

Critters are simulated in a virtual grid world, in which they may move, fight, die, and reproduce. The progression of these events occurs over time, represented by time steps. The user specifies the number of time steps to simulate in this model world from the command line with the “step” command. Other commands include “make” if the user chooses to make a Critter, “show” to show the current view, “seed” to seed the random number sequence, “stats” to retrieve information about the number of Critters in the grid, and finally “quit” to terminate the program.

All Critters have a location, and this location (x, y) may be represented as a coordinate object.

I created the Coordinate class as a blueprint for Coordinate objects. The Coordinate class contains x coordinate and y coordinate integer data fields as well as a constructor. It also contains toString(), hashCode(), equals(), and compareTo() methods. toString() was implemented for debugging purposes. hashCode() was overridden so that I could store Critters based on corresponding Coordinate objects based on their location into a HashMap. I implemented hashCode() and equals() so that two critters at the same location would be stored at the same coordinate in the HashMap. compareTo() was not used in my final project, but I added it in case I wanted to use Collections.sort() on all of the Coordinate objects in a list (for debugging).

I used a HashMap and the original population ArrayList to hold my critters. The HashMap had Coordinate objects as keys and an ArrayList of critters (alive or that had died during the current time step) at that location as the value. I also used the original population ArrayList to hold my Critters.