

Create a project called Ants and add source files as you see fit.

The goal for this programming project is to create a simple 2D predator-prey simulation. In this simulation the prey are ants and the predators are doodlebugs. These critters live in a 20 x 20 grid of cells. Only one critter may occupy a cell at a time. The grid is enclosed, so a critter is not allowed to move off the edges of the world. Time is simulated in steps. Each critter performs some action every time step.

The ants behave according to the following model:

Move: For every time step, the ants randomly try to move up, down, left, or right. If the neighboring cell in the selected direction is occupied or would move the ant off the grid, then the ant stays in the current cell. You are free to code this such that ants will always try to move to a free square if one exists or they can choose a direction at random and give up if it is occupied. You are not permitted to always choose the same direction though, it must be randomized.

Breed: If an ant survives for three time steps, at the end of the time step (i. e. after moving) the ant will breed. This is simulated by creating a new ant in an adjacent (up, down left, or right) cell that is empty. If there is no empty cell available, no breeding occurs. Once an offspring is produced an ant cannot produce an offspring again until it has survived three more time steps. If breeding fails because there is no room then the ant will also reset its timer.

The doodlebugs behave according to the following model:

Move: For every time step, the doodlebug will move to an adjacent cell containing an ant and eat the ant. If there are no ants in adjoining cells, the doodlebug moves according to the same rules as the ant. Note that a doodlebug cannot eat other doodlebugs.

Breed: If a doodlebug survives for eight time steps, at the end of the time step it will spawn off a new doodlebug in the same manner as the ant.

Starve: If a doodlebug has not eaten an ant within three time steps, at the end of the third time step it will starve and die. The doodlebug should then be removed from the grid of cells.

If a doodlebug starves in the same iteration that he is supposed to breed then he dies and does not breed.

During one turn, all the doodlebugs should move before the ants.

Write a program to implement this simulation and draw the world using ASCII characters of "O" for an ant and "X" for a doodlebug. Create a class named Organism that encapsulates basic data common to ants and doodlebugs and perhaps empty tiles. This class should have a virtual function named move that is defined in the derived classes of Ant and Doodlebug. You may need additional data structures to keep

track of which critters have moved. Ants should be colored green and doodlebugs should be colored red and this assignment requires you to work with windows.h.

Initialize the world with 5 doodlebugs and 100 ants placed randomly. After each time step prompt the user to press Enter to move to the next time step. You should see a cyclical pattern between the population of predators and prey, although random perturbations may lead to the elimination of one or both species.