NOVA
IMS
Information
Management
School

MAA

Mestrado em Métodos Analíticos Avançados
Master Program in Advanced Analytics

Simulation and study of an artificial ecosystem based on a cocoa plantation in Ghana, Africa

Simulação e estudo de um ecossistema artificial baseado numa plantação de cacau no Gana, Africa.

Catarina Ferreira Matos Quirino Palha

Dissertation submitted in partial fulfillment of the requirements for the degree of

Master of Science in Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Supervisor: Professor Vítor Manuel Pereira Duarte dos Santos from
Information Management School - NOVA University Lisbon.
Co-supervisor: Professora Elisabete Tavares Lacerda de Figueiredo Oliveira from
Agronomy School – University of Lisbon.

Simulation and study of an artificial ecosystem based on a cocoa plantation in Ghana, Africa

This document was created using the (pdf)LaTeX processor, based on the NOVAthesis template, developed at the Dep. Informática of FCT-NOVA by João M. Lourenço.

To all of those who believe that knowledge takes up no space.

ACKNOWLEDGEMENTS

Most certainly first in the list of people who deserve acknowledgement is my family, they have been there for every step of the way since I can remember, a special mention to my mom, dad and grandmother, that took out every stone of my way they could, I must say I am very privileged and lucky.

To my supervisors, professors Vítor Santos and Elisabete Figueiredo for accepting the challenge and for guiding me through with so much availability and good mood, I could have not chosen better.

To my knitting club, Mafalda Zúquete and Maren Leuthner without you two I most certainly not be here today.

To Leonor Furtado and Marta de Matos, small things have a great meaning sometimes, thank you.

To my dearest buddies Quira Josefa and Loki Lucius, you guys are absolutely therapeutic.

Last but not least, to Duarte de Matos, my beloved one, that watched me every day and never allowed me to give up, again, I am very privileged and lucky.

ABSTRACT

The scientific community has been warning for the dangerous consequences of human's exploitation on the planet. In the past thirty years the planet lost half of its wildness and the greenhouse gases emissions have being rising ever since.

This dissertation was focused on a particular ecosystem, a cocoa plantation in Ghana, Africa. Also, in this research the interaction between the species that live in the ecosystem as well as their rules and characteristics were studied.

The main objective of this dissertation was to build a system which is the result of an artificial ecosystem able to support artificial life and also to analyze how the species respond to an increase in the ecosystem temperature.

The artificial ecosystem was programmed to mimic the cocoa plantation. As such, the artificial life or species which live on it are: the cocoa trees, the midges and the mites.

These species were programmed as agents in an environment, the ecosystem. The system was programmed using python as programming language and it was based on the description that (Russell and Norvig, 2002) made on environments and agents construction and the description that (Rothlauf, 2006) and (Eiben, Smith, et al., 2003) made on genetic algorithms. The visualization of the environment and the agents was programmed in a parallel script that draws the previous objects at every iteration.

Finally, the environment temperature was increased by 1.5°C and by 3.0°C. The first increase had a small effect on the midges population only decreasing by 11% the initial number of individuals. The second increase had a major effect on the midges population leading to the species extinction. On the contrary, these increases of temperature did not affect the development of the mite species.

These results were not a surprise as climate change and global warming accelerates the previous scenarios. In fact, these phenomena are expected to happen a lot more across the world. Humans are causing an unbalance in nature, the planet conditions are changing rapidly, at a rate that nature cannot cope.

Keywords: Artificial Ecosystem; Artificial Life; Environment; Agents; Genetic Algorithms; Python; Climate Change; Global Warming

RESUMO

A comunidade científica tem avisado para as perigosas consequências da exploração humana no planeta. Nos últimos trinta anos o planeta perdeu metade da sua vida selvagem e as emissões de gases de estufa têm vindo a aumentar desde então.

Esta dissertação focou-se num ecossistema em particular, uma plantação de cacau no Gana, África. Para além disso, durante esta investigação a interação entre espécies que vivem no ecossistema assim como as suas respetivas regras e características foi estudada.

O principal objectivo da dissertação foi construir um ecossistema artificial capaz de suportar vida artificial. Depois, observar como é que as espécies que vivem neste ecossistema respondem quando a temperatura do ambiente é aumentada.

O sistema artificial foi programado para imitar a plantação de cacau, como tal, a vida artificial ou especies que lá vivem são: as árvores de cacau, as moscas do cacau e os ácaros.

As espécies foram programadas para serem agentes num ambiente, o ecossistema. O sistema foi programado usando Python como linguagem de programação e foi baseado na descrição que (Russell e Norvig, 2002) fez sobre a construção de ambientes e agentes e nas descrições que (Rothlauf, 2006) e (Eiben, Smith et al., 2003) fizeram sobre algoritmos genéticos. A visualização do ambiente e dos agentes foi programada em paralelo noutro ficheiro que desenha os objectos anteriores a cada iteração do ambiente.

Por fim, a tempertura do ambiente foi aumentada em $1,5°C$ e em $3,0°C$. O primeiro aumento teve um efeito pequeno na espécie das moscas do cacau, apenas diminuiu a população inicial das moscas. O segundo aumento teve um maior efeito na espécie das moscas conduzindo à extinção da mesma. Por outro lado a espécie das ácaros não foi afetada por nenhum dos aumentos da temperatura.

Estes resultados não foram surpreendentes com a aceleração das alterações climáticas e do aquecimento global. De facto, é esperado que estes fenómenos aconteçam cada vez mais por todo o planeta. A actividade humana está a provocar um desequilíbrio na natureza, as condições do planeta estão a mudar rapidamente, tão rápido que a natureza não consegue acompanhar.

Palavras-chave: Ecossistema Artificial; Vida Artificial; Ambiente; Agentes; Algoritmos Genéticos; Python; Alterações Climáticas; Aquecimento Global.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER

1

INTRODUCTION

Since the thirties, the planet Earth has lost more than half of its wilderness. This unfortunate event was the result of the comfortable life that humankind adopted, mostly based on intensive agriculture and activities that lead to greenhouse gas emissions. The human's way of living, created a unbearable unbalance in the natural world. The planet's ecosystems are changing fast, in a way that nature cannot cope (Attenborough, 2020).

Today, year of 2021, scientists are certain of the dangerous results of humankind's actions such as, the risks of drought, floods, extreme heat, poverty for hundreds of millions of people and the horrible consequences of the extinction of species and natural ecosystems. Unfortunately, scientists have been ignored in the past and they are still being ignored today. According to climate scientists, there is only around a decade to avoid global temperatures to be kept to a maximum of 1.5°C. Temperatures above this value will unleash a series of catastrophic events beyond human control. This is known as feedback loop.

A feedback loop is a vicious cycle where the output of a given sequence of events accelerates or decelerates that same sequence of events. In the case of global temperature, a feedback loop is a cycle that accelerates or decelerates a warming trend. A negative feedback decelerates a temperature rise and a positive feedback accelerates it, which is the global warning scenario (Ogunbode et al., 2020).

Table 1.1 presents a comparison of world population, carbon in the atmosphere and the remaining wilderness in the world between the year 1937 and the year 2020.

Year	**1937**	**2020**
World population (billion)	2.3	7.8
Carbon in atmosphere (parts per million)	280	415
Remaining wilderness (percentage)	66	35

Table 1.1: Description on the increase of world population, the carbon in atmosphere and the decrease of wilderness in the world, from 1937 to 2020.

1.1 Dissertation Background

Climate change has a major weight in the future of cocoa production and the communities that live from it, in Ghana, Africa. As climate change advances, it is expected an increase of temperature and unpredictable rainfall which accelerates the degradation of cocoa-producing areas consequently causes loss of trees.

Furthermore, interviews with the cocoa farmers revealed that they already are experiencing higher temperatures and unpredictable rainfall. Temperatures in Africa are projected to increase by 3°C until the year of 2100 (Hutchins et al., 2015).

Due to the urgency of the matter, it is very important to understand which events can actually cause a significant damage to these natural ecosystems and accelerate their collapse. This dissertation was focused on a particular ecosystem, a cocoa plantation in Ghana, Africa. The interaction between three species: the cocoa trees, the midges that pollinate the cocoa trees and the mites which were the midges predators, was studied.

1.2 Dissertation Objective

This dissertation aimed to create a system that simulated an artificial ecosystem which imitates the real cocoa plantation and its living organisms with their respective behavior, characteristics and interactions with other species.

With the computer power available, it is possible to simulate many generations but also to understand how small changes in the ecosystem can damage the species. In general insects are very sensitive to changes in temperature. Therefore, it was also set as a goal to observe what would happen to the species when the environment temperature was as high as in an extreme climate change scenario. Two changes of temperature were applied: the environment temperature was increased by 1.5°C and then by 3.0°C.

1.3 Dissertation Relevance

Despite the enormous potential of artificial life in ecosystem simulations, there are not many simulations focused on the kind of ecosystems explored in this dissertation.

Everything in nature is connected. It is known that the Saharan dust pollinates the Amazon rain forest (Koren et al., 2006) and, therefore, the extinction of one species might lead to the extinction of many others. For instance, in this study, if the midge species goes extinct, it is likely that the mites species also goes extinct because its source of food is gone, as well as the cocoa tree, because it loses the responsible insect for its pollination. Without pollination trees cannot reproduce. Hence, with time this tree species will eventually go extinct.

Besides the natural disaster that is to lose species, there are also economic disasters associated with the loss of biodiversity on the planet. This cocoa tree species is one of the thirteen most important crops in the entire world, where a crop is a cultivated plant that grows on a large commercially scale.

Cocoa insures the income of about 4.5 million families and globally, about 14 million people work in cocoa production. Without the cocoa ecosystem, at least 14 million jobs would be lost, which would have a devastating impact on the global economy (Claus et al., 2018).

In 2013, the cocoa production was produced on around 10 million hectares, which represents 7% of the global crop area. Today the area occupied by this production is bigger and plays an important role in carbon absorption, therefore, these ecosystems might have a important role in climate mitigation (Claus et al., 2018).

1.4 Dissertation Structure

This dissertation has a total of 6 chapters, without introduction included.

- Chapter 2 describes the methodology research used in order to build the system.
- Chapter 3 describes the scientific and theoretical support behind the research. It covers concepts like artificial life, artificial ecosystem, agents and environment and way of interaction, evolutionary algorithms and a biological study on the species used in the simulation.
- Chapter 4 describes the computational method and architecture used in order to build a solid environment able to have agents and to support their interactions, as well as the agents themselves. There is a deep description on how every agent type is defined and structured, together with the possible interactions between agents when in the environment. This chapter also describes the midge agent genetics, namely its chromosome and respective genes, as well as their association to physical characteristics. This chapter also describes the genetic algorithm that backs up the midge reproduction, the agent's fitness function, parent selection, crossover and mutation methods.
- Chapter 5 takes as baseline a system model built under the system foundations rules and parameters. This chapter analyses how an increase on the environment

temperature affects the species. The temperature was increased by 1.5°C and right after by 3.0°C.

- Chapter 6 presents the conclusions taken from the results and their discussion. This chapter also has a small reflection with ideas on how to improve the system in the future.

