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

Group 47: Elia Anzuoni (312827), Francesco Ballestrazzi (312494)

September 22, 2020

1 Implementation

Here, we detail the assumptions we made about the world model, and lay out the most important implementation choices.

1.1 Assumptions

We let the grass grow without bounds in each cell.

Besides the ones prescribed, we implemented sliders for the initial energy of a new-born agent, and for the energy gained by an agent after eating one unit of grass (we assumed that an agent eats all the grass in the cell it occupies, so that the energy it gains is $UnitGrassEnergy \cdot NumGrass$). To the contrary, we fixed the energy lost at every step to 1.

If the move chosen by an agent would make it collide with another, the move is not performed and the agent stays in the same cell, and is not given the chance to choose another move. In this case, the agent still loses one unit of energy, but is allowed to eat whatever amount of grass has grown in its cell during that time step.

We fixed the energy loss associated to reproduction to equal the initial energy of the new-born agent.

1.2 Implementation Remarks

We let the agents choose their move in a new random order at each time step, so as to avoid favouring one particular agent over the others.

New-born agents are spawned in a random cell of the grid, not near the father.

2 Results

Here, we present the outcomes (defined as the evolution over time of the rabbit population and the total amount of grass) associated to various experiments (identified by a concrete choice of the user-settable variables). The grid size was always left to 20.

Both quantities were consistently observed to stabilise, after an exponential-looking transient, around an equilibrium, which was perturbed by random fluctuations.

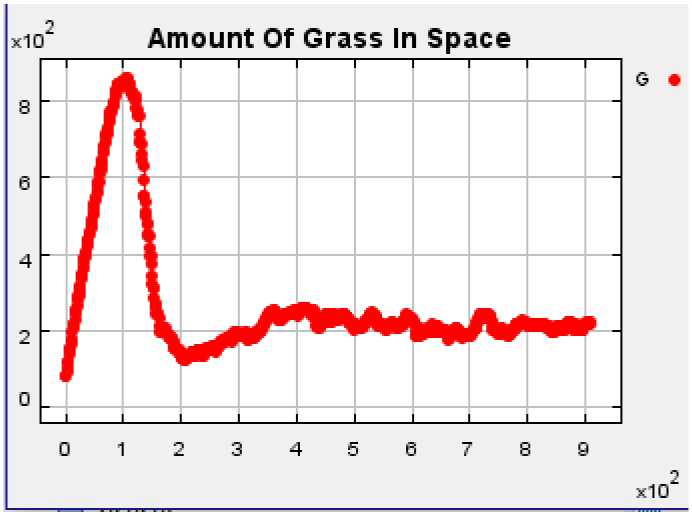
Quite surprisingly, the equilibrium rabbit population was (empirically) found to be $GrassGrowthRate \cdot UnitGrassEnergy$, while the equilibrium amount of grass was found to be inversely proportional to the $UnitGrassEnergy$.

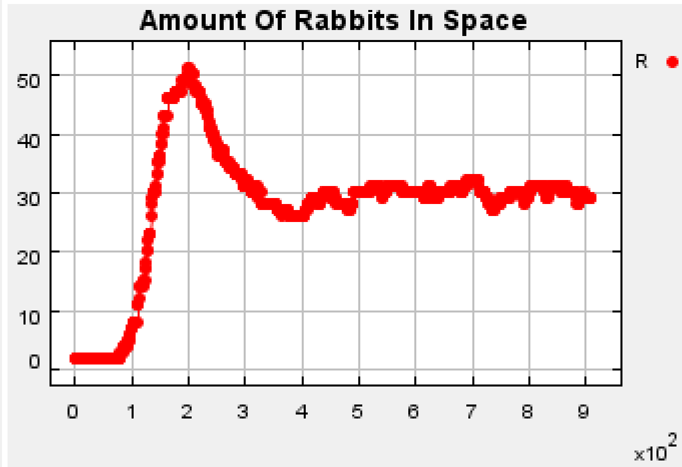
2.1 Experiment 1

2.1.1 Setting

AgentInitEnergy = 30; BirthThreshold = 100; GrassGrowthRate = 10; NumInitGrass = 70; NumInitRabbits = 2; UnitGrassEnergy = 3.

