Lab Assignment-4

IIIT-Delhi. 23rd August 2017. Due by 23:59pm on 25th August 2017

Instructor: Vivek Kumar

No extensions will be provided. Any submission after the deadline will not be evaluated. If you see an ambiguity or inconsistency in a question, please seek a clarification from the teaching staff.

NOTE: Attendance in the lab is mandatory.

Plagiarism: All submitted homeworks are expected to be the result of your individual effort. You should never misrepresent someone else's work as your own. In case any plagiarism case is detected, it will be dealt as per IIITD policy for plagiarism.

NOTE: You will have to create a "PRIVATE" git repository to manage your code for this lab assignment. This is a part of evaluation criteria and will be checked by TA during the demo. The timestamp on the git log should be before the deadline.

Problem Description:

In this lab you have to simulate herbivores and Carnivores game.The point of the game is to illustrate the food chain in nature, but it's not all fun and games. Biologists actually run serious simulations of this type on computers in order to study real animal populations.

Your program will run a simulation of herbivores and carnivores.The basic idea of simulation software is to loop through a bunch of objects (Animals, keep animal class as **abstract**) that can each take a turn. Notice that herbivores and carnivores will have a different way to take a turn. The class called World runs the simulation. Each animal is accompanied by a time instance at which it will take a turn (time-instance is an integer). Since the world manages a clock, it can tell the object to take its turn when its time comes. The simulation is *event-driven* (don't confuse with event-driven programming that we are yet to cover in lectures). Assume there will usually be an event scheduled for every time unit. Also note that several animals can take a turn at the same time.

The implementation of event-driven simulations is quite simple. The world stores the objects in a *priority queue*, which is like an ordinary queue except that its elements are ordered from high priority to low priority (you can google to get detailed information on priority queues). The highest priority object is the one with the soonest event time (i.e., the lowest time-stamp), and it is dequeued first. If two animals have same timestamp then animal with more health measure is given priority. If health measure is also same then the herbivores is given priority as compared to carnivores. Among two herbivores the one whose x,y coordinates are closer to 0,0 are given priority, similarly the same applies for carnivores. Assume that no two animals will have same priority.

The total final time is provided as input initially. This total final time is also equal to total number of turns allowed in the simulation.

The simulation algorithm goes like this: loop until the queue is empty or until maximum turns allowed is reached. In each iteration, dequeue an object from the priority queue and tell the object to take a turn, which might involve enqueuing the object for another turn. For enqueuing the object to another turn the new time-stamp for the animal assigned is by generating a random number between [max time-stamp uptill now, total final time] (exclusive).

The animal will be removed from the list and not go in the next turn:

1. If animal dies.
2. If new time-stamp of animal becomes total final time - 1 and max time-stamp will not be updated.

Assume each animal lives in a forest. All the simulations are performed in the forest. Each animal is accompanied by integer (x,y) coordinates denoting the current position of the animal in the forest. Also each animal has a health-measure. There are also grasslands in the forest for herbivores to eat. Assume every grassland is circular in shape. It is accompanied by x-y coordinate denoting the centre of the circle and also the radius. A grassland also has grass-availability as one of the measure denoting the amount of grass available.

Herbivores-

Each herbivore has maximum grass-capacity which basically tells how much quantity of grass a herbivore can eat at each turn. A herbivore will always try to avoid a carnivore and eat grass.
Herbivore takes the turn in the following manner-

1. A herbivore can choose among three actions. It can either
    a. go to the nearest grassland
    b. go away from nearest carnivore
    c. Choose to stay at current position and eat (if inside grassland).
2. If there is no carnivore left then the herbivore will either go to nearest grassland or stay at current position with 50% chance. Below points assume that there is at least one carnivore left.
3. If a herbivore is not inside the grassland -
    a. Then there is 5% chance that the herbivore will stay.
    b. If herbivore plans not to stay then there is 65% chance that herbivore will move 5 units in the direction of line joining the herbivore’s current position to the centre of nearest grassland. There is 35% (100-65) chance that herbivore will move 4 units

away the direction of line joining the herbivore’s current position to the position of nearest carnivore.

4. If herbivore is inside the grassland then-
    a. If the amount of grass-available is greater than or equal to grass-capacity of herbivore-
        i. Then there is 90% chance that herbivore will choose to stay and eat the grass up to its maximum capacity.
        ii. If the herbivore does not choose to stay then there is 50% chance that herbivore will move 2 units away the direction of line joining the herbivore’s current position to the position of nearest carnivore. There is also 50% chance that herbivore will move 3 units in the direction of line joining the herbivore’s current position to the centre of next-nearest grassland.
    b. If the amount of grass-available is less than grass-capacity of herbivore-
        i. Then there is 20% chance that herbivore will choose to stay and eat the grass and finish the whole grass of grassland.
        ii. If the herbivore does not choose to stay then there is 70% chance that herbivore will move 4 units away the direction of line joining the herbivore’s current position to the position of nearest carnivore. There is also 30% chance that herbivore will move 2 units in the direction of line joining the herbivore’s current position to the centre of next-nearest grassland.

At each turn the health-measure of the herbivore gets affected in the following fashion-

1. If a herbivore is inside grassland and chooses not to stay then it’s health gets reduced by 25.
2. If a herbivore eats grass upto maximum capacity then its health increases by 50% the current value.
3. If a herbivore eats grass not upto maximum capacity then its health increases by 20% the current value.
4. If herbivore’s health becomes equal to or less than zero then herbivore’s dies.
5. If herbivore has been outside grassland for more than 7 turns (it’s own turn) then its health starts getting decreased by 5 for every further turn until it reaches the grassland.
6. If a herbivore’s health is non zero after a turn, it will go for next turn.

