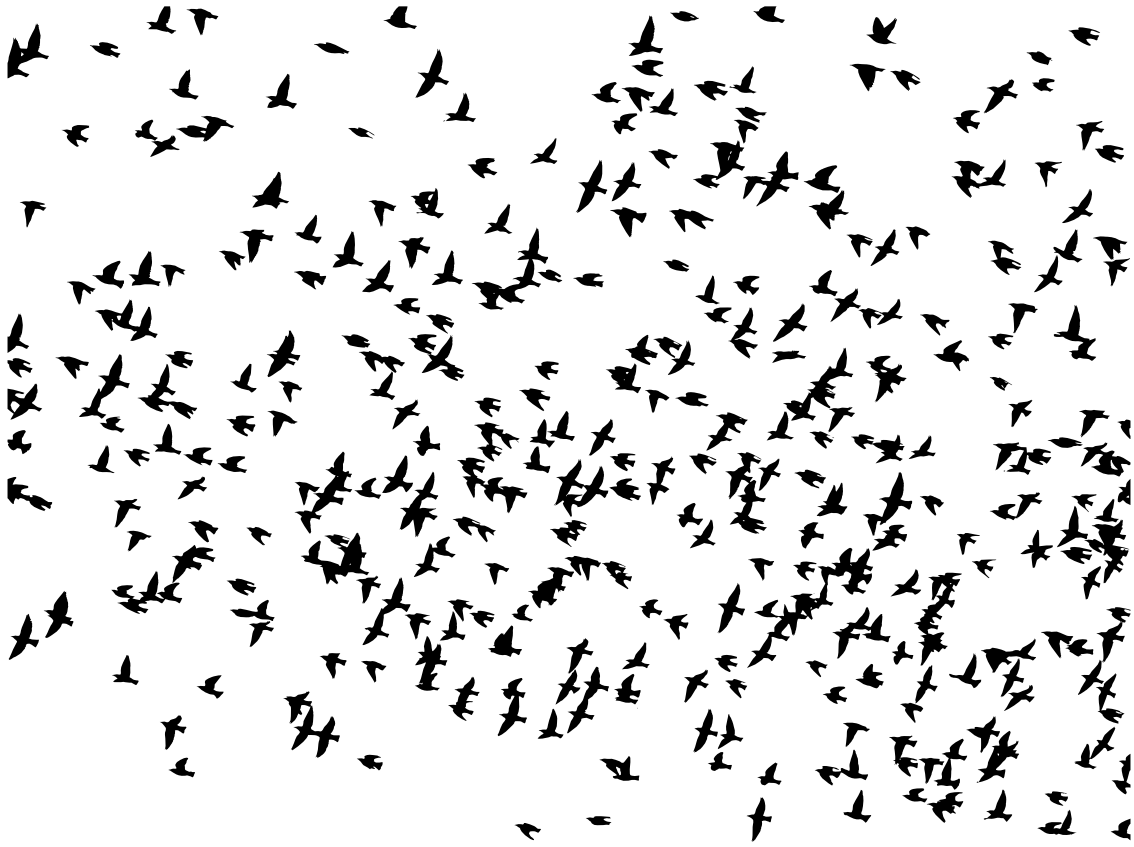
CHALMERS
UNIVERSITY OF TECHNOLOGY



Rules and Consequences

Bachelor's thesis

Group 21:
William Hjelm, Karl Strigén,
Samuel Håkansson, Simon Sundström,
Felix Jansson & Daniel Heurlin

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

Bachelor's thesis 2018:datx02-18-21

Rules and Consequences

Using NetLogo to Explore Behavior in Multi-Agent Systems

Group 21:
William Hjelm, Karl Strigén,
Samuel Håkansson, Simon Sundström,
Felix Jansson & Daniel Heurlin

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
Gothenburg, Sweden 2018

Rules and Consequences
Using NetLogo to Explore Behavior in Multi-Agent Systems
William Hjelm, Karl Strigén,
Samuel Håkansson, Simon Sundström,
Felix Jansson & Daniel Heurlin

Supervisor: K V S Prasad, Department of Computer Science and Engineering
Examiner: Our examiner wishes to remain anonymous in this report

Bachelor's thesis 2018: DATX02-18-21
Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Birds flying in flock.

Rules and Consequences
Using NetLogo to Explore Behavior in Multi-Agent Systems

William Hjelm, Karl Strigén,
Samuel Håkansson, Simon Sundström,
Felix Jansson & Daniel Heurlin, 2018.
Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

Emergence can be described as the result of a system that has properties greater than the sum of its parts. The result and individual actions are not traceable through each step of the process, but instead *emerge* from the behavior of all agents. Examples of emergence can be found in nature, such as birds flying in strange patterns and ants foraging for food. Understanding these systems is hard to for the human intuition, but since computer science enables techniques to examine emergence by introducing simulation tools we can see the results and get a deeper understanding of the system and its rules.

This thesis investigates what happens when the programmer does not micromanage how each component in the system interacts with other components. When multiple parts work together using simple rules, we sometimes get phenomena that we could not predict. This is what we call *rules and consequences*; when the rules for the components of the system give rise to emergent phenomena.

We have produced 25 different models and analyzed them using a guiding principle we refer to as *disciplined exploration*, meaning two things. First, results are not discarded even if they do not show what was intended or expected. Second, we systematically map out the scope for the parameters, as opposed to randomly choosing, by defining a "parameter space".

The thesis also provides insights about modeling systems this way using NetLogo. While it is a simple and easy tool to use, we also show that it does have drawbacks. Its internal scheduling and time complexity issues are problematic. Lastly, we provide conclusions to why emergent behavior is interesting and surprising. Emergence can be used to explain *how* complex system domains *could* be in reality and is always of interest due to our limited intuition of what will come out of a multi-agent system.

Keywords: NetLogo, emergence, agent-based modeling, agent, multi-agent systems, rules, consequences.

Sammandrag

Emergens beskrivs som ett resultat hos ett system som överskrider summan av dess delar. Resultatet är i det här fallet inte stegvis spårbart utan uppstår istället från agerandet hos samtliga agenter. Exempel på emergens i naturen är näst intill oändliga med allt ifrån svärmbeteenden hos fåglar till myrkolonier som letar efter mat. Att förstå systemen på mikronivå är ofta svårt med endast mänsklig intuition men genom att introducera simuleringsverktyg kan man studera resultatet och skaffa sig en djupare förståelse av systemet i sin helhet.

Vår tes undersöker vad som händer när programmeraren inte hanterar hur varje enskild komponent interagerar med andra komponenter. När flera delar samverkar under enkla regler får vi ibland beteenden som överraskar oss och detta kallar vi för *regler och konsekvenser*. Systemets regler ger upphov till emergens.

Vi har skapat och analyserat 25 olika modeller med hjälp av principen om disciplinerad utforskning och med detta syftar vi i huvudsak på två saker. Till en början så har alla resultat i detta projekt inte kasserats oavsett utfall. Samt att vi beskrivit och begränsat modeller systematiskt istället för att göra slumpmässiga val.

Vår tes bidrar även med insikter om systemmodellering med hjälp av NetLogo samt en diskussion om för- och nackdelar hos detta verktyg som främst behandlar schemaläggnings- och tidskomplexitetsrelaterade problem. Till sist så drar vi egna slutsatser om varför emergens och dess icketriviala beteenden är intressanta och ibland förvånande.

Acknowledgements

We would like to thank our supervisor K V S Prasad who has helped us during the project and given us many interesting ideas to model as well as fascinating conversations about everything under the sun.

William Hjelm, Karl Strigén, Samuel Håkansson, Simon Sundström, Felix Jansson & Daniel Heurlin , Gothenburg, June 2018

Contents

Glossary

Agent In the context of agent-based simulation, an agent is an autonomous decision-making entity.

BehaviorSpace A plugin for NetLogo that is used for running simulations multiple times and collecting data.

Reporter A kind of function in NetLogo that returns a value.

Scripting Language A language that is domain-specific and usually interpreted rather than compiled.

Turtle An agent that can move around the world.

1

Introduction

The world is full of strange and complex phenomena which might be hard to explain. It can be anything from the flocking of birds to the workings of world economics. However, these behaviors are not always as complex as seem at a first glance, or rather; the reason for the behavior might not be complex. When a system, such as the flocking of birds, moves without central coordination, one can say that the movement of the system as a whole *emerges* from the movement of each bird. Then a couple of questions arise: What are the rules that each bird follows? How well can reality be modeled by defining simple rules that give consequences for the system as a whole?

For notes on the references used in the project, see appendix A.

1.1 Background

The flocking of birds described above can be described as a multi-agent system. The same goes for many other things, both in nature and in society. Creating many models of multi-agent systems, referred to as "agent-based modeling" [1], has been the focus of this project. The benefit of agent-based modeling as opposed to other modeling techniques is the fact that it shows emergent behavior, as described by E. Bonabeau [2]. Only when looking at the combined components of a system which follows given rules, are we able to observe arising emergent behavior, which by this can not be achieved without multi-agent systems. Other interesting uses of agent-based modeling are for example:

- Trying to mimic a complex system can lead to a deeper understanding of its inner workings; what rules the agents might be following, what conditions must hold for the phenomenon to function as expected and the implications of altering parameters such as the environment, the number of agents, and starting conditions [2].

- In a system consisting of many independent agents, it may be desirable to achieve a certain behavior without the need for any central coordination. Simulating this system makes it possible to develop a set of rules for each agent

to follow which allows the desired behavior to emerge.

- In an environment where many independent agents interact unexpected emergent behavior can arise when new rules are introduced. Studying this could, for example, lead to a better understanding of how policy making in society and nature might have consequences other than those intended [2].

Emergence

The term emergence, as it is used in modern science, was coined in 1875 by the English philosopher G.H. Lewes [3]. It was originally intended as an account of Darwin's views of evolution. Emergence being thought of as the result of a special kind of process. According to R. Goldstein, the result is not traceable through each step of the process and the individual actions of each component, [4]. The concept was however observed even before the term was introduced, one notable instance of this is by Friedrich Engels in "The condition of the Working Class in England" [5].

We have seen that emergent properties arise in nature; life itself can actually be seen as emergence. So what is the role of computers and computer science in studies of this? A limitation when studying emergence in nature is the fact that it takes very long time before the results are seen. This holds for example regarding studies of evolution and genetics.

Experiments about how living organisms can adapt and overcome changes within their living environment have been made. One popular example of this are mutation analysis of Drosophila melanogaster (the common fruit fly) done by R. J. Konopka and S. Benzer [6]. Choosing flies for the experiment is based on their short living cycle, which is important to enable the appearance of mutations within a reasonable timeframe. Using computers instead, the results can be generated much faster.

Another thing that makes computer simulations of emergent behavior worthwhile is that results can be generated before they occur in real life. For example, when simulating economic systems, it is beneficial to know what will happen before it actually occurs.

Simply, the main reason computer science in is useful in studies of emergence is that the computational power of computers gives new possibilities in examining these systems through the construction of multi-agent simulations.

1.2 Purpose

The purpose of this project is to examine to what extent we can create multi-agent models with simple rules that produce interesting consequences; either visually or statistically.

1.3 Problem

It should be possible to describe simple models by listing rules that each agent follows. This allows models to be created quickly, allowing us to produce results at an early stage of the project. It is also good that each model can be created without extensive domain knowledge, which ensures that the focus of the project is on the multi-agent modeling and not on the study of the domains.

To make the results of the project interesting it will be beneficial if the emergent behavior of the models have somewhat of a "wow-factor", something that is unexpected and visually interesting. Therefore it is advantageous to have many models early in the project because it is not clear which of them could be interesting to present or draw conclusions from.

1.3.1 Choice of modeling software

To create models, some environment for programming is needed. We aimed to find a piece of software or programming language that is familiar or easy to learn due to the time frame. Representation of the results will be important, therefore a way to visualize the result must be provided or created. Running a simulation of a model multiple times while collecting data is preferable but not a deciding factor.

1.3.2 Modeling

Once it has been chosen what phenomenon to model, the challenge is to express the phenomenon as a set of simple rules. This amounts to mathematical modeling of real life behavior. Often this can be tricky to accomplish, since real life interaction tend to have many variables and unknown factors.

1.3.3 Delimitations

The scope of the project is to find simple rules which generate interesting consequences. Therefore, one delimitation is to not choose models with intricate real

world domains where simplifications are nontrivial. A consequence of simplifications is that in many cases, the models are drastically simplified versions of their real-world counterparts. Naturally, this limits the extent to which they can be used to gain understanding of the real systems.

Another delimitation of the project is that emergence is in the eye of the beholder. This means that ultimately what results will be deemed interesting and what emergence is examined, is based on what the beholder observes. By this, there is always a risk that things are missed. It is however the case that the term emergence is dependant on an observer, since the agents themselves have no means of observing the system on a macro level, as theorized by J. L. Dessalles, J. P. Müller, and D. Phan [7], but more on the properties of emergence in 2.1.

Implementing agents using a neural network could be highly interesting, as it most likely would allow models that more accurately mimic real situations to be created. Nevertheless, this possibility will not be explored since it is outside of the time frame of the project.

Lastly, this project will only use one software to simulate the multi agent systems. Even though we hope that a software will be found that fulfills the projects needs, it is still the fact that even more varied models could have been created if other software would be used. However, it seems to be a better choice to learn one instead of grasp the basics of many different.

1.4 Social and Ethical Aspects

The relevance of social and ethical aspects for this project is the fact that emergent behavior arises from social situations. While we make no claim that our simulations and models will mirror this behavior completely, it is important to be as correct as possible. A sensitive subject where emergence arises could be the grouping of people of different socioeconomic backgrounds in cities.

This project does not seem to carry the possibility of causing any concrete harm. However, when regarding different social scenarios there might be a possibility to do very rough generalizations regarding different types of persons. In this case, a project like this might run the risk of increasing existing stereotypes.

Models based on social phenomena will be considered more carefully and we do by no means imply that our models are a correct representation of real world issues and should not be treated as that.

Another issue that may have ethical implications is that emergence, by nature is unpredictable. By [3], the tools needed to accurately analyze systems with emergent properties do not currently exist, meaning that it is not possible to prove formally that a system will behave in a certain way. By this we mean to say that observations

made about the behavior of a model are not to be seen as proofs that the model will always exhibit this behavior.

Some models might have unexpected behaviors that could be very dangerous in certain cases. Therefore, if the model is used in situations where people may come to harm, it could have disastrous consequences. An example of this is the case of autonomous vehicles. It may seem tempting to employ concepts such as flocking as a means of coordinating vehicles moving in groups, such as cars or drones. However, if such a system were to become unstable and start to exhibit unintended behaviors, it is easy to imagine that a great risk could be posed to the public. Since it is not possible to guarantee the absence of such behaviors, we will advice that the results of this study not be used for any application where the safety of humans is a concern.

2

Theory

This chapter will cover the theory which is useful for understanding the project as a whole. Theory needed to describe the background of the models will not be covered here but rather in chapter 4.

