Computer Science Master, Course: Intelligent Agents
Students: Ivan Gavrilović, Boris Perović

Implementing a first Application in RePast: A Rabbits Grass Simulation

A brief solution description:

The solution was developed using JDK 7 and RepastJ 3.1. Provided template (*rabbits.zip*) was used as a base for the project and has been built upon with the help of tutorial mentioned on the assignment page. The names of the classes are descriptive enough to indicate their purpose.

Since most of the requirements were clearly stated in the assignment, but not all of them, here will be presented important project decisions:

- Constants used for the simulation:

```
// World size
private static final int WORLD_X_SIZE = 20;
private static final int WORLD_Y_SIZE = 20;

// Population constants
private static final int NUM_AGENTS = 30;
private static final int INIT_ENERGY = 8;
private static final int BIRTH_THRESHOLD = 15;
private static final int GRASS_GROWTH_RATE = 30;
```

- Rabbit's energy decreases by one with each step.
- Grass is allowed to grow on a cell that already contains grass. Grasses are merged, summing their individual energies.
- In order to make the simulation meaningful, it is required that both width and height of the grid are greater than 0. If not, user is prompted with message and simulation is stopped.
- Amount of energy that will be transformed into grass when simulation starts is `(int) (width * height * 0.2)`, where `width` and `height` are the dimensions of the grid.
- Distribution of energy after the reproduction is done in a specific way that will be explained in detail.

One of the first things that needs to be considered is the manner in which the rabbit reproduction is implemented. As explained in the assignment, a rabbit first makes a random move to a cell (if the cell is not occupied), then eats whatever grass there is on that cell and finally (method `execute` in `RabbitGrassStep` class), if the total amount of energy it possesses exceeds a certain value (`birthThreshold`), a rabbit reproduces (method `reproduce` in `RabbitsGrassSimulationAgent` class). The offspring rabbit is given initial energy value of floor of half of the energy its parent had. Naturally, the parent is left with a value of ceil of half of its energy before reproduction. Energy distribution is done in such a way as to try to preserve the total amount of energy in the system, mimicking natural world.

The addition of a rabbit to the simulation (at start or an offspring) is done through a maximum number of 10 * `WORLD_X_SIZE * WORLD_Y_SIZE` attempts. If a rabbit is not successfully placed on the grid in those attempts it is discarded, and user is presented with a message that some rabbits could not be placed.

The simulation step implemented through a method `execute()` of an inner class `RabbitGrassStep` of the method `buildSchedule()` of the `RabbitsGrassSimulationModel` class. First the grasses are planted randomly on the grid (initially `GRASS_GROWTH_RATE`, adjustable using slider). Next the rabbits' states are upadated. If a rabbit has an energy value that is below or equals to zero, it is considered dead and removed from the simulation. Conversely, if the energy value is a positive integer, the rabbit makes a step and tries to reproduce. The potential offspring rabbit is added to the simulation. When the number of rabbits reaches zero, the simulation stops.

Grid displays rabbits and grass. Cell with grass that has more energy is brighter. Rabbits that have amount of energy greater than 10 are colored red, others are colored yellow.

Results:

Charts show amount of energy and number of rabbits during time. Blue line represents the amount of energy available, and red line is the number of live rabbits. As shown on the plot below (*Figure 1*), the default values of the simulation provide us with simulation that have some periods of smaller variation, with occasional peaks. Increase of number of rabbits, induces reduction of energy count. Leaving the default parameters and incresing the grid to 40 by 40 removes all the rabbits at one moment (*Figure 2*). But if we reduce the `birthThreshold` to 0, increase the start `numRabbits` to 120 and `grassGrowthRate` to 100 we can reach state where we have population of rabbits varying from 100 to 300 (*Figure 3*).

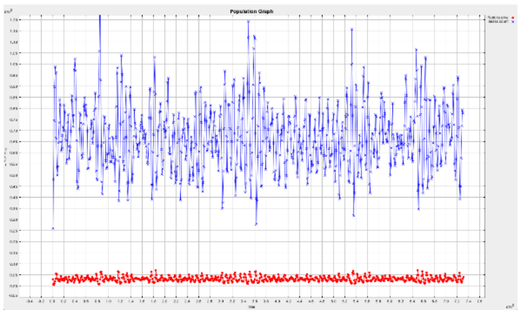

Figure 1 - BT 10, GG 40, NR 30, X 20, Y 20

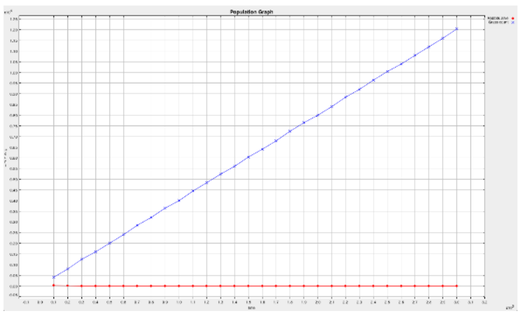

Figure 2 - BT 10, GG 40, NR 30, X 40, Y 40

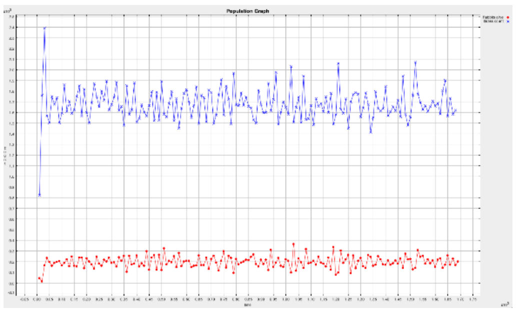

Figure 3 - BT 0, GG 100, NR 120, X 40, Y 40

Abbreviations: BT – Birth Threshold, GG – Grass Growth rate, NR – Number of Rabbits, X – width of the grid, Y – height of the grid.