April 2017

EvoSim: Genetic Evolution Simulation

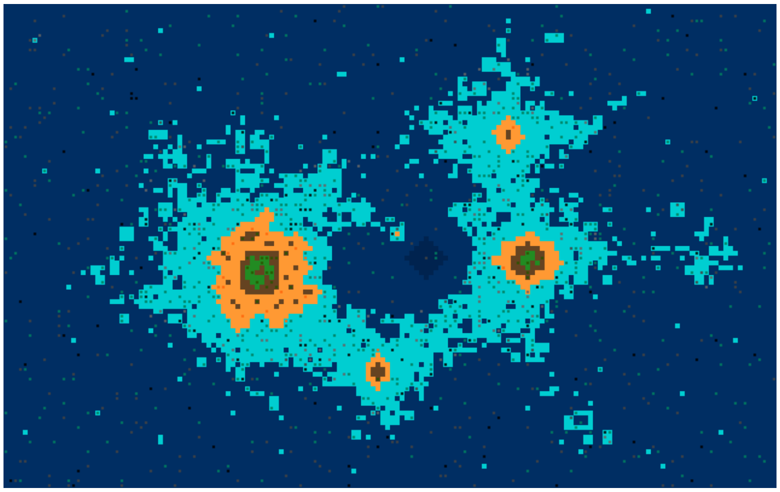
Yigit Uyan
Worcester Polytechnic Institute

EvoSim

Genetic Evolution Simulation

By Yiğit Uyan

MQP - MXC 0504



EvoSim - Genetic Evolution Simulation

Major Qualifying Project

Submitted to the faculty of Worcester Polytechnic Institute

In partial fulfillment of the requirements for

The Degree of Bachelor of Science

In Robotics Engineering

Date: 27 April 2017

Written By:

Yiğit Uyan

Project Advisor:

Michael Ciaraldi.

Abstract

Life is fascinating. Evolution, a fundamental concept behind life, is key to understanding how life as we know it originated on our planet. Computer simulations can be used to model the key aspects of evolution and to create artificial life for robotic agents in a simulated software environment. Through these simulations, we can better understand how random life is, perceive the key relations between the state of the environment and evolution, and find out the necessary factors for life to originate. Our main goal with this project is to create a simulation platform for evolution, combining the core principles of robotics with cutting-edge genetic algorithms.

Table of Contents

Abstract i
Introduction 1
Background 2
Personal Research 2
Work By Others 3
Design 4
Environment 4
Tiles 4
Objects 6
Creatures 6
Materials 7
Physics 8
Sensing 8
Actions 8
Genetic Algorithm 9
Gene Expression 9
Gene Mutations 10
Genetic Crossover 10
Simulation 11
User Interface 11
Data Collection 11
Results 12
Experiment 1: Prey vs. Predator 12
Experiment 2: Natural Selection 13
Future Work 15
Problem 1: City Congestion 15
Problem 2: Package Delivery 15
Conclusion 16
Acknowledgements 17
Useful Resources 18
Appendix 19

Table of Figures

Figure 1 - WPI Campus Map 2
Figure 2 - Weather Simulation 3
Figure 3 - Loaf Block Game of Life........ 3
Figure 4 - Braitenberg Vehicle 3
Figure 5 - Grid Based Environment 4
Figure 6 - Set of Biomes........ 5
Figure 7 - Various Maps........ 5
Figure 8 - Agent Environment Interaction 6
Figure 9 - Set of Creatures 7
Figure 10 - Various Sensors 8
Figure 11 - Basic Actions........ 8
Figure 12 - Complex Actions 8
Figure 13 - Simulated Genetic Sequence 9
Figure 14 - Gene Expression Sequence........ 9
Figure 15 - Genetic Variation........ 10
Figure 16 - Simulation User Interface........ 11
Figure 17 - Data Visualization Platform 11
Figure 18 - PvP Initial Layout........ 12
Figure 19 - PvP Population Data........ 13
Figure 20 - NS Initial Layout 13
Figure 21 - NS Population Data 14
Figure 22 - City Congestion Model........ 15
Figure 23 - Package Delivery Model........ 15
Figure 24 - The Islands........ 19
Figure 25 - Evolving Creatures 19
Figure 26 - Forest Map........ 20
Figure 27 - Desert Map 20
Figure 28 - Arctic Map........ 21
Figure 29 - Sea Map........ 21

Introduction

Humans have always been curious about evolution. Where did we come from; where are we going? It is these questions that drive us to research. Research on biology to learn about the building blocks of life. Research on ecology to understand what life needs to exist. Research on genetics to study how life changes over generations. Throughout the 21st century, we have devised remarkable ways to do this research. We can now sequence entire genomes of species and examine the effects of individual genes, or peek into the cells with precision microscopes and observe the genetic structures first hand. We use what we learn from these observations to create models of life and how it works. But these methods have their limits. Observations cannot tell us how life has evolved, what factors are necessary for it to exist or how it will be in the future. To answer these questions, we need a way to experiment with life as well.

EvoSim is built to be a platform to conduct experiments on life and evolution. We used fundamental ideas from robotics to model creatures as intelligent software agents. We built bio-inspired environments for these agents to exist in and interact with. Further on we built a genetic algorithm layer to simulate evolution of these agents, and finally built a data collection platform to gather insight on how life works inside a software simulation. In short, EvoSim is built to be the ultimate tool to simulate evolution in a software environment.

Background

EvoSim is a multidisciplinary project, where the ideas from robotics, genetics and computer sciences come together to model evolution in a software platform. We studied the fundamental knowledge of ecosystem biology to model environments, species evolution to create a genetic algorithm model, and multi agent systems to design an interactive agent-environment system.

The background is presented in two sections: The *Personal Research* discusses the past work and research we have done in related subjects prior to EvoSim. The *Work By Others* section goes over related areas of research, such as multi agent complex systems, cellular automata, and artificial life.

Personal Research

My personal research interest is using artificial intelligence to solve problems in complex systems. I have worked as a research assistant in the RAIL (Robot Autonomy and Interactive Learning) Laboratory at WPI, where I developed simulations to model real life maps in a software environment to solve navigation problems using machine learning. Some ideas from that research formed the groundwork for EvoSim.

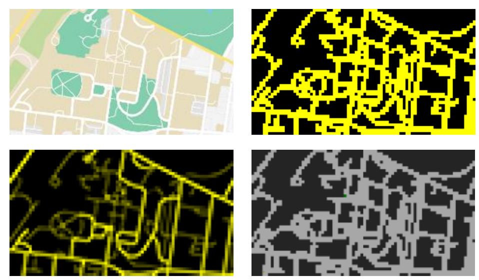
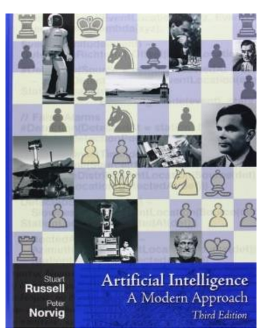
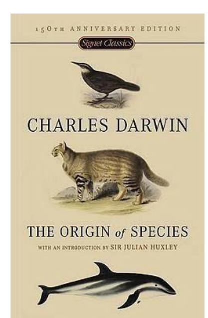


Figure 1 - WPI Campus Map

Aside from the classes I took during my studies on Artificial Intelligence and Data Mining, and a class on Plant Evolution, my research mostly consists of online Crash Course videos on Biology, Ecology and Genetic Diversity. My main source, both for implementation and ideas, are the following books:



Artificial Intelligence
A Modern Approach
Stuart Russell & Peter Norvig



Origin of Species
Charles Darwin

Work By Others

Naturally inspired simulations of complex systems are a popular study in academia. Research areas we investigated to get ideas for EvoSim are Complex Systems, Cellular Automata, and Artificial Life.

Complex Systems is a field of study, in which large systems are modeled using independent small parts, to accurately capture high level behaviors. Rather than using a simple model or equation, these complex systems are built on many small building blocks, sometimes on the order of millions. This usually outputs better results in capturing real outcomes, at the expense of higher computational demands. Geographic and Weather simulations are common examples in this area.

