

CS 1316 – Homework 11 – Continuous Simulation

Due: Friday, April 23rd, 2009 before 6pm

- 100 points

This is a pair programming problem! You are expected to work with the person you have been paired with in class, and you are both responsible for submitting the exact same code to T-Square. You may collaborate with other students in this class. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. You should not exchange code or write code for others. For pair programming assignments, you and your partner should turn in identical assignments. Collaboration at a reasonable level will not result in substantially similar code.

Simulation

In this assignment, student pairs must construct an object-oriented simulation. This continuous simulation must make use of agent objects. A graphical display must accompany your simulation. In most instances, this can be accomplished using Turtles; however you can choose to use Graphics objects, scene graphs or other means.

Your code should keep the user up-to-date with `System.out.println()` statements at relevant events. You must also provide for file output of appropriate statistics as your simulation runs. This should involve some means which allow the user to define the file name of the resulting document.

Your team must select what to model with your simulation. Depending on what you choose to simulate, you will be asked to implement at least one additional “improvement” to the basic simulation. See below for some basic simulation types. If you want, you may design your own system to model. If you choose to do this, email your grading TA with a specification describing what you intend to simulate by midnight on the Wednesday before the assignment is due.

- Predator / Prey
 - At least two distinct agent classes
 - Predator should have negative impact on prey
 - Predators should seek prey and prey should avoid predators
 - Possible Improvements (Pick one)
 - Diff-Eq. Equilibrium
 - Create a “stable” predator/prey environment where predators die if they don’t eat enough prey and each animal reproduces based on the number alive.
 - Object Variability
 - Have your agents evolve over the course of the simulation. E.G. deer that haven’t been eaten get faster and more perceptive, wolves that haven’t eaten get weaker.
- Disease Model
 - Agents have two distinct states that affect their interactions with other agents
 - Proximity or other factors may transmit this state between agents
 - Your agents must take action against other agents with the infection and some infected agents must take actions to avoid ‘clean’ agents.
 - Possible Improvements (Pick one)

- Invisible Infection
 - Have some agents who are infected but have no visible indication to other agents.
- Progressive Infection
 - Have the disease progress depending on how long a host has been infected.
Consider making a separate object to handle disease inside of the agents. Allow for agents to die from their infection.

Grading Breakdown:

- Simulation
 - Agents handled in an appropriate data structure ...20
 - Continuous simulation handled correctly ...20
- Agents
 - Agents subclass Turtle (if necessary) ...15
 - Appropriate behavior handled in act() ...15
- Output
 - System.out.println()'s contain relevant information ...10
 - Utilizes at minimum FileWriter and BufferedWriter ...20
- Bonus EC possibilities:
 - Disease Simulation [EC Bonus] ...+10
 - Custom Simulation [EC Bonus] ...+Lots