2.1.2 Observations





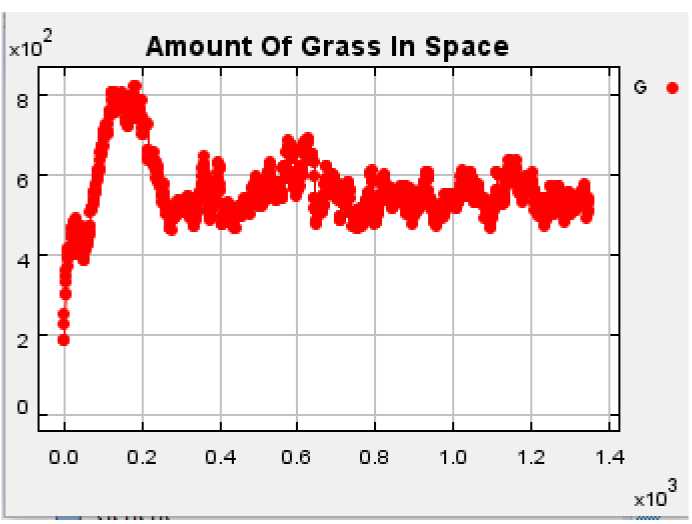
Both curves show an initial transient with a spike, which then stabilises towards an equilibrium regime, perturbed by stochastic fluctuations. As was noted, the equilibrium for the rabbit population is $GrassGrowthRate \cdot UnitGrassEnergy = 30$

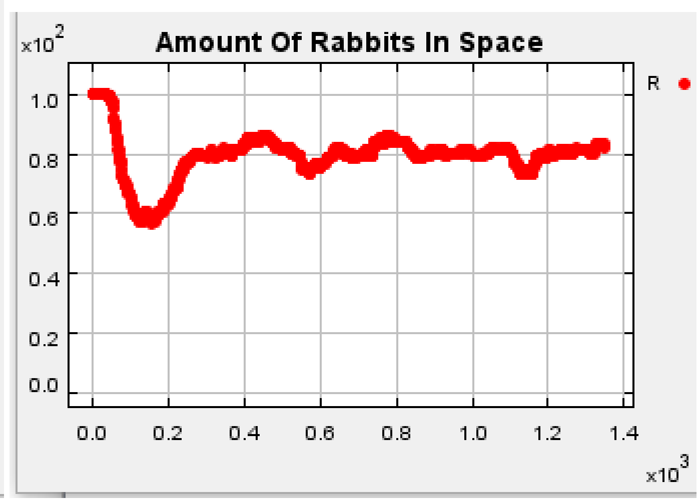
2.2 Experiment 2

2.2.1 Setting

AgentInitEnergy = 30; BirthThreshold = 100; GrassGrowthRate = 80; NumInitGrass = 150; NumInitRabbits = 100; UnitGrassEnergy = 1.

2.2.2 Observations





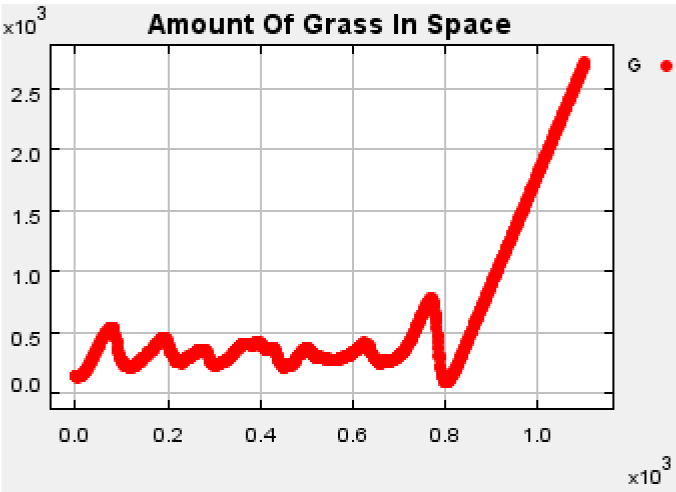
As was pointed out, the $UnitGrassEnergy$ has become a third of its previous value, and this resulted in a tripled value for the equilibrium amount of grass. This graph also shows that $NumInitRabbits$ and $NumInitGrass$ are completely inconsequential to the long-term evolution for the two quantities.