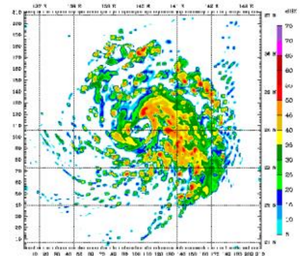


Figure 4 - Weather Simulation

Cellular Automata is a sub-category under complex systems, in which the system itself is governed by very basic rulesets to achieve life-like behaviors. These simulations are usually modeled on two dimensional grids where each cell can have multiple states (e.g., Dead or Alive). Each turn, the states of cells affect the neighboring cells, resulting in complex patterns across the entire grid. These simulations are thoroughly studied and divided into multiple categories according to the rulesets, initial layouts, and expected outcomes. One of the goals of cellular automata is to study self-replicating systems through basic structures.

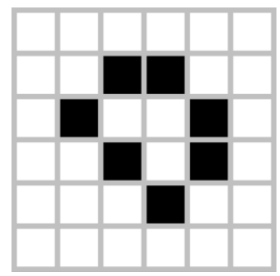


Figure 7 - Loaf Block Game of Life

Artificial Life simulations take the previous categories further by adding a layer of artificial intelligence to each building block. These simulations usually contain a naturally-inspired set of complex rules and feature a number of interacting agents in the environment. The algorithms range from reinforcement learners solving real life problems in a simulated environment, to genetic algorithms modeling real life dynamics in software, to neural networks for capturing life-like behaviors.

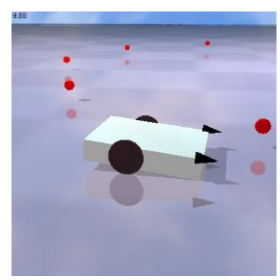


Figure 10 - Braitenberg Vehicle

EvoSim gets ideas from all three areas described above. We want our software to accurately model the real life as complex system simulations, be highly dynamic and result in complex patterns like cellular automata, and capture life-like behaviors in each agent like in artificial life models.

Design

The design of EvoSim focuses on four key features:

- Environment - Modeling Biomes
- Objects - Modeling Creatures
- Physics - Modeling Actions
- Genetics - Modeling Evolution

Each feature is modeled as a separate layer in code, giving high modularity to EvoSim. While building each layer, we followed the fundamental principles of robotics. Namely, on every turn, each agent should be able to sense their environment, take intelligent decisions, and act on the environment.

Environment

Environments are based on a grid based array of cells. Each cell contains a tile featuring a biome, a slot to contain an object (creature or material) and a set of neighboring cells to resolve physics. Every object in the simulation must exist in one of the discrete cells in the environment.

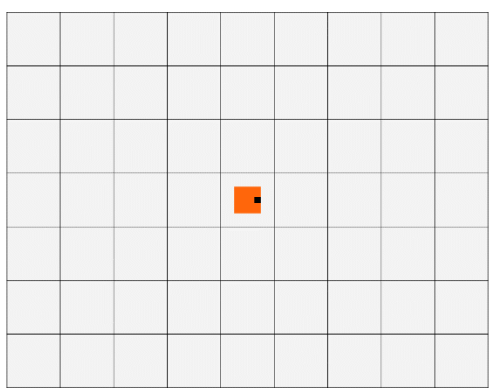


Figure 13 - Grid Based Environment

Tiles

EvoSim features a set of tiles to represent different biomes around the world. The tile of a cell has unique effects on the creatures standing on the cell, as certain actions can only be taken in certain tiles and certain creatures can only survive in certain biomes. Every tile has its native type of animals and plants (and corresponding spawn ratios). When the ‘spawn plants’ or ‘spawn animals’ option is selected, then the creatures of native types will start to spawn in these biomes. There are spawn rates for each tile, capturing the different fertilities of biomes, as well as a global spawn rate to the simulation.

Moreover, there are certain tiles representing dynamic biomes in the environment, such as flow (used to simulate rivers and sea currents). The effects of these tiles are modeled after real life physics and governed by the built-in physics engine in the simulation. Overall there are 20 biomes in EvoSim, capturing a range of different biomes around the world.

Figure 14 - Set of Biomes

Using the different tiles available in the simulation, we built real life inspired maps to simulate species dynamics in certain fauna. The ecosystems we modeled range from forests with lakes, to deserts with rivers, to islands in the Arctic. The detailed maps can be seen in the Appendix, Figures 26 – 29.

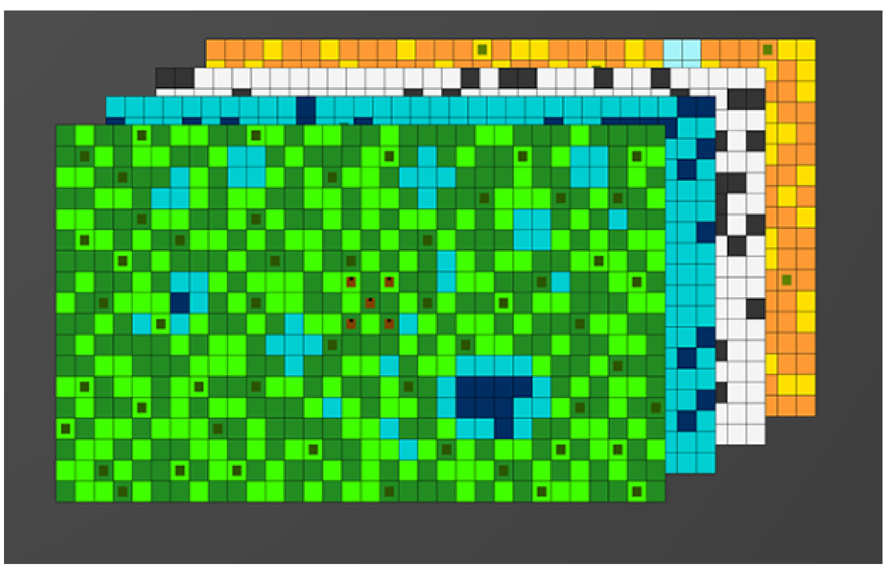


Figure 15 - Various Maps

Objects

The objects are anything used to simulate a physical entity in our simulation. Objects have types, such as a creature or a material, which further splits into categories, such as animals and plants, or minerals and chemicals. The main difference between creatures and materials is that creatures represent living entities and take actions every turn, while the materials are considered lifeless and are static.

Creatures

Creatures are the main agents for simulating life in our simulation. They are modeled based on the three fundamental ideas from robotics. Every turn, they scan the environment to gather information, use AI algorithms to come up with intelligent decisions, and take actions to interact with the environment. The interactions are resolved by the physics engine.

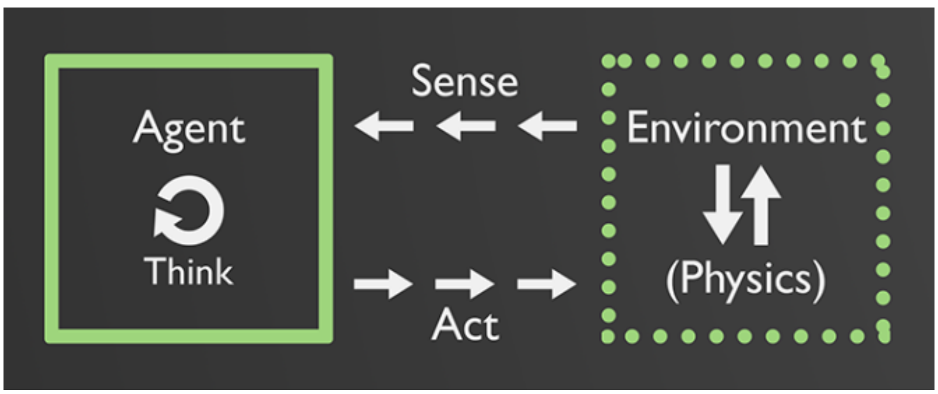


Figure 16 - Agent Environment Interaction

Each creature keeps track of the stats on its current condition, such as health, energy and age. Creatures start with maximum amount of health and energy they can have (specified in their genetic code) and further use these resources to explore the map and take various actions. Every action has an energy cost, as well as a duration, signifying that the creature cannot take any other action for that duration. If the health of a creature drops below zero (for example, when it is attacked), it leaves a corpse on the map which releases a portion of its remaining energy to the environment. When a creature's energy drops below zero as well, it (or its corpse) leaves the map permanently. The age comes into play on certain creatures. For simulations, we put an age cap on plants; they die when reaching that certain age. We disabled this for creatures, but it is implemented as a feature to further explore in the future.

