Space Bunnies Design Document

Winter is coming! The year is 2320 and the oligarchy of Earth must devise a way to lure sufficient biomass to the planet to feed its starving population. Luckily, a nearby civilization – much resembling what we know as "bunnies" – is extremely susceptible to "divine orders" to board spacecraft. Can you successfully control this voracious civilization to save humanity?

With this project, I intend to create a simple game engine incorporating aspects of network theory and inventory management, allowing the user to either edit parameters, or write more complicated rule-based python code to maximize the biomass that makes it to Earth in the simulation. I foresee the project being extensible, allowing the future creation of a graphical interface and visualization, and the potential for an evolutionary algorithm to be implemented on the user editable parameters and rules, using the biomass delivered as the fitness function. Furthermore, the structure of the code could potentially be repurposed to offer research opportunities in the fields of network theory or inventory management.

Classes

Planet – The planet class will have many attributes, including a bunny capacity, number of docking ports, list of resident bunnies, and connected spaceship routes. Most importantly, the planet class will serve as the node in our network. Resident bunnies will be able to reproduce while on a planet and will also be able to board spaceships that travel on the connected spaceship routes.

Route – The route class will have a few attributes, such as connected planets and route distance, but most importantly, will serve as the edge in our network of planets. Spaceships will travel along routes to transport bunnies from one planet to another, ultimately (hopefully) towards Earth.

Spaceship – The spaceship class will travel between planets along routes while carrying bunnies. Each spaceship will have many attributes, including a location, mass capacity, speed, fill/depart rate, and survival rate. Spaceships will dock at planets after their trips to unload bunnies, assuming a planet has an available docking port. Spaceships will contain the majority of interfaceable methods, including take off, travel, dock, request load.

Bunny – The bunny class will define an individual bunny and will include attributes such as an identifier, a mass, an age, and a propensity to pass on its genes (its relative mass) to offspring. The class will also include methods such as reproduce, board, debark, and die (of old age).

Model

The interaction between the various objects will be governed by a higher-level class or set of classes. This class or set of classes will iterate the necessary class methods over the course of the game time. Some actions will probably be more easily implemented at this level, in which case I will incorporate these elements outside of the individual class methods. For example, the aforementioned reproduce method might be easier to implement here. I plan to start the user interaction with the arguments to a method such as model.run() in this class. From there, I will seek to add more complex user interaction and address the extensibility ideas outlined above.