2.1 Emergence

According to Goldstein, there are a few properties of emergent phenomena that always hold [3]. This is regardless if the emergence is arising in nature or in computer simulations. The following properties identify emergent behavior according to Goldstein:

- **Radical novelty** Perhaps the most recognizable feature of emergent behavior is that they have features that are not observed previously in the system. This means that when the emergence arises it is something new and it makes the system behave in ways it has not previously done.
- **Coherence or correlation** The individual parts that comprise the system tend to appear as an integrated unit after the emergent properties show themselves. This is observable for example in bird flocking, when they all seem to move with a collective mind, as one unit.
- **Global or macro level** This is to describe the fact that the actual emergence occur and is observable on the macro level in a system, whereas it still consists of the components on the micro level in the system. In short, the emergence is observable when looking at the system as a whole.
- **Dynamical** It is not predetermined what behavior will emerge from a system and it change as the system changes over time.
- **Ostensive** That emergence is ostensive implies that it is recognized by showing itself. Due to the complexity of systems in nature, it is often the case that each ostensive occurrence of emergence differs a little from previous ones, even if it is basically the same behavior that emerges.

Attempts have been made by for example O. T. Holland to categorize emergent behavior and to formulate a taxonomy to present a sufficient mathematical theory about agent-based modeling [8]. This is where agent-based modeling is lacking compared to classical system dynamics.

Apart from the mentioned properties about emergent behavior, there is also a classification about different types of emergence by J. Fromm [9].

- **Type 1** Simple/Nominal emergence without top-down feedback.
- **Type 2** Weak emergence including top-down feedback.
- **Type 3** Multiple emergence with many sources of feedback.
- **Type 4** Strong emergence.

Type 1 emergence This is a system which has properties that the individual components within the system would not have. The components are however static in their interactions between each other and the outcome of the system will always be the same. An example of this is a constructed machine, such as a watch, which has the ability to keep time whilst none of the individual components are able to.

Type 2 emergence This exists in a system where feedback is present. This can be either through the agents interacting directly with each other or agents interacting with the environment to affect other agents. Examples of this are the fish model and ant spiral model, which will be introduced later. Systems that are unstable due to positive feedback also fall under this category. Type 2 emergence is the focus of this project.

Type 3 emergence It generally look like type 2, and the concept is somewhat hard to grasp. The difference is that there exists positive feedback that is negated due to negative feedback over time. An example of this is Conway's game of life, where the positive feedback of the cells destroying each other is negated by negative feedback of new cells being born [10]. In the scope of this project it will however be near impossible to distinguish type 3 from stable type 2 emergence. This is due to the fact that we have no means to deduce if one of our systems have short term positive feedback but gets stabilized by negative feedback or if there only exists negative feedback.

Type 4 emergence When the emergent properties of one system will together form a new system where emergence arises. An example of this is how culture arises from humans living together, while the life of the human in itself can be seen as an emergent property of the cells that make up the body. Type 4 emergence is outside the scope of this project.

2.1.1 Limitations of emergence

An issue with emergence is that it has a provisional status [3]. This means that what is called "emergence" might just be phenomena that no current theory can describe in a satisfying way. It could be that science has yet to describe how the properties arise of the agents, and not that it is emergent. This implies that there is no definite way of deriving emergence. Thinking this way presents the possibility that things thought as emergent behavior today will be described in some other way in the future. If there is a way to deduce the outcome of the system from the individual properties of the agents, the result can not be seen as emergence. This leads to the argument that a study of emergent properties would be futile. A reply to this critique is also given in [3] which is that the number of things described as emergent continue to increase even though our collective knowledge is increasing rapidly. This would imply that there will perhaps always be some things described as emergent.

2.2 Agent-Based Simulation

Agent-based simulation is a technique of creating systems modeled as a collection of autonomous decision-making entities called agents. Each agent individually assesses its situation and makes decisions on the basis of a set of rules [2]. The agents could be computer programs, in which case they are called software agents, or they may be people interacting in society as described by W. van der Hoek and M. Wooldridge [11]. Using several agents within the same environment are called multi-agent systems. These systems are used in a great variety of applications, such as air traffic control, process control, manufacturing, electronic commerce, patient monitoring, and games [1]. Agent-based systems open up a new perspective of how to solve problems. Small tweaks to variables for the agents or the environment can affect the result a lot. Therefore, agent-based simulation is a useful tool to examine how systems in the whole depends on its rules.

3

Methods

In order to explore the possibility of creating multi-agent systems based on simple rules, a "quantity-over-quality" approach was chosen: the goal was to initially accumulate a set of models as large as possible, where a subset could later be chosen to be developed further and examined in greater detail. This approach was employed since it was unknown in advance what systems would lend themselves to be modeled in this manner; thus, it was necessary to try many ideas in order to find such systems.

The range of things possible to examine is almost endless; the choice of what to model was based on what the group deemed interesting. However, once a model was chosen, the next step was simplification and stating its set of rules (this process is discussed further in section 3.2), followed by implementing the model in NetLogo. Finally, the focus was shifted to studying the model to see which consequences have arisen from the imposed rules and to analyze them.

3.1 Analyzing

The process of analyzing a model has varied from case to case, mostly depending on the number of variables that are of interest and should be varied throughout the testing. However, the general goal was to look at the behavior of the model and compare it to the system that it attempted to replicate. Looking for similarities as well as differences was important. A guiding principle at this stage was what we refer to as **disciplined exploration**: if the imposed rules did not produce the desired consequences, we did not discard them and try a different set of rules; rather, we embraced the consequences that were produced and considered them a new discovery.

It is important to note that even if certain behaviors appear to always be present in a model, it is not a proof that they will always occur. Instead, these observations should be seen as conjectures, or alternatively, proofs that the behavior *can* occur. Nevertheless, such a result can still be valuable; for instance, if we can find an undesired behavior in a system believed to be safe, then we have shown that the system is in fact **not** safe.

One more thing that has differed between models is whether or not their results could be analyzed visually, i.e. whether it was obvious by looking at the simulation if the desired behavior had been achieved. For some models this was not the case, meaning it was necessary to run the simulation many times and collect the results in order to perform statistical analysis. In these cases, external plugins were employed to quickly perform a large number of simulations.

3.2 Working Top-Down or Bottom-Up

When finding a concept to model, there are two perspectives from which it is possible to examine the concept: *top-down* or *bottom-up*.

In the *top-down* perspective, an existing phenomenon is chosen, the challenge of modeling being to find the rules which enable the system to mimic this phenomenon. If the model displays the desired behavior to some extent, any emergence that is found might be how it works in reality.

In contrast, *bottom-up* is based on the idea that you set up some rules and see how the system behaves. These rules may be altered if they show promise regarding interesting emergent properties. For bottom-up, the challenge is to successfully find emergent behavior and assert which rules lead to which consequences.

3.3 Modeling Software

Many different options are available as tools for modeling agent-based systems, such as Gama, NetLogo, or a general-purpose language such as Java.

Gama is a piece of free software available online. This software comes with an IDE which contains the ability of visualization of the model and the scripting language GAML [12]. Gama seems to have a steep learning curve and was therefore dismissed due to the time frame of the project.

General-purpose languages were also decided against due to the time-scope and relevance to the project. Building the environment in which to create and visualize the models would have had some advantages such as more control over how the program behaves; however, this would require a lot of time to be spent towards a goal that is not really relevant for the purpose of this project.

NetLogo is an open source programming environment for multi-agent systems with a graphical interface to visualize the model easily. It is authored by Uri Wilensky under continuous development at the Northwestern's Center for Connected Learning and Computer-Based Modeling. It is available for free download online [13].

NetLogo was chosen as the software for the project because of its simplicity without compromising the possibilities for intricate multi-agent models. It is easy to learn, which makes it possible to develop models within a reasonable time frame. The program has been around for a long time and has built up a considerable and helpful community. NetLogo also has a plethora of guides and models available in NetLogo's Models Library, which can help when looking for solutions to certain problems.

Another piece of software considered was StarLogoNova [14], a programming environment for multi-agent simulations similar to NetLogo. A difference between the two is that StarLogoNova supports collision detection out of the box, meaning that agents can perform specified actions or behaviors upon colliding with another agent or part of the environment. The reason this software was not chosen in favor of NetLogo was that it is inconvenient to use: rather than entering programs in code, it is necessary to enter commands using a graphical interface, a method of programming which is frustrating and time-consuming. In addition, not having access to the models' source code makes it impossible to use a version control system, which was a compromise we were not willing to make.

4

Model Catalogue

As discussed previously, the core of this project has been to create a large number of models using NetLogo. In total, 25 different models have been produced, with varying degrees of success. In this chapter, we will present a summary of, followed by the premise, results, and some discussion about those models which we found the most interesting, surprising, or otherwise notable. Next, we will briefly mention each of the remaining models. A more general discussion about modeling can be found in chapter 5.

4.1 Ant Spiral

Summary

To replicate an ant spiral, three rules were found. The first two being: one rule for movement and one for following scent trails. The resulting behavior of these two rules was that cycles occasionally formed but often they would quickly dissipate. A third rule was added to make ants prefer trails with a stronger scent. This, in turn, did give rise to ants regularly becoming stuck in circles but the model still shows noticeable discrepancies when compared to the real-life phenomenon.

Premise

In nature, some species of ants are almost blind and have developed a unique way of foraging for food. These ants rely on smell and follow the scent of others to find and bring back food to their colony. Together they exhibit an emergent self-organizing structure that is able to sustain itself very effectively. However, there is an inherent pitfall to this strategy. If a party of foraging ants accidentally stumbles back into their previous path, they tend to get stuck moving in a circular pattern and will not stop until either death of exhaustion or the circle is broken through external action [15]. This phenomenon is usually referred to as an ant mill or an ant spiral. The model explores the possibility of modeling this phenomenon using a top-down

approach. That is the emergent behavior is known, but the rules are not.

Rules and consequences

The model was first constructed with agents that follow two rules. A rule for leaving pheromone trails and one movement rule which governs when and how an agent should start following an encountered pheromone trail. Agents in the model are referred to as ants. The first iteration of rules are then specified as:

- Ants leave a trail of pheromones as they walk through the world.
- Ants wander around aimlessly until they encounter a trail of pheromones upon which they will start to follow the trail.

After testing the model it became clear that these rules were not sufficient to accurately model the phenomenon. The main issue is how an ant should move when presented with multiple possible paths of pheromones which in this version is not handled in any specific way. When an ant is presented with a choice, its decision is based on inherent randomness. In the real world, an ant would probably base its choice of which path to follow on the one that has the strongest scent. Therefore in the model, a rule of path selection has been implemented using a sorting algorithm that ranks paths with stronger scent first.

The rules are now:

- Ants leave a trail of pheromones as they walk through the world.
- Ants wander around aimlessly until they encounter a trail of pheromones upon which they will start to follow the trail.
- When presented with multiple trails, ants will prefer to follow the path where the scent is strongest.

This time the model shows more promise as ants frequently clump together forming cyclic paths as can be seen in 4.1. This behavior is not immediate but what has been observed after multiple runs is the eventual emergence of several circular structures that frequently exchange ants with each other.

Discussion

This model show that it is possible to illustrate the phenomenon using simple rules but it is still limited in several ways. For example, the movement in an actual ant mill is usually more of a spiral whereas the emerging structures in the model are only circular. It then appears to be one or perhaps several other underlying rules governing this behavior that has not been implemented in the model. With more

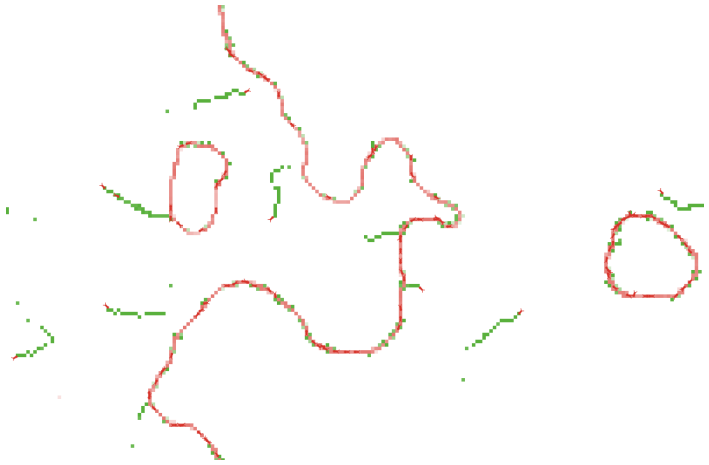


Figure 4.1: Result after letting the ants roam around for some time. Green patches represent the strength of scent trail, red agents are the ants.

intricate knowledge of the phenomenon, it can be worthwhile revisiting.

4.2 Airplane Boarding

Summary

This model attempts to mirror the process of passengers boarding an airplane. The goal is to replicate an experimental study which investigates in what order passengers should board in order to minimize the time it takes for everyone to become seated. The results are largely consistent with what was found in the experimental study[16], which speaks the fidelity of the model. However, certain inconveniences in NetLogo made the model more convoluted than desired.

Premise

When boarding an airplane, it is rarely the case that a passenger can go straight to their seat after entering the aircraft. Instead, they are usually forced to wait for other passengers who are in the way due to various delays. With this in mind, is it possible to adjust the order in which passengers board the plane so that passengers may board as quickly as possible? The goal of this model is to replicate a study by J.H. Steffen and J. Hotchkiss [16], which attempts to answer this question through an experimental test, on actual airplanes, of a few different boarding strategies.

As in [16], this model is set in a plane with 12 rows of seats, each having three seats on either side of the aisle, adding up to 72 seats in total. The world is populated by the same number of passengers, each being assigned to one of the seats. This setup is shown in figure 4.2.

Boarding strategies

The boarding strategies that were examined in the present model are the same as in the experimental study: *Random*, *Back to Front*, *Groups-2-3-1* (called "the Block Method" by Steffen and Hotchkiss), *Wilma*, and *Steffen*. In addition, a few variations of these were also examined: *Wilma Ordered*, *Passenger id*, and *Groups 3-2-1*. Each strategy is described in detail in table 4.1.

Rules and consequences

The following rules are executed in the given numeric order. A snapshot of these rules in action can also be seen in figure 4.3.

1. Walk down the aisle until you reach the row of your assigned seat.

2. When you have reached your row, put your bag in the luggage compartment.

Table 4.1: Description of the boarding strategies examined [16].

Method	**Description**
Random	Passengers board in a randomized order. Note that this does not mean that the seating is random; each passenger still has an assigned seat. Only the order in which they board is random.
Passenger-id	The order in which passengers board is determined by their seat number, i.e. the passenger seated at seat 1 boards first, the passenger seated at seat 2 second, and so on.
Back to Front	The passengers seated in the row furthest back in the plane board first, followed by the row the second furthest back, and so on. Within each row, passengers seated by the window go first, then the middle seats, and finally the aisle seats.
Groups-3-2-1	Passengers are grouped depending on the row of their assigned seat: passengers seated on three adjacent rows form a group. First, the group in the back of the airplane boards; next, the group in the middle; and finally, the group in the front of the plane. Within each group, the order is random.
Groups-2-3-1	Same as Groups-3-2-1, but the middle and back groups are swapped.
Wilma	Passengers are grouped by columns: first, passengers seated by the windows board; next, middle seats; and finally, aisle seats. The order within each column is random.
Wilma Ordered	Same as Wilma, but the order within each column is back to front.
Steffen	A boarding method conceived by the authors of the experimental study, designed to reduce congestion in the aisle. For details, see [16].

3. At this stage, you should be ready to take your seat. However, it may be the case that another passenger seated in the same row as you, but closer to the aisle, has already taken their seat.

   (a) If this is the case, wait for that passenger to get up, so that you can swap places. Next, take your seat.

   (b) If not, take your seat.

4. If you have already taken a seat, but are blocking another passenger from taking your seat as described above, get up and swap places with that passenger, and then repeat 3.