We have created a set of 20 animals and plants with distinct qualities to use in our simulations. There basic creatures like bacterium and plankton, which we used to test evolution and genetics. These creatures act pretty much randomly and multiply by mitosis, each creating a genetically identical copy of itself. We then chose animals based on certain biome groups. Creatures like mouse and deer form a group of land animals, while the fish and shark can only survive in water-based biomes. There are amphibious animals, such as turtle and mouse. The other factor in choosing the animals was diet. Half of our animals are herbivores, while the other half are carnivores. Animals like Deer-Wolf-Bear and Fish-Shark form Prey-Predator relationships, which we further analyzed in test cases. The plants are chosen to go along with the biomes we have in our simulation. The complete list of creatures can be seen below.

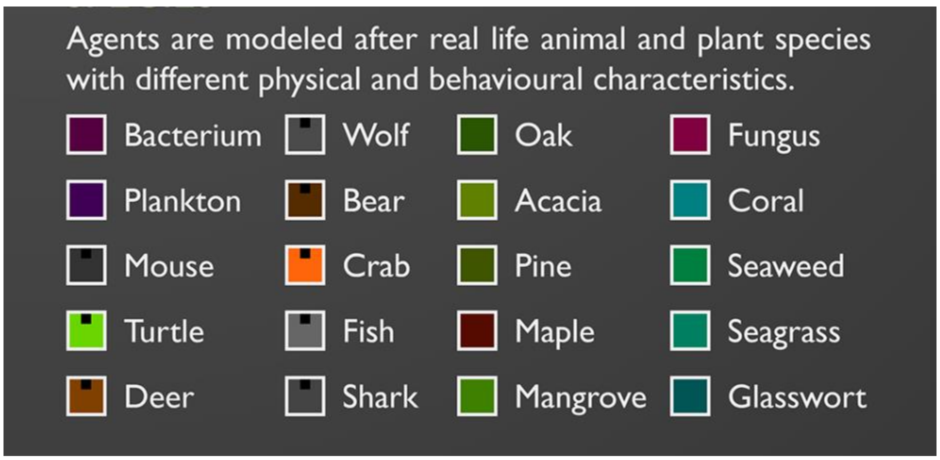


Figure 17 - Set of Creatures

We should note that the bacteria and plankton are modeled as animals, and fungi and coral are plants. This is done for convenience purposes and is not the case in real life. It is possible to add more creature types in the future to create a complete set of clades to model the phylogenetic tree.

Materials

Materials are based on lifeless entities in the world, such as a rock or a mineral. They do not take actions every turn, though they can deform over time. The two base types of materials we have in EvoSim are minerals and chemicals. They can act as static objects in the environment as well as provide food sources to certain creatures. We do not utilize materials in our test cases, however they are implemented in the simulation for future development.

Physics

The physics governs what creatures can see, where they can go, and which actions they can take in certain tiles. A sense-think-act model is built for creatures to interact with the environment.

Sensing

Sensors are defined during the creation of a creature. The first sensor we implemented in EvoSim is sight. There are different shapes and ranges for sight resulted by different eyes. Note that these parameters will change as the creatures evolve, as they are represented in the genetic sequence of a creature as well. When a creature is selected, its vision is displayed around it in a darker colored area.

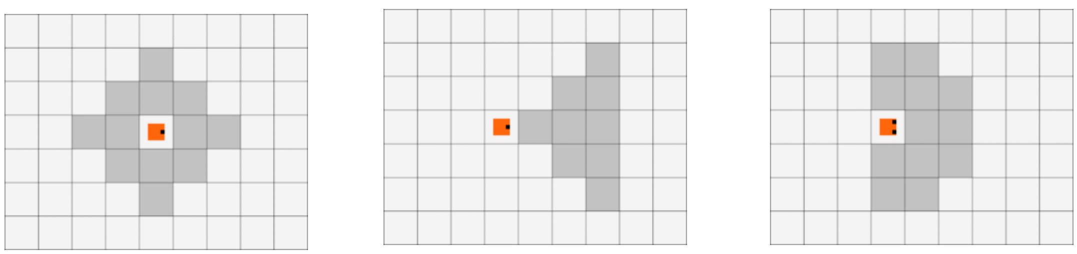


Figure 18 - Various Sensors

Actions

We have modeled different actions for creatures to interact with the environment. Every action has a minimum and maximum range of cells they can interact with, similar to sensors. These consist of basic directional actions such as move or swim, and usually more complex self-actions like eat and breed.

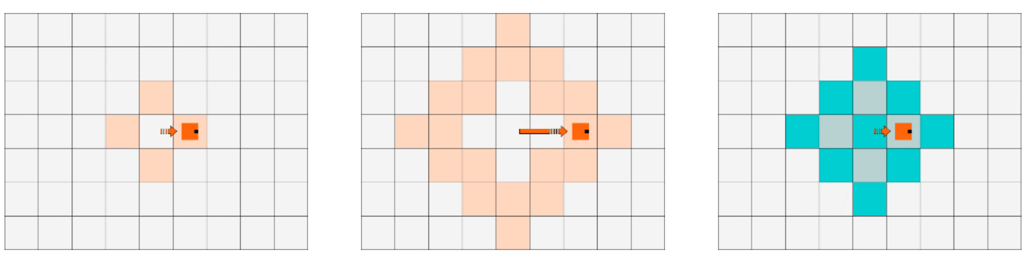


Figure 19 - Basic Actions

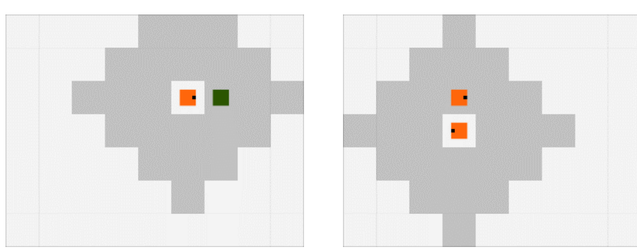


Figure 20 - Complex Actions

Genetic Algorithm

Many processes related to creatures and evolution are governed by genetic algorithms. A real life inspired genetics system is implemented, representing the visual and behavioral characteristics of creatures in a sequences of characters (genes) with values between 0 to F.

FF660B-082400-2227777979997979997-9988888...

Figure 21 - Simulated Genetic Sequence

There are 80 characters representing different qualities of each creature. This is a small number compared to the 20.000 genes in real life that give humans their physical and behavioral characteristics (or 2.000 for certain bacteria), but enough to capture the essence of life. Similar genes with similar functions are grouped into chromosomes, split by “-”. These chromosomes are read, then converted to arrays of numeric values, which later are parsed by the genetic system. As an example, the above gene sequence represents the DNA of a crab creature. The first chromosome FF660B is responsible for appearance, mapped to 1:1 to the RGB color space, giving the creature its orange (**#FF660B**) color.

It is possible to have creatures with more than one set of chromosomes, such as diploid creatures. One set of chromosomes for these creatures comes from a single parent (the mother) and the other comes from another (the father). While we did not use this in our test cases, the underlying genetic mechanisms for complex forms of breeding are implemented for future development.

Gene Expression

For creatures with more than one set of chromosomes, we created a genetic expression system.

HHHHHH-LLLLLL-RRRAAAARARRRARARRRAA-RRAAAAA...

Figure 22 - Gene Expression Sequence

This expression string maps 1:1 to the DNA string, featuring an expression (allele) for every DNA character. There are four possible characters for resolving conflicts in expression (for diploid creatures):

- H - The higher value gene will be expressed.
- L - The lower value gene will be expressed.
- A - The average of both genes will be expressed.
- R - One of the genes will be expressed at random.

Gene Mutations

The main driving force behind the genetic evolution system is gene mutations. At any turn, a random mutation can happen to a creature, changing one of the creature's genes at random. There is a global mutation rate, which controls the frequency of random mutations. Usually mutations have negative effects on creatures, and might even have deadly effects. Once in a while, though, a beneficial mutation might occur, increasing the chances of survival of a creature, which makes it more likely to pass its genes to future generations. The fundamental idea of genetic evolution states that, as generations progress, every new generation will be more fit to survive than the previous one. Also with more genetic variance, there will be more types of specialization across species to fill ecological niches.

