Evolving Motor Techniques for Artificial Life

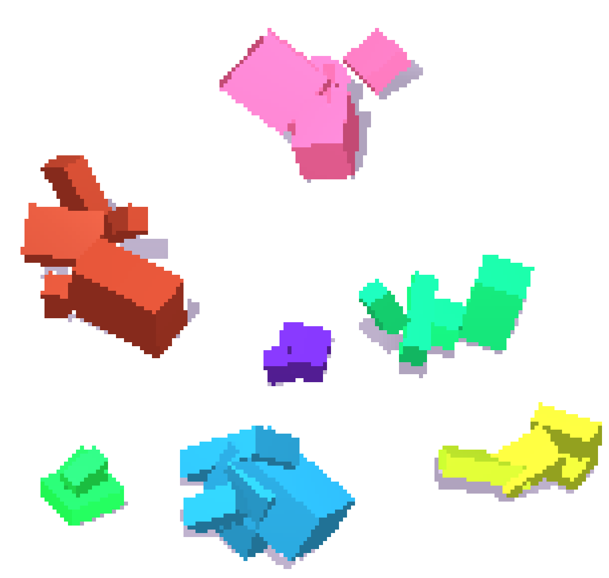
Kelley Hecker

TJHSST Computer Systems Lab 2007 - 2008

Abstract

I have created a program for simulating unique creatures in a 3D environment using co-evolution of the creatures' mental and physical structures. Creature data is stored in a one-dimensional genome consisting of various nodes for each physical body segment. The brain of the creature is controlled by neuron modification of sensor inputs. There is a system for converting genomes to physical representations to allow for physical simulation in the environment, and eventual selection of prime candidates through a genetic algorithm.

Examples of possible creature genomes. When each creature is displayed, the genome tree must be converted to an object.



Background

Related Research

Research done by Karl Sims is very similar to what I wish to accomplish. His work with evolving creatures led to a variety of organisms specialized in different areas and had very organic movements. The creatures were neuron controlled.

Yoon-Sik Shim and Chang-Hun Kim continued Sims' research and explored the possibility of flying creatures. They developed and explained a system of storing genomes as one-dimensional arrays.

Methodology

The simulation stores creatures in genomes in a way similar to Shim and Kim, however rather than being an array the structure is more like a tree. Creatures are controlled by passing joint-angle velocities received from sensors through neurons to produce joint velocity values. The data follows a circular pattern, moving between sensor, neurons, and effectors (output) each timestep.

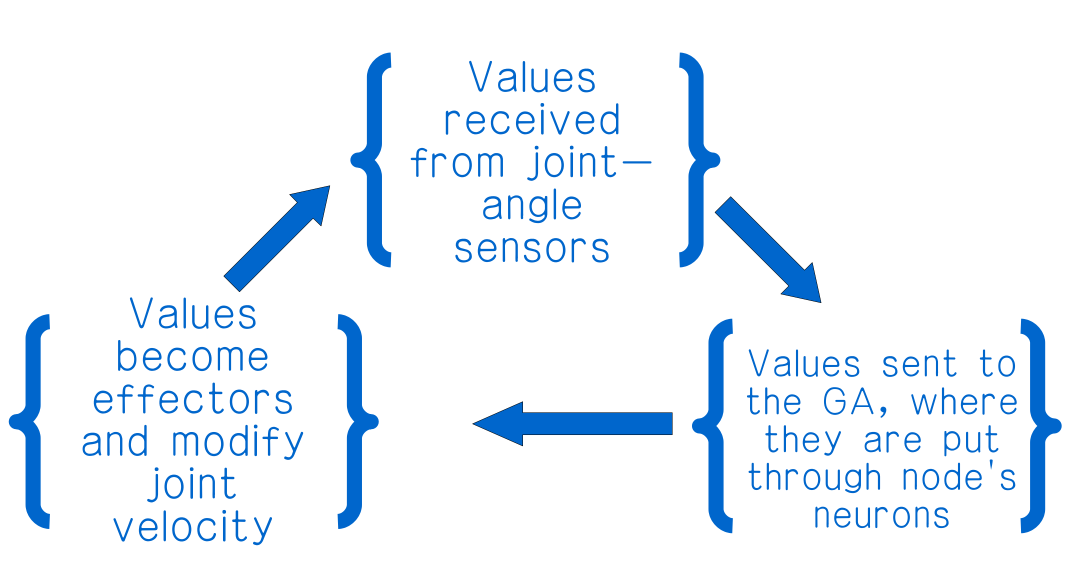


Image representing the circulation of data. Data originates from sensors measuring interaction with the environment, and then is modified by neurons within each node. Finally, the modified data is passed to the joint as velocity values. The cycle starts over again when sensors receive new data after the joints move.

