Shamma Kabir – sk38422
Allen Yang – jy8435

README

In our code, we created four new critter classes that extends Critter: weirdCritter, coolCritter, stupidCritter and randomCritter. All these critters Overrode the doTimeStep() and fight() methods. Some also had a "dir" field indicating which direction it would move. They all implemented the abstract methods in unique ways—for example weirdCritter ALWAYS fights itself, and SOMETIMES fights other critters, which we made random.

We held our critters in a variable called *population* that was an ArrayList<Critter>.

In order to implement our walk and run methods, we used a helper moveX and moveY and these helper functions helped implement the way the world wraps (ex: out of bound X means returning to 0).

We also implemented helper methods for fight: flipCoin, isLegalRun, fighting and encounter. flipCoin randomizes who dies (each critter gets a 50% of winning). With isLegalRun, we were able to determine if a critter wanted to run away, the run was valid. The fighting function puts two critters together when we know the two will fight, and the fighting helper function also implements flipCoin. The last helper method is encounter, which takes into consideration special cases (for example algae vs algae) and this also uses the other three helper functions. The helper functions are used until the fighting is resolved.