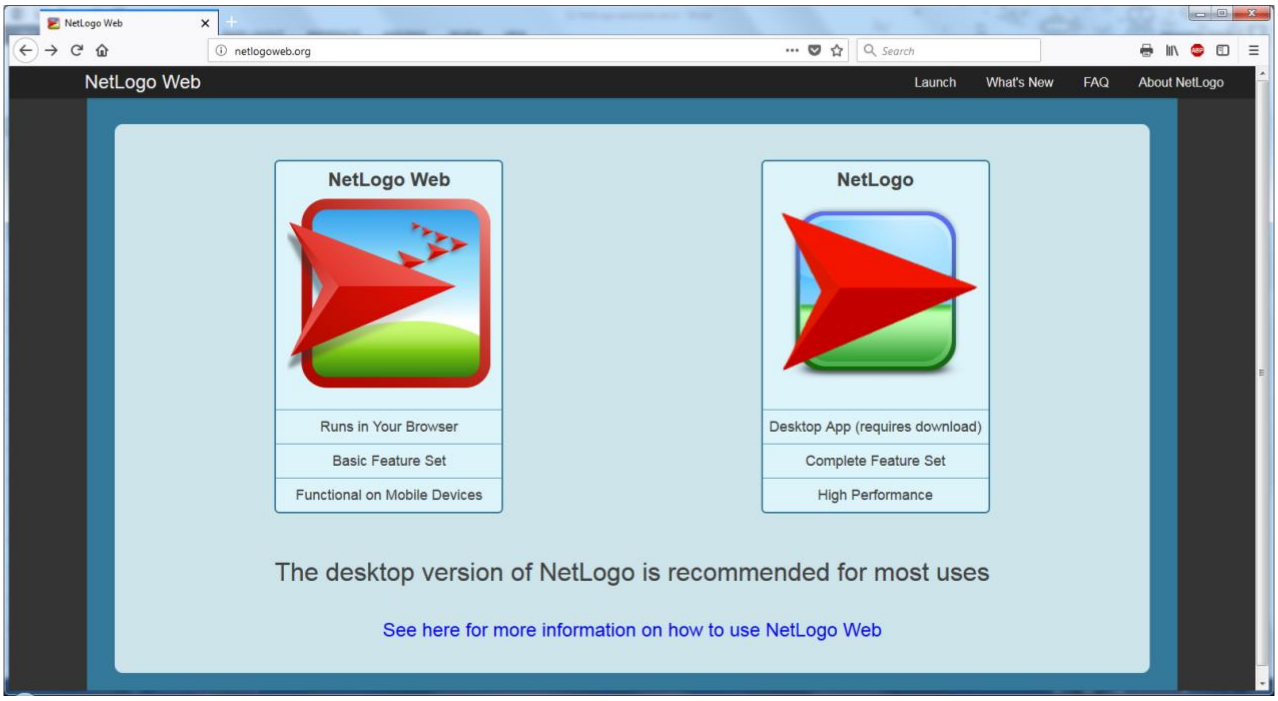
Athanasios Votsis
(athanasios.votsis@helsinki.fi)

