Functional Description of the ALife System used by the *Evolutionz* Software

Dimitri Marinakis, May 18th, 2004

The "Artificial life studies 'natural' life by attempting to recreate biological phenomena from first principles…" (Sante Fe Institute, 1994).

1.0 Introduction

Evolutionz allows a user to construct, compare, observe, and explore dynamic artificial ecosystems through a 3D interface. As well as a tool for investigating learning and open-ended evolution, the program is essentially a computer game or a "software toy" capable of producing environments of endless variety. The inhabitants of those worlds evolve to demonstrate behavior leading to self-preservation and propagation (i.e. realistic behavior). This document describes the simulation details of the ALife system used by the Evolutionz software.

2.1 The Environment

The artificial environment is constrained to a finite rectangular two-dimensional grid with a boundary surrounding the outer edge. Five classes of objects inhabit this artificial word: artificial plants, artificial animals, corpses, energy sources, and blocks. At initialization, the world is populated with an arbitrary arrangement of these objects, and at each time-step, all the objects are simultaneously updated. Different microhabitats are modeled by varying the rate of growth of the artificial plants in different regions depending on the location of the energy sources. There are two types of blocks: "trees" which are randomly grouped in clusters, and "mountains" which are roughly distributed along generated curves. These blocks provide obstacles for the animals, and create boundaries that can isolate different regions. Corpses represent dead animals and provide food for certain types of animals. User-defined parameters such as *World Size* and *Number of Blocks* control a number of settings that specify the rules of the world.

2.2 Plants and Energy Sources

The plants follow a simple life cycle. They begin from a small "seed" and grow larger over time, reproducing only when a maximum size has been reached. As an individual plant grows, it inhabits an increasingly larger area of the environment. Reproduction occurs by spreading seeds to nearby regions. After a certain age, each plant is given a small chance of dying each update, and this chance is increased over time.

There are three types of plants. Each type has attributes defining their maximum size, reproduction range, speed of aging and preferred energy level. Each type of plant is given a preferred energy level and an energy tolerance factor. The likelihood of a plant growing or reproducing at each round is determined by the amount of energy received at that location compared to its preferred energy level. Each energy source radiates a randomly determined amount of "energy". A plant receives energy from each energy source in inverse proportion to the square of the distance between it and the source. The probability of a plant growing is determined by a normal distribution about its preferred energy level, with a standard deviation equivalent to its energy tolerance factor (Equations 1 and 2).

Several user defined parameters determine the rate of growth and the starting population size. *Start Number of Plants* determines how many plants are initially placed in the world, and *Plants Head Start* is the number of times the plants are updated before any animals are added to the world. This gives the plants a chance to get established before foraging begins. Finally, *Plant Growth Period* determines how often each plant is given a chance of growing.

$$\text{Energy Recieved} = \sum_{\text{Energy Sources}} \frac{\text{Energy Emitted by Source}}{(\text{Distance to Source})^2} \tag{1}$$

$$\text{Chance Growth} \propto \frac{1}{\text{Tolarance Factor}} \exp \frac{(\text{Prefered Energy} - \text{Tolarance Factor})^2}{2(\text{Tolarance Factor})^2} \tag{2}$$

2.3 Animals

Like the plants, the animals grow throughout their lives, reproducing only when a maximum size has been reached. The animals gain and lose energy through their interaction with the environment, which they are able to sense through three 'antennae' or 'eyes'. Each animal is created with an amount of energy that it received from its primary parent. Energy is only gained through the consumption of food and is lost through growth and motion acceleration. A small amount is also lost each time step regardless of what actions are taken. The older the animal gets, the amount of energy it loses every round is increased. When its energy decreases to zero, the animal is considered dead, is removed from the simulation, and might be replaced with a corpse depending on the user-define parameter *Leave Corpse*. *Start Number of Animals* determines how many animals are initially placed in the world during the start of a simulation.

