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

Group №41: Mohammadreza Ebrahimi, Salar Rahimi

September 29, 2020

1 Implementation

In this section, we briefly describe the general framework of the project. The first script that is responsible for running the entire project is *MainRabbit.java* for which the user should provide two arguments. First one is always set to "" and hence the user can set the paramaters in GUI. The second argument determines that whether the program is going to be executed from GUI or not that is naturally a boolean.

The most important java script is *RabbitsGrassSimulationModel* that defined the skeleton of the project. In this file we build the model, determine the schedule and determine and initialize the main parameters of the setup.

In *RabbitsGrassSimulationAgent.java* Charactrestics of the agents, which in this project rabbits, are defined.we move the agents randomly around and add/subtract energy to/from the initial energy of each agent. In other words, agents behavior toward it's environments is defined.

Finally, in *RabbitsGrassSimulationSpace*, we define the 2D grid of the world and manage this space by placing grass and agents randomly in unoccupied cells. Also, we keep track of the existing agents and grass for plotting purposes.To summarize, the world and it's rules are defined in order to complete the communication between elements of the world.

1.1 Assumptions

There are number of core assumptions around which we base the reasoning of the simulation. The first main assumption is that in one cell, we can place multiple grass objects, i.e., the grass concentration may differ. This means the grass on one location can grow over time, and also new grass can grow at random locations. By applying a set of parameters that will be mentioned later, this assumptions leads to the saturation of the whole 2D space by grasses. Each rabbit would consume all the available grass on a single location (Value based on the intensity).

Another chief assumption is that we handle the collisions in a one time attempt fashion meaning in case that the target cell is occupied the agent doesn't move for that step. However, even if the agent doesn't move it loses the same amount of energy as if it has moved.

1.2 Implementation Remarks

For implementing torus grid, we manually move the agents around when they reach the boundaries of the map. This feature is provided for Repast-Symphony but we had to implement it by using if statements since we solely rely on RepastJ.

Naturally we assume the reproduction energy to be way higher than that of movement. Also, the initial energy shouldn't be enough for one reproduction of rabbits. Furthermore, we provided the most flexible case for the user that is running the .jar file meaning that all the decision parameters can be tweaked so that the contribution of all the parameters is potentially underlined.

2 Results

This section is aimed for providing different experiment parameter sets, the outcome of such setting and the intuitions that lead to this behaviour. We strive to bring experiment setups that correspond to natural phenomenons so that we can emphasise the validity of the proposed model.

Indeed, due to the very high flexibility of the parameters, the results of each of the following experiments can be reproduced with unconfined number of other settings.

2.1 Experiment 1

For the first experiment we tried to set the parameters in a manner that leads to equilibrium. The balance of the energy gained from eating grass and initial energy are appropriately in balance with the energy consumed by moving or reproducing.

2.1.1 Setting

InitialEnergy = 100 , NumInitRabbits = 20 , NumInitGrass = 20 , GrassGrowthRate = 5 , EnergyGainedFromGrass = 20, ReducedEnergyPerReproduce = 40, ReducedEnergyPerMove = 5, BirthThreshold = 120

2.1.2 Observations

By selecting the above setting for the parameter set, we reach to equilibrium. It might be the case that at different stages of the simulation the balance is not maintained but for this experiment and the proceeding ones we waited enough to observe that neither of the objects extinct or dominate the world. One can refer to Figure 1.

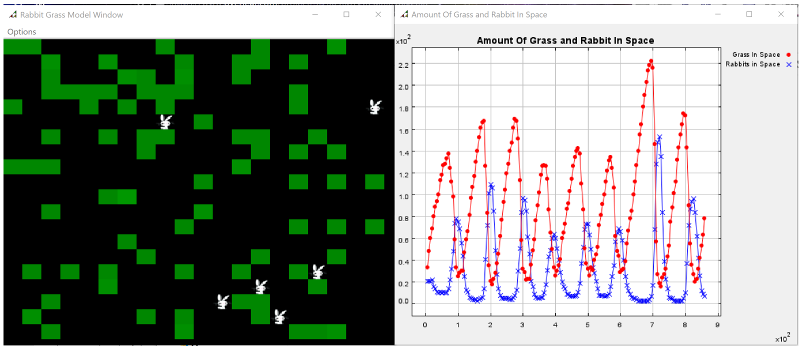
Rabbit Grass Model Window
Options
Amount Of Grass and Rabbit In Space
Amount Of Grass and Rabbit In Space
x10^2
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
6
7
8
9
x10^2
Grass In Space
Rabbits in Space