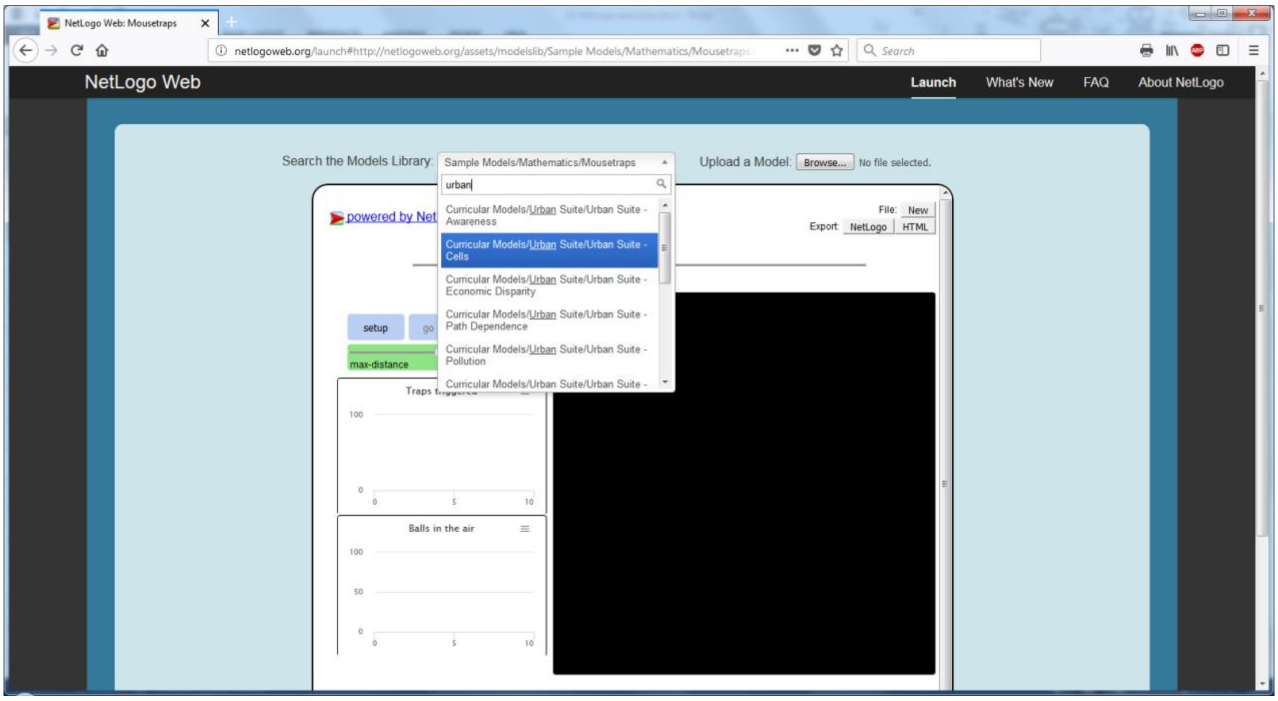
Getting familiar with NetLogo and urban-themed cellular automata

