**Aegis**

**What is Aegis?**

Aegis is an experimentation framework that makes it easy to conduct neuroevolution experiments using the BrainCraft NEAT library.

**How do I get started?**

The easiest way to learn how to use Aegis is by taking a look at FoodExperiment. The experiment works like this: critters and food pellets are placed on the stage. Each step, a critter is told whether there is a food pellet in the three squares in front of it. The outputs of its brain in response to these inputs are translated to movements and turns on the stage. The critter’s fitness is the number of pellets it eats in 100 steps. You should try modifying the experiment or the critter and observing the effects. Are critters more fit if you add more inputs to the critter’s brain? What if you allow the critter to move in any direction? Maybe add multiple critters of the same brain to the experiment?

**Organization**

The code is split into three packages: core, actors, and experiment. Core contains all of the files that are needed to run experiments. Actors is where you can place game agents like food and critters. Experiment contains objects that define and run a specific experiment.

**Creating new experiments**

An experiment in Aegis is a subclass of Experiment that is responsible for setting up the stage and running the simulation. It should create a stage object, get one or more brains from BrainCraft, create actor objects (possibly with those brains), add the actors to the stage, have the actors act for some time steps or until a condition is true, gather the fitness of the brains of those brains, and report them back to BrainCraft. You’ll need to create a BrainCraft population instance and specify the characteristics of the brains that you want in the constructor. You’ll ask this population to give you brains for your actors. You might want to pause the experiment for a few milliseconds before you call act on the actors, so that you can see the actors move on the stage.

**Creating new actors**

Objects that you want to place on the stage need to subclass Actor. Actor’s act() method should be called whenever you need it to pass inputs to its brain, gather the outputs, and act upon them. GetBrainInputs() returns an array of doubles between 0 and 1. It is up to you to decide how many inputs the brain should receive and what those represent. GetBrainOutputs() gives you the outputs of the brain. You need to decide how to use the outputs to change the simulation state. You need to specify how many inputs and outputs brains should have when creating a population of brains in the experiment class.