CHAPTER 2

METHODOLOGY

In order to guarantee the artificial ecosystem's similarity with the real world and its correctness it was necessary to define a comprehensive methodology which allowed to reach the main goal of the research: to build an artificial ecosystem with agents that closely simulated a cocoa plantation in Ghana, Africa, as well as the characteristics and behavior of the species that live within that environment.

The methodology applied had four stages: exploration, implementation, discussion and conclusion. Figure 2.1 presents the methodology workflow used in this research.

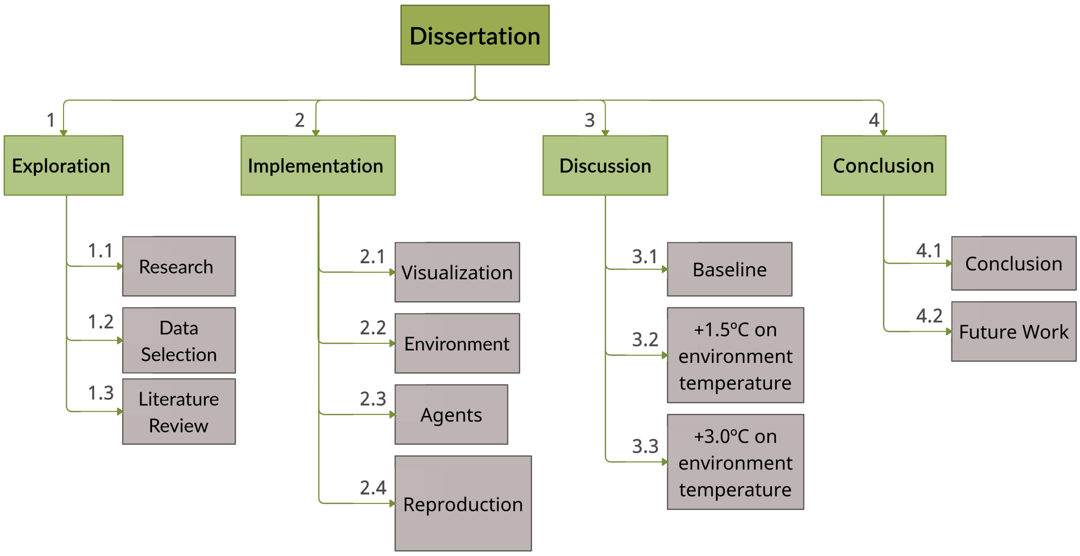

Figure 2.1: Methodology workflow.

The exploration had three steps. First, an initial research was done to understand how much and what information there was about the cocoa's plantation ecosystem and its agents. Second, a data selection was performed to filter from the previous research the relevant information which allowed to build the ecosystem. Third, a literature review with all the organized and relevant data was conceived.

The data collected and used was secondary. The description on the mites behavior and reproduction was collected by (Zhang, 2003), while, the description on the cocoa plantation itself and on the midges behavior and reproduction was described by (Kaufmann, 1975).

Although the current year is 2021, evolution of species takes up millions of years to occur, from 1975 until 2021 these species and ecosystem did not evolve in a way that (Kaufmann, 1975) is no longer valid or updated. This is one of the reasons why climate change is so dangerous for so many species the climate is changing so fast that species cannot adapt and evolve in time of assuring their survival.

The implementation stage also had three steps. First, a program was written in order to the visualize the environment and every agent in it, at every environment iteration. Second, an environment able to support agents was built. Third, agents with their respective behavior and characteristics were created according with the system foundations. Fourth, a genetic algorithm was built to enable the species reproduction.

The discussion had three steps. First, a baseline model was defined in order to act as a constant point for further comparisons. Second, the environment's temperature was increased by 1.5°C. Third, the environment's temperature was increased again by 1.5°C making a total increase of 3.0°C.

Last but not least, the conclusion stage had two steps. First, conclusions were taken on how the species had reacted to the environment temperature increase. Second, future work ideas for the system development were described.

Chapter 3

Literature review

This chapter describes in detail all the scientific concepts and ideas required to build the cocoa's plantation simulation, such as the definition of artificial life, the agent's description, the environment diversity, the interaction between the agents and the environment and the algorithms which the agent uses in order to reproduce in the environment.

3.1 Artificial Life

Artificial Life or ALife is defined as the study of synthetic systems that manifest behaviors and characteristics of natural ecosystems. It complements the traditional biological sciences concerned with the analyses of living organisms by attempting to synthesize life-like behaviors using technologies like computers.

(Langton, 1997) defined ALife as "the study of natural life where nature is understood to include rather than exclude human beings and their artifacts", or in other words, life made by man rather than by nature.

Artificial life is a field which has the purpose the use of a synthetic approach to study "life-as-is-could-be" instead of "life-as-we-know-it". It views life as a property of the organization of matter instead of a property of the matter which is organized.

While biology is concerned with the material basis of life, artificial life is only concerned with the formal basis of life. Also, biology starts from the top, viewing a living organism as a complex biochemical machine and working analytically downwards from there in its pursuit of the mechanisms of life, while, artificial life starts from the bottom, viewing an organism as a large population of simple machines and works upwards synthetically from there to build large aggregates of simple, rule-governed objects that interact with each other (Langton, 1997).

One possible application of artificial life is the creation and observation of organisms, populations and artificial systems built by humans in order to understand how these be generated, modified by external causes, or even evolve through time. This kind of research is very important if humankind wants to extend the life span of the existent ecosystems on planet Earth, but also to understand the evolution of species (Levy, 1992).

3.1.1 Agents and Environments

Biologists define natural ecosystem as the physical place where species interact with each other and compete for resources in order to survive and reproduce. Moreover, an artificial ecosystem is a natural ecosystem but artificially constructed with a computer simulation behind it, where artificial life forms adapt their behavior to best fit their environment (McCormack, 2007).

An artificial ecosystem has two main components: the agents and the environment. An agent can be defined as an object that perceives its environment through sensors and acts upon that environment through actuators. At every instant, or environment iteration, the agent receives inputs from the environment. An input is called percept and a sequence of inputs is called percepts sequence. An agent percept sequence represents everything the agent perceived in a specific environment iteration and its choice of action, which depends entirely on the percept sequence and not on anything that the agent has not perceived (Russell and Norvig, 2002).

Figure 3.1 presents a schema of agent-environment interaction through sensors and actuators.

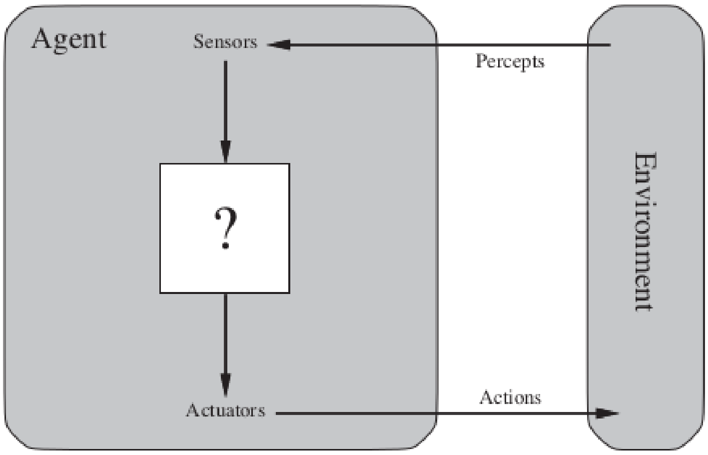

Figure 3.1: Agent interaction with the environment through sensors and actuators. This image was taken from (Russell and Norvig, 2002).

The agent's behavior can be described by an agent function, which is an abstract mathematical description that maps a perception sequence into the actuators. These actuators lead to actions that are to be performed in the environment, at every iteration (Russell and Norvig, 2002).

Internally, the agent behavior, or agent function, is implemented by an agent program which runs within some physical system, also known as the environment. It is very important to keep the agent behavior completely separated from the agent function because it can increase the complexity of the problem unnecessarily (Russell and Norvig, 2002).

The question of how does one agent know what is the "right thing" to do within its environment leads to the definition of rational agent. As mentioned before, while the agent is in the environment, it generates a sequence of actions according to the perceptions that it receives. This sequence of actions causes the environment to go through a different number of states.

If this sequence of actions is desirable then the agent performed well. Otherwise, it did not perform well. In order to evaluate the agent performance, a performance measure that evaluates every sequence of environment's states is used.

It is important to mention that the performance measure is calculated at every environment state, not at every agent state. By defining success in terms of an agent's opinion on its own performance, an agent can fool itself that its performance was perfect. As general rule, it's better to build a performance measure according to what is actually intended in the environment rather than how a agent should behave. To define a rational agent are needed:

1. The performance measure that defines the criterion of success.
2. The agent's prior knowledge of the environment.
3. The actions that the agent can perform.
4. The agent's perception sequence in that instant.

Therefore, a rational agent is an agent that for each perception sequence, selects the action that maximizes its performance measure. Importantly, choosing an action that maximizes its performance is completely different from the agent knowing the actual outcome of its actions, which, in reality is of course impossible, this kind of agent is called an omniscient agent.

The definition of rational agent not only gathers information but also learns as much as possible from what it perceives. An agent's initial configuration may reflect some prior knowledge of the environment but, as the agent gains experience this may be modified. A rational agent should also be autonomous, meaning that it should learn what it can in order to compensate the incorrect prior knowledge. After sufficient experience of its environment the behavior of a rational agent can become independent from its prior knowledge.

A problem to which a rational agent is the "solution"is called a task environment, which specification includes the performance measure, the environment, the actuators, and the sensors, also known as PEAS (Russell and Norvig, 2002).

3.1.2 Agent structure

An agent program should be able to implement, at every environment iteration. The agent function, which takes as input the current perception from the sensors and returns an action to the actuators. The agent program only takes the current percept as input, because nothing else is available from the environment. If an agent is to depend on the entire perception sequence, it will need to remember all of the percepts.

The computing device with the physical sensors and actuators is called architecture. The architecture makes way between the preceptors and the sensors in the program, in addition to running the program and feeding the action choices to the actuators.

$$agent = architecture + agentProgram \tag{3.1}$$

A way to keep track of the perception sequence is to index it to a table of actions and decide what to do. This table must contain the appropriate action for every possible perception sequence. This approach is not a good idea, let P be the set of possible percepts and T the life span of the agent, the table will contain too many entries if many different percepts are considered.

$$NumberOfEntries = \sum_{t=1}^{T} |P|^t \tag{3.2}$$

Instead, the agent selects the actions based on the current percept putting aside the idea of memory. This kind of agent is called a simple reflex agent, which is the simplest kind of agent that can be built. The simple reflex agent establish rules between the environment conditions and the actions to take in response. Moreover, there is a function that interprets the input and returns an abstract description of the current state according to the perception sequence. Next, another function matches the abstract description to a set of rules and returns the rule in the set that matches the given description. These rules lead to the execution of the possible actions (Russell and Norvig, 2002).

Figure 3.2 presents a schema of a reflex agent-environment interaction through sensors and actuators.

