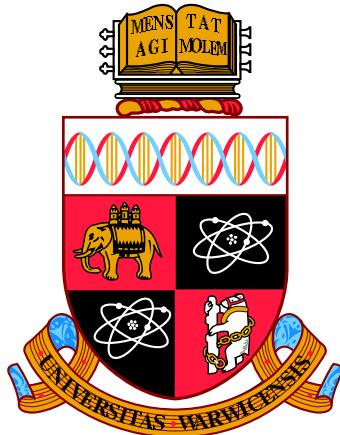


An interactive educational simulation of a marine ecosystem consisting of planktonic organisms and their evolution over time

Hassan Qadir

CS310 Computer Science Project Report



Supervisor: Dr Jianfeng Feng

Department of Computer Science
University of Warwick

2016-2017

Abstract

The marine ecosystem is characterised by many interesting natural phenomena, and is often viewed as a dynamic system comprising of evolving, interacting and self organising entities. The modelling and simulation of this system is usually confined to high academia, and is therefore inaccessible to students who wish to gain insight into the processes encapsulated by these systems. In an attempt to bridge the gap between accessibility and complexity, this project has created a discrete event simulation that models an abstract representation of the marine ecosystem. Seen through an interactive graphical application, the dynamics of this system are modelled through the interaction of various marine organisms and the environment. Moreover, these organism have genes which can mutate on an adaptive fitness landscape, enabling natural selection. The developed system design is based upon substantial research, and is carefully validated and verified using scientific methods. The result is a simulation that shows real promise both as a technical and educational tool.

Acknowledgements

The success of this project could not have occurred without the immense help and gratitude of a number of individuals, of whom I would like to acknowledge. First and foremost, my immense thanks must be given to Dr Jianfeng Feng, who provided excellent supervision and continual advice and support throughout the development of the project. Many thanks must also be given to Dr Nasir Rajpoot, whose marking, feedback and insight has added much needed rigour to the project.

I would also like to thank the researchers who helped provide advice and insight into concepts which were, at the time, very new and alien to me. Included in this is Dr Sara Kalvala, who provided face validation of the conceptual model and provided much insight into the potential of the project. Additionally, I would like to thank my testers, who provided significant help in the validation and verification of the simulation. These people include Kameron, Kulsoom, Amarjot, Gurkirat, Jeevan and Nabiha.

Contents

1	Introduction	9
1.1	Motivation and Background	9
1.2	Project Aim	11
1.3	Project Stakeholders	11
1.4	Report Overview	12
2	Research	14
2.1	The Marine Ecosystem	14
2.1.1	Abiotic variables	17
2.1.2	Biotic Variables - Producers	18
2.1.3	Biotic Variables - Consumers	18
2.2	Modelling the Marine Ecosystem	19
2.3	Modelling Variation and Fitness	22
3	System Requirements	26
3.1	Functional Requirements	26
3.1.1	Model Requirements	26
3.1.2	Software Requirements	29

3.2 Non-functional Requirements	30
4 Design	31
4.1 The Conceptual Model	31
4.2 The Ecosystem	33
4.3 Genetics and Fitness	35
4.3.1 Initial Experimentation	35
4.3.2 The Logistic Model	37
4.4 The Simulation Cycle	41
4.4.1 Selection	41
4.4.2 Living	42
4.4.3 Predation	43
4.4.4 Mating	43
4.5 The Dynamics of the System	45
4.6 System Architecture	48
4.7 Environment and User Interface Design	50
4.7.1 World Design	50
4.7.2 Analytics	51
4.7.3 Navigation and Controls	51
5 Implementation	53
5.1 Technologies Used	53
5.2 Structural Design Implementation	54
5.2.1 Biome and Zone	54

5.2.2	Chromosomes and Genes	57
5.2.3	Organism	59
5.3	The Ecosystem Controller	60
5.3.1	Initial Population	61
5.3.2	Running the Simulation	61
5.3.3	Analytics and User Input	62
6	Validation, Verification and Testing	71
6.0.1	Conceptual Model Validation	71
6.1	Verification and Testing	73
6.1.1	Unit Testing	73
6.1.2	Component Testing	74
6.1.3	Integration Testing	74
6.1.4	System Testing	82
6.1.5	User Acceptance Testing	82
6.2	Operational Validation, Results and Tuning	83
6.2.1	Photic Zone	84
6.2.2	Aphotic and Abyssal Zones	86
6.2.3	User Input	87
7	Legal, Social, Ethical and Professional Issues	89
7.1	Legal	89
7.2	Social and Ethical	89
7.3	Professional	90

8 Project Management	91
8.1 Methodology	91
8.1.1 Design	91
8.1.2 Software Development	92
8.2 Project Timetable	95
8.3 Management Tools and Techniques	95
8.4 Risk Management	96
9 Evaluation	98
9.1 Functional Evaluation	98
9.2 Non-functional Evaluation	98
9.3 Project Management Evaluation	98
9.4 Reflection and Limitations	99
10 Conclusion	105
10.1 Summary	105
10.2 Future Work	106
APPENDICES	110
A User Guide	111
B Keywords	112

List of Figures

2.1	Divisions of the marine ecosystem [32, chap. 1]	16
2.2	Distribution of O ₂ and CO ₂ with depth [concentration of dissolved gases in millilitres per litre (ml/l)] [32, chap. 4]	17
2.3	Common marine phytoplankton with unique shape and configuration	18
2.4	Abstracted trophic pyramid of the marine ecosystem	19
2.5	An example of a Biogeochemical model by [19], the NPZD model	22
2.6	A three-dimensional Wrightian fitness landscape [34]	24
4.1	The conceptual model of the system	32
4.2	A model of the physical space of the ecosystem	34
4.3	Plots for two alleles of Chromosome Pp	39
4.4	Abiotic constrain mapping	40
4.5	The simulation cycle representing one time step	41
4.6	A Punnett square showing a crossover of two chromosomes	44
4.7	The change in state density from time t to $t + 1$	45
4.8	A closed cycle within the model	46
4.9	Harmonic motion of a predator-prey relationship	47
4.10	Simulation system architecture	49

4.11 User Interface Design	52
5.1 UML class hierarchy for Ecosystem, Biome and Zone	55
5.3 UML class hierarchy for Chromosome and Gene	57
5.6 Code for the <i>main()</i> method	60
5.2 Code for the selection function	64
5.4 Code for the fitness function	65
5.5 UML class hierarchy for Organisms	66
5.7 Code for the <i>settings()</i> method	67
5.8 Instantiation of simulation objects	68
5.9 Initial Organism instantiation for the Aphotic Zone	69
5.10 Creation of the Biome	69
5.11 Function for displaying analytics	70
6.1 Simplified version of the model development process [40]	72
6.2 Data collected from the Photic Zone	85
6.3 Phytoplankton and Carbon Dioxide in the Aphotic Zone	86
6.4 Carbon Dioxide and Oxygen in the Aphotic Zone	87
6.5 Effects of user increased Carbon Dioxide	88
8.1 The incremental build cycle	93
8.2 The revised Gantt chart	94

List of Tables

4.1	Genetic algorithm experiment	37
6.1	Unit tests on the Gene class	75
6.2	Unit tests on the Organism class	78
6.3	Component tests on the Zone class	80
6.4	Integration tests on the System Components	81
9.1	Functional Evaluation	103
9.2	Non-functional Evaluation	104

Chapter 1

Introduction

Scientific modelling and simulation have always proven useful as a means of illustrating concepts within life science for educational purposes [10, 42, 46, 22]. Further, as computational resources have become more accessible, the opportunity to use simulation models as a way of analysing processes within biology and ecology increases. Despite this, very few accessible simulations of biological systems exist commercially that adequately bridge the gap between usability and mathematical rigour that is amenable to computational analysis [41]. This report documents the development of an ecological simulation that aims to fill this gap by focusing on scientific detail whilst maintaining usability and interactivity, with the aim that it could be conceivably used as a teaching tool.

This chapter explores the motivation and background of the project, and introduces the project aims and objectives. Finally, contemplation of the various stakeholders is given, followed by an outline to the document structure.

1.1 Motivation and Background

The considerable challenge posed by modelling processes in life science has historically hindered its advancement [1]. Despite this, there is a growing focus towards developing such systems as our readily available information processing power increases. Moreover, a particular modelling focus is often placed upon the systems approach. Typically, this is characterised by a desire to describe the states and relationships of certain phenomena quantitatively as an attempt to better understand their underlying mechanisms and

laws. Further, the semantic closeness that arises from a conceptual model to a computer model means that data and information gained from modelling and simulation can easily be translated into findings that are interesting and informative to a life scientist [15]. Despite this fact, it often remains to be seen whether such techniques can generate novel insight that cannot be uncovered by purer methods [41]. Indeed, this very challenge encourages the goal of this project; to conceptualise and develop a model and simulation of a system or process within life science. Such a system would attempt to provide insight into a chosen process whilst attempting to offer a visually engaging and interactive illustration of the process at hand. Moreover, this endeavour presents a computational challenge [20], and the application of computer science as a means of visualising phenomena within other fields is an exciting application.

Genetics and ecological dynamics bring together elements from biology and ecology and are principle in the understanding of systems within the natural world. As such, they are core within life sciences at a higher educational level - a good target area for simulation. Before formulating any original ideas, a consideration should first be given to existing implementations that aim to tackle this problem.

Commercially, there are many implementations of the simulation of natural selection. One such example is the 'Phet' [10] simulation of natural selection, which provides a level of interactivity to the user. However, it fails to properly model and simulate the dynamics of an ecosystem that contain evolving organisms, whilst also lacking the depth of the science. Further, it falls short of achieving a sufficiently high level of visual fidelity. Similarly, whilst 'Evol Sim' [22] manages to somewhat simulate the dynamics arising from evolving organisms, it fails to model any real world environments, ecosystems or species and as such is somewhat incomplete for academia. The 'Species' [42] game somewhat makes up for this by placing importance on scientific accuracy; it models an ecosystem along with the non-living (abiotic) characteristics of it. However, it suffers from aiming to simulate far too much complexity without adequate analytics. Finally, 'EvolveJS' [46] provides a more dynamic approach to simulation, but unfortunately fails to actually model anything in the real world.

Despite these simulations fulfilling the illustrative objective of scientific simulation, they still lack the crucial depth required of them to feasibly be used as tools which may enable novel insight and provide education. Furthermore, they lack a key integration with the computational approach to modelling biological and ecological phenomena, which brings many benefits over the traditional mathematical technique of scientific modelling [15]. Approaching the challenge of attempting to understand the interactions and evolution of life is tremendously exciting and is a key motivation for this project. Moreover, fulfilling

the objective of modelling the dynamics of the natural world containing interacting evolving organism requires an in depth study of ecology and biology, with a particular focus on the areas pertaining to the topics listed above. Given this, an ecosystem should be identified and modelled with appropriate abstraction along with the variables, both living and non living, that characterise it. An understanding of system dynamics will allow the derivation of a dynamic evolving model of an ecosystem. The majority of this research is outlined within the research section of this document, which precedes the Design and Implementation.

1.2 Project Aim

There are many interesting natural phenomena in the marine ecosystem that may benefit from being abstracted, modelled and simulated, with a goal being to provide insight into their function. Further, creating such a system could be used as an educational tool for those studying concepts within these fields. A marine ecological system will be modelled and implemented as a discrete event simulation through a graphical, interactive software application. This system will also model the natural selection of organisms as they adapt to a dynamic environment. In building this system, a conceptual model is devised based on substantial research and background both on the nature of the real ecosystem and the techniques employed to model it.

1.3 Project Stakeholders

The project leader, Hassan Qadir, along with the supervisor Dr Jianfeng Feng and the co supervisor Dr Nasir Rajpoot are the primary stakeholders in this project. As such, they will support the development of a software solution based on the research conducted, which was also influenced by these essential stakeholders. The project leader in question holds considerable influence in not only the product but the future of the project in general. As a former student and avid reader of biology and ecosystem dynamics, the theme of the project holds considerable interest to him. As a result, he will continue to invest into improving the accuracy and validity of the model, as well as the quality of the software implementation upon project completion.

As a commercial academic product, any users of the system will be declared as secondary stakeholders. Dr Sara Kalvala was consulted to provide validation for the system, and

is therefore also a key stakeholder.

1.4 Report Overview

Research

This section outlines the research that was performed preliminary to the derivation of any models of the system. Initially, research into the marine ecosystem and its characteristics is given, including a study of the organisms that reside within it. The morphology and the evolutionary history of such organisms are also researched. Finally, research into the dynamics and the systems approach to the modelling the marine ecosystem is shown which will inform the process of deriving a model.

System Requirements

This chapter outlines the specification for the requirements for both the conceptual model and the system that implements it.

Design

These chapter document the design of conceptual models based on the research conducted, followed by a presentation of the design of the system architecture that encapsulated these models.

Implementation

A high level description of the methods used to implement the design presented earlier in software is presented in this chapter.

Validation, Verification and Testing

In keeping with the methodology of simulation modelling, the conceptual model is validated scientifically. Following this, the software implementation is verified through extensive software testing. Finally, the operation of the simulation is validated against expected outcomes.

Legal, Social, Ethical and Professional Issues

The various potential issues surrounding the development of the project are contemplated in the section, along with suggestions on how to mitigate such issues.

Project Management

Details regarding how the project was managed from start to finished is outlined here, with consideration given to changes made to the project timetable as time progressed.

Evaluation

The functional and non-functional requirements outlined in the System Requirements chapter is individually evaluated for success.

Chapter 2

Research

This chapter documents the background research that was conducted in order to inform the model development. Initially, the aim was to understand the background required to begin the consideration of developing a simulation within biology and ecology, along with a justification as to why certain elements were chosen as a point of focus. Thereafter, a review is given of various methods of implementing simulation within the context of the initial research. This provided inspiration for the design that is outlined in the sections following.

2.1 The Marine Ecosystem

Before documenting the features of the marine ecosystem, including why it was chosen to model, it is worth understanding the fundamentals of an ecosystem. An ecosystem can be considered as consisting “*of all the organisms and the abiotic pools (quantities) with which they interact*”, where pools are regarded as abiotic (non-living) or biotic (living) components of the ecosystem. Moreover, we understand ecosystem processes to be “*the transfer of energy and materials from one pool to another*” [14, chap. 1]. Energy will enter an ecosystem when light energy is captured by photosynthetic organisms and directed into organic matter, driven by the light reduction of carbon dioxide. This organic matter (and hence energy) will travel within an ecosystem through the trophic levels of a food chain during predation, with energy being lost as heat when consumer organisms respire with the use of oxygen. Organisms, upon death, may be decomposed as detritus by decomposing organisms, and so we observe another transfer of energy through organic matter. Crucially, the flow of energy implies the flow of biotic matter

including “*the absorption of minerals by plants, the death of plants and animals, the decomposition of dead organic matter by soil microbes, the consumption of plants by herbivores, and the consumption of herbivores by predators*” [14, chap. 1]. These flows are sensitive to environmental variables and hence play a role in regulating the dynamics of interacting organisms and communities. Indeed, the derivation of a model representing such a system will inform the development of a simulation. Beforehand, however, an actual ecosystem must be identified and studied.

Given that the fundamentals of an ecosystem has now been loosely defined, the task then became to identify one that was simple enough to model. An aquatic system seemed appropriate given that it is simple in comparison to other well defined ecosystems on earth. For instance, a land based ecosystem carries with it the complexity of land organisms and large fauna; these creatures are vastly complex and are therefore out of the scope of the project. Aquatic ecosystems on the other hand benefit from containing simple organisms that occupy all levels of its trophic pyramid. As such, the task was then to well understand the ecology of an aquatic system and thus identify an ecosystem within it to model.

The aquatic ecosystem may be divided into freshwater and marine ecosystems. The former is considered in the study of limnology; the study of the biological, chemical, and physical features of lakes and other bodies of fresh water. The latter, on the other hand, refers to oceanic ecosystems and hence oceanography [26]. Within the scope of this project, it does not seem important which of the two the simulation is based on, given that the key feature of the system will likely be common in both. That being said, the evolutionary patterns and the variety of species found within a marine ecosystem is rather vast. The marine environment is also rather interesting, and is distinguished from freshwater ecosystems by the presence of dissolved compounds in the water [3]. This feature might be an interesting inclusion in the simulation, and so the research was focused on the marine ecosystem.

The marine ecosystem can be divided into benthic (substratum based) and pelagic (aquatic) zones. The benthic compartment of the ocean starts from the shoreline and extends up to the hadal zone. Furthermore, the entire water mass above the ocean floor (benthic substratum) with all the diverse types of organisms is referred to as the pelagic zone [32, chap. 1]. The benthic zone represents a wide diversity of ecological features, but given that the simulation should aim to model an aquatic or oceanic ecosystem, the simulation will be based on the pelagic (aquatic) zone of the marine ecosystem.

The pelagic zone can also be differentiated into neritic and oceanic zones [32, chap. 1],

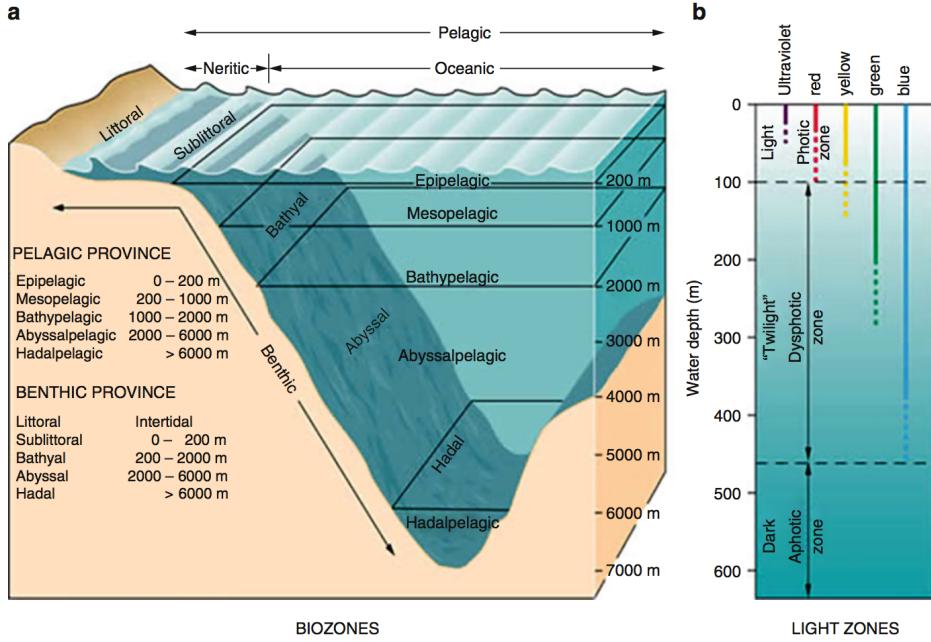


Figure 2.1: Divisions of the marine ecosystem [32, chap. 1]

with the neritic zone representing the shallow extension of ocean from the intertidal zone to the continental shelf and has an average depth of 200m. With the oceanic zone being far deeper, it has more interesting ecological characteristics and likely a greater level of biodiversity; the oceanic compartment will be focused on.

The oceanic compartment can be divided into the photic and aphotic zones. The photic zone is restricted up to the layer where light is available and usually ranges up to 200m from the surface. Oceanic producers usually occupy this space, such as algae and phytoplankton, who serve as the nutrient base of the consumers. By contrast, the aphotic zone lies below this 200m depth and receives very low levels of illumination; the water absorbs more than 95% of the sunlight. Photosynthesis is therefore rarely possible [32, chap. 1]. The exponential decrease in light intensity due to depth can be attributed to absorption and refraction, a phenomenon known as vertical light attenuation which can be mathematically expressed by Beer Lamberts law [36, chap. 1]

$$E_d(z) = E_d(0) \cdot e^{-kz}$$

where $E_d(0)$ and $E_d(z)$ are light intensity at surface and at depth z respectively. This relationship will likely be of importance in the modelling of the ecosystem and will be discussed later on.

2.1.1 Abiotic variables

The most abundant gases in the atmosphere and the oceans are Nitrogen (N_2), Oxygen (O_2) and Carbon Dioxide (CO_2) [32, chap. 4]. Moreover, in photosynthesis, plants (and all producers, including Phytoplankton) use carbon dioxide to form organic molecules and Oxygen as a by product. However, the light needed for this is usually confined to the upper 100m of the ocean, where the producers of the marine ecosystem usually live and survive. Oxygen is therefore typically produced in the surface water and carbon dioxide is usually consumed here.

By contrast, respiration breaks down organic substances, providing energy. The process requires Oxygen and produces Carbon Dioxide. We also observe that decomposition, the process in which non-living organic material is broken down, also requires Oxygen and releases Carbon Dioxide.

Oxygen usually enters the oceans at the surface due to exchange with the atmosphere or as a waste product of photosynthesis. Carbon Dioxide may also enter through similar exchange, but is usually available at all depths due to respiratory and decomposition activity in organisms. We also observe that an oxygen minimum occurs at around 800m depth due to the absence of the photosynthetic process and the presence of the respiratory events of organisms that live in this aphotic zone [32, chap. 4].

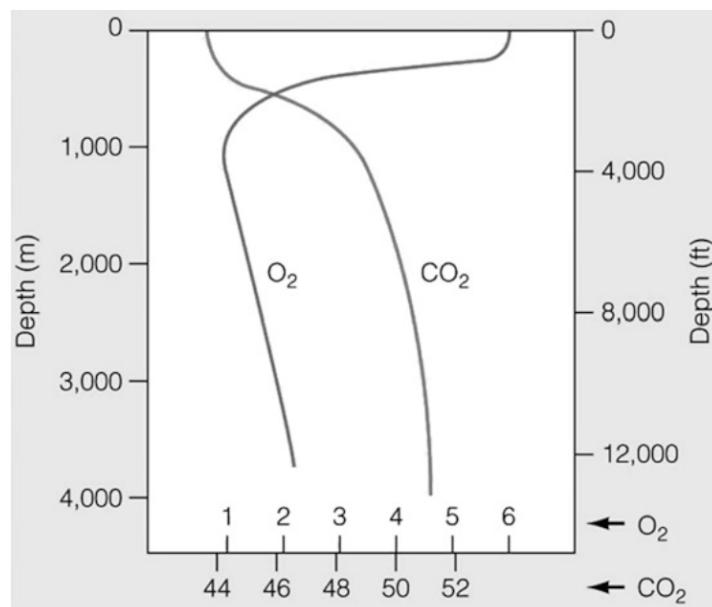


Figure 2.2: Distribution of O₂ and CO₂ with depth [concentration of dissolved gases in millilitres per litre (ml/l)] [32, chap. 4]

Given this information, abiotic factors such as Carbon Dioxide and Oxygen levels should

be modelled in the simulation, with a clear separation between photosynthetic and respiratory organisms. Differing populations of these organisms, as well as the abundance of such factors will likely drive the evolution of species.

2.1.2 Biotic Variables - Producers

Most of the primary production in the marine ecosystems is carried out by phytoplankton [32, chap. 5]. Phytoplankton are free-floating, photosynthetic, aquatic microorganisms, which move from one place to another, either actively by their locomotory organs (flagella) or passively by water currents [36, chap. 1]. Some other primary producers are seaweeds, salt marsh grass, seagrass and mangroves.

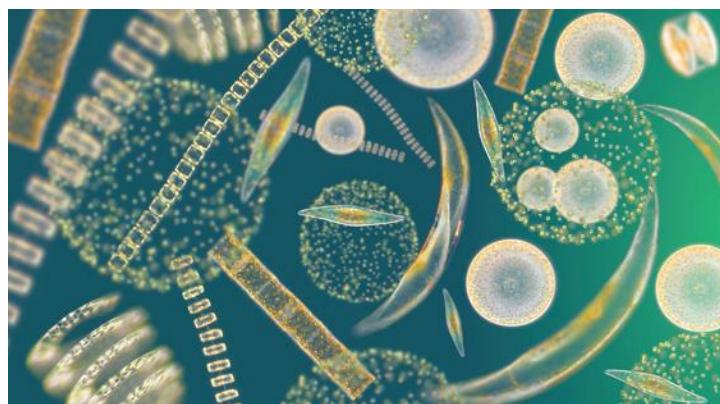


Figure 2.3: Common marine phytoplankton with unique shape and configuration

Phytoplankton are the primary producer organisms of the aquatic compartment of the marine ecosystem. In effect, they support zooplankton, fish and other members of aquatic fauna [36, chap. 1]. As a result, they are placed at the base of the trophic strata or bottom of the aquatic food web.

2.1.3 Biotic Variables - Consumers

The biodiversity of consumers in the marine ecosystem is immense, ranging from planktonic organisms to large mammals. On the smaller end, we encounter a category of species known as zooplankton. Like phytoplankton, zooplankton cannot move against the currents of marine ecosystems [32, chap. 6]. Furthermore, they are heterotrophs, which means they gain their organic matter from the predation and hence consumption of other organisms. Many are herbivorous (consuming phytoplankton) whilst omnivores

and carnivores can consume smaller zooplankton. They are typically the consumers in the second tier of marine and estuarine food chains.

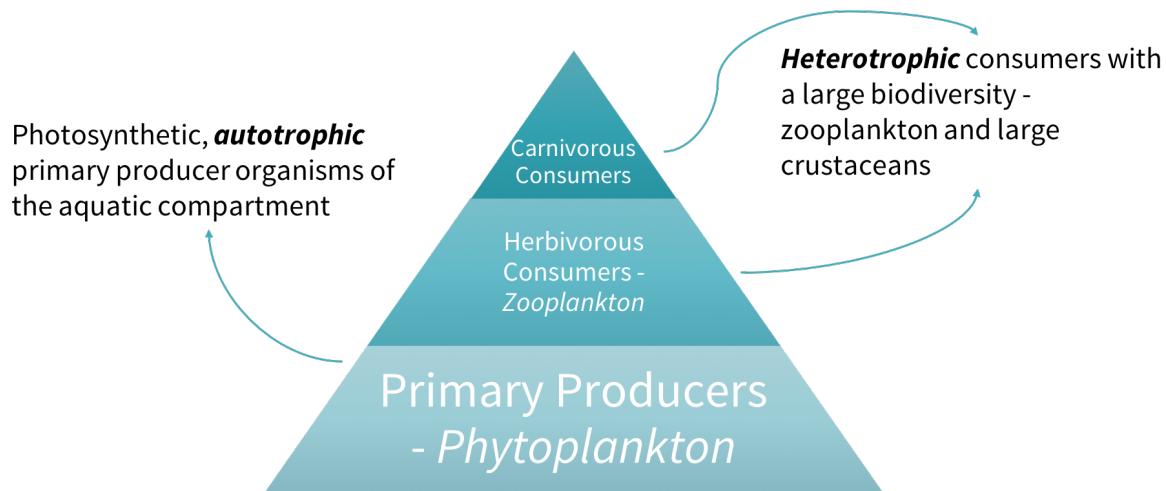


Figure 2.4: Abstracted trophic pyramid of the marine ecosystem

The inter-relationship between producer and consumer organisms within the marine ecosystem can be seen in Figure 2.4, where a dependency emerges between all the organisms at each trophic level. More formally, the primary producers support the populations of herbivorous and thus carnivorous consumers via the transfer of energy within the ecosystem. This trophic energy transfer is the basis of supporting the functioning of the ecosystem and is therefore a serious consideration for the dynamics of it.

2.2 Modelling the Marine Ecosystem

Approaching the modelling of the marine ecosystem begins with appreciating the dynamics that are inherent within them. Chapin et al. [14] makes the observation that “*ecosystem dynamics are a product of many temporal scales*”, and that the rate of ecosystem processes are particularly sensitive to fluctuations and activities of organisms on various time scales. Moreover, whilst early studies of ecosystems proposed that ecosystems were in equilibrium with their environment, conceptual advances within the field have “*increased the recognition of the importance of past events and external forces in shaping the functioning of ecosystems*”. This involves the recognition that most ecosystems experience disturbance as a natural component of their dynamics and do not exhibit one single stable equilibrium. More formally, ecosystems are seen as being in their ‘steady state’ if the balance between inputs and outputs to the system remains constant over time,

where this constant state requires continual work [14, 45]. Conversely, the equilibrium condition is maintained when no work is required to maintain it, and there is no input or output to the system. The open system nature of an ecosystem comes into conflict when defining an ecosystem as possessing a closed system equilibrium.

Yackinous [50] also makes the effort of describing ecosystem dynamics as that of representing an open dynamical system. He further defines an ecological system within a framework consisting of self organisation, regulation/adaptation and propagation. With self organisation, he cites the work of Odum [33] when describing the concept in terms of non-equilibrium thermodynamics. In particular, complex systems consisting of smaller parts will “*tend to organise to achieve some stable, pulsing state in the absence of external interference*” which in turn can only be maintained by a constant flow of energy through them, and are therefore not in thermodynamic equilibrium. In this respect, Yackinous [50] and Chapin et al. [14] agree, alongside many others within systems ecology. Yackinous [50] describes regulation and adaptation as being an inherent concept within ecological systems, noting that it is “*required for system survival*”. He also states that this regulation is homoerhetic, meaning that flow is stabilised and describes a fluid rather than static state of the internal environment, which would describe homeostasis [5], and is also a crucial part of any living system [31]. Finally, he describes propagation as the means of interaction between nodes within an ecological network and its environment, and notes that propagation as a concept is fundamental to synthesis and modelling. The unique interplay between this framework and a computational model provides invaluable insight to begin the ecological modelling of the marine ecosystem. That being said, it is worth appreciate ideas within complexity science as a comparison.