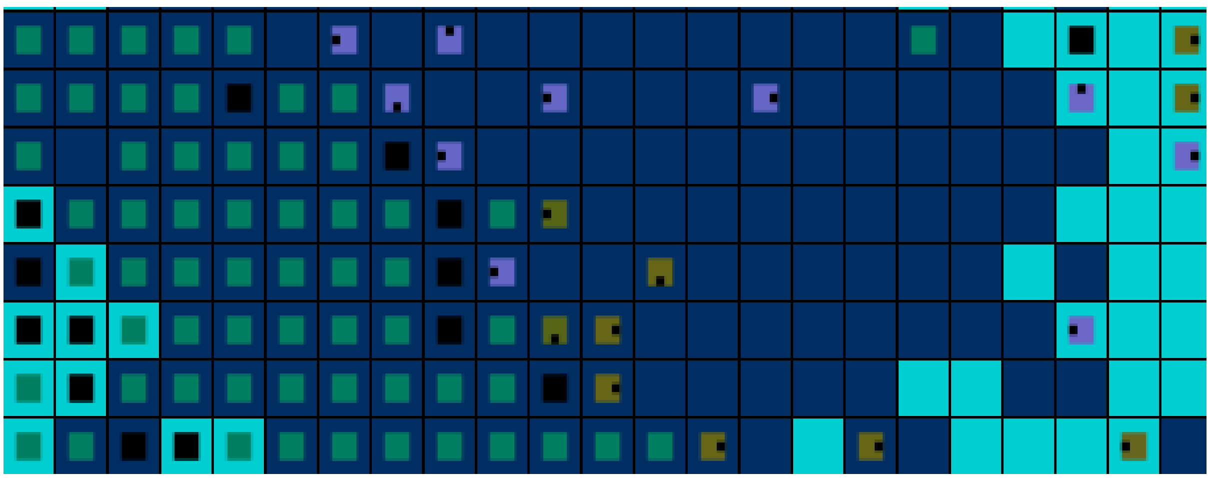


Figure 24 - Genetic Variation

Color is another feature that can mutate, though it does not have positive or negative effects on creatures. However, it is useful for the observer to track the evolutionary paths of the creatures. The image above features two local fish species with different colors, each evolved from the same ancestor a few generations before.

Genetic Crossover

Another way for nature to create genetic variance is the crossover. When new creatures are created, the genes which come from the parent can be shuffled to ensure that each creature is unique. We did not use genetic crossover in our test cases, but the features are implemented for future development. We intend to use this feature with creatures that sexually produce i.e., have (multiple parents), with the genes of each parent getting crossed, and one of the outcomes is selected to pass to the newborn individual.

Simulation

The simulation is the main component of EvoSim, which reads, initializes, and runs maps. The simulation can be played out creature by creature, turn by turn, or in free running mode. The main screen for simulation contains UI controls to display and run the map, as well as the controls to switch back and forth between the simulation and the data collection and visualization platform.

User Interface

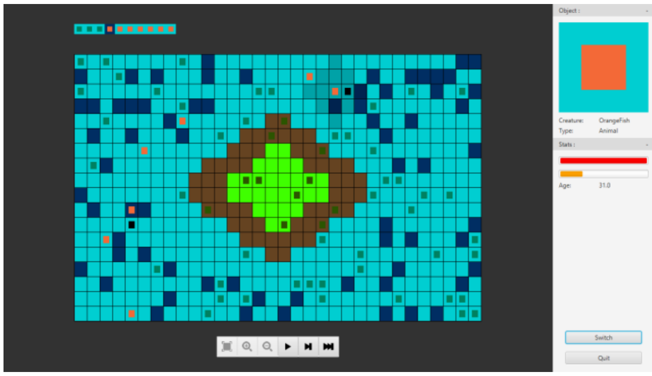


Figure 25 - Simulation User Interface

We designed the user interface in a clean and understandable fashion. The map in the center features the biome tiles and objects. Objects can be selectable; the sidebar on the right will display the detailed stats on the selected object. The UI also features a control panel (bottom), where the users can run the simulation in different modes. The upper left corner features the action order of creatures, and there is a turn counter in the upper right corner. Finally, the sidebar also features buttons to exit or proceed to the data collection.

Data Collection

The data collection platform is always active when a simulation is running. Users can switch back and forth between the simulation and the data visualizer to see relative populations of each creature.

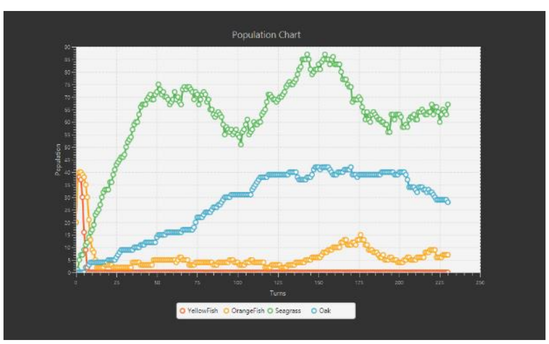


Figure 26 - Data Visualization Platform

Results

The goal of EvoSim is to simulate real life ecosystems and species interaction in a simulated environment. To test how accurate our model is, we chose to test it with two problems popular in ecology. The first experiment features two creature species, testing the prey / predator interactions between species. The second experiment is to test natural selection, where we study effects of genes in the species' survival.

Experiment 1: Prey vs. Predator

The first case we present is the predator vs. prey. For predators, we put two fish in the map in a central location. The initial prey population consists of 60 seagrass, laid out in a ring shape around the fish. The plants are to act as the initial food source, as well as the initial population to spread around the map.