Three different types of animals exist obtaining energy for growth and reproduction from different sources of food. The categories are: herbivores, which eat only plants; omnivores, which may eat plants, corpses, and herbivores smaller than themselves; and predators, which may eat corpses, and smaller herbivores or omnivores. The amount of energy that a herbivore receives for consuming a plant is determined by a parameter called *Herb Plant Mult*. An herbivore receives energy equal to the square of the size of the plant multiplied by this value. A user-defined setting called *All Plants Edible* determines if a herbivore can eat all plants, or only plants smaller than its self. A predator receives the energy of the animal or corpse it consumes multiplied by a constant called *Predator Animal Mult*. An omnivore receives the same energy as an herbivore from consuming a plant, but discounted by a multiplier called *Omnivor Plant Mult*. Similarly an omnivore receives the same energy as a predator from consuming a corpse or animals, but discounted by a multiplier called *Omnivor Animal Mult*. Currently, *Herb Plant Mult* is a user-defined parameter, while the others are set at compile time. Table 1 describes the values currently assigned to these parameters.

Table 1: Food Multiplers for the Different Types of Animals

Parameter	Value
Herb Plant Mult Predator Animal Mult	User Defined 0.9
Omnivor Plant Mult	0.5
Omnivor Animal Mult	0.5

When an animal reproduces, the last animal of the same species it made contact with becomes the "father" of the newly created baby. The attributes of each animal are passed down with mutation from the animal's parent or parents. These attributes determine at which energy stage the animal should grow, its maximum growth stage, how much energy it should give to its offspring when it reproduces, how far it is able to see, and who it may mate with. Table 2 gives a list and quick description of the inheritable attributes of the animals.

The first three attributes of Table 2 concern the manner and rate of growth and reproduction of the animal. *Growth Energy Quotient* determines when it grows based on its energy level according to the inequality expressed in Equation 3. To grow costs a set amount of energy, which also increases with the square of the size of the animal. Once an animal has grown to its maximum size, subsequent attempts to grow result in reproduction. The new animal is placed randomly near its parent, and it is given energy equal to the *Birth Energy* attribute of its primary parent. The parent's energy value is then reduced by this value. When an animal dies, the energy allotted to the corpse is equal to the amount of energy the animal spent on growth during its life.

The *Species Flag* is a thirty-two bit number used for enforcing speciation among the animals. This number is inherited only from the animal's primary parent. If the user-define parameter *Allow Speciation* is set, then the Hamming distance between this attribute in two animals determines if they are able to mate. In addition, this value can be used to give an estimate of the most recent common ancestor for any group of animals.

The *Vision Range* determines the range of each of the animals' "eyes". The sensing capacity of the animal is very primitive. One ray is traced out directly in front of the animal, and two rays are traced out at forty-five degree angles from the front of their body. The first object encountered by each of these ray traces is perceived by the animal at a strength inversely proportional to the distance of the object. Although this sensing method has no true biological parallel, it was implemented in this system because of the speed in which it can be simulated.

$$\text{Current Energy} \quad > \quad \text{Growth Energy} * (\text{Current Size})^2 \qquad (3)$$

Table 2: Inheritable Attributes of the Animals

Attribute	Description
Growth Energy Quotient	Dictates amount of energy required to grow. Reproduction occurs when maximum growth has been reached.
Birth Energy	Amount of energy given to offspring.
Maximum Size	Largest size obtainable.
Vision Range	Range of their sensing capabilities.
Species Flag	32 bit flag used for speciation.

2.4 The Neural Nets

The artificial animals are controlled by neural networks. At each time step of the simulation, intrinsic and extrinsic sensory information is input into the net and the animal's motion parameters for that time step are determined based on its outputs. The nets take 8 bit signed values as inputs and outputs (Tables 3 and 4). The first nine values are intrinsic information, which reflect the animal's current motion and internal variables such as current energy level. The other inputs reflect what the animal perceives with its primitive 'eyes', and its limited sense of 'touch'. The two outputs from the net control the animal's acceleration and its heading. Only the four cardinal direction headings are allowed.