With the greater accessibility of powerful hardware, the notion of modelling ecosystems as complex systems has become popular in literature [30]. Wu and Marceau [49] defines a complex system as being

- Thermodynamically open, meaning that they exchange energy and/or mass with their environment
- Composed of a large number of diverse components
- A system where components interact with each other non-linearly, and frequently have response delays and feedback loops among them
- Exhibiting a high degree of heterogeneity in both time and space

In relation, Anand et al. [12] notes the increasing acceptance of ecological systems as

being characteristic 'complex adaptive systems', which are defined as having "*macroscopic system properties such as trophic structure, diversity, productivity relationships, and patterns of nutrient flux emerge from interactions among components, and may feed back to influence the subsequent development of those interactions*". Anand et al. [12] further describe examples of how complex systems science has enabled the production of models which have provided considerable insight into the dynamics of ecological systems. This has been enabled by "*by modelling ecosystems as simplified systems of interacting particles (e.g., individuals, agents, species), often in well-defined, unchanging spatio-temporal environments*". They also vividly describe how the complex systems approach has modelled and enabled discovery within sub field in Food Webs, Patch Dynamics and Population Fluctuations. Despite the clear advancements made within the Complex Systems approach in academia, attempting to build simulation based on such theoretical and complex definitions remains a difficult task. In similar fashion, other methods of modelling such as Silow et al. [44]'s proposal of thermodynamic modelling using the concept of Eco-Exergy produce similar results that are potentially difficult to translate to a valid computer model within the scope of the project. As a result, using the insight gained from systems and complexity theory other literature was sought out to better inform the problem at hand.

Fennel and Neumann [19] rigorously describes conventional methods to model marine ecosystems with ample consideration for the systems and complexity approach. In this book, a 'model' is described as "*theoretical descriptions in terms of sets of differential equations that describe the food web dynamics of marine systems*". These food webs in question are defined by a complex flow of materials through all trophic levels of the marine ecosystem. Therefore, unlike the models defined by Silow et al. [44] through the notion of exergy, Fennel and Neumann [19] considers the dynamics of the system to be driven by processes such as nutrient uptake, respiration, grazing, or indeed physical processes such as light and turbulence variability. Moreover, models are defined by state variables defined as "*well defined and measurable quantities, such as concentration of nutrients and biomass or abundance*" which characterise a system. Individual based models will consider individuals within the marine ecosystem as the basic units of the system. Conversely, state variable models will look at numbers or mass of very many individuals per unity of volume. Individual based models benefit from being able to represent genetic variation amongst individuals within the system, an advantage given that this relates to the project aims.

The principles of modelling postulated by Fennel and Neumann [19] enable an approach that better suits the other aims of the simulation, namely, to also model genetic vari-

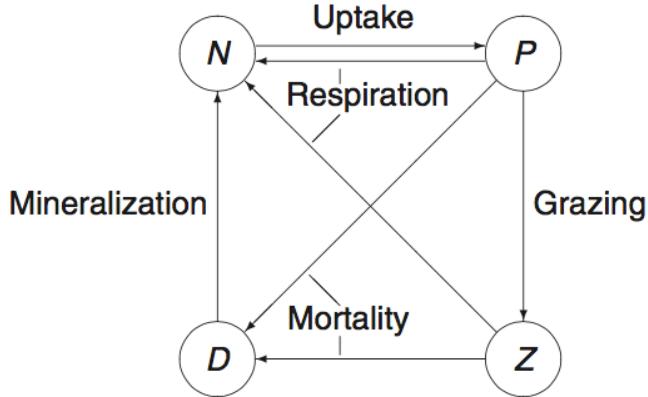


Figure 2.5: An example of a Biogeochemical model by [19], the NPZD model

ation and selection. Furthermore, the ideal focus on the marine ecosystem make their work particularly valuable in adopting a model approach. In building simple models to represent nutrient flow dynamics within the marine ecosystem, they make use of modelling entities as state variables in a dependency network as in Figure 2.5. In this model, the yearly cycle of nutrients and plankton is defined, and the rates of uptake, grazing, mineralisation, mortality and respiration are defined by differential equations. The rates of these equations will inevitably drive the dynamics of this model. Chen et al. [15] makes the important case that differential equation based approaches to biological (and hence ecological) modelling is limited in the sense that they assume the internal structure of the model is static, which does not hold when unpredicted system evolution occurs with time as our previous research has suggested. They also make the case for a closer semantic relationship between a conceptual and a computer model, and note the better suited nature of computational modelling in regards to tractability when model resolution or complexity is high. Like Anand et al. [12], however, Chen et al. [15] defines ecological systems as complex adaptive systems but posit a computational approach to their modelling.

2.3 Modelling Variation and Fitness

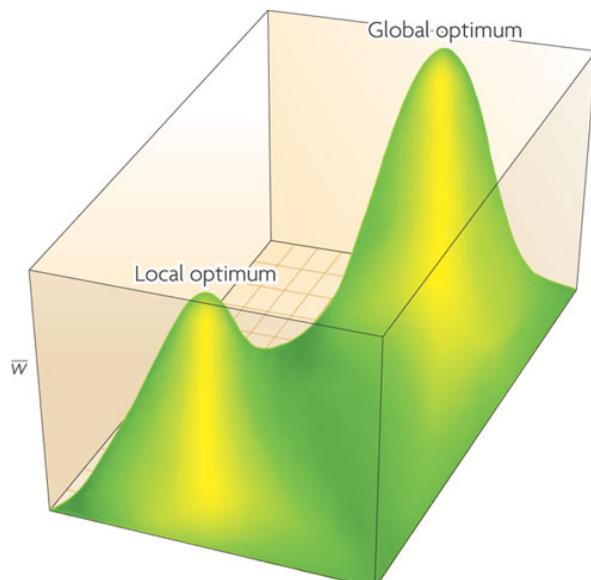
Modelling the dynamics arising from the inter-relationships and processes within the marine ecosystem is not enough to fulfill all the project objectives; we still require a way of representing the genotype (and hence the phenotype) of any one organism in our model. Given this, it is worth understanding important principles behind genetics in order to abstract and eventually model them within the organisms. However, some

thought must be given to the complexity of a genetic model for each organism and how the selection of the genes expressed in their phenotypes would occur based on the characteristics of their environment; there must be some measure of fitness. A review of research within this area was required prior to any modelling considerations.

The fundamentals of DNA are well understood in academia; genes that determine the physical characteristics of an individual are encoded within its structure, which in turn are maintained within a structure found within each cell - the chromosome. Moreover, organisms are considered *diploid* if they contain two copies of each chromosome. Each one in turn is inherited from either parent during sexual reproduction. Given this fact, genes will have two expressions if they are carried between diploid organisms, known as alleles. Both of these expressions are each a different version of the gene, and so an offspring will inherit a particular combination of their parent alleles within some probability. Moreover, alleles may be *dominant* or *recessive*; wherever a dominant allele is found within a genotype, the dominant one is expressed in the phenotype. The exception occurs in the case where both alleles are recessive, in which case it will be expressed [8]. Whilst this may be a vast simplification of the principles of genetics, the core of this concept remains and builds the foundational blocks of selection and thus evolution. More formally, within any interbreeding population there exists a gene pool characterising the sum of every gene in every organism within it. Selection will occur, based on the characteristics of the environment (and thereby the abiotic factors) on every organism; the environment will effectively determine the reproductive success of any expressed genotype in the gene pool. This measure may even be quantified, deemed the fitness of the genotype with respect to its environment [16]. Crucially, during the process of reproduction a mutation may occur in the genotype of an organism, and is the main cause of genetic variation in the gene pool [9]. Whilst the overwhelming frequency of these mutations produce no drastic effect toward the fitness of an individual, occasionally they may result in an increase in the fitness as perceived by the environment. As a result, selection will favour this genotype and we can expect an increase in the frequency of the now beneficial mutated allele in the gene pool.

To consider an approach to modelling the genotype of our organisms we must consider some method of abstracting the actual mechanism of DNA to something simpler and more expressive of the nature of *genes*. Moreover, the way genes are modelled must take into consideration the nature of the project, in that we require some quantitative measure of the effectiveness of the gene in performing the very task it is encoded for. Inevitably, this fitness measure should vary as the environment changes. A consideration for existing methods of addressing this issue is presented in the following section.

The idea of a Genotype-Phenotype mapping as some sort of function between the genotype and the resultant phenotype is a powerful concept introduced by Alberch [11] in 1991. Pigliucci [37] re-examines this original concept by contextualising it amongst the newer research produced in the domain. He reaffirms Alberch [11]'s original hypothesis that “*genes do not specify development ... but are instead one of several causal factors that are jointly determinant of the phenotype*”, and that an abstract $G \rightarrow P$ mapping (far more complex than a one-to-one relation) exists and can be modelled. Furthermore, he presents a summary of research conducted to unearth a $G \rightarrow P$ mapping of RNA folding and protein functions, and represent a direct empirical approach to mapping, noting the success of the experiments. However, he and many others who have attempted to model these maps acknowledge the complexity of the task at hand; Pigliucci [37] notes the incredible difficulty in creating a protein function. Otwinowski and Nemenman [35] reinforces this by noting that the incredible complexity of DNA has yet to be modelled within a $G \rightarrow P$ mapping context. Nonetheless, they present a mapping of the *E. coli* lac Promoter with significant depth and detail based on long nucleotide sequences of the actual bacteria. Both Pigliucci [37] and Otwinowski and Nemenman [35] summarise methods that abstract nucleotide sequences in a systematic method whilst still being able to produce some representation of genes. Crucially, this reinforces the obvious view that modelling the genotype of organisms within a marine ecosystem will take a great deal of abstraction. The question as to whether an expression in any visible form of the phenotype is required remains to be seen. Regardless, we still require some measure of the effectiveness of the eventual phenotype in the environment it manifests itself in.



Nature Reviews | Genetics

Figure 2.6: A three-dimensional Wrightian fitness landscape [34]

The notion of fitness can be described as “*the ability of organisms or, more rarely, of populations or species to survive and reproduce in the environment in which they find themselves*”, where the notion of mathematical fitness is deemed the most robust interpretation [34]. Further, Orr [34] defines the terms absolute and relative fitness, the former is assigned to the genotype and will refer to the genotype’s total expected fitness. The latter describes the same quantity but normalised, usually in relation to the fittest individual in the environment. He also introduces introduces the fitness landscape, an abstract concept first introduced by Wright [48] as seen in Figure 2.6. Wright [48]’s landscapes are seen as a three dimensional abstraction of allele frequencies at two different loci in two dimensions, with the third representing the mean absolute fitness for that allele in the environment space. This idea has since been extended for these structures to represent a discrete representation underlined by DNA or protein sequences [34]. Such a surface would be multi-dimensional and would involve adaptive walks promoted by natural selection across the fitness surface and mechanised through random mutation [34, 28, 29, 17]. These walks can be seen as an optimisation problem with which there exists many local and one global optimum [28], as illustrated in Figure 2.6. Indeed, such a surface would be classified as *rugged* [17], with many peaks and troughs to model the variability of fitness throughout the space. Such structures would be very useful in predicting the adaptation of certain sequences should the corresponding $G \rightarrow P$ mapping be correct. It seems, therefore, that if the project simulation were to have some representation of a genotype from which a fitness could be surmised through some function, we would be able to generate some N-dimensional landscape which would be explored through via natural selection, driven by mutation. However, given that our simulation explores the variability of the abiotic characteristics of the environment rapidly, we would expect the landscape to change - the fitness of a genotype may indeed be suddenly less fit in the newer, altered environment. The notion of a dynamic fitness landscape is therefore required.

Kauffman and Weinberger [29] defines adaptive evolution as “*uphill walks on rugged fitness landscapes*”, and presents the NK model which has some dynamic properties. More specifically, the model has a parameter K which increases the ruggedness of the landscape. However, the ruggedness is tuned within strict parameters; it is not variable with abiotic characteristics or indeed any notion of these characteristics. We require a model were the actual fitness value, be it discretely encoded in binary or continuous, be adjusted as the environment changes. Some adjustment to the NK model may bring us closer to this goal, but it seems a different representation of fitness measurement may be more appropriate. Nevertheless, the notion of fitness and the adaptive exploration of a landscape is a concept that this project will aim to explore.

Chapter 3

System Requirements

Given the intensely experimental nature of this project, including periods of research and prototyping followed by short development bursts, an agile methodology has refined the requirements specification of the of the project from the initial one outlined in the Specification and the Progress Report.

3.1 Functional Requirements

These requirements detail what the system should do, along with a statement of services the system should provide, and a detail on how the system should react in certain scenarios [27]. Preliminary to any software development, a model had to be designed that is defined by it's own requirements - these are also outlined below.

3.1.1 Model Requirements

F1: *The model must be modelled as a discrete event simulation*

F2: *The physical space of the marine ecosystem must be appropriately divided into subsections with respect to light availability*

1. Three zones will divide the ecosystem, namely
 - (a) The Photic Zone
 - (b) The Aphotic Zone

(c) The Abyssal Zone

F3: *The abiotic components of the marine ecosystem must be modelled, where components must vary between zones*

1. The chemical compounds involved in key life processes in the ecosystem, including
 - (a) Carbon Dioxide
 - (b) Oxygen
2. Light intensity as a representation of solar light energy
3. Temperature as a representation of multiple heat sources, primarily as solar heat energy

F4: *An abstract representation of some oceanic organisms found within the marine ecosystem must be modelled, including*

1. Autotrophic producers, modelled after Phytoplankton
2. Herbivorous heterotrophic primary consumers, modelled after Zooplankton
3. Carnivorous autotrophic secondary consumers, modelled after Large Crustacean Zooplankton

F5: *A trophic hierarchy supported by Phytoplankton must be modelled*

1. Zooplankton graze on Phytoplankton at the base of the hierarchy
2. Crustacean prey on Zooplankton and occupy the topmost level in the heirarchy

F6: *Essential life supporting processes must be modelled in each organism*

1. Phytoplankton perform Photosynthesis, depleting a certain amount of Carbon Dioxide from the zone it occupies and augmenting Oxygen as a by-product
2. Zooplankton and Crustacean perform Respiration, depleting a certain amount of Oxygen from the zone it occupies and augmenting Carbon Dioxide as a by-product

F7: *Each organism in the ecosystem must possess a genotype*

1. The genotype must possess genes that contribute to the cumulative fitness of the organism, these being, with respect to the organism they are found in:
 - (a) Phytoplankton - LightGene (Photosynthesis)
 - (b) Phytoplankton - CarbonDioxideGene (Photosynthesis)
 - (c) Zooplankton/Crustacean - OxygenGene (Respiration)
 - (d) Zooplankton/Crustacean - TemperatureGene (Respiration)
2. Each gene in turn must possess a continuous value that represents the fitness of the gene with respect to the abiotic variables in the zone it currently occupies at the time
3. Each gene must have the capacity to mutate from its continuous value by some randomly distributed displacement
4. Each genotype must possess two allele expressions of each gene in every genotype, each inherited from either parent, grouped into a Chromosome
5. Each allele will have a dominant or recessive characteristic that describes its expressive potential in the genotype

F8: *Each organism must have the capacity to mate and produce offspring*

1. Offspring genotypes must be generated from crossover pairing of parent allele pairs. A probability is associated with which pairs will end up in the offspring genotype
2. Random mutations must occur in the genotypes of these offspring at the mating stage
3. Each mating pair may have a number of offspring; the probability of having a higher frequency of offspring increases as the cumulative fitness of both parents increases

F9: *Predatory organisms must be able to prey (or graze) and remove the prey from their zone if predation is successful*

1. The likelihood of predation occurring paired with the frequency of prey removed from the zone it occupies in one predation event is bounded by some probability specific to that species

F10: *The ecosystem must possess some selection mechanism which removes organisms from the zone they occupy if their cumulative genotype fitness falls below some survival threshold*

3.1.2 Software Requirements

F11: *The system must provide a cross-sectional view of the ecosystem in two-dimensional space*

F12: *Each organism must be represented by an ellipse, where the colour and size of which is a differentiation between species*

F13: *Each organism must possess some movement in the space it occupies*

1. The movement is bounded by the width and height in world co-ordinates of the zone it occupies
2. Movement is random and non-direction

F14: *Each frame in the system must represent one discrete time step in the simulation*

F15: *The system must provide the functionality to adjust parameters in the simulation*

1. The user must be able to increase or decrease the speed of the simulation (enacted by varying the frames per second)
2. The user must be able to increase or decrease abiotic variables for each zone in the ecosystem, including
 - (a) Carbon Dioxide
 - (b) Oxygen

F16: *The user must be able to move around the ecosystem using the mouse, including pan and zoom functionality*

F17: *The interface should provide analytics regarding the state of the zone the user is currently surveying updated at each time step, including*

1. The levels of each abiotic variable for that zone

2. Every gene present in the zone, the dominance of the gene, the value and finally the frequency of the gene in the ecosystem
3. The frequency of every species in the zone

F18: *The user must be able to cycle through the analytics for each zone*

3.2 Non-functional Requirements

NF1: *The system should be executable on a range of operating systems and form factors*

NF2: *The system requirements should be low, thus increasing the accessibility of the tool*

NF3: *The system must be visually informative and obvious to use, simulating at the required depth whilst omitting irrelevant detail*

NF4: *The system must be extensible and provide the ground work for extension*

NF5: *The system should be able to execute without the requirement of a dedicated graphics processor; an integrated CPU based solution should be sufficient*

NF6: *The system should be vigorously documented and developed within a framework to allow easy extension and better understanding for future work*

Chapter 4

Design

This chapter describes the formulation and specification of the design for the system. Initially, the conceptual model is presented, along with an analysis of the behaviour of the system that encapsulated it. This model in question was designed based on the research that was conducted, and thus the evolution of the model is also discussed to provide better context to the iterative and experimental nature of the project. Following this, the architecture of the software implementation of this model is briefly described, followed by a description of the user interface design that accompanies the back-end.

4.1 The Conceptual Model

The research section presented a small snapshot of the vast amount of literature surrounding the topic of ecological modelling. As mentioned in the section, [19] makes the case of ensuring that the modelling objectives are well understood before descriptions of a model can begin to take shape. The requirements specification in the previous section outlined what we would like our model to do, and the mechanism of the simulation based on these requirements is what the conceptual model describes.

Figure 4.1 illustrates the conceptual diagram of the model that the simulation is built upon; a high level description of an abstracted marine ecosystem. Every entity in this description represents a dynamic quantity which fluctuates over time. The entities P, Z and C represent the three species that were chosen to be represented in our model; Phytoplankton, Zooplankton and Crustacean respectively. As highlighted by the requirements specification, the abiotic components of the ecosystem chosen to be simulated are

Carbon Dioxide, Oxygen, Light and Temperature. Of these, only Carbon Dioxide and Oxygen will fluctuate in quantity over time through the photosynthetic and respiratory effects of the organisms in the ecosystem, and are hence represented as entities in the model. The other factors will remain as constants. These processes are represented in the model diagrams through the arrows between the gases and the organisms. More specifically, Phytoplankton will perform photosynthesis, depleting a certain quantity of Carbon Dioxide from the ecosystem and augmenting a similar amount of Oxygen in the process. Similarly, the respiratory organisms Zooplankton and Crustacean will perform respiration, depleting Oxygen and augmenting Carbon Dioxide in the same fashion.

As seen in the diagram, the arrows represent uptake or output from the organism entity. The uptake arrows Grazing and Predation describe the food chain in this model. Specifically, Phytoplankton are at the base of the food chain and are therefore the primary producers in this model. Zooplankton will graze on the Phytoplankton, making them the primary consumers. Finally, Crustacean make up the top of the food chain as the secondary consumers by predating on Zooplankton. By virtue of these fluctuating entities which are controlled by the processes just described, we can begin to see how the dynamics of the model ecosystem can be very vaguely described by Photosynthesis, Respiration and Predation.

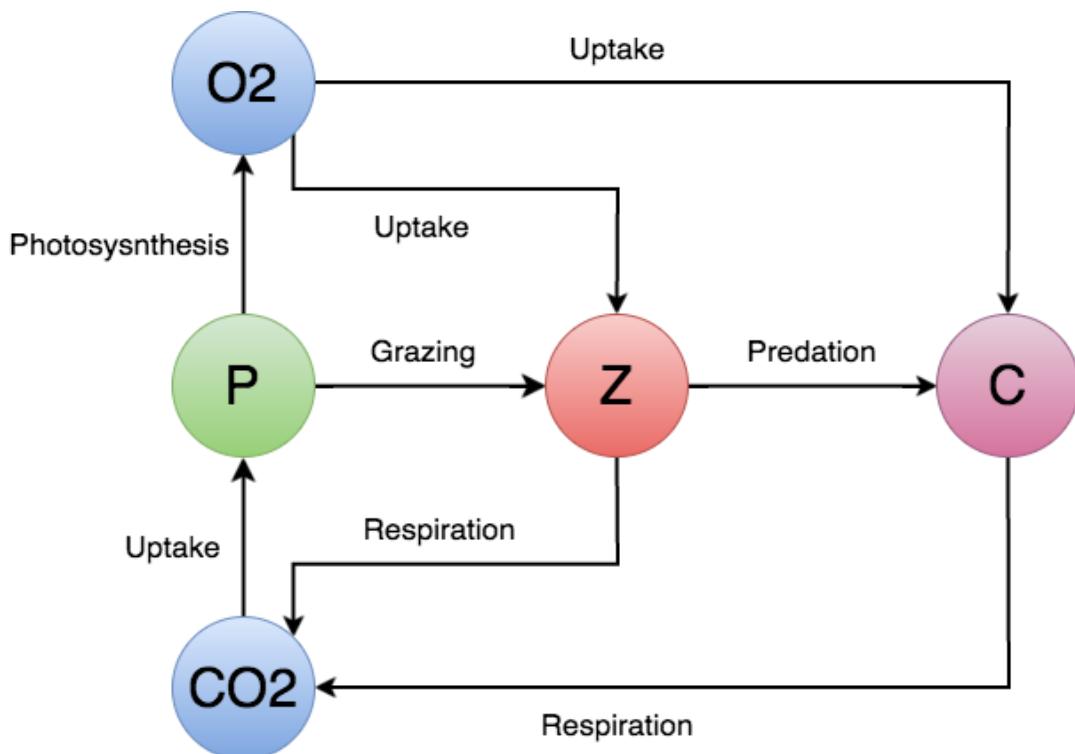


Figure 4.1: The conceptual model of the system

This diagrammatic description shows the dependency relationship that exists between the entities in the ecosystem. It represents a vast abstraction of the marine ecosystem, whilst preserving the features of the ecosystem we wish to model for the sake of the project goals. These features still require a lower level description, and will be discussed in the following sections. However, just by observing Figure 4.1, one can clearly see a dynamic system emerging that is driven by the key processes outlined, where states in our system are the entities in the conceptual model. Moreover, this system can be described mathematically.

The dynamics of the system will also be discussed in appreciation of the lower level mechanics that are yet to be outlined. Furthermore, in beginning to appreciate the notion of a simulation, a sense of time must be invoked into the system. More specifically, whilst the conceptual model can be described as continuous, our system will be simulated as a discrete event simulation. As Ewald et al. [18] importantly points out, continuous system models within Biology (and Ecology) are useful and are usually described by partial differential equations. On the other hand, stochastic discrete simulations present a different perspective on the deterministic nature of biological and ecological processes, which Ewald et al. [18] notes has also seen incredible success. Finally, given that our model must be built as a piece of simulation software, it makes sense to treat the simulation as discrete and stochastic in nature.

4.2 The Ecosystem

The research conducted into the features of the marine ecosystem indicated the ecosystem can be divided with respect to the light penetration proportional to the depth of the ocean. Simply by this fact, the abundance of species and nutrient levels fluctuates (see Figure 2.2). Figure 4.2 illustrates the physical space model for the simulation, and is based on the marine ecology discussed. More formally, the space is divided into zones, where each zone is characteristically different depending on the biochemical implications of the varying light level.

The Photic Zone receives the most light in the ecosystem and is built to model the upper region of the marine ecosystem (and indeed the actual Photic Zone as described by Mitra and Zaman [32]). As such, it is unlikely that the light level will be a limiting factor for photosynthesis. Therefore, Phytoplankton will likely thrive here, resulting in a high uptake of Carbon Dioxide and a correspondingly high attenuation of Oxygen. The zone also receives a constant supply of Carbon Dioxide from the environment, supporting

the ever growing Phytoplankton population. The abundance of Phytoplankton and the high level of Oxygen provides a suitable habitat for grazing Zooplankton, whose population will also flourish here. As a result, the suitable level of prey results in the presence of many predatory Crustacean, who will inevitably compete for Oxygen with the Zooplankton. The interesting dynamics of the interdependent populations in this zone will be discussed further in this chapter.

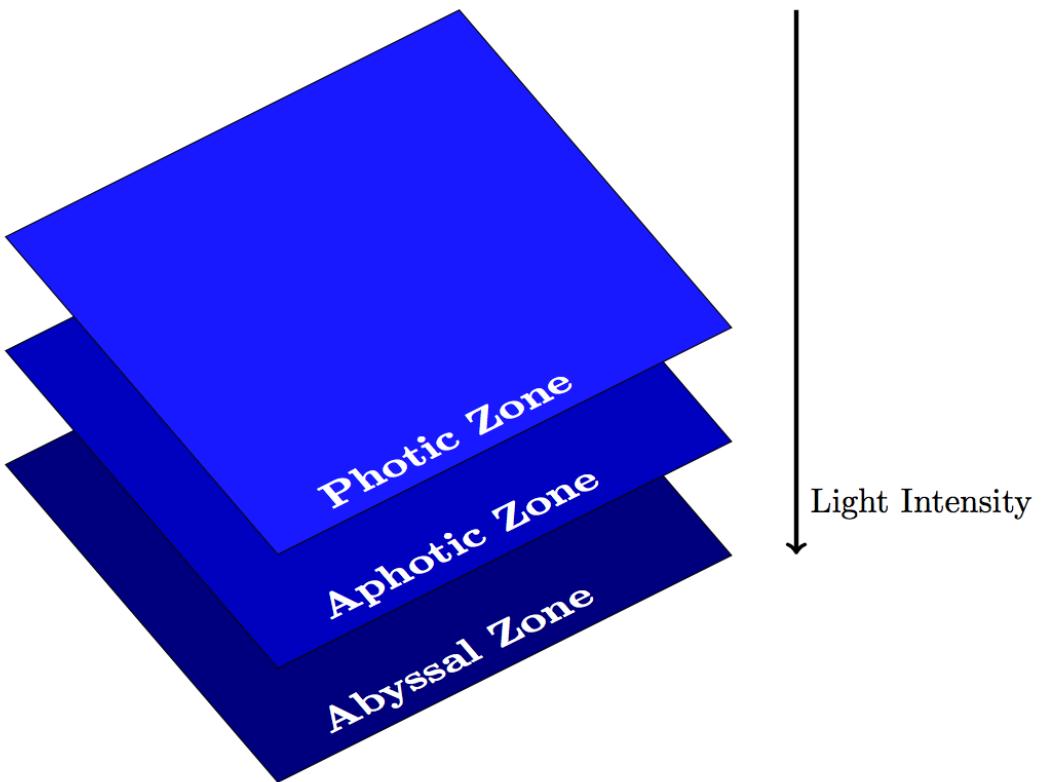


Figure 4.2: A model of the physical space of the ecosystem

The Aphotic Zone is appropriately named due to the absence of light in the marine ecosystem. In the model, however, the light is simply low enough such that it limits the success of Photosynthesis for most genes that occupy it. This limitation will limit the Phytoplankton population and hence limit the Zooplankton and Crustacean population as a result of the low amounts of Oxygen produced by the few Phytoplankton and the hence the few prey for the consumer population.

Finally, the Abyssal Zone is so named because it models the oceanic abyss, where absolutely no light will penetrate. The limitation of any realistic photosynthesis effectively eliminates the possibility of any sustainable Phytoplankton population. The dynamics that result from preventing the extinction of organisms is interesting and is discussed

further. It is worth appreciating the physical space model in the context of the conceptual model before the simulation mechanics and the genetic models are introduced. Specifically, the rates of uptake and hence the rates of population growth as highlighted by the model in Figure 4.1 will inevitably differ in each zone due to the variance of abiotic characteristics between them. Therefore, this simulation will effectively possess three models, each of which possesses its own entities independent of each other.

4.3 Genetics and Fitness

4.3.1 Initial Experimentation

The research extensively discusses the various ways that genotypes have been modelled in academia, along with the notion of fitness, mutations and evolution through selection. In the exploration of modelling genes in the organisms in the model ecosystem, it was important to understand the motivations and goals of this particular model in question.

An essential component of the conceptual model is the dynamics that occur when organisms perform photosynthesis and respiration which drive the exchange of gases and hence fluctuate population sizes. Further, each organism in turn should possess a genotype which contains genes which encode the function of this process. Indeed, genetic variation infers that the fitness of genes that perform photosynthesis or respiration will vary within a population. It is this variation due to random mutation and mating that enables the selection of the fittest individuals in a gene pool, thus driving evolution.