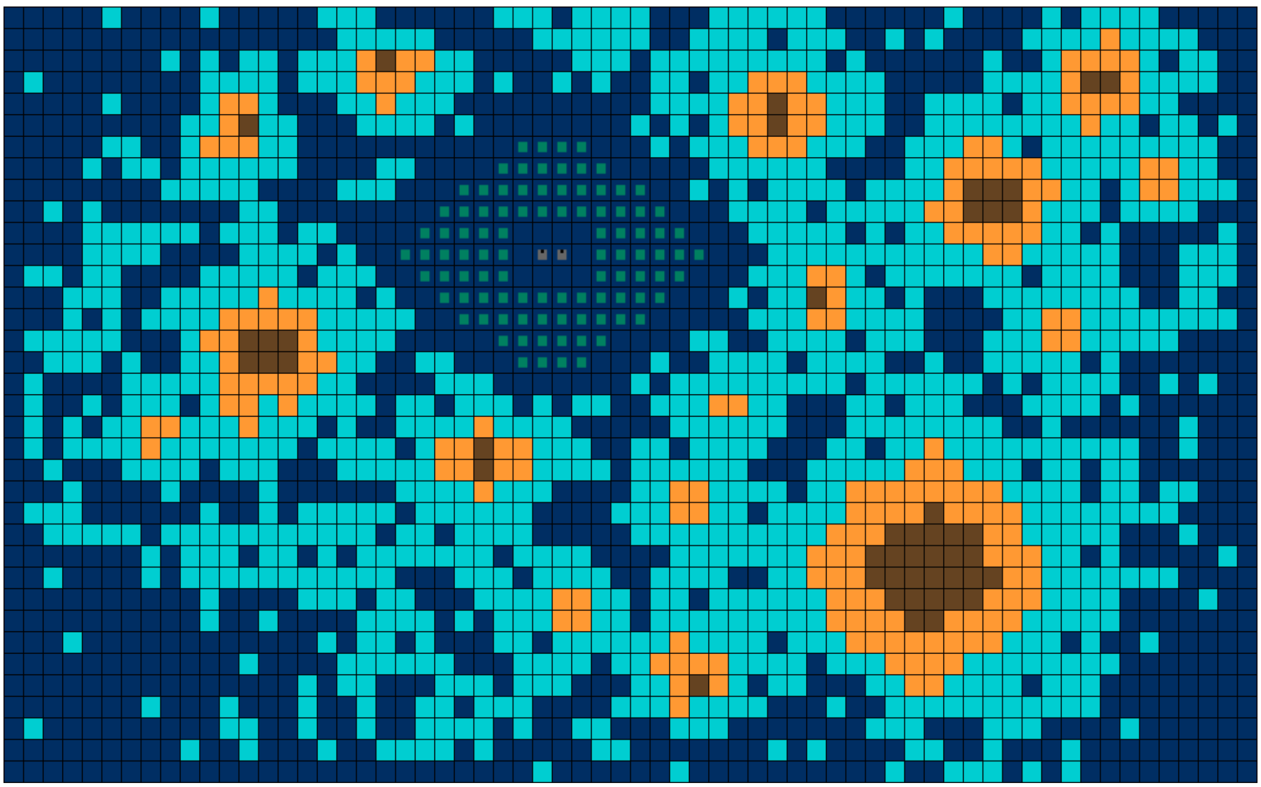


Figure 27 - PvP Initial Layout

Prey-predator dynamics is a thoroughly studied subject in ecosystem science. The populations of both species present interlocked sinusoidal curves, with a slight phase delay in the predator population. The size, frequency, and phase delay of the two relative curves depend on several factors, such as the initial layout of the map, genetic fitness of the creatures, the number of initial food sources, etc.

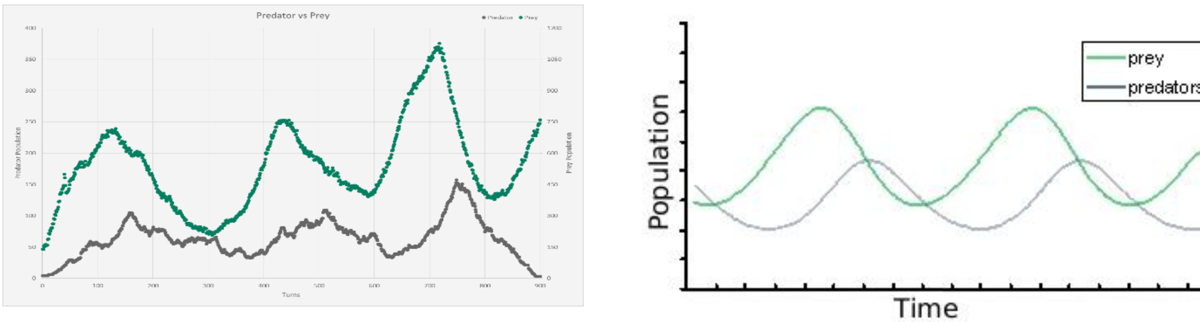


Figure 28 - PvP Population Data

The outcomes can be observed in the graphs above, which present the data collected from EvoSim (left) and a graph displaying ideal curves (right). The ideal model comes from Lotka-Volterra equations and is a widely accepted model for population dynamics in nature.

Experiment 2: Natural Selection

The second experiment is for natural selection. We have two fish species, orange and yellow. The map features a central island, which creates a separation and presents a dangerous terrain for the fish. Several plants are added as the initial food source, in addition to the random plant spawn across the map.

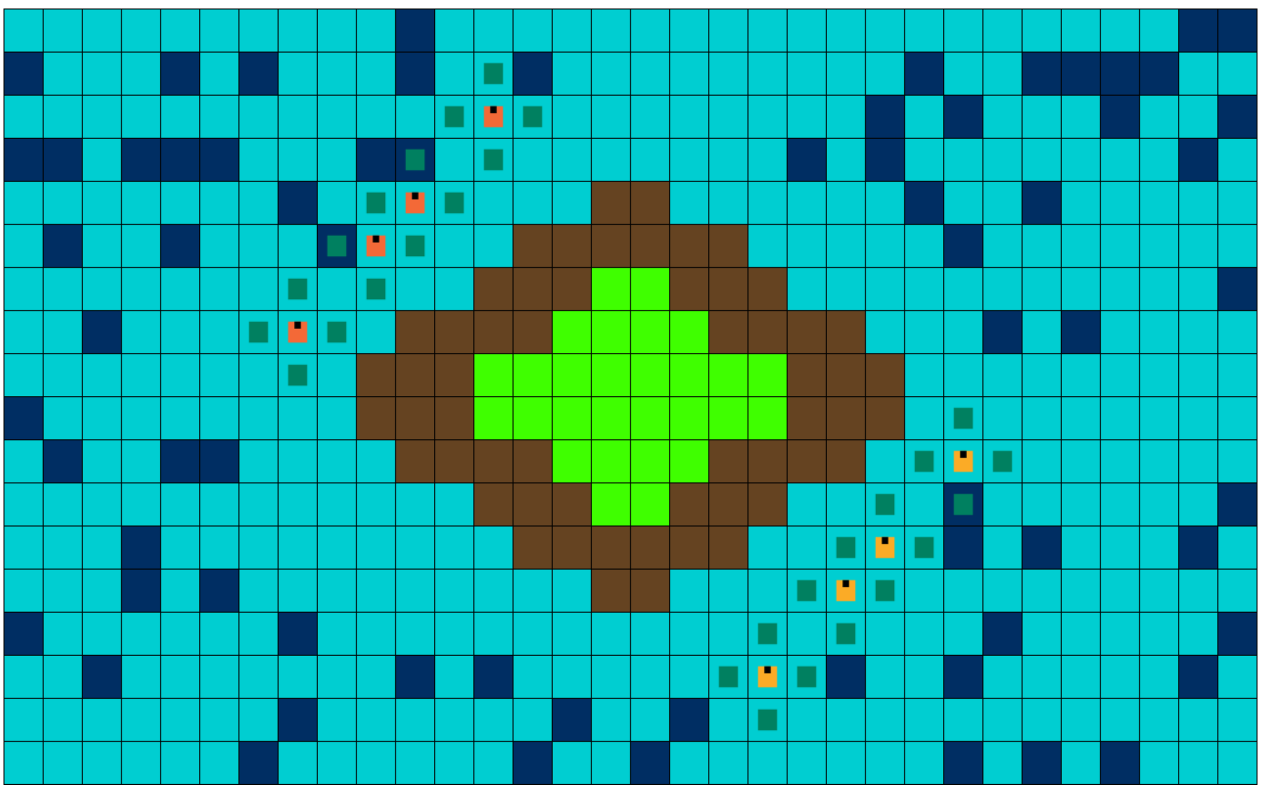


Figure 29 - NS Initial Layout

The entire genetic information for both fish species is identical, except for two key differences. The color genes are different so the observer can distinguish between the two as the populations progress. And there is an instinctive difference: the orange species can distinguish between land and water and try to stay away from the land. The yellow cannot make such a distinction, and so is indifferent to the type of tile. Over time, the changing populations of these species progress as follows:

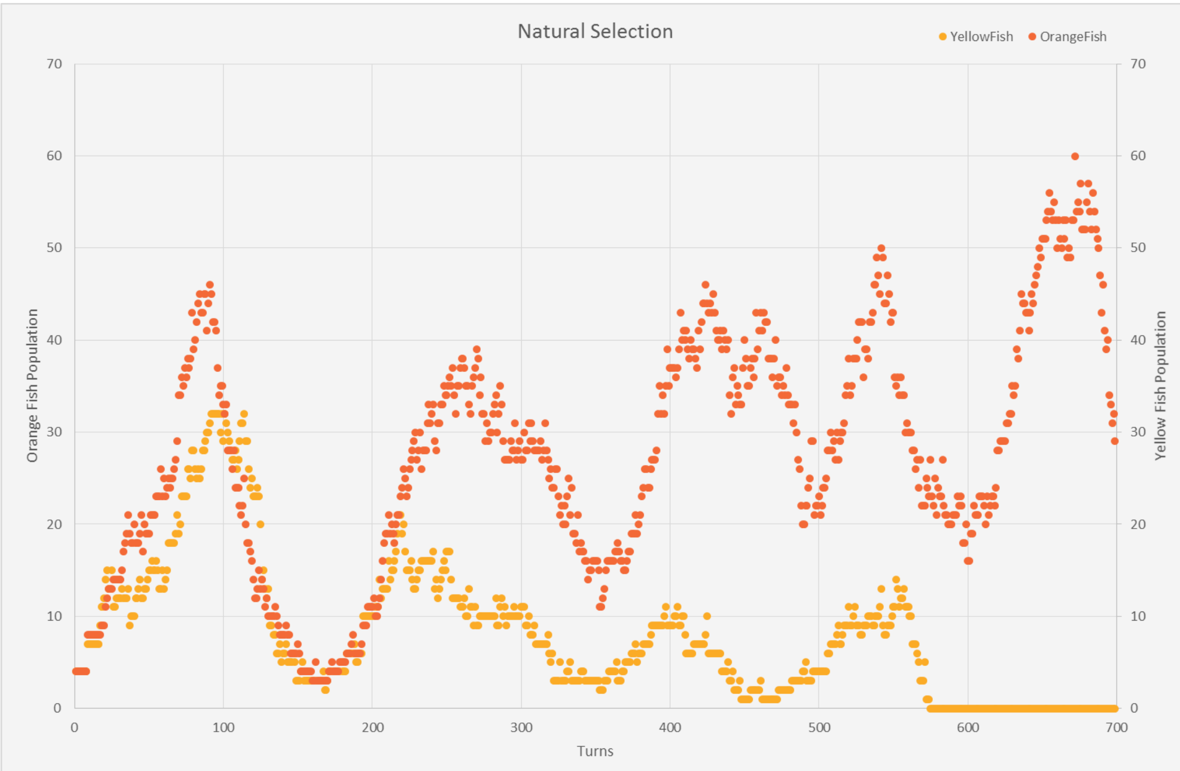


Figure 30 - NS Population Data

The sinusoidal curves are visible, as these species are in a prey-predator relationship with the spawning plant population (not shown). Moreover, the orange species thrive as the generations progress, while the yellow species go extinct over time. The experiment proves that a natural phenomenon like natural selection can be simulated in a software environment, and EvoSim can accurately output such results.

Future Work

While EvoSim targets key problems in biology and ecosystem science, in theory any complex system problem can be modeled in EvoSim. For this, we propose two popular problems in robotics.

Problem 1: City Congestion

It is possible to model city traffic in EvoSim. The cars can be modeled as creatures with special sensors and actions. Roads, houses and highways can be modeled as special tiles. The entrance to the city will be special tiles that will spawn cars, and the exits will be tiles that remove objects which enter them.

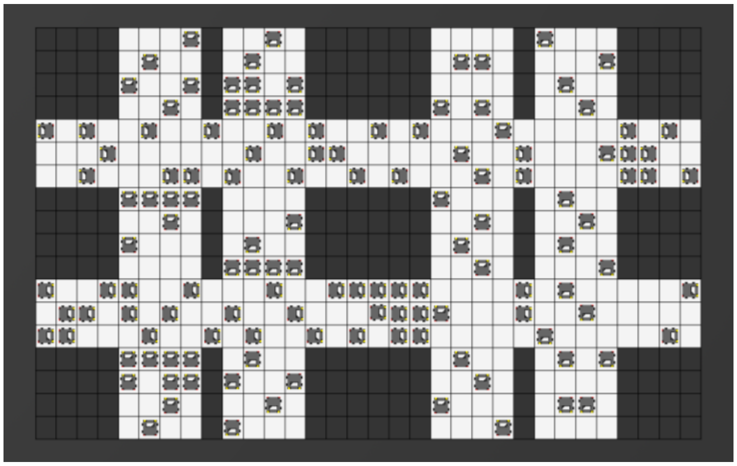


Figure 31 - City Congestion Model

Problem 2: Package Delivery

Another type of problem that can be modeled with EvoSim is package delivery with robots. Since the creatures in EvoSim possess strong AI over action decisions, it is possible to give creatures a global sensing mechanism (such as GPS) and let them solve navigation problems using evolving AI algorithms.

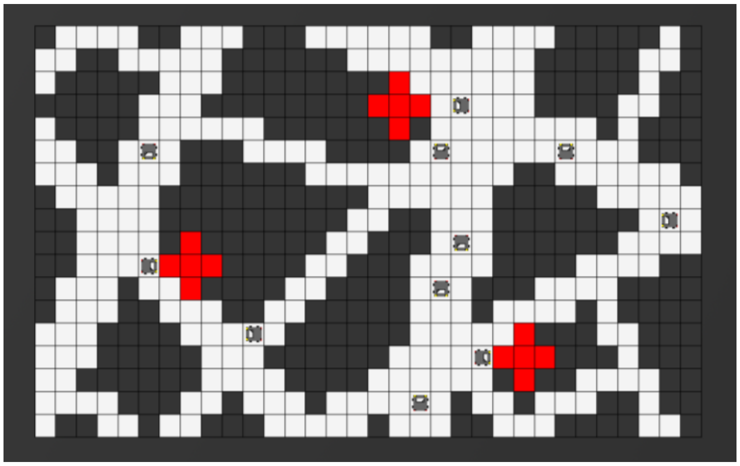


Figure 32 - Package Delivery Model

Conclusion

EvoSim is built to be a tool to simulate ecological systems and model species interaction, but in reality it is much more than that. It is a simulation platform with living environments, presenting living creatures, constantly changing with genetic algorithms. Every creature is unique in EvoSim, in a way to try to maximize its chance of survival with evolving DNAs and intelligence.

Our experiments have shown that EvoSim can accurately capture biological phenomena like predator-prey interaction and natural selection. But these are already well studied subjects in biology; what about questions for which we do not have answers yet? Maybe, someday, new discoveries can be made in EvoSim.

Acknowledgements

I want to thank Professor Michael Ciaraldi for being my advisor throughout this project, helping me to deal countless problems, providing valuable insight and ideas. Many thanks to Professor Sonia Chernova for teaching me everything I know in Artificial Intelligence. Finally, I also want to thank the vending machine in Atwater Kent building, which helped me to survive many sleepless nights building robots.