These rules give rise to two sources of congestion in the aisle:

- Passengers who are putting their luggage away (from 2)
- Passengers who are swapping seats (from 3a and 4).

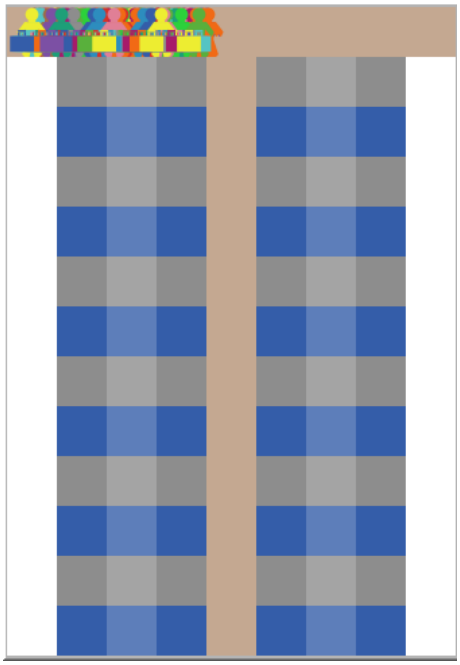


Figure 4.2: The setting of the Airplane Boarding model. Each gray or blue square represents a seat, and the white columns represent the overhead luggage compartments.

In order to evaluate all boarding strategies, an experiment was performed where the model was run 100 times for each of the strategies. At the beginning of each run, a timer is started when the passengers begin the boarding and stopped when every passenger has taken their seat. This way, the average time for each strategy was calculated, as shown in table 4.2.

Table 4.2: Average time to board (in ticks) for the different boarding strategies.

Method	**Time [ticks]**
Random	1049.4
Passenger-id	568.04
Back to Front	1226.73
Groups-3-2-1	1200.48
Groups-2-3-1	1207.24
Wilma	780.36
Wilma Ordered	507.9
Steffen	597.57

Discussion

The results are largely consistent with [16], with a few notable differences: In [16], grouped boarding was found to be the slowest method. In the present model, "Back to Front" boarding is slightly slower than either group boarding method. In addition,

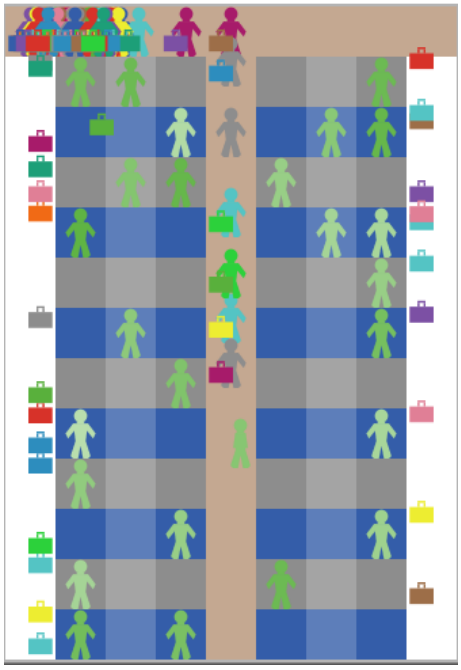


Figure 4.3: The Airplane Boarding model in action.

the fastest boarding method was not the Steffen method, but instead the ordered Wilma method, which was not even considered in [16].

More could be said about these results, but a more detailed discussion of boarding times would be outside the scope of this project. The main conclusion that can be drawn is that the results are quite similar to what was found experimentally, which shows that the rules have the potential of modeling airplane boarding to a meaningful extent.

Although the rules did produce the desired consequences, implementing them was not as simple as hoped because of the limitations in NetLogo. The first complication was implementing the *movement* of the agents: Consider rule 1 "Walk down the aisle until you reach the row of your assigned seat". Stated this way, the rule seems simple; however, implementing it proved more difficult, as NetLogo lacks the concept of "moving towards" a destination. Instead, the passengers have to take a step each frame and then check whether they have reached their row. If so, they move on to rule 2; else, they keep moving. In other words, the agents are essentially implemented as state machines.

Another inconvenience appeared with collision detection. In order to have the passengers stand in line without passing through each other, they must be able to detect whether another passenger is in front of them or not. There is no built-in mechanism to deal with this in NetLogo, so it is necessary for the agents to look at a range of coordinates in front of them in order to determine if the path is clear. If NetLogo had the ability to move the agents forward until they hit an obstacle, as well as a notion of moving towards a destination (as mentioned above), the logic could likely

have been simplified greatly. In its present state, the model comprises over 500 lines of code, which is more than desired for a simple set of rules.

4.3 Children in a Playground

Summary

Agents moving around and colliding creates an environment where it is possible to achieve different consequences by changing what happens upon a collision. Three models based on what children could do on a playground were created to explore this. The movement also showed to be important as the two methods used resulted in very different consequences.

Premise

These models build upon the concept of children running around on a playground and what rules could be given to them to get interesting consequences. Even though the results presented here are somewhat obvious they still classify as emergent since the agents are following rules that are not related to the results that emerges.

Rules and consequences

Model 1 The first model was considered with the rule where if two agents collide, the one with the lower value of a size variable is removed. The rules for the agents, hereafter referred to as children are then:

Movement rule Children walk around randomly.

Main rule If two children bump into each other, the smaller one is removed.

The emergent result of this model is a maximization algorithm. It is visually pleasing albeit inefficient and somewhat predictable. However, the resulting movement of the children is perhaps more ambiguous. Children seem to have a strong tendency to converge towards the middle of the playground. This is because a child has more options to move towards the middle, as can be seen in figure 4.6.

Model 2 Sorting is a common approach to solve problems within computer science. This model is created to see what changes are needed to get a sorting algorithm from the previous maximization model. Firstly the original rule was altered to not remove the smaller child completely, but rather that it should stop playing. Also, two more rules were added; if you are the only child left, go stand in line. If you are not in line and there is no one left playing, go play again.

With these rules a sorting algorithm was achieved. It can be argued that it is possible to build upon rules which have similar properties to the maximization ruleset and create other sorting functions. This can be seen as a selection sort algorithm, where the biggest child is selected each iteration by using the rules described in Model 1. Regarding limitations, because it is based on the model of maximization the same limitations apply here too.

Model 3 In this model, the collision rule of Model 1 was altered. Instead of removing the smaller child, the bigger child gives some size to the smaller. The children's visual size is scaled corresponding to their size variable. Because of that, the result is presented intuitively through visual means, instead of displaying the values of each child. The consequence was also rather unsurprising as it is a mean value algorithm, where all children end up with the same size. This is equal to the sum of all children's sizes divided by the number of them. However, the experience of watching the model was improved with the change.

Model 4 When the previous models were created and the main rule gave rise to emergent properties, it became of interest to take another look at the movement. The altered movement rule of model 1 became as following: if another a child is ahead of you, move towards that child, otherwise continue moving randomly around. What that rule caused was that in some starting positions, the algorithm never finished due to all children running on a line, see figure 4.4. This was unexpected and it was decided to create another model based on this phenomenon; model 4.

In this model, all other rules than the aforementioned movement rules are removed since this was the interesting part. The rules are created with three changeable parameters:

view-distance How far you can see.

field-of-view How wide you can see.

deviation How much you veer when moving randomly.

From this, it was possible to draw some conclusions. All setups we tried will lead to a line of children. However, some parameters will make it impossible to hit the deadlock since the key part of this emergence is that the world is wrapping and the agents always have someone in sight. This means, to hit the deadlock, the variables need to be set such that the agents have the possibility to always have another agent in sight. Therefore, with a low view-distance it is important to have many children to be able to create this line. If the view-distance is altered to be higher, the demand of many children is lowered.

Discussion

The general conclusion of all the models is that it is an inefficient way to solve the mentioned problems using multi-agent systems, because the movement of the entities are random. On the other hand, it is a good way to visualize an algorithm to get a deeper understanding how it works and how the result can be seen. It is also an interesting example of how well-known algorithms can be represented as agent-based simulations. For education, it is a good way to specify the rules which at first glance do not hint what the result will be and which of the rules have the biggest impact. In the sorting example, it might not be obvious that the children will stand in a sorted line when the simulation ends, only by looking at the rules.

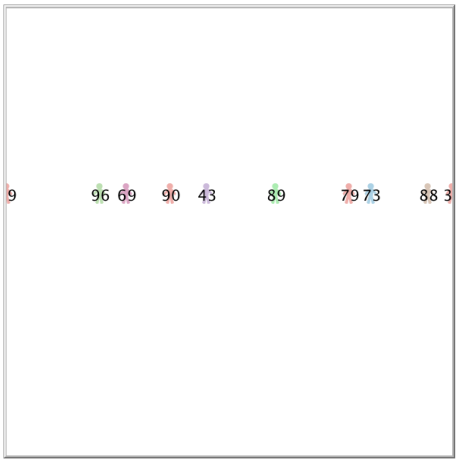


Figure 4.4: Children running in a straight line.

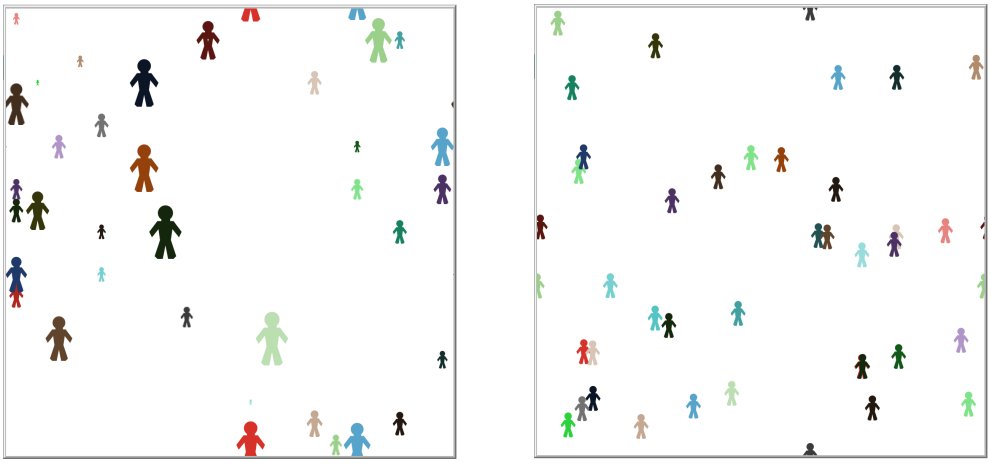


Figure 4.5: Start and end of mean value simulation.

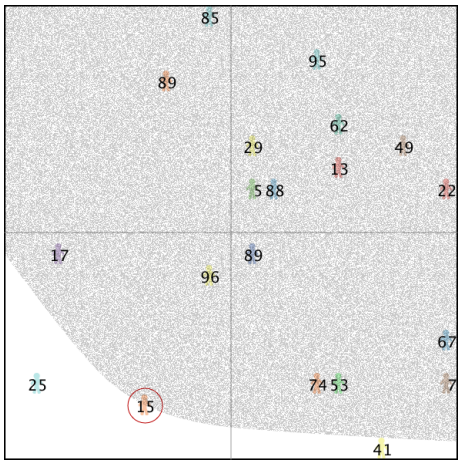


Figure 4.6: The gray area represents all points that child 15 can take a step towards to get closer to the middle.

4.4 Clock - Repressilator

Summary

The rules for this system is that three persons are standing in an equilateral triangle, and each of them tries to hit their respective target with a ball. When hit by a ball, a person may not throw balls for a while. The result of this becomes a clock like behavior. It can also be seen as a simplified version of something called a repressilator.

Premise

The idea of this system is that it consists of three agents existing in an equilateral triangle. The rules for the agents are simple; they throw balls towards one other agent. When hit by a ball, the agent "sits down" meaning that they can not be hit again and won't throw balls for a while. Since the behavior is specified rather than the results, this is constructed with a bottom-up approach. This is a simplified model of something called a repressilator [17] which is a chain of proteins working as a control system and repressing each other. The key word for this model is *stability*, how do we generate a stable repressilating behavior?

Some things are of great importance when creating this model. The first is that the distance between the agents has to be the same, since otherwise the balls will not hit after an equal amount of time which will give an unbalance in the behavior. Another thing that needs to be introduced is some sort of deviation from the straight line when throwing the balls. If this is not the case, every throw would be a hit, which would lead to no interesting behavior at all, since there would never be any change in the system.

Rules and consequences

To implement a system whose behavior is at a reasonable pace, some variables are introduced to the model as seen in 4.3.

Table 4.3: Variables in clock model

Variable	Description
frames-between-balls	How long time the agents wait before throwing a new ball, if they are not hit.
time-down	How long time the agents wait until they throw a new ball after they have been hit.
deviation-throw	The variance, on a standard normal distribution of the throws.
speed-balls	How many patches the balls move for each tick.

Regarding the implication of the variables on the outcome of the model, some actions are of more interest. First of all it should be mentioned that the distance between the agents is 12 patches. *Deviation-throw* has the greatest impact on the result since it influences how often a throw misses or hits. At values larger than 60 degrees, the model completely breaks down since it is no longer guaranteed that each agent will only hit one other agent with its throws. In order to create the repeating behavior of one agent standing at a time, testing shows that values lower than ten degrees seems to be needed. Regarding the other variables, the testing was mainly made with *speed-balls* ≤ 2, since higher values introduced some visual bugs in the model. For these values, it is possible to calculate the values of *frames-between-balls* and *time-down* that will generate a stable behavior. First, we conclude that the time needed for a ball to either hit or miss, is due to the distance between agents

$$\frac{12}{\textit{speed-balls}}$$

We also know how to calculate the probability of a hit, which is the probability that a normal curve with variance *deviation-throw* will be more than 8 away from its mean value (which is when a miss occurs). When *probability-hit* is calculated, the variable *time-down* can be set to get the behavior of one agent standing at a time(referred to as clock-like behavior)

$$\textit{time-down} = 12 * \frac{1}{\textit{probability-hit}}$$

.

And finally, since *frames-between-balls* needs to be somewhat larger than this because an agent might otherwise get hit two times in a row without having time to throw in between. Testing suggests that two extra frames are needed for this to not occur, most likely due to one frame spent on sitting down and one spent on rising. Thus, we have

$$\textit{frames-between-balls} = \textit{time-down} + 2$$

.

Discussion

If these calculations were correct, it seems the system would show a very stable behavior. This is however not the case, the stable clock behavior arises, but also disappears for some time, no matter what values are chosen. In general, it holds that the lower the *deviation-throw* the longer it takes for the stability to show but the longer it holds and vice versa.

The fact that the stability is not as good as expected is because how the collision detection is modeled. Since that is tricky to accomplish in NetLogo, the consequence is that the number 12 used in all the equations above is not always 12, but for the outermost angles that still hit it actually takes 13 frames to know if the ball is a hit or a miss which affects the stability of the system. It is also in its nature not to be completely stable since there is always a probability of two throws in a row missing, which breaks the regular behavior. This can be circumvented by adjusting the slider for *deviation-throw* to 0 when the model is in the stable behavior. Changing the parameters during run-time does not make the model more stable however, it only introduces the possibility to create stable behavior.

Changes to increase stability

The result was not as stable as hoped, therefore some changes to the rules were made. The rule that the agents cannot be hit if they sit down was removed. Together with changes to the parameters, the consequence was a more stable model. It was important to have a higher ratio between time-down and frames-between-balls since it became preferable to hit with many balls each time the agent is active. The variance matters less, because it is important that enough balls hit the target. With very high speeds and balls thrown at a high rate, the model can achieve very precise intervals.

