Excercise 1.
Implementing a first Application in RePast: A Rabbits Grass Simulation.

Group №10: Cosmin-Ionut Rusu, Sorin-Sebastian Mircea

October 1, 2019

1 Implementation

We followed the provided tutorial which served as a base for our implementation. We carefully adopted the architecture described in the RePast tutorial and implemented the other requirements to fit our specific problem requirements. We've also followed Object-Oriented Programming principles as guidelines for the project.

Visualization

In our graphical visualization of the simulation, the grass is coded with the green color, and the rabbits are coded with red, for best contrast. Empty cells are represented by black.

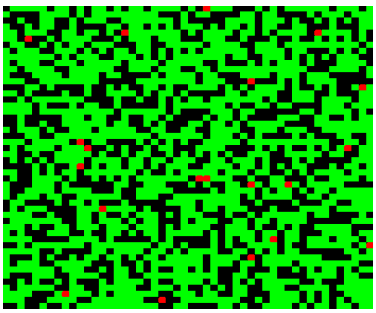

Figure 1: Simulation world

Charts

To better visualise and understand the world we have created two charts:

1. Showing the evolution of the grass available in the world and the population of rabbits. The rabbit population is multiplied by a constant (in our case 100) to bring them to similar scales.
2. Histogram of the rabbits energy at the current step

Extra parameters

We considered that the the the following parameters would need to be added (besides the already provided ones) in order to have a better control of the world:

- **EnergyLostByMoving** - the energy that a rabbit is going to lose for moving;
- **EnergyLostByReproduction** - the energy that a rabbit is going to lose in order to reproduce (once it reaches the reproduction threshold);
- **MinRabbitInitEnergy** and **MaxRabbitInitEnergy** - rabbits are randomly going to receive a value from the interval as initial energy.

1.1 Assumptions

Actions order

The order in which our world is going to execute the possible actions can have an impact on the outcome (even thought this is negligible). Arbitrarily we chose the following order (that is executing during one step):

1. Move agents
2. Agents eat (if they are on a grass)
3. Agents lose energy
4. Agents die (if energy = 0)
5. Agents give birth (if $energy \geq birthThreshold$)
6. New grass tiles are spawned

Grass spawning

Grass tiles cannot collide with other grass tiles (during spawning). We had to decide if we were going to spawn the grass only on the tiles that are unoccupied by the rabbits, or randomly everywhere. We chose the second one. When this happens, the rabbits are instantly going to benefit by the extra food, allowing them a second change before they might die. We believe this simulates a real life environment where rabbits can eat grass grown overnight.

Initial energy levels

We also make sure that the initial energy level is always below the $birthThreshold$. This is a fair assumption since if a rabbit were to be born with an $energylevel \geq birthThreshold$, then it might reproduce right away and we will have more rabbits than we expected (set by $numInitRabbits$ variable).

Rabbit in impossibility of movement

In the case that a rabbit cannot move due to it being surrounded by other rabbits, we chose for the rabbit to stay on place but still lose energy (as a penalty for not being able to search for food).

Collision system

Rabbits cannot collide with other rabbits (during spawning and movement), it can move onto a grass tile (and at that moment the grass will be consumed). A drawback on this iterative process is that the order in which we move the rabbits have an impact on the possibility of the movements of the individual rabbits. For example, if two rabbits were next to each other, after one of them moves, the other one will now be able to move to the position where the first one was initially. We assumed we always move the agents in the order they were generated.

1.2 Implementation Remarks

Movements

For moving the rabbits one step in the 4 adjacent cells, we defined two constant delta vectors. The first vector tells the change of the x position $dx = [0, 1, 0, -1]$ and the second vector tells the change of the y position $dy = [1, 0, -1, 0]$. We believe this is a very elegant solution and in order to make a legal move we first draw a random integer d in the interval $[0, 3]$ and then use the following formulas to update the directions:

$$newx = (x + dx[d] + N)\%N$$
$$newy = (y + dy[d] + M)\%M$$

Where N and M are the number of rows and the number of columns the matrix, respectively. This formula takes care of all the corner cases and will make the space behave like a torus.