The use of simple reflex agent as an agent program is a good approach, because, it is an idea very simple to implement. This approach can fail if the correct decision cannot be made on the basis of the current percept, which means the environment is not fully observable. An environment task is fully observable if an agent's sensor gives access to the complete environment state at each point in time, or, in other words, if

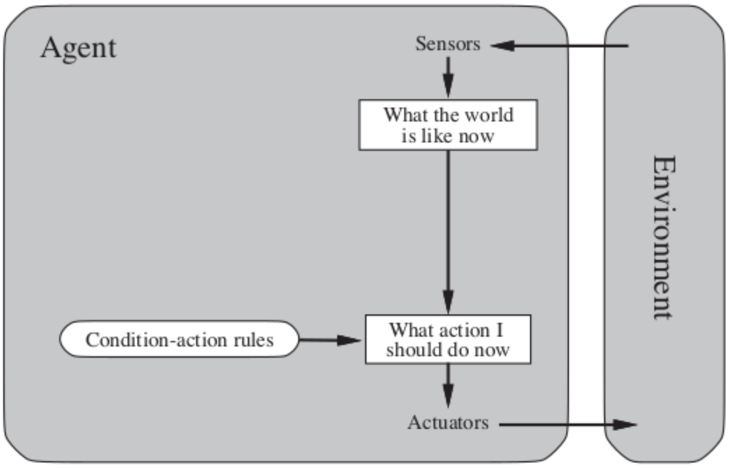

Figure 3.2: Simple reflex agent interaction with the environment through sensors and actuators. This image was taken from (Russell and Norvig, 2002).

the sensors detect all aspects that are relevant for the choice of action. Otherwise, the environment task is called partially observable.

The most effective way to deal with partial observable environment tasks is for the agent to keep track of the part of the environment that it cannot see. For this to be possible, the agent should have an internal state that depends on the percept history. In order to update this internal state through time two types of knowledge are required to be encoded in the agent program. First, it is necessary to have information about how the environment evolves independently of the agent. Second, it is needed information about how the agent's actions affect the environment. This knowledge of "how the environment works" is called a model of the environment and an agent that uses such model is called model-based agent.

Figure 3.3 presents a schema of a model-based agent.

Yet, knowing the current environment's state might not be enough to decide which action to take, for that, alongside with the internal state the agent might need a goal information that describes the situations which are desirable. The agent's program can combine this goal information with the model in order to choose the action that achieves that goal. This type of agent program is called goal-based agent. This agent program type is extremely flexible because the knowledge that supports the decisions is explicitly represented and it can be modified.

Figure 3.4 presents a schema of a model-based and a goal-based agent.

Although goals provide a binary distinction between what is "correct" or "incorrect", goals alone might not be enough to generate a high quality behavior in some environments. For this purpose, an agent's utility function can be created. An agent's utility function is an internalization of the performance measure. If the internal utility function and the external performance measure are in agreement then an agent that chooses actions to maximize its utility will be rational according with the external

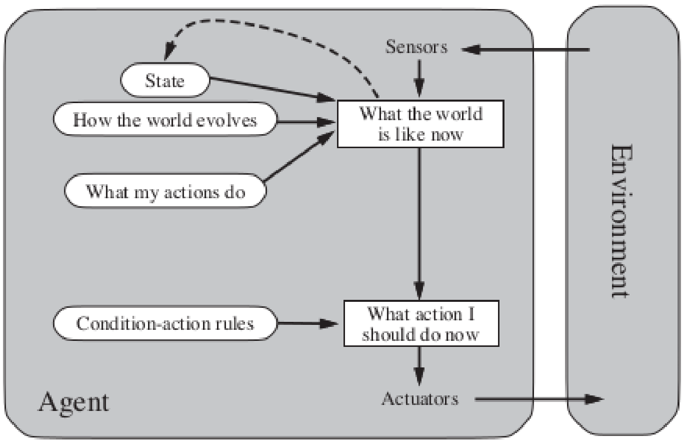

Figure 3.3: Model-based reflex agent with internal state. The picture shows how the current percept is combined with the old internal sate in order to generate the updated current state based on the agent's model of how the environment work. This image was taken from (Russell and Norvig, 2002).

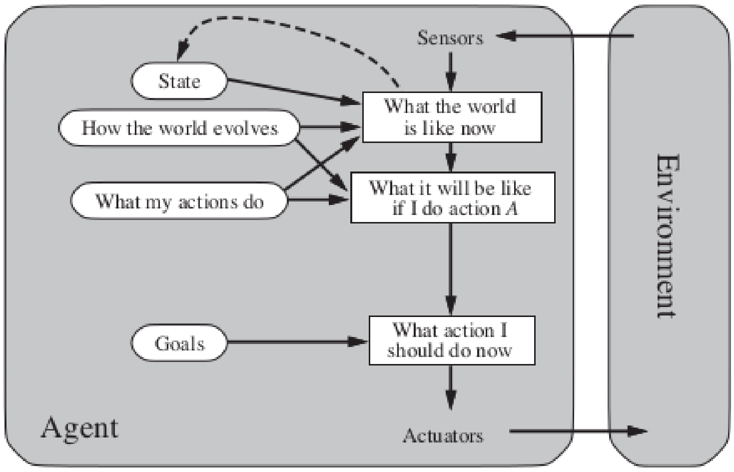

Figure 3.4: Model-based and goal-based reflex agent, which keeps track of the environment state as well as the goals that is trying to achieve. This image was taken from (Russell and Norvig, 2002).

performance measure. Notice that this is not the only way of an agent to be rational. Just like the goal-based agents an utility-based agents has some advantages in terms of flexibility and learning Even if the goal is inadequate an utility-based agent can still make rational decisions.

First, when there are goals conflicts, where only some of which can be achieved, the utility function specifies the appropriate outcome. Second, when there are several goals that the agent aims for and none of them can be achieved with certainty, the utility provides a way in which the likelihood of success can be weighed against the importance of the goals. A rational utility-based agent chooses the actions that maximize the expected utility of the actions outcome, which is, the utility the agent expects given the probabilities and utilities of each outcome.

Figure 3.5 presents a schema of model-based and utility-based agent.

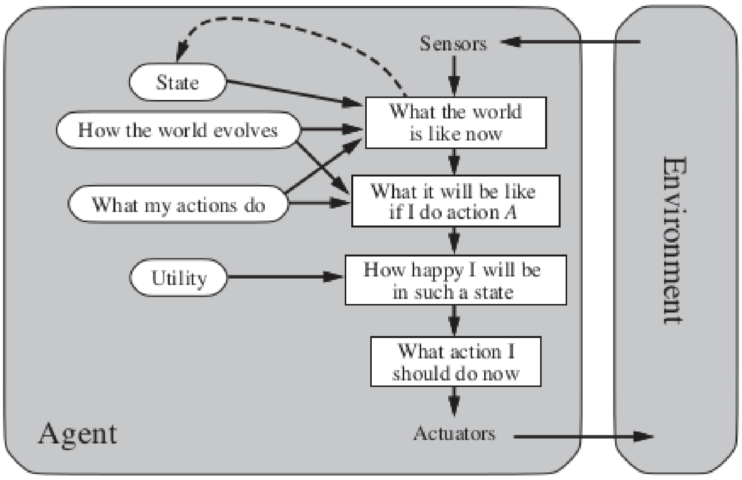

Figure 3.5: A model-based and utility-based agent. They use a model of the environment along with an utility function that measures the agent preferences among states of the world then, it chooses the action that leads to the best expected utility. This image was taken from (Russell and Norvig, 2002).

To sum up, there are four possible types of agents programs, all described above:

- Simple reflex agents.
- Model-based reflex agents.
- Goal-based agents.
- Utility-based agents.

Where simple reflex agents respond directly to percepts, model-based reflex agents keep an internal state to track aspects of the world that are not evident in the current percept, goal-based agents act in order to achieve their goals and utility-based agents try to maximize their own concept of correct.

3.1.3 Learning agents

Back in 1950, Alan Turing had the idea of programming his intelligent machine by hand, of course, he quickly understood how much work this might take. Therefore, he proposed to build learning machines and then to teach them. Learning allows the agent to operate in initial unknown environments and to become more competent than its initial knowledge alone allows.

All agents can improve their performance through learning. A learning agent is divided in four components: critic, learning element, problem generator and performance element.

The performance element is responsible for the selection of external actions, which was defined before as the agent itself. Therefore, the performance element takes as input the percepts and decides on which action to take. The learning element is responsible for making the improvements. It uses the feedback from the critic, which says how the agent is doing with respect to fixed standard performance and also determines how the performance element should be modified in order to do better in the future. Notice that the critic is essential because the percepts alone cannot provide indication on the agent's success. Also, the performance standard distinguishes the input percept as reward or penalty which, as expected, provides a feedback on the quality of the agent's behavior.

At last, the problem generator is responsible for advising actions that will mostly lead to new and informative experiences, in other words, the problem generator allows the agent to explore new actions even if these actions are not the best pointed out by the performance element.

Figure 3.6 presents a schema of a learning agent.

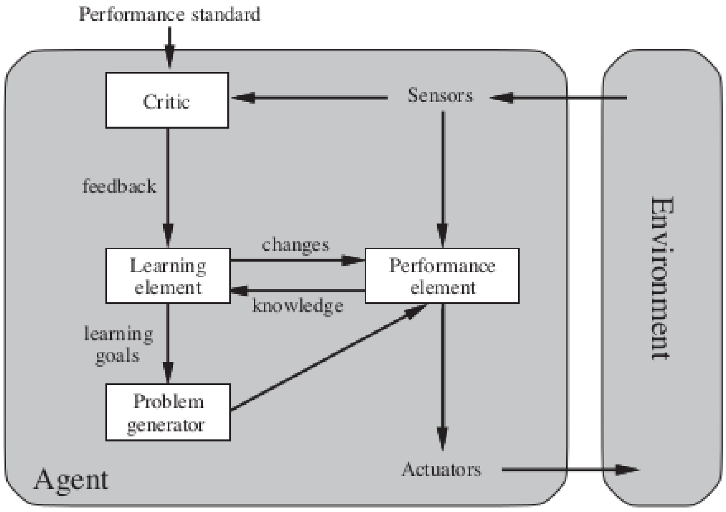

Figure 3.6: A learning agent schema. This image was taken from (Russell and Norvig, 2002)

3.1.4 Genetic Algorithm

Given a population with a finite number of individuals within an environment which has limited resources, the competition upon these resources generates what, in computation, is called an evolutionary algorithm. This competition also origins a phenomenon by (Darwin, 2004).

Natural selection backs up those individuals that better adapt to the environment conditions, these are more likely to survive, and ultimately leads to the species evolution. Evolution ensures that individuals vary genetically and physically from each other. This is advantageous to a population because it enables some individuals to adapt to the environment ensuring the survival of the species.

In computation, this phenomenon is called survival of the fittest. Those individuals that better adapt to their environment are more likely to survive, to reproduce and to pass their genetic material into to the next generation (Eiben, Smith, et al., 2003).

When two individuals reproduce in order to create offspring, this offspring will carry a combination of genetic material from both parents. A chromosome is the place where all the information of an individual is stored, usually is represented as a string of symbols where each symbol is called an allele. Therefore it can be said that a chromosome is a sequence of alleles (Rothlauf, 2006).