In exploring the problem of modelling the genotype of marine ecosystems, an abstraction of DNA was initially sought that could be quantised to some fitness value which could undergo selection. Such an approach could be likened to a meta-heuristic solution to an optimisation problem similar to a genetic algorithm [25]. A novel approach would be to encode genes with some binary string quantised and evaluated by some cost function which represents the fitness of this gene. The algorithm would select the fittest individuals in the population and pair them up for mating, crossing the strings over to produce a new gene for an offspring as a combination of the parent genes. Finally, a random point mutation somewhere in the sequence will introduce slightly more variation before the new solutions are re-evaluated in selection and the process begins again. This approach borrows key concepts in genetics to produce a solution which (should) in theory converge to global optimum on the fitness landscape. In adapting a genetic algorithm for the model, a cost function will enable us to appropriately engineer an encoding for

photosynthesis and respiration.

Haupt and Haupt [25] presents many optimisations for genetic algorithms, and stresses the importance in selecting a cost (or fitness function) that can evaluate the success of any genetic solution produced by the algorithm. A simple function that would produce some continuous value for the fitness of a gene would be an easy solution to the problem. However, an important feature of the model is that abiotic variables will be a parameter in our fitness function. More specifically, we have a dynamic fitness landscape; a vast alteration of an abiotic variable will result in genes that were once local or even global optimum on our fitness landscape to reduce in altitude in our n-dimensional space. A static cost function does not take into account the dynamics of ecosystem selection; we desire a cost function that is variable to the abiotic factor important to that process.

Early experiments using this approach were conducted in order to assess the suitability of this approach as a solution. Genes are represented as nine bit binary strings in these experiments , and the Lyapunov energy function in 4.1 as adapted for the neural hopfield network was chosen as the cost function. This function was chosen due to the pliability of the weight matrix; the fitness of a gene will be different depending on the weight matrix. If the abiotic factors of the environment could be encoded within it we may have achieved a dynamic fitness landscape.

$$-\frac{1}{2} \sum_{i,j} w_{i,j} g_i g_j \quad (4.1)$$

Table 4.1 shows the results of an early experiment using this method conducted in Mathematica. The leftmost table shows a small gene pool all of which have been given a fitness as determined by 4.1 using the inhibition constant $i = 0.112455$ which was trivially determined. The lowest cost solution in this fitness pool is the fittest in this context. Selection occurred on the fittest half of the gene pool before mother and father pairs were selected, mated and crossed over to provide new combinations of the previous gene pool. Point mutations were not included in this genetic algorithm for the purposes of illustrating the convergence of basic re-combination.

The rightmost table describes the resulting fitness after one iteration; the average fitness has diminished regardless of any change to the weight matrix. Given that a genetic algorithm is designed to find optimal solutions after a large number of iterations, this approach seems unsuitable as a model of genetics which attempt to model that of actual organisms that live within an ecosystem. More specifically, it is unrepresentative to expect some offspring to inherit genes drastically less fit than it's parents if it performs

fitness	encoding	fitness	encoding	fitness	encoding
-3.37365	1,1,1,0,0,1,1,0,1	-3.37365	1,1,1,0,0,1,1,0,1	-1.14874	1,1,1,0,0,1,1,1,0
-1.34946	0,0,1,0,0,0,1,1,1	-1.34946	0,0,1,0,0,0,1,1,1	-0.22491	0,0,1,0,0,0,1,0,0
-0.67473	0,0,1,0,0,0,1,0,1	-0.67473	0,0,1,0,0,0,1,0,1	-0.22491	1,0,0,1,0,0,0,0,0
-0.67473	0,0,1,0,1,0,0,0,1	-0.67473	0,0,1,0,1,0,0,0,1	0.87545	0,0,0,0,1,0,1,1,1
-0.22491	1,0,0,1,0,0,0,0,0	-0.22491	1,0,0,1,0,0,0,0,0	-0.67473	1,0,0,0,1,0,1,0,0
-0.02419	1,0,0,1,1,0,1,0,1	-0.02419	1,0,0,1,1,0,1,0,1	-0.67473	0,0,1,0,0,0,0,1,1
0.87545	0,0,0,0,1,0,1,1,1	0.87545	0,0,0,0,1,0,1,1,1	-0.22491	0,0,1,0,1,0,0,0,0
0.87545	0,0,0,1,0,1,0,1,1	0.87545	0,0,0,1,0,1,0,1,1	0.87545	1,0,0,1,1,0,1,0,0
0.87545	1,0,0,1,0,1,0,0,1	0.87545	1,0,0,1,0,1,0,0,1	1.55018	0,0,0,1,0,1,0,0,1
0.87545	1,0,0,1,1,0,0,0,1	0.87545	1,0,0,1,1,0,0,0,1	1.55018	1,0,0,0,1,1,0,0,0
0.87545	1,1,0,0,1,0,0,0,1				
0.87545	1,1,0,0,1,0,1,0,0				
0.87545	1,1,0,1,0,0,0,1,0				
1.55018	0,1,0,1,0,1,0,0,0				
1.55018	1,0,0,1,0,1,0,0,0				
3.10036	0,1,0,1,1,0,1,0,0				
3.10036	1,0,0,1,1,0,0,1,0				
4.17653	1,1,0,1,1,0,1,1,1				
4.17653	1,0,1,1,1,1,1,1,0				
9.97581	0,1,0,1,1,1,0,1,1				

Table 4.1: Genetic algorithm experiment

the same function and has not experienced any mutation. This approach fails to model chromosomes by breaking apart genes in the pursuit of an optimal solution, which would remain an issue with the use of genetic algorithms despite the nature of the cost function. A better, more representative model will be developed that better meets the project goals in trying to simulate and model the genetics of oceanic diploid organisms.

4.3.2 The Logistic Model

Following the failure of the genetic algorithm to appropriately model the genotype of the marine organisms, another encoding and associated fitness function was sought. The logistic function is commonly used in biology to model population growth, and it so happens to possess features which could be beneficial to modelling genes.

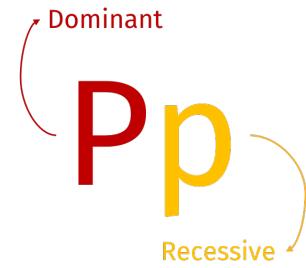
Before introducing the solution, it is worth reiterating the design of the genotype. The marine ecosystem model will contain four different genes:

- LightGene

- CarbonDioxideGene
- OxygenGene
- TemperatureGene

The former two occur in the genotype of Phytoplankton and contribute to the success of photosynthesis, whilst the latter genes occur in the genotype of Zooplankton and Crustacean and contribute to the success of respiration. Moreover, every organism will have two expressions of every gene in the form of an allele (forming a Chromosome) each of which is inherited from either parent. Each allele will possess a quality of either dominance or recessiveness; only the dominant allele will actually be expressed in the phenotype of the organisms, unless two recessive pairs exist in the Chromosome, in which case either recessive allele will be expressed. This forms the basis of Mendelian genetics.

We now require some encoding for the genes which may be considered for fitness and be susceptible to random mutations. Consider the logistic curves in Figure 4.3, which represents two different alleles of the chromosome Pp. If our fitness is simply an output of the function that produces these curves, it is clear that when the associated abiotic factor is high the dominant allele will be fitter in the gene pool and selection will hence favour it. Conversely, if the abiotic factor is low, the opposite is true. Therefore, by representing genes by these curves, we can measure fitness and achieve genetic variation in our gene pool with enough variability such that selection can be driven. The plots are given by Figure 4.3



$$f(x) = \frac{1}{1+e^{-\theta x}} \quad (4.2)$$

$$\begin{aligned} f(x) &= \text{fitness} \\ x &= \text{abiotic factor} \\ \theta &= \text{encoding} \end{aligned}$$

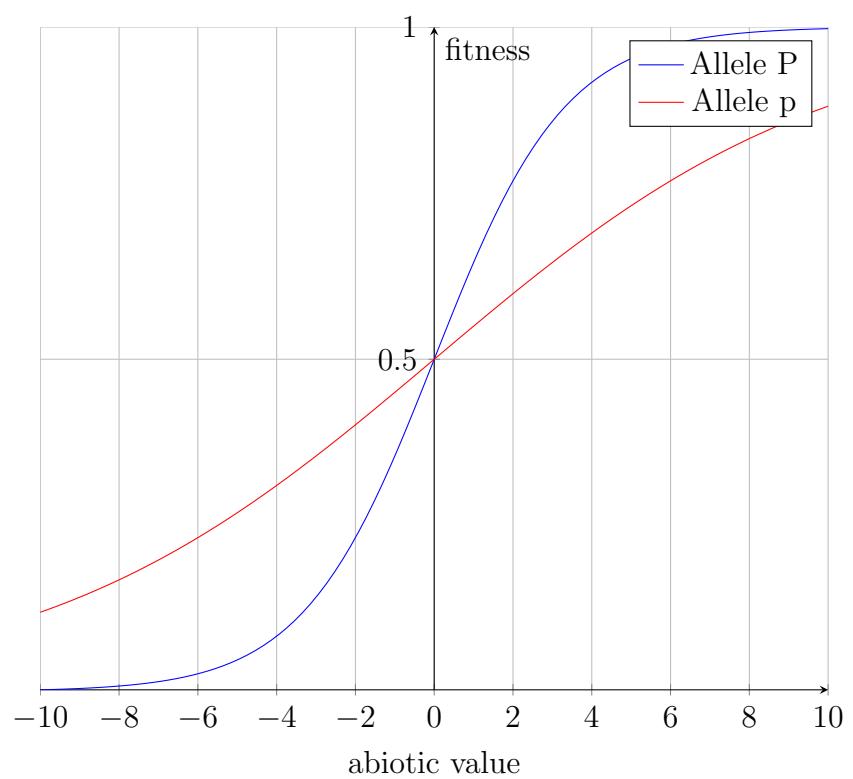


Figure 4.3: Plots for two alleles of Chromosome Pp

The encoding is determined by θ and serves to displace the logistic curve from its original position. Therefore, if different alleles had different values of θ we will simulate variation amongst alleles in the gene pool. Further, the curve produces a constraint on the fitness naturally, meaning that an increase in the abiotic factor past a certain point should not result in any significant increase in fitness. To further constrain the abiotic factor, the value of any factor in any zone at any one time will be mapped to 4.3. This function serves to constrain the abiotic value to the domain of the gene function, where α is regarded as the effective carrying capacity of an abiotic factor in the zone. This is so named because it constrains the population size of any species that utilises that factor, as it puts bounds on how many organisms can effectively deplete a factor before the level in the zone becomes too low to support the process in the first place; the larger the domain of this function, the more factor is available for this zone. In this example, we denote the abiotic carrying capacity to be $[-40, 40]$.

$$\frac{10}{1+e^{-\alpha x}} - 5 \quad (4.3)$$

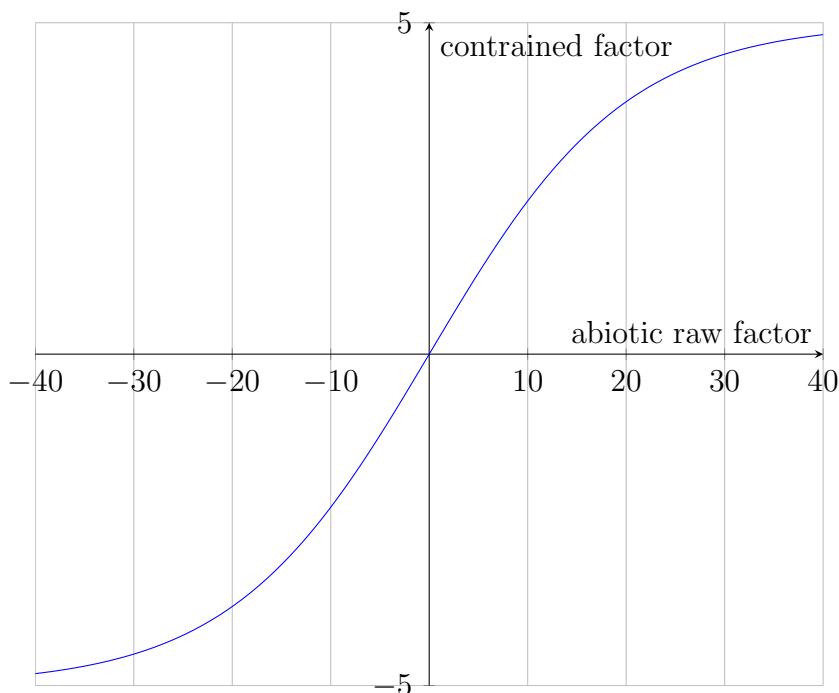


Figure 4.4: Abiotic constrain mapping

4.4 The Simulation Cycle

The static structure of the simulation thus far has been described in the conceptual model, the physical space model and the genetic model. In order to perform any meaningful simulation, an architecture that drives the simulation must manipulate these abstract structures based on certain rules. The discrete time simulation is described in this section; each step in the cycle performs a crucial process that contributes to the dynamics of the system. Following this, the system architecture that enacts this system is described following an explanation of the software system. Figure 4.8 illustrates the simulation cycle. This cycle will be performed for each zone in the ecosystem continuously until the simulation is terminated.

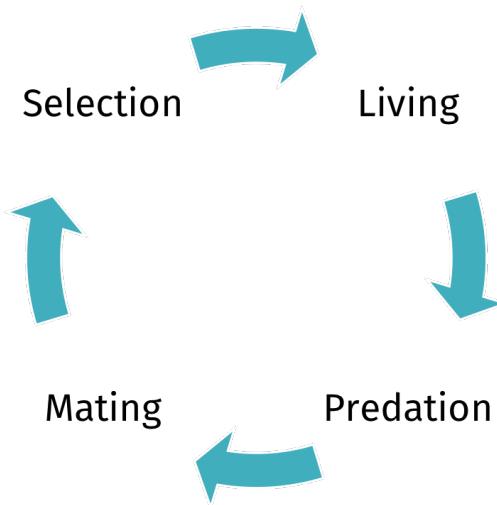


Figure 4.5: The simulation cycle representing one time step

Algorithm 1 Simulation cycle procedure

```
while simulation is running do
    for all Zones do
        ZONE.SELECTION
        ZONE.LIVING
        ZONE.PREDATION
        ZONE.MATING
    end for
end while
```

4.4.1 Selection

During the selection procedure, each organism of every species in every zone is evaluated for survival. More specifically, a survival threshold is introduced to the logistic fitness

function described in 4.2 which sets a lower bound on the survival of an organism. This value is determined to be the level at which any allele that produces a fitness below the threshold will be ineffective in their desired task entirely, and should all the genes in an organism's genotype produce a fitness value below this threshold, it will be removed from the population. Thus, only the fit will survive to pass on their genes to their offspring and hence the gene pool in the following sequences.

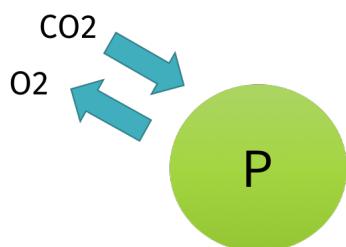
When considering the shape of the logistic gene curve, one can see how constraining the abiotic variable of either Carbon Dioxide or Oxygen will apply a selective pressure on the population, as curves that are steep (and therefore better) will likely survive over curves that aren't and will produce fitness values below the survival threshold. Moreover, should a user be tempted to reduce an abiotic factor dramatically, they would see the selection procedure begin to kill off the unfit alleles, leading to some alleles being removed from the gene pool entirely. Indeed, the natural depletion due to the fluctuating dynamics can achieve the same effect.

Algorithm 2 Selection procedure on Zone population

```

for all organism in zone do
    if organism.genotypeFitness < organism.threshold then
        ZONE.REMOVE(organism)
    end if
end for
```

4.4.2 Living



Following the selection performed on the zone population, the organisms that have survived will now be given a chance to *live* by systematically allowing each organism to perform their life function; photosynthesis or respiration. The mechanics of the life process will be determined by the allele an organism has expressed in their phenotype (the dominant allele if homozygous dominant or heterozygous, or either recessive allele if homozygous recessive). In particular, each organisms will deplete either Carbon Dioxide or Oxygen by an amount as determined by the the fitness value of CarbonDioxideGene or OxygenGene respectively. Similarly, it will augment the opposite gas by the exact same amount, leading to an effectiveness replacement of gases.

During the iterative process, the input gases for the process under consideration will

gradually deplete from the zone as Living occurs. Indeed, if the population size for the species under consideration exceeds the carrying capacity, the population could effectively be entirely removed in the next iteration of the simulation through selection, leading to mass extinction. In order to prevent this from occurring, selection is also assessed on every organism during the iteration of the Living procedure. In essence, this means that if an organisms is considered during this procedure, it will assess if the organisms considered before it has depleted the abiotic factor enough such that the allele under consideration produces a fitness below it's survival threshold. If it falls below, it is clearly unfit for the population at the time as theoretically it should not be able to perform the life process successfully in the first place. This selection procedure will of course consider the cumulative fitness of the genotype.

4.4.3 Predation

Selection has been performed and the surviving organisms have performed their essential life function, thus adjusting the states of the model entirely. During Predation the consumer organisms Zooplankton and Crustacean are given the chance to graze on Phytoplankton or predate upon Zooplankton respectively. To enable this, each consumer is considered iteratively for this procedure.

In order to assess the effectiveness of predation in a predator, another property is introduced to all organisms (including Phytoplankton) called the *Energy* of the organism. The energy in question corresponds to the cumulative fitness of the genotype. Using this value, we set the number of organisms to be preyed upon by any considered predator during predation to be a random number between the *Energy* of the predator and an upper bound experimentally determined by the designer. As such, weak organisms will have less likelihood to successfully predate, mirroring the energy impact on real predators in the marine ecosystem. The dynamics that result from predation are crucial to the simulation.

4.4.4 Mating

The final stage in the simulation cycle is Mating, where the surviving organisms after Selection, Living and Predation are divided into maternal and paternal sub-populations and are mated with each other, producing offspring which inherit the alleles of their parents and are thus added to the zones where the mating occurred. This is a crucial

part of the simulation and the principles of Mendelian genetics are heavily considered during this procedure.

Initially, a random (even sized) mating pool is selected whose size is randomly determined between two and the effective population size. This pool is then split into the mother and father groups, and each is paired with the other and considered for mating.

Building on the principles of genetics established earlier, we need to create a new genotype for each offspring determined by crossing over the chromosomes for each chromosome pair. More specifically, a Mendelian cross is determined for each chromosome in the genotypes of the parents, and a random permutation of alleles (a new chromosome) is selected for the offspring genotype. An example crossover is illustrated in Figure 4.6.

Crucially, before a random combination is assigned to the genotype of an offspring, a random mutation may occur on the encoded value of a randomly selected allele in a chromosome in this genotype. This procedure will introduce the genetic variation into the gene pool which will drive evolution in the sense of a random displacement of the genetic logistic curve; a random mutation may produce a steeper curve, or even a gentler curve that performs better at lower abiotic values. Either way, the fitness will change and either be successful or detrimental to the allele in question as dependent on the abiotic factor in the environment in the future. The amount of the encoding to displace (θ in 4.2) will be a random amount constrained between the *Energy* (as mentioned before) of the organism and some experimentally determined parameter. This parameter in question will determine how readily mutations occur, and indeed by how much even during the simulation and hence remains important to the dynamics of the simulation.

Much like the predation, a parent pair may be offered the chance to produce zero to many offspring, the likelihood of which is determined between the *Energy* of the parents and an experimentally determined upper bound. This parameter is also crucial to the dynamics of the simulation as it influences the mating rate of a species.

P	p		
P	PP	Pp	
p	Pp	pp	

Figure 4.6: A Punnett square showing a crossover of two chromosomes

4.5 The Dynamics of the System

The simulation cycle enables the ecological system to work and enables the dynamics of the system illustrated by the conceptual model in Figure 4.1. In the context of the simulation cycle, we can express the expected change in the state variables from an arbitrary time step t to $t + 1$ mathematically as a combination of all changes due to the actions of each step in the cycle. These succinct equations are expressed in Figure 4.7.

$$\Delta P = Mating(P, t) - Predation(Z, t) - Selection(P, t) \quad (4.4)$$

$$\Delta Z = Mating(Z, t) - Predation(C, t) - Selection(Z, t) \quad (4.5)$$

$$\Delta C = Mating(C, t) - Selection(Z, t) \quad (4.6)$$

$$\Delta co_2 = \sum^Z \frac{1}{1 + e^{-\theta_{Zco_2} \cdot Abiotic(o_2, t)}} + \sum^C \frac{1}{1 + e^{-\theta_{Cco_2} \cdot Abiotic(o_2, t)}} - \sum^P \frac{1}{1 + e^{-\theta_{Pco_2} \cdot Abiotic(co_2, t)}} \quad (4.7)$$

$$\Delta o_2 = \sum^P \frac{1}{1 + e^{-\theta_{Pco_2} \cdot Abiotic(co_2, t)}} - \sum^Z \frac{1}{1 + e^{-\theta_{Zco_2} \cdot Abiotic(o_2, t)}} - \sum^C \frac{1}{1 + e^{-\theta_{Cco_2} \cdot Abiotic(o_2, t)}} \quad (4.8)$$

Figure 4.7: The change in state density from time t to $t + 1$

If we can appreciate the mass of any state in the model at any one time, we can begin to describe the state changes as described by Figure 4.7. In particular, the states P , Z and C represent the accumulated biomass of all organisms of their respective species; the higher the population size, the larger the biomass (if we assume the unit biomass of each organism is uniform). Conversely, we regard the mass of the gases Carbon Dioxide and Oxygen to be an arbitrary quantity which is constrained by the logistic constraining function in 4.3 within some range $[-x, x]$ deemed to be the effective carrying capacity of the zone. Furthermore, the equations express functions that enact some change in this state. $Mating(Species, t)$ describes the addition of organisms due to Mating occurring on a species at time t . $Predation(Species, t)$ describes the loss of species as a result of the predation enacted by the organism at time t . Finally, $Selection(Organisms, t)$ describes the loss of species to due Selection at time t .

Closer examination of the equations in Figure 4.7 that describe the dynamics of the model in Figure 4.1 indicate the presence of a dynamic system driven by the simulation. Indeed, the model itself possess many cycles, indicating the presence of many equilibrium around states within the model. This is also apparent in the equations in Figure 4.7.

Figure 4.8 illustrates the cycle that surrounds the gaseous masses in the model, and is

described by the equations 4.7 and 4.8. These equations describe the change in each gas measured as the difference between the augmented and depleted amount of gases during the Living stage in the simulation cycle. Specifically, 4.7 describes the difference between the augmented amount of Carbon Dioxide produced by the sum of all respiration genes within Zooplankton and Crustacean, and the depleted amount through photosynthesis in the genes of Phytoplankton. Note that this quantity is described by the sum of every gene logistic equation, where the θ value is combined with the abiotic factor in question for that specific time step. Similarly, 4.8 describes the difference in Oxygen as a result of the augmentation through the sum of all photosynthesis and the depletion of which through the sum of all respiration.

Closer examination of the terms in 4.7 and 4.8 reveals an interesting property of this system. Specifically, should the third term denoting the photosynthetic effects from Phytoplankton be larger than the first two in 4.7, there will be a net loss in Carbon Dioxide in that time step. Further, should this term consistently dominate as t grows larger, Carbon Dioxide should, in theory, be completely depleted. The same principle will apply to 4.8, and can be seen visually in Figure 4.8. It seems clear that if these terms are balanced out an equilibrium is reached between the gain and loss of gaseous terms through the rates of photosynthesis and respiration.

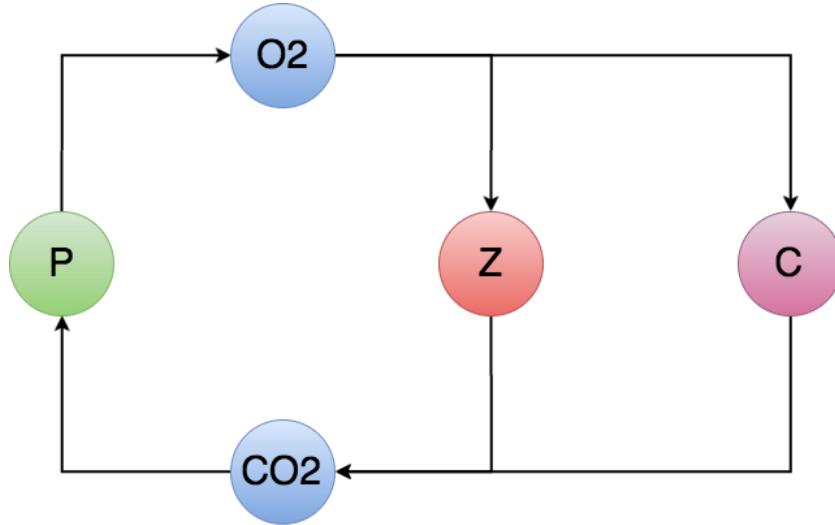


Figure 4.8: A closed cycle within the model

In attempting to reach this state of equilibrium, the rates of photosynthesis and respiration must be balanced out. The presence of randomness and noise within the parameters of the simulation makes this difficult to mechanically achieve. Moreover, the equations 4.4, 4.5 and 4.6 impact the dynamics of the gases due to the fact that the population sizes of organisms intrinsically impact the rates of photosynthesis and respiration. Although the actual amount of depletion is very much due to the randomly mutating value

of the encoding within genes, the mating and predation rates can impact the dynamics and can be somewhat controlled due to the open parameters dictating their likelihood at each evaluation. Indeed, we observe that if the effects of Selection and Predation is cumulatively larger than mating in 4.4, Phytoplankton will eventually go extinct. This gradual extinction will have a direct impact on the masses of the gases; Carbon Dioxide will gradually grow whilst Oxygen gradually depletes due to the imbalance of the terms in 4.7 and 4.8 as a result of less photosynthesis. This in turn may limit the success of Phytoplankton and Crustacean due to the absence of Oxygen being a limiting factor to their growth; Selection will eliminate these populations due to the lack of Oxygen, resulting in extinction.

An extinction event is not very useful to the running simulation; should the simulation be accelerated enough, the very nature of the system means that it is highly likely an extinction will occur. Rather than restart the simulation every time, the design will enforce a lower bound on any population size. This action should allow the dynamics of the system to gradually regain the equilibrium state in the face of an effective extinction.

This sequence of events highlights the intrinsic interconnectedness of the system. Specifically, the loss of Phytoplankton will eventually lead to a loss in Zooplankton and Crustacean. Moreover, this behaviour is typically expected of a marine ecosystem in which there exists some dynamics and a trophic food chain and is therefore a successful model of real life dynamics. Indeed, if the population of Zooplankton and Crustacean decrease, we expect the predation rate on Phytoplankton to fall. If the Phytoplankton population is not extinct, it may have a chance to grow again due to the smaller term in 4.4. This behaviour creates harmonic motion defined by the Lotka-Volterra equations [7] and illustrated in Figure 4.9.

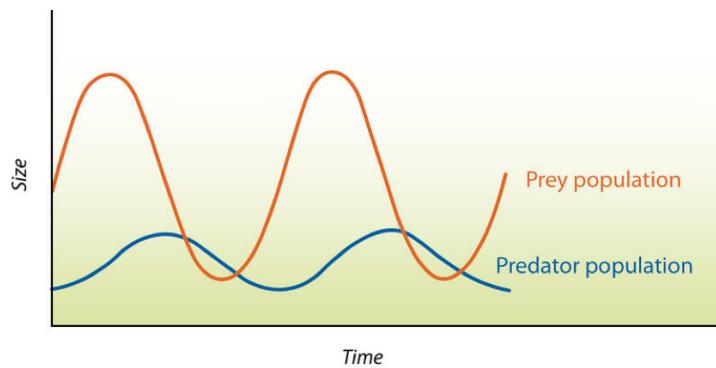


Figure 4.9: Harmonic motion of a predator-prey relationship

Another interesting feature of the dynamics results when population blooms occur. These may happen due to random mutation producing very fit alleles which quickly

disperse themselves into the gene pool through selection, and an increased mating rate due to greater individual genotype *Energy*, which was touched upon earlier. In particular, as the bloom occurs we see a higher domination of the corresponding life process for that species (photosynthesis or respiration) and hence the term in their respective equation will be greater. If this process is photosynthesis, we have a Phytoplankton bloom, and the level of Carbon Dioxide will gradually deplete and more Oxygen will be pumped into that zone. However, as the Carbon Dioxide is depleted and Phytoplankton compete with each other for this resource, it will become scarce in the zone to the point where it is too low for photosynthesis to occur, thus enabling Selection. As a result, a large proportion of the population is removed and the equilibrium is re-established. A knock off effect of this bloom is a short lived increase in the Oxygen level, which should in theory increase the Zooplankton and Crustacean population. This behaviour corresponds to the predator-prey relationship illustrated by Figure 4.9.

