://www.coursehero.com/file/p26rjdd/Thus-a-model-is-a-simplified-representation-of-a-real-world-situation-that/

Kiesewetter, B. (2020, October 5). *tierlexikon*. Retrieved from kindernetz.de: https://www.kindernetz.de/wissen/tierlexikon/steckbrief-ameise-100.html

Kress, U. (2021). *lexikon-der-schaedlinge.de*. Retrieved from Glänzendschwarze Holzameise: https://www.lexikon-der-schaedlinge.de/glaenzendschwarze-holzameise/

Lipa, C. (2022). *Pi.math.cornell.edu*. Retrieved from Conway's Game of Life: http://pi.math.cornell.edu/~lipa/mec/lesson6.html

Ltd, T. (n. D.). *twi-global.* Retrieved from faq-what-is-simulation: https://www.twi-global.com/technical-knowledge/faqs/faq-what-is-simulation

Ludwig, M. (2019, 11 05). *tierwelt.ch*. Retrieved from Die Unbesiegbare: https://www.tierwelt.ch/artikel/wildtiere/die-unbesiegbare-408511

mefics.org. (2020, 11 01). *mefics.org*. Retrieved from Rote Ameisen: https://mefics.org/rote-ameisen/

Nanaimo Pest Control. (2016, 2 12). *nanaimopestcontrol.com.* Retrieved from Carpenter Ant Parent Colony Trail: http://nanaimopestcontrol.com/wp-content/uploads/2015/01/E2DyBa6ZckmSrHh4ERpYEK5scQT6hyx8YPfsEARXfseqT21BXrQSzbtXa2j2dhKFf-576x1024.jpg

Ratchford, D. (2017, 9 1). *howstuffworks.com.* Retrieved from Floating Fire Ant Rafts Are Horrifying, Dangerous, Really Cool:

https://media.hswstatic.com/eyJidWNrZXQiOiJjb250ZW50Lmhzd3N0YXRpYy5jb20iLCJrZXkiOiJnaWZcL2ZpcmVhbnRyYWZ0LWZsaWNrcmRvcmlzcmF0Y2hmb3JkLmpwZyIsImVkaXRzIjp7InJlc2l6ZSI6eyJ3aWR0aCI6ODI4fSwidG9Gb3JtYXQiOiJhdmlmIn19

schaedlingskunde.de. (n.d.). *schaedlingskunde.de*. Retrieved from Glänzendschwarze Holzameise - Lasius fuliginosus: https://schaedlingskunde.de/schaedlinge/steckbriefe/ameisen/glaenzendschwarze-holzameise-lasius-fuliginosus/glaenzendschwarze-holzameise-lasius-fuliginosus/

schule-bw.de. (n.d.). *Die Verständigung der Ameisen*. Retrieved from schule-bw.de: https://www.schule-bw.de/faecher-und-schularten/mathematisch-naturwissenschaftliche-faecher/biologie/unterrichtsmaterialien/um56bnt/wirbellos/insekt/ameisen/verstaendigung.html

Shields, J. (2017, 9 1). *howstuffworks.com*. Retrieved from Floating Fire Ant Rafts Are Horrifying, Dangerous, Really Cool: https://animals.howstuffworks.com/insects/fire-ant-rafts-colony-flooding.htm

Thanukos, A. (n.d.). *berkeley.edu*. Retrieved from Understanding the strategies: https://evolution.berkeley.edu/fire-ants-invade-and-evolve/understanding-the-strategies/

The Field Museum. (n.d.). *expeditions.fieldmuseum.org*. Retrieved from ANT EVOLUTION AND THE ENVIRONMENT: https://expeditions.fieldmuseum.org/australian-ants/ant-evolution-and-environment#:~:text=Ants%20first%20appeared%20on%20the,species%20like%20pines%20and%20ferns.

Walker, J. (2018, Juni). *fourmilab.ch*. Retrieved from Simulated Annealing: https://www.fourmilab.ch/documents/travelling/anneal/

Wild, A. (2008, 9 13). *alexanderwild.com.* Retrieved from Polyergus: https://www.alexanderwild.com/Ants/Taxonomic-List-of-Ant-Genera/Polyergus/i-Nm95qN2/A

Yao, X. (1999). *avellano.fis.usal.es.* Retrieved from yao99evolving.pdf: http://avellano.fis.usal.es/~lalonso/compt_soft/articulos/yao99evolving.pdf

8. Table of tables

Table 1: Acceptance criteria ant simulation....14
Table 2: Simulation colors....17
Table 3: Behavior legend....19
Table 4: Ant path findings....25
Table 5: Glossary....28

9. Table of figures

Figure 1: Fire ant....5
Figure 2: Carpenter ant....6
Figure 3: Fire Ant Raft (Ratchford, 2017)....7
Figure 4: Carpenter ant trail (Nanaimo Pest Control, 2016)....8
Figure 5: Mockup....15
Figure 6: Fire ant....18
Figure 7: Carpenter ant....18
Figure 8: Ant directions....19
Figure 9: Scavenger ant distance....20
Figure 10: Deliverer ant distance....21
Figure 11: Carpenter ant color set....23
Figure 12: Fire ant color set....23
Figure 13: Button bar and fps displayer....23
Figure 14: Fire ant activator....24
Figure 15: Carpenter ant activator....24
Figure 16: Parameter sliders....24
Figure 17: More food, lesser food stock....25
Figure 18: Less food, more food stock....25
Figure 19: Adaptability of ants....26

10. Statutory Declaration

We hereby declare that we have prepared the present interdisciplinary project work independently and without any unauthorized external help and that all sources, aids and internet sites have been used truthfully and are documented.

Luzern, 23.05.2022

Cédric Gasser	Dániel Polgár
Vincent Hischier	Dominik Costa