Carnivores-

A carnivore will always try to catch a herbivore. A carnivore is intelligent and knows that a herbivore is likely to be found at a grassland.
Carnivores takes the turn in the following manner-

1. A carnivore can choose among three actions. It can-

   a. Eat a herbivore.
   b. Move towards a herbivore.
   c. Choose to stay.
2. If there is no herbivore left then carnivore will always stay at its position (will do nothing). Rest points assume there is at least one herbivore left.
3. If a carnivore is within a 1 unit radius of a herbivore then it will kill and eat the herbivore. Below points are the cases when it is not inside the 1 unit radius of the herbivore.
4. If a carnivore is not inside grassland -
   a. There is 92 % chance that a carnivore will move 4 units in the direction of the line connecting the current position of it to the nearest herbivore. With 8% chance it will stay at its current position
5. If a carnivore is inside grassland -
   a. There is 25% chance that carnivore will stay inside the grassland.
   b. If it does not stay inside the grassland then it will move 2 units in the direction of the line connecting the current position of it to the nearest herbivore.

At each turn the health-measure of the carnivore gets affected in the following fashion-

1. If the carnivore is not inside grassland and it chooses to stay then its health will get reduced by 60.
2. If it is inside the grassland and it chooses to stay then its health gets reduced by 30.
3. If it has not come near a five mile radius of herbivore for more than seven of its turns then its health start getting reduced by 6 until it comes inside a five mile radius of the herbivore.
4. If it eats the herbivore then the ⅔ rd of the health of herbivore gets added to the health of carnivore.
5. Rest in all the cases there is no effect on health.
6. If health of carnivore becomes equal to or less than zero then the carnivore dies else it goes for next turn.

Since the simulations can become increasingly complex if large number of herbivores, carnivores and grasslands are taken, **therefore you are allowed to make following assumptions:**

1) **There are only two herbivores, two carnivores and two grasslands for the simulations to be performed**.
2) **Health and Grass Capacity is same for all herbivores and similarly health is same for all carnivores.**
3) **A grassland remains a grassland even if there is no grass but health of herbivore does not get increased if there is no grass (rest all the actions will remain the same).**
4) **Also assume that no two grasslands will overlap.**

A sample menu-driven input and output-

Blue lines are print statements. Anything within the brackets in green color is for explanation. Inputs by the user are represented by black color.

Menu-

Enter *Total Final Time for Simulation:*

25 (This is just a time unit and does not mean 25 seconds or 25 minutes, etc.)

Enter *x, y centre, radius and Grass Available for First Grassland:*

0 5 4 15

Enter *x, y centre, radius and Grass Available for Second Grassland:*

0 0 1 25

Enter Health and Grass Capacity for Herbivores:

15 20

Enter x, y position and timestamp for First Herbivore:

2 2 5

Enter x, y position and timestamp for Second Herbivore:

0 -5 15

Enter Health for Carnivores:

25

Enter x, y position and timestamp for First Carnivore:

1 5 12

Enter x, y position and timestamp for Second Carnivore:

2 7 10

The Simulation Begins -

(Status of priority queue at start will be: First Herbivore > Second Carnivore > First Carnivore > Second Herbivore. So, First Herbivore is dequeued. Since it is inside First Grassland, and the amount of grass-available is less than grass-capacity of herbivore then with 20% chance herbivore chose to stay and eat the grass and finish the whole grass of First grassland)

It is First Herbivore.

It's health after taking turn is 22.5

(Timestamp for First Herbivore will randomly selected between (15, 25), say it is 18. Status of priority queue will be: Second Carnivore > First Carnivore > Second Herbivore > First Herbivore. So, Second Carnivore is dequeued. Since it is not in any Grassland, with 92 % chance the carnivore will move 4 units in the direction of the line connecting the current position of it to the nearest herbivore which is First Herbivore.)

It is Second Carnivore.

It's health after taking turn is 25

(Timestamp for Second Carnivore will randomly selected between (18, 25), say it is 19. Status of priority queue will be: First Carnivore > Second Herbivore > First Herbivore > Second Carnivore. So, First Carnivore is dequeued. Since it is in Grassland, with 25% chance, carnivore stays inside the grassland.)

It is First Carnivore.

It is dead.

(First Carnivore dies as health is < 0 after its turn. Status of priority queue will be: Second Herbivore > First Herbivore > Second Carnivore. Now Second Herbivore is dequeued. It is not inside the grassland and with 65% chance it moves towards the Second Grassland.)

It is Second Herbivore.

It's health after taking turn is 15.

(Timestamp for Second Herbivore will randomly selected between (19, 25), say it is 20. Status of priority queue will be: First Herbivore > Second Carnivore > Second Herbivore. So, First Herbivore is dequeued.)

It is First Herbivore.

It's health after taking turn is 22.5

(Timestamp for First Herbivore will randomly selected between (20, 25), say it is 21. Status of priority queue will be: Second Carnivore > Second Herbivore > First Herbivore. So, Second Carnivore is dequeued. It kills First Herbivore which is in the radius of 1 unit.)

It is Second Carnivore.

It's health after taking turn is 40.

(Timestamp for Second Carnivore will randomly selected between (21, 25), say it is 22. Status of priority queue will be: Second Herbivore > Second Carnivore. So, Second Herbivore is dequeued. It chose not to stay and with 50% chance it move 2 units away the direction of line joining the herbivore's current position to the position of second carnivore.)

It is Second Herbivore.

It is dead.

(Second Herbivore dies as health is < 0 after its turn. Status of priority queue will be: Second Carnivore. It is dequeued and it stays in the grassland)

It is Second Carnivore.

It's health after taking turn is 10.

(Status of priority queue will be: Second Carnivore. It is dequeued and it again stays in the grassland)

It is Second Carnivore.

It is dead.