1 – Producing forms consistent with observed urban morphologies

1.1 Go to http://netlogoweb.org/ and chose the "NetLogo Web" version that runs straight from your browser:

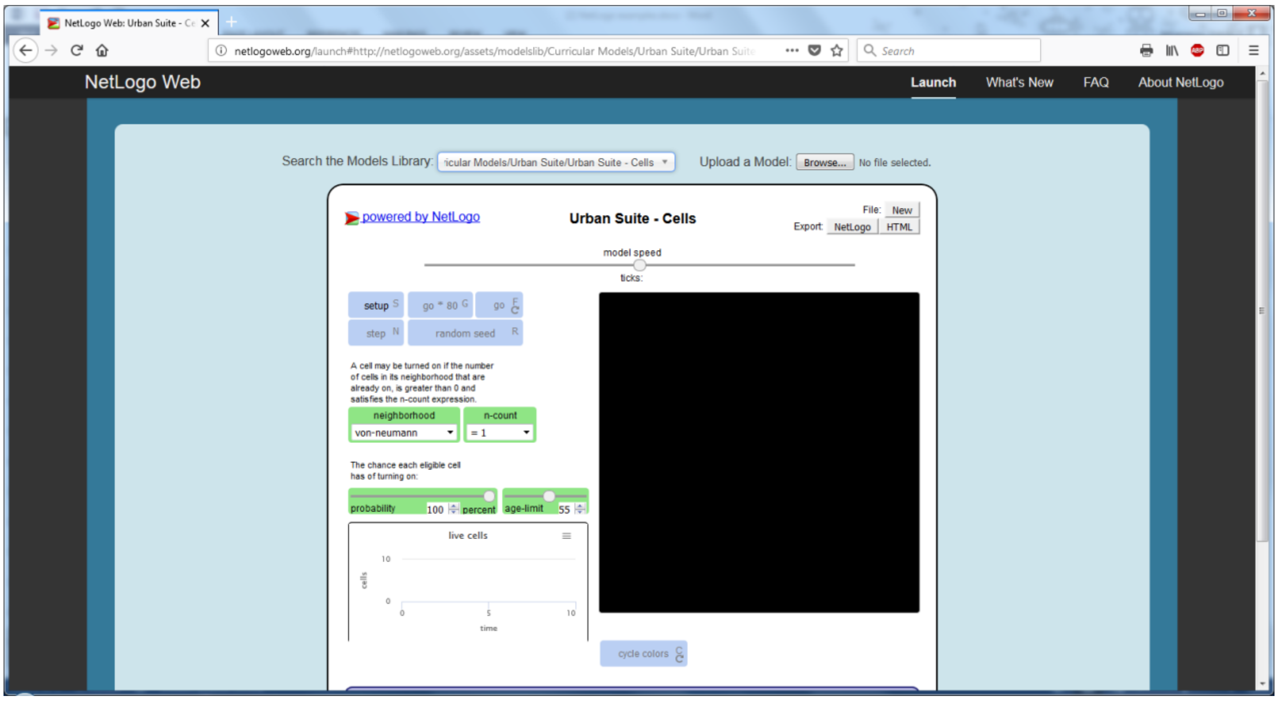

NetLogo Web
netlogoweb.org
Search
NetLogo Web
Launch
What's New
FAQ
About NetLogo
NetLogo Web
Runs in Your Browser
Basic Feature Set
Functional on Mobile Devices
NetLogo
Desktop App (requires download)
Complete Feature Set
High Performance
The desktop version of NetLogo is recommended for most uses
See here for more information on how to use NetLogo Web

1.2 From the `Select Models` drop-down list, select the `Curricular Models/Urban Suite/Urban Suite – Cells option`:


NetLogo Web: Mousetraps
netlogoweb.org/launch#http://netlogoweb.org/assets/modelslib/Sample Models/Mathematics/Mousetraps
NetLogo Web
Launch
What's New
FAQ
About NetLogo
Search the Models Library:
Sample Models/Mathematics/Mousetraps
urban
Curricular Models/Urban Suite/Urban Suite - Awareness
Curricular Models/Urban Suite/Urban Suite - Cells
Curricular Models/Urban Suite/Urban Suite - Economic Disparity
Curricular Models/Urban Suite/Urban Suite - Path Dependence
Curricular Models/Urban Suite/Urban Suite - Pollution
Curricular Models/Urban Suite/Urban Suite -
Upload a Model:
Browse...
No file selected.
File:
New
Export:
NetLogo
HTML
powered by Net
setup
go
max-distance
Traps triggered
Balls in the air

Athanasios Votsis
(athanasios.votsis@helsinki.fi)

1.3 The `Urban Suite – Cells` model is a simple CA model that lets you explore what happens to a generic growth process (more accurately: growth by spreading) when you apply different neighborhood rules. When you load the model, it looks like the following:


NetLogo Web: Urban Suite - Cells

netlogoweb.org/launch#http://netlogoweb.org/assets/modelslib/Curricular Models/Urban Suite/Urban Suite

Search the Models Library: icular Models/Urban Suite/Urban Suite - Cells

Upload a Model: Browse... No file selected.

Urban Suite - Cells

A cell may be turned on if the number of cells in its neighborhood that are already on, is greater than 0 and satisfies the n-count expression.