Development

The entire simulation will be run by the Controller object. At the beginning of each simulation the Controller will create an array of genomes, and maintain this array throughout. It also displays the creatures in the physical environment and measures their fitness levels. After the fitness levels are compared, the Controller will manage the reproduction of the creatures and update the genome array with the next generation.

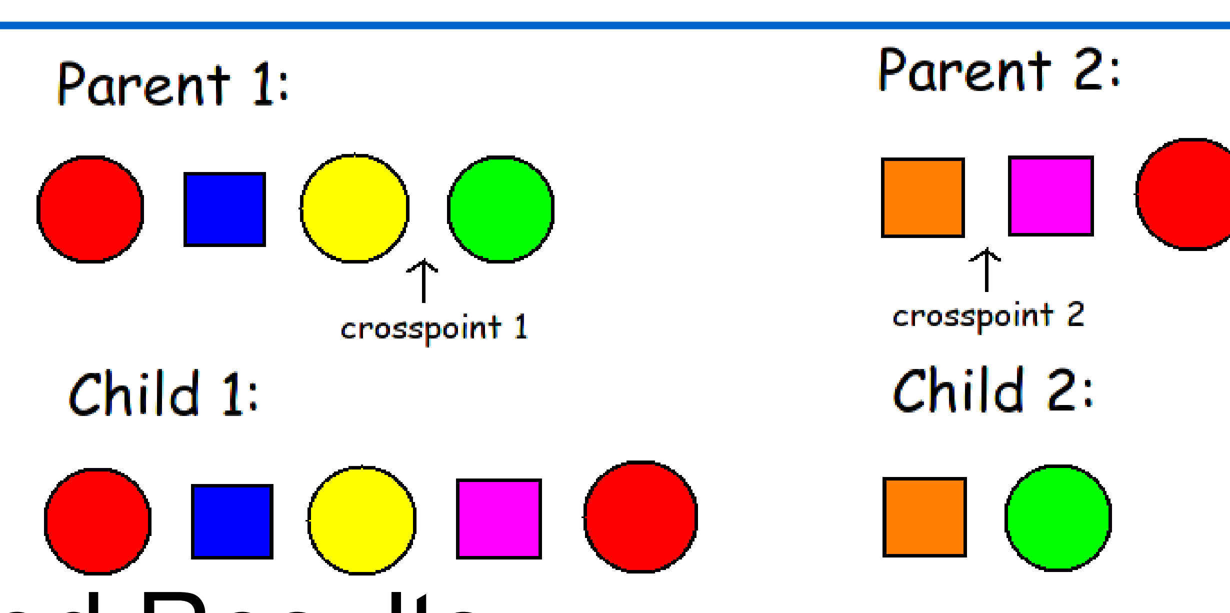
Each genome is make up of several nodes, each representing a body segment in the physical creature. The nodes store physical dimensions for the limb, a list of connected limbs or children, points where the segment connects to its parent and children segments, and the neurons which will control the joints.

At each time-step the joint-angle values for each node are measured and passed to the Creature Genetic Algorithm. This algorithm passes the sensor values through the neurons for that node and produces an effector, which will be the joint velocity. Possible neuron functions are sin, cos, atan, sum-threshold,sign-of, min, max, if, mem, saw-wave, log, expt, devide, interpolate, and differentiate.

Evolution

At the end of a generation, the best creatures are chosen based on their fitness value, which is how far they have moved since the start of the simulation. The top 20% of creatures are reproduced asexually (copied directly to the next generation). The remaining creatures are crossed over to produce new offspring.

An example of the crossover process between two parent creatures.



Expected Results

My final goal is to create a simulation that can create a unique variety of creatures with advanced motor techniques. Since creatures are stored as genomes it should be easy to allow for both mental and physical evolution.

There is also a possibility for specialized creatures. Different fitness tests could be implemented to select creatures which excel at different techniques, such as swimming or jumping.