Evolving Fish Behaviours

Adrian Boeing

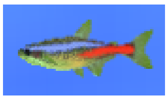
Abstract:

This is a small program demonstrating fish flocking behavior. The flocking implementation is fairly similar to (and was based on) Craig W. Reynolds original “boids” model. However, the flocking model has been modified to include predator avoidance, and seeking food. This simulation differs from most flocking systems in that the behaviors of the fish are evolved via a genetic algorithm.

The simulation:

The fish demonstration program animates three different types of fish flocking, and contains four basic entities:

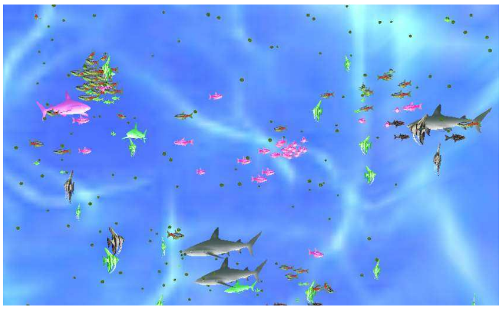
- Food – This is simply meant to represent various plant material which can be eaten by the small fish. It is distributed randomly within the “fish tank”



- Neon Tetras – The smallest fish, they eat the green food. They avoid the angel fish predators, but are unaffected by the sharks presence.

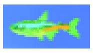
- Angel Fish – The medium sized fish, they eat only the neon tetra, and avoid the sharks.

- Sharks – The largest fish, they have no larger predators, and eat only angel fish.



Screen shot showing evolved fish movements. Note the individualistic behavior of sharks, and the tight schooling behavior of the tetras.

Each fish can be in four different states:



- Infant, the fish has only just been born, is not of a full size, and moves slower than adult fish. Infant fish are also incapable of mating. Infant fish slowly mature into adult fish, but mature faster if they are well fed. Infant fish are indicated by a green glow and smaller size.

- Adult fish, these are standard fish, move faster than infants and can mate if they are well fed.

- Mature adult fish, these are identical to adult fish, but have collected enough food to enable mating. These fish are indicated by pink glows.

- Dead fish, these are simply normal fish that have died, however are still edible by predator fish for a short period before they disappear.

It can take quite some time for the fish to evolve realistic behaviors (up to one hour). Generally you will notice that the fish move around rather randomly at the beginning, however after a few minutes they will begin to school. Given an hour the fish behave as you would expect. Given the original settings, you will generally notice the tetras acting intelligently the first, resulting in worse behavior for the angle fish, but improving the shark's behaviors. Eventually the angel fish behave intelligently, and you will begin to see the neon tetras flocking around the sharks for protection from the angel fish. This behavior was not explicitly programmed in, and is truly 'evolved'.

Implementation details:

Fish flocking:
A flocking model generates flocking behavior by analyzing neighboring fish movements and generating a set of movement vectors. The flocking algorithm considers the following forces:

- Fish 'cohesion' – This indicates the central point of neighboring fish movement. This enables fish to school if they desire this behavior.
- Fish 'seperation' – This indicates the vector the fish should move along to avoid collisions with neighboring fish.
- Fish 'alignment' – This aligns fish in the same direction. This enables a more flowing movement for schooling fish.
- Fish 'food' – This points fish to the closest edible item. This allows the fish to seek out the food it requires to survive.
- Fish 'mates' – This points fish to the closest fish that can also mate. This allows fish to seek other fish with which it can mate and generate new offspring.

Each of these vectors are then summed to generate a final fish direction value. Each vector is multiplied by a weighting value that determines how strongly each behavior affects the fish.

Finally if the fish is heading towards the fish tank walls it is given a wall avoidance vector to stop the fish from colliding with the walls.

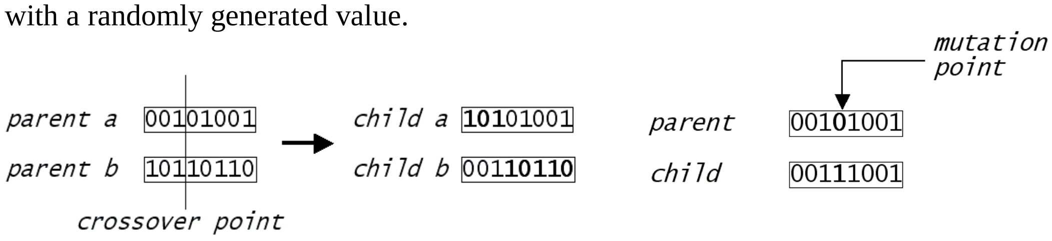
Each fish then moves in the direction of the final vector. Some scaling with past movement vectors is applied to generate smoother motions.

Genetic Algorithm:
The genetic algorithm (GA) is loosely based on Darwin's theory of natural selection. Conventionally a population of 'chromosomes' is maintained, and ranked according to a fitness value and then combined with other chromosomes to generate new offspring. Thereby theoretically increasing the overall fitness of the population.

Unlike the conventional algorithm this fish simulation improves the fitness of the general population of the species by 'terminating' unsuccessful fish (fish that have not eaten) and only reproducing with other successful fish (fish that have eaten plenty). As a result the fitness function and selection mechanism are essentially the entire simulation itself.

When a fish mates, it may either perform a 'crossover' operation, or a mutate operation. Unlike most GA's the operations actually perform byte-wise operators. (This is because the fish chromosome is encoded in bytes).

The crossover operation randomly selects a position within the fish chromosome and copies the starting section from one parent's gene with the ending section of the other parent's chromosome. The mutate operation simply replaces a randomly selected byte with a randomly generated value.



The fish chromosomes simply consist of the fish weighting vector values, and are encoded into a fixed point one-byte 4.4 representation for the genetic algorithm.

Enjoy!

Additional notes:

The flocking algorithm was originally developed for a 64kb intro entry available from www.scene.org:
http://www.scene.org/file.php?file=%2Fparties%2F2003%2Fbuenzli03%2Fin64%2Ffishtro2_final_1_0.zip&fileinfo

The graphics engine behind the simulation is not publicly available because it is in a very unfinished state. The engine used for this program is very out of date and is in fact my old test platform; I would not recommend copying the design.

The simulation requires DirectX 8.1 to run.

Update:

I have updated the program to provide acceleration control, and changed some of the fish's controlling parameters. The food model was also modified to produce dense clumps of food rather than randomly distributed food. This was done as problems arose with controlling the fish accurately when the fish controlled their acceleration. Adding a 2nd order integrator appears to have also helped the situation. The calculation for the separation distance was also modified. Code was added to allow fish to target only one specific food item, but was disabled. Code was also added for generating new fish from the existing fish gene pool, but was also removed, as this did not generally improve the "performance" of the simulation.