



antarctic ocean life forms simulation



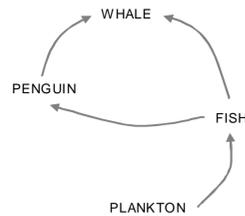
:: Problem Statement

Marine ecosystems are complex, multifaceted systems that play host to numerous organisms that form key components of nutrition for many communities around the world, as well as organisms that are vital to the stability of numerous food chains. Today's world is resource-scarce, and hence it is critically important for there to be an extensive knowledge base to enable our policy makers to formulate solutions that meet both short-term and long-term objectives.

Every ecosystem consists of interactions between the organisms that constitute the ecosystem. Indeed, in addition to the dependence and interdependence regimes between different organisms of separate species, organisms of the same species interact with one another to ensure collective survival. It is, therefore, integral to any computer modelling of an ecosystem that it encapsulates the food chain, as well as key characteristics of how life forms actually move in real life. That is, although the system should respond in feasible ways to changes in its state, there must be sufficient randomness to reflect the inherent chaotic nature of the world around us. Furthermore, in order to reach its full potential, the simulation must be easy to use, and allow for changes to parameters in order for experimentation to occur.

:: Solution Description

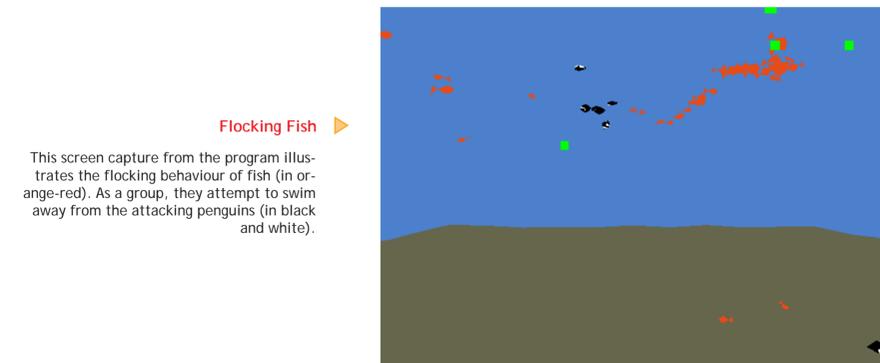
The virtual environment of the simulation has been set in the frigid waters of the Antarctic, an area that will be increasingly thrust into the spotlight as competing interests fight over access to the largest remaining unspoilt area of the Earth. Autonomous agents representing fish, penguins, whales and different types of plankton (the primary producers in the food chain) move around in the virtual three-dimensional environment. The interactions between different species are predator-prey relationships, with the life forms towards the top of the food chain acting to eat the ones further down, and the life forms towards the bottom reproducing to ensure survival of their species. Intra-species relationships include flocking and reproduction.



Simulation Food Chain

The different types of plankton are the primary producers in the system, and they are eaten by fish. Fish are the primary food source for the two animals above it, penguins and whales.

Because the simulation is designed to model the behaviour of the real world in order to understand the behaviour of real world systems, scientific accuracy was a concern of the developers in this project. Research into behaviour and characteristics of Antarctic marine life forms was carried out so that the formation of the simulated food chain would be representative, and so that the distinguishing characteristics of movement of each of the life forms would be maintained. For example, for fish, different methods of flocking were investigated, while for penguins, the ways in which real penguins may choose to come up for breath is reflected in the simulation.

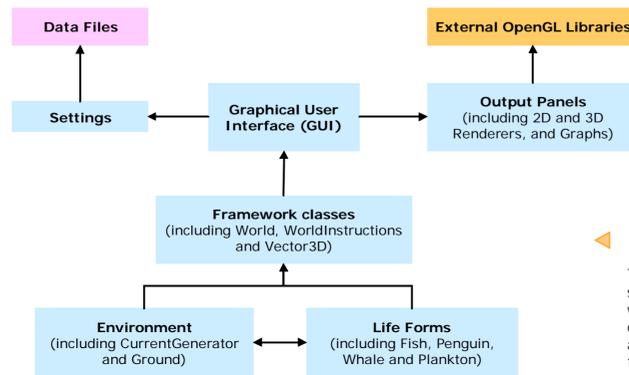


Flocking Fish

This screen capture from the program illustrates the flocking behaviour of fish (in orange-red). As a group, they attempt to swim away from the attacking penguins (in black and white).

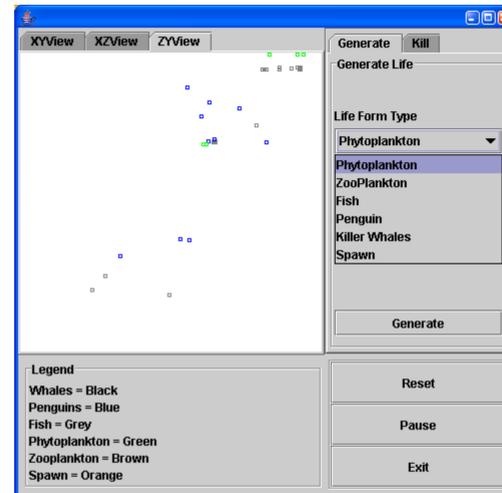
:: Strengths and key features

The solution, written in the Java programming language, utilises object-oriented principles, allowing for easy extensibility and adaptability. The project can be roughly divided into three sections. Firstly, the autonomous life forms are part of an inheritance structure, simplifying code production through common routines, and allowing for polymorphism in the next section, the framework classes. The framework classes control the simulation environment including the ground and current, communication between the life forms and the environment and vectors. The upper tier is the user interface, consisting of a main program executable, into which different panels can be inserted, such as the two- and three-dimensional renderers and the set of graphs. The three-dimensional renderer relies on the set of JOGL libraries (a Java implementation of OpenGL).