Genotype and phenotype are two other important concepts that also need to be defined. The genotype is the set of genetic characteristics encoded in a DNA's organism, in other words, it is a collection of genes stored into chromosomes. All the information is stored in the chromosomes, which is what makes possible to describe an individual genetically. The phenotype is the set of physical characteristics that result of the interpretation of the genotype in the context of the environment where the organism lives (Langton, 1997).

The region in the chromosome responsible for a specific phenotype characteristic is called a gene (Rothlauf, 2006).

Figure 3.7 presents an example on a possible chromosome representation.

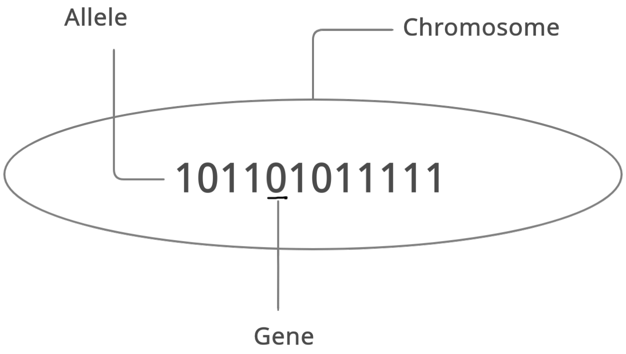

Figure 3.7: Example on a binary representation of a chromosome, allele and gene. This image was inspired by (Rothlauf, 2006).

An example of an evolutionary algorithm is a genetic algorithm (short GA). A genetic algorithm can be defined as the method that allows to generate offspring from a parent population. The principal concern is to produce new generations that have a high probability of surviving in the environment (Langton, 1997).

Genetic algorithms imitate the process of natural selection and reproduction described in the previous section. Usually, these algorithms are nature-inspired methods used for optimization problems, copying basic ideas of life by applying genetic operators like crossover, mutation and selection of the parents' chromosomes. A primary element of evolution is an operator named crossover. It is throughout crossover that offspring is created with a combination of both parents, by randomly choosing one or more alleles to interchange, in order to create one or two new chromosomes that represent the offspring. This operator helps maintaining the diversity of the population by bringing up the best combinations over time (Rothlauf, 2006).

Additionally, mutation aims to create offspring by randomly flip, delete or insert alleles in the chromosome, these can occur in every allele with a certain probability (usually with a probability of 0.001). These random variations in the alleles may introduce changes that sometimes can originate a huge advance in evolution by refreshing the population, with the introduction of new patterns in the chromosomes (Rothlauf, 2006).

On the genetic algorithm context, the initial population, or generation zero, is a set of random chromosomes that represent solutions for a certain problem. These solutions were referred before as individuals. Next, it is required to define a fitness function that evaluates every individual in every generation according to the problem purpose. The fitness of each individual is independent from each other (Rothlauf, 2006).

Once the individuals meet the criteria that allow them to start breeding, three genetic operators are applied to the population: selection, crossover and mutation. The outcome of these operations is the next generation, or the offspring, which should be better individuals than the current ones.

The selection operator is responsible for choosing individuals from the current population that can reproduce and have better changes of surviving, or in other words, is responsible for choosing the fittest individuals. For instance, roulette wheel selection and rank-based are heavily used selection methods (Rothlauf, 2006).

The crossover operator creates offspring from the selected individuals, or parents. The mutation operator might randomly introduce a change to one or more new individual chromosomes. Once the reproductive cycle is over, the fitness of each individual is calculated again (Rothlauf, 2006).

Last but not least, in order to end the algorithm, a stopping condition is required. Until this condition is met, the algorithm loops on itself. Without loss of generality, a genetic algorithm follows the following steps:

1. Population initialization.

2. Calculation of the fitness of each individual.

3. Application of genetic operators.

4. Repetition from 2. until a stop condition is met.

In the year 1995, the idea of evolutionary algorithm took a step further. A genetic algorithm was used to be a method that achieves the creation of solutions for a certain problem. Now, together with the artificial life, the genetic algorithm is the solution itself (Russell and Norvig, 2002).

3.2 *Forcipomyia squamipennis*

All around the world there are diverse pollinators species of *Theobroma cacao* L. (also known as cocoa tree) living in different temperatures and regions, but *Forcipomyia squamipennis* Ingram & Macfie (Diptera: Ceratopogonidae), which common name is the chocolate midge. It is known for being the most important of them all, due to their constant feeding, reproduction and development near these trees (Kaufmann, 1975) (Frimpong et al., 2011).

Before starting to describe this species it is important to notice that like every insect this midge is poikilotherm and ectotherm. It means, the species internal temperature varies considerably and also the main heat source comes from the external environment.

Therefore, the development of this midge as well as its reproduction, longevity, mortality rate and population characteristics depends directly on external environment's temperatures.

3.2.1 Biological Characteristics

A complete midge life cycle has about 28 days at temperatures of 20 – 25°C. This species is holometabolous, it means its life cycle is split into four stages: eggs, larvae, pupae and adult. Although *F. squamipennis* is abundant and reproduces throughout the entire year, the population drifts seasonally. The largest populations occur in the rainy season and the highest midge numbers recorded were from June to November, declining sharply in February and March (Frimpong et al., 2011).

The following stages descriptions were made and collected by (Kaufmann, 1975).

3.2.1.1 Egg

Usually, the eggs are laid in humid places like rotten wood, hallow dead trees roots, old trees cracks, decayed cocoa husks or even moss. The eggs take around 2 – 3 days until hatch at temperatures of 20 – 25°C. Scientists believe that each female can produce about 200 eggs, from which, about 5 – 6% of the eggs become adults.

3.2.1.2 Larva

This stage is often seen during the rainy season, but also exists during the dry season, which can be seen in places that did not dry out. Therefore, as expected, larvae live in a dark and humid environment. The midge lives in this stage during 10 – 15 days at temperatures of 20 – 25°C and then pupate.

They feed on cocoa husks, decayed wood, banana stems and other vegetables. Some scientists say larvae feed on fungi, bacteria and algaes rather that plants products.

3.2.1.3 Pupa

The pupa is yellow, and it can be found in dry places. Although it seems a delicate phase, the pupa is able to defend itself when under attack. This period takes around 2 – 3 days at temperatures of 20 – 25°C until the emergence of the imago.

3.2.1.4 Adult

The midge body in this stage is black and yellow. The adult stage last a week, 7 – 8 days at temperatures of 20 – 25°C. The species produces at least 12 generations per year. The midges can live 24 hours without food or water and were never observed preying upon other insects or animals.

Usually, males outnumber females when spotted on cocoa flowers depending on the cocoa plantation with a ratio 1 : 4 and also during reproduction with a ratio 1 : 2, where the ratio means female:male. Mating occurs during swarming and a male often mates with 2 – 3 females during one swarming.

Right after the mate, a female either remains in the resting place or goes elsewhere to lay the eggs. Table 3.1 describes a midge stages duration at 20 – 25°C.

Midge Stage	Duration (days)
Egg	3
Larva	10
Pupa	3
Adult	8

Table 3.1: Midge stages and respective duration in days, at 20 – 25°C.

3.2.2 Nutritional Value

It is necessary to define a parameter that measures the midge life state. This parameter needs to measure the cost that essential behaviors represent to the agent, namely the costs of flying around or reproducing.

In some countries, insects are used as a food source. Although there is no research on *F. squamipennis* nutritional values, probably because it is not eaten, there are some similar insects, more specifically, insects of the same order that are used in the food system of these countries, namely the black soldier fly, *Hermetia illucens* (Diptera, Stratiomyidae).

In general an adult insect weights between $77g - 100g$ where per $100g$ there are $409.78kcal$ (Dobermann et al., 2017).

Due the lack of data on *F. squamipennis* different weights stages, in order to calculate the nutritional value of every stage the lengths described in table 3.2 will be used (Dobermann et al., 2017).

Mite Stage	Length (mm)
Egg	0.22
Larva	0.5
Pupa	3.5
Adult	0.22

Table 3.2: Midge stages and respective lengths in millimeters.

The nutritional value calculation starts with the adult stage, by generating a random number between $77g - 100g$. Knowing the adult weight, the real nutrition value can be calculated by simply doing a proportion using a rule of three, once it is known that an adult has $409.78kcal/g$.

$$NutritionValue_{Adult} = \frac{409.78 RandomAgentWeight}{100} \tag{3.3}$$

Then, the adult nutrition value is associated with its length. The respective nutrition value for the other stages can be defined using again a rule of three. The following equation is valid when state is other than adult.

$$NutritionValue_{State} = \frac{Length_{State} NutritionValue_{Adult}}{Length_{Adult}} \tag{3.4}$$

3.2.3 Ecosystem

In order to build the artificial ecosystem, it is crucial to describe in detail the ecosystem where the midge population lives. This research was based on investigations made in Ghana, where there are two main seasons, wet and dry season. Adult midges can be found in the shade of large trees, on cracks in the trees, in decayed wood, in hollow rotten trunks and, also, in cocoa husk piles (Kaufmann, 1975).

Overall, previous investigations on the midge biology had shown that the perfect ecosystem is a place where a great amount of shade and a high level of humidity exists (Kaufmann, 1975).

Apart from these important conditions, the ecosystem includes the cocoa tree, *T. cacao*, but also other types of trees that contribute to the large shades mentioned before, such as:

- *Piptadeniastrum africanum*, odahoma,
- *Chlorophora excelsa*, odum,
- *Pseudospondias microcarpa*, katawani,
- *Bosqueia angolensis*, okuri,
- *Ceiba pentandrd*, silk cotton.

There are some important aspects to notice about the cocoa tree, the cocoa flowers are available from August to November and the *T. cacao* grows mainly in the South of Ghana, where the rainy season goes from April until mid-November (Frimpong et al., 2011).

3.2.4 Predators

As every other species, midges have predators, which are: mites, ants, springtails, dermapterans and centipedes. However, the ones that make the most damage to this midge are the mites. A mite development goes from egg to adult in a fortnight at temperatures of 15–28ºC and its development consists in four stages: egg, protonymph, deutonymph and adult (Zhang, 2003).

Table 3.3 describes a mite stages and respective duration, at 15 – 28°C.

Mite Stage	**Duration (days)**
Egg	3
Larva	2
Protonymph	1
Deutonymph	2
Adult	10

Table 3.3: Mite stages and respective duration in days, at 15 – 28°C.

Except for the egg stage, every other mite stage is a midge predator, but they only feed on egg, larva or pupa midge stages. As mites are terrestrial beings and adult midges fly and stay on higher levels on the trees they never experience each other in the environment (Kaufmann, 1975).

As well as the midge, mites start their life cycle near the resting places, the trees, as eggs. This stage does not move or interact with the environment. When a mite reaches the deutonymph stage males are attracted to the female sex pheromone. They guard their territory and fight against any other males.

When a mite reaches the adult stage it can reproduce and a female can lay until 10 eggs per day. The unfertilized eggs originate males, which are haploid, and the fertilized eggs originate females, which are diploid (Zhang, 2003).

