Rabbit Grass Simulation Report

Junxiong Wang SCIPER 254478, Zixuan Liang SCIPER 255019

Code Descriptions

1. RabbitsGrassSimulationModel.java

main: creates a RabbitsGrassSimulationModel object and load the object.
begin: builds the model, schedule and display respectively. Then displays the dynamic grid and real-time statistic graphs.
getInitParam: gets initial values of parameters.
setup: disposes old grid and graphs, and creates new ones.
buildModel: builds a RabbitsGrassSimulationSpace object, and spread certain number of grass and adds rabbits in the space.
addNewAgent: adds a new rabbit to rabbit list and space.
buildSchedule: spreads some new grass, and then update every rabbit's status, and remove dead rabbits, and update the number of rabbits and grass in the graphs.
reapDeadAgents: removes dead rabbits whose energy is zero.
buildDisplay: sets colors of grass and grid and displays graphs.

2. RabbitsGrassSimulationSpace.java

spreadGrass: randomly spreads certain amount of grass in the grid.
isCellOccupied: checks whether there is a rabbit in one cell.
getEnergyAt: gets the amount of energy in one cell.
obtainEnergy: a rabbit eats grass and obtains the energy. Then the grass disappears.
moveAgentAt: move rabbits in the random direction.
getGrassNum: gets the amount of grass in the grid.

3. RabbitsGrassSimulationAgent.java

setVxVy: randomly sets the direction for the next move of a rabbit.
step: moves a rabbit in its direction and lets it obtain energy if move succeeds.

Code Running Procedure and Result

First it calls the "main" function to load model. After the user sets parameters, it spreads the amount of grass and add some rabbits randomly based on those parameters. For each step in the schedule, it spreads the certain amount of grass in the available cells. The rabbits move in random directions without collision. The program updates the status of each rabbit, including death or reproduction based on its energy. And the statistic graphs record the change of rabbit, grass and the total energy of this ecosystem in the entire phrase.

For illustration, we run the application, set parameters as figure 1. GrassEnergy: energy of one unit of grass; GrassGrowNum: grass growth rate; GrassNum: initial number of grass; GridX: X size of grid; GridY: Y size of grid; RabbitIntialEnergy: new rabbit's energy; RabbitNum: initial number of rabbits. Then start the simulation. The running program shows as figure 2. In the grid, black, green and white cells represent empty cells, grass cells and rabbits respectively. These three line charts demonstrate changes of number of grass, number of rabbits and total rabbit energy over time. The number of grass and rabbits fluctuate like this: first much grass grows in the space, providing enough energy for rabbits to reproduce. As the number of rabbits increases, more grass is eaten and its number drops. Then some of rabbits lack energy and die, providing more space for grass to growth. In this way, the "eco-system" maintains homeostasis.

