Generated Darwinism

Team: Kindon Smith, Nicholas Forester, Reece Sagun

Theme:

Our project aims to explore AI as a designer, by crafting suitable creatures for a given environment. The main idea behind our AI is that of a tool, used by game designers to populate there preconstructed worlds with creatures that could survive there. If we were to take our project many steps further, it would end up as a Unity tool for prospective game designers, ready to take in any environment and evolve creatures thusly.

Overview:

Our project idea started with a genetic tournament of sorts, where we would have simulated a range of environments and creatures, simulating evolution through creature interactions and creature evolution. It is an interesting idea, but not necessarily unique or innovative, so we came up with the idea of a tool for game designers to populate worlds. This could potentially save game designers a lot of time and money, because they wouldn't have to hand craft every creature they put in their world, and the creatures would belong in a evolutionary standpoint, according to our program.

We ended up going with a variably input, static environment that has a list of traits, and a randomly generated creature with its own list of traits, some of which corresponding to specific elements of the environment. We then simulate a population of this new 'species' with similar values, each individual having its own fitness, and run a genetic algorithm on the resulting population. This process repeats for 500 generations or so, adjustable depending on what the user wants.

Most of the work for our project came from finding a solid path to go forward with, and then once we had that, figuring out what makes a 'fit' creature in our world. We spent several days trying to find and work with open source 3D morphing software, but we couldn't find anything suitable, so we decided to stick with text based creatures, and have some friends draw the creatures for our program. Our next thought was how to start a 'species'? Any idea that made sense to us would have detracted from the rest of the program, because we wouldn't have time to implement it all, so we started with a random base creature, and a population with similar traits. It's challenging to think what makes these creatures fit; why are 3 toes better than 5, what makes a slower creature better than a faster one, etc. We are happy with how it turned out, but it took a lot of time and thought to work through.

Value:

With AI as a designer in species creation, we look for this AI to help alleviate and shorten the process of hard creating unique creatures for given environments. For example, take MapleStory, a MMO there will be multiple maps and many play spaces. With many play spaces, it would take a significant amount of time to storyboard and create many creatures. This AI would shorten the process by taking in abiotic factors of a play space and generating a species with unique features and stats for that environment.

Although currently out of the scope in the time frame given, another possibility with this AI would be to observe and simulate possible species interacting with other species and evolving in given environments. Instead of just using AI just to design a creature fit for an environment, the AI would take on the role as a simulation of nature. The interface that the user would interact with would be a dial that determines how many generations or how much time has elapsed

during the simulation along with a chart that shows the traits and characteristics of varied species in the given environment. The value of this AI could be used in terms of game design mechanics in which a game progresses over a certain amount of time. Examples of such games would be the

Novelty:

As mentioned before, the novelty behind this isn't that it would be its own game, like No Man's Sky, or used to create your own creatures, like Spore, but as a tool used to generate creatures for an environment. If you look at World of Warcraft, there is a massive world filled with unique environments and hand crafted creatures. Every one of those creatures has been meticulously created, and often times is just a recolored model of a different entity. Imagine if instead of spending a good chunk of time and effort on the creature creation, Blizzard could focus on gameplay and story, leaving the creature creation up to a program like ours. Our idea is a long way off from that point, but this is the vision we've been pursuing during these past weeks.

Technology/Breakdown:

The project can be broken down in to two main parts. These parts are the fitness function and the genetic algorithm.

