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

**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.

**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 it’s pointer to the critter vector.
- **virtual ~Simulation();** The destructor is required. Loop over the vector of pointers and delete all of them.
- **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.

Your Critter class must implement the following changes.

- **Critter( int x, int y, int level );** Add a mBreedStep data member. Initialized to 0.
- **virtual bool reproduce( std::vector< Critter* >& critters, CritterPtr& ) = 0;** Details to come (not required for assignment Fall 2017)

Your Lion class must implement the following changes.

- **virtual bool reproduce( std::vector< Critter* >& critters, CritterPtr& );** Details to come (not required for assignment Fall 2017)

Your Zebra class must have the following methods.

- **virtual bool reproduce( std::vector< Critter* >& critters, CritterPtr& );** Details to come (not required for assignment Fall 2017)

**Show Off Your Work**

To receive credit for this assignment, you must submit your working code project to Canvas. No unit tests available.