CHAPTER 4

SYSTEM DEVELOPMENT

The system intended to simulate a Ghana's cocoa plantation ecosystem by using two completely separated structures: the environment and the agents. The environment was the cocoa plantation itself and was responsible for every possible interaction between the agents. An agent was everything else that co-exists within that environment. The possible agents in the simulation were: a mobile agent, a tree, a day, or a year. In this simulation, a day had 24 hours, a year had 360 days and a mobile agent was a special type of agent that could move around the environment. These special agents could be of two types: midges or mites.

4.1 System Foundations

This section highlights the assumptions that emerged from the literature review. These assumptions were used as rules or as parameters of the environment or agents required to develop the system:

- The wet season takes place from April to mid-November and the dry season from December to February.
- Midges survive at 20 – 25°C and mites survive at 15 – 28°C.
- Adult midges pollinate and feed on tree flowers early in the morning and late in the afternoon.
- The adult midges flight range never exceeds 6 meters from the tree and the larvae cannot go far from the tree.
- The midges egg and pupa cannot move or eat, as well as mites egg.

- The midges and mites life cycle and respective stages duration.
- Midges and mites reproduction happens near the resting places. And that it can happen anytime of the day and during the entire year.
- Midges have external fertilization and mites have internal fertilization.
- Apart from the egg, all mite stages feed on midge's egg, larva or pupa.
- Mites and midges lay eggs near the resting places.
- Genetic algorithm idea and design.
- A PEAS task environment method.
- A Simple reflex agent program.

4.2 System Construction

The system was built by using Python as the programming language and the following libraries: random, math, numpy and pygame.

The random library allowed to generate pseudo-random numbers. The math library allowed the use of mathematical functions. The numpy library allowed to use multidimensional array objects and insured a faster manipulation of mathematical objects. Last but not least, the pygame library allowed to visualize the simulation itself by drawing the objects at each environment iteration.

Python was the programming language chosen because it has great libraries that allowed the use of complex objects in a simple way. This language has good readability and is very popular among programmers. Therefore, it is in constant grow and has a strong community support.

A class named visual was programmed by using the pygame library in order to draw alongside with the environment all the agents in it. This class insured to the user the visualization of every object within the simulation at every environment iteration.

The visual class when initialized drew a white window with "Cocoa Plantation Ecosystem" as title and with a size given by a width and a height. This window was the representation of the environment itself. Figures 4.1 and 4.2 present what the user should see when a simulation within an environment class, a day class and a year class is created. Visually, the environment was a white window the day and year classes were represented by a color bar where each color represents a stage. In fact, every change of color that may occur within this simulation represented a change of stage for the respective agent.

Alongside the environment all agents with their respective characteristics and perceptions-rules results were drawn. The midges were represented by polygons which changed colors at every stage. Mites were horizontal triangles which also

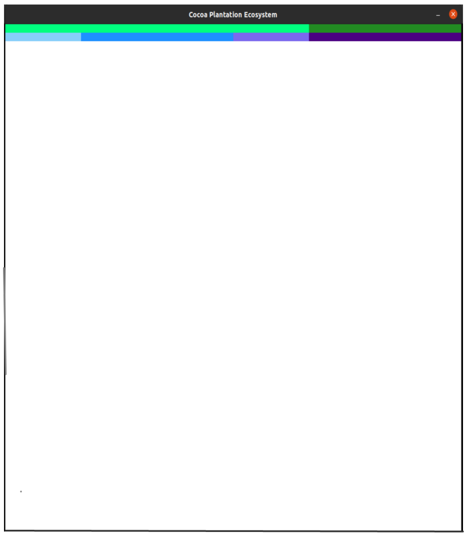

Figure 4.1: Visual result of an environment class with a day and a year agent. The first bar is the representation of the year agent where each color represents a stage, the second bar, right bellow, is the representation of the day class where each color also represents a stage.

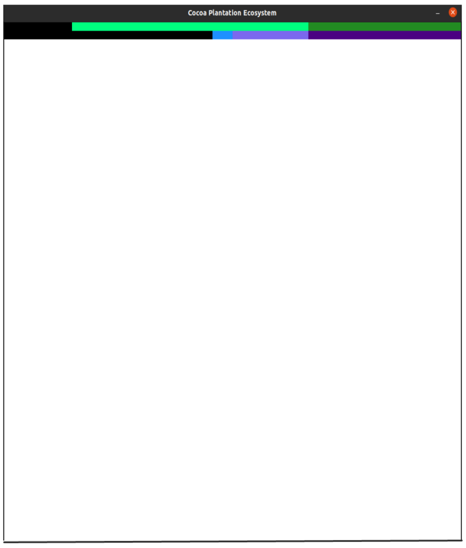

Figure 4.2: An environment with a day and a year agent. The black bar upon the day and year agent represent the life cycle course of each agent in the simulation. At each environment iteration, the bar moves forward, when the black bar reaches the end, starts again.

changed colors at every stage. Trees were green circles only changing colors when the season, or year stage, changed.

Table 4.1 describes the agents types with the respective stage and color chosen to represent them in the system.

Figures 4.3 and 4.4 show the visual representation of trees and all possible mobile agents, and a complete environment with all possible agents on it.

4.3 System Architecture

This section describes in detail all the 11 classes and respective methods present within the system which allowed to simulate the cocoa plantation ecosystem. Figure 4.5 represents the system classes diagram with the respective relationships between classes.

4.3.1 Environment

The simulation began with the creation of an environment class, initialized with an age variable set to zero, which was increased at each environment iteration, a width and height variables set to 1024 each which set the environment boundaries.

At each iteration, the environment knew every piece of information about the agents that lived in it. Also, when the environment class was initialized, a dictionary

Agent Type	Stage	Color	RGB
Midge	Egg	Burlywood	(222, 184, 135)
Midge	Larval	Wood Medium	(165, 128, 100)
Midge	Pupal	Gold	(255, 215, 0)
Midge	Adult	Orange	(255, 165, 0)
Mite	Egg	Dark gray	(169, 169, 169)
Mite	Larval-Protonymph	Steel Blue	(35, 107, 142)
Mite	Deutonymph-Adult	Black	(0, 0, 0)
Tree	Wet season	Spring green	(0, 255, 127)
Tree	Dry season	Forest green	(34, 139, 34)
Year	Wet season	Spring green	(0, 255, 127)
Year	Dry season	Forest green	(34, 139, 34)
Day	Dawn	Light sky blue	(135, 206, 250)
Day	Day	Dodger blue	(30, 139, 34)
Day	Twilight	Medium state blue	(123, 104, 238)
Day	Night	Indigo	(75, 0 130)

Table 4.1: System color representation of every type of agent stage.

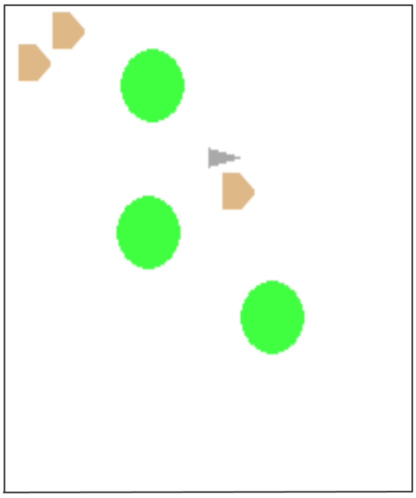

Figure 4.3: The trees are represented by a circular shape, the midges are represented by a pentagonal shape and the mites are represented by a triangular shape.

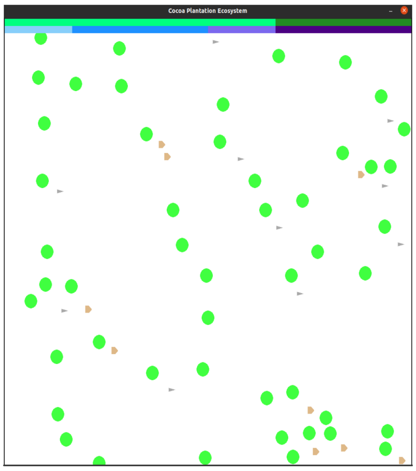

Figure 4.4: An example of an environment with trees, midges and mite agents.

able to hold the agents information was created. Consequently, when an agent class was created, the respective information was appended to this dictionary. By using this dictionary and some class methods, the environment was capable of knowing every detail that it needed from an agent in order to perform a certain task.

Besides knowing the agents which lived in it, the environment also knew when eliminate an agent from it. Thus, along with the environment class a list object was

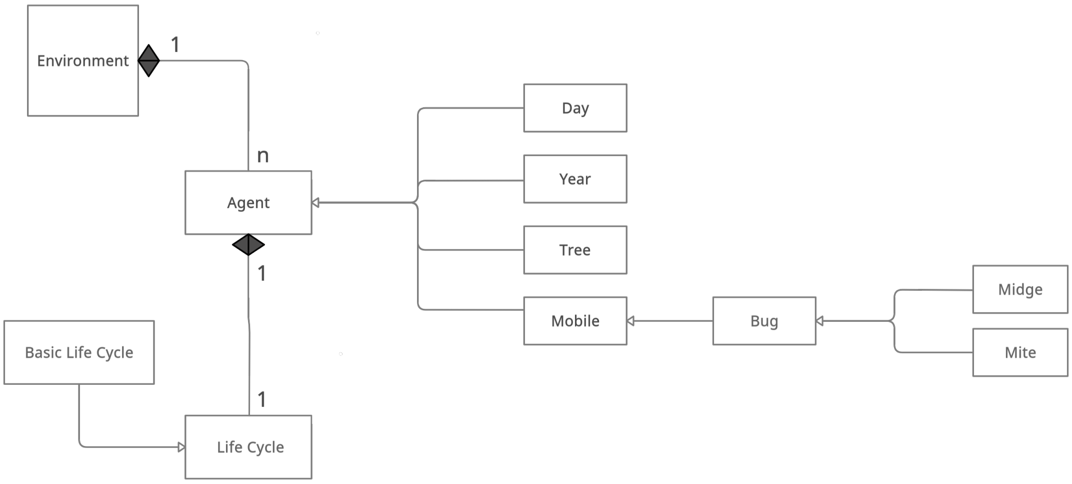

Figure 4.5: Diagram classes.

also initialized to hold the agents ids that were killed during the simulation. Finally, the environment was also created with an initial temperature variable set to 23°C, but, this temperature could vary according with the year stage.

The environment class had functions which allowed it to perform operations at each iteration:

- Set temperature: Does not receive any arguments as input. This function looks up at the year stage and changes the temperature accordingly. When the stage is wet the temperature is a random number between 20 and 25, else, if the stage is dry the temperature is a random value between 21 and 30. At the end, the function returns the set environment's temperature.
- Map: Given a list of agents id and an operation, the map function applies the previous operation to every agent in the list and returns a list with the result after applying the operation.
- Filter: Given a list of agent ids and a condition, the filter function applies the given condition to every agent in the list and returns a list with the agents that satisfy the previous condition.
- Add agent: Given a new agent, adds it to the environment by appending it to the agents dictionary described previously.
- Kill agent: Given an agent id, removes the agents from the environment and appends its id to the kill agent list described previously.