Applying selective pressures through an adjustment of abiotic factors is a key component of the simulation design. Such action will apply the pressures on the organisms it is involved in and ultimately impacts the terms in the equations described in Figure 4.7. The capability of the system to re-balance as it deviates away from the equilibrium is also a key feature of the design. This is achieved by preventing the extinction of species whilst preserving the capacity for population blooms; a user will still be able to recognise which alleles are the most successful simply by noting the clear difference in population sizes. Experimentation of various parameters which can impact the simulation are discussed later in the document.

4.6 System Architecture

The design has thus far been presented in an abstract sense without an architecture that could form the basis of the software implementation. Figure 4.10 illustrates the overall system architecture that encapsulates the simulation cycle and the model, along with an interactive user interface.

More formally, the architecture follows the Model-View-Controller design pattern with some modification. The View in this architecture is the UI that the user sees and renders each frame as a distinct time step in the simulation. This View is constantly updated from the simulation in each frame by the Ecosystem Model, which performs the simulation cycle on each zone in the ecosystem. The state of the zones that the model enacts upon is encapsulated in the Ecosystem Data storage, which at any one time

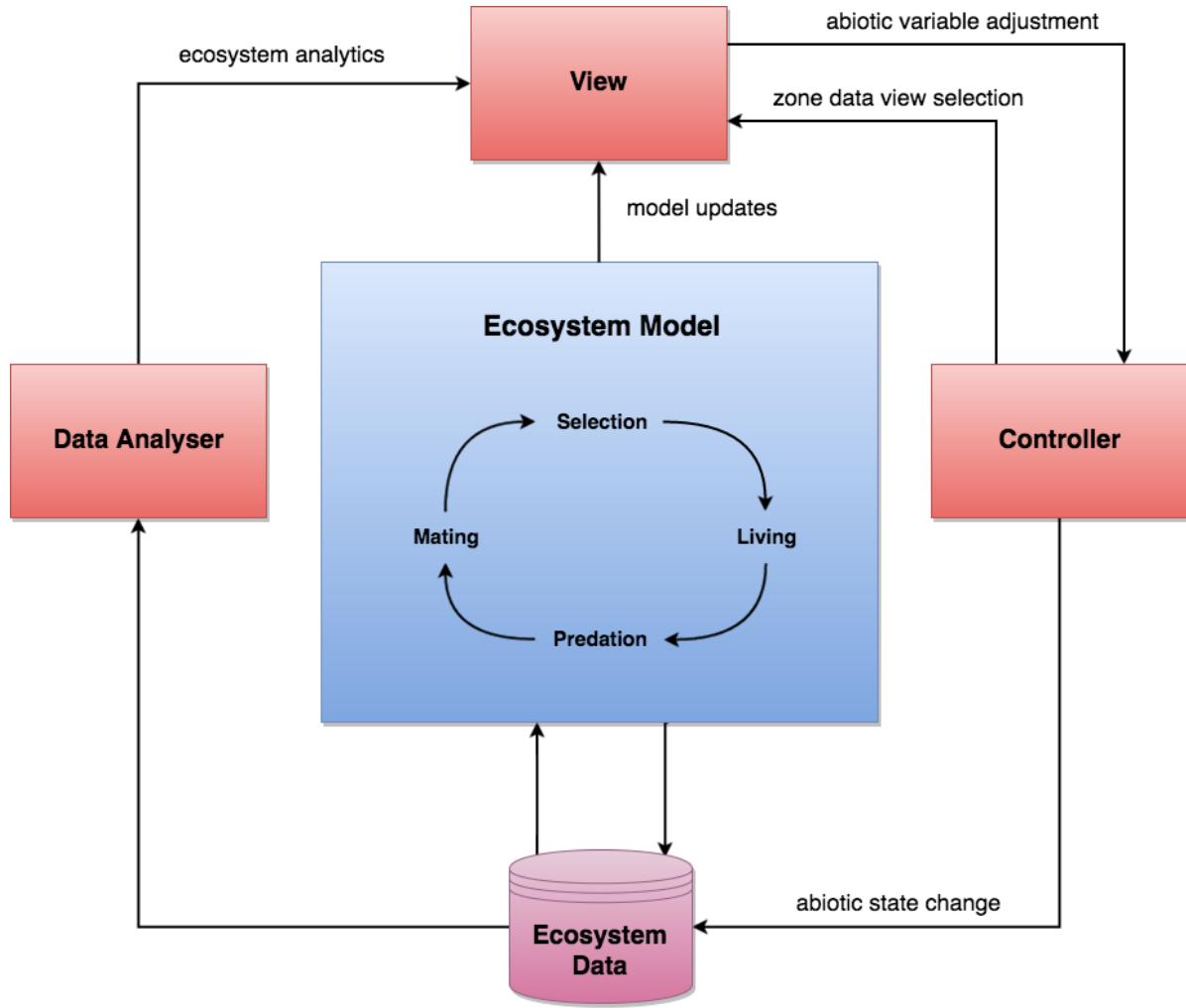


Figure 4.10: Simulation system architecture

stores the state of the entire ecosystem. Namely, it stores information pertaining to the abiotic features of every zone, the state and values of every gene, and the number and detail of every organism that encapsulates them. Given no interruptions, the simulation should be able to continuously run via this cycle as time progresses, updating the view with the new state of the ecosystem as altered by the model. Furthermore, every frame receives an update to the analytics window in the user interface which is independently fetched from the ecosystem data store. The data is analysed by the Data Analyser which performs sorting and interpretation in a way that makes it more useful and obvious to the user.

The user has the opportunity to adjust the abiotic variables of the zone they are considering. In doing this they will send a state change to the ecosystem data store, thus enacting the change. In the next time step, the ecosystem model will note the change and adjusting itself accordingly. The Controller also enables the switching between the

controls and analytics for different zones, providing greater feasibility to the user.

The design of the system is built around a discrete event simulation. In order to enact this, the ecosystem model constantly cycles, manipulating the data in the data store as it does. Our design effectively listens for events that can manipulate the data in the store, which can alter the model itself. A suitable graphics library with an object orientated back-end suitably implements this architecture, the documentation of which is featured in the next chapter.

4.7 Environment and User Interface Design

The user interface was designed with industry principles in mind, drawing on Shneiderman and Ben [43]’s principles and guidelines on user design.

We identify the users of the system to be *Expert Intermittent Users* [43]. In essence, users are expected to understand the fundamentals of evolutionary biology and ecology, but utilise this tool as a means of facilitating an understanding of it. The system also aims to provide greater analytical detail to users who have a thorough and grounded background knowledge of the subject domain. Given this, the design should facilitate the needs of the first type of user, while hiding away detail that can be unearthed easily for the more advanced one. The intermittent quality of the target user assumes a background in using graphical systems, and is also factored into the visual design.

4.7.1 World Design

Figure 4.11 illustrates the visual design, formulated to be implemented in a 2D graphics application. The camera in our world is positioned such that the user is given a cross-sectional view of a three dimensional model of the marine ecosystem, projected onto a two dimensional plane in order to reduce computational costs and make the simulation more visually intuitive. The height of the world will correspond to the depth of the ecosystem, providing a clear separation between each of the three zones in the model. A basic gradient effect will align the background to mirror the effect of oceanic light penetration.

Each organism in the simulation is modelled as an ellipse. Whilst this structure is not indicative of an actual marine organism, it makes our organisms differentiable and

simple to identify by the use of different colours and sizes between species, reducing computational overhead when rendering many thousands of these creatures. A basic design principle is to be adhered to in the models, and the size of each ellipse increases down the trophic food chain for increased differentiability.

- Phytoplankton - Green
- Zooplankton - Amber
- Crustacean - Dark Red

4.7.2 Analytics

An analytics window will be superimposed onto a small portion of the Abyssal Zone. This is initially hidden from the user and set to a toggle in order to hide detail that is not useful to more basic users of the system. In this sense, usability is preserved across the spectrum of expertise. Given this, the analytics can preserve fine detail pertaining to the simulation. The detail of the analytics is specified in **F17** of the Model Requirements in the System Requirements chapter.

4.7.3 Navigation and Controls

The window is kept non-interactive due to the complexity of the information on the screen. Therefore, navigation is kept isolated to the keyboard and mouse. More specifically, the user should be able to make abiotic, speed, toggle and zone switching adjustments via keys. This moves buttons away from the interface in order to free up space for the world to better occupy the window. These information pertaining to which keys to press are highlighted in plain white text on the screen; the bold colour stands out against a darker blue background, making it more bold and obvious.

The simulation will at any one time contain many elements on screen at once. During blooms, it is likely that the world space will be completely covered by many organisms, many of which will superimpose each other. Therefore, the system should allow the user to manipulate the camera of the world around ecosystem. More formally, the user should be able to use the mouse to pan and zoom within the marine world. This allows the user to identify a section of the ecosystem of interest and navigate closer to provide a better view of ecosystem interactions.

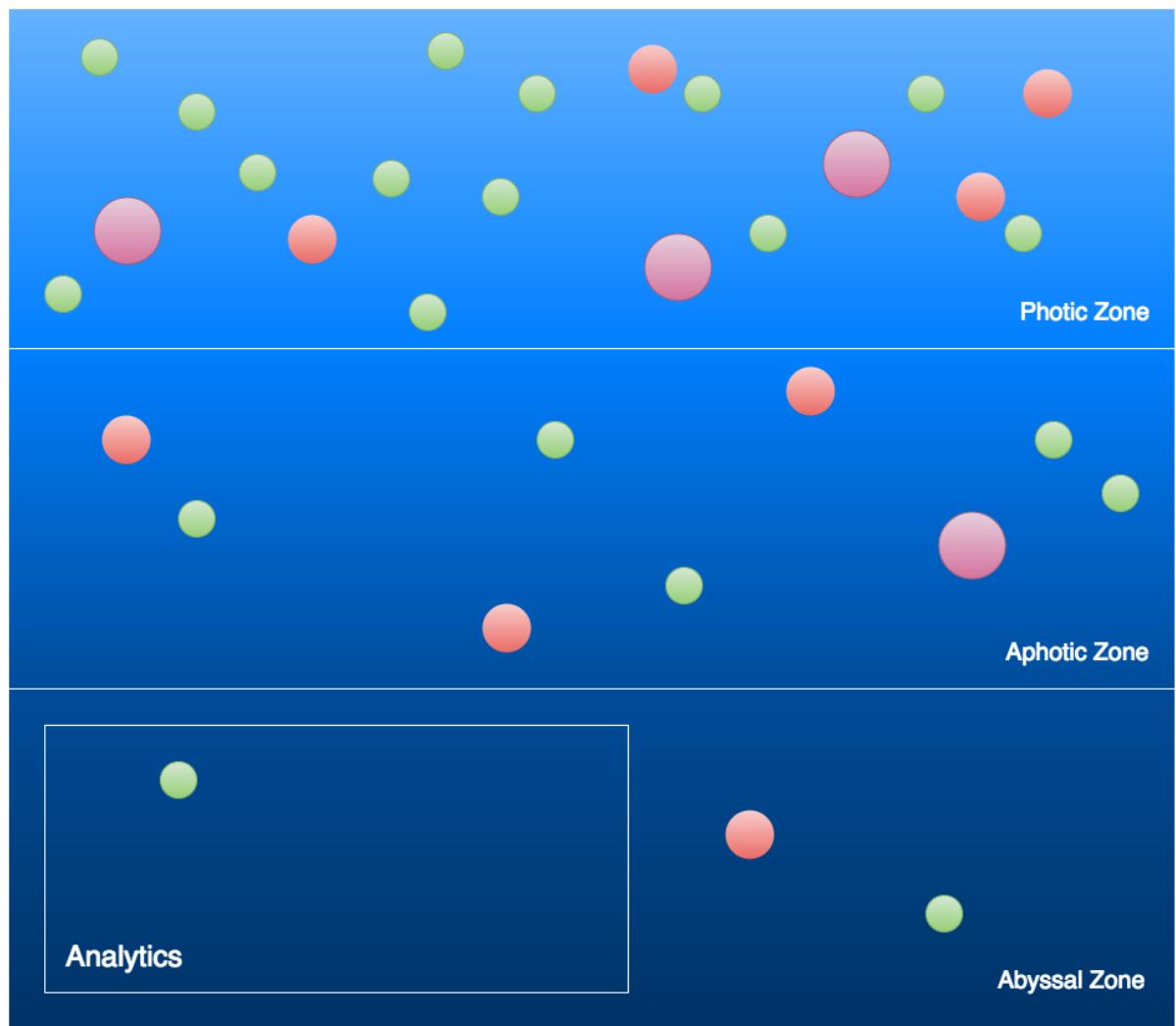


Figure 4.11: User Interface Design

Chapter 5

Implementation

This chapter describes how the conceptual model outlined in the previous section was implemented within software by adherence to the system design. Further, justification is given for the design of the implementation and for the choice of technologies used.

5.1 Technologies Used

As per the system requirements, the user is to be provided with a graphical interface which reflects the status of the model ecosystem. Given this, it is imperative to use a graphics application that could support the complexity of the simulation and successfully implement the user interface design outlined in the Design chapter.

Processing is a free and open source library which provides the functionality to build interactive graphics applications in 2D. Moreover, it supports an accelerated OpenGL framework, is well documented and offers cross platform support [21]. As a lightweight alternative to heavier graphics API's, Processing consumes far less computational power than some of its counterparts, a feature which supports a key non-functional requirement of the project. Crucially, it is written in Java, whose powerful object oriented foundation can easily support the depth and complexity of the simulation, as well as the architecture that was designed to encapsulate it.

The conventional means to developing a Processing application is to use the in built IDE, which is preloaded with Processing methods and classes, along with the libraries for use in compilation. That being said, it offers a limited development tool-set and was deemed unsuitable for the larger scale development required by the code. As such, an

eclipse based IDE named IntelliJ IDEA [6] was used for development.

In complement to Processing, additional libraries were used to add power and control to the simulation. The giCentre [4] provide a powerful extension to Processing such as the HashGrid and Zoomer. The former enables objects to be stored efficiently whilst preserving spatial location properties, whilst the latter provides pan and zoom functionality around the world through the mouse controls. These libraries supplemented the core program, the details of which are elaborated on in the remainder of this chapter.

5.2 Structural Design Implementation

The system makes effective use of the object orientated programming capabilities associated with Java in order to give representation to entities in the system. In particular, Genes and Organisms are represented as objects which are stored within data structures in each Zone. Further, the simulation will effectively interpret and adjust data associated with each of these objects in order to enact the cycle described in Figure 4.8.

This section will detail the class hierarchy and the resulting implementation of each class that builds up the static structure of the system outlined in the Design section.

5.2.1 Biome and Zone

Figure 5.1 describes a portion of the static class hierarchy. Specifically, the Zone class describes a section of our ecosystem as touched upon in the Design chapter, whilst Biome encapsulates the entire ecosystem. The classes effectively store information regarding the abiotic and biotic components of the ecosystem, bringing context and structure to the dynamic entities it contains. The Ecosystem class dually serves as the controller and the view of the system as described in Figure 4.10, and is described in greater detail further on in the chapter.

Zone

As explained by the Design section, the ecosystem is divided into three zones; the Photic, Aphotic and Abyssal zones, each of which is an instantiation of the Zone class. Each instantiation will store two Collections of all the Organisms that reside in the Zone,

each of which serves a different purpose. Crucially, we keep persistence between two data structures when changes are made to an Organism.

The first of these stores is a HashGrid [4], which provides efficient storage and retrieval of objects that have some spatial location within the world co-ordinate system of each Zone. By storing Organisms here, we can easily discern where an Organisms is in space and paint an ellipse at that location. Should an Organism move, the location data will change and the ellipse will reflect this accordingly. Moreover, the HashGrid can be used for collision detection by preventing movement outside of the space bounds assigned to it by the designer.

The Zone class also possess a Map containing every species as a key, where the value contains a List of all the organisms associated with that species. This structure is required to enforce the separation of species, an important feature when performing operations on the organisms. For instance, should we wish to perform an operation on all members of a species, we would have to sort through every element of the HashGrid data structure if this were the only storage facility. Given this clear inefficiency, the usage of a species data structure that works alongside the HashGrid is justified.

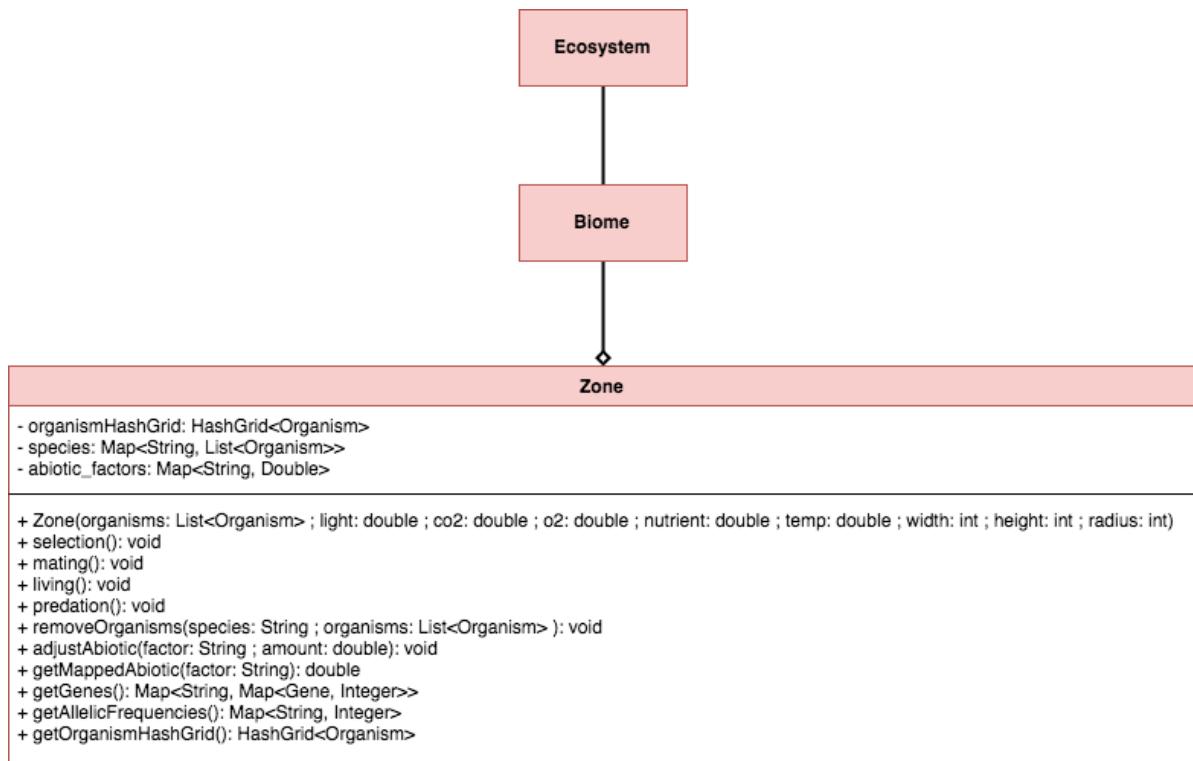


Figure 5.1: UML class hierarchy for Ecosystem, Biome and Zone

The Zone will also keep a Map of the abiotic factors, where the key is the name of the factor and it's present quantity is the value. This, alongside the Organism data stores

are used by the methods of Zone in order to perform the simulation cycle, manipulate them accordingly. As one can imagine, frequent access as every time step will be required of these data structures. As such, Maps were chosen given the efficiency of their access.

The *selection()* method follows the algorithm outlined in the Design chapter. Specifically, it will iterate through all Organisms, removing one from both data sets using the *removeOrganisms()* method if the ecosystem deems it unfit for survival 5.2.

Conversely, the *mating()* method is more involved; it follows the Design principle of creating a sublist of each species List randomly as a mating pool, dividing these lists into mother and father populations and iterating through them concurrently. Each one of these pairs are mated, and the lists of all offspring that are returned from Organism will be appropriately added to each Organism data structure.

The *living()* method builds on similar iteration principles as the previous two methods. Namely, we consider every organism, and depending on the nature of the organism, we run either it's respiratory or photosynthetic methods associated to the object, passing in the relevant abiotic factor by reference to the Map storing them. Crucially, we ensure that the abiotic factor that we extract from the map is constrained through the constraining function defined in 4.3 via the method *adjustAbiotic()* before we run the life function, ensuring it fits the domain of the gene function. Importantly still, we perform selection on each organism in question beforehand to ensure that it can survive with that factor. Moreover, should the organism return a value for their life process, we alter the abiotic factor Map with the relevant adjustment as returned by the Organism.

Finally, the *predation()* method performs the same iteration procedure, but only considers consumer species. The method will pass in the list of all possible prey to predation method found in each predatory Organism, which will return the list of prey it wants to kill. Given this, we remove every instance of prey from the data structures reflecting the predation.

The remaining methods found in Zone are used in the analytics implementation, and provide information regarding the Zone at any one time. In particular, *getGenes()* will iterate through the species Map and add every allele that it encounters. The frequencies are stored in a Map containing Gene names, whose associated value field contains a pair of alleles along with the frequency of the occurrence of them in the Zone. The *getAllelicFrequencies()* method fulfils it's criteria of providing the frequency of specific allele pair occurrences using similar intuition, and is also used in analytics. Accessor methods for each abiotic factor are also implemented.

Biome

The Biome class acts as a container for the every Zone instantiation in the ecosystem and is thus the first point of contact by the main system in conducting the simulation. Namely, it contains methods which iterate through each Zone as part of a greater call to run each step in the simulation cycle sequentially. *biomeSelection()* would be one of them, a method which iterates through every Zone contained in a field list, calling the *selection()* method in each Zone as it goes. The remaining cycle methods *biomeLiving()*, *biomePredation()* and *biomeMating()* follow similar intuition.

5.2.2 Chromosomes and Genes

Gene

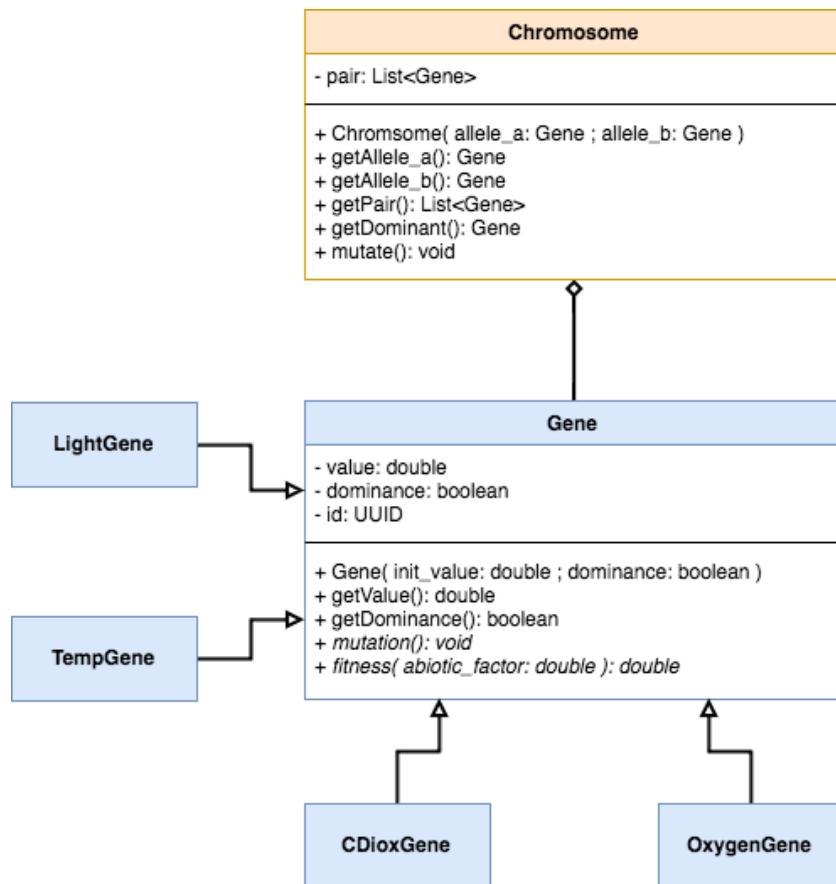


Figure 5.3: UML class hierarchy for Chromosome and Gene

The Gene is the most basic unit in the system. As detailed in the design section, they are carried by every organism that populates the theoretical physical space of the system

in their genotypes. Moreover, they are of four types:

- CDioxGene
- LightGene
- OxygenGene
- TempGene

Of these, the former two are carried by Phytoplankton and encode to perform photosynthesis, whilst the latter two are carried by Zooplankton achieving respiration. The implementation of genes will follow the design outlined in Design section, the fitness of which is formulated by the equation in 4.2.

The implementation includes an abstract Gene class which is sub-classed by LightGene, TempGene, CDioxGene and OxygenGene in order to group together common characteristics but separate specific differences in the sub-classes. Such a structure allows all Gene types to be grouped into the same specific Collection.

All genes will have a value with which describes the encoding (value), along with a description of it's dominance as an allele and a unique *ID*. Moreover, each sub-class must implement a *fitness(abiotic_factor)* method, which implements the the gene logistic equation described in 4.2. This method uses the field variable *value* as the θ term, along with a retrieved abiotic factor to determine a fitness derived from the equation. As an abstract method we allow for alterations of the gene logistic function in specific gene sub-classes should it be desired. For instance, alterations to the function may increase the fitness range that the function outputs.

The self explained *mutate()* method serves to displaces the the gene encoded value by a random amount to provide a mutation. A random displacement from the gene encoding is determined as a random number specified between $[-\text{mutation_rate}, \text{mutation_rate}]$, where the *mutation_rate* differs between different genes.

Chromosome

Each Gene object must form part of the Chromosome class in order to be understood by the mating mechanics of the system. This class simply holds the List of two alleles, making a pair. Alongside some obvious accessor and mutator methods, we can randomly select an allele to mutate through the *mutate()* method held in this class.

5.2.3 Organism

Organisms have already been discussed extensively in the Design section and their implementation requires an adherence to the model principles and use of the Gene and Chromosome classes. Much like the Gene hierarchy, the Organisms class in Figure 5.5 is abstract and groups common functionality across it's subclasses whilst also allowing generic Collections to be built consisting of different organisms.

In regards to the abstract Organism class, we define several field variables important to it's functionality. Specifically, a unique *ID* is defined along with a *colour* for the ellipse that will shade the Organism, as well as the PVectors defining the *location* and *movement*. A crucial field is the *genotype*, which will store the Chromosomes that make up the genetic description of the organism in a Map for efficient retrieval. An *energy* value stores a concept of organism health and is described in the Design section. Finally, a *mutation_rate* will define an upper bound for a probability with which to determine how frequently a mutation will occur within the genotype of the organism.

An Organism is instantiated with a random initial position and movement vector, and is given genetic information either from either parent genotype combinations or as raw chromosomes which are assembled in the constructors of the sub-classes. Furthermore, each organism is instantiated with a random position and movement vector, and is given the spatial bounds of the zone it is instantiated in. With this information, the organisms can detect the collision of the outbounds of the zone in the *move()* method. This method in particular is called at every time step and serves to change an organism position in the direction of the movement vector by some magnitude, but will reverse the direction of this vector when the expected position is to be outside the zone boundary.

Each sub-class will have a *mate(Organism father)* method which is called when the Mating step of the simulation cycle is called. This method will generate a random number between the *energy* of the organism and the *mating_proba* as defined in the field variable, where the former is set to the cumulative fitness of the genotype and the latter is experimentally determined as so is of a *final* type. Each offspring generated will call the generic *crossover()* method in the Organism class which performs the Mendelian crossover of two genotypes that are used to produce offspring, the mechanism of which is briefly explained in the Design section. This method will serve to create the genotypes of offspring by selecting random Punnett combinations of the parent Chromosomes, returning a new genotype map to the *mate* method in order to instantiate a new Organism offspring with it. This method in itself is called by Zone, and so the return value will be a list of Organisms which Zone will add to it's Organism data stores.

The *survival(abiotic_factors)* method is another abstract method that each sub-class must implement, and is used by Zone to determine whether an Organism will survive given the abiotic factor. Just as the design has already re-iterated, this method will unpack the *abiotic_factors* map passed in by Zone and use it to determine the fitness of the dominant allele in each Chromosome of genotype. The function will return true if each fitness value passes the survival threshold, which is stored as a *final* variable in each sub-class field.

In addition to the generic methods defined by an Organism, Phytoplankton contain a *photosynthesis(co2_intensity)* method, whilst Zooplankton and Crustacean contain a *respiration(o2_intensity)* one. Each method in turn will extract the relevant gene from the genotype field (Pco2 for photosynthesis and Po2 for respiration), extracting the dominant allele and returning the fitness to be used by Zone to augment and deplete the abiotic factors as per the design principles.

5.3 The Ecosystem Controller

The previous section detailed the hierarchy of the static structure of the system, along with the methods that work to mechanise the simulation steps. A controller interface is required to manage and manipulate this structure in order to run the simulation as described in the Design chapter. This controller is implemented within the Ecosystem class.

```
public static void main(String[] args) {PApplet.main(new String[]
 {"Ecosystem"});}
```

Figure 5.6: Code for the *main()* method

As an externally developed Processing application, the system must make use of the core library and as such is required extend the PApplet class. This enables the client class of our system (Ecosystem) to be recognised as a Processing Application.

As a PApplet, the Ecosystem class must adhere a structure that defines a Processing graphics application. In particular, every Processing application must implement *main()*, *settings()*, *setup()* and *draw()* methods.