4.5 Crowded Hallway

Summary

When leaving a classroom you might experience some queuing. This model aims to show what happens if all students leave at the same time or independently of each other. Movement and human interactions are a central part of this model and therefore it might be slightly unrealistic. The results indicate that even for a small difference in release time for different classes it is possible to reduce the individuals queuing substantially.

Premise

Evacuation is a field where agent-based modeling is commonly used [18]. Generally speaking, these models are made with tools more complex and specialized than NetLogo, for example by modeling the interactions between people as a sort of magnetic field that controls the flow of people [19]. This model can be used for simulating evacuations, both in emergencies and not. On the other hand, there is critique aimed at modeling crowds as just a grouping of people, when crowd psychology is much more complex [20].

The model contains a school with six classrooms that are to let its pupils out, see figure 4.7. Normally, this is done with a bell, but in order to avoid congestion in the hallway, another option can be explored.

Rules and consequences

The rules for the agents are very simple, they move towards the closest open door. The question is whether or not it is faster to let all the students out of the classrooms at once when the bell rings or when the teacher is finished. We can see the model as a top down view of the school and the rule is to either let all students out at once, or to let them out randomly. The consequences of these models are almost the same except for how much time it takes to get out.

Discussion

The results of interest are the total time until everyone is out and the average time for a student to get out. A benchmark is set up, where the bell is rung so that all students have to evacuate at the same time, see table 4.4.

Table 4.4: Times when all students leave at once.

Total time	124.8
Average time	65.2

To compare other results with this, an approximate time when the bell would have rung is needed. This is set as the time when there is a 50% chance of having left the classroom. This is specified as *Approximate bell time* in 4.5.

The tables 4.5 shows that there are many options when not using the bell that result in better times. For the 10% chance of leaving the classroom, both the total time and the mean time is better. For 3.5% however, the mean time gets 17.7 ticks less while the overall time only increases by three ticks. Since this would mean that all 60 students in the simulation save 17.7 ticks, it seems like a very small price to pay that the total increases by 3. For 2.5% the average is even smaller while the total

Table 4.5: Times when classes are released randomly.

2.5% Chance	
Total	
Approximate bell time	152.1
Average time	175.3
Time diff.	23.2
Single student	
Average time	41.4
Time diff.	-23.9

3.5% Chance	
Total	
Approximate bell time	144.2
Average time	147.3
Time diff.	3.1
Single student	
Average time	47.5
Time diff.	-17.7

10% Chance	
Total	
Approximate bell time	131.3
Average time	130.1
Time diff.	-1.2
Single student	
Average time	61.4
Time diff.	-3.8

is 23.2 ticks longer. This might be slightly too large, depending on the goal of the evacuation.

A conclusion from the results is that a bell might be best if there is an emergency, as it is probably most important to make the total time as small as possible. On the other hand, if it is for a regular break in school it might be more interesting to make the average as small as possible and thus not letting all students out at the same time.

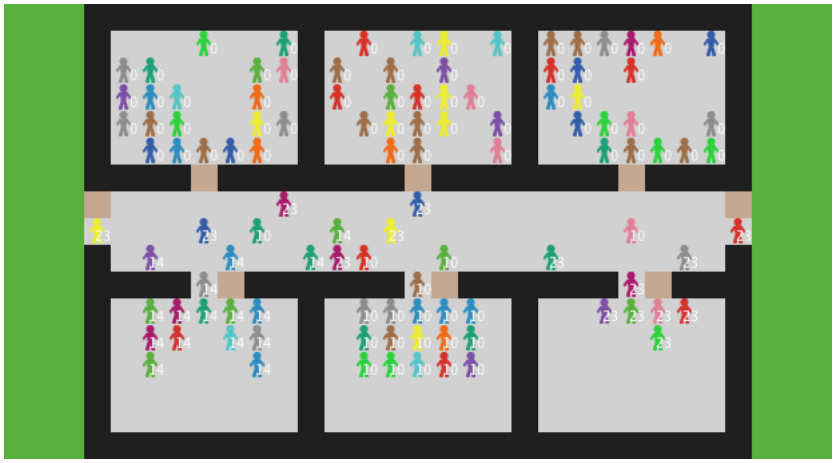


Figure 4.7: Example of an evacuation without the bell.

4.6 Drunkard's Walk

Summary

Here, a random walk model has been created in both one and two dimensions The rules are very similar for both of the models. The rules are that the agents take a step in a random direction for each tick. The result is that the root mean square value (RMS) of the distance for the agents to the origin will be equal to the square root of the ticks.

Premise

The idea of this model is a drunk person, who is to find his way somewhere after heavy drinking. Because of this, his steps are completely random, it is equally likely that he will take a step in any direction. This is an example of the mathematical concept know as **random walks** which have been studied extensively, and is of scientific interest within many fields, for example in biology [21]. It is worth noting that this model is only looking at one very specific random walk but the studies of random walks referred to as Brownian motion [22] is what is of significant scientific interest, whereas this model in itself might not be.

The simplest form of the random walk that will be considered is the "Drunkard's walk in one dimension" also known as "The simple isotropic random walk" [21]. It is to be understood as an agent walking along a line of uniform lattice, for example the x-axis of a standard coordinate system. The agent starts at the point 0 and then move a distance of δ for every step of time τ. This model is simulated with $\delta = 1$ and $\tau = tick$.

Discussion 1D

When producing a graph of the results, we can confirm the root mean square value of the distance from origin of the agents is equal to the square root of the time they have spent walking, as can be seen in figure 4.8.

Rules and consequences 2D

With this model working along one axis, the same idea can be used to extend the model into both two and three dimensions. For the project, some different models were created to show the same idea in two dimensions. As can be seen in picture 4.9, the model colors the patches after how long it has been since they were visited by an agent. The agents themselves now move in two dimensions. This implies that for every step they have an equal probability of going left, right, up, and down.

Until they reach the walls of the room, the same as in one dimension occurs; there will only be agents on every other square after an even/odd amount of steps. This model can be seen with the agents on colored patches and with a border dividing the room in pictures 4.9 and 4.10.

Apart from this version in 2D, there is also one that works without borders, giving the same result as in the one dimensional model; the root mean square value of the distance from origin of the agents is equal to the square root of the time:

$$\sqrt{\frac{\sum \delta^2}{n}} = \sqrt{\tau}$$

where n is the number of agents.

Rules and consequences 3D

Lastly, there is also a model that is created in NetLogo3D. This does however not explore the same random walk in three dimensions but was instead created to visualize what happens in two dimensions in a superior way as in 4.11. The random walk is modelled in the same way as before, but this time it is more visual by stacking agents currently on the same square, giving a height to show the concentration of agents. This model also has the feature that after a chosen amount of steps (300 in the pictures) it starts to visualize the square root of the time value through a post-processing algorithm. This is seen in picture 4.12 with the finished result in picture 4.13. The post-processing algorithm for this is the following:

- Select one of the agents with the largest distance to origin $agent_{max}$.
- Select one of the agents with the smallest distance to origin $agent_{min}$.
- Move $agent_{max}$ one step towards the middle.
- Move $agent_{min}$ one step out from the middle.
- Repeat until every agent has the same distance to the middle.

This generates a circle of agents with the radius $\sqrt{time}$.

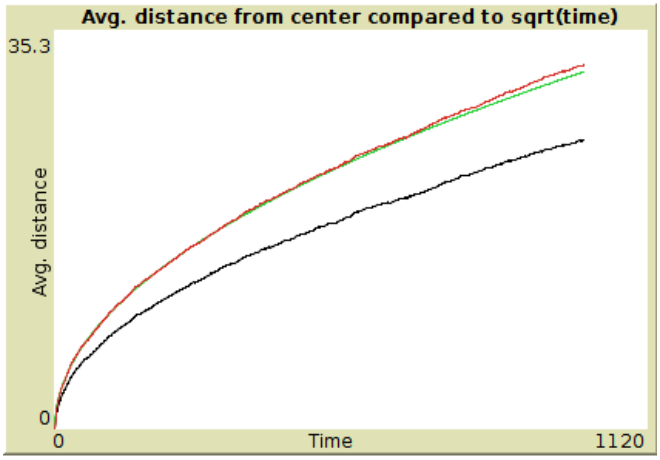


Figure 4.8: Results of drunk walk in 1D. The black line is the average distance to the middle, the green line is the RMS value of the distance and the red is the square root of the time.

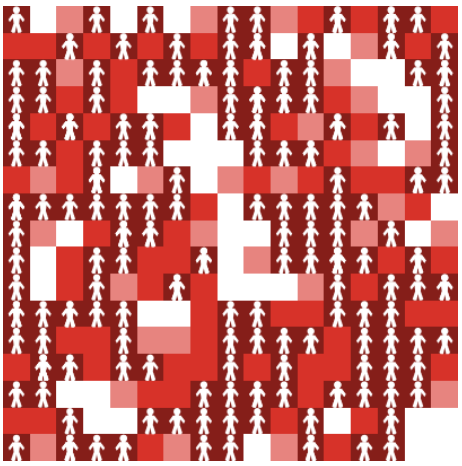


Figure 4.9: Drunk walk in 2D.

Figure 4.10: Same simulation with a border dividing the room.

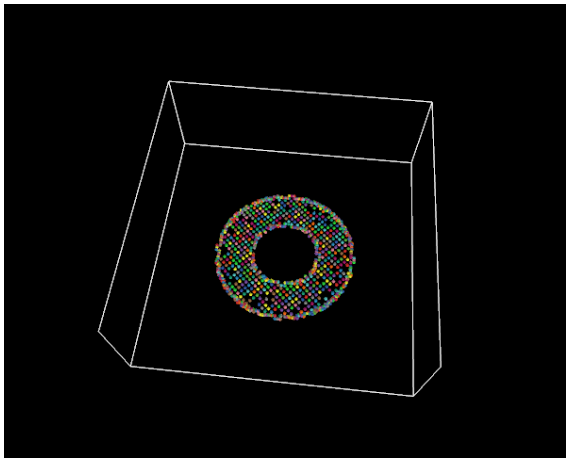


Figure 4.12: Running the algorithm to visualize the square value.

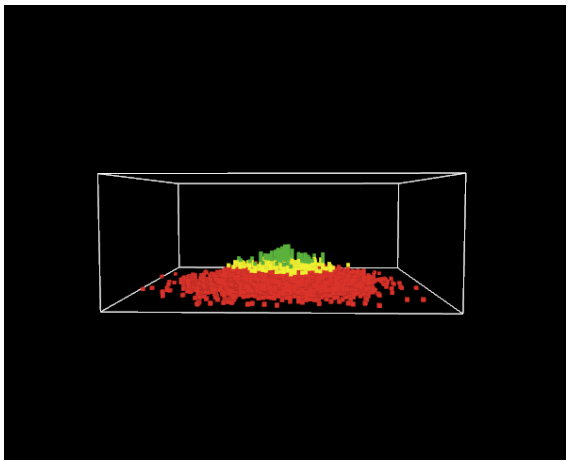


Figure 4.11: Results of drunk walk in 3D.

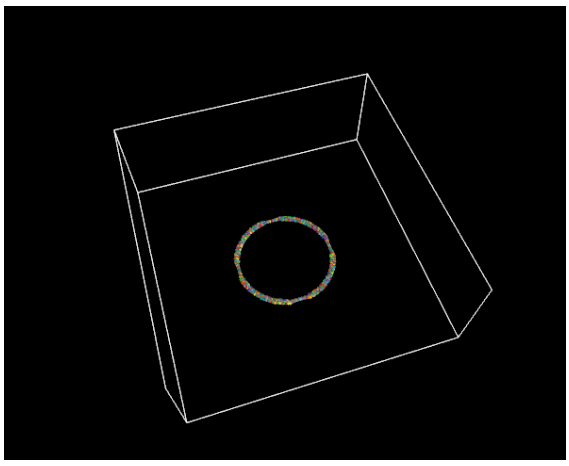


Figure 4.13: A circle of the drunks at distance $\sqrt{time}$.

4.7 Fisher's Principle

Summary

Fisher's principle states that the sex ratio in a population always moves towards 1:1. The rules each agent follows have been derived from the work of W.D Hamilton and simplified in NetLogo to a model where an agent can be either a man or a woman. These agents find mates and produce offspring according to a variable meant to replicate some gene.

Premise

In most sexually reproducing species there is an interesting emergent phenomenon. The populations of both sexes almost always seem to move towards a 1:1 ratio. In

1930, Ronald Fisher proposed that the reason why this happens must be due to natural selection [23]. The theory was then further strengthened by W.D Hamilton who provided a set of six steps by which the agents of the system follows and that causes this emergent behavior to appear [24]. This model will then explore the application of Fisher's principle to a multi-agent system.

Rules and consequences

Hamilton defines the rules of Fisher's principle [24] as follows:

- Suppose male births are less common than female.
- A newborn male then has better mating prospects than a newborn female, and therefore can expect to have more offspring.
- Therefore parents genetically disposed to produce males tend to have more than average numbers of grandchildren born to them.
- Therefore the genes for male-producing tendencies spread, and male births become more common.
- As the 1:1 sex ratio is approached, the advantage associated with producing males dies away.
- The same reasoning holds if females are substituted for males throughout. Therefore 1:1 is the equilibrium ratio.

These rules have then been translated and simplified to the following list:

- Each agent has some probability of giving birth to a child of a certain sex.
- Agents are looking for one partner of the other sex. At each turn, the minority sex will choose one mate if they did not previously have one.
- At each turn a couple has a chance to produce offspring, the sum of the probability of the two individuals will determine the sex of the child.
- The offspring will inherit a gene which in turn determines their probability of giving birth to a certain sex. some random sway is added to account for mutation.
- An agent dies sometime between the age of 70 and 90.

The model is initialized with 1000 agents with about 50/50 males and females. It has two buttons that will add 100 agents of one of the sexes.

The result when running it as initialized is that the sex probability seems to affect the population more than the opposite. It doesn't converge exactly to 1:1, but

instead oscillates at values close to it, as the deviation has some impact. It takes some time before it stabilizes.

Discussion

When adding some agents of one of the sexes, the probability of giving birth to the majority sex is declining. An interesting effect with this model is that when you add a lot of extra people of a certain sex e. g. females, the gene for producing females decreases. An equal amount of the added females won't be able to find a partner, so the population will be almost the same after about 80 ticks (years) as when the button was clicked, but now the sex probability has changed and sometimes causes the whole population to die because of its inability to give birth to females. This property is also described by Hamilton in a similar scenario where he discusses the effects of introduction and subsequent spread of a mutation that inhibits the ability of one of the sexes to be born [24].

4.8 Football

Summary

This model tries to mimic football (soccer). The agents follow two simple rules for movement and kicking the ball. The result resemblances kindergarten football. General thoughts have been mentioned how we think professional football can be modeled. The model is made with a top-down perspective.

Premise

In this project, football has been top-down analyzed since it is well known how the game should look. This model's domain is well known by most people which can help understanding and see the emergent properties arise.

Rules and consequences

The first rules implemented for the agents with the following rules:

- Players move towards the ball.
- If a player is at the ball he kicks it towards the goal.

Given this set of rules, with the following implementation **??**, the model displays some kind of resemblance to kindergarten football. Each team starts at a differen

side of the pitch and the ball in the middle. Simplifications are made to the model such as the entities are reset to their start positions when the ball leaves the pitch or a goal is scored. The simulation has a few parameters to be adjusted, such as variance in speed and strength with a random deviation.

