

Programming Project: Predator-Prey Simulation

Aygün BAYIR

161044119

Introduction

This is a 2D predator-prey simulation. In this simulation, predators are doodlebugs and preys are ants. Also there is a world that is simulated by grid of cells, organisms are living in this world. Grid size is adjustable by changing global variables.

Process of Simulation

There are two types of organisms that are doodlebugs and ants. They behave according to their roles. Predators which are doodlebugs hunting other organisms. Preys which are ants not hunting but being hunted. Predators never eat other predators, they always eat ants. They move and if they could find any food they will eat.

Ants are prey. They never starves so they do not need to eat other organisms. They move around and sometimes they breed to keep their population. After breeding, sometimes their babies mutate to poisonous ants.

Poisonous ant are same as ants. The only difference is being poisonous. If a doodlebug eats poisonous ant, doodlebug will die after two time steps.

Implementation

There are four classes. Their implementations are seperated in different files.

- World
- Organism
- Ant (derived class from Organism)
- Doodlebug (derived class from Organism)

World class keeps information of world and methods. Its content is below:

- Standart constructor
- Custom constructor takes three parameters (organism pointer, x, y)
- Virtual destructor
- setCell method that puts an organism into world
- getCell method that returns an organism in the world
- printWorld method that prints content of earth
- tick method that simulates time for one time-step
- private 2D array of Organism pointers which simulates the world

World class is different from others. Other classes are derived from Organism. These classes for organisms. They present organisms and their methods simulate life functions of organisms.

Organism class is base class for organisms that are Doodlebugs and Ants. Its content is below:

- private integer numbers for x, y coordinates
- private breed_time for keeping breeding time information
- private boolean is_moved as move information
- private worldPtr: world pointer that points to World class that gives access to world
- Organism standard constructor
- Organism three parameter constructor takes world pointer, x and y
- Virtual destructor
- Pure virtual functions that will be defined in organism classes (breed, move, starve)
- World pointer getter function

Ant class is simulates organism Ant which is prey in this simulation. Its content is below:

- Private boolean is_poison that keeps status of being posionous
- Default constructor
- Three parameter constructor that takes world pointer, x and y
- getType getter function
- isPoison getter function that returns status of poison
- setPoison setter function for is_poison value
- breed function for breeding
- move function for moving
- starve function that required because of virtual definition, always return false because ant never starves.

Doodlebug class is simulates organism Doodlebug which is predator in this simulation. Its content is below:

- private integer starve_times that keeps status of starve
- private boolean poison_eaten that keeps status of being poisoned
- private integer moves_after_poison that counts moves before dying because of poison
- Default constructor
- Three-parameter constructor that takes world pointer, x and y
- Breed function for breeding
- Move function for moving
- Starve function that returns status of starving which is boolean

Conclusion

This simulation program simulates a world and life functions of organisms which are in this world. There is a cyclical pattern between populations. Ant population increasing by breeding and decreasing by eaten by doodlebugs. Doodlebug population is increasing by breeding and decreasing by dying because of starving or eating poisonous ants. But there is a certaing thing, their population never become zero. Their populations always be.