Each net is made up of two output neurons and an arbitrary number of internal neurons. Each neuron contains a sigmoid operator and may have weighted input connections to any other neuron or any sensory input value. Hence recursive connections are allowed. The sigmoid takes two parameters that determine its steepness and its offset (Equation 4). In addition, the neurons have a temporal parameter, which dictates how many time-steps must pass before the output value is updated with respect to the inputs. The neurons also have an initial state which determines the starting value of their output and the starting time-step in their update cycle.

Those animals created during the initialization phase of the simulation are given nets for which both the topology and the weights are randomly determined. They are created in a cascade like manner as follows. A random number of internal nodes are added sequentially to an empty net. Each new node is connected to a random selection of inputs, which may include previously added internal nodes as well as its self. The output nodes are then added sequentially in a similar manner. Figure 1 gives an example of a randomly created net using this process.

Table 3: Neural Net Inputs

Input	Description
LeftHeading	127 if current heading is left 0 otherwise
RightHeading	127 if current heading is right 0 otherwise
UpHeading	127 if current heading is up 0 otherwise
DownHeading	127 if current heading is down 0 otherwise
XSpeed	proportional to horizontal component of velocity
YSpeed	proportional to vertical component of velocity

Energy	current energy level (normalized by max energy obtainable)
Age	proportional to current age
Size	proportional to current size
Touch	127 if collision with food object last time-step -127 if collision with non-food object last time-step 0 otherwise
For Each 'Eye':	
Food Sensed	0 if nothing detected, non-zero value proportional to distance to food (object edible to this animal)
Block Sensed	0 if nothing detected, non-zero value proportional to distance to block (object capable of blocking path of this animal)
Mate Sensed	0 if nothing detected, non-zero value proportional to distance to mate (animal of same species)
Danger Sensed	0 if nothing detected, non-zero value proportional to distance to danger (larger animal of species capable of eating this animal)

Table 4: Neural Net Outputs

Output	**Description**
Accelerate Forward	If output is larger than 15, accelerate forward in proportion to output value
Rotate Left or Right	Rotate left if value less than –15 Rotate right if value greater than 15

The net for each animal is inherited with mutation from its parent or parents. The mutations can be characterized as affecting the entire net, an individual node, or a connection. The likelihood of each mutation occurring is determined by a parameter. The new net is initialized with its starting values before its first update. A mutation at the net level adds a randomly created neuron, removes a randomly selected neuron, or switches the position of two neurons. If a node is added, some new random connections to it are also added to the net. The appropriate connections are removed in the case of a deleted node. A mutation at the node level either alters node parameters or changes the connections to the node. Each node in the newly created net has a chance of undergoing this mutation. A parameter mutation alters the sigmoid parameters, the initial output value of the neuron, the number of time-steps in-between updates, or the initial time-step of the update cycle. The sigmoid parameters and the initial update value can be altered by up to 15%, while the update values can be shifted either up or down by one time-step. A connection mutation either randomly adds a new connection to the node or removes an existing connection. A mutation at the connection level simply alters the weight of the connection up or down by 15%. Each connection in the new net has a chance of being altered.

During crossover, a copy of the reproducing animal's net is used as a template for the new net, but a randomly selected set of its nodes is replaced with nodes from the father's net. Mutation is applied afterwards.

$$\text{output} = \frac{1}{1+e^{-x}}, \text{ where } x = \frac{\sum_i \text{weight}_i * \text{input}_i}{\text{steepness quotient}} + \text{offset} \quad (4)$$

3.0 Conclusion

The goal of this project was to model the evolutionary process in a closed ecosystem. Although the simulation is relatively simple, complex behaviors and interactions between the inhabitants emerge "for free" through evolution without specific AI coding.