# TJHSST Senior Research Project
# Simulating Evolution
# 2007-2008

Tasha Wallage

October 31, 2007

**Abstract**

The main purpose of this program is to accurately simulate the genetic evolution of a species.It will attemp to do so using methods such as genetic mutation, genetic drift, and natural selection by means of both microevolution and macroevolution.

**Keywords:** genetic algorithms(methods of representation, methods of selection, methods of change), evolutionary computation, genetic mutation, genetic drift, optimal state, microevolution, macroevolution, Darwanism, natural selection, genetic variation, recombination, gene flow***, speciation***, ***maybe (if time)

## 1 Introduction

### 1.1 Rationale

The computer will simulate an environment and the user can modify that envirnment. Modication of an organism environment can force it to adapt and in essence, better survive. Because of this, those that are not cut out for the change will die off and those that are will live on and reproduce, thus creating a genetic drift in the species. This is the theory. According to this theory, one should be able to predict the changes in a species genetic make-up due to a change in its environment. If the environment becomes hotter, those creatures with higher temperature tolerance should be less affected than those without a high temperature tolerance and so one would expect to see the species evolve to have a greater tolerance to higher temperatures.

### 1.2 Purpose

Evolution often thought of as the changes that occur in an organism to better suit them to their envirnoment. However, this is not completely true. Evolution occurs in both positive and negative directions. It is completely random and the result could be in favor of the organism or it might not. I am trying to simulate evolution and track the change in a species' traits to better understand how evolution really works.

## 2 Background

"Genetic changes do not anticipate a species' needs and those changes may be unrelated to the selection pressures on the species. Nevertheless, evolution is not a fundementally random process."[3]

1. Mechanisms that Decrease Genetic Variation[1]

   (a) Natural Selection

   Natural selection was introduced by Charles Darwin. It is when the frequency of the more prolific members of a species increases because they are better adapted to the environment.

   (b) Genetic Drift

   This occurs when the allele frequency changes (can allow mutant alleles drift into fixation).

2. Mechanisms that Increase Genetic Variation[1]

   (a) Genetic Mutation

   This occurs when the gene sequence altered because the copy of "DNA" is corrupt.

   (b) Recombination

   This includes crossover of genes from the mother and the father to produce genes of the child(gene shuffling).

(c) Gene Flow

Gene flow occurs when genes drift into a population from a different population through mating.

3. Types of Evolution[1]

   (a) Macroevolution

   Macroevolution includes speciation, or the seperation of one species into two. It is an evolutionary change at or above the species.

   (b) Microevolution

   Microevolution is evolutionary change below the species level.

4. Types of Genetic Variation[4]

   (a) Variation Under Domestication

   (b) Variation Under Nature

# 3 Development Sections

## 3.1 Requirements, Overview, Limitations, Development Plan

The minimum requirement for this project would be an environment that includes the species under evaluation. There obviously must be a species to study and that species must have traits that can be altered, affecting its ability to survive. The environment must also provide a regenerating food source for the organism being studied else it will not live long enough to produce various generations leading to evolutionary results.

The project's main limitation is time. I am not sure how far involved my program will be because it really depends on how many problems I am faced with while creating the simulation and how easy they can be resolved. If I do manage to get pretty far to the point I want to include gene flow and speciation, I may find myself limited by the processing speed of teh computer. This may also be a problem in the earlier stages if the population of a Species becomes too large and the time taken to calculate each Species' actions becoms too much to handle.

Software

1. Java
2. Java3D

## 3.2 Research Theory and Design Criteria

Methods

1. Food Methods
   - breed()

     The Food class has a simple breed method because it is asexual and does not have a limitation on the number of times it is able to breed in one lifetime. Food breeds at random, but only if there is room for offspring around it.
   - act()

     The act method merely consists of randomly determining whether a Food will breed and/or whether it will die.
   - die()

     The die method acts randomly, generating a number between one and zero. If that number is below the given deathRate of the Food, it will die and the World will remove it from the Species HashMap.
2. Organism/Predator Methods
   - act()

     This method determines the next action of the Organism or Predator depending on its age. For example, younger and older Species will not be able to breed and will focus more on eating.
   - move(Location loc)

     This method moves an Organism or Predator one step closer to their target (which may be food or prey).

- breed()

  This method calls the crossOver method to determine the genetic code of the child with respect to that of its parents. If there is no space to breed or the Species is too young or too old, it will not have access to this method.

- crossOver(int trait1, int trait2)

  This method takes a trait from the father and a trait from the mother and produces a trait that is equal to one or between the two. If it is mutated, it might not reflect the traits of either the mother or the father.

- die()

  Same as Food's die().

3. World Methods

   - step()

     This method calls act() on each of the Species located in the World (stored in a giant HashMap).

   - getClosest(String type, Location loc)

     This method return the closest specified Species to the Location given. This is useful when an Organism is searching out food, or a Predator is hunting an Organism. It goes throught the HashMap of all the Species in the world, comparing the distances from the Location given and returning the specified Species with the shortest distance from the given Location.

   - add(Species s)

     This adds a Species to the World and is called by the breed() method in each Species when it gives birth to a child. add(Species s) also tallies the amount of each Species that is present in the World, so when a Species is added, the number count for the added Species (Food, Organism, Predator) is increased by one.

   - remove(Species s)

     Simular to add(Species s), this method removes the given Species from the World and is called by the die() method of each Species. It also subtracts the count of the given Species by one.

- aggAve()

  This method calculates the average aggression values of the Predators living in the World.

- defAve()

  This method calculates the average defense values of the Organisms living in the World.