Useful Resources

http://genetics.thetech.org/

http://learn.genetics.utah.edu/

http://genetics.thetech.org/

Appendix

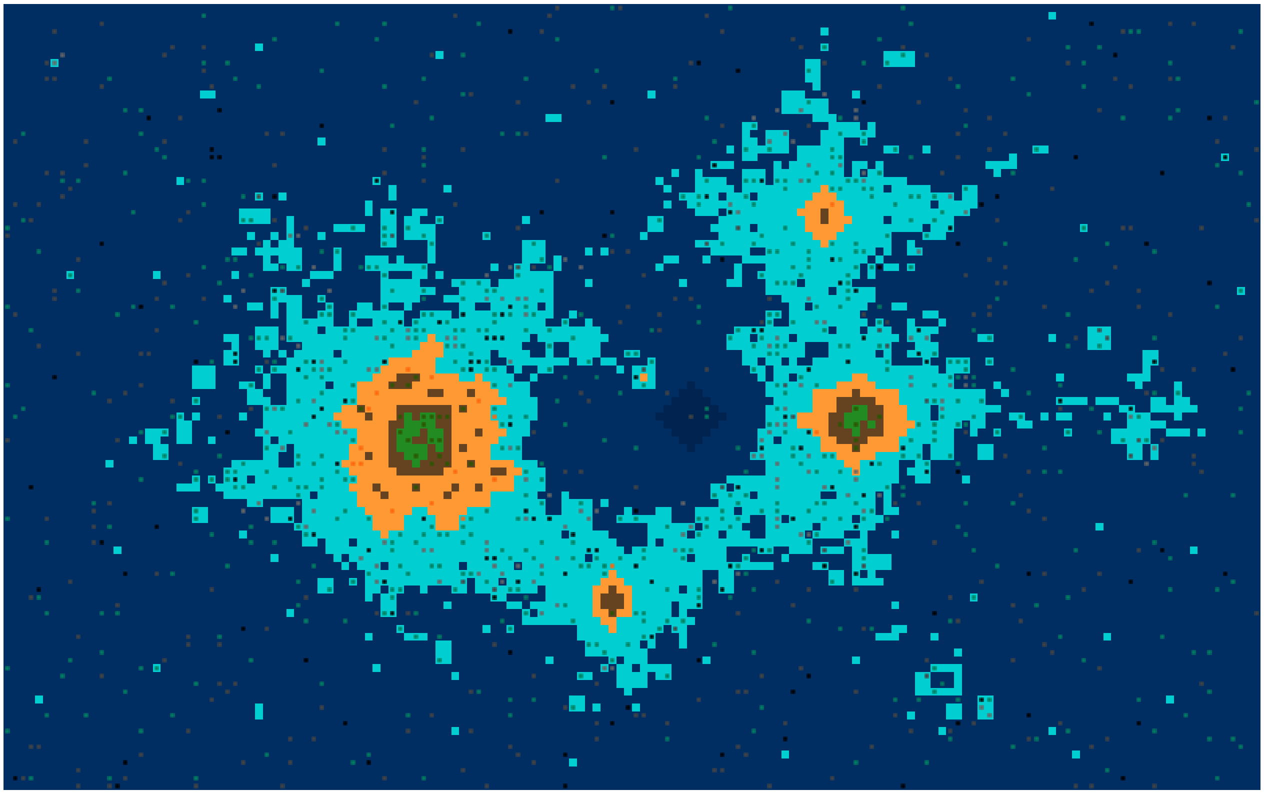


Figure 33 - The Islands

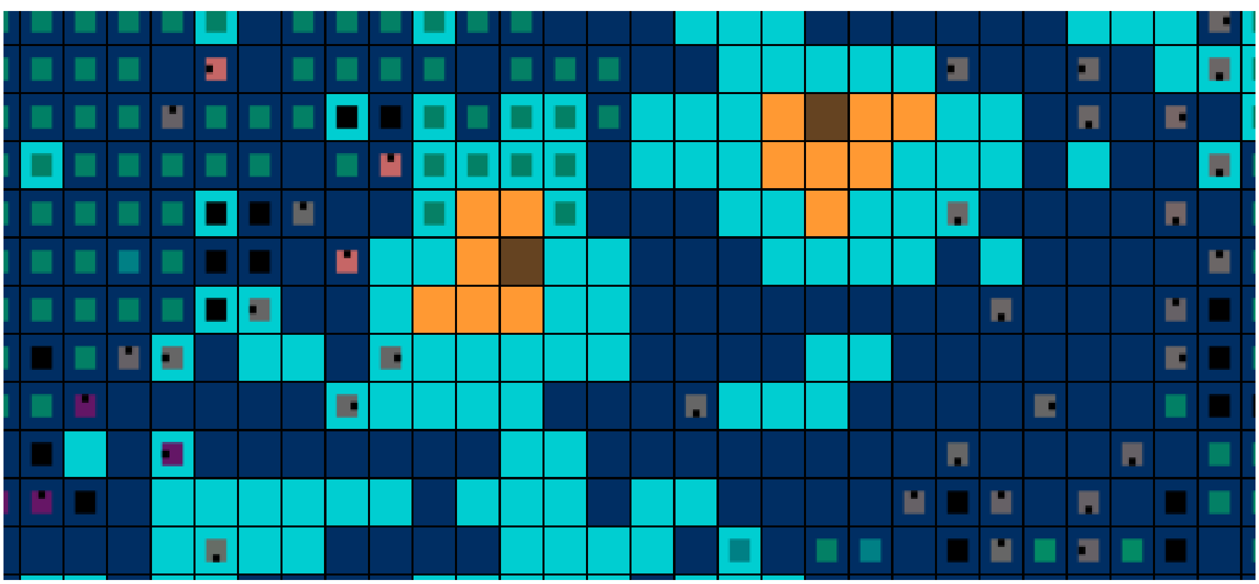


Figure 36 - Evolving Creatures