The *main()* method points to where the Processing application should begin in Figure 5.6, whilst the *settings()* will define the window size along with a specification of the

type of graphics we wish to render (P2D in our case) in Figure 5.7. Similarly, the *setup()* method serves to define some initial environment properties in complement to *settings()*. Finally, the *draw()* method, called immediately after *setup()*, is where the bulk of the program exists. It continuously executes the lines of code contained inside its block until the program terminates. Crucially, every frame is redrawn at the end of this method.

5.3.1 Initial Population

Upon initialisation of the simulation, an initial population must be created that can populate the ecosystem. Indeed, we require every initial Gene, Chromosome and Zone to be instantiated before any real simulation can occur. We take advantage of the nature of the *setup()* method mentioned earlier by placing all of the initial instantiation within it in Figure 5.8.

Specifically, we begin by instantiating all the initial Genes in the ecosystem by providing them with a dominance characteristic and an initial gene encoding (θ), eventually encapsulating them into every possible Chromosome pair. A homozygous dominant, recessive and heterozygous pair for each Gene will be defined in their own respective Chromosomes.

Once the Chromosomes have been created, they will find themselves in an initial Organism pool. Furthermore, given enough initial variation such that their offspring can theoretically possess every allele combination in Figure 5.9. Finally, all that was required was for the organisms to be packed into a list pertaining to their respective Zones. Every Zone instantiation also requires an initial value for each abiotic factor. Once they have been shuffled, a Biome object is instantiated that is associated with all of the three Zones we have just defined in Figure 5.10.

5.3.2 Running the Simulation

Given the creation of the Biome now containing the initial population, we can begin the actual simulation in the *draw()* method.

Initially, the organisms are retrieved from each Zone of the Biome and are drawn to the screen as per the position and state that were in the time step prior. In principle, this involves retrieving every Zone associated HashGrid from the Biome object, performing

an iteration over it and drawing an ellipse at the position retrieved. Once every Organism has been considered, they are moved to their next designated position using the *move()* method associated with each Organism. This process is detailed in Figure 5.11, which illustrates the method called by *draw()* to paint the Organisms in the Photic Zone using the method just described. Crucially, the Aphotic and Abyssal zones will have to have each ellipse drawn to be translated down the y axis by a certain amount, such that it falls within the boundaries of the zone it is meant to be in. This is easily achieved by pushing and popping the transformation matrix prior to drawing the ellipse, the code of which lives in related methods.

Following this, the simulation cycle is performed on the very same Organism pool, the effects of which will be reflected in the visual representation of them in the next call to *draw()*. In essence, this simply involves a sequential call to each simulation step method defined in Biome. This will perform the functionality of the simulation internally, changing the state of the Biome which will be visually reflected by the *draw()* function when it loops back again to draw the Organisms. In particular, Organisms will move, appear and disappear at every step due to normal movement, mating and predation/selection. Finally, we can assign a delay at the end of the function to slow or speed up the simulation.

5.3.3 Analytics and User Input

The final function of the continuously executing *draw()* method is to display the analytics window as a reflection of the internal state of each zone. This is achieved by reference to the *getGenes()* method of Zone, which will return a Map containing information about the frequency of each Gene in the ecosystem. Using this data structure, along with other retrieval methods in Zone, we print information to the screen regarding each Zone. Moreover, the specific nature of the information is highlighted in the Functional Requirements and the UI Design. As characteristic of the *draw()* function, the analytics will be updated at every frame refresh to reflect the new state of the ecosystem at every time step.

Finally, user input is achieved using conventional Processing methods, and serve to either change the state of the ecosystem or control the display of the window. Specifically, the key controls are as follows:

- **i** - toggle info

- **z** - switch between zones
- **s** - speed up
- **S** - slow down
- **c** - reduce CO₂
- **C** - increase CO₂
- **o** - reduce O₂
- **O** - increase O₂

The **i** keys will hide or display the analytics, allowing a user to select a preference of view. The **z** key will increment a pointer which is referenced by the analytics display method to switch between the zones. The remainder of the keys are self explanatory and result in the an adjustment to the designated abiotic factor for the zone in consideration, as determined by the pointer. Moreover, the effects of this are explained in the Design chapter extensively.

Finally, as touched upon earlier, the ZoomPan method *transform()* will sit in the *draw* method. Depending on the user input to the mouse, it will transform the entire ecosystem to reflect the camera position. By placing this call at the beginning of the *draw()* method we ensure that glitches in transformation are avoided and the graphics appear smooth.

```
public void selection() {
    /*
     Selection function
     Iterates through species list, if an organism does not survival
         remove it
    */

    // iterate through organisms list of every species and check
    // survival
    Iterator species = this.species.entrySet().iterator();
    while (species.hasNext()) {

        Map.Entry<String, List<Organism>> next_species = (Map.Entry)
            species.next();
        List<Organism> species_population = next_species.getValue();

        // iterate though each organism in species list
        for (Iterator<Organism> iter =
            species_population.listIterator(); iter.hasNext(); ) {
            Organism organism = iter.next();

            // cap the population size to prevent extinction - only
            // species five organisms can be selected
            if ( (!organism.survival(getMappedAbioticAll())) &&
                (species_population.size() > 5) ) {

                // to avoid ConcurrentModificationException do
                // iter.remove()
                iter.remove();
                removeOrganism(next_species.getKey(), organism);
            }
        }
    }
}
```

Figure 5.2: Code for the selection function

```
public double fitness(double abiotic_factor) {
    /*
    The logistic function gene function is calculated here based on
    the abiotic factor in question and the value

    @param abiotic_factor - the x value for the function fetched from
        the ecosystem
    */
    double theta = this.value;

    // logistic function (see documentation)
    double fitness = 1/(1 + Math.exp(-theta*abiotic_factor));

    return fitness;
}
```

Figure 5.4: Code for the fitness function

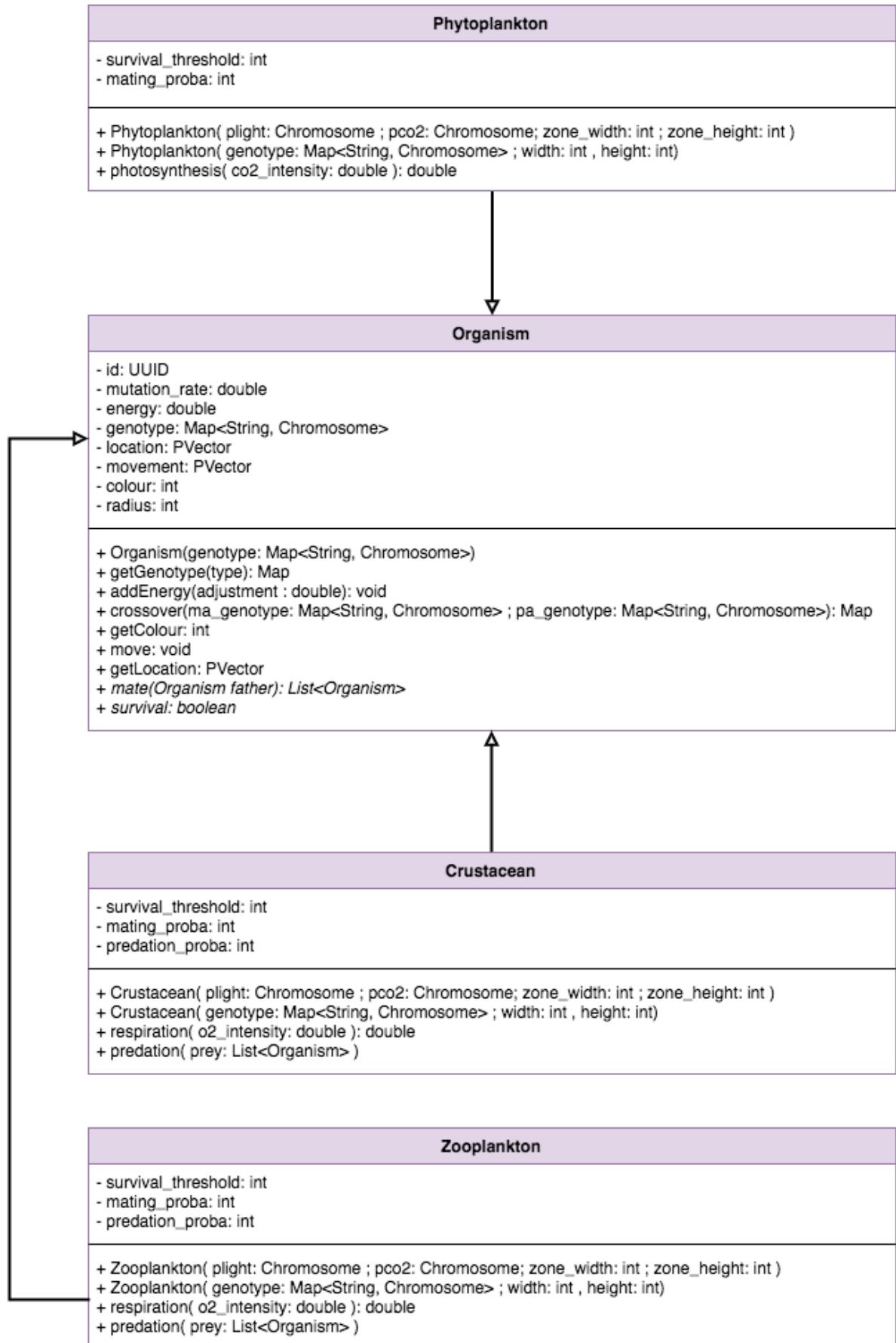


Figure 5.5: UML class hierarchy for Organisms

```
public void settings() {
    /*
    Processing function defining initial settings such as window size
    and ZoomPan instance
    */
    size(1024, 700, P2D);
    zoomer = new ZoomPan(this);
    smooth(3);
}
```

Figure 5.7: Code for the *settings()* method

```

// Define initial genes with pre-determined values. One dominant and
// one recessive
Gene plight_dominant = new LightGene(0.8, true);
Gene plight_recessive = new LightGene(0.3, false);

Gene pco2_dominant = new CDioxGene(0.8, true);
Gene pco2_recessive = new CDioxGene(0.3, false);

Gene ro2_dominant = new OxygenGene(0.8, true);
Gene ro2_recessive = new OxygenGene(0.3, false);

Gene rtemp_dominant = new TempGene(0.7, true);
Gene rtemp_recessive = new TempGene(0.2, false);

// Define all possible Chromosomes for population by combining alleles
// plight (LightGene)
Chromosome plight_hom_dom = new Chromosome(plight_dominant,
    plight_dominant);
Chromosome plight_het_dom = new Chromosome(plight_dominant,
    plight_recessive);
Chromosome plight_hom_rec = new Chromosome(plight_recessive,
    plight_recessive);
// pco2 (CDioxGene)
Chromosome pco2_hom_dom = new Chromosome(pco2_dominant,
    pco2_dominant);
Chromosome pco2_het_dom = new Chromosome(pco2_dominant,
    pco2_recessive);
Chromosome pco2_hom_rec = new Chromosome(pco2_recessive,
    pco2_recessive);
// ro2 (OxygenGene)
Chromosome ro2_hom_dom = new Chromosome(ro2_dominant, ro2_dominant);
Chromosome ro2_het_dom = new Chromosome(ro2_dominant, ro2_recessive);
Chromosome ro2_hom_rec = new Chromosome(ro2_recessive, ro2_recessive);
// rtemp (TempGene)
Chromosome rtemp_hom_dom = new Chromosome(rtemp_dominant,
    rtemp_dominant);
Chromosome rtemp_het_dom = new Chromosome(rtemp_dominant,
    rtemp_recessive);
Chromosome rtemp_het_rec = new Chromosome(rtemp_recessive,
    rtemp_recessive);

```

Figure 5.8: Instantiation of simulation objects

```
// APHOTIC Phytoplankton Population
for (int a = 0; a < 5; a++) {
    aphotic_population.add(new Phytoplankton(plight_hom_dom,
        pco2_hom_dom, width, zone_height));
}
for (int b = 0; b < 3; b++) {
    aphotic_population.add(new Phytoplankton(plight_het_dom,
        pco2_het_dom, width, zone_height));
}
for (int c = 0; c < 2; c++) {
    aphotic_population.add(new Phytoplankton(plight_hom_rec,
        pco2_hom_rec, width, zone_height));
}
```

Figure 5.9: Initial Organism instantiation for the Aphotic Zone

```
// Create zones
photic = new Zone(photic_population, 10.0, 40.0, 40.0, 5.5, width,
    300, RADIUS);
aphotic = new Zone(aphotic_population, -3.0, 40.0, 40.0, 5.5, width,
    300, RADIUS);
abyssal = new Zone(abyssal_population, -10, 10, 10, 10, width, 300,
    RADIUS);

// Add newly populated zones to the Biome
zones.add(photic);
zones.add(aphotic);
zones.add(abyssal);

// Create Biome
marine = new Biome(zones);
```

Figure 5.10: Creation of the Biome

```
public void drawPhoticOrganisms() {

    // PHOTIC zone
    // retrieve photic_hash_grid
    HashGrid<Organism> photic_hash_grid = marine.getPhoticHashGrid();

    for (Organism org : photic_hash_grid) {
        fill(org.getColour());

        if (org instanceof Phytoplankton) {
            ellipse(org.getLocation().x, org.getLocation().y, RADIUS,
                    RADIUS);
        } else if (org instanceof Zooplankton){
            ellipse(org.getLocation().x, org.getLocation().y,
                    RADIUS+5, RADIUS+5);
        } else {
            ellipse(org.getLocation().x, org.getLocation().y,
                    RADIUS+8, RADIUS+8);
        }
    }

    for (Organism organism : photic_hash_grid) {
        organism.move();
    }
}
```

Figure 5.11: Function for displaying analytics

Chapter 6

Validation, Verification and Testing

This chapter outlines the evaluation of correctness and validity of the simulation model determined with respect to the original purpose. More formally, the conceptual model is validated with respect to the system theories it is based on. Following this, the computer model developed to implement the conceptual one undergoes verification with respect to the modelling objectives. Finally the operation of the computer model is validated by assessing the output of the system and comparing it against the expected output with respect to the system theory and background.

6.0.1 Conceptual Model Validation

The overall conceptual model is explicitly described in the Design section as an implementation of the model requirements outlined in the System Requirements section, where numerous component models make up the overall system. As Sargent [40] importantly points out, any conceptual model built to represent a system must undergo some determination of correctness based on the real world problem it represents. More formally, Sargent [40] defines conceptual model validation as “*determining that the theories and assumptions underlying the conceptual model are correct and that the model representation of the problem entity is reasonable for the intended purpose of the model*”.

The simulation models developed for this project involve an abstraction of real ecological and biological phenomena, making it difficult to measure the correctness of the model against real world data. Rykiel [39] acknowledges the difficulty in validating ecological models with respect to the level of abstraction typically required by them. Specifically, he notes that detail will inevitably be left out of the model due to complexity. Clearly,

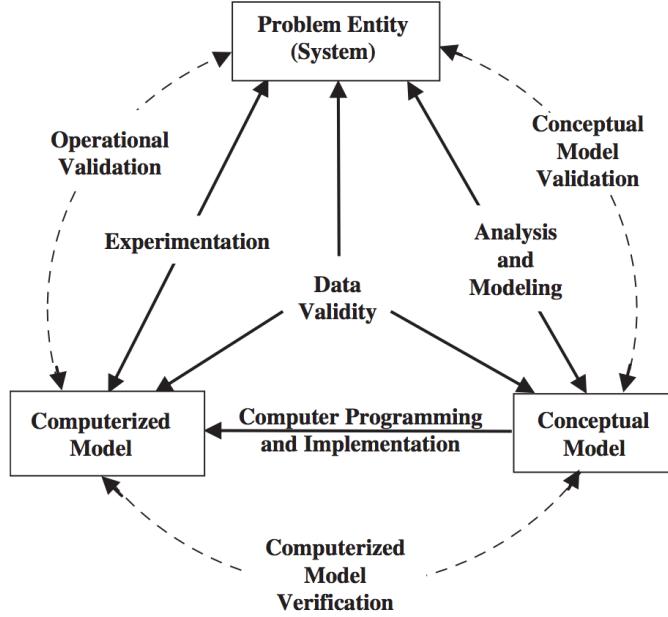


Figure 6.1: Simplified version of the model development process [40]

this practice will not always lead to invalidity, as justification can be provided for the simplification of known processes. Moreover, providing such assurance can establish conceptual validity. Alternatively, Rykiel [39] states that “*Conceptual validity depends on providing a scientifically acceptable explanation of the cause-effect relationships included in the model*”.

The simplifying assumptions made by our conceptual model are not only made to reduce complexity. Rather, they are made to highlight the presence of dynamical relationships in ecosystems, and how processes such as mating, predation and natural selection can influence the dynamics of this system. As per Rykiel [39]’s recommendation, we aim to justify the simplifications made by the conceptual model as a means of establishing validity.

Face validation, as suggested by Sargent [40], involves presenting the conceptual model to domain experts. Such experts can provide assurance that the theories and assumptions that underline the model are scientifically correct, and that the simplifications made were justified. Motivated to establish conceptual validity, Dr Sara Kalvala (an associate professor at the University of Warwick and a specialist in computational biology) was presented with a structured walkthrough of the conceptual model along with the finalised system. Positive feedback included a conformation that the conceptual model was scientifically valid. Furthermore, a motivation was expressed by her to use the simulation as a tool to aid in the teaching of ecological modelling, thus fulfilling one

of the core aims of the project and providing further validation.

6.1 Verification and Testing

Figure 6.1 illustrates the model development process, a structure that has been adhered to in the development of this project. Sargent [40] states that once a computer implementation of the conceptual model has been created, it must be verified by ensuring that “*the computer programming and implementation of the conceptual model are correct*”. Before this can be done, assurance of the software implementation must come from testing.

Given that the system was developed around an incremental build methodology [24], tests could be easily integrated within component development cycles. Moreover, as development took shape in phases, where smaller components were built first, testing took a similar approach. Specifically, the irreducible components of the system that were paramount to the overall functionality underwent unit testing. This ensured that individual modules behaved as expected when isolated from the system, and was particularly important given the that the system relies on numerical precision. Once these units were declared functional, integration testing assessed the performance of individual units working together. Indeed, once integration tests had passed, the next development phase would occur as per the incremental methodology. Finally, systems testing evaluated the entire system with respect to the requirements outlined in the System Specification. Crucially, each test is evaluated with respect to the System Requirements, the fulfillment of which denotes some measure of success and provides verification of the computer model with respect to the conceptual one.

6.1.1 Unit Testing

This section denotes the unit tests that were performed on each of the simple components of the system, namely Gene and Organism. Whilst not exhaustive, the tests presented were crucial to ensuring the system performed effectively, hence justifying their inclusion.

Gene

The Gene class can be seen as the simplest component of the system. The implementation for the genetic logistic function outlined in Figure 4.3 is found here, the correctness of which is crucial to the functioning of the Selection and Living steps of the simulation cycle. Unit testing was performed on this class and its subclasses, presented in Table 6.1.

Organism

The Organism class contains methods which enable much of the processes crucial to the system, and encapsulate the Gene class in doing so. Given this, the unit tests performed on these modules were performed after the Gene tests had passed, and can therefore be seen as an integration test of similar modules. Table 6.2 outlines this procedure.

6.1.2 Component Testing

The unit testing provided assurance that the basic static units of the system were ready to be integrated into the larger dynamic system; the Zone. Here, Organisms (and hence Genes) are grouped into data structures which are manipulated by the simulation step methods. Given that such methods involved multiple components coming together, we can consider this component testing, which had to be performed before any other stage could commence. Table 6.3 presents such procedures.

6.1.3 Integration Testing

The interface (The Ecosystem class - a system component in itself) integrates with the back-end, the bridge being the Biome class which manages each Zone. The marriage of these crucial system components required integration testing once the unit and component testing had been performed for each integration component. Unit and component testing has already been performed before the interface was developed. Crucially, the majority of the tests that involved the interface were considered integration testing in themselves, given that the interface required processed data from the back-end in order to function. This being said, tests were performed on individual units, the results of which can be inferred from the integration tests documented in Table 6.4.

Table 6.1: Unit tests on the Gene class

F#	Description	Input	Expected Output	Actual Output	Result
F4	The fitness method implements the logistic gene function in Figure 4.3 correctly, given a value of θ	An abiotic factor within a domain of [5, 5]	A fitness value within a range of [0, 1] consistent with the output of the function	A fitness value equal to the output of the function	PASS
F4	The mutate method produces a random double within a specified range which displaces θ by some amount	A call to the method	An average displacement within the range specified to θ	Successfully displaced θ by some amount within the range specified	PASS

F#	Description	Input	Expected Output	Actual Output	Result
F5/9	Zooplankton and Crustacean can perform predation on a random number of prey	A List of Organisms consistent with the type of prey suitable	A List of Organisms which is smaller than the input List	A List of Organisms which contains fewer organisms than before, indicating successful predation	PASS

F5/9	Predation success determined by a random number generated between the organism energy and the predation probability	Values for energy and predation probability	Increasing each input will increase the rate of predation, on average	The average predation rate increased with an increase in the input rate	PASS
F7	A Map of Chromosomes representing the genotype has accessor and hence mutator functionality	A request for a Gene responding to the request	A dominant Gene corresponding to the request	Correct genes returned, of the dominant type	PASS
F6	Respiration will provide a performance retrieved from the dominant Ro2 gene in the genotype, incrementing energy in the process	A value denoting Carbon Dioxide intensity as an input of the gene function	A performance correlating to the fitness of Ro2. The energy increases correspondingly	The value returned matches Ro2 fitness, which is added to the organisms energy	PASS

F6	Photosynthesis will provide a performance retrieved from the dominant Pco2 gene in the genotype, incrementing energy in the process	A value denoting Oxygen intensity as an input of the gene function	A performance correlating to the fitness of Pco2. The energy increases correspondingly	The value returned matches Pco2 fitness, which is added to the organisms energy	PASS
F7	Mating between two Organisms will produce new offspring	An Organism with which to mate	A List of Organisms, each of which contains genotypes as a combination of parent ones	Successfully generated offspring which contain parent genes	PASS
F7	Mating success determined by a random number generated between the organism energy and the mating probability	Values for energy and mating probability	Increasing each input will increase the rate of mating, on average	The average mating rate increased with an increase in the input rate	PASS
F7	Crossover will generate a genotype created as a combination of parent genotypes, in which the genes may have been mutated	Mother and Father genotypes	A genotype where Chromosomes consist of combinations of parent genotypes, the values of which may be different	Successfully generated predictable combinatorial genotypes with occasional mutation footprint	PASS

F8	Survival can be evaluated by assessing genotype fitness with respect an an abiotic factor	A Map of abiotic factors	A Boolean correctly evaluating survival based on genotype fitness	Successfully evaluated fitness of all genes, returning True when all genes passed survival threshold	PASS
F13	An organism can move within the direction of a specified vector by some velocity, abiding by collision boundaries	A PVector representing placement in a 2D space, and another one representing movement	Organism should move in direction, reversing course when out of bounds	Organisms fully adhered to collision boundaries	PASS

Table 6.2: Unit tests on the Organism class

F#	Description	Input	Expected Output	Actual Output	Result
F4	The storage, addition and removal of Organisms in a HashGrid and a Species Map	Query of Organism(s)	Retrieval of Organism(s) from stores	Successfully able to remove, add and store Organism in both data stores, keeping consistency between them	PASS

F3	Representation and adjustment capability of all abiotic factors in a Map	Query of abiotic factor	Retrieval of abiotic factor from data store	Successfully able to adjust abiotic factor as specified from the Map	PASS
F3	Mapping of abiotic factor through mapping function (Figure 4.4)	A (raw) abiotic factor within a domain of $[-x, x]$	A mapped abiotic factor within a range of $[-5,5]$	Successfully mapped abiotic factors through to the domain of the gene function	PASS
F10	Selection on population through survival method in Organism	Map of Species	Organisms removed from population if survival unsuccessful	Successfully removed Organisms which failed the selection criteria	PASS
F6	Organisms perform their life process if they can survive Selection, adjusting abiotic variables accordingly	Map of Species	Dynamic abiotic factors fluctuate, and Organisms removed from population if survival unsuccessful	Abiotic factors adjusted accordingly as per life process, with Selection removing Organisms appropriately as abiotic factors deplete	PASS

F5/9	Predation on population from predatory population through predation method in Organism	Map of prey Species	Prey removed from population if predation successful	Successfully removed Organisms when predation was successful	PASS
F8	All Organisms within a mating pool can mate with each other, adding to population	Map of Species	Offspring adding to population if mating successful	Successfully added Organisms when mating was successful	PASS
F17	Retrieval of the frequency of each Gene in the population	Map of Species	Map containing Genes along with the frequency of it's occurrence	Successfully retrieved Gene map, where frequencies are correctly determined	PASS

Table 6.3: Component tests on the Zone class

Description	Input	Expected Output	Actual Output	Result
Retrieval and drawing of Organisms from data store	Query for Organism HashGrid	Ellipses drawn at each Organism location	Organisms drawn at designated place given stored co-ordinates	PASS

Re-draw of Organisms at each time step	Query for Organism HashGrid	Ellipses moved at each frame refresh	Organisms disappear and appear as per movement, predation/selection and mating mechanisms	PASS
Organisms in Aphotic and Abyssal Zones translated down the screen as per ecosystem depth	Translation of ellipse corresponding to depth	Ellipse translated into correct Zone	All Organisms placed in correct locations after translation	PASS
Analytics reflects true Zone data	Query for Zone data	Zone data is displayed on screen correctly	The analytics window is consistent with Zone data	PASS
User adjustment to abiotic factors propagate to Zones	User adjusts an abiotic factor	Zone reflects change both in analytics and behaviour	Adjusting abiotic factors effects population sizes and is reflected by analytics	PASS

Table 6.4: Integration tests on the System Components

6.1.4 System Testing

In compliance with testing standards, systems testing was performed following the success of unit and integration testing. This tested the system as a whole with respect to the requirements, taking a black box approach. Specifically, this meant that testing occurred without consideration of the inner workings of the system, such that it could be evaluated from a different perspective.

In an initial testing period, the developer would test the system using a black box approach, exhausting all possible use cases and isolating the system until errors were shown. Thankfully, the success of the unit and integration tests meant that bugs were relatively insignificant and semantic, and were quickly corrected as a result. Finally, in an effort to remove bias, beta testers were employed to test the full system who possessed domain knowledge, evaluating the performance and usability of the system. Feedback was relatively positive, and adjustments were made where suitable. It is worth noting that this testing was not to prove model validity, but rather the robustness of the software implementation.

6.1.5 User Acceptance Testing

As part of the overall validation and testing procedure, it was important to the success of the project to test the system with students who made up the target demographic of this educational software. Using similar black box means as in the systems testing, A-Level Biology students were given the system to play with in order to assess their feedback and inform the validity of the system.

The overall feedback was positive; users found the system to be intuitive and clear, with significant detail and user input. However, common complaints were of the usability of the system, alongside the depth of the analytics. As a result, the Implementation chapter outlines the final iteration of an incrementally built software that took into account this user feedback. The testers reacted positively to changes, providing some sense of model validation.

6.2 Operational Validation, Results and Tuning

Following the verification of the software through rigorous testing, the operational validity of the simulation model had to be established. Sargent [40] describes operational validation as “*determining whether the simulation models output behaviour has the accuracy required for the models intended purpose over the domain of the models intended applicability*”. Indeed, a simulation may be completely tested and verified in software, but the model output may be inconsistent with what we expect. Moreover, Sargent [40] stresses the need to collect output data from the simulation and compare them to expected results. Given this, output graphs were collected that displayed quantities over time for each Zone in the ecosystem.

The expectations of model behaviour can be inferred from the equations defined in Figure 4.7, where the inter-connected relationship of all the entities in the model drive the dynamics of the system. Whilst the expected behaviour may seem predictable, there exists some crucial unpredictability, in that random variables are periodically generated that influence the rates of the processes defined in these equations. Specifically, this is the:

- Mating Rate
- Predation Rate
- Mutation Rate

which serve to constrain the probabilities generated that influence these rates, having already been discussed in this document in different contexts. Furthermore, they can be viewed as parameters which influence the behaviour of the model, the tuning of which occurred during the systems testing phase until a group of acceptable values were determined that seemed to provide good performance. In this context good is seen as behaviour that is expected (given the model) and correlates with data collected in the Research and from experimental sources. Specifically, Wasmund and Uhlig [47] and Gasiūnaitė et al. [23] provide data on marine population fluctuations in the Baltic Sea, which were also used as basis for validation.

Both logical inference and reference to literature suggests that the mating rate should exceed the predation rate for Phytoplankton in order to mirror a typical pyramid of biomass (Figure 2.4). The initial population chosen for every Zone will reflect this pyramid as best as possible. Furthermore, the Crustacean population is modelled after

a consumer on the extrema of a food chain. Typically, these consumers will have a smaller mating rate as compared to species lower down the chain, and is hence reflected in its own mating rate. The mutation rate is uniform across all species, and is chosen to regulate the speed of natural selection.

