CS-430
Participants
Grades
General
16 September - 22 September
23 September - 29 September
30 September - 6 October
7 October - 13 October
14 October - 20 October
21 October - 27 October
28 October - 3 November

Intelligent agents

Programming: Implementing a first Application in RePast: A Rabbits Grass Simulation

This exercise is graded with 50 points and has to be done in a group of two persons (obligatory).

The goal of this exercise is to get you familiarized with RePast, a Java agent-based simulation platform. During the Intelligent Agent course, you will work on a platform for simulating the Pickup and Delivery Problem, written on top of RePast, which is to be introduced in the next exercise. It is therefore important that you understand the underlying mechanisms by designing and implementing a simulation yourself.

In this exercise, you should first go through the RePast documentation on their web page: http://repast.sourceforge.net/repast_3/index.html. Make sure that you work with Repast 3.1 (Repastj) and not Repast Symphony! We recommend you to then read the tutorial described by John T. Murphy from the University of Arizona: http://liapc3.epfl.ch/repast/main.htm and use this as a template for implementing the Rabbits Grass simulation.

Setting up a directory structure

First create the root directory, for example, CourseIntelligentAgents. Download rabbits.zip and unzip into this directory. It contains skeleton code, libraries, latex template for the documentation and a text file with instructions.

The Rabbits Grass simulation

The Rabbits Grass simulation is a simulation of an ecosystem: rabbits wander around randomly on a discrete grid environment on which grass is growing randomly. Rabbits have an initial amount of energy and with each move they lose a part of it. Once their energy is 0, they die. But when an alive rabbit bumps into some grass, it eats the grass and gains some energy. If a rabbit gains enough energy, it reproduces. The reproduction takes some energy so the rabbit can not reproduce twice within the same simulation step. The grass can be adjusted to grow at different rates and give the rabbits differing amounts of energy. It has to be possible to fully control the total amount of grass being grown at each simulation step. The model can be used to explore the competitive advantages of these variables.

This model has been described at http://ccl.northwestern.edu/netlogo/models/RabbitsGrassWeeds for the NetLogo simulation toolkit.

You have to program the Rabbits Grass Simulation in RePast, using the following requirements:

- **Grid:** the size of the world should be changeable. The default is a 20x20 grid. The world has no borders on the edges (thus, it is a torus).
- **Collisions** : different rabbits cannot stay on the same cell.
- **Legal moves:** only one-step moves to adjacent cells (north, south, east and west) are allowed.
- **Eat condition:** a rabbit can eat grass when it occupies the same cell.
- **Communication:** we assume that agents can not communicate with one another.
- **Visible range and directions:** all rabbits are blind and move randomly.
- **Creation**: at their births, rabbits are created at random places.

Implement sliders for the following variables of the simulation:

- **Grid size** of the world. We assume a square world of size (GridSize * GridSize)
- The **number** of rabbits defines the initial number of rabbits spread in the world
- The **number** of grass defines the initial number of grass spread in the world
- The **birth threshold** of rabbits defines the energy level at which the rabbit reproduces.
- The **grass growth rate** controls the rate at which grass grows (total amount of grass added to the whole world within one simulation tick). If it is set to 50, 50 units of grasses are spread to the world in total at every simulation tick and you should randomly put them into a number of cells.

You should **not** change the variable names of the above variables in the skeleton code we provide. You can add new variables if you want. Furthermore, remember to complete all "set" and "get" functions for all variables.

Report

- Explain your important model assumptions for the simulation your implement.
- Create a **population plot** to observe the evolution of the rabbits and the grass.

Running simulation

The main function accepts two arguments:

- args[0]: the parameter file to specify the variable values in the slider bar. By default, we set it to "" such that you can manually modify them in the GUI.
- args[1]: whether to use batch mode to run a simulation. By default, we set it to false to use the GUI mode.
- Remark: we only require you to implement the GUI model in this project. If you want to play with the parameter file and the batch model, feel free to so. In this case, you can use your IDE to pass these to arguments. For example, in Eclipse, you can edit them through "Run Configuration".

Before you submit your final version, please make sure that your runnable jar file can also be executed with passing the two arguments. for example, try the following command:
java -jar ***.jar "" false

Deliverable (Due on Tuesday 01.10.2019 at 23:55):
- Your source code, compiled code, runnable jar, and report. All in one (!) zip file
- The report is a short description of your code and results, must be in PDF, maximum of three pages
- Follow the instructions given in the materials you downloaded

UPLOAD THE zip FILE USING MOODLE.
NAME CONVENTION (IMPORTANT): *lastname1-lastname2*-in.zip/pdf

Submission status

Submission status	Submitted for grading
Grading status	Not graded
Due date	Tuesday, 1 October 2019, 23:55
Time remaining	Assignment was submitted 4 hours 21 mins early
Last modified	Tuesday, 1 October 2019, 19:33
File submissions	kimble-afresne-in.zip
Submission comments	Comments (0)

Feedback

Grade	40.00 / 50.00
Graded on	Saturday, 19 October 2019, 19:45
Graded by	Panayiotis Danassis

◄ Slides: Introduction to RePast and the first e: Jump to... Supplementary reading: Computational Rationality ►