Discussion

As the simulation runs, the result is that all agents follow the ball in a cluster and try to kick the ball towards the opposite team's goal. A result noticed was that the fastest player on the pitch would outrun the other players and score a goal. Therefore, if the speed variation is set to zero, a more equal game would emerge.

The model have shown interesting results so far and it seems possible to extend the stated rules and go from kindergarten football to a professional level. The set of rules that could be required to achieve professional football can only be speculated, but the following rules might be reasonable:

- The player will pass if possible and beneficial
- The players are not allowed to be offside.
- The players will run towards beneficial positions.

However, with these rules, there is a lot of other concepts that need to be introduced such as team-play, offense, and defense. Beneficial passes involves decision making and adaptation to different situations on the pitch. Most of these issues are not considered when starting the implementation, but become apparent as the rules are implemented. This model was discontinued due to these implications and the idea of football being modeled as kindergarten football was deemed a success.

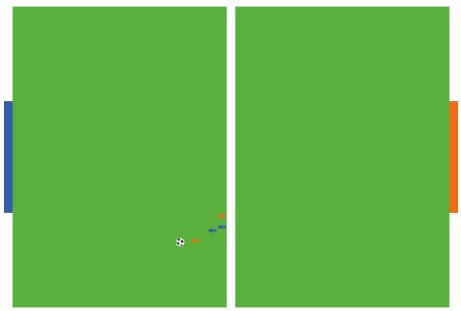


Figure 4.14: Example 1: Game with 2 player in each team.

4.9 K'NEX

Summary

By connecting lines at different angles, you can create shapes which are very unique. Some of them seem to have different properties depending on what angles are chosen, for examples using 60°we get hexagon patterns in 2D and cover the space efficiently in 3D. The modeling and simplifications were easy with the chosen implementation but difficult to analyze; it shows that given a simple ruleset, it might be easy to implement but very hard to draw conclusions.

Premise

There is a toy called K'NEX that simply is rods of plastic which in each end can be placed in cog-like parts that connect them at an angle. Now, imagine a group of children playing with these things using only a given angle, only choosing the direction it is placed in. What can be created? Do we get some special features from the different angles? What are the consequences of choosing one angle instead of another? The model that was created aims to visually show what could be built using these parts.

Rules and consequences

The rules are:

1. Choose an angle between the rods.
2. Place a rod.
3. Pick a random rotation, for 2D only 0 and 180 is available.
4. Connect a new rod to end of the previous rod at the angle in step 1.
5. Repeat step 3 - 4, until finished.

These rules create a continuous line of rods with some different properties for different angles.

Discussion

The model was created to allow for both 2D and 3D structures. To make the models simpler it was decided to let the world wrap which might be the cause of some issues, namely structures not overlapping perfectly. There are a few parameters to alter for

this model, the angle of which the pieces are placed and if it should be in 3D. There is also an option of having different colored pieces and to resize the world for better visualization.

When running this model, it is quite hard to quantify the results in a valuable way since it mostly creates visual feedback and it is a very subjective interpretation of each simulation. There are some parallels to be drawn for some of the angles that are chosen. The 60° in 2D, figure 4.15, results in a hexagons pattern which is very common in nature in for example carbon molecules or beehives. The 90° structure, figure 4.16, just creates squares and is pretty much as expected.

In 3D we get consequences that are harder to define. There are mainly three different angles that create structures that appear to be similar to other structures. The first one is around 20° and looks like a root system, as can be seen in figure 4.17. Secondly, choosing 60°, figure 4.18, it looks more blocky but covers the space much more efficiently. The third degree is 90°, figure 4.19, which is even more blocky and does not cover the area well, it results in higher concentration of pieces in certain areas. To draw conclusions from these results is difficult and other than the visual observations there are not any to speak of.

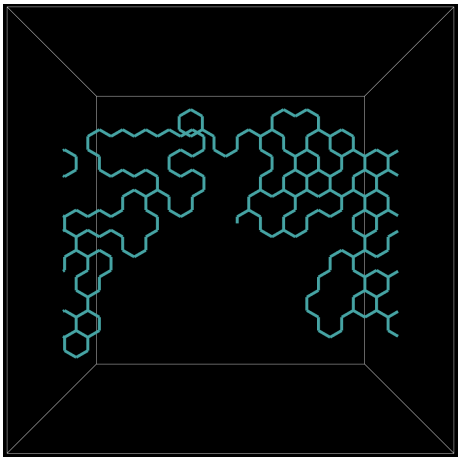


Figure 4.15: 2D structure made from 60° angles.

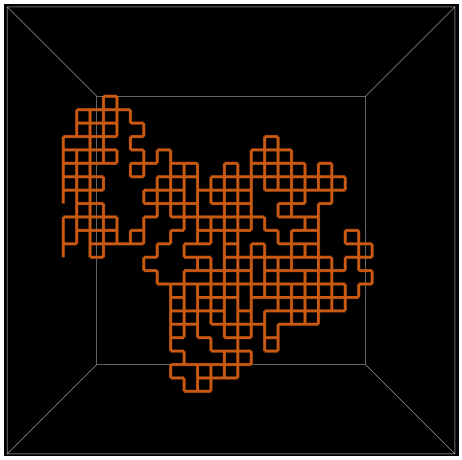


Figure 4.16: 2D structure made from 90° angles.

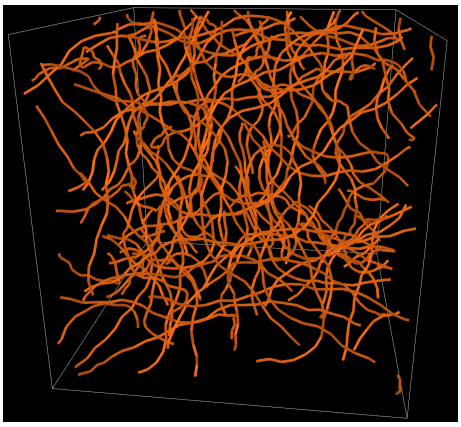


Figure 4.17: 3D structure made from 20° angles.

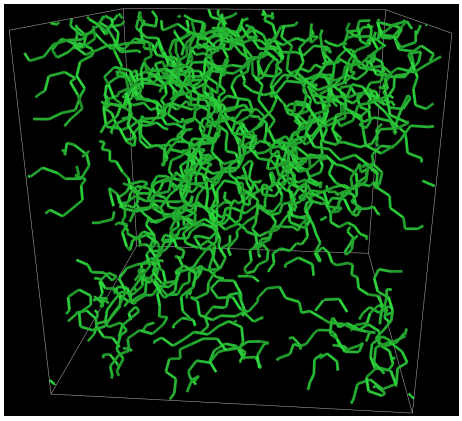


Figure 4.18: 3D structure made from 60° angles.

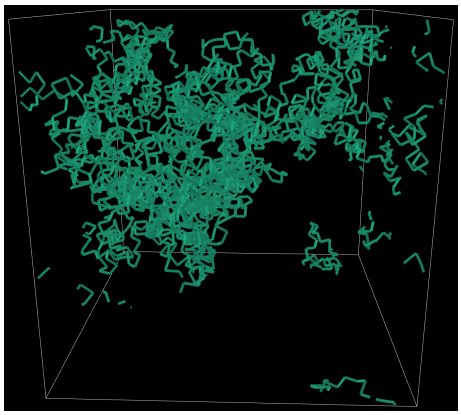


Figure 4.19: 3D structure made from 90° angles.

4.10 Kolam Grammars

Summary

Kolam is an indian art form consisting of a grid and lines drawn around points. There are many ways to resemble Kolam grammars, here we present two; replicating an existing Kolam grammar and generating new ones. The result is that replicating a Kolam pattern works fine if it is possible to explain using simple rules. To generate new Kolam grammars is a bit more difficult, especially to setup rules to make it look neat.

Premise

Kolam Grammars is an art form primarily found in the southern regions of India. It is unique in the way that it heavily relies on repeating patterns which makes many Kolams fractal-like in appearance. Another feature of some Kolams is that they

follow very simple rules that when iterated on create large complex patterns.

Two models have been constructed. One which explores ways of replicating an already existing Kolam, namely, The Anklets of Krishna, see figure 4.23 using a Multi-agent system. The other model focuses on creating Kolam grammars using random movement. The purpose of creating the first model is to be able to compare them and see how they are similar and also how well we could actually reproduce the algorithm explained below with a multi-agent system.

Model 1

Rules and consequences

Marcia Ascher describes the grammars for drawing the Anklets of Krishna with a picture language where a turtle moves across a canvas [25].

Ascher's base rules are:

- F = Move forward
- R1 = Move forward while making a half turn to the right
- R2 = Move forward while making a full loop to the right

Rewriting rules have also been defined by Ascher to create a more complex Kolam. The rewriting rules have been specified so F remains untouched, R1 becomes R1FR2FR1 and R2 becomes R1FR2FR2FR2FR1.

In its simplest form, the rules for drawing does not need to undergo any major changes when making the transition to NetLogo code and can mostly be interpreted directly. However, when implementing the rewriting rule to draw more complex Kolams, some questions arise on how to tackle the actual rule. In this case utilizing multiple agents prove useful in maintaining the simplicity of the rules for each agent. In the model an agent then perform its rules 4 times in descending order to form one part of the Kolam. A picture illustrating how agents move and replicate can be seen in figures 4.20 to 4.21.

The rules are:

- Agent moves forward
- Agent does a half circle, checks condition, spawns a new agent if true
- Agent does a half circle

Discussion

While this does produce the more complicated pattern, several issues have come to light in this iteration. The pattern grows very irregularly due to all agents on a specific level not hatching at the same time. The agents are also currently not aware of what areas of the Kolam have already been drawn, resulting in the growth of agents being exponential and thus very taxing on the CPU. This issue is solved by introducing a patch-specific variable that marks a part of the Kolam as drawn.

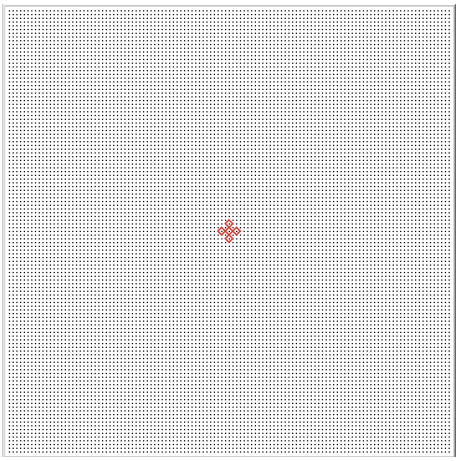


Figure 4.20: Results after clicking on the button step once.The first Agent has finished drawing a Kolam. One new agent has been spawned at each edge for a total of 4 agents

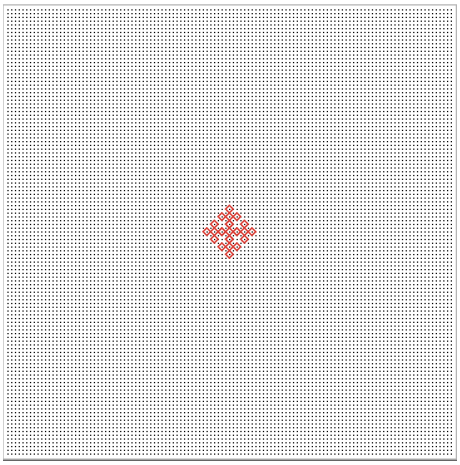


Figure 4.21: Results after clicking on the button step twice.

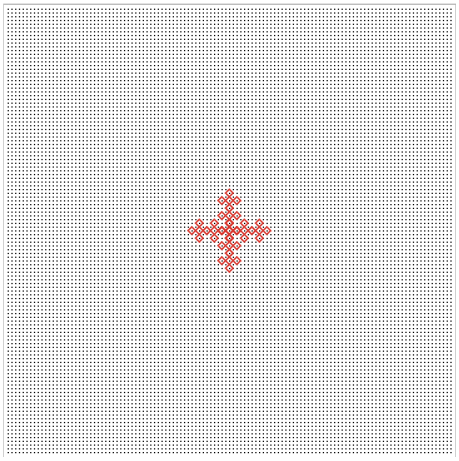


Figure 4.22: Results after clicking step three times. There are now 12 agents in total.

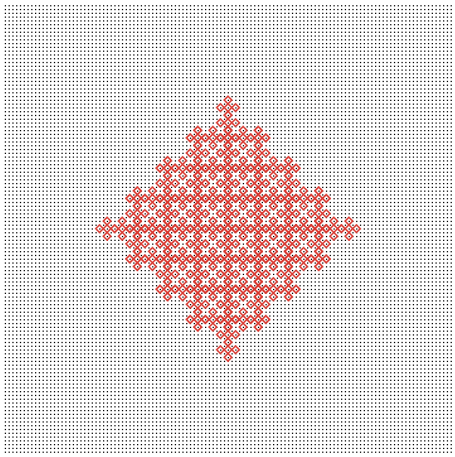


Figure 4.23: The Anklets of Krishna from model 1.

Model 2

Rules and consequences

This model creates random Kolam grammar patterns. The algorithm for the turtle is:

1. Choose a random white patch where you have not been before.
2. Pick a side of the white patch (right or left).
3. Go there and turn half a lap around the patch.
4. If the number of steps are done, finish by going back to the start.
   Otherwise, go back to 1.

How it is actually done in NetLogo is a bit different since it is sometimes hard to follow paths. The model consists of four turtles, two painters and their respective butlers. Two painters are needed to more easily create the reflection and thus symmetry. The butlers are needed because it is an easy way of selecting a patch in close proximity to a certain patch, which here is a white dot. The procedure starts with the first butler selecting a random white patch, distinguished from the one it is currently on, and goes there. Its painter then asks for the patch the butler is on and decides if it should go around it on the left or right side. The painter goes there and turns half a spin around the patch. The other two turtles then do the same pattern, reflected.

Discussion

The modeling was not trivial since it is hard to get turtles to follow paths and make it look nice. A next step would be to evaluate the aesthetics of the patterns in some way, for example be checking if there is more than one crossing close to each other. Unfortunately, the built-in pen-down command is used to draw the lines, which one can not check after drawing. The alternative is then to save all the points and the order they were executed in, which leaves a hard and subjective problem.

An attempt to make a model based on simpler rules were tried. The idea was that a turtle would wander, choosing between going left, forward or right. The problem is that it is hard to keep the rule of not going in the same circle as you have drawn in NetLogo, as it was hard to track or check. An attempt to make this in 3D was made without any success.

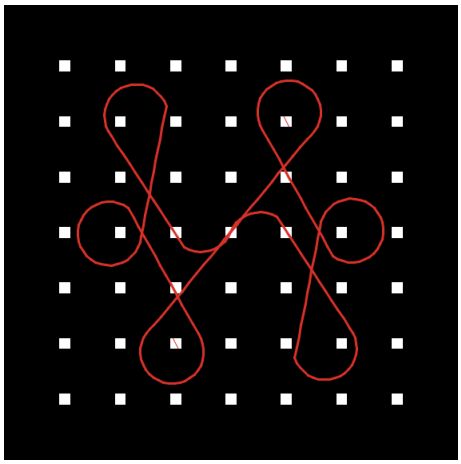


Figure 4.24: Example Kolam from model 2.

4.11 School of Fish

Summary

The Boids model of flocking uses separation, alignment, and cohesion to resemble fish, among other species. This model uses only two of the rules and looks like a school of fish. One can change what happens to the formation if the sight of the fish is limited. A conclusion is that the fish need to see at least two patches far and around 60 degrees of vision to school.