The initial population must be populated with Genes. Specifically, two alleles of each gene will be associated to each Chromosome that makes up the genome of every species. By encoding a value for each dominant allele such that the logistic gene curve is steep, we can ensure that the initial genes are very fit for their purpose when the abiotic factor is high. Conversely, every recessive allele is encoded as a steeper curve for low abiotic factors. This enables Selection to favour homozygous recessive organisms when the abiotic factor suddenly falls, similar to the allele illustrated in Figure 4.3. In doing this, we provide room for natural selection.

The remainder of this section will discuss the results collected from each Zone after running the simulation for some time, and explain how they serve to provide operational validation. Finally, each test was performed starting from the very same initial population, and the very same parameter rates to keep consistency in experimentation.

6.2.1 Photic Zone

As discussed in the research, the Photic Zone is defined by the abundance of light that readily enables photosynthesis. As such, a large Phytoplankton population supports this Zone, with regular additions of atmospheric Carbon Dioxide to support it. Correspondingly, the initial population reflects this ratio, with Phytoplankton being large as compared to the consumers. Figure 6.2 illustrates a test that was performed on the Photic Zone for around 500 time steps. Note that the *y – axis* represents standardised quantities; the Phytoplankton population has a high density compared to others, which may not be apparent in the graph. Rather, this graph is meant to illustrate the dynamics of the Zone.

As time increases, we see Phytoplankton undergo heavy fluctuation, presumably due to the variability of having a high mating rate. Moreover, we can attribute Phytoplankton blooms to a number of causes. For instance, pure chance may cause a sudden increase in mating, most likely due to the mating pool being large for one time step, leading to exponential growth. Alternatively, a random mutation in the genes that regulate Photosynthesis may propagate through the gene pool very quickly, increasing the mating rate due to an average energy increase in the species. Finally, a direct increase in Carbon

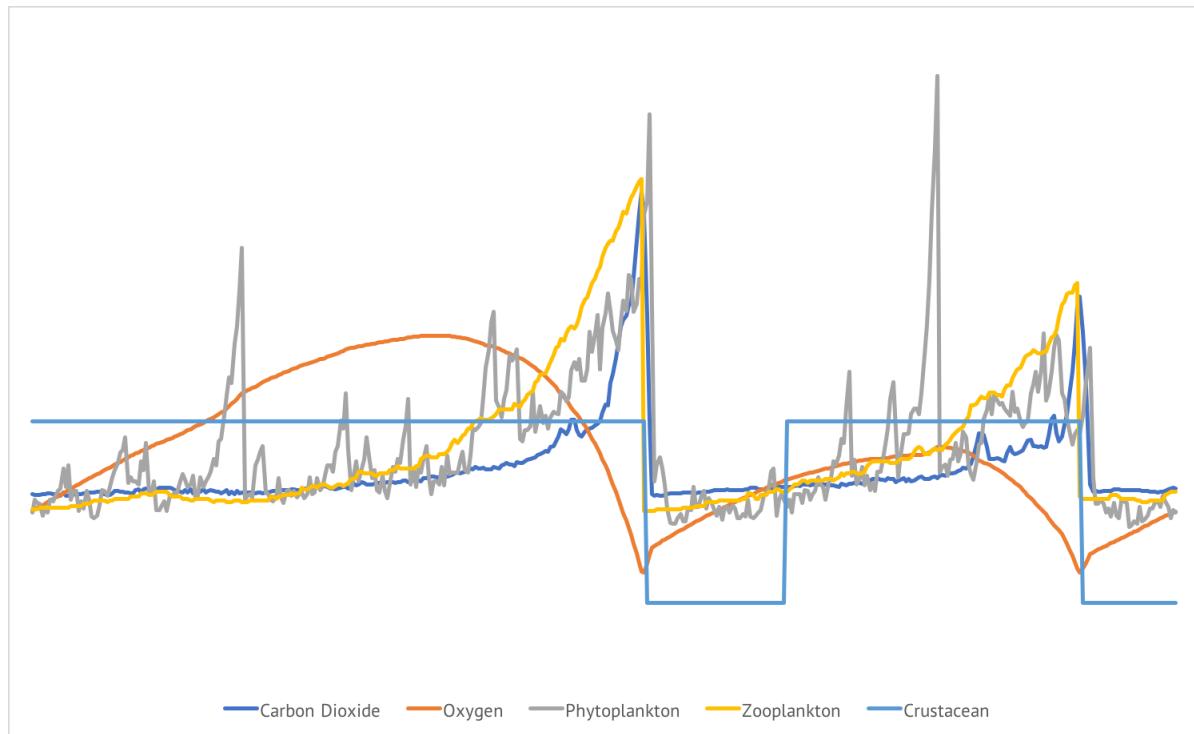


Figure 6.2: Data collected from the Photic Zone

Dioxide will lead to a proportional increase in Phytoplankton, as selection will lessen due to the abundance of the factor. Equally, every bloom will lead to a crash when selection takes place soon thereafter.

Leading on from this, we can explicitly see this effect in Figure 6.2. Specifically, Zooplankton experience growth due to the initial abundance of Oxygen. As this occurs, the Carbon Dioxide level grows as the rate of respiration increases, leading to a proportional increase in Phytoplankton. Similarly, the rapid growth of Zooplankton will leads to a gradual depletion of Oxygen; this factor does not experience continual replacement as seen in Carbon Dioxide. Effectively, a point is reached where the contribution that Phytoplankton makes to Oxygen level in the zone is less than the depletion from the growing Zooplankton population, resulting in a net decrease. This behaviour is expected of the model, and can be proven by observing the details of the equations in Figure 4.7.

As the populations grow, selection enacts when the effective carrying capacity has been reached; there is not enough Oxygen or Carbon Dioxide to sustain the massive population any further. The population will pick up again gradually and result in another bloom soon thereafter, reflective of the expected behaviour of the model.

6.2.2 Aphotic and Abyssal Zones

The Aphotic Zone has purposely very little light present to mirror the conditions of this depths in the marine ecosystem, showing very interesting characteristics as a result. For reference, Figure 6.3 shows the relationship between Phytoplankton and Carbon Dioxide in this zone, which consists of blooms of Phytoplankton, which can be attributed to many factors.

The Phytoplankton population is typically small when it is not in bloom here as a consequence of the lack of light. As a result, Carbon Dioxide is consistently augmented by the consumer population, which itself sees steady growth. The growing abundance of Carbon Dioxide means that it is no longer a limiting factor for photosynthesis, and when a random mutation produces a gene that can withstand the poor light, it will propagate through the population, leading to Phytoplankton blooms like the ones seen in Figure 6.3. As expected, Carbon Dioxide will proportionally deplete as a result; it is not being constantly replenished by the atmosphere and the consumer population is too small to replace it. Selection will enact at the reaching of the carrying capacity, explaining the peaks and drops in the graph.

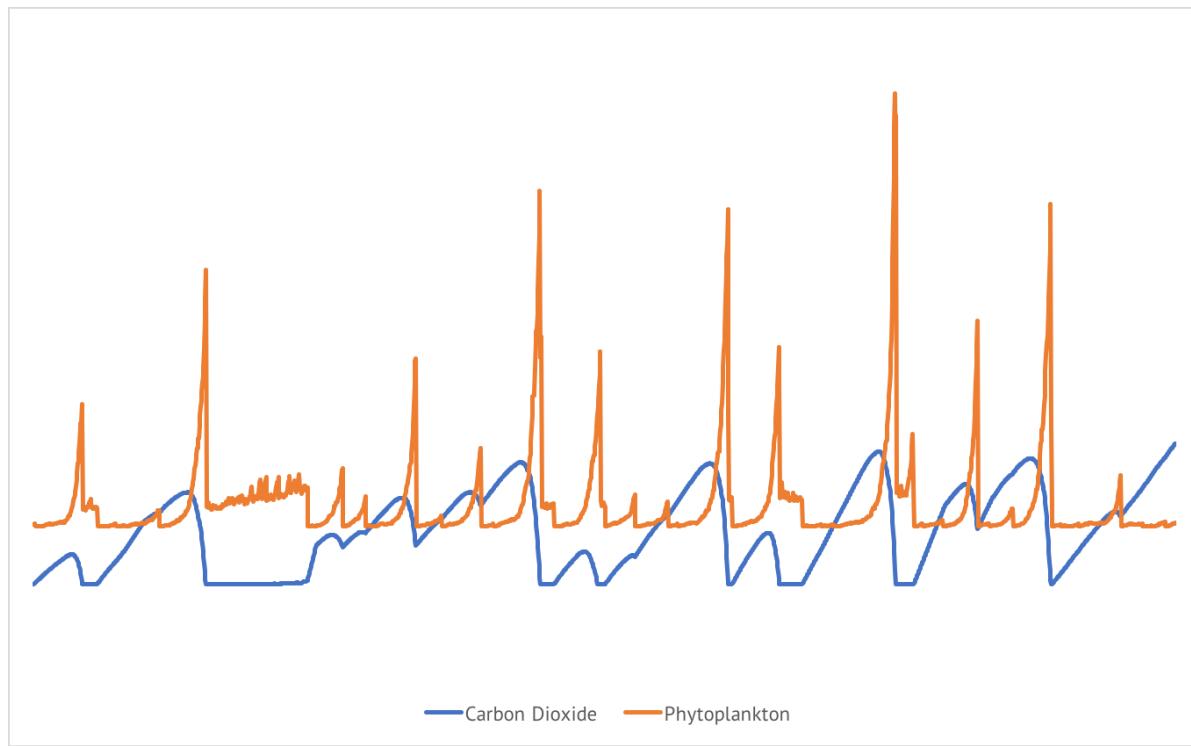


Figure 6.3: Phytoplankton and Carbon Dioxide in the Aphotic Zone

The consequences of the blooms in Figure 6.3 is further illustrated by Figure 6.4, were the gaseous quantities seem to mirror each other as a result. More formally, as Phy-

toplankton deplete Carbon Dioxide and augment Oxygen, the resulting high rates of photosynthesis will dominate in the dynamics as described by the equations in Figure 4.7. These results are expected of our model, and serve to validate the system further.

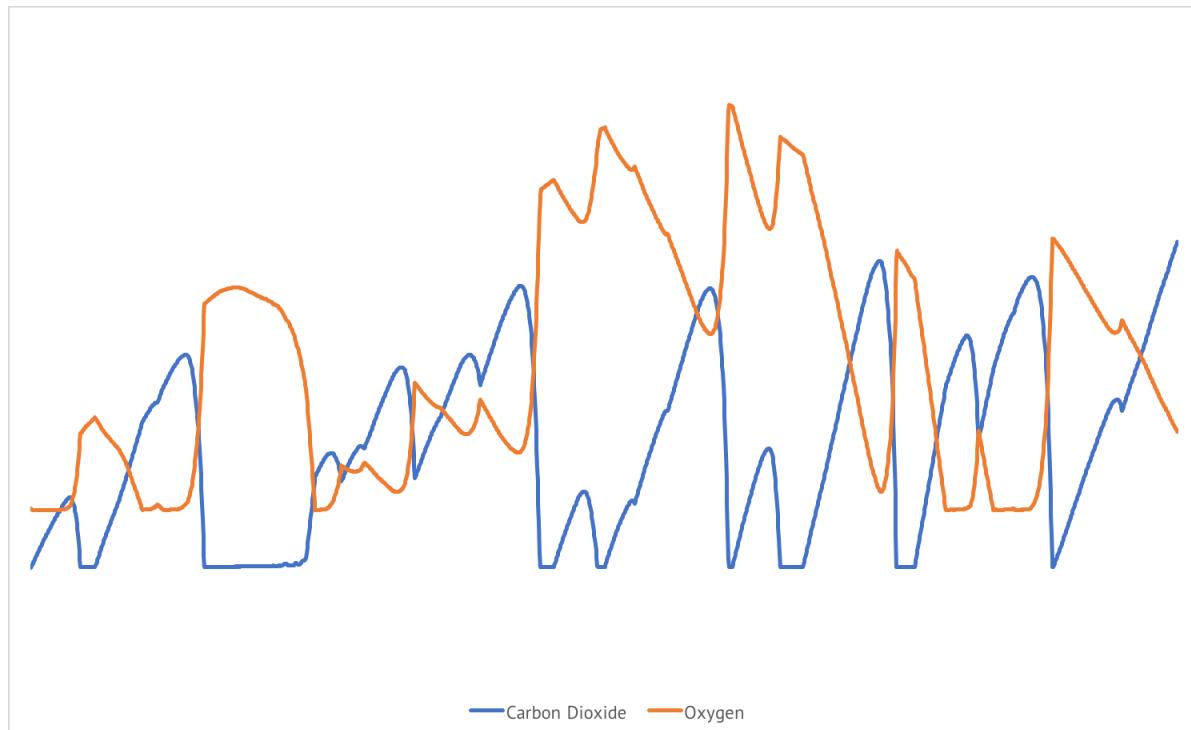


Figure 6.4: Carbon Dioxide and Oxygen in the Aphotic Zone

The Abyssal Zone is purposely set to receive virtually no light and an extremely low temperature, serving as an experimental area of the ecosystem. Close observation of the zone displays results expected of the models. In particular, the population remains alive only by the grace of the implementation disabling extinction, as the light and temperature are too low to enable any realistic photosynthesis or indeed respiration. Despite this, very small fluctuations have been observed due to a chance mutation producing unrealistically fit genes for a life process. Overall, this zone is relatively inactive, which confirms our model behaviour. Specifically, genes that live in such abyssal conditions should not survive, and typically do not in the real world (see Research). Indeed, actual marine ecosystems possess highly adapted organisms that live at these depths which our model does not consider.

6.2.3 User Input

The system provides the functionality to manually alter the abiotic factor of any given zone, in effect manually driving the dynamics of the system. Using the model descrip-

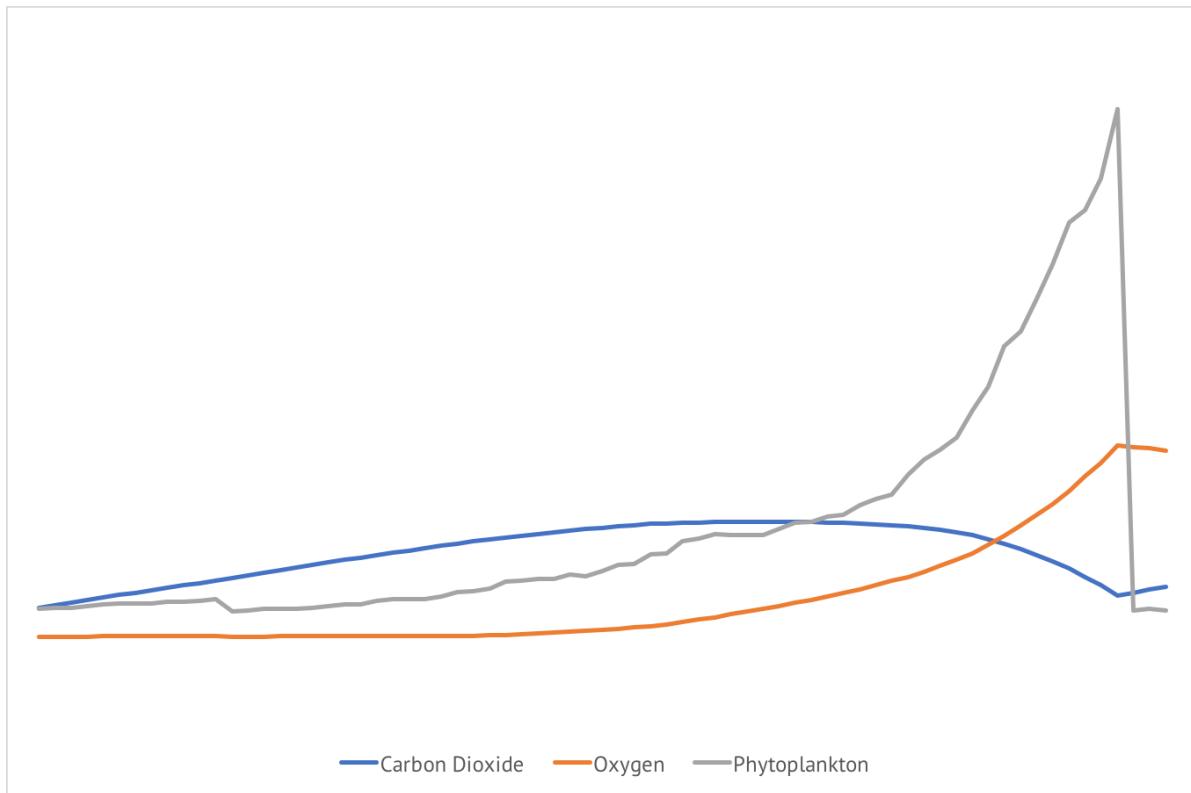


Figure 6.5: Effects of user increased Carbon Dioxide

tions we can predict the effects of doing so on the behaviour of the system. Figure 6.5 illustrates the effect of manually increasing the Carbon Dioxide level in the aphotic zone of the ecosystem. In this graph, the increase in the factor clearly correlates with a rapid bloom of Phytoplankton as a result of the lesser selection impact on the population. Indeed, the population will continually grow at an exponential rate as long as the user continues to augment more Carbon Dioxide to the zone, and is simple consequence of a net Carbon Dioxide gain. Eventually, however, the Phytoplankton population catches up to the rate of increase; the population is so large that the net depletion of Carbon Dioxide due to respiration will outweigh the user defined increase and selection will enact. Equally, ceasing the increase will also result in selection via the same effects. Crucially, the same effects are seen with Zooplankton and Crustacean populations when Oxygen is increased instead. Such behaviour is expected of the model, and further validates it's operation.

Chapter 7

Legal, Social, Ethical and Professional Issues

A project as large, experimental and research heavy as this one requires careful consideration of the legal, social, ethical and professional issues. Aims to mitigate any ramifications that result from these are presented in this chapter.

7.1 Legal

The use of third party libraries such as Processing may indicate the need to consider the licencing implications of using such libraries. As of writing, Processing is open source and open to use under the GPL [38]. Moreover, as Processing allows the export and distribution of Applets, it is legal to distribute this software for educational purposes.

7.2 Social and Ethical

Assuring that the accessibility of the software was maintained was a key issue that had to be tackled in regard to social issues. In particular, the UI was designed to be clear and intuitive to cater for all users, including those who may find it difficult to navigate a heavily scientific and analytic system. Moreover, the colour choice was bold and the primitives used to display the graphics were simple to further ensure visual clarity.

As a project that involves user testers to provide feedback, it was imperative to ensure

that privacy was maintained in the collection of feedback data. For instance, a secure database was used to store user feedback data that was accessible only to the developer.

7.3 Professional

Ensuring that the software implementation is maintained thoroughly following any publication is imperative to maintain professionalism. Further, a thorough documentation and a clear, commented coding style will aid in this further. Given the scientific nature of the project, all theories, background and assumptions will be included in the documentation to aid in the understanding of the system. Finally, strict adherence to the BCS code of conduct will further ensure that the project is seen as professionally as possible [2].

Chapter 8

Project Management

This chapter details the methodologies implemented during the development of the project, including an overview of the approach taken toward project management. Indeed, the research heavy, experimental nature of the project required a specific alteration of a typical methodology. This is also discussed in detail.

8.1 Methodology

The nature of the project dictated that the conceptual model was to be designed and validated before any implementation took place. Given this, the model design and the software development stages could be separated and would therefore encapsulate their own methodologies. Ideas from Black and Johnsonbaugh [13] and Ghahrai [24] were equally considered in approaching the methodology for the project.

8.1.1 Design

Black and Johnsonbaugh [13] presents a methodology for the design of simulation conceptual models which was adhered to in the model design. Specifically, the development was divided into distinct phases as per Black and Johnsonbaugh [13]'s recommendation.

The first of these phases is *defining the problem*. This involved defining the objectives of the study and the specific issues to be addressed. This is shortly followed by determining the boundary of the study and the abstraction level desired. In this context, this involved

identifying the marine ecosystem as a focus for modelling, followed by gathering research pertaining to it in order to understand the area and define a boundary with which to adhere, eventually leading a suitable level of abstraction. The end of this phase involves a formal proposal in the form of the project specification.

Following from this, *designing the study* involved investigating some steps of the previous phase in greater detail. In particular, the broader assumptions are evaluated with respect to the research conducted, and the tools to be used are researched and identified. It was determined to use Processing as a graphics API via some experimentation as an assessment of its validity. Finally, a functional specification was laid out that outlined the requirements of the conceptual model.

Conveniently, the *design of the conceptual model* took place following the background and elicitation stages prior. Specifically, the decision to model a discrete time simulation informed the design of the entities and relationships in the model. Similarly, the level of detail in each component was determined. The functions and experimentation of gene representation would lead way to the mathematical definitions of the entire dynamic system that involved each species entity.

Finally, *validation of the model* took place, which involved a consultation of the research, validation data and domain experts. As expected, revisions of the model took place until a final, acceptable and robust group of models was devised ready for a software implementation. Ensuring that this stage followed a plan based methodology such as this one ensured that development could encapsulate its own methodology without having to change due to an alteration of the model design.

8.1.2 Software Development

As defined by Ghahrai [24], the incremental build model involves design, implementation and testing in small stages. Each stage will develop a section of the system a little at a time until an entire system has been devised which satisfies all the requirements. As a methodology that combined elements of the waterfall model and iterative prototyping, it was perfect for the experimental nature of the project where each experiment required precision and detail.

Figure 8.1 illustrates a typical cycle that encapsulated an increment in the system development. In particular, each stage would identify a system requirement and consult the conceptual model for a detail on the design of the module. Analysis and design of

the model component based on the conceptual counterpart would typically inform the prototyping and precede several implementations until an acceptable model would be developed. Unit (or integration) testing would ensure the rigour of the module, and success would be evaluated with respect to the conceptual model before the next increment would begin. The order of each increment was intuitive; the smallest components such as Gene and Organisms taking precedence in development.

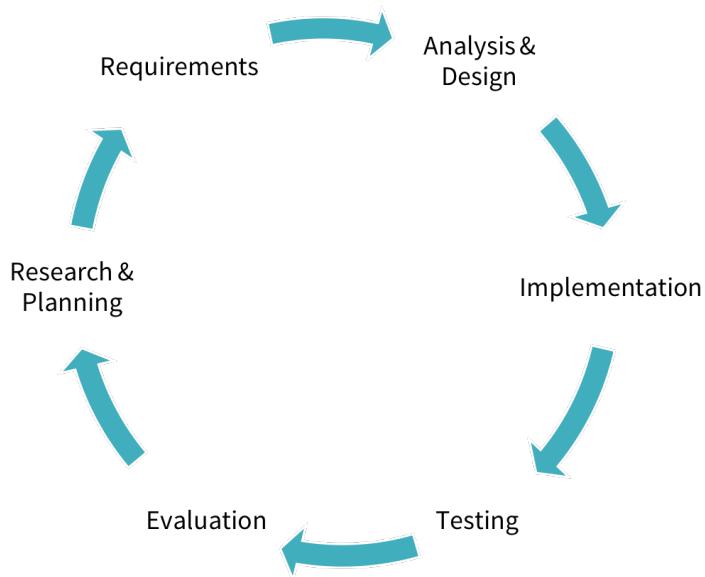


Figure 8.1: The incremental build cycle

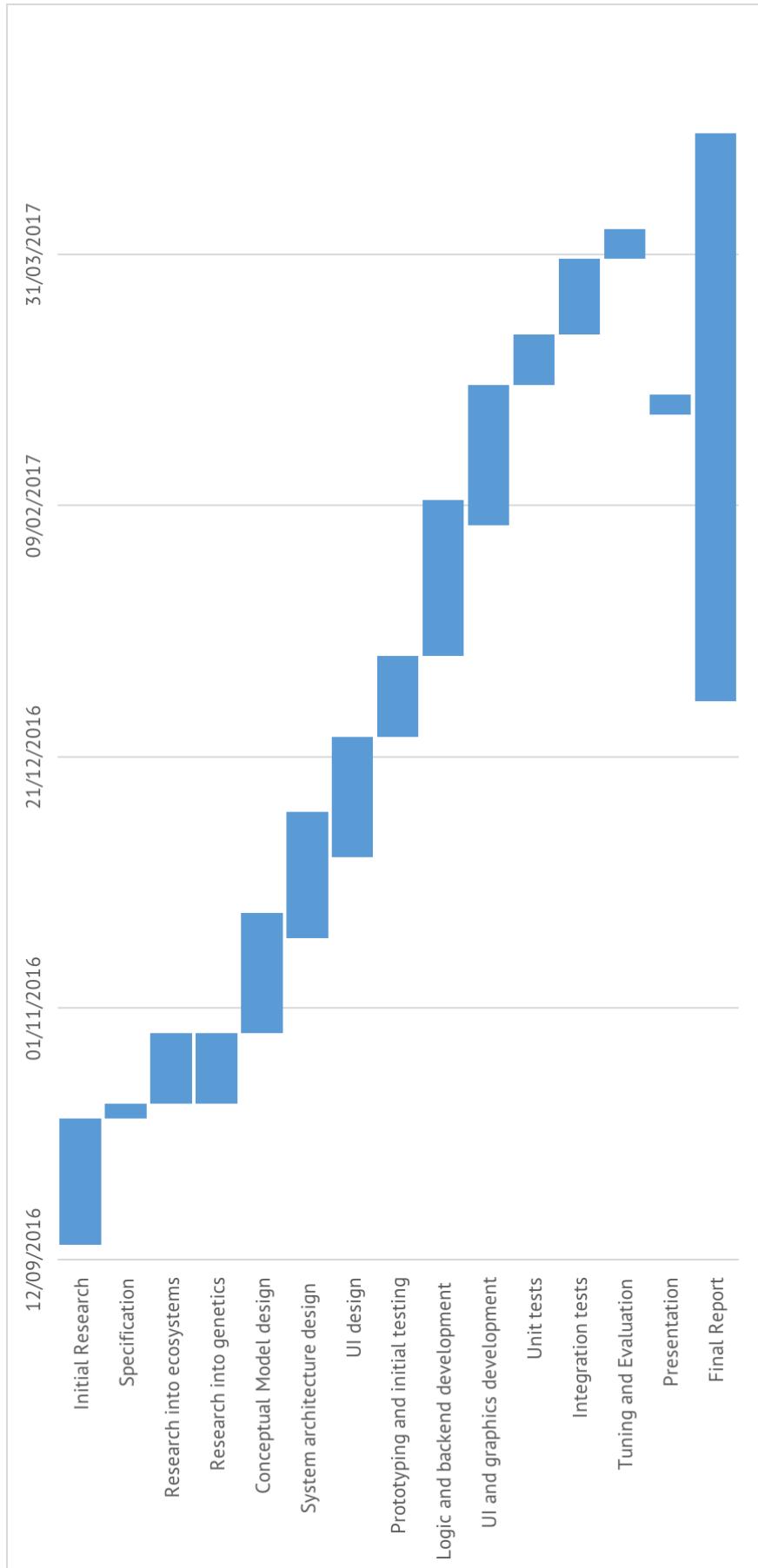


Figure 8.2: The revised Gantt chart

8.2 Project Timetable

As touched upon in the Progress Report, the original timetable outlined in the Project Specification underwent significant alteration and can be seen in Figure 8.2. This occurred in most part due to the unpredictable nature of the project as a significant research piece. Furthermore, additional sections of the timetable have been added as the project underwent better structure as it progressed.

For one, the initial research period outlined in the original timetable took far longer than expected. This was caused by an underestimation of the difficulty of the subject matter, and the literature around ecological modelling proved equally difficult to navigate. As a consequence, the conceptual model design proved longer than expected, in large part due to the failure of the initial genetic modelling experiments to produce a viable results. Additionally, the production of a viable system architecture involved unforeseen experimentation with Processing, also pushing this longer.

The reevaluation of the project in the conceptual model design resulted in less software development time as a happy consequence. In particular, modules had been better planned, such that the software could be built quicker as a result. Finally, as a result of the unpredictability of the system, the final report was pushed to begin far later than originally planned, as reflected by the new timetable.

8.3 Management Tools and Techniques

Multiple third party tools were used for the implementation and management of the project, where the choice of tool was informed by a feasibility study as highlighted below.

IntelliJ IDEA CE

This development tool was vitally important to the success of the project [6]. Specifically, the eclipse style IDE offered fast compilation and coding suggestions, thus improving the quality of the software implementation. Integration with Processing allowed instant deployment of the program, enabling incremental debugging.

Mathematica

This was used extensively during experimentation. The power and precision of this software make the results more valid, making the system more trustworthy.

Processing

The Processing library allows efficient and fast development of graphics applications [21]. Further, the OpenGL style syntax and the wealth of supplementary documentation providing tutorials for the API aided in the development of the application.

LaTeX

The documentation was implemented through LaTeX. The scientific typesetting offered a professional look to the documentation, and the powerful coding and mathematics presentation proved useful in illustrating some of the contents of the project.

8.4 Risk Management

An important part of the project management is the handling of potential risks that may hinder the success of the project. As a result, it became important to identify such risks and develop contingency plans to avoid any detrimental effects.

