Using Genetic Programming to Find Expressive Swarm Behaviors
For a Swarm-Based Performance System
Ruben Martinez Jr., 2015.

Computer scientists have been utilizing swarms for a wide variety of purposes for decades. The project at hand looks at them in an untraditional light, by mapping musical rhythms or patterns to interesting swarm behaviors. My research, specifically, focused on using the process of genetic programming to create these visually stimulating behaviors and storing them for use later in the project. I found that I could automatically generate and store behaviors, manually rate their usefulness for our purposes, and breed them to produce new swarm behaviors, though whether this process over time produces increasingly interesting behaviors remains to be seen.

Throughout the first parts of the project, I used a combination of Java and Processing to encode and manipulate the swarm, the swarm behaviors, and the swarm visualization. We had already identified the different constituent variables that define a behavior; for example, flock size, speed, neighbors within a radius, and radius size can affect separation, alignment and cohesion within the flock. Utilizing these factors, I was able to move from having static, manually created behaviors to a more general template for creating dynamic behaviors, or one in which the values of constituent variables change over time. Using a tree-like structure to model behaviors, I successfully wrote a Behavior class to automate the creation of behaviors. I integrated these behavior objects with the existing code at the swarm level, so that every time a swarm was generated, a new behavior would be created as well. I now had Behavior objects that I could modify and potentially breed using a genetic program.

Next, I moved onto laying the web foundations for the genetic program; this involved transitioning some of the existing code from Java into a web-friendly format. I chose PHP for this simply because it was the most versatile web-language for the purposes of this project. The end goal here is to speed up the process of swarm behavior evaluation, by crowdsourcing fitness data so as to avoid the slowness of manually rating how interesting each behavior was. Essentially, this step involved creating a “dummy” Behavior class in PHP: that is, a simplified class that would be able to randomly generate the identifiers that made up the Behavior class I had previously defined in Java; though meaningless numbers on their own, they could be easily bred and stored on the web in an SQL database and interpreted by the “smart” Java Behavior class at a later point.

Once that was finished, I transitioned to writing the genetic program itself to evaluate and breed these “dummy” behaviors. A population of “dummy” Behaviors would be created on the web; then, the “smart” Java program would parse each of these Behaviors one at a time and display a swarm exhibiting each behavior to an end-user graphically until the user had evaluated said Behavior. The user would submit an evaluation back on the web, and the fitness data would be stored until the entire population of Behaviors had been evaluated. Next, it would continue the breeding process by selecting from the population which individuals’ constituent elements, analogously DNA, would be chosen to continue to the next generation based off of which are evaluated the most highly. Once selected, crossover would recombine this Behavior DNA by choosing two parent-Behaviors, picking a single-point in their genetic code, and swapping the first portion of their DNA. The newly created Behavior DNA would move on to mutation, which, with a certain probability, changes one or several constituent elements at random. After repeating this process for the entire population, the resulting individuals would form the next generation of Behaviors, and the entire process would repeat itself, starting back at the evaluation. At this point, I created a front-end interface whereby users could create an account on a website, create their own initial population of Behaviors, and breed them.

Faculty Mentor: Stephen Majercik
Funded by the Gibbons Summer Research Program
Using Genetic Programming to Find Expressive Swarm Behaviors
For a Swarm-Based Performance System
Ruben Martinez Jr., 2015.

At the time of writing, I cannot yet say with certainty that this genetic program produces increasingly well-rated behaviors. However, I was able to create randomly-generated dynamic behaviors, evaluate and store them in a web environment, modify the Java code to construct Behavior objects based on data queried from the web, and finally, successfully breed these Behavior objects across multiple generations.

Faculty Mentor: Stephen Majercik
Funded by the Gibbons Summer Research Program