Premise

A school of fish is a group of fish swimming together. The behavior of this and bird ability to move in unison inspired Craig Reynolds to make the Boids model; a computer program meant to simulate coordinated movement in a group of animals [26]. The name boids refers to the term bird-like object.

Rules and consequences

Boids model consists of three rules from Reynolds original computer program and concern other boids in its proximity:

Separation avoid collision.

Alignment steer towards the same direction as the agents in close proximity.

Cohesion steer towards average position.

The boids model is used in several areas, for example stochastic optimization algorithms [27] and movie making to resemble birds [28]. For our model, only separation and alignment are needed. All fish start at random positions. If the fish does not find any fish in its proximity, it keeps swimming straight.

Discussion

When we added sliders for the field of vision of the fish and tested the variables, one can see that in this case, the fish need to see at least two patches far and around 60 degrees of vision to school. Also, for the purpose of this model, cohesion is not implemented since the two first rules give the requested behavior of creating a school of fish, which is sought after when looking at the smallest ruleset.

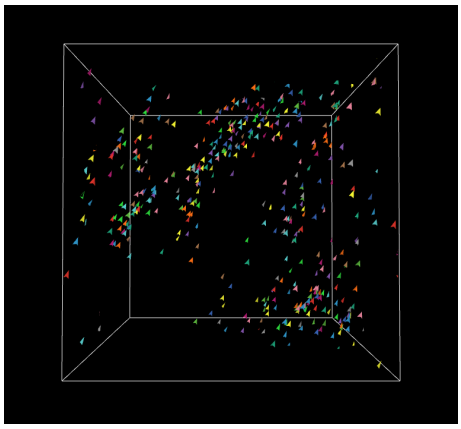


Figure 4.25: A school of fish.

4.12 Tree Growth

Summary

The rule for the tree is: grow in the direction which is most suitable in current state. It needs to keep a balance between nutrition which is found in the ground and sunlight.

Premise

The idea behind this model is to find what rules are needed in order to build an agent-based tree. The agent in this case could be interpreted as a cell or a group of cells in the tree and the model explores what rules for the cells constructs something that looks like a tree.

Trees are thought of as having two basic needs in this model; sunlight and nutrition. This is highly simplified since there are many types of nutrition a tree needs. The growing in this model is random and does not directly regard that trees need to grow in order to spread its seeds, gain stability, and intake carbon dioxide.

Rules and consequences

The model is created with differently colored patches. Every patch in the model has a value dependant on how much nutrition it contains and how much sun it gets. The availability of sunlight is dependent on if there is a yellow patch above and how many solid/semi-solid patches are in the way. Ground is denoted by a gray color and the tree-parts are represented by coloring a patch brown, there is also a green layer for grass. The initial setup of the model is simply a stump.

The rule for each patch in the model is implemented as follows:

- With regular intervals, attempt to grow.
- Determine how profitable it is to grow in each direction.
- If the best direction is worth growing to, grow that way.

Discussion

Every part of the tree decides whether or not it is going to branch. The main feature with the model at this stage is that all parts of the tree are following the same rules, but because of what resources are available to them, the function of each part changes. The model creates branches and roots with different structures due to which resources are available and how it is gathered. However, this might not be the case in reality, but it gives some credibility towards the cells being able to follow the same rules even though the functions are different.

The result of the rules set up is a tree-like structure, see figure 4.26, that is similar to the structure of a Savannah tree, figure 4.27. It is somewhat intuitive that this will be the result if the rules are analyzed deeply. However, the model is indeed interesting to look at due to its similarities with a real plant and that this pattern of growth can be emulated by this simple set of rules.

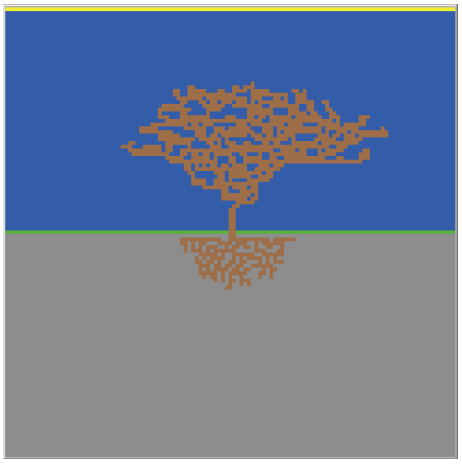


Figure 4.26: Tree made from model.

Figure 4.27: Savannah tree (available under CC0 license).

4.13 Turning Birds

Summary

Birds avoid collision with each other by always turning to the right [29]. This model explores why they turn right and not any other direction. The result

was that birds flying in circles survived most times. The result is not realistic because birds, in reality, have a destination and can not circle forever. The reason is most likely that modeling only that single behavior of a bird is a great simplification of their movement.

Premise

Human inventions sometimes draw inspiration from nature. As drones and pilot-less aircraft become more popular, the question of how to not collide becomes important. According to [29], birds follow two rules: veer right, change altitude. The question then arose, why is it right? Why not left? What happens if you do not turn at all or maybe turn randomly?

Rules and consequences

To look for answers to these questions a model with four different breeds of birds were created and examined. The breeds differed in how they act when they see a bird in front of them. All simulations were run 2000 times with each setting. The model is created to look at what preference works best if it at one time was an equal amount of each bird. For this model, there are then two parameters that matter, how far away a bird is and in what field of view the bird is before veering.

Low angles in the field of view show that it is beneficial to turn randomly instead of turning either left or right. Fly straight is, however, the best option, see table 4.6. This is interesting because this is not what was found in [29] as real birds would not turn in that case.

The result changes when the field of view is larger. The threshold when the birds with a preference for turning got the upper hand was around 90-100 degrees, as can be seen in table 4.7 and 4.8. This seems preferable at first as the agent can veer from more birds, but what actually happens is that the birds hit a deadlock; an endless turning loop which makes sure it does not collide.

total birds	13,88
right preference	2,72
center preference	4,08
left preference	2,99
random preference	4,07

Table 4.6: Average birds remaining with FOV 60.

total birds	15,02
right preference	3,47
center preference	3,88
left preference	3,53
random preference	4,13

Table 4.7: Average birds remaining with FOV 90.

total birds	16,17
right preference	4,36
center preference	3,71
left preference	4,34
random preference	3,76

Table 4.8: Average birds remaining with FOV 100.

Discussion

When looking at this result it is important to remember that these models are highly simplified and a lot of physics and other factors are ignored. For example, it might not be possible for a bird to fly in circles or that a bird will turn even if not on a collision course but rather because it sees another bird.

The overall impression of this was that it seemed unlikely that flying straight was better than veering to avoid collision. It does makes sense that it is better to let the other one turn but nothing in the model really gave that impression from the start. There is also a point to be made about how the birds were flying, as it might not give the correct preconditions for the issue at hand. This was obvious when the birds started to fly into each other when flying side by side, or exposed with a T-crossing and one bird continuously turns toward the bird passing by and thus finally colliding. This proved to be a good example of how the simple rules might not be good enough to model a problem.

4.14 Other Models

The models previously presented can be seen as descriptive for the project as a whole, and the themes they contain regarding agent based modeling is what the conclusions are built upon. The rest of this chapter describes the remaining models, including an explanation for why we chose to exclude them from the main part of the report.

Atom Collision

From the K'NEX model the idea to create carbon molecules arose. It could be interesting to see what kind of molecules could be generated. This was hard to model with our current knowledge about NetLogo. The structures led to a very simple model that created unrealistic molecules and did not seem to produce good enough results to continue.

Bird Chase

This model attempts to replicate the stability presented in the Wolf Sheep Predation model from the NetLogo Models Library [30]. The premise of the model is to have two species of birds – a predator and a prey – which interact together with each other in a 2D environment. Predators would chase prey through a cone of vision and prey would actively try to avoid being spotted. Later revisions also included ways for both species to reproduce either as prey by moving to specific positions or as predator by eating prey.

While the model showed some promise with both populations being dependent on each other what tended to happen was that one population massively out produces the other after a few swings in size. Sometimes the system is stable for some time but then there is always some event which destabilizes it.

We didn't include this model because it did not differ enough from [30]. Most other interesting features were visible in the other models that were chosen. It is also less visually interesting than other models.

Broken Window

The "Broken Window" theory is a theory about how serious crimes can be reduced by stopping minor violations [31]. What could be interesting with the model is to see if it actually works and if possible, to what extent. It turned out to be too

exhausting to model since the guessing of interdependent variables was used in too many places. There was also no intuitive way of representing it visually.

Camouflage Mutation

The idea is to simulate how a predator-prey system could adapt to mutations. It is known that animals use colorful patterns and bright colors to show that they are poisonous. The model then consists of four different agents: predators, poisonous frogs, non-poisonous frogs, and mutated non-poisonous frogs with the same colorful skin as the poisonous ones. This model could turn out to be interesting regarding if the predators adapt to changes in the system.

Collision Detection on a Grid

This model is set on a network of randomly generated paths which intersect each other at multiple points. Agents are traveling along these paths from a start point to an end point and back. The goal was to find rules that decide which way an agent should turn in an intersection, such that there would be no collisions between agents.

One rule that worked fairly well was the *always right* policy: whenever you reach an intersection, turn right if possible, otherwise, continue forward. With this rule in place, there are typically no collisions (except for a special case where paths wrap around the world in a certain way, but this will not be discussed here).

The reason that we did not consider this model to be one of the more interesting ones is the lack of interaction between agents. We did consider coming up with rules which would introduce some sort of communication or feedback between the agents, but these were not explored further due to time constraints.

Game of Life 3D

Game of life is usually in two dimensions, but is possible to make in 3D. This model is basically the same as Life 3D in the models library of NetLogo [32] which made further work on it feel redundant.

Marriage

The idea for this model is to examine how wealth concentrates in families through marriage. This can be linked somewhat to the “stable marriage problem”, which

is a famous computer science problem [33]. The similarity is that this model also aims to make people as satisfied as possible with their marriages. The simple rules are that women prefer as rich men as possible and men prefer women as fertile as possible. The model was however dropped partly because these simple rules were quite difficult to model, and many of the variables had to be chosen arbitrarily. Moreover, the results were hard to visualize in a reasonable way, and we also feared that some might take issue with the ruleset, which could be considered to represent an aged vision of marriage.

Power Creep

This model is based on a phenomenon in the world of online gaming known as "power creep". This implies, for games with different playable characters, that newly released characters get more powerful as time goes. It is then as the term implies, that the general power level of the characters is rising. It is interesting to examine what rules in creating the game give this behavior.

This model was not included, mainly due to two major things. The first is that this phenomenon is really hard to visualize. Many attempts were made at making the model intuitive and pleasing to look at, but this was ultimately unsuccessful. The other thing is that, while the model can still produce results, they are almost impossible to evaluate since most of the variables and functions in the code are somewhat arbitrary. To elaborate, most of the functions are not based on research but rather on what seemed reasonable, since no research was found that they could be based on. This implies that the results are to be seen with some skepticism.

Proxy Voting

Due to low voting attendance it is interesting to investigate the concept of voting by proxy, i.e. allowing voters to yield their vote to someone else. With this idea there are some problems that arise. The model aimed to show how voting results would differ with a proxy vote system versus a regular system. The problem with this model is that it shows some results but not in a very intuitive way. An interesting observation made by manipulating the percentage of people who actually vote, is that proxy voting makes a much larger difference if there are fewer voters. When you have a high attendance – more than 80% – you find that most of the time it would not have altered the results if the non-voters voted or not. This is a bit concerning for democracy and quite interesting but not really within the scope of this project to investigate further.

Social Groups

Is it possible to model the behavior of social groups? This model used a group of people who were each assigned a social score of 0-9. The idea was that the difference in social score of two people talking was the level of not fitting together. A stochastic time variable was also added, since you would probably get tired of talking to someone for a long time even you fit very well together. The time variable turned out to be the killer of the model. What it did was to simply reset all agents at some random time, making it impossible to see any patterns.

Two Bird Species

The idea is two species of birds that interact together with each other. Collision between different species means that both die, collision between two of the same birds gives a chance of hatching a new one.

One model was produced by each group member with slightly varying results. The models were produced very early on in the project and they were used more as learning experience about NetLogo, multi-agent systems and emergence which was also the reasoning behind not including the models as a focus of the report.

Vaccination

Agent-based modeling is often used for modeling diseases, due to the obvious interest in examining diseases and the impact they have on modern society [34]. The idea with this model is to imitate a disease and to incorporate the possibility for vaccination while having the people represented as agents.

The model was dropped partly because it proved difficult to create a realistic model of a disease. However, even with simple rules, some interesting results can be found when looking at a single generation. The real problems occur when looking at people dying and getting born, at which point the model gets out of hand. It was also very arbitrarily modeled regarding the distribution of the vaccine.

War

The premise of this model was that a battle was being fought between two groups: one trying to take over the world, another trying to stop them. The idea was that every time the group trying to take over the world appeared to have been defeated, they always came back. The model was scrapped as the rules ended up being too strict, leaving no room for any emergence or unexpected consequences to appear.

5

Discussion

A lot of models have been simulated throughout the project. It has sometimes been hard to measure successes and failures for each model. Because of the nature of this project it might be hard to determine what to call a result or when a model should be discontinued. Since it is somewhat exploratory, everything is a result in one way or another and thus sifting out the most valuable observations was a challenge. In the following chapter we present our thoughts and experiences of the project. This involves discussing what to model, NetLogo and how to model.

5.1 Models Worth Making

Many of the ideas for the models in the project sprung from when the project group had meetings with our supervisor and discussed what to further work on. Many ideas were presented and discussed that are not a part of this final report; this is our view on why that is the case.

What models that are reasonable to create will often depend on two separate things. The first would be that there are not always obvious ways to create the model with the tools that NetLogo presents. The other reason that ideas were dropped were because the behavior discussed seemed too complex for us to be able to reduce it into a set of simple rules.

This might partially be because of our limited experience of modeling real life behavior. But it might also be that some of these ideas are not suited to be reduced into a simple ruleset. The use of the word "simple" in this case is quite significant since a big part of the goal of the project was defined as just that. There is of course no general conclusion about what would decide whether a model is simple or not, and it is not always the case that simple rules can be described by short sequences of code. If there are no simple rules to be found, the model is not in the scope of this project.

The differences in difficulty of the concepts were not fully realized until a couple of weeks into the project. This had the implications that the ideas for systems to model got much better as time went by. Many of the ideas that were present at the

beginning of the project was later dropped due to the previously mentioned reasons. All in all, the ideas that went into the models got better as the project went on.

5.2 NetLogo

Since we decided to only use NetLogo, it had an impact on the result of the project. Therefore, a discussion of the pros and cons with NetLogo is important. A lot of time has been devoted to NetLogo and therefore, the conclusions can help others who find the software interesting to understand the benefits of using it.

The NetLogo language It is easy to read, write, and understand the code in smaller programs. We think it is appropriate to use as an introduction to multi-agent simulations and agent-based modeling. Especially for models which can be written with few lines. It could be a problem if the domain becomes more complex because the code tends to be unreadable as shown at row seven, in figure **??**. Even if the language has many internal functions with understandable names, using function after function has the tendency to make it hard to understand what happens. NetLogo enables the code to be divided into different files which helps with the readability.