neighborhood: von-neumann

n-count: = 1

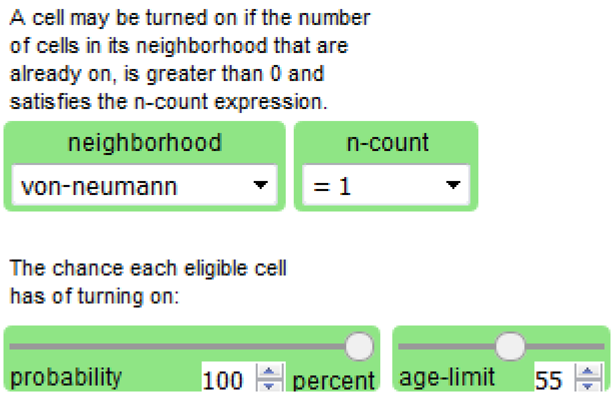
The chance each eligible cell has of turning on:

probability 100 percent

age-limit 55

1.4 For now, the most important components you should be aware of are the following:

- The settings controls, which let you set the parameters that control the behavior of this particular CA model:


A cell may be turned on if the number of cells in its neighborhood that are already on, is greater than 0 and satisfies the n-count expression.

neighborhood: von-neumann

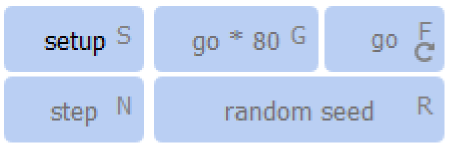
n-count: = 1

The chance each eligible cell has of turning on:

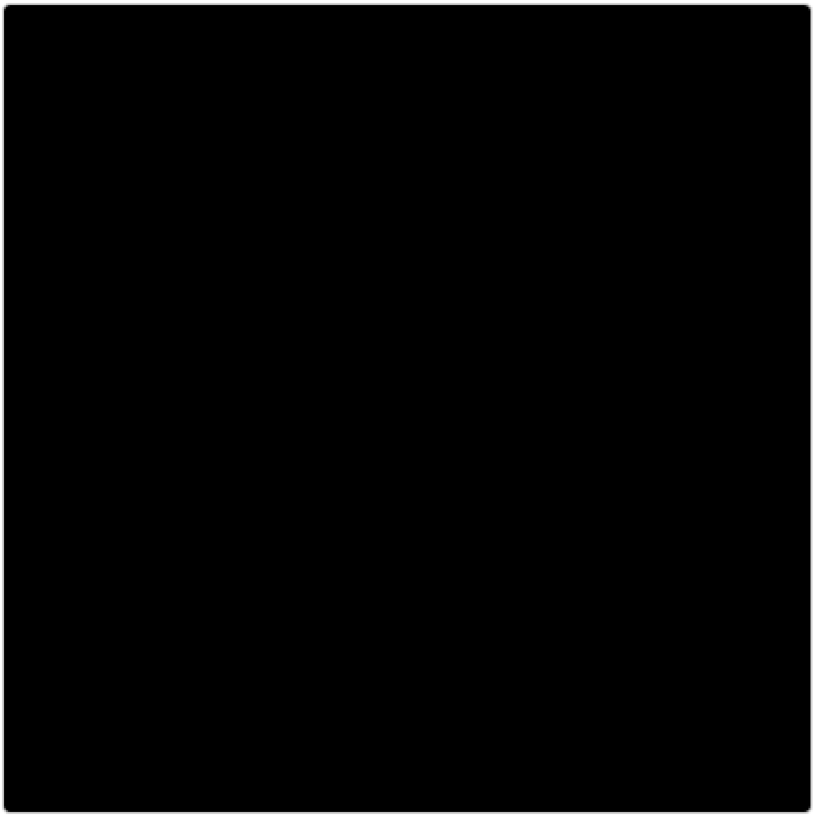
probability 100 percent

age-limit 55

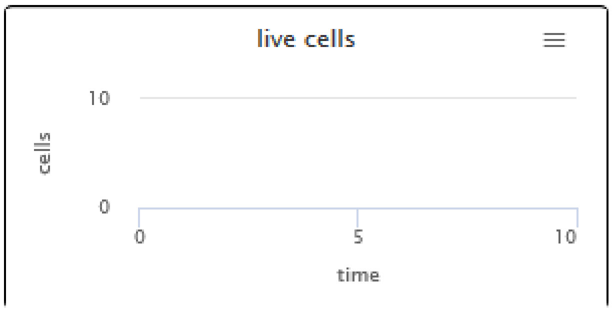
- The initialization and runtime controls. The `setup` button initializes the model with the parameters you have defined in the settings. The `go` button starts running the model indefinitely, the `go*80` button runs 40 time steps (the "80" is a typographical error), and the `step` button runs just 1 time step at a time:



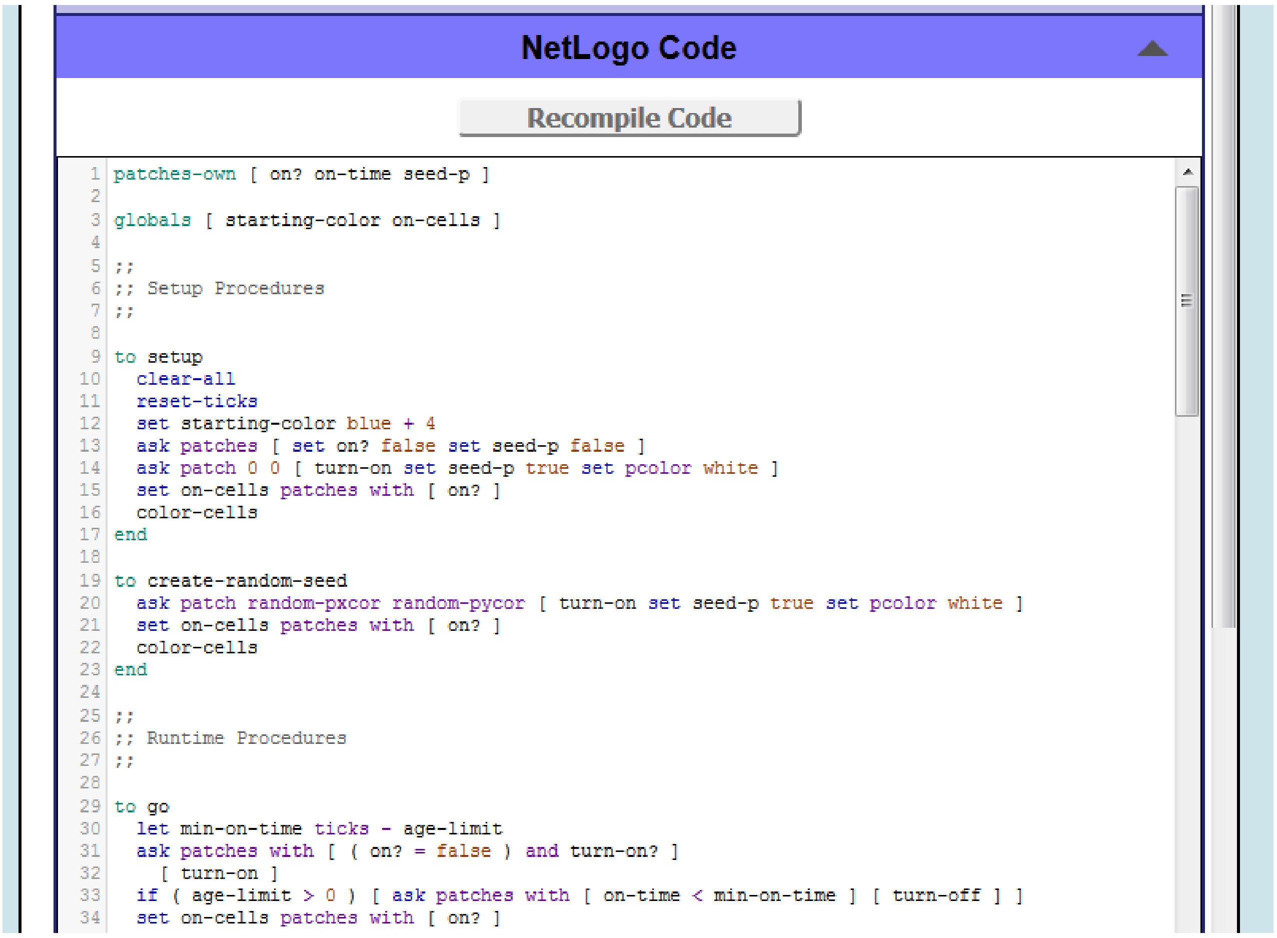
- The spatial output of the CA growth process:



- And lastly, a graph of how many cells have been urbanized over time:

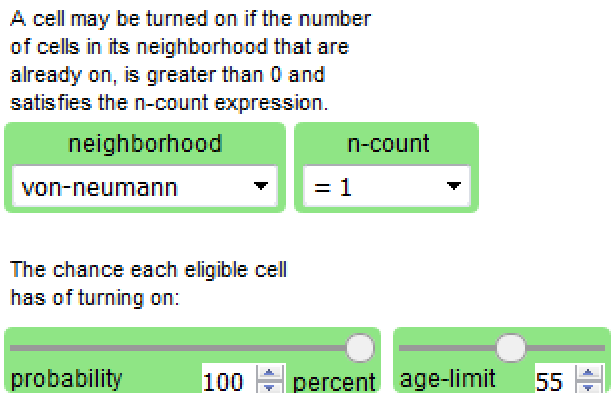


1.5 All the above are completely programmable. If you click on the `NetLogo Code` tab below the abovementioned visual elements, you will see the code that has created them, which is written in the NetLogo language:



```
patches-own [ on? on-time seed-p ]

globals [ starting-color on-cells ]

;;
;; Setup Procedures
;;

to setup
  clear-all
  reset-ticks
  set starting-color blue + 4
  ask patches [ set on? false set seed-p false ]
  ask patch 0 0 [ turn-on set seed-p true set pcolor white ]
  set on-cells patches with [ on? ]
  color-cells
end

to create-random-seed
  ask patch random-pxcor random-pycor [ turn-on set seed-p true set pcolor white ]
  set on-cells patches with [ on? ]
  color-cells
end

;;
;; Runtime Procedures
;;

to go
  let min-on-time ticks - age-limit
  ask patches with [ ( on? = false ) and turn-on? ]
    [ turn-on ]
  if ( age-limit > 0 ) [ ask patches with [ on-time < min-on-time ] [ turn-off ] ]
  set on-cells patches with [ on? ]
```

1.6 Let's initialize and run the model with the default parameters:



A cell may be turned on if the number of cells in its neighborhood that are already on, is greater than 0 and satisfies the n-count expression.