Failure of Hardware

This can be viewed as a risk of high consequence, with low probability. The entire software development was performed on a single machine (Macbook Pro 2014), and as such, a failure in the hardware would have ceased development until a replacement machine were to be found. Indeed, such a risk could lead to a loss in data. In order to manage this risk, the software was stored in a BitBucket git repository, with the repository undergoing continuous updates to ensure that roll backs were possible should something go wrong at any stage of development. Furthermore, the cross platform capability of IntelliJ and Processing meant that alternative hardware could be sought that allow development and testing to continue.

Underestimation of time management

As touched upon earlier, the research heavy and experimental nature of the project opened the risk that time would be managed poorly. Indeed, the Design chapter highlights failures in modelling that resulted in many modules having to be restarted. Thankfully, this was a known possibility even at the beginning of the project, so the project timetable was given enough room for adjustment should things change unexpectedly.

Chapter 9

Evaluation

Evaluation of the project must take into consideration the original System Requirements. More specifically, each functional and non-functional requirement has been assessed for completion and evaluated for success. Following this, the chapter will describe an evaluation of the project management of the system. Finally, a presentation of an assessment of the overall project from the author's perspective is given.

9.1 Functional Evaluation

Each of the functional requirements are given a result of either pass or fail, along with a comment on the evaluation of each object. Much like the original specification, we can divide the evaluations into the model and system requirements. Table 9.1 presents this.

9.2 Non-functional Evaluation

Following similar procedure, the non-functional requirements are presented in Table 9.2.

9.3 Project Management Evaluation

The two methodologies employed for the project have significantly contributed to its success. For one, the methodology adhered to for the design of the conceptual model was

plan based and rigid with some room for experimentation. This was crucial to the success of a model design that implements highly scientific concepts and ideas. Furthermore, ensuring that proper validation took place gave the model confidence and increased the stability and robustness of the yet to be implemented software implementation.

On the other hand, the agility of the incremental build methodology employed for the software implementation allowed flexibility for experimentation whilst combining the plan based approach that was crucial in building a very technical simulation. Small increments could be continually evaluated and validated as the project went on with respect to the requirements. This ensured that validation could be performed on basic elements of the system - had this not been performed, invalid elements may have made their way into the final system.

Finally, the tool set used proved to be highly successful. A surprisingly small learning curve and the use of fast development tools made development easier than expected and ensured that the final implementation was as robust as possible. The project timetable was revised from it's original one, which in itself was left open to change. This was crucial for an experimental project such as this one, and thus contributed to the success of the project management in general.

9.4 Reflection and Limitations

The project, since it's conception, has undergone many alterations and changes of approach. Indeed, the Research chapter outlines only a small snapshot of the areas that were studied in order to garner an understanding of the natural world and how to model it. This is further reflected by the drastic change in the contents of the documentation from the specification onwards; new insights and experiments periodically changed the requirements. Moreover, as the background was better understood, and the notion of how to model a dynamic ecological system was formulated within the context of the research, the model became more rigid. Namely, the requirements became stronger, and the model itself became more specific as a result.

The final system was successfully evaluated against the model requirements and was validated and verified. Furthermore, the simulation was unpredictable - the dynamic nature of the ecological system actually provided unforeseen insight into the notion of an ecological system, which only became apparent once the simulation has been completed. The capturing of results far after verification had been performed provided

it's own insight into evolutionary theory and population dynamics. Given that one of the aims of this simulation was to develop a system that could provide insight into natural phenomena in an educational context, we can deem this a success. The positive feedback from domain experts and the target audience during validation reinforces this success.

Equally, with any simulation of natural phenomena, there are limitations to this system. Much of these arise from the high level of abstraction, made for one due to the scope of the project, but also because of the immense complexity related to the field of study. For instance, the genotype contains but two genes which are represented solely as mathematical functions. This is far from the DNA encoding that actually encodes the genetic representation of organisms. Moreover, with reference to the Research chapter, many of the illustrated ideas were either heavily abstracted or omitted entirely. The genotype makes no reflection visually onto the genotype in any of the organisms, a limitation given that the notion of $G \rightarrow P$ mapping was researched. Furthermore, the only abiotic variables modelled are Carbon Dioxide and Oxygen - in actuality many factors influence photosynthesis and respiration. Furthermore, the Research brings together the arguments for modelling ecosystems as open, pulsing and self organising dynamic systems which incorporate a notion of energy transfer. Conversely, the conceptual model is a closed system without an energy transfer mechanisms.

These limitations, whilst numerous, do not necessarily indicate failure. Rather, they are areas that were explored but that were deemed as too complex or unimportant given the scope and the aims of the project. The final system still manages to successfully illustrate the effects of the environment on population dynamics and fitness. Therefore, if we evaluate the success of the project on the adherence to the System Requirements as outlined in the early portion of the document, we can conclude positively.

Requirement	Comment	Result
F1	The simulation cycle illustrated in the Design chapter represents a discrete time step. Equally, each frame of the simulation encapsulates this cycle, and a collection of these steps provides the illusion of time	PASS
F2	The physical space model divides the ecosystem into the aforementioned zones and applies properties to them as a consequence of light availability. Equally, the implementation succeeds in dividing the ecosystem into zones, where it treats each zone separately by associating distinct organism and abiotic stores with each. Moreover, the nature of each abiotic store differs based on the light availability	PASS
F3	The abiotic components have been divided into fluctuating quantities and constants. Namely, Carbon Dioxide and Oxygen are represented as entities in the dynamic model, and Light and Temperature are constants which differ between zones. The implementation succeeds in creating an abiotic map for each zone	PASS
F4/5	Each organism has a representation as a distinct entity in the models, each of which is effected by the food chain as defined by the dynamics. Moreover, the software implementation succeeds in representing organisms as objects, with each species encapsulating its own properties in order to fulfil this requirements	PASS
F6	The processes of Photosynthesis and Respiration were represented as processes in the conceptual model enacted by the genetic logistic function, and contribute to the dynamics of Carbon Dioxide and Oxygen. In the implementation, these processes are represented as methods within the species classes, which return a value corresponding to the fitness of the genes that determine their success that enacts the abiotic change	PASS

F7	The genetic logistic curve function successfully models each aforementioned gene encapsulated into a genotype along with it's fitness representation as a continuous value. The encoding given to each curve allows genes to differ and hence mutate as a result of a change in the curve itself. Moreover, the design and implementation successfully represents alleles as different expressions of the same gene by associating them within a Chromosome, and each gene can be given a property representing the dominance of the allele	PASS
F8	Mating works intrinsically with the Chromosome representation of alleles in the genotype by performing Mendelian crosses on the alleles and the designation new combinations for the genotype of offspring. The implementation succeeds in this by the use of data structures, and assigns probability by reference to fitness determined energy and a random number generator.	PASS
F9	A concept in the Design, the implementation succeeds in this by grouping organisms into zone associated data structures which allow retrieval and removal of organisms on the success of a predation event	PASS
F10	Selection is represented as a time step in the simulation cycle and successfully uses the logistic gene fitness to evaluate survival in the implementation	PASS
F11	The implementation uses Processing 2D in order to give the illusion of a cross sectional view, where the window height corresponds to the depth of the ecosystem.	PASS
F12	Successfully implemented using trivial Processing primitives	PASS
F13	The HashGrid data structure, along with Processing vector types enabled the movement of organisms in a cartesian space. Collisions were successfully implemented using the height and width of zones as a boundary	PASS

F14	The Processing draw method calls a redraw of every frame, and as such was used as a basis for representing discrete time. Specifically, the simulation cycle is encapsulated successfully within each frame redraw call	PASS
F15	Successfully utilised Processing user input methods to capture user input. Changing variables in the back-end implementation proved trivial provided that abiotic factors were safely represented within data structures that were visible to the front of the system. Similarly, speed changing took user determined adjustments and applied them to a delay, which occurred prior to every frame refresh	PASS
F16	Utilisation of the Zoomer package in Processing allowed mouse input to access the camera in the world, such that the user would be given the ability to pan and zoom around the ecosystem	PASS
F17	Effective use of the data structures associated with each Zone allowed interface to easily access data and use trivial Processing methods to print them to the screen through text	PASS
F18	Key presses can change a pointer which would be referenced by a text display method, making use of zone ID's to select different data	PASS

Table 9.1: Functional Evaluation

Requirement	Comment	Result
NF1	As a cross platform tool, building the simulation in Processing allows it to be deployed to multiple different operating systems and environment and hence results in a success	PASS

NF2	By using a lightweight tool such as Processing as the graphics API, as well as utilising data structures with efficient time access, the computational costs of the simulation have been kept to a minimum. Additionally, the simple primitives used to draw entities in the application further lowered the drawing and hence computational costs. All of these factors keep that the system requirements at a low, making it as accessible a tool as possible	PASS
NF3	The use of simple primitives and a consistent UI ensures that the system is simple. Further, bold colours separate entities, and nothing is shown on the screen that is not explicitly part of the requirements. Hence, a suitable abstraction of a marine ecosystem is the only thing visible to the user.	PASS
NF4	The strict use of well understood and robust software development approaches ensured that the system can be easily extended if need be. Additionally, classes were kept well commented and clean, and data structures remain extensible to the addition of more objects. In particular, adding more genes and species is trivially done by subclassing, and adding deeper zones or changing the properties of the ecosystem is a readily available extension.	PASS
NF5	The lightweight nature of Processing does not require the use of a dedicated GPU, and can use CPU cycles to draw the primitives that the simulation utilises	PASS
NF6	Adherence to the incremental build software methodology as well as scientific methods in simulation modelling have given way to a well documented and extensible project backed by robust referencing and an intuitive design approach	PASS

Table 9.2: Non-functional Evaluation

Chapter 10

Conclusion

10.1 Summary

The final system simulates a heavily abstracted marine ecosystem that contains fluctuating biotic and abiotic entities which intrinsically influence each other. This simulation is implemented as an interactive graphical application modelled in discrete time.

Research into the characteristics of the marine ecosystem was conducted to better understand it, including details regarding its biotic and abiotic components. With a goal in mind, further research provided insight into ecological modelling techniques, as well as exploration of the problem of modelling genetics and fitness.

Armed with the key insight provided by the literature review, the conceptual model was designed, which encapsulated dynamic entities by abstracting the marine ecosystem to some degree. Alongside this model, a physical space model was devised that an implementation could adhere to, along with a adaptable fitness function modelled after the logistic curve. Finally, the dynamics of this model was contemplated before an implementation was introduced. The implementation was then discussed, which used Java and Processing to simulate the model devised in the Design. It uses an object orientated structure to effectively and efficiently create a graphical application that closely resembles the posited design.

Finally, validation and verification was performed through testing and evaluation of the output of the system, comparing it to the expected output and with data obtained as a validation mechanism. Correctness was established after this phase - the system behaved as expected and was evaluated with this in mind.

10.2 Future Work

The system developed is based on a design that omits a portion of the detail discussed in the research. This was done in most part due to the scope of the project. However, should future work be conducted on the simulation, this detail could be revisited.

For one, more genes could be added to the genotypes of the organisms. This would better represent the vast complexity of the organisms chosen to model. Furthermore, more genes may indeed correspond to a greater number of abiotic factors; nutrients were entirely omitted from the model, a potentially important inclusion. Additionally, predator-prey mechanics could have greater effect to the dynamics of the system in any future implementation. This would tie in with an inclusion in a representation of energy travelling through the biome, a core part of the marine ecosystem. A proper $G \rightarrow P$ mapping would ensure another research area was fulfilled, whilst simulating better, more intuitive movement would add greater realism to the simulation. Moreover, the inclusion of predator based movement or even evasive prey would increase the realism greater still.

The visual fidelity, whilst kept purposely simple, could also undergo significant improvement. For one, a consideration for making organisms look more like they do could underpin a 3D implementation that would display greater detail. Increased detail would also be welcome in the analytics, with the potential of a dynamics graph view that could visualise population trends over time.

Bibliography

- [1] Life sciences. URL <https://www.3ds.com/products-services/simulia/solutions/life-sciences/>.
- [2] Code of conduct — membership — bcs - the chartered institute for it. URL <http://www.bcs.org/category/6030>.
- [3] Marine ecosystem. URL <https://www.britannica.com/science/marine-ecosystem>.
- [4] About us. URL <https://www.gicentre.net/about/>.
- [5] homeorhesis. URL <http://medical-dictionary.thefreedictionary.com/homeorhesis>.
- [6] IntelliJ idea the java ide. URL <https://www.jetbrains.com/idea/>.
- [7] Lotka-volterra equations. URL <http://mathworld.wolfram.com/Lotka-VolterraEquations.html>.
- [8] Nature news, . URL <http://www.nature.com/scitable/ebooks/cntNm-8/118523179#bookContentViewAreaDivID>.
- [9] Nature news, . URL <http://www.nature.com/scitable/topicpage/genetic-mutation-1127>.
- [10] W. Adams. Phet interactive simulation. URL <https://phet.colorado.edu/en/simulation/natural-selection>.
- [11] P. Alberch. From genes to phenotype: dynamical systems and evolvability. *Genetica*, 84(1):5–11, 1991. ISSN 1573-6857. doi: 10.1007/BF00123979. URL <http://dx.doi.org/10.1007/BF00123979>.
- [12] M. Anand, A. Gonzalez, F. Guichard, J. Kolasa, and L. Parrott. Ecological systems as complex systems: challenges for an emerging science. *Diversity*, 2(3):395–410, 2010.

- [13] J. J. Black and B. Johnsonbaugh. Simulation methodology: A practitioners perspective.
- [14] F. Chapin, M. Chapin, P. Matson, and H. Mooney. *Principles of Terrestrial Ecosystem Ecology*. Springer New York, 2006. ISBN 9780387216638. URL <https://books.google.co.uk/books?id=shsBCAAAQBAJ>.
- [15] C.-c. Chen, S. Nagl, and C. Clack. Computational techniques for modeling and simulating biological systems. *ACM Computing Surveys*, 34(3):1–36, 2006.
- [16] C. Darwin. *The Voyage of the Beagle*. Floating Press, 1839.
- [17] J. A. G. De Visser and J. Krug. Empirical fitness landscapes and the predictability of evolution. *Nature Reviews Genetics*, 15(7):480–490, 2014.
- [18] R. Ewald, C. Maus, A. Rolfs, and A. Uhrmacher. Discrete event modelling and simulation in systems biology. *Journal of Simulation*, 1(2):81–96, 2007.
- [19] W. Fennel and T. Neumann. *Introduction to the Modelling of Marine Ecosystems*. Elsevier oceanography series. Elsevier, 2004. ISBN 9780444517029. URL <https://books.google.co.uk/books?id=I2PwAAAAMAAJ>.
- [20] A. Finkelstein, J. Hetherington, L. Li, O. Margoninski, P. Saffrey, R. Seymour, and A. Warner. Computational challenges of systems biology. *Computer*, 37(5):26–33, 2004.
- [21] P. Foundation. Processing.org. URL <https://processing.org/>.
- [22] E. Games. Evol sim. URL <https://evolsim.itch.io/evol-sim>.
- [23] Z. Gasiūnaitė, A. Cardoso, A.-S. Heiskanen, P. Henriksen, P. Kauppila, I. Olenina, R. Pilkaitytė, I. Purina, A. Razinkovas, S. Sagert, et al. Seasonality of coastal phytoplankton in the baltic sea: influence of salinity and eutrophication. *Estuarine, Coastal and Shelf Science*, 65(1):239–252, 2005.
- [24] A. Ghahrai. Incremental model - advantages and disadvantages, Nov 2008. URL <http://www.testingexcellence.com/incremental-model/>.
- [25] R. Haupt and S. Haupt. *Practical Genetic Algorithms*. Wiley-Interscience publication. Wiley, 2004. ISBN 9780471455653. URL <https://books.google.co.uk/books?id=4vpQAAAAMAAJ>.
- [26] R. F. Horton. Aquatic ecosystems, jun 2001. URL <http://www.encyclopedia.com/science/news-wires-white-papers-and-books/aquatic-ecosystems>.

- [27] S. Jarvis. Lecture notes in software engineering, January 2017.
- [28] S. Kauffman and S. Levin. Towards a general theory of adaptive walks on rugged landscapes. *Journal of theoretical Biology*, 128(1):11–45, 1987.
- [29] S. A. Kauffman and E. D. Weinberger. The nk model of rugged fitness landscapes and its application to maturation of the immune response. *Journal of theoretical biology*, 141(2):211–245, 1989.
- [30] S. A. Levin. Ecosystems and the biosphere as complex adaptive systems. *Ecosystems*, 1(5):431–436, 1998.
- [31] E. Mamontov. Modelling homeorhesis by ordinary differential equations. *Mathematical and computer modelling*, 45(5):694–707, 2007.
- [32] A. Mitra and S. Zaman. *Basics of Marine and Estuarine Ecology*. Springer India, 2016. ISBN 9788132227076. URL <https://books.google.co.uk/books?id=1D6FCwAAQBAJ>.
- [33] E. P. Odum. *Fundamentals of ecology*, volume 3. JSTOR.
- [34] H. A. Orr. Fitness and its role in evolutionary genetics. *Nature Reviews Genetics*, 10(8):531–539, 2009.
- [35] J. Otwinowski and I. Nemenman. Genotype to phenotype mapping and the fitness landscape of the e. coli lac promoter. *PLoS one*, 8(5):e61570, 2013.
- [36] R. Pal and A. Choudhury. *An Introduction to Phytoplanktons: Diversity and Ecology*. Springer India, 2014. ISBN 9788132218371. URL <https://books.google.co.uk/books?id=3aCVngEACAAJ>.
- [37] M. Pigliucci. Genotype–phenotype mapping and the end of the genes as blueprint-metaphor. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365(1540):557–566, 2010.
- [38] Processing. processing/processing. URL <https://github.com/processing/processing/wiki/FAQ>.
- [39] E. J. Rykiel. Testing ecological models: the meaning of validation. *Ecological modelling*, 90(3):229–244, 1996.
- [40] R. G. Sargent. Verification and validation of simulation models. *Journal of simulation*, 7(1):12–24, 2013.

- [41] H. M. Sauro, D. Harel, M. Kwiatkowska, C. A. Shaffer, A. M. Uhrmacher, M. Hucka, P. Mendes, L. Stromback, and J. J. Tyson. Challenges for modeling and simulation methods in systems biology. In *Simulation Conference, 2006. WSC 06. Proceedings of the Winter*, pages 1720–1730. IEEE, 2006.
- [42] J. Schumacher. Species: Artificial life, real evolution. URL <http://www.speciesgame.com>.
- [43] B. Shneiderman and S. Ben. *Designing the User Interface*. Always learning. Pearson Education. ISBN 9788131721636. URL <https://books.google.co.uk/books?id=8A4d0-nFeukC>.
- [44] E. A. Silow, A. V. Mokry, and S. E. Jørgensen. *Some applications of thermodynamics for ecological systems*.
- [45] J. Steele. Steady state vs equilibrium in biology. URL <http://education.seattlepi.com/steady-state-vs-equilibrium-biology-6085.html>.
- [46] G. Tatum. Evolve js. URL <http://evolvejs.com>.
- [47] N. Wasmund and S. Uhlig. Phytoplankton trends in the baltic sea. *ICES Journal of Marine Science*, 60(2):177, 2003. doi: 10.1016/S1054-3139(02)00280-1. URL [http://dx.doi.org/10.1016/S1054-3139\(02\)00280-1](http://dx.doi.org/10.1016/S1054-3139(02)00280-1).
- [48] S. Wright. Evolution in mendelian populations. *Genetics*, 16(2):97–159, 1931.
- [49] J. Wu and D. Marceau. Modeling complex ecological systems: an introduction, 2002.
- [50] W. Yackinou. *Understanding Complex Ecosystem Dynamics: A Systems and Engineering Perspective*. Elsevier Science & Technology Books, 2015. ISBN 9780128020319. URL <https://books.google.co.uk/books?id=beAGrgEACAAJ>.

Appendix A

User Guide

The system is presented as a .jar file and is already packaged with Processing. It is also a requirement for Java 8 to be installed in order to run it. Furthermore, it should be noted that the system has only been tested on Mac, and it is not known how another system may run it. Given this, it is advisable to run the software on Mac.

```
java -jar Ecosystem.jar
```

Appendix B

Keywords

simulation The production of a computer model of something, especially for the purpose of study.

abiotic The living components of an ecosystem.

autotroph An organism that can produce its own food using light, water, carbon dioxide, or other chemicals.

biotic The nonliving physical and chemical components of an ecosystem.

flagella Flagellum, plural flagella, hairlike structure that acts primarily as an organelle of locomotion in the cells of many living organisms.

heterotroph An organism that cannot fix carbon and uses organic carbon for growth.

morphology The branch of biology that deals with the form of living organisms, and with relationships between their structures.

prokaryote A prokaryote is a single-celled organism that lacks a membrane-bound nucleus (karyon), mitochondria, or any other membrane-bound organelle.

protozoan A single-celled microscopic animal of a group of phyla of the kingdom Protista, such as an amoeba, flagellate, ciliate, or sporozoan.

CS310 - Progress Report

An interactive educational simulation of a marine ecosystem consisting of planktonic organisms and their evolution over time

Hassan Qadir

Contents

1	Introduction	2
2	Progress	2
3	Research	3
3.1	The Marine Ecosystem	3
3.2	Ecology of the Marine Ecosystem	3
3.3	Abiotic variables of the Marine Ecosystem	4
3.4	Producers of the Marine Ecosystem	6
3.5	Consumers of the Marine Ecosystem	6
3.6	Evolution of planktonic organisms	7
4	Approaching a Design	7
4.1	Ecosystem Model	8
4.2	Genetic and Selection Models	8
5	Requirements Specification	10
5.1	Non-functional Requirements	10
5.2	Functional Requirements	10
6	Appraisal and Reflection	11
6.1	Timetable	11
6.2	Methodology	11
	Glossary	12
	References	12
A	Background	14
B	Problem Discussion	14

C Objectives	15
C.1 Abiotic factors	15
C.2 Biotic factors	16
D Methodology	16
E Timetable	17
F Resources	17
G Project Risks	17
H Legal, Social, Ethical and Professional issues	18

1 Introduction

The fields of Ecology, Evolutionary Biology and Population Genetics (amongst other related fields) seem to lack an accessible simulation tool that can visualise concepts that are core to the subject area. This project aims to develop a rigorous simulation of a marine ecosystem, building on models inspired by research into the oceanic biome. Furthermore, it aims to simulate populations of autotrophs and heterotrophs within this ecosystem, (based on planktonic organisms) with the intention of observing evolution through natural selection as a response to changes in user controlled abiotic variables.

This document will briefly detail the research undergone preliminary to any design. It will then outline how such designs have been approached, followed by a revised requirements specification and a detail on (early) system models. Finally, the progress that has taken place will be reflected upon and compared to original timetable outlined in the specification document, noting key changes and consolidating new discoveries.

2 Progress

Since the original conception of the title, much work has been done. For one, the title has undergone a change as a result of the research that has taken place since the specification submission. The new title better represents the real world influences that the project is based on, and will be explained further in the report.

The ecology of marine ecosystems and the morphology of the organisms that live within them has been better understood. Research has been conducted in this area, and has taken up a significant portion of allocated time. By abstracting characteristic features of these environments appropriately, early models for the ecosystem have been devised that will be scrutinised further and used in the development of the simulation. Furthermore, research and thought has been devoted to the genetic representation of organisms in the environment, and how offspring of individuals in the environment will be tested for fitness and be selected. Again, this has lead to the development of early models for the simulation.

Prototyping of an ecological and evolutionary simulation using various graphics platforms has taken place, taking inspiration from the concepts discussed in the research and design plan. That being said, progress has been slow in actual development; the research period has taken a greater amount of time than anticipated.

3 Research

The simulation can be divided into two key areas; the aquatic ecosystem and the morphology and reproduction of the communities of organisms that reside within them, which (should) result in their evolution. A key research period was set aside in order for these to be properly understood.

3.1 The Marine Ecosystem

An aquatic ecosystem can be of the marine or the freshwater type. Within the scope of this project, it does not necessarily matter which of the two the simulation is based on, as the key feature of the system will likely be common in both. That being said, there is much academic literature available on the evolutionary patterns in certain marine organisms. The marine environment is also characteristically rather interesting, and is distinguished from freshwater ecosystems by the presence of dissolved compounds, especially salts, in the water (Kingsford 2006, p. 137). This feature might be an interesting inclusion in the simulation, and so the research was focused on the marine ecosystem, hence the title change.

3.2 Ecology of the Marine Ecosystem

The marine ecosystem can be divided into benthic (substratum based) and pelagic (aquatic) zones. The benthic compartment of the ocean starts from the shoreline and extends up to the hadal zone. Furthermore, the entire water mass above the ocean floor (benthic substratum) with all the diverse types of organisms is referred to as the pelagic zone (Mitra & Zaman 2016, p. 3). The benthic zone represents a wide diversity of ecological features, but given that the simulation should aim to model an aquatic or oceanic ecosystem, the simulation will be based on the pelagic (aquatic) zone of the marine ecosystem.

The pelagic zone can also be differentiated into neritic and oceanic zones (Mitra & Zaman 2016, p. 12), with the neritic zone representing the shallow extension of ocean from the intertidal zone to the continental shelf and has an average depth of 200m. With the oceanic zone being far deeper, it has more interesting ecological characteristics and likely a greater level of biodiversity; the oceanic compartment will be focused on.

The oceanic compartment can be divided into the photic and aphotic zones. The photic zone is restricted up to the layer where light is available and usually ranges up to 200m from the surface. Oceanic producers usually occupy this space, such as algae and phytoplankton, who serve as the nutrient base of the consumers. By contrast, the aphotic zone lies below this

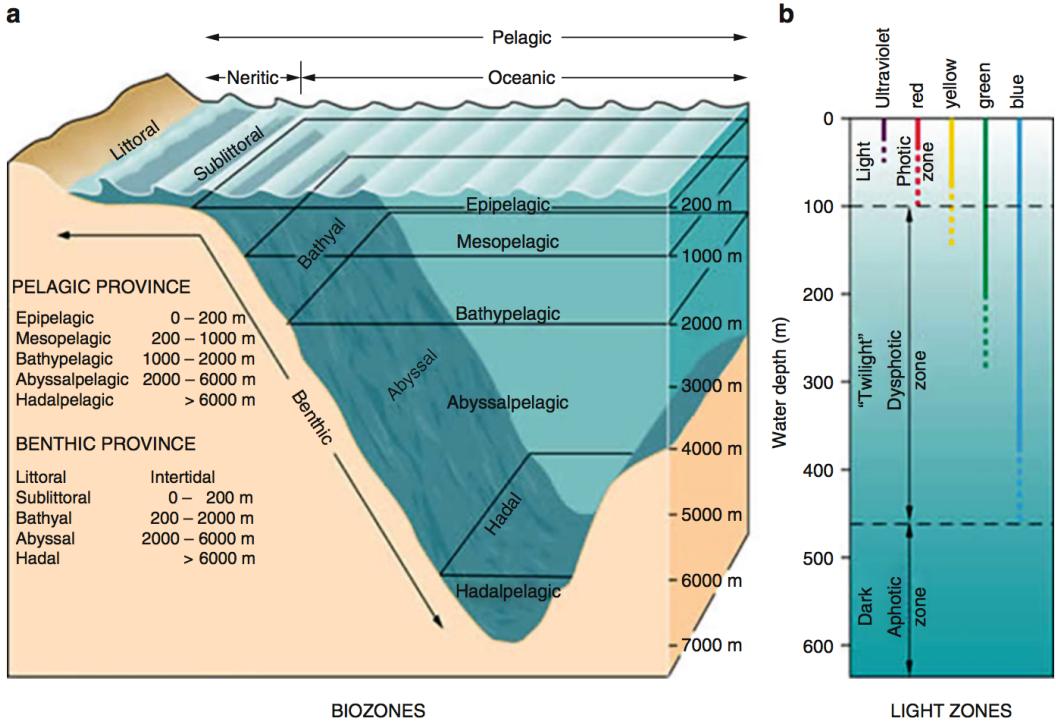


Figure 1: Benthic compartment of a marine ecosystem (Mitra & Zaman 2016, p. 4)

200m depth and has receives very low levels of illumination; the water absorbs more than 95% of the sunlight. Photosynthesis is therefore rarely possible (Mitra & Zaman 2016, p. 12). The exponential decrease in light intensity due to depth can be attributed to absorption and refraction, a phenomenon known as vertical light attenuation which can be mathematically expressed by Beer Lamberts law (Pal & Choudhury 2014, p. 43).

$$E_d(z) = E_d(0) \cdot e^{-kz}$$

where $E_d(0)$ and $E_d(z)$ are light intensity at surface and at depth z respectively. This relationship will likely be of importance in the modelling of the ecosystem, which will be discussed later on.

