Predator Prey Simulation (Wandering Beasts)

A variety of critters wander in the grasslands. Some are looking for grass to eat. We will call these the prey. Others are looking for prey to eat. We will call these the predators.

Assignment

In this step of the assignment, you will create an application that can generate 5 predators and 10 prey, and place them in a 20x15 grid. We call this collection the critters.

When a critter is asked to move, it must randomly choose from the 4 cardinal directions (up, down, left, right). The critter will then move in that direction, unless the movement would cause it to leave the grid. In this case, the critter doesn’t move. The critters must be created in random order For example, not all predators and then all prey. This order must be preserved.

This will be a visual simulation. Your application must display all of the critters on a grid. The size of the display is your choice. But remember the grid is 20x15.

The simulation will allow one critter to move at a time. The critters must be allowed their turn in a cyclical order. Each movement must be initiated by the user pressing the $s$ key on the keyboard.

Requirements

- Program must use OpenGL to display the grid and critters.
- Program must use OpenGL to receive the $s$ key presses to advance the simulation.
- Program must have a 20x15 grid.
- Program must have 5 predators.
- Program must have 10 prey.
- Program must create and store the critters in an initially random order.
- Critters must move one at a time.
- Critters must randomly choose direction of movement.
- Critters must not leave the grid.
- Predators and prey must be displayed differently.
- The display must be kept up to date.
- The executable program must be called PP.
- Use inheritance and polymorphism.

Hints

- Look for common features that both predators and prey have. Make a base class.
- Use polymorphism to track your critters. (A vector of base class pointers may be handy.)
- Your program is allowed to have additional features, as long as they do not break any of the required features.

Show Off Your Work

To receive credit for this assignment, you must upload the source code (.h and .cpp files) and the Makefile to the Canvas submission system.

Additionally, the program must build and run. Any incorrect performance or memory errors will be counted against the assignment score.