neighborhood: von-neumann

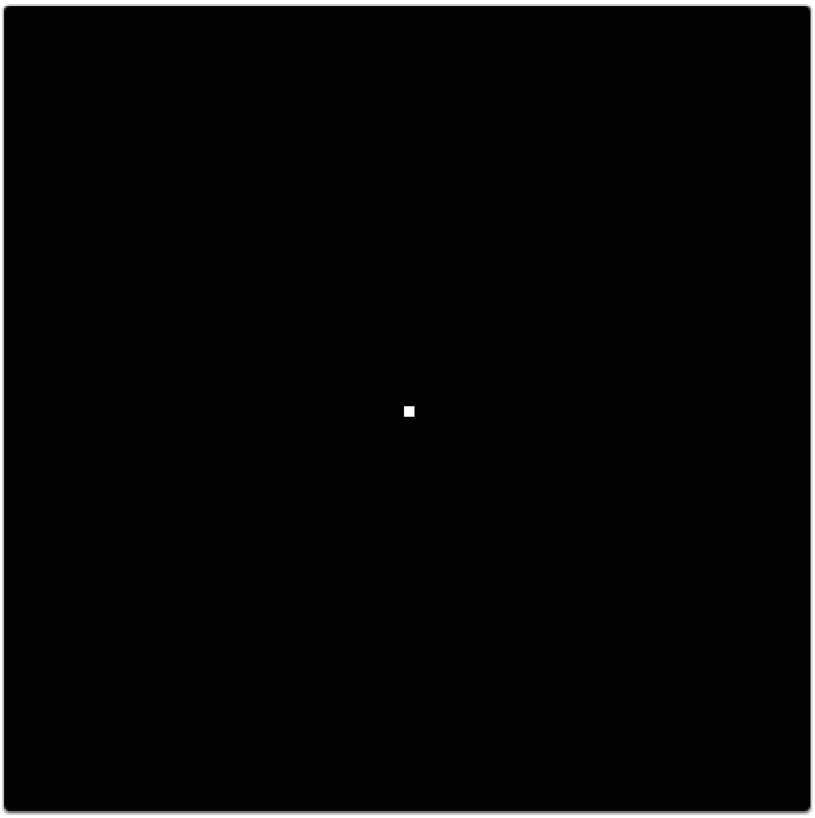
n-count: = 1

The chance each eligible cell has of turning on:

probability 100 percent

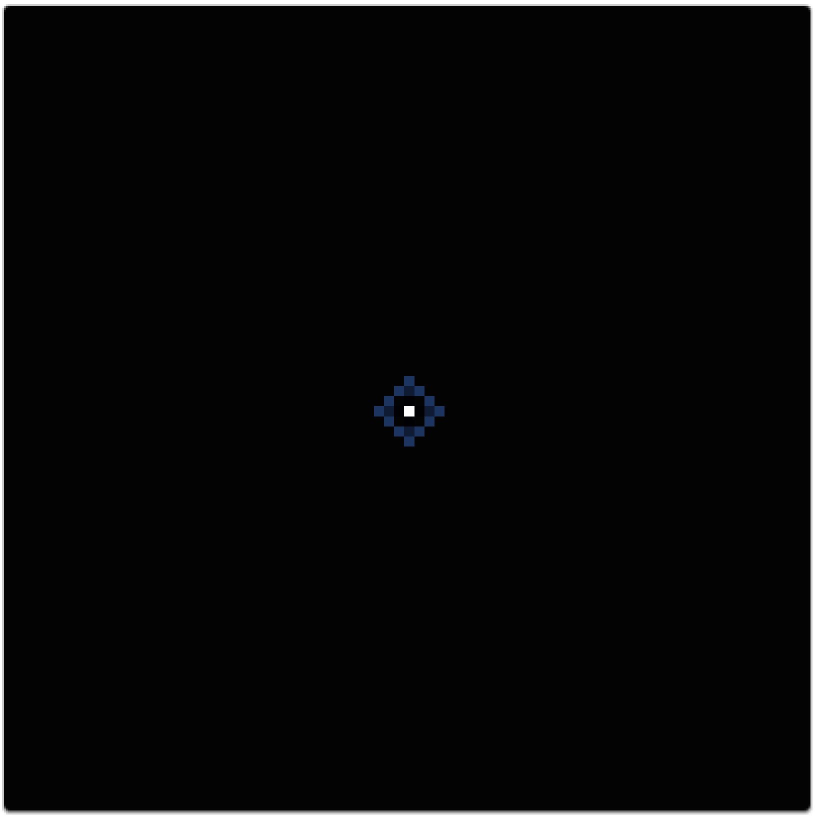
age-limit 55

1.7 After you press `setup` to initialize the model, you will see one white pixel (= "urbanized" or "developed") at the center; in modeler's jargon this cell is often called the "seed". Any idea why?

