CS 474 – Object Orient Languages and Environments

Programming Project 3, Spring 2016

Simulation of a Predator/Prey Ecosystem

Due: Tuesday, April 12, 2016 at 11:59 pm

This project is to be written using the C++ Language. You may work on this project by yourself or with one other person. When working on the project with another person, both students will earn the same grade on the project. Only one submission of the project is to be turned in for the final grading (so make sure it clearly indicates the names of the students involved.)

For this assignment, you are to create a GUI display so show the generation-by-generation population/distribution of creatures in a particular "region". An example of such a display can be found at:

http://www.shodor.org/interactivate/activities/RabbitsAndWolves/

We will be starting out using some of the basic information from this simulation to define the behavior of our creatures. This simulation has 3 different creatures: grass, rabbits and wolves. Your simulation is to start with these three creatures and then add at least 3 more creature types of your own design.

The intent of this project is that you are to use the principles of object oriented design to help make the coding simpler. We expect to see the use of inheritance, polymorphism, abstract classes, pure virtual methods in your code. We hope to see the use of design patterns in your code also.

The GUI will use the Simple and Fast Multimedia Library – SFML, which can be found at:
http://www.sfml-dev.org/index.php
We intend to have some sample code for you to use so this project doesn't become a graphics project but instead stays an object oriented design project.

The creatures are to have operations like:

- Move
- Hunt/Eat
- Evade/Chase
- Reproduce
- Die
- Age/Grow

There will be 3 basic types of creatures:

- Plants
- Herbivores (an animal that eats plants)
- Carnivores (an animal that eats other animals)

For the three creatures types you are required to implement: grass is a plant, rabbit is a herbivore, and wolf is a carnivore.

The following characteristics are to exist for each required creatures:

Grass
- Starts with Food Level of 20
- Maximum Food Level of 100
- Metabolism Rate of -1 (adds 1 to its food level at every stage/day/generation)
- Never Move
- Never Hunt
- Never Evade
- Never Reproduce
- Never Die

Rabbit
- Starts with Food Level of 10
- Maximum Food Level of 45
- Metabolic Rate of 3 (subtract 3 from its food level at every stage/day/generation)
- **Moves** up, down, left or right one space every stage, to the neighboring location that has the highest plant food level that is unoccupied.
- After it has moved (and does not evade a carnivore), it will **eat** up to 5 food levels of the plant at that space, subtracting those 5 food levels from the plant. If the plant have less than 5 food levels, the rabbit can only eat however much food that plant has. The rabbit will not eat food so that it goes over its maximum food level. The rabbit will also eat 1 extra food level from the plant for each 20 food levels the plant has.
- After it has moved, if the rabbit is next to a carnivore and if it has more than 30 food levels, a rabbit will **evades** a Carnivore by moving away from the carnivore by 2 spaces. Note the rabbit does not get to eat or reproduce during a stage that it evades a carnivore.
- After a rabbit has moved, the rabbit has a 50% chance that it will **reproduce** a baby rabbit provided that adult rabbit is at least 10 stages/days/generations old, it has a food level (after eating) of 40 or greater, it is not currently next to a carnivore, and it did not evade a carnivore on that stage/day generation. The baby rabbit will be born in the recently vacated space and will start with a Food Level of 10.
- A rabbit will **die** after it has lived for 25 stages/days/generations, or it reaches a food level of 0 (or gets eaten by a carnivore).

Wolf
- Starts with a Food Level of 150
- Maximum Food Level of 200
- Metabolic Rate of 2
- If the wolf is next to a herbivore and its food level is <= 190, the wolf will hunt and eat herbivore by moving to the herbivore’s location. The wolf will gain 10 food levels for eating a rabbit. The herbivore is killed when eaten.
- Moves up, down, left, right one space every stage to the neighboring location that is unoccupied if it have not hunted and eaten a herbivore.
- Wolves do not evade rabbits, but do they chase rabbits if hungry. If the wolf has less than 40 food levels and there is a herbivore within 3spaces of the wolf, it will move2 spaces closer to the herbivore.

- After a wolf has moved, the rabbit has a 40% chance that it will **reproduce** a baby wolf provided that adult wolf is at least 10 stages/days/generations old, and it has a food level (after eating) of 120 or greater. The baby wolf will be born in the recently vacated space and will start with a Food Level of 150.
- A wolf will die of old age after 50 stages/days/generations or of starvation if its food level drops to zero.

The simulation is to start with 20 herbivores and 5 carnivores randomly distributed around a 40x30 grid area. There should also be a plant at every location in the grid. You are also required to create at least 3 additional creatures of your own design. These creatures could be a plant, a herbivore, or a carnivore, but at least two different types of creatures must be chosen (i.e. you can't create 3 additional herbivores). You create an omnivore if you wish (which eats both plants and animals, so would be both a herbivore and a carnivore).

Programming Style and Restrictions

Your program must be written in good programming style. This includes (but is not limited to) the follow.

- Multiple Source Code Files
- Meaningful Variable Names
- Proper Indentation of Code
- Blank Lines between Code Sections
- Use of Methods and Functions
- Use of Multiple Classes
- In-Line Commenting
- Header Comment for the File
- Header Comments for each Method, Function and Class.

The work you turn in must be 100% your own (or your own group if working with another student). You are not allowed to share code with any other person outside of your "group" (inside this class or not). You may discuss the project with other persons/groups; however, you may not show any code you write to another person/group nor may you look at any other person's/group's written code.
This program is expected to be written using multiple .cpp files and at least one .h file. The use of a makefile is also expected to compile your program. The linked list classes need to be in separate file(s) than the main() function. The linked list classes would ideally each have their own .h and .cpp files; however, it is acceptable to have the entire linked list class hierarchy to be in one .h/.cpp file pair. The file that contains main() should have the remaining code for the user interface.

Project Submission

Directions will be post on the class web page for submission of the project.