When executing the rules for the agents, NetLogo iterates the defined rules over all agents. This is a potential downside of the language because the iterations are done sequentially which can create biases. For some models this is fine but for others it can affect the outcome of the results. This is not good for models where things must happen in r random order. This could be prevented with concurrent agents which are executed without a predefined order. The iteration often leads to complexity issues, due to iterating over all of the agents at least once, that becomes a problem for some models which leads to the question: how scalable is NetLogo?

Scalability Scalability is a concern when using Netlogo. As mentioned above, the problem mostly comes from the iteration over agents. What we have noticed is that more agents seem to drastically increase the time between each tick. Having agents perform more complex calculations such as sorting seems to be particularly demanding. We do not exactly know how the NetLogo code base has been implemented, but creating models using these methods are restricted to a couple of thousand of agents before computational requirements get too demanding. This could be a problem because the result might not emerge with only few agents nor with many due to how slow the simulation runs. An example of this is our Growth model.

General thoughts Our impression of NetLogo is positive. The software as mentioned above has some problems with scalability of the models and code modularity,

but the language is easy to learn and there is a huge community with tutorials and discussions at your disposal. The software is a great tool for introduction to multi-agent systems and how it can be used to solve or model problems. NetLogo was a good choice of software for this project since the focus has been to look for *simple rules* that produce interesting behavior or mimic a specific phenomena. It was beneficial to choose a software that was easy to get started with, and the concerns about time complexity did not become very important until we started with some more intricate models. All in all, NetLogo is a good tool for multi-agent simulations, especially if experience with similar simulation software is lacking, or many models with wide range of topics are modeled. If one is looking to model a very specific problem and wants it to mimic the reality there is probably another software that is better suited, more specialized, for that exact purpose.

5.3 Visual Representation of Models

Many of the models created that were deemed a failure, were given that label due to problems with visualizing the results of the models. Visualization is of great importance since we are looking for models that are interesting to watch. On the other end of the spectrum we have models that are considered quite successful due to them being fun and interesting to look at. While these problems sometimes have been amplified due to the use of NetLogo, they would not have been avoided entirely with another software that we know of. These problems arise from the fact that models simply work better in a mathematical setting but also because one of the main goals of the project have been to **show** interesting results. This would imply that the search is not only for data and diagrams but also something that visually represents the model. Because of this, models that are interesting due to consequences after many iterations rely on data rather than visual feedback. This is very much the case in the turning birds model as one iteration is not enough to see anything interesting, but requires statistics to show consistency.

5.4 Collision

Many of the models consist of the agents interacting with each other when colliding or within proximity of each other. In these cases it is sometimes required that a collision be registered when two or more agents touch, with all its consequences, in the code. This is not always easy to accomplish and NetLogo did not supply any means of making this easier. This was a problem for example in the clock model, when hits registered at different times due to the visual representation of the model, which in turn leads to less stability in the system as a whole.

Another instance where collision detection was necessary was in the Airplane Boarding model. In order to have the passengers stand in line without passing through

each other, they must be able to detect whether another passenger is in front of them. This is described in more detail in section 4.2.

If a similar project were to be undertaken in the future, we would recommend looking into options for making collision detection easier, as we feel this would greatly simplify some aspects of modeling. The fact that software such as StarLogoNova [14] does provide such functionality is a testament to its importance in modeling multi-agent systems (although we still stand by our decision to use NetLogo instead for the reasons discussed in section 3.3). One option might be to implement an extension [35] to NetLogo which would handle collision in a similar way to StarLogoNova.

5.5 Movement

Several of the models involved movement in their rulesets. Those who did can be placed in two categories, where movement is a central part of the model and where it is not. For those where movement is a central part it was sometimes complicated. Most often this was because of issues with how NetLogo handles a position of an agent in comparison to how the grid-based field of NetLogo is constructed. These issues are however possible to work around but was an issue for some models that caused a disruption in the development. The other case, where the movement is not a central part, seemed to cause different results dependent of the movement. Whether this is an issue or not can be discussed as the different implementations might cause a discrepancy in the results. We can clearly see this in the children models 4.3, where the movement changed and this caused the simulation to sometimes not terminate.

Another thing we noted is that it is sometimes difficult to translate rules about movement into NetLogo code. Rules such as “move towards X” or “keep walking until X, then Y” are easy to express in words or as pseudo code, but become quite convoluted when expressed in NetLogo. In fact, it is often necessary to implement agents as state machines to achieve such rules, as discussed in 4.2. In order to create models using sets of simple rules, we believe it would be beneficial to have a higher level of abstraction in this area, allowing movement rules to be expressed more concisely.

5.6 Similarity in Models

A striking conclusion when working with some of the models is that they are more similar than what one might presume. For example, when working with the Drunkards’ walk and researching about random walks, it seems that many of the other models have this incorporated in them. The K’NEX can be seen as another sort of random walk, and the same can be said for many other movement rules, where it is intended for the agents to move at random in order to not introduce bias.

One thing that makes this interesting is the fact that when having a goal of creating models that are very varied and that are not correlated they still might be that. It could be due to individual programming preferences or the fact that they are all built in the same language but the point still stands that some models are quite similar.

5.7 Complexity of Models

When making models of such various things that have been done during this project, it is quite often that the papers found when searching for references are complex. For example when studying the repressilator model, the paper in which it is introduced requires a lot of knowledge about biology to fully understand. This holds also for the vaccination model where a lot of papers about agent based modeling of diseases require specific knowledge about medicine.

All in all, this has made the research a bit tricky since a lot of time had to be spent on researching domains which are not related to the project in itself, or even computer science at all. There is however not an apparent solution to this; if the goal of the project is to create models of various phenomena then the research will have to be in many varied fields.

6

Conclusions

The purpose of the project was to examine to what extent we could create multi-agent systems with simple rules and still provide interesting consequences. In order to do this, a number of models were created using NetLogo, the process of which lead to some insights and conclusions about multi-agent simulation and the ability to model phenomena using simple rules. This chapter will briefly present what we found out regarding this.

- Creating a multitude of models proved an effective method of conducting a study of emergent behavior in multi-agent systems since it is a very wide concept. By this we found many systems were simple rules provided us with interesting and unexpected consequences.

- Some things are not very well suited to be reduced to simple rules and might altogether be very hard to create with agent-based modeling.

- NetLogo is a good tool to use as an introduction to agent-based modeling. It is also good to use when a project aims to use the same software to model very different phenomena. It is however not perfect and has some issues, mainly with collisions and scalability.

- Multi-agent systems surprises people because our intuition is not well-developed to deal with this.

The general conclusion of this report and our experiences with agent-based modeling is first and foremost that the application field is amazingly broad. With all the different things that this project has described in terms of agent-based modeling there is still a staggering amount of interesting phenomena in the world. Emergence is the term that connects all these things, from life itself to the culture in the society we live in.

6.1 Recommendations for Further Studies

Creating a multitude of models exposed some flaws and inconveniences of the NetLogo language and programming environment. In particular, it was not possible to easily express rules about movement and collision using built-in mechanisms of NetLogo, requiring such cases to be dealt with ad hoc. Since this type of rule appeared often in models, we would recommend those interested in further research in this area to look for programming environments which can express such rules more concisely, since this would allow many models to be created more easily. If no such tool is found, it could perhaps be a worthwhile effort to create one.

Due to the broad spectrum of phenomena that have been examined and modeled in this project, there are many areas where further studies could be done. Many of the models could be further studied in themselves, for example, it is relevant to continue to try and model the spread of various diseases by agent-based modeling as we tried to do. The same goes for many other phenomena where models were created for this project, such as the evacuation of a building. It could also be beneficial to continue working on some of the models in order to produce proofs of the results that we have observed in this project.

Apart from this, when looking at the bigger picture, there are many ways one could further examine agent-based modeling regarding to what extent it is possible to program with it. While it has not been the main focus of the project from our perspective, research about how agent-based modeling can be incorporated in traditional programming is highly relevant.

Lastly, more research can be done about NetLogo in itself and how it could be of use for computer science students. Research about if it could be of use as an introduction to agent-based modeling for students with competence similar to what we had when the project started.

Bibliography

[1] "Multi-agent systems", in *Objective Coordination in Multi-Agent System Engineering: Design and Implementation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 9–32, ISBN: 978-3-540-44933-1. DOI: 10.1007/3-540-44933-7_2.

[2] E. Bonabeau, "Agent-based modeling: methods and techniques for simulating human systems.", *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99 Suppl 3, no. suppl 3, pp. 7280–7, May 2002, ISSN: 0027-8424. DOI: 10.1073/pnas.082080899.

[3] J. Goldstein, "Emergence as a Construct: History and Issues", *Emergence*, vol. 1, no. 1, pp. 49–72, 1999, ISSN: 1521-3250. DOI: 10.1207/s15327000em0101_4. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1207/s15327000em0101_4.

[4] R. Goldstein, "Emergence in Complex Systems", *The SAGE Handbook of Complexity and Management*, vol. 7, no. 10, pp. 65–78, 2011, ISSN: 1530437X. DOI: 10.4135/9781446201084.

[5] F. Engels, *The condition of the working class in England*. 1887, ISBN: 1-4069-2036-3.

[6] R. J. Konopka and S. Benzer, "Clock mutants of drosophila melanogaster", *Proceedings of the National Academy of Sciences*, vol. 68, no. 9, pp. 2112–2116, 1971, ISSN: 0027-8424. DOI: 10.1073/pnas.68.9.2112. eprint: http://www.pnas.org/content/68/9/2112.full.pdf. [Online]. Available: http://www.pnas.org/content/68/9/2112.

[7] J. L. Dessalles, J. P. Müller, and D. Phan, "Emergence in multi-agent systems: conceptual and methodological issues", 2007. [Online]. Available: http://www.gemass.org/dphan/papers/dessallesMullerPhan2007.pdf.

[8] O. T. Holland, "Taxonomy for the modeling and simulation of emergent behavior systems", in *Proceedings of the 2007 Spring Simulation Multiconference - Volume 2*, ser. SpringSim '07, Norfolk, Virginia: Society for Computer Simulation International, 2007, pp. 28–35, ISBN: 1-56555-313-6. [Online]. Available: http://dl.acm.org/citation.cfm?id=1404680.1404684.

[9] J. Fromm, "Types and Forms of Emergence", 2005. arXiv: 0506028 [nlin]. [Online]. Available: http://arxiv.org/abs/nlin/0506028.

[10] M. Gardner, "MATHEMATICAL GAMES by", *Scientific American*, vol. 223, no. 4, pp. 120–123, Aug. 1965, ISSN: 0036-8733. DOI: 10.1038/scientificamerican1070-120.

[11] W. van der Hoek and M. Wooldridge, "Chapter 24 multi-agent systems", in *Handbook of Knowledge Representation*, ser. Foundations of Artificial Intelligence, F. van Harmelen, V. Lifschitz, and B. Porter, Eds., vol. 3, Elsevier, 2008, pp. 887–928. DOI: https://doi.org/10.1016/S1574-6526(07)03024-6. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574652607030246.

[12] A. Grignard, P. Taillandier, B. Gaudou, D. A. Vo, N. Q. Huynh, and A. Drogoul, "Gama 1.6: Advancing the art of complex agent-based modeling and simulation", in *PRIMA 2013: Principles and Practice of Multi-Agent Systems*, G. Boella, E. Elkind, B. T. R. Savarimuthu, F. Dignum, and M. K. Purvis, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 117–131, ISBN: 978-3-642-44927-7.

[13] U. Wilensky, *NetLogo*, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL., 1999. [Online]. Available: http://ccl.northwestern.edu/netlogo/.

[14] MIT Scheller Teacher Education Program, *StarLogo Nova*. [Online]. Available: http://www.slnova.org/.

[15] F. Delsuc, "Army ants trapped by their evolutionary history", 2003. DOI: 10.1371/journal.pbio.0000037. [Online]. Available: http://journals.plos.org/plosbiology/article/file?id=10.1371/journal.pbio.0000037&type=printable.

[16] J. H. Steffen and J. Hotchkiss, "Experimental test of airplane boarding methods", *Journal of Air Transport Management*, vol. 18, no. 1, pp. 64–67, 2012, ISSN: 0969-6997. DOI: 10.1016/j.jairtraman.2011.10.003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0969699711000986.

[17] S. M. D. Oliveira, J. G. Chandraseelan, A. Häkkinen, N. S. M. Goncalves, O. Yli-Harja, S. Startceva, and A. S. Ribeiro, "Single-cell kinetics of a repressilator when implemented in a single-copy plasmid", *Mol. BioSyst.*, vol. 11, no. 7, pp. 1939–1945, 2015, ISSN: 1742-206X. DOI: 10.1039/C5MB00012B. [Online]. Available: http://xlink.rsc.org/?DOI=C5MB00012B.

[18] X. Zheng, T. Zhong, and M. Liu, "Modeling crowd evacuation of a building based on seven methodological approaches", *Building and Environment*, vol. 44, no. 3, pp. 437–445, Mar. 2009, ISSN: 0360-1323. DOI: 10.1016/J.BUILDENV.2008.04.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360132308000577.

[19] S. Okazaki and S. Matsushita, "A study of simulation model for pedestrian movement with evacuation and queuing", *Engineering for Crowd Safety*, no. 1, pp. 271–280, 1993. [Online]. Available: http://www.mukogawa-u.ac.jp/~okazaki/OK/PMOVE/paper1/London.pdf.

[20] I. von Sivers, A. Templeton, F. Künzner, G. Köster, J. Drury, A. Philippides, T. Neckel, and H. J. Bungartz, "Modelling social identification and helping in evacuation simulation", *Safety Science*, vol. 89, pp. 288–300, 2016, ISSN: 18791042. DOI: 10.1016/j.ssci.2016.07.001. arXiv: 1602.00805.

[21] E. a. Codling, M. J. Plank, and S. Benhamou, "Random walk models in biology.", *Journal of the Royal Society, Interface / the Royal Society*, vol. 5, no. 25, pp. 813–34, 2008, ISSN: 1742-5689. DOI: 10.1098/rsif.2008.0014. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2504494&tool=pmcentrez&rendertype=abstract.

[22] M. Kac, "Random Walk and the Theory of Brownian Motion", *The American Mathematical Monthly*, vol. 54, no. 7, p. 369, Aug. 1947, ISSN: 00029890. DOI: 10.2307/2304386. [Online]. Available: http://www.jstor.org/stable/2304386?origin=crossref.

[23] R. A. Fisher, *The Genetical Theory of Natural Selection*. Oxford University Press, 1930, p. 318, ISBN: 0016-6731. DOI: 10.1038/158453a0. arXiv: t8jd4qr3m [13960]. [Online]. Available: https://books.google.se/books?id=sT4lIDk5no4C&printsec=frontcover#v=onepage&q&f=false.

[24] W. D. Hamilton, "Extraordinary sex ratios", 1967. DOI: 10.1126/science.156.3774.477. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.464.5709&rep=rep1&type=pdf.

[25] M. Ascher, "The Kolam Tradition A tradition of figure-drawing in southern India expresses mathematical ideas and has attracted the attention of computer science", *American Scientist*, vol. 90, 2002. [Online]. Available: http://www.jstor.org/stable/pdf/27857597.pdf?refreqid=excelsior%7B%5C%%7D73.

[26] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model", *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 25–34, 1987, ISSN: 00978930. DOI: 10.1145/37402.37406. arXiv: 0208573 [cond-mat]. [Online]. Available: http://portal.acm.org/citation.cfm?doid=37402.37406.