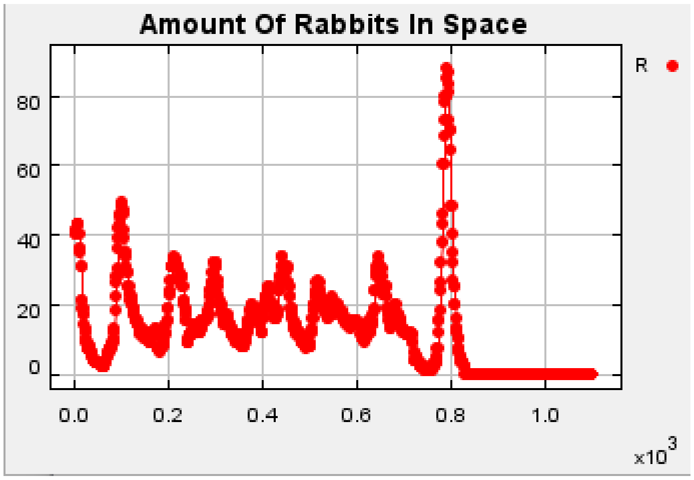
2.3 Experiment 3

2.3.1 Setting

AgentInitEnergy = 10; BirthThreshold = 15; GrassGrowthRate = 9; NumInitGrass = 150; NumInitRabbits = 40; UnitGrassEnergy = 2.

2.3.2 Observations





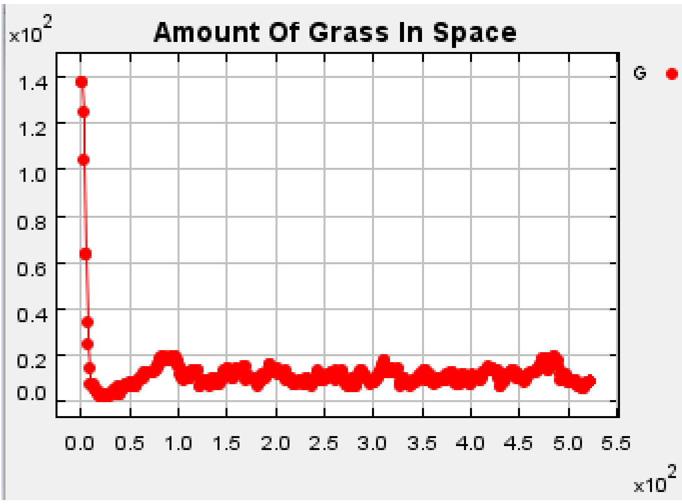
When the equilibrium rabbit population is low (i.e. comparable to the fluctuations) it is likely that, sooner or later, the population hits 0, and the rabbits go extinct (obviously forever).

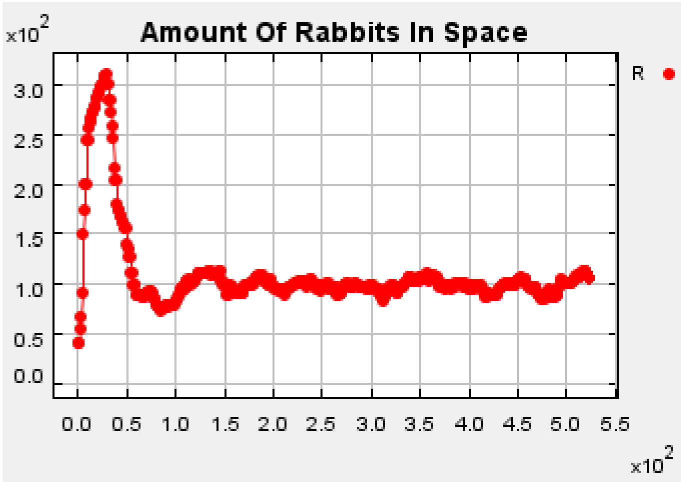
2.4 Experiment 4

2.4.1 Setting

AgentInitEnergy = 30; BirthThreshold = 50; GrassGrowthRate = 2; NumInitGrass = 150; NumInitRabbits = 40; UnitGrassEnergy = 50.

2.4.2 Observations





The total extinction of rabbit does not happen when the equilibrium amount of grass is low because, even if the grass is 0 at some point, it still regenerates at the next time step, unlike rabbits.