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

Group №: 14; Launay Clement

October 2, 2017

1 Implementation

1.1 Assumptions

In addition to the requirements given in the exercise statement, the following assumptions and choices have been made:

- Rabbits' energy : Rabbits start with a set given number of energy which can be modified with the variable *initialEnergy*.
- Grass : As suggested by NetLogo simulation, each patch of grass is worth a certain amount of energy. It is controlled by the variable *grassValue*. With the requested *grassGrowthRate*, it allows for more precise tuning. It is also assumed that grass can spawn anywhere on the grid at any time, increasing the worth of a cell in energy if it happens to already contains some grass. There is no limit to the amount of grass one cell can contains. However, rabbits always eat all the grass available on the cells they move to.
- Rabbits' collisions: Considering that newborn rabbits should randomly be added to the grid but cannot overlap with others, the method responsible for this, *addNewRabbit* may fail if it does not find an appropriate space after a certain number of tries. In that case there are likely very few spots actually available, so the number of rabbits is too high to be affected. Another issue is rabbits trying to move to a cell already occupied by another rabbit. In that case they do not get to pick another destination and simply do not move for this turn (they still lose energy though).

1.2 Implementation Remarks

- Schedule order: The different part of the *step* function are resolved in the following order.
  1. A *grassGrowthRate* number of patches of grass are randomly added to the space.
  2. One after the other, each rabbit eats all the grass at its position, gaining the corresponding energy, then choose a random direction to move in, effectively moving if the destination cell is free, and loose one energy
  3. Each rabbit is checked to see if it has died (energy=0) and should be remove, or if it should give birth, reducing its energy and adding a new rabbit in the space.
- Movements : Rabbits move by picking a random number between 0 and 3 which tell them how they should try to modify their coordinates (e.g. $0 \longrightarrow NORTH \longrightarrow (0,1) \longrightarrow (x+0, y+1)$). They can then compute their destination's coordinates with a modulus to comply with the torus space design and request the simulation space to handle the movement.

- Simulation feedback : Alongside the simulation, two plots display the evolution of the rabbit population and the amount of grass energy available on the field. They have separated windows because their scales can greatly vary depending on the simulation's parameters. It is also possible to access and modify the details of any given cell by clicking on them. On the display, rabbits are represented by white circles while cells with grass are colored with brown when empty and with darker and darker shades of green as their amount of grass increase (up to black when they reach 16 patches).

2 Results

2.1 Experiment 1

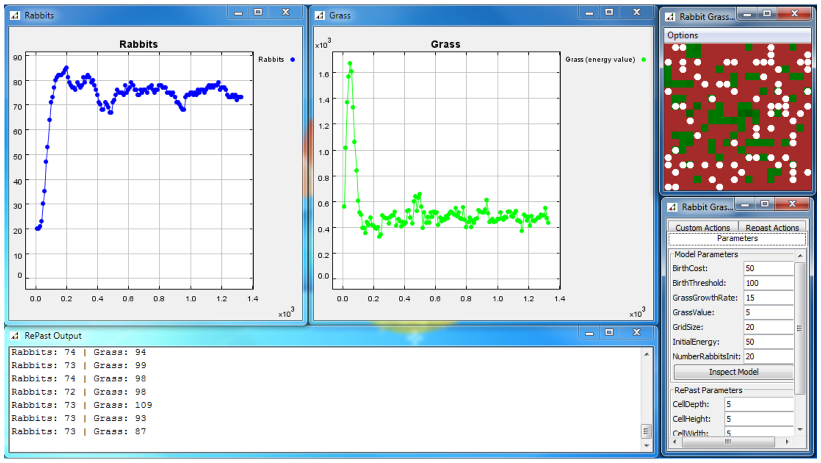
2.1.1 Setting

birthCost	50	gridSize	20
birthThreshold	100	initialEnergy	50
grassGrowthRate	15	numberRabbitsInit	20
grassValue	5		

A simple experiment with the default values to observe the evolution of the model. The parameters related to birth are a bit intuitive with $initialEnegery + birthCost = birthThreshold$

2.1.2 Observations

The evolution of the grass and rabbit populations first undergo a transient phase. At the beginning grass have a small window where its number reaches its peak because it is not spread everywhere while rabbits are underpopulated. Since rabbits move at random, they actually rely on grass spawning near them than moving to it. This quickly happens and the number of rabbits then skyrockets (all rabbits tend to give birth at around the same time). Both populations then stabilize, with the final number of rabbits being around 74. The final regime is still subject to small oscillations, clearly visible on the rabbit graph (grass graph is more edgy because rabbits eat all grass in a cell). They are due to the model being unstable by nature: it is a predator-prey systems where the rabbit population grows until it surpass grass's natural growth, at which point some rabbits dies from the lack of grass, allowing more grass to grow since there are now less rabbits, and the cycle starts again. This is coherent with the final average rabbit population being close to $75 = 15 * 5 = GRASSGROWTHRATE * GRASSVALUE$