- Scan at: Given an agent position and an agent radius the function returns a set of other agents within its radius.
- Random position near agent: Given an agent id and a maximum distance returns a random position within the maximum distance and near the agent id.
- Lay agent: Given a position and an agent, the lay agent function adds an agent in a random position near the closest tree id.
- Agent distance: Given an agent id and a position, the agent distance function returns the distance between the agent and the position.
- Get nearest agent: Given a position and an agent type, the function filters the agents around to be of the chosen type and returns the closest agent id from the given position.
- Get position: Given an agent id, the function returns the agent position.
- Parent selection helper: Given an agent id, this function builds a dictionary with the information about the chromosome and energy perception of all the agents within the given agent id sensor range. The function returns the dictionary. This dictionary will later feed the genetic algorithm's parent selection function.
- Midge tree sensation: Receives nothing as input, this function looks up at the day agent stages. Returns true if the stage is day or night and returns false otherwise.
- Get agent gender: Given an agent id, the functions returns its gender.
- Get agent stage: Given an agent id, the function returns its stage.
- Get agent chromosome: Given an agent id, the function returns its chromosome.
- Get agent energy perception: Given an agent id, the function looks up its perception list and returns the energy perception value.
- Get random position: Given a radius and a number of retries, the function returns a random position inside the environment.
- Step: Does not receive input arguments, the function uses the set temperature function in order to define the environment's temperature. Then it increases the environment age by one and updates the the agents dictionary and the kill list according to the agents in the environment.

With the environment class creation, the notion of time in the simulation was measured by the environment step function, where each environment step represented a time unit, or an iteration. Moreover, at each environment iteration every other agent also took a step further by activating their step function. This concept of time did not correspond to the reality. Hence, with the environment class a day and a year class were created. Both day and year classes were static agents in the environment and both had a life cycle associated with them. Actually, every agent type had a different life cycle associated with, which are described in the following section.

4.3.2 Life Cycle

The life cycle was also represented by a class which started the simulation with zero age, with a function that returned the current life cycle stage and a step function that made its own age increase by one at each environment iteration.

The life cycle class was a super class to a class named basic life cycle, it means the basic life cycle class inherited all attributes and properties from the life cycle class.

The basic life cycle class was initialized with a repeating variable set to true. When this variable was true the life cycle looped on itself and it never ended, when the variable was set to false the life cycle ended when it reached its total duration.

Along with the repeating variable, a duration list with the duration of each stage, a change stage list which holds the environment iteration number where stages should change, a count cycle variable equal to zero, a stage changed variable set to true, a cycle duration variable that is set to be the last position of the change stage list and a current stage set initially to zero were also initialized. In order to push the life cycle forward were defined four functions:

- Count stages: Returns the length of duration list, in other words, how many stages an agent has.

- Total duration: Returns the cycle duration variable. This cycle duration variable, saves the last position of change stage list, this value is the total duration of the cycle.

- Cycle age: Returns the remainder of age variable by cycle duration variable. This remainder value helps to control the change of stage.

- Step: This function first activates the life cycle class step function, sets the stage changed variable to false and sets a cycle age variable with the returned value of cycle age function. If the cycle age variable is zero and the repeating variable is true the current stage variable stays zero, the count cycles variable is increased by one and the stage changed variable is set to true. If the cycle age variable is equal to the change state value in the current stage position of the change stage list, the current stage variable is increased by one and the stage changed variable is set to true. Otherwise, it does nothing.

4.3.3 Agents

Before running the simulation, the user is able to choose not only the number of agents that are to be initialized along with the environment, but also their type.

Each time an agent class was initialized the respective agent information was added to the correspondent variable in the environment class. Also, when an agent class was created, an id that uniquely defines the agent was assigned to it, an age variable set to zero and an energy variable set to 100 energy units. Besides that, the agent class also had a life cycle variable that held the duration of each agent life span and respective stages.

Because the agent class intended to imitate a living being, it was necessary to define some functions in order to insure basic agent functionalities. For instance, the metabolism function increased the age variable by one at each environment iteration and activated the step function from the life cycle class in order to push forward the agents life cycle.

Besides metabolism, an agent needed to respect certain basic physic laws for that, a physics function was created to define these rules. At each environment iteration, the agent was able to receive inputs from internal sensations and from the environment called percepts. The update perception function defined how these percepts were to be measured according the agents characteristics and rules. This function returned a list with all the percepts the perception list, at each environment iteration.

The previous perception list was given as input to a function called behave. The behave function held the set of rules which allowed the agent to decide which action to perform according to the internal and external sensation given by the perception list.

Finally, the step function combined all the previous functions. If an agent had an energy greater than zero, an environment to live in and a defined life cycle the step function executed the following functions by this exact order: update perception, behave, metabolism and physics.

There were four types for possible agents: mobile agents, trees, year and day. The mobile agent class had a subclass named bug which defined agents that could be of two types: midge or mite, also classes.

A mobile agent class was also a subclass of the agent class. When a mobile class was created it was initialized with a life cycle, a position, a radius, a weight, an energy and a maximum velocity that the agent could achieve. This class also had a variable called heading which was set to zero and two more variables, velocity and acceleration, also set to zero. When the agent position was not given, a new random one inside the environment boundaries was generated. Also, if the maximum velocity was not given the maximum velocity could not pass the agent's radius value.

As well as any agent, a mobile agent needed to respect its physics rules. Therefore, the mobile agent class also had a physics function. The physics function, at each environment iteration, increased the velocity variable by adding the acceleration value to the previous velocity value. The velocity was, by definition, always between zero and the maximum velocity set previously. Also, at each iteration the agent position was calculated and regulated by verifying if that position was within the environment boundaries or that there was not other agent on it. The mobile agent also had a metabolism function that called out the agent class metabolism function and slightly decreased the agent energy at each environment iteration, consequence of being alive or of moving. Table 4.2 describes this loss of energy at each environment iteration.

Metabolism	Energy Loss
Being alive	−0.01
Moving	−0.015*Aceleration

Table 4.2: Mobile agents energy loss by metabolism function.

The mobile agent had a behave function, a head to function which was responsible for pointing the agent's head to the a given position, in other words, this function ensured that the agent moved in the direction of its head, for instances, if the heading was pointing to the left, the agent would move to the left and never in other direction.

Finally, the bug class inherited all the attributes and properties from the mobile class. This class was also initialized with an initial energy variable that saved the initial energy value of the respective agent and an alive variable set to true, which was turned to false when the agent's age was greater or equal than the agent's cycle duration variable or its energy lower or equal than zero. This class had only two functions. The first was the metabolism function, that activated the mobile agent metabolism function. The second one was the step function, that activated the mobile agent step function and verified if the agent alive variable was still true. If the alive variable was false the agent was no longer alive and the environment removes the agent from itself.

4.3.3.1 Agent Perceptors

At each environment iteration the mobile agents, the midges and the mites, update their perceptions according to what existed in their sensor range's surroundings and within their internal sensations. These perceptions were key for the way the agent behaved. The agent behavior was the bridge, or the connection, between the perceptions list and the possible actions to take.

The perception list translated, not only the physical elements around the agent, what the agent could "see", but also internal sensations, what the agent could "feel". The perceptions related with internal sensations could only have one of three values, low, medium or high. This measure was thought to scale an agent's need. If the value was low the agent does not need it. Otherwise, the value was medium and the agent had a neutral needing. When, the value was high and the agent really needed it.

When the agent was a midge, the perception list had a range of 12. Next, a list with the respective description of the possible percepts that a midge could have is presented.

1. Tree count: Returns how many trees are within the agent's radius.
2. Tree M_x: Returns the x medium point of the tree, or trees, within the agent's radius.
3. Tree M_y: Returns the y medium point of the tree, or trees, within the agent's radius.
4. Mites count: Returns how many mites are within the agent's radius.
5. Mites M_x: Returns the x medium point of the mite, or mites, within the agent's radius.
6. Mites M_y: Returns the y medium point of the mite, or mites, within the agent's radius.
7. Midges count: Returns how many midges are within the agent's radius.
8. Midges M_x: Returns the x medium point of the midge, or midges, within the agent's radius.
9. Midges M_y: Returns the y medium point of the midge, or midges, within the agent's radius.
10. Energy sensation: Returns the agent's need of getting energy.
11. Tree Sensation: Returns the agent's need of being around a tree.
12. Sexual sensation: Returns the agent's need of breeding.

Although the perception list length was always the same, the perception values themselves might change with the midge stage or position. As mentioned before, the midge agent had four stages: egg, larval, pupal and adult, at each stage the perceptions followed the stage's characteristics.

Only the adult stage was able to reproduce, therefore, all stages except for the adult had the sexual sensation marked as low, for the adult stage the sexual sensation was always marked as high.

While the agent was at egg and pupal stages it did not move or it did not needed to be concerned about its energy, the energy sensation was marked as being low during these stages. On the other hand, when the stage was adult or larval, the energy sensation was constantly changing at each environment iteration. When the current energy was lower than half of the initial energy, the energy sensation was marked as high. Otherwise, was marked as low.

Also, the egg, larval and pupal stages were always located around trees consequently, the tree sensation perception for these stages was always high. The tree sensation on adult stage changed with the day stage, when the day stage was day or night the midges would be near the resting place known as the trees therefore, the tree sensation was marked as high. Otherwise, it was marked as low.

Table 4.3 holds the midge sensations that remain unchanged during an entire stage.

Midge Stage	**Percept**	**Value**
Adult	Sexual sensation	High
Larval	Sexual sensation	Low
Pupal	Sexual sensation	Low
Egg	Sexual sensation	Low
Egg	Energy sensation	Low
Pupal	Energy sensation	Low
Egg	Tree sensation	High
Larval	Tree sensation	High
Pupal	Tree sensation	High

Table 4.3: Midge fixed perception on the respective stage.

Last but not least, when the agent was a mite, the perception list had 11 percepts.

1. Mites count: Returns how many others mite agents are within the agent's radius.
2. Mites M_x: Returns the x medium point of the mite, or mites, within the agent's radius.
3. Mites M_y: Returns the y medium point of the mite, or mites, within the agent's radius.
4. Midges count: Returns how many midges are within the agent's radius.
5. Midges M_x: Returns the x medium point of the midge, or midges, within the agent's radius.
6. Midges M_y: Returns the y medium point of the midge, or midges, within the agent's radius.
7. Trees count: Returns how many trees are within the agent's radius.
8. Midges M_x: Returns the x medium point of the midge, or midges, within the agent's radius.

9. Midges M_y: Returns the y medium point of the midge, or midges, within the agent's radius.
10. Energy sensation: Returns the agent's need of getting energy.
11. Sexual sensation: Returns the agent's need of breeding.

As well as the midge agent, the mite perception list was updated at each environment iteration. When the stage was egg, the energy and sexual sensation were marked as low because this stage did not eat or reproduce. When the stage was larval, protonymph, deutonymph, or adult, the energy sensation was marked as high if the current energy was lower than half of the initial energy. If not, was marked as low.

The deutonymph and adult stages were able to reproduce, the sexual sensations in these stages were marked as high. On the other hand, at larval and protonymph stages the sexual sensations were marked as low, because these stages did not reproduce.

Mite Stage	Percept	Value
Deutonymph	Sexual sensation	High
Adult	Sexual sensation	High
Egg	Sexual sensation	Low
Larval	Sexual sensation	Low
Protonymph	Sexual sensation	Low
Egg	Energy Sensation	Low

Table 4.4: Mite fixed perception on the respective stage.

