Readme

Danny Vo dpv292
Zain Modi zam374

CritterWorld.java

Class was created to represent the world on which the Critters interact with

public static LinkedList<Critter> getLiveCritters();
Accesses our private LinkedList<Critter> (all Critter objects that are alive(and returns it

public static Critter[][] getWorld();
Accesses our private Critter[][] array (represents the XY display).

public static void addToCrib(Critter c);
Instantiates type of passsed Critter and adds to our ArrayList<Critter> crib

public static void removeDeadCritters();
Removes the dead critters in the liveCritters list

public static void resetWorld();
Clears the Critter[][] array

public static void birthBabies();
Places babies from ArrayList<Critter> crib into LinkedList<Critter>.

public static void addCritter(Critter c);
Contructs critter specified and adds to LiveCritter array.

LinkedList<Critter> to represent all live Critter objects
ArrayList<Critter> to hold all reproduced Critters before placing them

Sub-Critter classes (Mia.Java, Lexi.Java, Kennedy.java, Asa.Java)

The special Critter classes that we created, each with their own behavior sets (as described in the actual code)

private int[] behavioralPreference = new int[6]
Array field that contains either 0 (rest), 1 (walk), 2 (run). When doTimeStep is invoked, a random number between 0 and 5 is rolled, and that is the index for accessing this array which then determines if the Critter will rest, walk, or run

private int[] directionalPreference = new int[16]
Array field that contains the 8 radial directions (0 - 7). When doTimeStep is invoked, a random number between 0 and 15 is rolled, and that is the index for accessing this array which then determines the direction the Critter will move in (if it's moving at all)

private boolean hasMoved
Field that tells if the Critter has moved this timeStep or not

@Override
public String toString()
Returns the first char of the Critter's name

@Override
public void doTimeStep()
Sets hasMoved to false
Determines if the Critter will rest/walk/run
Determines the direction the Critter will move (if at all)
Executes Critter movement if necessary
Reproduces if passes energy requirement

@Override
public boolean fight()
Returns whether a Critter will fight a certain species or not

General Methods

All methods we were to write from the instructions implement the specified behaviors. worldTimeStep() will iterate through the LinkedList of liveCritters and call each Critter's doTimeStep() then resolve any encounters between Critters. displayWorld() places all live Critters onto a 2D array and prints to the console (if Algaes were placed on top of Critters, the Critter will be displayed instead if WorldTimeStep() has not been invoked). reproduce() clones the parent, halves the parents energy and sets that as the child's energy, then place the child in an unoccupied spot adjacent to the parent. makeCritter() adds a specified Critter to the list of liveCritters.

User Inputs

make <String Critter> #amount adds that many Critters of specified type to the world
step #amount calls worldTimeStep() the specified amount of time (default = 1)
show displays the graphical representation of the current world state
stats shows the amount of each Critter in existance