Functional Structural Diagram

This diagram represents the general structure of the simulation program, with the arrows representing primary data flow direction. Although this is not a class diagram, it visualises the separation between groups of physical and non-physical classes and dependency between different sections of the program.



User Interface Panel

The user interface panel allows the user to select between different two-dimensional slices of the simulation, as well as buttons to change the state of the simulation.

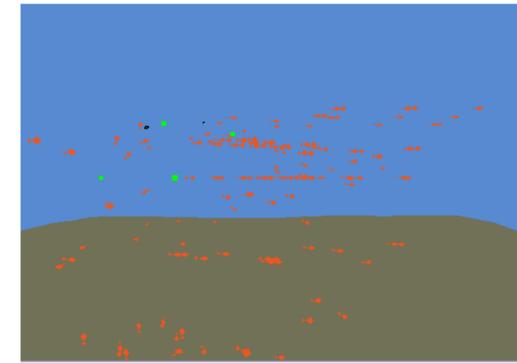
Upon initiation of the world, given a set of life forms and a description of the ground, for each time step (a "tick") created by the simulation engine, the world iterates through all of the life forms and allows them to make a decision as to which course of action they would like to perform. Instructions are passed back to the world from the life form; these include "accelerate" and "kill" instructions. The use of the instruction set was implemented to enforce strict rules about the behaviour of life forms, to ensure conformance with object-oriented design principles, and to keep the world's indexes accurate, because the life forms are prevented from changing their position and velocity directly. After the world has been updated by one time step, the output panels, such as the renderer and the graphs, are updated to reflect changes during that time step.

An important design principle that should be noted is that the life forms are able to move around freely using vectors for position and velocity, unrestricted by a grid-based simulation environment. This allows for the smooth movement of the life forms, and for the life form agents to have greater autonomy.

The 3D renderer is a key strength in the user interface component. This is because the user can visualise the scene more effectively by extracting information from both two-dimensional and three-dimensional renderers at the same time. This is because we are more accustomed to conceptualising three-dimensional scenes in our everyday experiences. In addition, the interactions between the life forms, such as flocking, which take place in three dimensions, is best seen using three dimensional output. Furthermore, because the 3D renderer uses OpenGL instead of a proprietary system, the simulation program is not restricted to being run on one operating environment.

Three-dimensional renderer

The three-dimensional renderer is able to render ground and a blue background representing water, in addition to complex scenes involving hundreds of life forms simultaneously.

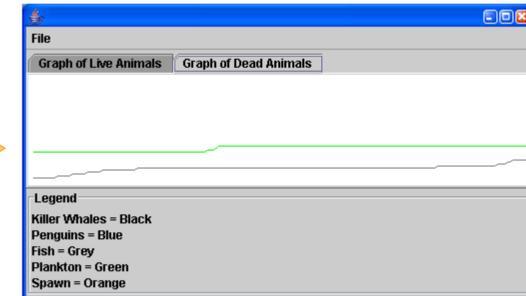


The World simulation environment class relies heavily on indexes to speed up execution. For example, the index that sorts life forms by position is relied upon heavily when the world reports to individual life forms about creatures near it. Furthermore, indexes are used to pre-sort each of the life forms in the simulation into their respective categories so that the rules regarding the order of execution (whales, penguins, fish, plankton and then spawn) are easily maintained.

Graphs showing the simulation environment over a period of time are available, allowing for study into population trends over a period of time. This is far more intuitive than reading off numbers representing the state of the system.

Graph of Live Life Forms and Cumulative Deaths

In this particular example, the number of dead fish (in grey) are rising because they are being hunted and eaten by hungry penguins. However, plankton (in green) not only die when eaten by fish, but the death of plankton is also reported when two clumps of plankton come together and merge.



The user interface is intuitive and functional. Graphs show the state of the simulation environment over a period of time, allowing for study into population trends as the population of each of the different species is changed; this is far more intuitive for the average user than reading lists of numbers. Furthermore, because the two-dimensional and three-dimensional renderers are running simultaneously, more detailed and quantifiable information can be extracted when conducting investigations. The user interface also provides buttons for pausing and resetting the simulation, which are important for productivity purposes, as well as commands for changing the number of life forms present in the simulation environment.

The world simulation environment class relies heavily on indexes to speed up execution. For example, the index that sorts life forms by position is relied upon heavily when the world reports to individual life forms about creatures near it. Furthermore, indexes are used to pre-sort each of the life forms in the simulation into their respective categories so that the rules regarding the order of execution (whales, penguins, fish, plankton and then spawn) are easily maintained.

Key features and strengths of the life forms in this computer model include **flocking** in fish and penguins, the **"porpoising"** style of rising up to the surface of the water to breath in penguins, and **reproduction** in fish.