

David Chun and Faisal Mahmood
dc37875 fai265?
Project 4 Group 36

Readme

This project involves creating a world populated by Critters who can move and can interact through encounters. Each encounter involves either an attack or a retreat. Movements for each Critter are walking or running, depending on the situation.

We will use a set to contain all the critters currently alive. It will be a static data component in the Critter class. When critters die or run out of energy, they will be removed from this set.

There will be 4 Critter subclasses called MyCritic1, MyCritic2, MyCritic3, and MyCritic4. In each of these subclasses there will be a constructor for the object as well as the 2 abstract methods specified in Critter.java. These two methods are doTimeStep() and fight(String opponent). doTimeStep() return a void value and takes no parameters. Fight returns a boolean and takes in a string of its opponent. Depending on the critter, they will invoke the walk or run method from the Critter class in both the doTimeStep and the fight methods. In both these situations, the critter will check for life variables. If it runs out of energy or is killed, it will be removed from the set of alive critters in the Critter class. In addition to these two methods defined in the Critter class, each critter may call the reproduce method which takes two parameters, a Critter and a direction to go to.

The custom critter classes must also contain at least a field storing its current direction and its remaining energy. These fields will be private and can be accessed through getters and setters if need be.

For the view of the world, there will be a public static displayWorld() method in Critter.java that will neither take nor return parameters. The grid displayed by this method to System.out will involve a space for each unoccupied location, the String representation of a critter if present, and the border of the world.

The controller for this world exists in Main and will involve a method taking 6 different input options. The quit option terminates the program, halting the game. The show input displays the world by calling the displayWorld method. The step option takes either a string "step" or "step some_value" to invoke either the default 1 world step or the specified amount from the second part labeled some_value. The seed string sets the seed in the critter class to produce duplicatable results. The make input will accept either a string "make" or "make some_value" similar to the step option. This optional some_value will be passed to the makeCritic class in Critter.java as a parameter. The stats option includes the optional addition as well and invokes the getInstances method from Critter.java. This getInstances method returns a list which must be traversed to call runStats for each critter in the list.

After each command is processed, the controller will prompt the user for a new input, quitting when the quit option is invoked.