Figure 1: Simulation and plot results of the first parameter set

One of features of this setting is that it makes sense with the natural intuition of the phenomenon. Also, the 2D map doesn't get saturated meaning when the population of the rabbits grow, the total number of grasses decrease and there is not enough food for everybody to reproduce and therefore the majority of the rabbits will die after a few steps and the system recovers its natural balance.

2.2 Experiment 2

For the second experiment, we increased the energy consumption of each move to double to simulate an extreme living condition.

2.2.1 Setting

ReducedEnergyPerMove = 10 (rest are same as the first case)

2.2.2 Observations

Expectedly, this will lead to the distinction of rabbits which leads to unbounded generation of grass (we have constant grass growth rate). The blue curve (number of alive rabbits) converges to zero quickly and population grass curve gets saturated at 400 which is number of cells times the maximum density of grass a each cell. This result is evident in Figure 2.

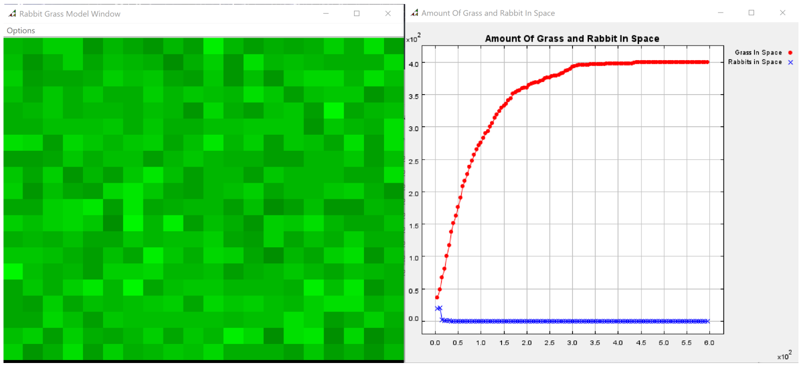

Figure 2: Simulation and plot results of the second parameter set

2.3 Experiment 3

In this experiment we like to simulate the limited source case. The only way to simulate that is to put grass growth rate to zero. To achieve the case that we have some rabbits and no grass, we put too many rabbits initially and enough grass to have some steps.

2.3.1 Setting

NumInitRabbits = 100 , NumInitGrass = 200 , GrassGrowthRate = 0 (rest are same as the first case)

2.3.2 Observations

It's simple to observe that in the case of limited food source, the living agents quickly vanish. This is a very natural consequence of the tailored parameters observable in Figure 3.

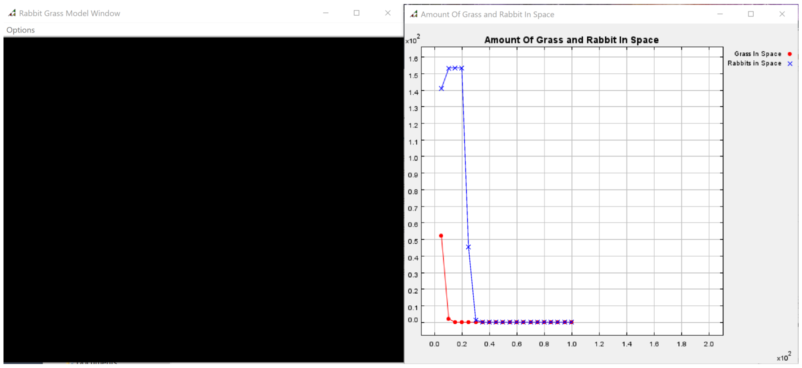

Figure 3: Simulation and plot results of the third parameter set