Instead of five stages, in this simulation a mite agent had only three stages, which were: the egg, the larval-protonymph and the deutonymph-adult. The stages duration in days are described in table 4.4. Table 4.5 describes the mite stages duration in the system.

Mite Stage	Duration (days)
Egg	3
Larval-Protonymph	3
Deutonymph-Adult	12

Table 4.5: Mite stages and respective duration in days.

4.3.3.2 Agent Behavior

The agent's behavior function connected perceptions to actuators, each mobile agent in the ecosystem needed to have a behavior function that translated what they could or not do.

For each stage, a midge or a mite had a different behavior function which contained the midge rules and characteristics that applied on the respective stage. The agent program chosen to use in this simulation was a simple reflex agent. This choice was done based on simplicity, the simple reflex agent program is a simple model and also fits the midge and mite behaviors very well when using the data collected in literature.

The midges possible actions were: to approach a position, to flee from a position, to stop at a position, to feed on a tree, randomly walk on the environment and to mate with another midge if a certain reproduction criteria was met. The mites were also able to perform all the previous actions except from: to flee from a position and to stop at a position.

An agent turned its alive variable to false if its energy was lower or equal to zero or its cycle ends. There were two exceptions to this rule. When a midge was at egg or pupal stage if it was caught by a mite its energy was set to zero and the midge immediately died. The midge larval stage lost 3.0 energy units, 2.0 energy points for meeting with a predator and 1.0 energy units for activating the flee function. Unless the larval energy sensation was high, the larval stage randomly walks in the environment but it cannot go further in distance than three times the tree radius.

Unfortunately the energy that an mobile agent wasted on performing its possible actions or the energy wasted on meeting a predator was data which could not be found. Therefore, in order to find a reasonable energy value that the agent could gain or lose, several simulations with the agents perception and behavior functions working properly were ran. Table 4.6 describes the energy cost or gain of each possible action that a midge can perform, chosen from the simulations results. This results were set as default energy parameters for the midges and mites in the system.

Midge Action	**Energy Cost**
Approach	−0.25
Flee	−1.0
Mite in Range	−2.0
Random Walk	−0.25
Stop	0
Feed	+250
Mate	−0.25
Successful reproduction	+200

Table 4.6: Midges energies cost or gain based on actions.

This choice was made by trial and error in order to get a simulation which imitated well enough the real cocoa plantation ecosystem.

An agent should not lose much energy when performing basic necessities like being alive or being able to move. Also, when performing basic actions like approach, random walk, stop and mate an agent should not lose much energy. Otherwise, the results observed were that the agents would die in few days before achieving the adult stage and being able to reproduce. To find a mite or to flee from a mite were considered bad for the midges, which means, they lose energy units when such happens. The results shown that midges could lose more energy units dealing with those previous events than by performing the basic ones. Mites did not have any predator therefore they did not lose energy with such kind of events.

When an agent needed to get energy, it would feed, such action increased the agent's energy. The simulations results shown that if the agent did not get a large amount of energy it was always closer to the trees not having enough energy explore the environment. When an agent performed the action to mate it was considered good for the agent because allowed the continuity of the species. If the reproduction was successful the agent energy was increased.

The female agents always laid the eggs near the resting places and reproduction happen all around the environment. This rule is a part of the mate functions that are described in detail at section 4.3.4.

The following rules apply when a midge was in adult stage and its energy sensation was different than high:

- Because the sexual sensation is always high, the midge always tries to mate with another midge.
- After, if the offspring list is not empty the female goes to the nearest tree and lay its eggs while the male randomly walks around the environment.
- For both genders after a successful reproduction the sexual sensation changes to low and the midge increases its energy variable by 200 units. If the offspring list is empty regarding the gender the midges randomly walk around the environment.
- When the tree sensation is high the midges can not go further than five times the radius tree of the closest tree found. When it is low the midge can randomly move around the entire environment.
- In every other circumstance the midge randomly walks on the environment.

On the contrary, when the midge energy was high, if the count trees percept was zero, there were no trees around the agent. Therefore the midge used its approach function in order to go to the closest tree. Otherwise, if the count trees percept was different than zero the midge activated its feeds function feeding on the closest tree and increased its energy variable by 250 units.

Similarly to the midges species, the mites also followed certain rules by performing certain actions given a list of perceptions as input.

On larval stage, when the energy sensation was different than high the mite randomly walk around the entire environment. But when the energy sensation was high, if the mite had no midge in its surroundings it used its approach function in order get close to the closest midge which stage was not adult. Otherwise, the mite fed on the closest midge which stage was not adult and increased its energy variable by 300 energy units.

Table 4.7 describes the cost or gain of each possible action that a mite can perform.

Mite Action	**Energy Cost**
Approach	−0.25
Random Walk	−0.25
Feed	+300
Mate	−0.25
Successful reproduction	+200

Table 4.7: Mites energies cost or gain based on actions.

The following rules apply when a mite was on adult stage:

- When the mite's energy is different than high, because the sexual sensation is high, a mite always tries to mate with another mite.
- For both genders after a successful reproduction the sexual sensation changes to low and the mite increases its energy variable by 200 units. When the sexual sensation is low the mite randomly walks in the environment.
- When the mite's energy perception is high, if the mite has no midge within its surrounding it activates the approach function in order to get closer to the closest midge which stage is not adult. When the mite has midges in its surroundings it feeds on the closest one and increases its energy by 300.
- In every other circumstance the mite randomly walks around the environment.

To conclude the section, table 4.8 gives an overview of the complete environment task for this simulation.

Agent Type	**Environment**	**Actuators**	**Sensors**
Tree	Cocoa Plantation	-	-
Day	Cocoa Plantation	-	-
Year	Cocoa Plantation	-	-
Midge	Cocoa Plantation	F, M, A, FL, S	Internal, External
Mite	Cocoa plantation	F, M, A	Internal, External

Table 4.8: Environment task. Where, F stands for Feed, A for Approach, M for Mate, FL for flee and S for Stop.

4.3.4 Reproduction

Each midge agent initialized within the system had a chromosome which was represented by a list. This list held information about the genes that made the midge's survival possible in the environment. A chromosome consisted of five genes: the weight, the radius sensor, the skill, the adaptation and the sex. The first four genes could be one of three alleles: low, medium or high. The fifth gene could be one of two alleles: male or female. In total there were 162 possible combinations, where each represented a different chromosome.

Within the system these alleles were represented by a categorical variable. At the first four genes low was 0, medium was 1 and 2 was high. In the fifth gene, the alleles were codified as 0 for male and 1 for female, as follows table 4.9.

Alleles	**Codification**
Low	0
Medium	1
High	2
Male	0
Female	1

Table 4.9: Alleles and respective codification.

The first generation of chromosomes was randomly initialized. Each gene was defined by randomly choosing an allele among the alleles types mentioned before. From the second generation, the new chromosomes, or new individuals, were generated by using a genetic algorithm that allowed the midges to change genetic information between them and to create offspring.

In order to observe and create species evolution in the environment, the alleles need to have an effect on the midge's physical condition. Moreover, the weight was linked to the midge initial energy variable, the sensor range was linked to the midge radius, the skill gene was linked to the maximum velocity that a midge can achieve, the adapt gene was linked to the temperature on which the midge was able to survive and the sex gene was linked to the midge gender which was used for breeding effects.

The higher the weight, the sensor range, the skill and the adapt genes, the higher the midge initial energy, sensor range, maximum velocity and temperature where it was able to survive.

$$Chromosome = (Weight, SensorRange, Skill, Adapt, Sex) \tag{4.1}$$

An insect's weight might vary between 70 and 100 *grams*, where an insect with 100 grams has 409.78 *kcal*, which represents its energy threshold. When the weight allele was low, a random number between 70 and 78 was generated to be the midge's weight value. If the weight allele was medium a random number between 79 and 89 was generated. Otherwise, the weight allele was high and a random number between 90 and 100 was generated.

Once the weight variable value was known the midge's initial energy could be calculated. The following equation, has the formula to calculate a midge's initial energy given its weight.

$$InitialEnergy = \frac{409.78 MidgeAgentWeight}{100} \tag{4.2}$$

The sensor range gene defined how far a midge could detect other agents from its radius in the environment. When its allele was high, the midge sensor range variable was five times the midge radius variable. When its allele was medium and the sensor range was three times the midge radius. When its allele was low the sensor range variable was two times the midge radius.

The skill gene defines how fast a midge could move by limiting its maximum velocity in the environment. If the allele was low, a midge's maximum velocity was one. If the allele was medium where the maximum velocity was three. Otherwise, the allele was high and the maximum velocity was five.

Because an insect's well-being depended directly on the environment's temperature, the adapt gene defined the minimum and maximum temperature where the midge was able to survive. If the allele was different than high the midge was able to survive at $20 - 25$°C. If the allele was high and the midge was able to survive at $20 - 26.5$°C.

The sex gene simply represented the midge gender which also was female or male. For instance, the chromosome in figure 4.6, codified according with these system rules, represents a midge with low weight, high sensor range, medium skill, low adaptation and it is a male.

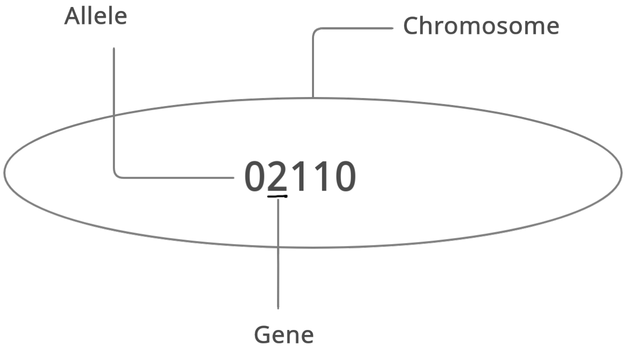

Figure 4.6: Example of a midge chromosome.

On the other hand, the mite species were genetically identical, therefore they had no chromosomes associated with.

4.3.4.1 Genetic algorithm

The user has the chance to choose the population that is to be initialized before the simulation begins. The midges number chosen by the user generates the initial midge population which initially feeds the genetic algorithm.

Without loss of generality, a species goal was to survive in the environment and to reproduce in order to continue the species and this genetic algorithm was no different.

In order to do so, a method was built to evaluate not only how good genetically an individual was but also its performance in the environment. For this to be possible, the midge chromosome and the energy perception, because it described a history of the midge performance in the environment. Within the system this method was represented by a function called fitness function.

The fitness function took as input a midge chromosome and an energy perception value and it operated by calculating a fitness value in two separate steps. First, the function evaluated the midge chromosome by going through all the genes and checking the respective allele code in order to count how many high alleles there were in the chromosome. The number of high alleles was saved into a variable called fitness genetic. Second, the function evaluated at the midge's energy perception value and gives a certain score according to the previous value. This score was saved into a variable called fitness energy. The fitness energy value followed the following rules: if the energy sensation was low the fitness energy was 2.0 units, if the energy sensation was medium the fitness energy was 1.0 unit and if the energy sensation was high the fitness energy was 0.0 units.