Model Parameters
BirthCost: 50
BirthThreshold: 100
GrassGrowthRate: 15
GrassValue: 5
GridSize: 20
InitialEnergy: 50
NumberRabbitsInit: 20

RePast Parameters
CellDepth: 5
CellHeight: 5
CellWidth: 5

```
Rabbits: 74 | Grass: 94
Rabbits: 73 | Grass: 99
Rabbits: 74 | Grass: 98
Rabbits: 72 | Grass: 98
Rabbits: 73 | Grass: 109
Rabbits: 73 | Grass: 93
Rabbits: 73 | Grass: 87
```

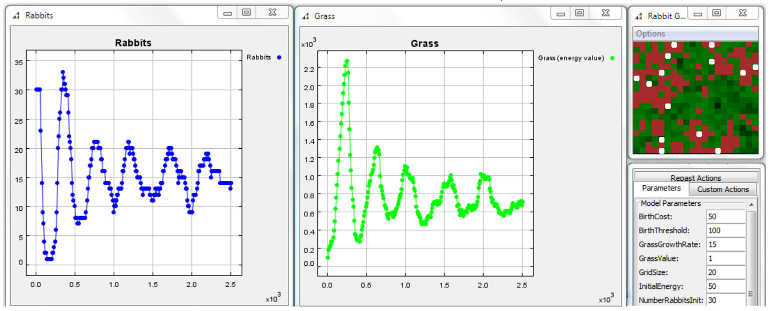
2.2 Experiment 2

2.2.1 Setting

birthCost	50	gridSize	20
birthThreshold	100	initialEnergy	50
grassGrowthRate	15	numberRabbitsInit	30
grassValue	1		

2.2.2 Observations

In this experiment, the initial number of rabbits is deliberately higher than the available grass production. Hence the rabbit population starts by declaiming, which is further guaranteed by the *grassValue* being 1 (so rabbits are very unlikely to be able to reproduce before most of them went extinct, allowing path of grass to start being actually worth more than the cost of moving to them). We can then see the establishment of the stable regime with damped oscillations (that will not actually fully disappear).



BirthCost: 50
BirthThreshold: 100
GrassGrowthRate: 15
GrassValue: 1
GridSize: 20
InitialEnergy: 50
NumberRabbitsInit: 30

2.3 Experiment 3

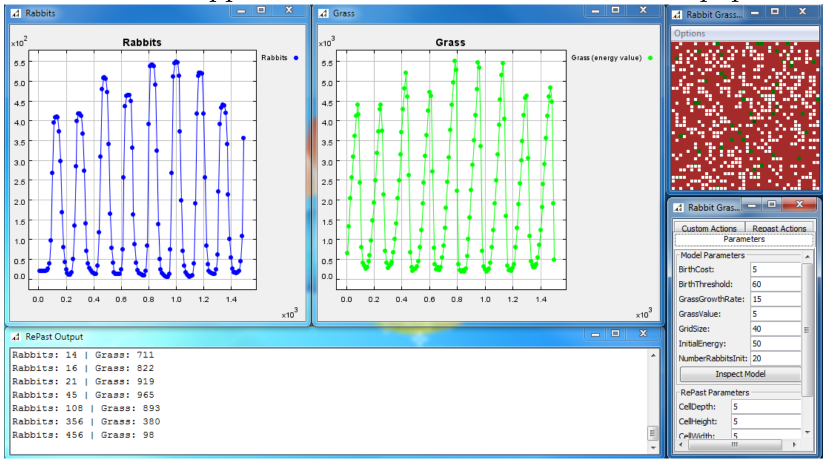
2.3.1 Setting

birthCost	5	gridSize	40
birthThreshold	60	initialEnergy	50
grassGrowthRate	15	numberRabbitsInit	20
grassValue	5		

These values where found by tinkering with the parameters to obtain the wanted result.

2.3.2 Observations

This experiment features more extreme values for the variable in order to emphasis the inherent instability of this simulation. The results are opposite harmonic oscillations for both populations



```
Rabbits: 14 | Grass: 711
Rabbits: 16 | Grass: 822
Rabbits: 21 | Grass: 919
Rabbits: 45 | Grass: 965
Rabbits: 108 | Grass: 893
Rabbits: 356 | Grass: 380
Rabbits: 456 | Grass: 98
```
BirthCost: 5
BirthThreshold: 60
GrassGrowthRate: 15
GrassValue: 5
GridSize: 40
InitialEnergy: 50
NumberRabbitsInit: 20
CellDepth: 5
CellHeight: 5
CellWidth: 5