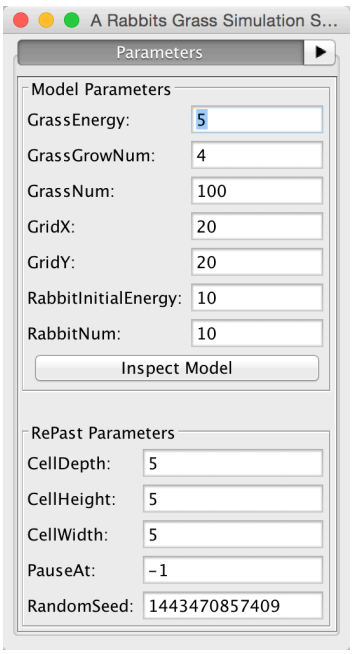

Figure 1

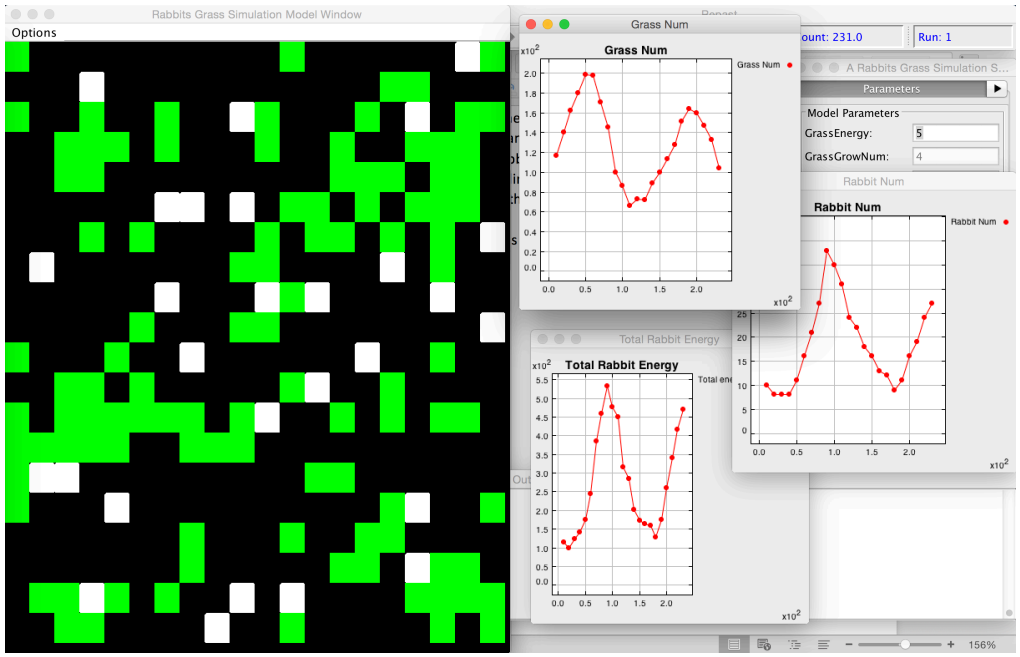

Figure 2

Next we change the initial parameters to somehow an extreme degree. For example, there are very few rabbits and very little grass as Figure 3. As Figure 4 shows, rabbits die out at the beginning because they consume too much energy in random movement before eating enough grass to survive. Then grass slowly occupies all the cells as it is no longer threaten by rabbits.

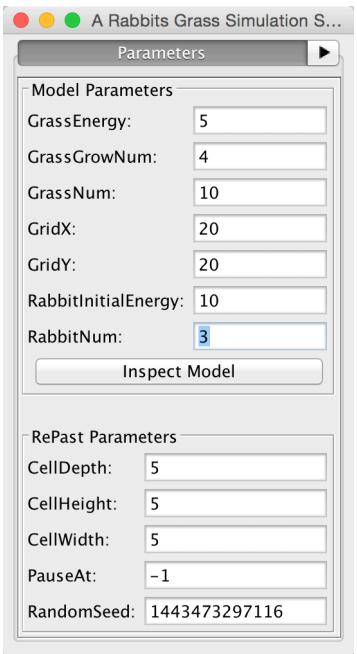

Figure 3

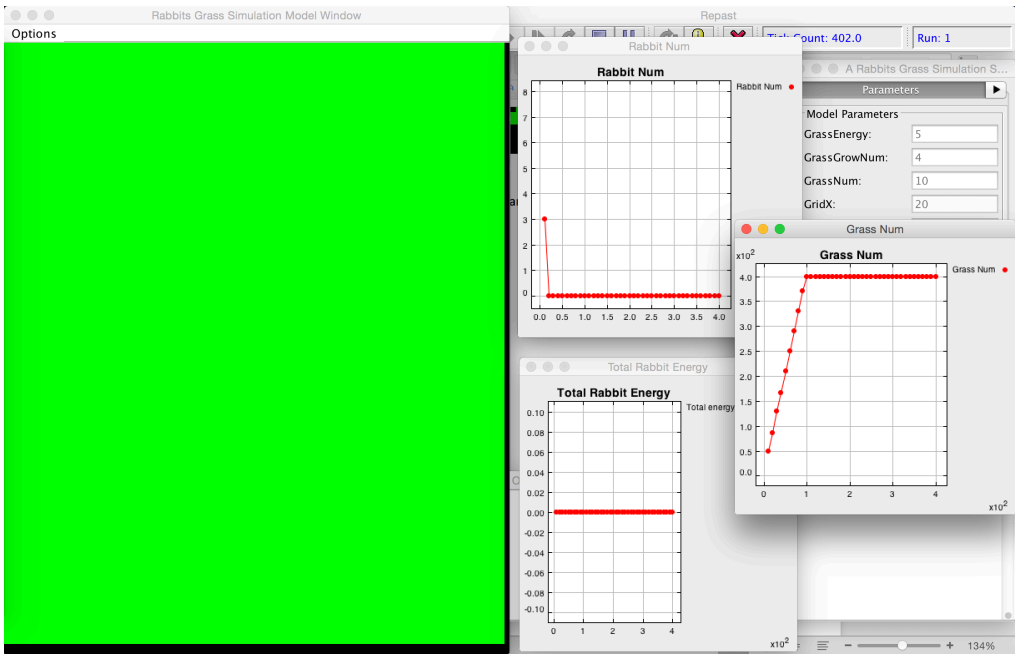

Figure 4