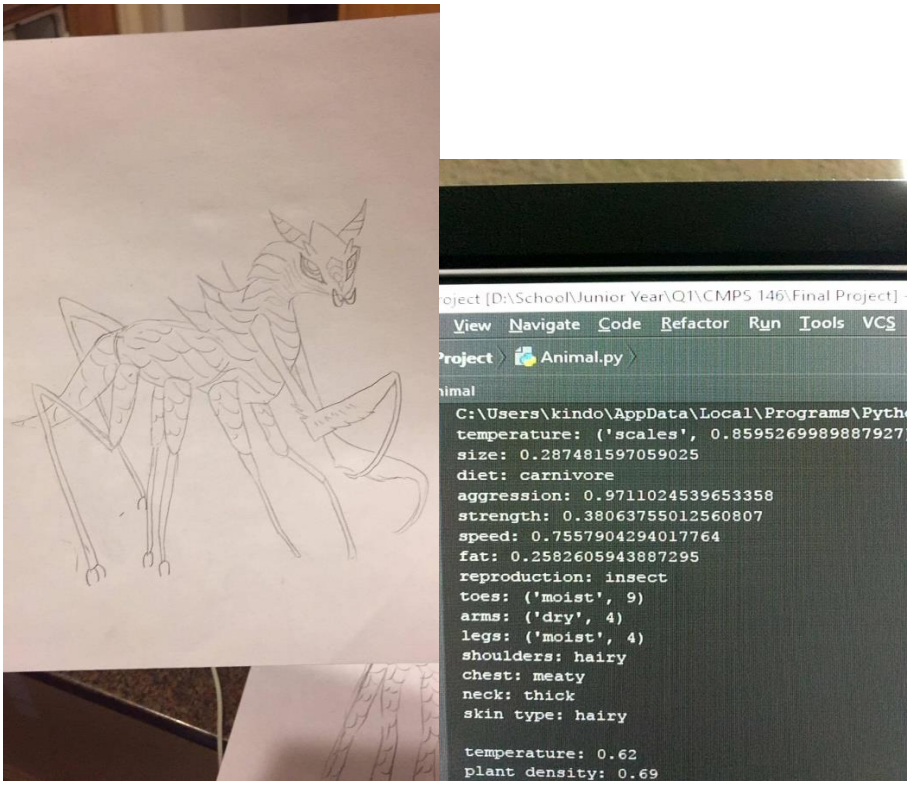
For the fitness function we took in to account different abiotic factors about a given environment as well as different traits on a species. In this project, we randomly generated an environment with given factors to perform a simulation on. In a finished and polished version of this project, the environment and its factors should be supplied by the user to describe their ideal environment space. We used the environmental statistics to provide what we call importance

coefficients. The coefficients would determine to how much extent an abiotic factor is weighed. An example of a factor and its coefficient is natural disaster. The more extreme the value for natural disaster, the higher the coefficient is weighted in the total fitness of that category.

As for the traits on animals, there were a variety of characteristics to run different test on. With the different traits, there was a huge variance on how fitness was calculated. We took a fair amount of time determining the ideal situation for each trait and how it is impacted the environment statistics. For example, our temperature fitness took in the environment temperature and the species' temperature attribute. If the temperature attribute was fur, then an ideal environment temperature would be a low moderate temperature. Each part of the fitness function has its own unique thought process on how the numbers were calculated giving a uniqueness to each variant of traits on a speices.

The code for the genetic algorithm itself was mostly copied over from the Mario level generator, but there were a couple of important changes that we had to make. We used an elitist selection mixed with a roulette selection to form the next generation. Our crossover just involved choosing alternating traits between the two parents for the children. By default, there is a .5% chance that a child will be born with a mutation, and the available mutations include changing any single trait to another variation of that trait. For the numeric traits like speed and strength, we capped how much those can be altered by a mutation, because it doesn't make any sense for a slow population to suddenly give birth to a sprinter.

Mementos:



last - Notepad

File Edit Format View Help

```
aggression: 0.6238256083787217
chest: meaty
fat: 0
neck: flexible
shoulders: 4
skin type: moist
strength: 0.7032084863983269
toes: 8
arms: 4
diet: carnivore
legs: 4
reproduction: insect
size: 0.6718778814610585
speed: 0.525448342384082
temperature: ['hide', 0.37906957368493766]

temperature: 0.83
plant density: 0.47
prey density: 0.45
humidity: 0.14
elevation: 0.15
geovariance: 0.08
fertility: 0.74
sun exposure: 0.72
precipitation: 0.68
wind: 0.66
barometric pressure: 0.26
natural disaster: 0.54
pollution: 0.43
hostility: 0.08
```