[27] Z. Cui and Z. Shi, "Boid particle swarm optimisation", *International Journal of Innovative Computing and Applications*, vol. 2, no. 2, p. 77, 2009, ISSN: 1751-648X. DOI: 10.1504/IJICA.2009.031778. [Online]. Available: http://www.inderscience.com/link.php?id=31778.

[28] I. L. Bajec and F. H. Heppner, "Organized flight in birds", *Animal Behaviour*, vol. 78, no. 4, pp. 777–789, Oct. 2009, ISSN: 0003-3472. DOI: 10.1016/J.ANBEHAV.2009.07.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0003347209002966?via%7B%5C%%7D3Dihub.

[29] I. Schiffner, T. Perez, and M. V. Srinivasan, "Strategies for Pre-Emptive Mid-Air Collision Avoidance in Budgerigars", *PLOS ONE*, vol. 11, no. 9, D. Osorio, Ed., e0162435, Sep. 2016, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0162435. [Online]. Available: http://dx.plos.org/10.1371/journal.pone.0162435.

[30] U. Wilensky, *NetLogo Wolf Sheep Predation model*, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL., 1997. [Online]. Available: http://ccl.northwestern.edu/netlogo/models/WolfSheepPredation.

[31] B. E. Harcourt and J. Ludwig, "Broken Windows : New Evidence from New York City and a Five-City Social Experiment Broken Windows : New Evidence from New York City", vol. 271, no. 1, pp. 271–320, 2006. [Online]. Available: http://home.uchicago.edu/ludwigj/papers/Broken_windows_2006.pdf.

[32] U. Wilensky, *Netlogo life 3d model*, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1998. [Online]. Available: http://ccl.northwestern.edu/netlogo/models/Life3D.

[33] D. E. Knuth, *Stable marriage and its relation to other combinatorial problems : an introduction to the mathematical analysis of algorithms*, English. Providence, R.I. : American Mathematical Society, 1997, Includes bibliographical references (p. 67-68) and index, ISBN: 0821806033 (alk. paper). [Online]. Available: https://trove.nla.gov.au/work/15032597.

[34] L. Perez and S. Dragicevic, "An agent-based approach for modeling dynamics of contagious disease spread", *International Journal of Health Geographics*, vol. 8, no. 1, pp. 1–17, 2009, ISSN: 1476072X. DOI: 10.1186/1476-072X-8-50.

[35] U. Wilensky, *NetLogo Extensions*, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL., 1999. [Online]. Available: https://ccl.northwestern.edu/netlogo/docs/extensions.html.

[36] F. Marini and B. Walczak, "Particle swarm optimization (PSO). A tutorial", *Chemometrics and Intelligent Laboratory Systems*, vol. 149, pp. 153–165, Dec. 2015, ISSN: 0169-7439. DOI: 10.1016/J.CHEMOLAB.2015.08.020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169743915002117.

[37] Rennard Jean-Philippe. and IGI Global., *Handbook of research on nature-inspired computing for economics and management*. Idea Group Reference, 2007, pp. 28–74, ISBN: 1591409853.

A

Bibliographic notes

Software

U. Wilensky, NetLogo [13] The software that was used to create all the models. Further reading is available on the website or on the GitHub page for NetLogo.

A. Grignard, P. Taillandier, B. Gaudou, D. A. Vo, N. Q. Huynh and A. Drougoul; Gama 1.6: Advancing the art of complex agent-based modeling and simulation [12] The paper describes an alternative programming language to NetLogo, with more features but a steeper learning curve.

U. Wilensky, NetLogo wolf sheep predation model [30] On of the models that comes with the free download of NetLogo. This is of a system with wolf, sheep and grass and how they eat and reproduce.

U. Wilensky, NetLogo life 3d model [32] On of the models that comes with the free download of NetLogo. This model is of Conways' game of life, but expanded to 3D.

U. Wilensky, NetLogo Extensions [35] NetLogo supports external extensions that provide additional functionality to the software.

MIT Scheller Teacher Education Program, StarLogo nova An alternative software for multi-agent systems which enables program to be written with a graphical interface.

Background/Theory

Multi-agent systems, in *Objective Coordination in Multi-Agent System Engineering: Design and Implementation* Springer, Berlin, Heidelberg [1] A chapter from a book which aims to give an introduction to multi-agent systems and autonomous agents. It also tries to make the terms easier to use, since many people

E. Bonabeau, Agent-based modeling: methods and techniques for simulating human systems [2] A paper that discusses four areas of application of agent-based simulation: flow simulation, organizational simulation, market simulation, and diffusion simulation. It also discusses business applications for each one.

J. Goldstein, Emergence as a Construct: History and Issues [3] Discusses the construct of emergence in a general philosophic way, it also places emergence in a historical context and presents some issues with the concept.

J. Konopka and S. Benzer, Clock mutants of drosophila melanogaster [6] A study about changing the living environment for different mutation of a fly to see how they evolve and adept to the changes.

L. Dessalles, J. P. Müller, and D. Phan, mergence in multi-agent systems: conceptual and methodological issues [7] Explains the different meanings of emergence such within philosophy, complex adaptive systems literature and computer science.

R. Goldstein, Emergence in complex systems [4] Discusses emergence as a philosophical concept. Was used to help describing emergence in understandable terms, while the paper as it is might not be very related to this project.

F. Engels, The condition of the working class in England [5] A book by Friedrich Engels regarding the working class in England. Used because what he saw and described draws some parallels to emergence in society regarding socioeconomic factors. The term emergence was however not coined by the time of writing.

O. T. Holland, Taxonomy for the modeling and simulation of emergent behavior systems [8] A paper that tries to establish a consistent lexicon of

emergent behavior in order to better understand emergent phenomena. It presents a taxonomy based on five levels of emergence.

J. Fromm, Types and forms of emergence [9] This paper describes different types of emergence and what properties they have. The proposed types are somewhat different to those in [8] but many things are similar.

W. van der Hoek and M. Woolridge, Chapter 24 multi-agent systems [11] This chapter discusses representation formalisms for multi-agent systems. It also describes four logical frameworks for representing the cognitive state of rational agents and how these framework can help with engineering.

F. Marini and B. Walczak, Particle swarm optimization (PSO) [36] A paper that presents the potential for particle swarm optimization for solving various optimization problems. We used it to present some things about stochastic optimization.

M. Gardner, MATHEMATICAL GAMES [10] This paper discusses the cellular automaton simulation "Conway's Game of Life". In particular, it explores the behaviors of various starting states over time.

Rennard Jean-Philippe and IGI Global, Handbook of research on nature-inspired computing for economics and management [37] This book discusses applications of so-called "nature-inspired computing" in areas such as multi-agent systems among others.

Models

F, Delsuc, Army ants trapped by their evolutionary history [15] A journal article detailing the phenomenon of ant mills within certain species of army ants.

J. H. Steffen and J. Hotchkiss, Experimental test of airplane boarding methods [16] This is a study of different ways to board airplanes and how fast they get all passengers seated.

S. M. D. Oliveira et. al., Single-cell kinetics of a repressilator when implemented in a single-copy plasmid [17] This paper describes one type of

repressilator and how it can work in real life. While our model was not built to be like this, we used it to present the idea.

R. A Fisher, The genetical theory of natural evolution [23] A paper where Ronald Fisher first describes the phenomenon on the sex ratio in reproducing species. W.D Hamilton later builds on these theories in [24]

W. D. Hamilton, Extraordinary sex ratios [24] A paper which explains Fishers' principle in greater detail and situations where it does not hold. It produces a set of rules that was useful for us when modeling Fishers' principle.

C. W. Reynolds, Flocks, herds and schools: A distributed behavioral model [26] In this paper Craig Reynolds explores an approach based on simulating animal movement as an alternative to scripting their paths individually by describing what would later become the boids model.

Z. Cui and Z. Shi, Boid particle swarm optimisation [27] Explains how the inner workings of particle swarm optimisation (PSO) works and that it is a population-based stochastic optimisation algorithm inspired by the Reynolds' boid model.

I. Schiffner, T. Perez and M. V. Srinivasan; Strategies for Pre-Emptive Mid Air Collision Avoidance in Budgerigars [29] A paper on how birds of a special species make decisions when flying in order to avoid a head-on collision.

M. Ascher, The Kolam Tradition tradition of figure-drawing in southern India express mathematical ideas and has attracted the attention of computer science [25] An article that describes the kolam drawing tradition in greater detail.

E. a. Codling, M. J. Plank and S. Benhamou; Random walk models in biology [21] A paper about some phenomena in biology that can be approximated and modeled as a random walk.

M. Kac Random walk and the theory of Brownian motion [22] An older paper on the theory of Brownian motion, which is a mathematical approach to the random walk.

X. Zheng, T. Zhong and M. Liu; Modeling crowd evacuation of a building based on seven methodological approaches [18] This paper tries to model evacuation by using different approaches to try to produce the best result.

S. Okazaki and S. Matsushita, A study of simulation model for pedestrian movement with evacuation and queueing [19] This papers presents a model for evacuation where the agents are modelled with something similar to a magnetic field to be as realistic as possible.

I. von Sivers et. al., Modeling social identification and helping in evacuation simulation [20] A paper that presents critique towards modeling agent based evacuation without taking group psychology into consideration.

L. Perez and S. Dragicevic, An agent.based approach for modeling dynamics of contagious disease spread [34] One of many studies that simulates the spread of a disease through multi-agent simulations. Was used when constructing the vaccination model.

D. E. Knuth, Stable marriage and its relation to other combinatorial problems: and introduction to the mathematical analysis of algorithms [33] Older book that was one of the first to discuss the famous stable marriage problem, among many other things.

B. E. Harcourt et. al., Broken windows: New evidence form New York City and a Five-City social experiment [31] An article about the broken window principle and how it can be seen in different cities in the United States.

I.L Bajec and F. H. Heppner, "Organized flight in birds", *Animal Behaviour* vol. 78, no.4, pp. 777-789 [28] This article discusses various types of organized flight in birds, and the various scientific questions associated with this topic. It also discusses the contributions of computer science in this area.

B

Contribution report

B.1 Daniel Heurlin

In this project, I was responsible for producing some of the models and writing part of the planning report and final report. In addition, I handled certain administrative tasks such as booking rooms, attending fackspråk-sessions and handling some of the communication with the supervisor and course administrators.

The models for which I was the main contributor were:

- Airplane Boarding
- Collision Avoidance on a Grid
- War

Although I have made the fewest models of all members, it should be noted that all of my models are relatively large in size, each being aroud 500 sloc. Whether this is a good or a bad thing can be discussed, but it does explain why each model took more time to make than usual, hence why I was not able to make more.

In the planning report, I was the main author of the introduction, as well as the part of "Social and Ethical Aspects" about the unpredictable nature of emergence. Since these were partly transferred into the final report, I am also the main authors of their equivalents in the final report. I also wrote the foundations for the introductory texts to the "Methods" and "Model Catalogue" chapters in the final report, as well as the sections on "Airplane Boarding" and 'Collision Avoidance on a Grid" in the latter. In addition, I wrote parts of the discussion on collision and movement in the "Discussion" chapter, and contributed to the "Conclusions" chapter. Apart from these things, I also made innumerable smaller contributions all over the final report. Finally, I have helped iterate over and finalize parts of the report that were mainly written by other group members.

During meetings with our supervisor, as well as during fackspråk-sessions, I often acted as a de facto secretary / note-taker. I also handled some of the communication

with the course administrators and our supervisor by e-mail, although the latter was mainly handled by Samuel.

B.2 Felix Jansson

During the project have I either alone or together with another member of the group been responsible for producing and writing about the following models:

- Children in a playground
- Football
- Tree growth
- Camouflage mutation

Throughout the project have I also discussed and analyzed the other group members models to find emergent behaviors. In addition to the models have I written different parts of the planning report and took responsibility with Simon to created the time plan. I was one of the presenter during the half time presentation where I took the role to write what to be presented and to create a digital presentation.

In the final report have I actively worked with literature studies, research about our models and the creation of text. More in detail am I the co-author of the "abstract" as well for the parts about "emergence" and "delimitations" in the *Introduction*. In the *Methods* have I written about "top-down and bottom-up", and contributed to "modeling software" by researching about other software. In the *Models Catalogue* have I written about the models mentioned above as well helped to produce text for other models which its creator later fill in with the detailed information, such as turning birds. In the *Discussion* have I contributed to "successes and failures" and written the text about "Netlogo". I have also written parts of the "bibliographic notes". Beside the text I have produced have I iterated over existing work several times to find grammatical errors, reduce wordiness, make sure each part have correct tense and structure of the report.

During the whole project have I handled administrative tasks such as booking rooms each week. I have attended compulsory sessions as well as "fackspråk-sessions" which not is compulsory. Throughout the project have I always been active in discussions and been trying to move the project forward.

B.3 Karl Strigén

Regarding the model catalogue I have had the primary responsibility for the following models:

- Clock- Repressilator
- DrunkWalk
- Vaccination
- Marriage
- Power Creep

Apart from this I have also been contributing in discussions with the other group members about the modeling in many other cases, mainly for the Crowded Hallway and Growth model.

In the report I have focused a lot on the introduction and conclusion, while also writing about emergence in the theory chapter. Naturally I also provided most of the text about the models for which I had the responsibility as well as a lot of text about the children algorithms.

During the course of the project I have been to almost all of the meetings we have had as a group. I have also attended some lectures and been responsible for some of the administration with bookings.

B.4 Samuel Håkansson

I have created the following models:

- School of Fish
- Kolam (with William)
- Fisher's principle (with William)
- Game of Life 3D
- Broken Window
- Social

I have been responsible for mailing with our supervisor and making the poster together with Simon. In the report I wrote about my models and mainly about

Stochastic Optimization Methods as well as small stuff here and there. I have also focused on fixing bad sentences, deleting unnecessary wording and other smaller errors in the report. I have, like all of us, contributed to discussions about the creation of, and issues with modeling.

B.5 Simon Sundström

In the project I feel that I have been a part of every major section. In an administrative sense I have been responsible for our meetings. I have also been booking some of the rooms we've been working in, we split this over almost everyone in the group.

For content creation I have been responsible, either alone or together with someone, for the following models:

- Children
- Football
- Tree growth
- Turning Bird
- Crowded Hallway
- K'NEX
- Atom collision
- Proxy voting

There are also models that I have contributed to, either by helping out with debugging or simply developing the model. Those models are the clock, power creep, school of fish and camouflage mutation.

We spent some time reading papers and then took a session where we presented what we've learned and how we thought it might relate to our project. I also went to 2 of the voluntary lecturers and relayed information back to the group.

For the report I have produced some main content mostly in the model sections of the models I was responsible for. Other than that I have worked with the report, iterating over the text.

I have, together with Samuel, produced the poster.

X

B.6 William Hjelm

I have been personally responsible for keeping track of what all members have been doing by means of keeping a group diary. In this diary I have summarized what each member has been doing, issues we have faced and how we eventually solved them.

Together with the rest of the group I have been developing and discussing models. Later on in the project I have also been responsible for booking some of the rooms.

I have been involved in developing or improving the following models

- Ant Spiral
- Fisher's principle (together with Samuel)
- Kolam (together with Samuel)
- Bird chase (other models)
- Crowded Hallway (visualization)

I have been active in discussions during meetings both with the group and with our supervisor to move the project forward.

In the report I started to write about the models I helped work on or created on my own. After this was done I started to help out other members where I could with writing as well as doing a lot of proofreading and correcting grammatical errors and misstakes.

Like my other members I have been present during almost all of our meetings as well as a majority of the lectures.