3.3 Abiotic variables of the Marine Ecosystem

The most abundant gases in the atmosphere and the oceans are nitrogen (N_2), oxygen (O_2) and carbon dioxide (CO_2) (Mitra & Zaman 2016, p. 91). Moreover, in photosynthesis, plants (and all producers, including phytoplankton) use carbon dioxide to form organic molecules and oxygen as a by product. However, the light needed for this is usually confined to the upper

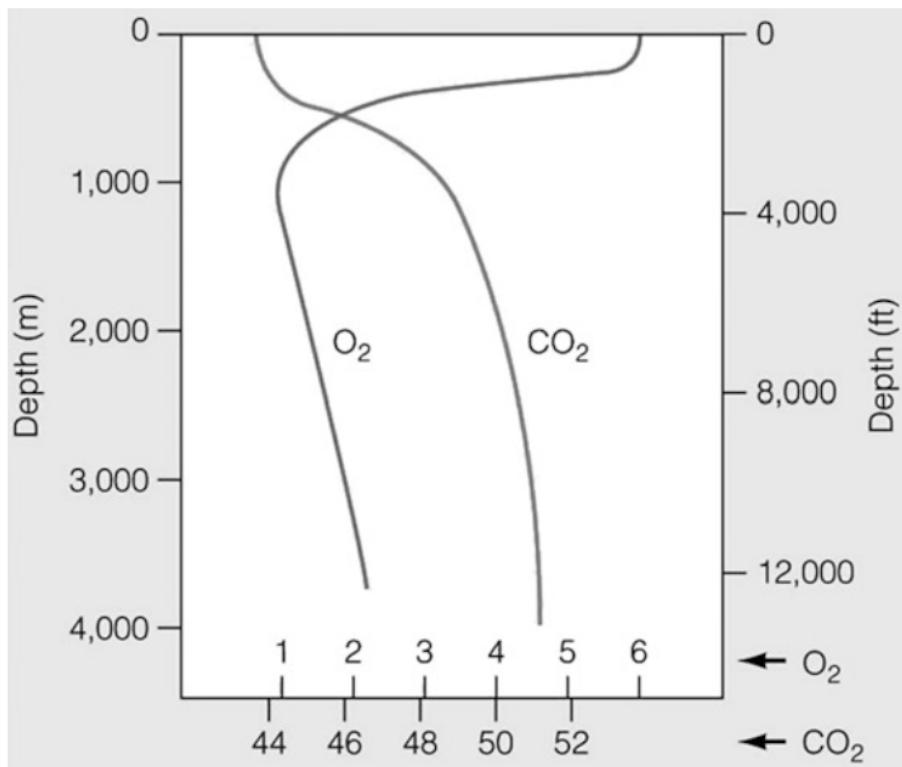


Figure 2: Distribution of O₂ and CO₂ with depth [concentration of dissolved gases in millilitres per litre (ml/l)] (Mitra & Zaman 2016, p. 92)

100m of the ocean, where the producers of the marine ecosystem usually live and survive. Oxygen is therefore typically produced in the surface water and carbon dioxide is usually consumed here.

By contrast, respiration breaks down organic substances, providing energy. The process requires oxygen and produces carbon dioxide. We also observe that decomposition, the process in which non-living organic material is broken down, also requires oxygen and releases carbon dioxide.

Oxygen usually enters the oceans at the surface due to exchange with the atmosphere or as a waste product of photosynthesis. Carbon dioxide may also enter through similar exchange, but is usually available at all depths due to respiratory and decomposition activity in organisms. We also observe that an oxygen minimum occurs at around 800m depth due to the absence of the photosynthetic process and the presence of the respiratory events of organisms that live in this aphotic zone (Mitra & Zaman 2016, p. 92).

Photosynthesis in the oceans produces a substantial amount of carbon. Most goes into short-lived micro-organisms and is rapidly returned to the ocean surface water when these organisms die or are eaten and burned by larger organisms. Some of the skeletal material produced in the mixed layer ends up in the deep ocean through sinking dead microorganisms.

Given this information, it would be clear that abiotic factors such as carbon dioxide and oxygen levels should clearly be modelled in the simulation, with a clear separation between photosynthetic and respiratory organisms. Differing populations of these organisms, as well as the abundance of such factors will likely drive the evolution of species.

3.4 Producers of the Marine Ecosystem

Most of the primary production in the marine ecosystems is carried out by phytoplankton (Mitra & Zaman 2016, p. 137). Phytoplankton are free-floating, photosynthetic, aquatic microorganisms, which move from one place to another, either actively by their locomotory organs (flagella) or passively by water currents (Pal & Choudhury 2014, p. 1). Some other primary producers are seaweeds, salt marsh grass, seagrass and mangroves.

Phytoplankton are the primary producer organisms of the aquatic compartment of the marine ecosystem. In effect, they support zooplanktons, fishes and other members of aquatic fauna (Pal & Choudhury 2014, p. 23). As a result, they are placed at the base of the trophic strata or bottom of the aquatic food web.

The phytoplankton population, controlling the life cycle of each species, is again controlled by several factors, like:

- The availability of nutrients
- Degree of thermal stratification
- Algal movements relative to the water current
- Zooplankton (heterotrophs) grazing
- Intra-algal competition
- Parasitism by protozoans
- Fungi
- Bacteria or viruses

(Pal & Choudhury 2014, p. 23).

There exists a number of pathways in which photosynthesis occurs within Phytoplankton, and thereby algae. Algae are photosynthetic organisms and they possess chlorophyll in their chloroplasts. The primary photosynthetic pigment of algae is chlorophyll and is the light receptor in photosystem I of light reaction (Pal & Choudhury 2014, p. 18).

3.5 Consumers of the Marine Ecosystem

The biodiversity of consumers in the marine ecosystem is immense; ranging from planktonic to large mammals. On the smaller end, we encounter a category of species known as zooplankton.

Like phytoplankton, zooplankton cannot move against the currents of marine ecosystems (Mitra & Zaman 2016, p. 194). Furthermore, they are heterotrophs. Many are herbivorous (consuming phytoplankton) whilst omnivores and carnivores can consume smaller zooplankton. They can also include jellyfish, and therefore have a large biodiversity and size range. They are typically the consumers in the second tier of marine and estuarine food chains, and can also be subdivided into holoplankton and meroplankton depending on the time they spend in free-floating form in their life cycles.

3.6 Evolution of planktonic organisms

There are many theories as to how certain marine organisms came to possess certain features. Phytoplankton is heavily studied in this regard and there exists a number of theories postulating how they came to photosynthesise.

An interesting evolutionary process known as endosymbiosis is detailed as a theory for the adoption of chloroplasts in eukaryotic algae, in where a prokaryote(ic) algal cell was captured in a food vesicle by a phagocytotic non-photosynthetic protozoan. Instead of digesting, the algal cells started maintaining it as endosymbiont, where the endosymbionts are used to get shelter from the host, whereas the host is used to get the supply of photosynthetic products from the endosymbiont. In the course of evolution, the endosymbiont turned into the photosynthetic cell organelle or chloroplast modifying the photosynthetic protozoa to eukaryotic algal cell. Eventually, in the process of evolution, the plasma membrane of the endosymbiont became the inner membrane of the chloroplast and the food vesicle membrane of the host became the outer membrane of the chloroplast envelope (chloroplast of Glaucophyta represents the intermediate stage). This process is termed as primary endosymbiosis (Pal & Choudhury 2014, p. 10). (Falkowski et al. 2004) proposes a similar theory in relation to the evolution of oxygenic photosynthesis in phytoplankton, in which endosymbiosis is the prime candidate.

Whilst endosymbiosis postulates a theory for the evolution of photosynthesis, another interesting area of research aims to discover how motile organisms came to evolve. A particularly interesting structure is the flagella. Such as structure has been described as 'irreducibly complex' and impossible to have evolved through natural selection. (Pallen & Matzke 2006) makes an argument involving ideas of self-assembly and mergers between several modular subsystems. Inevitably, such evolutionary incidents are hugely complex and falls vastly out of scope of this project to simulate. Incidentally, endosymbiosis falls under the same category.

4 Approaching a Design

The research undertaken over the past eight weeks has informed a clearer design and a re-evaluation of the specification laid out for the system since the original conception of the project. In particular, studying the ecology of the marine ecosystem and evolution of planktonic organisms that typically live within them, has shed light on what is important to simulate in order to most accurately meet the original design goals.

4.1 Ecosystem Model

The oceanic compartment of the pelagic zone of a marine ecosystem seems appropriate to model, given that it supports a large biodiversity. For instance, research indicated that it is divided into two distinct zones; the photic and aphotic zone, where the former receives enough light to support photosynthetic microorganisms such as (autotrophic) phytoplankton. Conversely, as the amount of available light decreases at greater depths, light becomes a severe limiting factor for photosynthesis; we have reached the aphotic zone. Here, heterotrophic organisms such as zooplankton that cannot fix carbon, instead acquire carbon through organic compounds produced by producer organisms like phytoplankton. The inversely proportional relationship between increasing depth and light availability influences the concentrations of carbon dioxide and oxygen within the ecosystem.

As observed by the research, carbon dioxide levels remains low in the upper regions, (photic zones) whilst oxygen levels remain high due to abundance of photosynthetic phytoplankton. Clearly, this relationship changes in the photic zone, where the absence of photosynthesis results in greater concentrations of carbon dioxide and smaller concentrations of oxygen, due to the increased presence of heterotrophs that require oxygen to respire. This relationship can be seen in Figure 2. These simple abiotic factors, namely light, carbon dioxide and oxygen levels are prime candidates for user adjustable variables in our simulation. The factors will control the populations of autotrophs and heterotrophs, whose own individual populations in turn will be controlled by each other in the emergent food web. Here, phytoplankton (or simply photosynthetic organisms) are the primary producers and thereby occupy the bottom-most trophic level in the simulated marine biome. Such an ecosystem can be divided into the aforementioned light dependent zones.

Figure 3 illustrates an (early) ecosystem model devised from the ideas mentioned above. The user will likely see a top-down view of this ecosystem, being able to 'zoom' in as the depth increases. The ecosystem is divided by light zones, where the aphotic and abyssal zones both receive minimal to zero light and makes photosynthesis impossible. Naturally, phytoplankton persist in the photic zone which is likely abundant in oxygen. This, along with plenty of food (phytoplankton) offers a suitable habitat for passive and perhaps motile zooplankton.

The aphotic zone will show some interesting adaptations. Likely only (motile) larger zooplankton will live here, potentially of the omnivorous sort. The harsher environment will likely apply greater selective pressure on the gene pool here, resulting in speciation. Food is likely going to be other zooplankton and the decaying matter of phytoplankton that drift downwards. Meanwhile, the abyssal zone will be deprived of key abiotic features; very adapted organisms will live here. The distribution of gases will be similar to the model in Figure 2.

4.2 Genetic and Selection Models

We can clearly see a model begin to emerge that appropriately represents the abiotic component of the ecosystem. Such a model is young and will be held to scrutiny during its full design and eventual development, a likelihood highlighted in the original project specification. However, a key second model needs development: the selective feature of the simulation that was meant to showcase the adaptation and eventual evolution of the organisms.

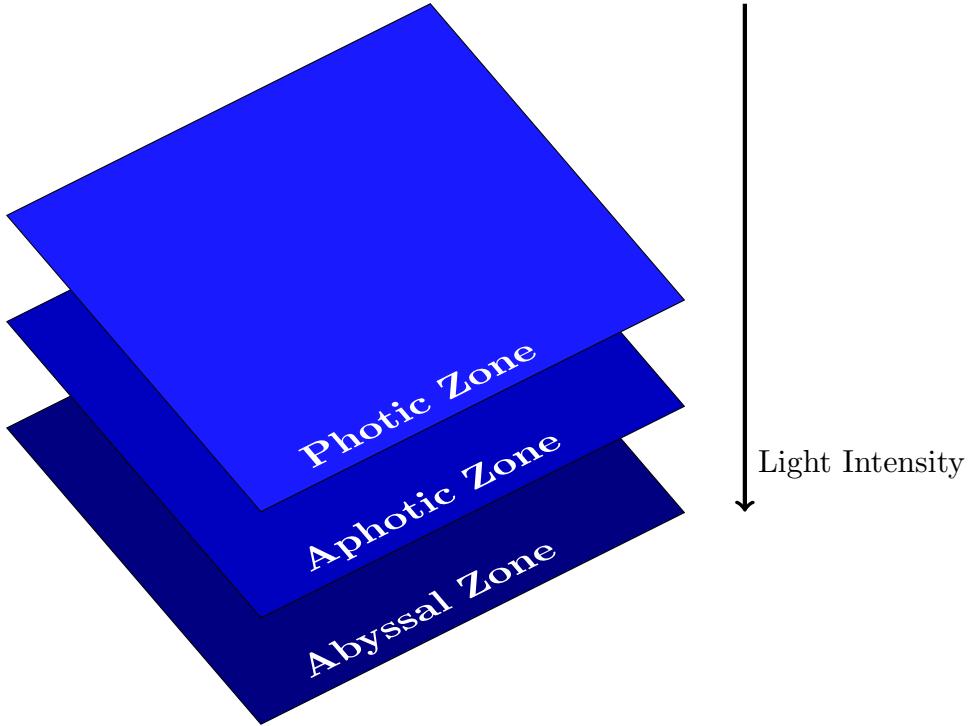


Figure 3: Light Intensity variation across Ecosystem Model

Taking inspiration from real world adaptations in planktonic species can help inform a model design. For one, in accord with the vertical limitations of the phytoplankton, actively growing zooplankton also concentrates in the upper waters close to its food supply. Much of the plankton also has developed some means of motility and some zooplankton migrates vertically in order to feed in the upper water layers and then sinks to deeper and cooler waters to rest, thereby conserving energy (Tappan & Loeblich 1973).

Aside from this, common adaptations that can be a selective advantage become apparent: faster motile zooplankton may gain food quicker, and a high surface-to-volume ratio is particularly effective in aiding suspension of the passive plankton (Tappan & Loeblich 1973). Moreover, differing chlorophyll pigments may be better adapted to capturing light energy for a particular wavelength or intensity of light - the list is endless. However, it is not the job of this simulation to predict adaptive advantages; that should occur naturally. At a basic level, we require an abstract genetic code that transcribes a particular phenotype in an organism. The code should contain genes which are measured by fitness in the ecosystem at any one time. The frequency of better alleles or genes should increase when the user applies a selective pressure by adjusting an abiotic variable in the ecosystem model. Random mutations should result in genetic variation within the population that allows for natural selection, genetic drift or gene flow within the gene pool. Other evolutionary processes such as endosymbiosis are too complex for our model.

This genetic code must feature in the genetic model. A parser should read the code of an

organism and visually represent this in the phenotype. Furthermore, the parser should apply a fitness heuristic to each 'gene' and measure the success of it in relation to its relative objective. At this point in time, such a model still needs to be derived, and further research around evolutionary theory must be conducted beforehand.

5 Requirements Specification

The research undertaken over the past eight weeks has informed the collection of a more rigorous requirements specification for the system, on which the design will be built.

5.1 Non-functional Requirements

1. Simulate the abiotic component of the ecosystem
 - (a) Model solar energy as the primary energy source for the ecosystem
 - i. Accommodate the inversely proportional relationship of oceanic depth and light intensity
 - ii. Model light as a general collection of varying wavelengths, with different ones contributing to different depth of aquatic penetration
 - (b) Model chemistry that drives certain processes in the ecosystem
 - i. Include carbon dioxide and oxygen for photosynthesis and respiration
 - ii. Include other nutrients and soluble compounds needed for biochemical processes
2. Simulate the biotic component of the ecosystem
 - (a) Represent basic physiological or biochemical processes within organisms
 - i. Simulate photosynthesis, with different species having varying productivity depending on availability of limiting factors and different wavelengths of light
 - ii. Simulate decomposition of autotrophs (such as phytoplankton)
 - iii. Simulate respiration in heterotrophs of organic compounds
 - (b) Simulate predator-prey relationships within communities of zooplankton
 - (c) Visually and informatively represent such organisms based on an abstract genetic encoding
 - i. Have such organisms reproduce, producing offspring defined by parent genes
 - ii. Reproduction should feature the facility of random mutations in order to drive selection and eventually result in the evolution of species

5.2 Functional Requirements

1. Allow the user to control abiotic factors such as light intensity and atmospheric carbon dioxide and oxygen

2. Provide the functionality to speed up or slow down time in the simulation
3. Allow the user to 'zoom' through varying depths of the ecosystem model
4. Have the system produce a phylogenetic tree representative of organisms in the simulation

6 Appraisal and Reflection

The original timetable set out to have completed system architecture and UI designs by this point in time; this goal has not been achieved. For one, the research period has taken significant time due to the unfamiliarity of the subject area. During the research period, many routes were taken that eventually lead to out of scope ideas being explored too deeply. Moreover, various models have been devised, but further research lead to revisions and back-tracking.

Prototyping has taken place using a variety of engines. These include: Unreal Engine 4, Unity and Processing. It is yet to be finalised which engine will be used for development, a decision that has admittedly run over its timescale. Nonetheless, it shall be decided in the last two weeks of term (not timetabled).

6.1 Timetable

An updated timetable has been devised which better suits the project scope at this point in time. Omitted activities have been kept the same and remain in the Gantt chart in the original specification.

Start Date	End Date	Activity
28th November 2016	30th November 2016	Genetics and Evolution research
30th November 2016	8th December 2016	Genetics and Evolution modelling
8th December 2016	15th December 2016	System Architecture Design
15th December 2016	1st April 2017	System Development

6.2 Methodology

The original specified incremental build methodology has been largely adhered to, although development has yet to take significant progress. Nonetheless, models and ideas have been continuously and iteratively scrutinised and re-evaluated as a result of further research and feedback from academics and peers. The same methodology will be adhered to in the coming weeks when development begins to take shape.

Glossary

abiotic The living components of an ecosystem (Gates et al. 2013). 2, 10

autotroph An organism that can produce its own food using light, water, carbon dioxide, or other chemicals. 2, 8, 10

biotic The nonliving physical and chemical components of an ecosystem (Gates et al. 2013).
10

flagella Flagellum, plural flagella, hairlike structure that acts primarily as an organelle of locomotion in the cells of many living organisms. 7

heterotroph An organism that cannot fix carbon and uses organic carbon for growth. 2, 6, 8, 10

morphology The branch of biology that deals with the form of living organisms, and with relationships between their structures. 2, 3

prokaryote A prokaryote is a single-celled organism that lacks a membrane-bound nucleus (karyon), mitochondria, or any other membrane-bound organelle. 7

protozoan A single-celled microscopic animal of a group of phyla of the kingdom Protista, such as an amoeba, flagellate, ciliate, or sporozoan. 6, 7

thermal stratification A change in the temperature at different depths in the lake, and is due to the change in water's density with temperature. 6

References

Adams, W. (n.d.), ‘Phet interactive simulation’.

URL: <https://phet.colorado.edu/en/simulation/natural-selection>

Darwin, C. (1839), *The Voyage of the Beagle*, Floating Press.

EpicGames (n.d.), ‘Unreal engine 4’.

URL: <https://www.unrealengine.com/unreal-engine-4>

Falkowski, P. G., Katz, M. E., Knoll, A. H., Quigg, A., Raven, J. A., Schofield, O. & Taylor, F. J. R. (2004), ‘The evolution of modern eukaryotic phytoplankton’, *Science* **305**(5682), 354–360.

URL: <http://science.sciencemag.org/content/305/5682/354>

Games, E. (n.d.), ‘Evol sim’.

URL: <https://evolsim.itch.io/evol-sim>

- Gates, D. M., Thompson, J. N. & Thompson, M. B. (2013), 'Biosphere'.
URL: <https://www.britannica.com/science/biosphere>
- istqbexamcertification (n.d.), 'What is incremental model- advantages, disadvantages and when to use it?'.
URL: <http://istqbexamcertification.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/>
- Kingsford, M. J. (2006), 'marine ecosystem'.
URL: <https://www.britannica.com/science/marine-ecosystem>
- Lenton, T. M., Boyle, R. A., Poulton, S. W., Shields-Zhou, G. A. & Butterfield, N. J. (2014), 'Co-evolution of eukaryotes and ocean oxygenation in the neoproterozoic era', *Nature Geosci* 7(4), 257–265.
URL: <http://dx.doi.org/10.1038/ngeo2108>
- Maxis (n.d.), 'Spore'.
URL: <http://www.spore.com/what>
- Mitra, A. & Zaman, S. (2016), *Basics of Marine and Estuarine Ecology*, Springer India.
URL: <https://books.google.co.uk/books?id=1D6FCwAAQBAJ>
- Pal, R. & Choudhury, A. (2014), *An Introduction to Phytoplanktons: Diversity and Ecology*, Springer India.
URL: <https://books.google.co.uk/books?id=3aCVngEACAAJ>
- Pallen, M. J. & Matzke, N. J. (2006), 'From the origin of species to the origin of bacterial flagella', *Nat Rev Micro* 4(10), 784–790.
URL: <http://dx.doi.org/10.1038/nrmicro1493>
- Schumacher, J. (n.d.), 'Species: Artificial life, real evolution'.
URL: <http://www.speciesgame.com>
- Stanford, L. L. & Spacie, A. (1994), *Biological monitoring of aquatic systems*, CRC Press.
- Tappan, H. & Loeblich, A. R. (1973), 'Evolution of the oceanic plankton', *Earth-Science Reviews* 9(3), 207 – 240.
URL: <http://www.sciencedirect.com/science/article/pii/0012825273900925>
- Tatum, G. (n.d.), 'Evolve js'.
URL: <http://evolvejs.com>

Project Specification

A Background

Evolutionary biology taught within the context of GCE Advanced Level education lacks a key visual element that could accompany the theoretical concepts it teaches. Whilst simulations of biological evolution already exist both commercially and in academia, the accessibility of such software is often poor. Moreover, the actual depth of the simulation is not often wholly appropriate for properly teaching the concepts introduced at such an educational level. Given this, approaching the problem of creating an evolutionary simulator poses an interesting challenge.

Simulation games that do exist such as 'Species' (Schumacher n.d.), 'Evol Sim' (Games n.d.), 'Phet' (Adams n.d.) and 'Evolve JS' (Tatum n.d.) focus primarily on the first principles of evolutionary science; namely variation, mutation and natural selection. This fails to accommodate additional concepts such as game theory, behavioural ecology or even evolutionary psychology. Furthermore, most simulations do not model an organism's internal biology. Rather, a few seem to focus on showcasing natural selection outside of a biological context, simply implementing trivial genetic algorithms. An abstracted genetic encoding is also rarely featured, making these simulations limited for education. Finally, existing simulations seem to lack a level of interactivity that would support the visual element in teaching the theory.

The lack of the aforementioned features in a commercial educational simulation leads on to the problem discussion by attempting to meet these needs.

B Problem Discussion

The overall aim of the project would be to develop some interactive software with a high graphical fidelity. Games like Spore (Maxis n.d.) were acclaimed for invoking a sense of excitement towards learning about evolution; the accessibility and UI being instrumental in achieving this. Given the target audience, it would be wise to attempt to emulate this. Achieving such a standard will nonetheless pose a significant design challenge.

Beyond this, classifying which species to base the simulation on is important to consider. On one hand, modelling land based organisms seems appealing; concepts of natural selection are often illustrated by mammals (Darwin 1839). However, modelling mammalian behaviour will potentially be far out of scope. On the other hand, unicellular organisms are far easier to simulate, particularly in the representation of a genome. An accurate representation of a organism's genome would obviously be too difficult and out of scope, so this will have to be heavily abstracted. Works such as (Falkowski et al. 2004) and (Lenton et al. 2014) reinforce the choice of focusing on aquatic eukaryotes; a large body of research exists in the study of their evolution, and the simplicity of their ultra-structure aligns with the scope. The characteristics of a oceanic ecosystem is also far simpler to simulate than a land based one.

To formalise the problem deeper, consideration must be given as to what should actually be simulated. Organisms must live in an ecosystem in which they can evolve in, although every discovered feature of an aquatic ecosystem would be too difficult to simulate. Rather, key abiotic features that can change and apply selective pressure on the organisms within it should be simulated. Users should also be able to alter these features in order to drive evolution. Similarly, the organisms themselves must live within communities. Sexual reproduction may lead to genetic variation in the gene pool, potentially leading to the natural selection of more suitable organisms.

Crucially, an entire system must be created that transcribes organisms based off of an abstracted genetic encoding that can often change. Moreover, whilst this is a very basic overview of what the project should aim to achieve, research into evolutionary biology will likely more intimately define the problem, resulting in further changes.

C Objectives

The aquatic ecosystem should be simulated appropriately; biotic communities should be structured by biological interactions and abiotic environmental factors. This should drive the simulation.

C.1 Abiotic factors

- Substantial research will need to be conducted in order to formalise which features of the environment to simulate. Some factors of aquatic ecosystems include substrate type, water depth, nutrient levels, temperature, salinity, and flow (Stanford & Spacie 1994).
- The abiotic environment will be abstracted in order to accommodate the scope of the project. Whilst all documented features are important to model in the consideration of true ecological simulation, this is not the focus of the project. Rather, basic ecological features that apply selective pressure on organisms that occupy specific niches should be present in order to drive the simulation.
- The level of interactivity between the user and the control of the environment will be important in defining the use case and ultimately the simulation. The user may be able to apply alterations to the environment, whilst the control features may be light intensity, nutrient levels or food abundance.
- An element of time control will be useful; users should be able to speed up a simulation in order to see the effects of evolution over a substantial period of time.
- The characteristics of the environment should be based on one that has shown evidence to have influenced significant evolution. For instance, (Lenton et al. 2014) discusses the Neoproterozoic era and how eukaryotes influenced ocean oxygenation, thus driving evolution.
- The visual element of the environment will need to be developed with a decently high degree of graphical fidelity.

- The ecosystem may exhibit features of an ecological cycle, such as the cycle of energy or carbon. This could interact with the biotic factors in the simulation and ultimately contribute to the simulation.

C.2 Biotic factors

- The depth of the simulation of biotic factors will need to be formalised and abstracted so as to stay within the bounds of the project scope; the focus should not be to simulate internal cellular mechanisms. However, some abstract representation of such mechanisms should be considered.
- The genetic representation of each organism will need to be appropriately abstracted in order to illustrate the evolutionary concept, such is the actual aim of the project.
- The classification of organisms that occupy the environment should be decided. The consideration of simulating autotrophic and heterotrophic organisms and their interaction is a potential idea of extension.
- The abstract genetic encoding will need to be developed such that it transcribes a certain structure to an organism whilst still being constrained within a logical framework. Random mutations between generations of organisms will be tested against the environment, potentially developing new species.
- Evolutionary Stable Strategies should drive the eventuality of the gene pool, although further research into the mechanics of evolution need to be done to formalise this.
- Behavioural mechanics should be researched and may be a feature in the simulation.

D Methodology

Substantial research must be conducted before an initial plan is developed, particularly in exploring the ecological and biological concepts outlined earlier. Once the relevant research is performed, an idea of what should be implemented will form the basis of the initial system design. This will likely be subject to further change as more research is discovered and the scope of the project is gradually verified.

Nonetheless, it is important to establish clear goals, so that models can be written that fit into a system architecture. This should be based on heavily verified and well documented research, but allow incremental development and prototyping so that theoretical concepts can be tested and either adjusted or removed based on feedback from domain experts.

For these reasons, an incremental build model will be used for this project, given that it combines elements of the planned based waterfall model with the iterative philosophy of prototyping (istqbexamcertification n.d.). Initially, models will be written based on rigorous research. A system architecture will then be incrementally prototyped until the development phase begins properly. The architecture is expected to adjust based on the findings of continual research and a clearer alignment of scope.

E Timetable

The Gantt chart at the end of the document details the rough timetable for the project.

The initial phase of the project will be research; principles of ecology and evolutionary theory must be well understood before actual planning can get underway. Investigation into the tools to use to develop the simulation should also be performed during this period.

Once the initial research phase is concluded, an architecture derived from models based on the research can begin development. How the ecosystem will be simulated, as well as the system of evolution are crucial elements that need to be planned carefully here.

Iterative development will test early prototypes; continual research may fill in gaps or even replace concepts deemed out of scope or broken through development. During this phase, the architecture is gradually formalised, being documented as it evolves.

F Resources

One of the most substantial challenges is the development of the visual element of the system. This will be achieved by using an existing graphics engine that provides resources that allow easy development of visual assets to facilitate the simulation.

Unreal Engine 4 (EpicGames n.d.) provides key features required by the project; a rich library of assets and tutorials to quickly develop graphical elements based on a powerful engine. C++ source code access allows the back-end logic to define the behaviour of the simulation. Artificial intelligence features of the product may also prove useful in the development of the simulation.

A goal of the project is to minimise the system requirements as much as possible, thus increasing the accessibility of the software and closer meeting the educational requirement of the project. Should the Unreal Engine prove too intensive, a web based approach should be explored. Advanced JavaScript and Java backend would aim to provide the tools for the simulation; modern browsers make good use of client hardware and would serve to bring the simulation to life.

Git and github will be used for version control and asset storage. Furthermore, SQL will be used as a database to store the growing amount of information collected or generated as the simulation occurs.

G Project Risks

An element of the research phase of the project may involve field work, which carries an associated risk of physical injury. To minimise this risk, care will be taken to choose a field site that is safe and appropriate for research. Ensuring that a professional is present during

research will also serve to minimise such risk.

The risk of equipment damage will be minimised by ensuring that project assets are backed up in multiple storage locations.

H Legal, Social, Ethical and Professional issues

Licencing of the Unreal Engine is free; (EpicGames n.d.) simply asks for a 5% royalty at the point of commercial distribution, so this should not be an issue. All other licencing terms will be reviewed and adhered to.

It may be the case that during field research certain legal requirements are met (land ownership laws). If this is the case, all issues will be well researched before any research is conducted. All of this will be well documented in the final report.

