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

Group №: 24: Lucas Burget, Lucas Waelti

October 2, 2018

1 Implementation

1.1 Assumptions

World Configuration The world is built on a discrete torus grid where grass can grow and rabbits can move around. Each cell can only contain **one unit** of grass at a time as well as maximum one rabbit. This allows a simpler visualization of the space by directly seeing if there is something or not at a given position (it is possible that a rabbit covers some grass though).

Rabbits (agents) Two rabbits (agents) can never be on the same position. At each step, each rabbits move left/right/up or down. If they encounter some grass after moving, they eat it and gain **a given amount** of energy. Otherwise, they lose **one unit** of energy. If they have enough energy, they will reproduce.

Grass The grass initially covers 50% of the grid. The parameter GrassGRate (Grass Growing Rate) defines how much grass must be spread across the grid at each time step. As mentioned above, there can only be one grass object at a given place in the grid at a time. The energy a unit of grass delivers to a rabbit is contained in the parameter GrassEnergy and is set to 1 by default. It means the grass amount per cell is bounded.

1.2 Implementation Remarks

Procedure during one time step First, all rabbits who are out of energy are killed and removed from the simulation.

Then, it is checked whether any rabbit can reproduce. The rabbit class (RabbitsGrassSimulationAgent) indicates if it has enough energy to reproduce. If it is the case, it autonomously **divides its own energy by half** and indicates it can reproduce. The model will then spawn a new rabbit at a random location in the grid. If no rabbit can be spawned (no more room), the one that reproduces, gets its energy divided by 2 anyway. This allows a more dynamical behavior of the population.

All rabbits can now move by choosing a new location and going onto it, if it is valid. If some grass is encountered, the rabbit can eat it and gain energy. Otherwise, it just looses one unit of energy.

Once each rabbit could update, some grass is added to the grid in quantity corresponding to the parameter GrassGRate. More on that later.

The display is then updated to show the changes in the model.

How rabbits choose their target location and move towards it Each rabbit will randomly choose to move on one of the adjacent cells. It is checked whether the chosen location is not currently occupied by another rabbit and if it is free, the rabbit moves towards it. Otherwise, a new destination selection sequence is started. An agent can try up to 4 times to find a new target location. If it fails to find a new place, it will just stay where it is and therefore lose some energy. Indeed, it will not be able to eat the grass that may grow below him at the end of the step as grass grows only after all agents tried to eat the grass in the cell they just moved to.

Rabbits color code To make the simulation a little more readable, a simple color code was implemented to the rabbits' display:

- White: This is the standard color of a rabbit.
- Gray: The rabbit has low energy level (energy = 1). It might die at next time step.
- Red: The rabbit will reproduce at next time step as its energy exceeds the birth threshold.

How is grass implemented Grass is implemented through Integer objects spread across the Object2DGrid object of Repast. A grass unit has always a value of energy defined by the **GrassEnergy** parameter. Only one Integer object can occupy a cell at a time.

Details about the grass spreading As only one unit of grass can be assigned to a location, when spreading grass across the grid, the program will try to find a free spot at random. It does not have an infinite amount of tries and therefore it is possible that less grass grows at a certain time step than what is indicated with the grass's grow rate. It made sense to proceed in this way because if some grass cannot be placed, it means that the space is already saturated with grass, hence the population of rabbits is low or has even vanished.

Case where a rabbit and some grass exist together at the same location It is possible for a rabbit to be on a location with some grass there at the same time. This can happen when every rabbit has moved and eaten the grass found where they went and that some new grass was spread across the grid.

Parameters Here is a list of parameters that can be modified to tune the model:

- AgentEnergy: Energy all rabbits have when created.
- BirthThres: Energy required for a rabbit to reproduce.
- FullSpeed: Simulation speed. Uncheck to have a delay of 1 second between steps.
- GrassEnergy: Energy delivered by one unit of grass.
- GrassGRate: Number of grass units to be spread across the grid at each step.
- GridXSize: X-size if the grid (number of cells, call setup before changing this parameter).
- GridYSize: Y-size if the grid (number of cells, call setup before changing this parameter).
- NumAgents: Number of agents at the beginning of the simulation.

2 Results

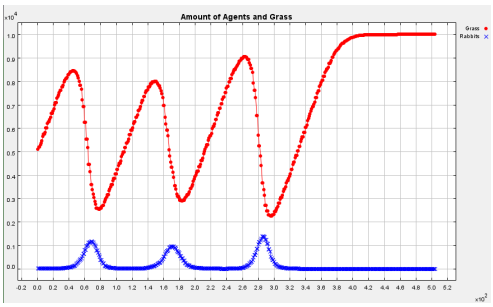
2.1 Experiment 1

2.1.1 Setting

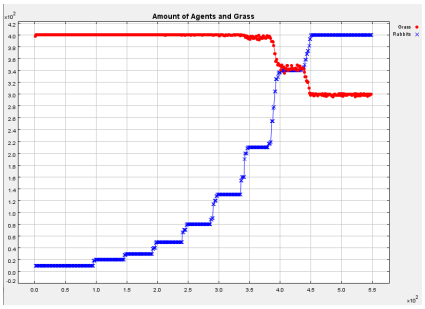
First, let us consider a simulation using the default parameters: AgentEnergy=5, BirthThres=10, GrassEnergy=1, GrassGRate=20, GridXSize=20, GridYSize=20, NumAgents=10.

2.1.2 Observations

The rabbit population as well as the amount of grass oscillate pretty strongly but both evolve around their respective average (≈ 40 and ≈ 240 respectively). As we expected, both populations are inversely proportional.



(a) Evolution of the rabbit population and the grass quantity in case of Experiment 2



(b) Evolution of the rabbit population and the grass quantity in case of Experiment 3

Figure 1: Experiments 2 and 3 illustrations (rabbits: blue, grass: red)

2.2 Experiment 2

2.2.1 Setting

Let us use some more extreme parameters this time: AgentEnergy=9, BirthThres=10, GrassEnergy=1, GrassGRate=100, GridXSize=100, GridYSize=100, NumAgents=10.

2.2.2 Observations

As we can see in Figure (1a), the chosen settings provoke bursts in the rabbits population, which cause the grass amounts to collapse. Then the population of rabbits goes near to zero while the grass quantity builds up again. Here the population vanished after three bursts.

2.3 Experiment 3

2.3.1 Setting

With parameters: AgentEnergy=5, BirthThres=100, GrassEnergy=1, GrassGRate=300, GridXSize=20, GridYSize=20, NumAgents=10.

2.3.2 Observations

The grid is saturated with grass. The fact that agents have a very high birth threshold produces this exponential sort of growth as seen in Figure (1b). The "stairlike" shape of the curve is due to the fact that most of the rabbits reach their reproducing threshold at the same time.