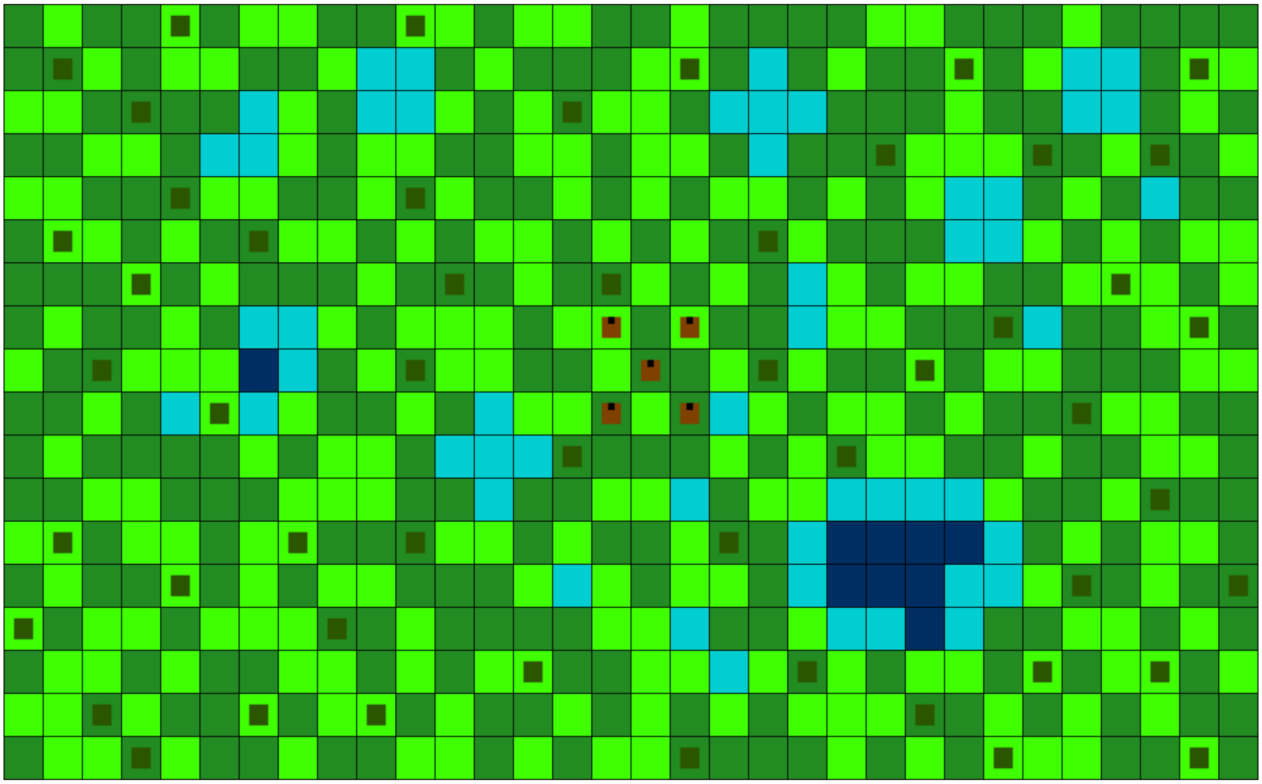


Figure 38 - Forest Map

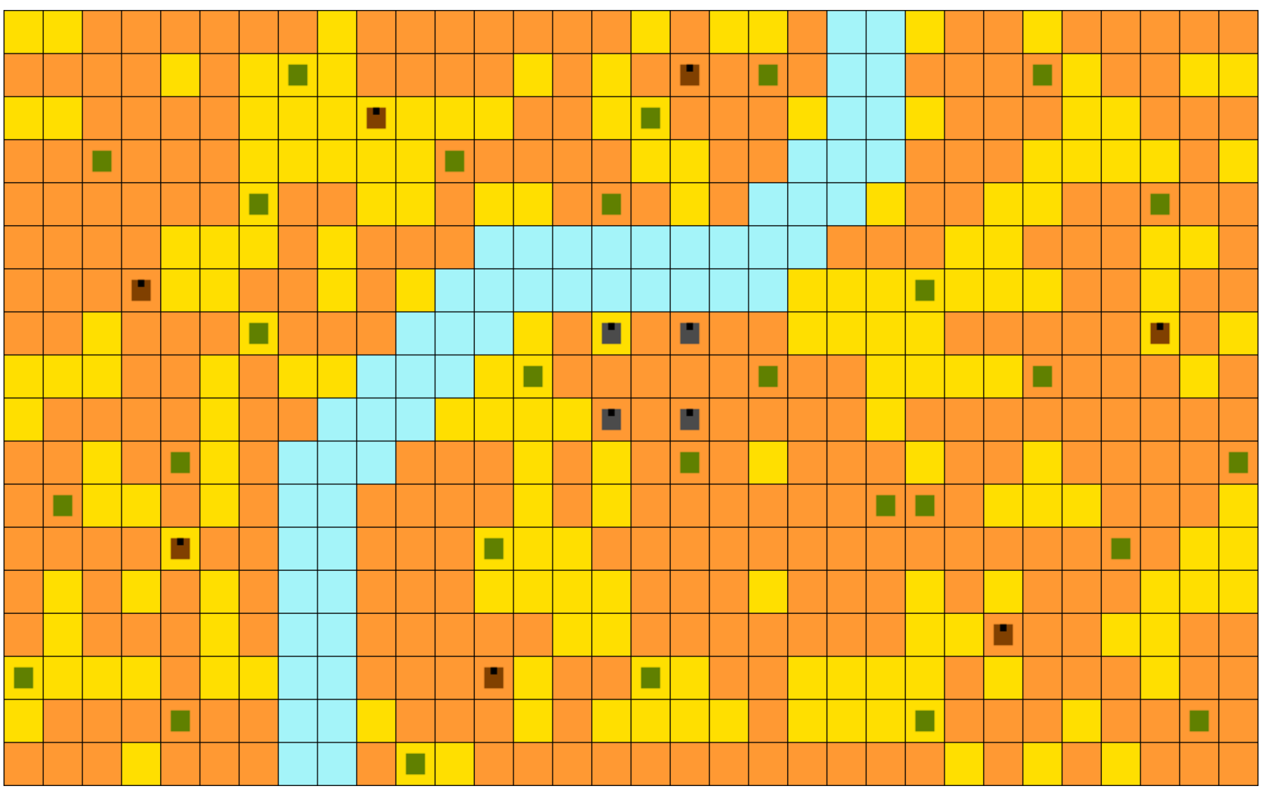


Figure 37 - Desert Map

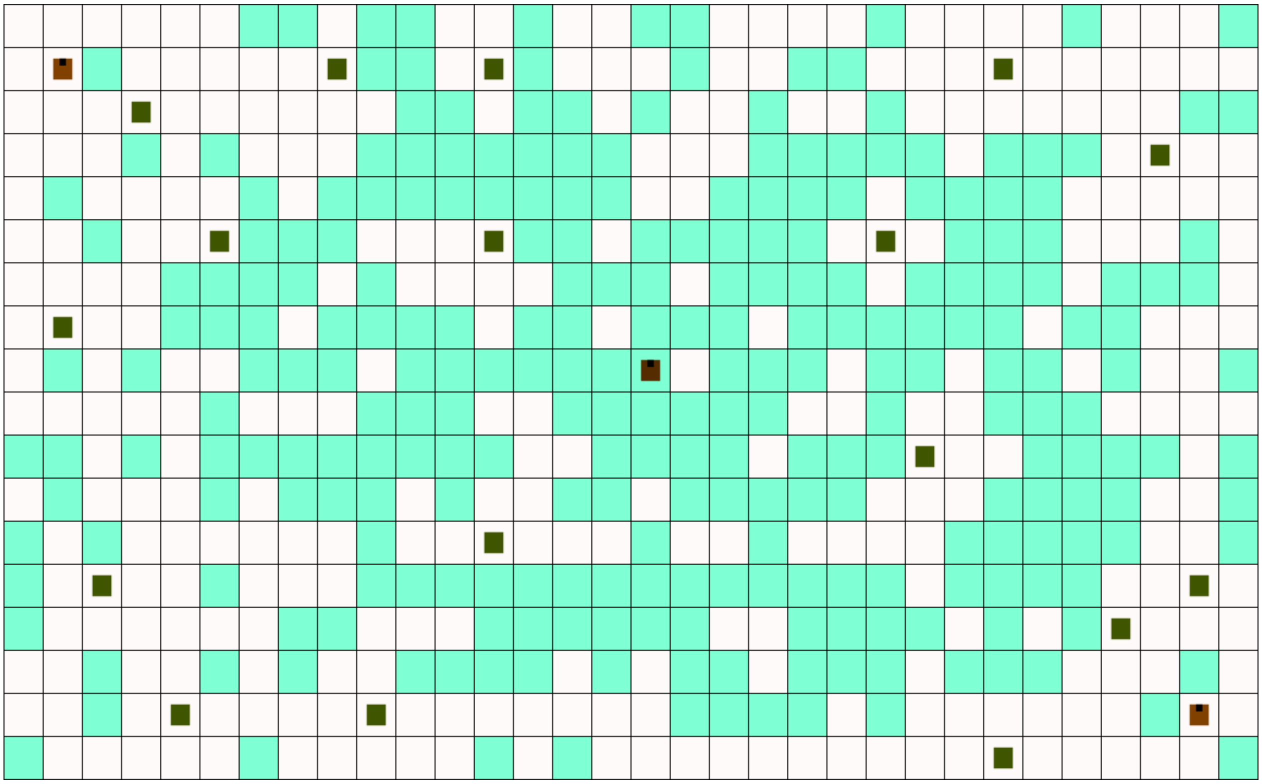


Figure 40 - Arctic Map

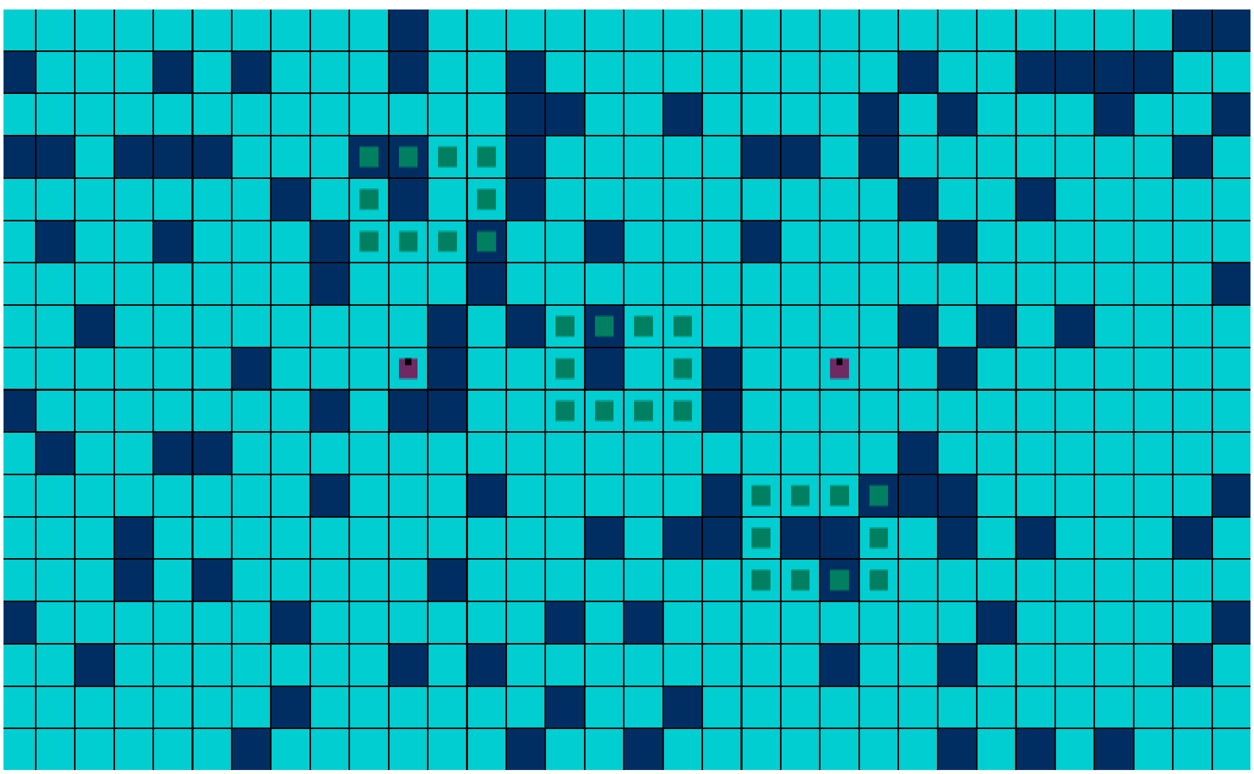


Figure 39 - Sea Map