4. Sim Methods

   - actionPerformed(ActionEvent e)

     This method is called repetitively by a Timer that is fired faster than the eye can see. This method calls step() on the World and then draws every Species, specific to their type, giving the simulation animation.

Algorithms

1. Process for Recombination

   The process for creating a new organisms with a new combination of genes mixed from its parents (and sometimes randomly mutated) takes the traits from both parents and gives the child a trait that is either equal to one of the parents, or is a mix of the two (something in between). The assignment of the trait is semi-random.

2. Randomization for Mutation

   The process by which genes are mutated is completely random. In fact, it is double random because the swapping of genes is random and the chance that it is mutated is also random.

## 3.3 Testing and Analysis

There have been many tests done to the program because it is so fragile and succeptible to errors. Currently, I have been testing the functionality of the program through system outputs. This includes running the program with only Organism and few Food and individually tracking each one to make sure they are working correctly. Then, I run the program again, but with many Organisms and Foods keeping track of the average values of traits to see if they are changing or remaining the same. If they are the same, the program is not working properly and if they are changing by too great of a degree or too rapidly, there is also a problem.

## 3.4 Visual Representation of Data and Results

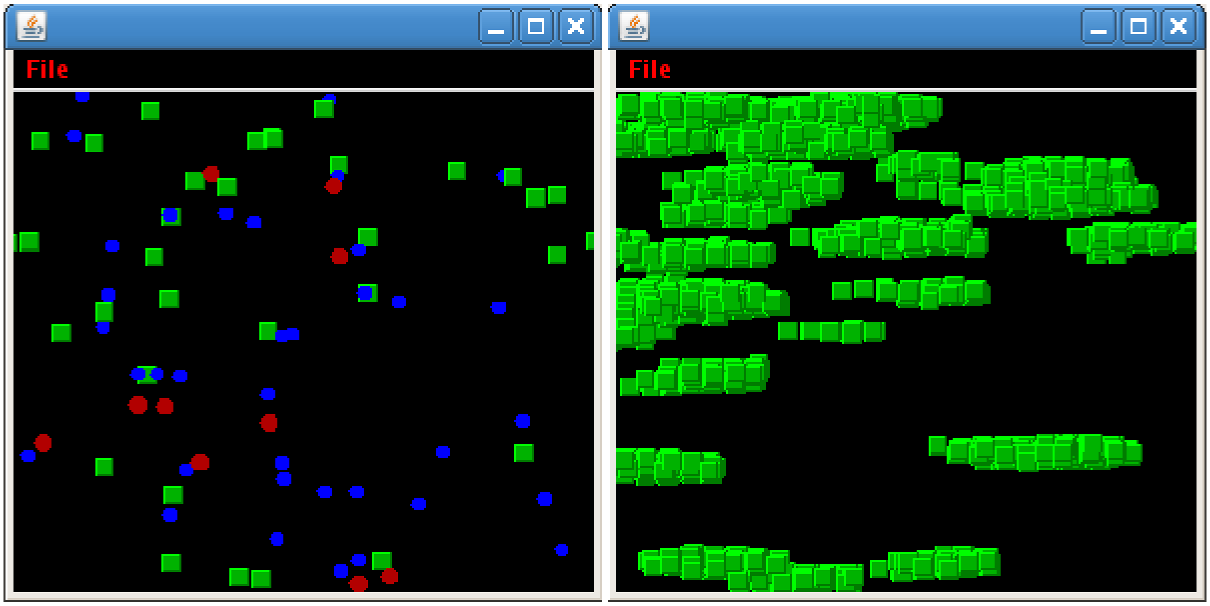
Currently, I have not been able to produce graphs and charts to display the information from the simulation because I am not far enough in my program to do so. I just finished getting a semi-working model up and running and when the model is stable, I should be able to make graphs and charts that are updated by the program itself. This is most efficient because the program has to the information in the simulation.

## 3.5 Development Procedures

Steps to Simulating Evolution

1. Create a changing environment with which a species may interact
2. Create a food source for the species
3. Create a species with designated traits to be tracked
4. Possibly create an herbivousous species and a predator
5. Define how the species may evolve (genetic algorithms)
6. Track the changes in traits and make obsevations
7. Adjust the model until a balance is achieved

# 4 Results, Discussion, Conclusion, Recommendations




My program is not stable enough to show any clear results. However, it does show that when Predators are not present, the defense of an Organism usually drifts to lower and lower values in order to boost speed because there is no need for a good defense if they are not being hunted. When you add Predators to the World, the balance is thrown off and either the Predators die off because they cannot catch the Organism or they eat all the Organisms and then have no food left. Either way, they die of starvation. Also, if there are too many Organisms to start off, they eat all the food and end up in the same predicament as the Predators. Overall, balancing the system is extremely difficult.

# 5 Literature Cited

## References

[1] Chris Colby, "Introduction to Evolutionary Biology", http://www.talkorigins.org/faqs/faq-intro-to-biology.html

[2] Charles Darwin, “The Origin of Species”,
http://www.talkorigins.org/faqs/origin.html

[3] John Wilkins, “Evolution and Chance”,
http://www/talkorgins.org/faqs/chance/chance.html

[4] Laurence Moran, “Random Genetic Drift”,
http://www.talkorgins.org/faqs/genetic-drift.html

[5] Jaroslaw Puszcynski, “Artificial life Portal”,
http://www.alife.pl/portal/main/e/index.html

[6] Jeff Smith, “Genetic Algorithms: Simulating Evolution on the Computer”,
http://www.developer.com/tech/article.php/964131