COSC343- Evolve a Species

Nick Sparrow (4742998)

Department of Computer Science

University of Otago

May 10, 2016

For this assignment I decided on using java because it’s the language that I have the most experience in and before starting I had a good idea of the layout that I was going to use which was having a Creature class, Monster class and a main class (EvolveSpecies.java). By splitting it into the multiple classes it helped me to clearly see what was essential for just the creature or just the monster. I didn’t see it necessary to make a separate class for the strawberries and mushrooms because they didn’t have any special attributes that I saw requiring it. Java made the structure easy to understand for me however I’ve never done any GUI work with java minus a couple labs in comp160, so getting a handle on the it took a lot of time I ended up using characters to represent the different elements in the simulation; **C** for Creatures, **M** for Monsters, **s** for Strawberries and **m** for Mushrooms. I was in the process of changing them to be coloured rectangles however was beginning to run out of time so decided on getting everything else done first and returning to it if I had time due to a detailed GUI not being a major part of the assignment.

To run the program simply compile and run, it will render a new world and step through however many *timesteps* is specified in the EvolveSpecies data fields then will render the next generation and continue this for as many generations *total\_generations* is set to. The size of the world, amount of creatures, monsters, and probability of there being a strawberry or mushroom can all be easily changed in the data fields also along with the *pause\_time* between the renderings of the frame making it easy to speed up or slow down the simulation.

Throughout the generations I could see that the creatures generally were much more likely to run away from monsters and head towards strawberries. This showed at the end of each generation with the number of surviving creatures increasing and typically having more health than what the previous surviving generation had. To create the off springs I attempted to implement the roulette wheel selection that was discussed in lecture 14 to find the parents. Before creating the next generation I first sorted the array of creatures by their energy levels then find the fitness of each creature by taking the sum of all the creatures energy levels at the end of the generation and then set each creature’s *fitness\_norm* by dividing its *energy\_level* by the sum of the generations energy’s combined. Using that I set the creature’s *fitness\_accum* values by getting the sum of each of the creature’s *fitness\_norm* that was before it in the array, meaning the last creatures in the array are the ones that survived with the most energy and they will have the highest *fitness\_accum.* This determines the likely hood of a creature being selected as a parent because as the creatures that survived with the most energy have the highest *fitness\_accum* they have the best chance of having a higher value than a randomly generated number in the range 0-1.

To make the graph I made the average energy levels of each generation to be added to a file called *graph\_data.txt* that would get created on running, I then put the values into an excel spreadsheet which made it very simple to create the graph:

The data fields that I used in making the graph were:

* World\_width: 40
* World\_height: 40
* Total\_timesteps: 50
* Total\_generations: 10,000
* Num\_creatures: 50
* Num\_monsters: 15
* Chance\_of\_strawb: 0.05
* Chance\_of\_mushroom: 0.05