CS 3005: Programming in C++

Predator/Prey Simulation Part 2

This assignment is the second of two assignments in which you will construct a simulation program to observe the effects of balance in a predator/prey food chain. We are concerned with two types of Critter: Lions (which are predators) and Zebras (which are prey).

To represent the simulated system, we will use a two dimensional grid of squares, with zero or one critter per square. Each critter is located in one square. It may move to a horizontally or vertically adjacent square. No diagonal moves are allowed.

Critters may die from starvation, or from being eaten by another critter.

All critters can reproduce periodically, with the offspring spawning in an adjacent square.

Assignment

Create a class Simulation. The simulation will hold a vector of Critter pointers. Some of the critters will be Lions and others will be Zebras. You are required to create unit tests.

Also create a main program that uses OpenGL to display your simulation.

Programming Requirements

Your Simulation class must store the following data.

- Two integer values, describing the width and height of the simulation grid.
- A vector of Critter *.

Your Simulation class must have the following methods.

- Simulation( int width, int height ); Sets the data members correctly. For each allowed x,y location, randomly choose to create a Zebra (40%), Lion (25%) or nothing (35%). If a critter is created, do so with dynamic memory and add its pointer to the critter vector. Assume the random number system is already seeded.
- virtual ~Simulation( ); The destructor is required. Loop over the vector of pointers and delete all of them.
- const std::vector< Critter* >& getCritters( ) const; Returns the vector of critter pointers.
- void eatAll( ); Calls eat on all critters in the vector.
- void reproduceAll( ); Calls reproduce on all critters in the vector.
- void moveAll( ); Calls move on all critters in the vector.
- void removeDead( ); Removes all critters that are no longer alive from the vector. Must delete the member for each of the removed critters.
- void step( ); Calls eatAll, reproduceAll, moveAll and removeDead.

You must create unit tests for the Simulation class.

Program Programming Requirements

The program must:

- Launch an OpenGL program.
- Create a single simulation object.
- Call the step( ) method every time the user presses the s key.
- Display the background and all critters when display( ) is called.

Show Off Your Work

To receive credit for this assignment, you must upload the source code (.cpp and .h files) and the Makefile to the Canvas submission system. This should include critter.h, critter.cpp, lion.h, lion.cpp, zebra.h, zebra.cpp, simulation.h, simulation.cpp, Makefile, and all of the unit test files from both assignments. The Makefile must build a program named unit-test that can be run to execute all of the unit tests. Also,
all of the files required for the OpenGL program must be included.

You may assume that `gtest`, `OpenGL` and `glut` have been installed on the system used to check your code. Or, you may use your own version of Google Test.