Project 1 Design Proposal

Document Author(s): Nikhil Shirsekar

Date: 9/22/16

Design Rationale

The objects that are most important to the implementation of the ecosystem project are the animals themselves. All of the animals will be contained underneath the superclass Animal as HighEndAnimal, MidAnimal and LowEndAnimal. This will allow for simplified access to the animals themselves through the use of inheritance, and also the ability to add and remove animal types in the future without intricate changes to the overall design of the system.

Most of the data required to implement this system revolves around how the animals interact with the ecosystem and each other. The constructor for animal involves the number of steps an animal can breed after and how many steps it can move before dying, these two integers are integral components to this system as they create distinctions between the different levels of animals in the ecosystem. The integer rank, which is also part of the constructor for animal also further solidifies the differences between the animals and allows for a simple comparison between 2 animals to determine if one can eat the other. Another important piece of data that is contained in the Animal constructor is the boolean isDisabled. This piece of data is important to implement as it will allow dead animals to become food for other animals.

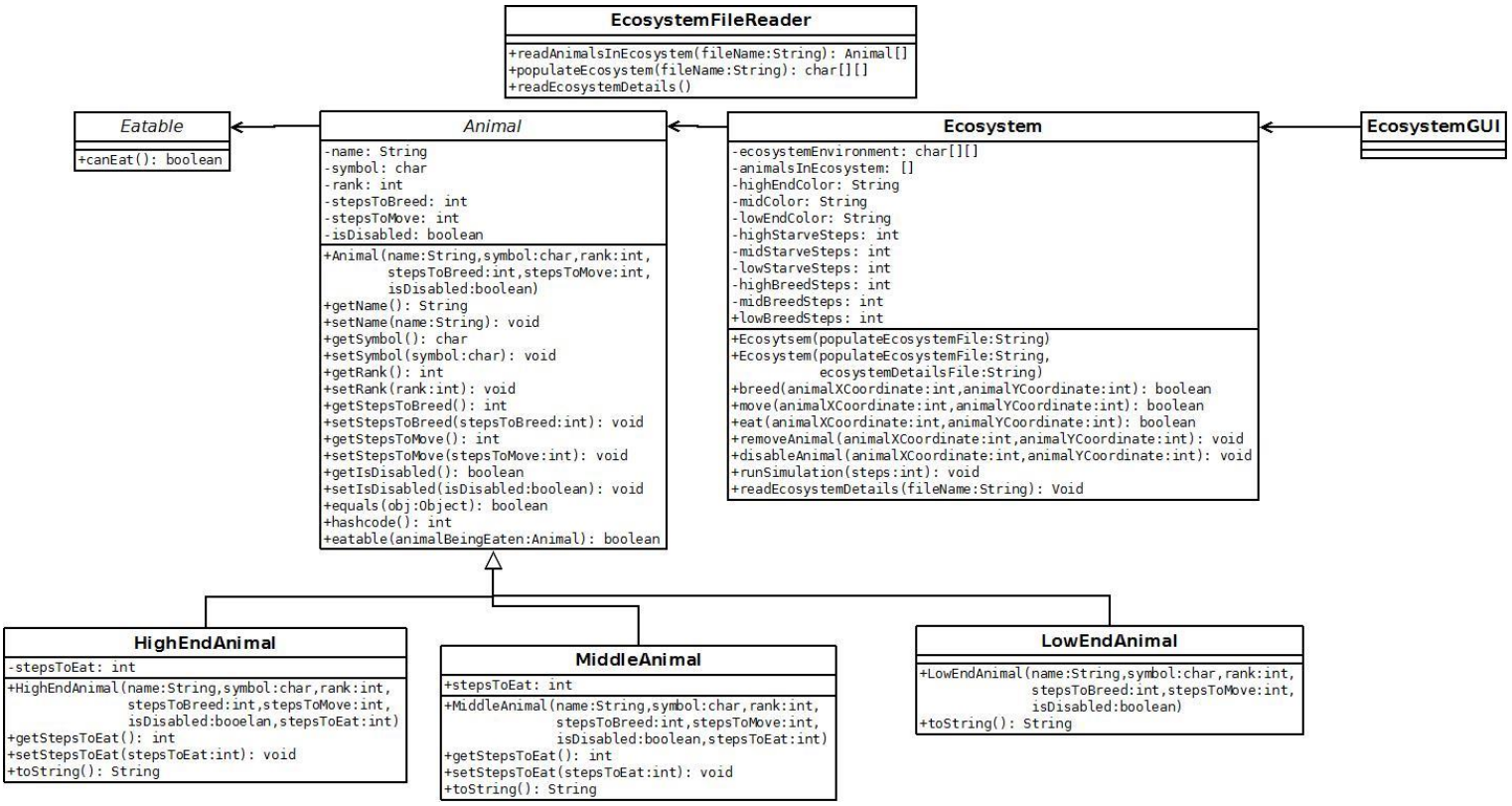
The responsibilities that I have assigned to the superclass Animal are contained there because they are all common between animals regardless of their rank in the food chain. By doing this, I am able to use inheritance to further simplify my design by allowing the engine class ecosystem to only interact with Animal. This will also allow for future growth or reduction of different food chain rankings.

One of the main patterns being used in this system is the builder pattern. This pattern can be seen in the hierarchy of the animals on the food chain. By using the Animal superclass, the system is able return the animal object in various ways by creating animals that are ranked differently on the food chain.

The main limitation that this system has is how it receives all of its information. It is highly dependent on specifically formatted files for its input. While the system does incorporate default values, these may frequently not match up with what the user specifies in their files. This could become problematic if a user is not formatting their files properly.

The GUI in this system will be interacting with the Animal superclass directly, enabling it to access the different ranks of animals in the food chain. This will allow the GUI to show important data that is vital to making an effective simulation of an ecosystem. This data includes which animals are in the ecosystem, and the symbol that represents them. By accessing this information, the GUI will be able to create an accurate grid representation of an ecosystem that the user can understand and interact with. To access this data, the GUI will be accessing a one dimensional array in the Ecosystem class that contains all of the animals in the ecosystem. This will allow the class to get access to all of the getter and setter methods that have been made public in the animal classes.

Figure 1: Class Diagram for Project 1



Document Revision History

Date	Author	Change Description
9/22/16	Nikhil Shirsekar	• Created Design Rationale and Added UML diagram.