The function returned the fitness value that was a linear combination of the fitness genetic and fitness energy. The midge with the highest fitness value was the fittest.

$$Fitness(Chromosome, EnergyPerception) = 0.3FitnessGenetic + 0.7FitnessEnergy \tag{4.3}$$

When it came to creating offspring, three genetic operators were applied: parent selection, crossover and mutation, by this exact order.

The environment played a major role in parent selection because given a midge it returned a dictionary with the potential parents gene and energy sensation information at that environment iteration.

This dictionary was fed to the genetic algorithm that calculates the respective fitness functions of each potential parent and returned a new dictionary with the parents candidate ids as keys and the result of the fitness function as values. Then, the dictionary was sorted and at the first position was the fittest parent.

The case that there were more than one parent with the higher fitness value, the midge reproduced with the first one that appeared on the dictionary ordered by fitness values.

If the length of the parents dictionary was one, the midge approached the other midge without any competition or parents selection from the genetic algorithm. When the dictionary length was greater that two, the parent selection from the genetic algorithm was activated and the midge approached the fittest parent given by the previous parent dictionary.

Once the parents were selected, a crossover method was performed between the two midges in order to create offspring.

This method chose six random points in the chromosome, named crossover points, and exchanged the alleles in those positions between the chromosomes, the new swapped chromosomes were the resulting offspring.

Next, a random number between zero and one was generated. This random number was called mutation rate. If the mutation rate was greater that 0.001, a mutation point method was applied to the offspring chromosome. The mutation point method went to the crossover point and randomly change the allele again.

Besides being responsible for generating the offspring, the mate function also guaranteed that the reproduction occurred between different genders and at the correct midge and mite stages. This function was also responsible for placing the offspring within the environment. If the midge it was a female it would go to the closest tree in order to lay the eggs at a random position near that closest tree and increased its energy variable by 200 energy units.

Although the mite species were genetically identical they also have a mate function which allowed the species to reproduce. When an agent was a mite the mate function operated in different ways according with its gender.

If the gender was female the goal was to lay eggs near the closest tree. Then, if the mite could detect trees within its sensor range it approached the position of the closest tree. When a female could detect trees it would lay male eggs on a random position near the closest tree and increased its energy variable by 200 energy units. A female mite could lay four eggs in its life span.

When the gender was male the mite approached the closest egg around it and swapped the egg gender to female by fertilizing it. Increased its energy variable by 200 energy units. A male could fertilize two eggs in its life span.

When both midges and mite regardless of the gender reached their maximum number of eggs to lay or fertilize they would randomly walk in the environment.

CHAPTER

5

DISCUSSION

This chapter aims to exhibit and to describe three different results associated with the mobile agents if the environment's temperature was modified. This discussion followed three steps: a baseline simulation definition, an increase of 1.5°C and an increase of 3.0°C to the baseline temperature simulation.

The baseline simulation was used as a base to comparison of experiments done with the environment's temperature parameters. The baseline was the system built in the previous chapter where the environment had a temperature between 20 – 25°C when the year season was wet and a temperature between 20 – 28°C when the year season was dry.

Depending on genetic factors, a midge usually survived at temperatures between 20 – 25°C, but in a few tests it survived at temperatures between 20 – 26.5°C. The mites when compared with the midges were able to survive at a larger temperature interval. This species was able to survive at temperatures between 15 – 28°C.

Table 5.1 holds the information on the parameters used within the baseline simulation.

Parameters	**Value**
Environment Temperature (°C)	20 – 25 or 20 – 28
Midge Temperature (°C)	20 – 25 or 20 – 26.5
Mite Temperature (°C)	15 – 28
Trees Number	150
Midges Number	100
Mites Number	20

Table 5.1: Baseline simulation input parameters.

The baseline, the +1.5°C and the +3.0°C simulations were all initialized with 272 agents in the environment. Where:

- 150 agents were trees.
- 100 agents were midges where 50 of them were females and the other 50 were male.
- 20 agents were mites where 10 of it were females and 10 were males.

All three simulations ran within a year, 360 days, alongside with a maximum of 18 generations, a maximum of 20 generations that the midges and the mites supported, respectively, and an unlimited number of generations.

The baseline simulation revealed that both species were able to survive in the environment at random temperatures within the temperature intervals 20 – 25°C or 20 – 28°C. The temperature interval depended on the year stage respectively.

5.1 Environment Temperature Increase: +1.5°C

In this simulation the baseline environment temperature was increased by 1.5°C. Therefore, the new environment temperature was a random number between 21.5 – 26.5°C or between 21.5 – 29.5 °C.

This simulation reveled a small but relevant alteration within the midge population. Out off the 100 initial midges, the number of midges that were able to survive in the new environment temperatures when the simulation began were in average 89 midges. Within the temperature increase of 1.5°C the midge species had a decrease of 11% on its population.

Although the midge species had started the simulation with less population numbers than at the baseline simulation, the species was still able to overcome the initial population decrease and survived through it by holding enough generations which allowed to continue the species.

On the other hand, the mite population was not affected at all by the increase of 1.5°C on the environment temperature. The entire population was able to survive at these new environment temperatures.

Table 5.2 contains the parameters used within the temperature increase by 1.5°C simulation.

5.2 Environment temperature increase: +3.0°C

To represent an environment temperature increase by 3.0°C simulation it was set the environment temperature to a random number between 23.0 – 28.0°C or between 23.0 – 31.0°C. This simulation revealed a complete incapability for the midge species to live at such environment temperatures.

Parameters	**Value**
Environment Temperature (°C)	21.5 – 26.5 or 21.5 – 29.5
Midge Temperature (°C)	20 – 25 or 20 – 26.5
Mite Temperature (°C)	15 – 28
Trees Number	150
Midges Number	100
Mites Number	20

Table 5.2: Environment temperature increase by 1.5°C simulation parameters.

On average, out of the 100 initial midges, only 27 midges were able to survive at such environment temperatures. For the midge species this situation represents a decrease of 73% on the midges population.

In these circumstances, which represent an extreme climate change scenario, the midge species was not able to produce enough offspring in order to ensure the continuity of the species. Therefore, in this simulation the midges species went extinct.

The mite species was, again, unaffected by this change of temperature in the environment.

Table 5.3 has the parameters used within the temperature increase by 3.0°C simulation.

Parameters	**Value**
Environment Temperature (°C)	23.0 – 28.0 or 23.0 – 31.0
Midge Temperature (°C)	20 – 25 or 20 – 26.5
Mite Temperature (°C)	15 – 28
Trees Number	150
Midges Number	100
Mites Number	20

Table 5.3: Environment temperature increase by 3.0°C simulation parameters.

CHAPTER 6

CONCLUSION

For the last chapter of this dissertation a reflection on the discussion results was took based on the predictions and warnings that the scientific community has been giving for more than thirty years, previously explained in chapter 1.

From the discussion chapter it was possible to conclude that an increase of 1.5°C on the environment's temperature had no major change in the midge species. On the other hand, an increase of 3.0°C on the environment temperature had a huge impact on the midge's survival, which led to the species extinction.

Although the mite species remains unaffected by both temperature changes they feed on midges, with the midge's extinction at some time in the simulation the mites were also to be extinct due to the lack of food in the environment.

Last but not least, the midge species is the major pollinator of the cocoa trees in Ghana and with the midges extinction at some point in the simulation the trees would also face extinction or most probably would suffer a huge decrease in its numbers.

Besides affecting the lives of million of people that somehow are connected to the cocoa farms in Ghana, the damage to the cocoa sector could have a catastrophic impact on the global economy.

The results observed in this dissertation were not a surprise. They were, in fact, expected. It is known that insects are very sensitive to temperature. The cocoa plantation is not an exception for the consequences of global warming. This means that thousands of other species are similarly at risk.

This dissertation was focused on the effect on the species if one environment parameter, the environment temperature, was changed. The midges extinction was a possibility from the very beginning. Hundreds of other approaches and combinations can be tested in order to observe different and probably more accurate results, which is another research on its own.

6.1 Future Work

A future work idea for this system is to program the mobile agents in a way that allowed them to be intelligent agents. The mobile agents could be transformed into learning agents and experimented under the agent programs described in chapter 2. Having said that, it would be possible to simulate an ecosystem with agents on it with the ability to decide if an action was good to take or not.

Furthermore, if a proper brain could be added to the agent using a neural network it would be possible to build an agent with a reinforcement learning method which learns what is the best action to take at a certain moment while exploring the environment and following its rules and characteristics.

Another possible idea for the future research purposes that came up while writing the dissertation is to build a method which allows the year stage to limit the amount of flowers available in each tree. This method would reflect on mobile agents as a limit to the number of agents that can be around a tree agent at the same time. It would create an extra competition for food in the environment for the midge species.

BIBLIOGRAPHY

Attenborough, D. (2020). *A life on our planet: My witness statement and a vision for the future*. Random House.

Claus, G., Vanhove, W., Van Damme, P., & Smagghe, G. (2018). Challenges in cocoa pollination: The case of côte d'ivoire. *Pollination in plants, 39*.

Darwin, C. (2004). *On the origin of species, 1859*. Routledge.

Dobermann, D., Swift, J., & Field, L. (2017). Opportunities and hurdles of edible insects for food and feed. *Nutrition Bulletin*, *42*(4), 293–308.

Eiben, A. E., Smith, J. E. et al. (2003). *Introduction to evolutionary computing* (Vol. 53). Springer.

Frimpong, E. A., Gemmill-Herren, B., Gordon, I., & Kwapong, P. K. (2011). Dynamics of insect pollinators as influenced by cocoa production systems in ghana. *Journal of Pollination Ecology*, *5*.

Hutchins, A., Tamargo, A., Bailey, C., & Kim, Y. (2015). Assessment of climate change impacts on cocoa production and approaches to adaptation and mitigation: A contextual view of ghana and costa rica. *International Development Studies*, 1–22.

Kaufmann, T. (1975). Studies on the ecology and biology of a cocoa pollinator, forcipomyia squamipennis i. & m.(diptera, ceratopogonidae), in ghana. *Bulletin of Entomological Research*, *65*(2), 263–268.

Koren, I., Kaufman, Y. J., Washington, R., Todd, M. C., Rudich, Y., Martins, J. V., & Rosenfeld, D. (2006). The bodélé depression: A single spot in the sahara that provides most of the mineral dust to the amazon forest. *Environmental Research Letters*, *1*(1), 014005.

Langton, C. G. (1997). Artificial life: An overview.

Levy, S. (1992). *Artificial life: The quest for a new creation*. Random House Inc.

McCormack, J. (2007). Artificial ecosystems for creative discovery. *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, 301–307.

Ogunbode, C. A., Doran, R., & Böhm, G. (2020). Exposure to the ipcc special report on 1.5 c global warming is linked to perceived threat and increased concern about climate change. *Climatic Change*, *158*(3), 361–375.

Rothlauf, F. (2006). Representations for genetic and evolutionary algorithms. *Representations for genetic and evolutionary algorithms* (pp. 9–32). Springer.

Russell, S., & Norvig, P. (2002). Artificial intelligence: A modern approach.

Zhang, Z.-Q. (2003). *Mites of greenhouses: Identification, biology and control*. Cabi.