Spatio-Temporal Data in Society – Exercise 02

Christopher Stephan

November 27, 2013

Using the example programs provided, select some examples of spaceships, oscillators and methuselahs and provide the TerraME code.

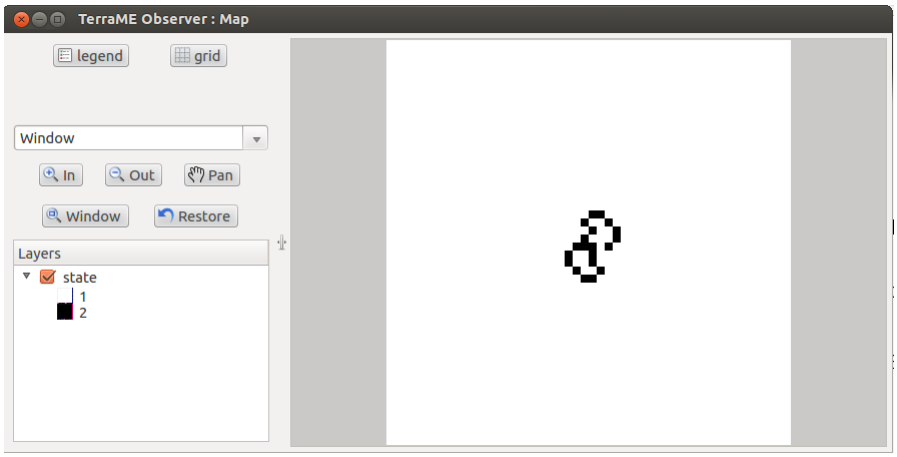
Additional patterns were added in the patterns.lua file: LWSS (spaceship), blinker (oscillator) and century (methuselah). The helper function insert_pattern was then used to show each pattern along a diagonal axis in the cellular space.

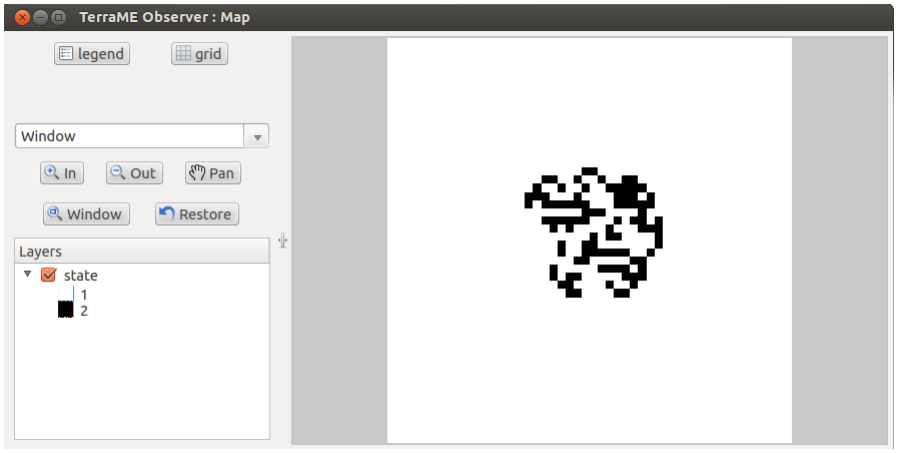
Please outline how you would design a TerraME program that could find all patterns of a certain type. For example, how would you design a program that finds all oscillators of period five (5)?

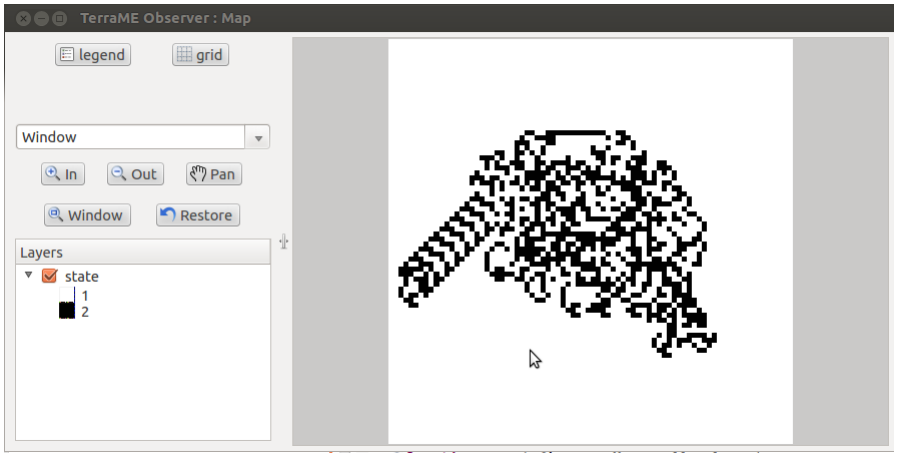
I would try to implement the concept of a listener function. This function has one parameter to pass the periodicity and a second one to pass a model that tries to generalize the characteristics of each type of pattern (i.e. spaceships or oscillators). The listener is attached to all cells in the cellular space. It records the cell's state at each step and compares the resulting pattern against the model.

Implement Langton's ant and verify its complex behavior.

The cellular space is initialized by setting all of its cells to the state "white". The initial starting position of the ant is defined in coordinate called "antPosition". The direction of the ant's movement is stored as global variables denoted with NORTH, SOUTH, EAST and WEST. The initial heading of the ant is set to NORTH. The behaviour of the ant's movement is modeled in the function called rules. This functions determines whether the current position of the ant is either a black or a white field. Depending on the field's color the direction of the ant's heading is determined and updated. After updating the heading, the ant moves one step forward in this direction. The following screenshots verify the complex behaviour at 100, 800 and 12000 steps.


TerraME Observer : Map
legend
grid
Window
In
Out
Pan
Window
Restore
Layers
state
1
2


TerraME Observer : Map
legend
grid
Window
In
Out
Pan
Window
Restore
Layers
state
1
2


TerraME Observer : Map
legend
grid
Window
In
Out
Pan
Window
Restore
Layers
state
1
2

Implement Brian's Brain in TerraME and see if you can reproduce some of these effects.

The cellular space is initialized by using the math library to pseudo-randomly choose a value between 1 and 3. These values indicate if a cell's state is ON, OFF or DYING. Within the rules function maximally three if-statements are checked to set the state of a cell in each step. A helper function, called hasTwoNeighborsOn(cell), is used that returns a boolean value. Effects like diagonal waves are shown in the following screenshot. The ON cells are white and the DYING cells are red.