

Name: Young-suk Lee, Vincent Liu
CS315H
Assignment #3: Critters

Overview

This program's purpose is to simulate critters that behave according to instructions in a certain file. The objective to this assignment is to create an interpreter that reads and performs instructions in a file appropriately.

Program Description

(for Interpreter)

= When loading the file, instead of converting the file to an ArrayList of String Arrays, we stored everything in terms of Integer objects to an ArrayList of Integer Arrays. The index of the ArrayList plus one represents both the value of nextCodeLine and the line number of the file.

= We chose to parse everything in the loadSpecies method instead of the executeCritter method to avoid the slow down of tokenizing and parsing the values of the code line. We even did this for the command itself and assigned static 'int' values to represent each command. This also had the effect of only having to retrieve the value once for the switch statement in the executeCritter method. Before this, we had to retrieve the String array instruction line from the ArrayList in each if statement. In addition, almost all problems with the behavior file itself are now caught in the loadSpecies method instead of at runtime.

= We handled relative jumps using an extra element in the instruction array. A method is used to distinguish between relative jump types and absolute jumps by parsing the raw command and storing the constant int representation of the jump type in the instruction array. Another method is used to determine the actual line number where the interpreter is to jump.

(for CritterTest)

= We also created a CritterTest class, which is used to test the behavior file and Interpreter class. CritterTest implements Critter and has the methods in the Critter interface. It also simulates the environment and the scenario using its static methods.

= The class uses a method newX and newY which returns the int x, y position of the adjacent position based on the bearing of the cell. It is used to convert an absolute heading to its corresponding cell.

= A heading is the direction a critter is facing with respect to the grid, a bearing is the direction of an object relative to the critter's heading, and both are measured in degrees clockwise. We need the bearing adjust method because the direction of critters and grid objects are stored in absolute headings, while almost all the instructions use relative headings.

= The infect method without a parameter simply uses the infect method with a parameter and assigns the parameter to 1.

= The CritterTest class efficiently tested our interpreter by simplifying the environment. Only one critter is created and the rest of the environment is simulated using GridObjects. The GridObjects have a type, a heading, and, within the HashMap, a position. The HashMap was used so that only actual objects would be stored and they could be accessed using their position, without wasting space for empty cells or providing bounds for the environment.

Testing Procedures

= Every command was tested using specially crafted combinations of GridObjects and behavior files. Information about the critter and any GridObjects are printed out after each execution of the executeCritter method. The executeCritter method was mostly run 10 times because less than 10 runs were required for all test cases. The number of runs was increased and decreased as needed to test longer term effects.

= When writing the command methods, we first coded 'hop' and 'go' to implement the Hop critter. Later, we built the methods one-by-one starting from the methods required by each given critter behavior file. When all the example critters were correctly implemented, more advanced critters were crafted to test the remaining instructions.

Assumpti on

= We assume that if the critter uses the hop method incorrectly, (i.e. when the cell content is already occupied) then the critter uses its turn without moving, turning, eating, or infecting.

Karma Points

= Critters were designed by each person with input and a lot of help from the other partner. They were then pitted against each other to find the strengths and weaknesses in them. The critter named Young plays very defensively by staying in one place and turning until either it infects a critter or it sees an enemy not in front of it. If so, the executing critter runs away. In addition, if the cell in front of the critter is not empty, then the critter turns left or right depending on whether the cell at bearing 45 or 315 is empty. (right has the priority).

Strength: the critter runs away from enemies, and avoids enemy contact; Weakness: critter is not aggressive, and is easily infected by enemies.

= The Vincent critter hops in a straight line until it is blocked, or it sees an enemy. It is much more aggressive because it attempts to chase any enemy that it has an advantage over in terms of angle. If the critter does not have an advantage, it runs away. Strength: Aggressive and hunts for enemies. Weakness: Critter cares more about attacking than escaping from certain death.

= When Young and Vincent co-exist with 30 critters each on the field using GUI, Young prevails over Vincent. However, if each were given 50 or 100 critters, Vincent prevails.

Pair programming log

= Each person took approximately 20 min turns driving, with some flexibility if the turn ends while the driver is in the middle of typing a line. Code was never written or even looked at without both members present until the classes were completed. Comments for each file were written separately, as evidenced by the differences in formatting, and then checked/edited by the other member. Critters were also typed separately, but they were planned together and were typed in the presence of the other member.

= (09/25, at the 1st Blanton Lobby) Young drives for 30min; Vincent drives 50min; Young drives 20min; Vincent 10min

= (09/26, at the 2nd Blanton Lobby and at the Q) Vincent 20min; Young 20min; rotation for 5hr from 4:50pm to 9:30pm (diner break 6:30-7:30)

= (09/28, at the 2nd Blanton Lobby) Vincent 20min; Young 20min; rotation for 1hr 9:35 to 10:35

= (10/02, at the 2nd Blanton Lobby) Vincent 20min; Young 20min; rotation for 1hr from 7:50 to 9:00

= (10/03, at Vincent's dorm room) Vincent 50min alone; Young 1hr and 20min alone (however, discussed the strategy of each critters); Young comments on CritterTest for 20min; Vincent comments on Interpreter for 20min; Young writes report for 10min; Vincent writes report for 10min; Young writes report for 10min; Vincent 10min; Young 10min; Vincent 10min

= (10/04, at Vincent's dorm room) Vincent edits the report for 30min alone; Young writes log for 30min

We are using Vincent.cri in the CritterFest.