2 Results

Unless otherwise stated, the value of each variable is the default one depicted in the table 1.

Parameter	Default Value
GridSize	50
NumInitRabbits	20
NumInitGrass	1500
GrassGrowthRate	10
BirthTreshold	20
MinRabbitInitalEnergy	10
MaxRabbitInitalEnergy	20
EnergyLostByReproduction	10
EnergyFromEeating	5
EnergyLostByMoving	1

Table 1: Default values for the parameters

2.1 Experiment 1

2.1.1 Setting

In this experiment, we changed the default values for the number of rabbits. We experimented with 20, 100 and 300 rabbits to see how this affects the long term behaviour of the population.

2.1.2 Observations

Eventually all of the population got extinct, no matter how big the population was. The reason was that our grass growth rate is too low, and the population reaches a point when there is not enough available grass or it is too far, hence dying.

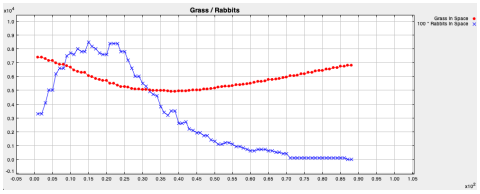

Figure 2: 20 rabbits

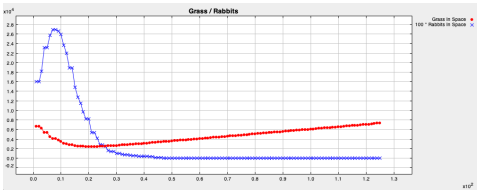

Figure 3: 100 rabbits

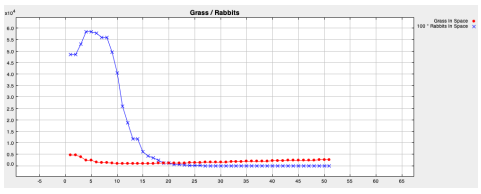

Figure 4: 300 rabbits

2.2 Experiment 2

2.2.1 Setting

Following our previous experiment, we decided to changed the growth rate of the grass and see how it impacts the whole population. The rabbit population size was 20 and we variate the grass growth rate to 20, 100 and 300.

2.2.2 Observations

When the population equals the grass growth rate, the population of rabbits fluctuates and eventually get extinct. However, we saw that for higher growth rates, the rabbits population will tends to have smaller variation (i.e. converge). We found this is true even if set different birth thresholds.

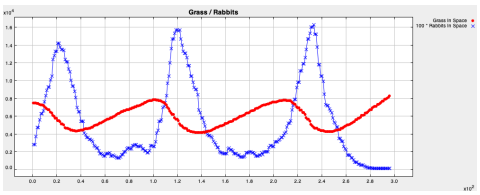

Figure 5: 20 growth rate

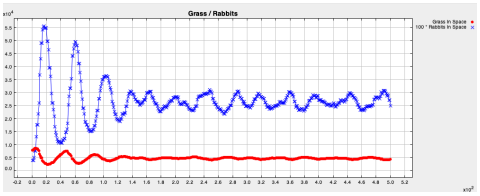

Figure 6: 100 growth rate

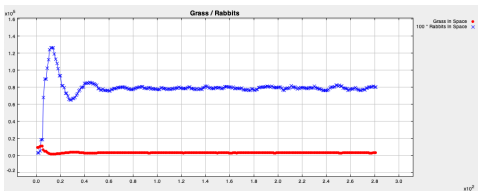

Figure 7: 300 growth rate

2.3 Experiment 3

2.3.1 Setting

Based on our previous experiments, we decided to change the birth threshold and the energy from eating a unit of grass. We used different values for this.

2.3.2 Observations

The behaviour of the population over the long term didn't affect the same patterns we saw previously. The population is still either converging or dying based on the growth rate of the grass.