

INTELLIGENT AGENTS

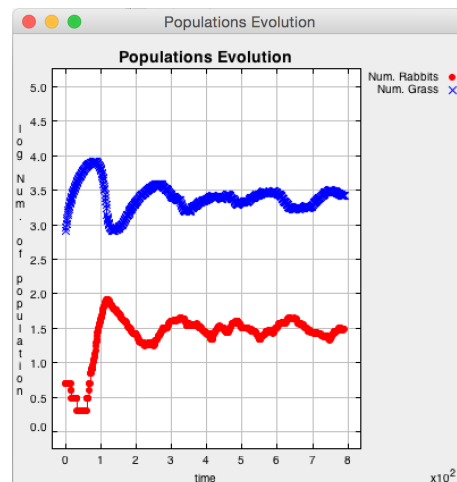
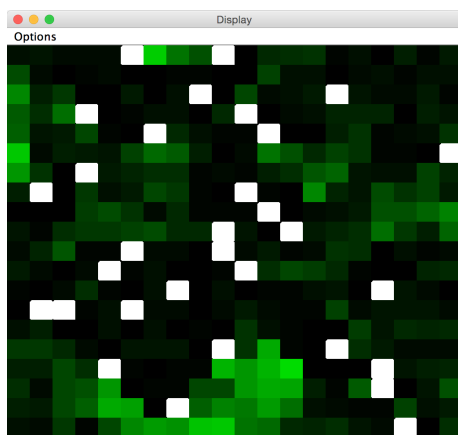
CS-430

Lab1: Rabbits

Submitted By Group 36:

Christopher Salotti & Marco Antognini

September 2015



Introduction

In this first lab we implemented a very simple biological model of rabbits living in a discrete, torus world made of grass and nothing else; a relatively peaceful paradise for these little fluffy creatures.

In this simulation, our anosmic rabbits wander around blindly and eat some grass they can find. When they are fat enough, they perform an asexual reproduction and generate an offspring exactly identical to its parent, except for its level of energy and its position on the world since no rabbit is allowed to be on top of another one.

The amount of grass on each discrete cell of the world is also discrete and ranges from 0 (no grass at all) to 500¹. In our universe, the time is governed by a discrete clock. At each clock tick, the total amount of grass of the world is increased by a parametrised amount which is randomly distributed onto the different cells.

Additionally, at each clock tick, every rabbit agent does the following:

1. It randomly chooses one of the four direct neighbouring cells and if it is currently rabbit-free, the agent moves there and loses `MoveEnergyCost` amount of energy; if the destination is currently held by another rabbit, it simply stays where it currently is.
2. It eats at most `MaxEatQuantity` amount of the grass available on the current cell it is on.
3. It consumes a bit of energy (`EnergyConsumptionRate`) for biological purposes.
4. If it has enough energy for reproduction (i.e. more than `BirthThreshold`) and a neighbouring cells is free, then it duplicates itself; the new creature has an amount of energy equals to `InitialAgentEnergy` and the parent rabbits loses that same amount of energy in order to preserve the total amount of energy.
5. If the agent has no energy (i.e. less than or equal to zero), the rabbit is considered as dead and removed from the world.

Simulation Parameters

In our simulation, the following parameters can be used to tune the system:

GridSize The size of the world; cannot be modified after the simulation was started.

¹which was arbitrarily chosen due to some limitation of RePast's `ColorMap` class that doesn't have a default colour

GrassGrowthRate The amount of grass being added per clock cycle into the world.

InitialAmountOfGrass The overall amount of grass available in the world when the simulation starts.

InitialRabbits The number of rabbits already living when the simulation start.

InitialAgentEnergy The initial amount of energy a rabbit has when it is given life.

BirthThreshold How much energy a rabbit must have to reproduce.

MaxEatQuantity The maximal amount of energy a rabbit can eat per clock cycle.

MoveEnergyCost The amount of energy a rabbit consumes when it moves to another cell.

EnergyConsumptionRate The amount of energy a rabbit loses per clock cycle, in addition to the energy consumed for moving around.

Graph Evolution In our simulation, with default parameters, we see that the populations of grass and rabbits, even if they oscillate, stabilize around a certain mean. We plotted in log scale to have a good vision of the two populations.

Implementation

The simulation was implemented using RePast in, hopefully, a usual manner. The model class (`RabbitsGrassSimulationModel`) manages the different parameters of the simulation, plus a list of rabbit agents (instances of `RabbitsGrassSimulationAgent`) and a representation of the world capable of being displayed (`RabbitsGrassSimulationSpace`), in addition of the traditional scheduler. The simulation program can be started using the main function of the `MainRabbit` class

We used `uchicago.src.sim.gui.ColorMap` to graphically represent the amount of grass on each cell using RGB colors with this mapping formula:

$$\text{amount of grass } i \mapsto (R = 0, G = \min \{255; i * 5\}, B = 0)$$

How to run the simulation In a terminal, simply run “`java -jar salotti-antognini-in.jar`”.