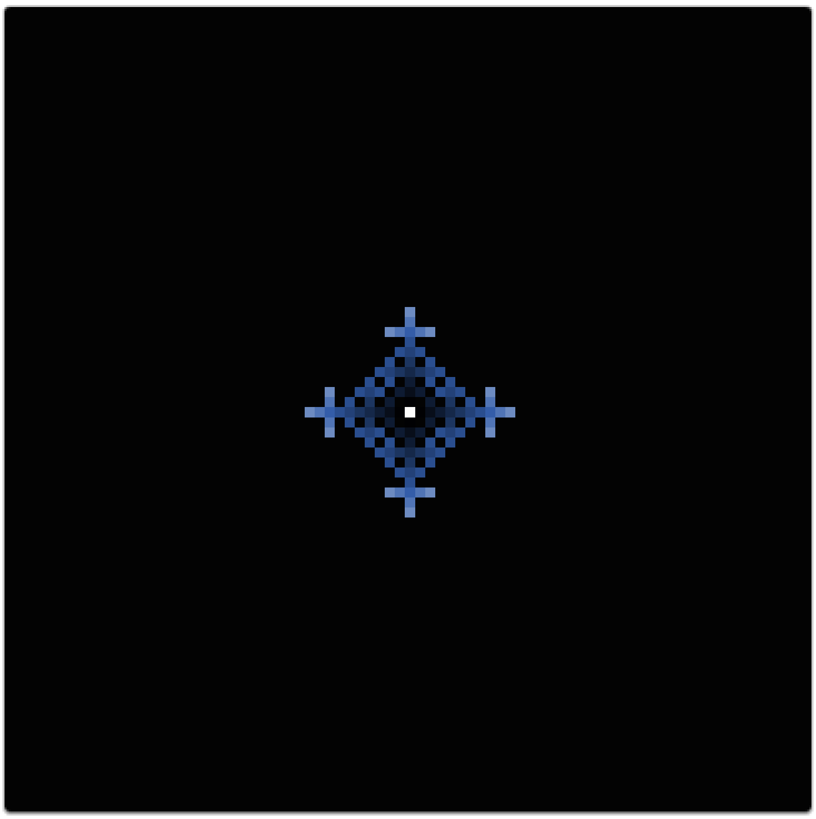


1.8 Now press `step` several times to see how the growth process evolves:

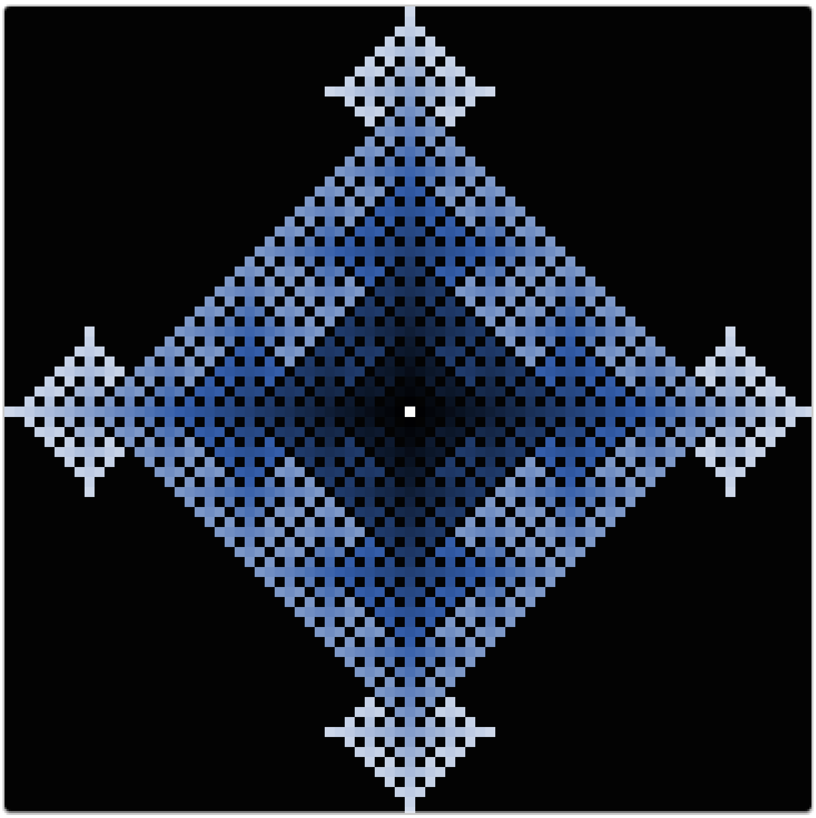
After 3 time steps:



After 10 time steps:

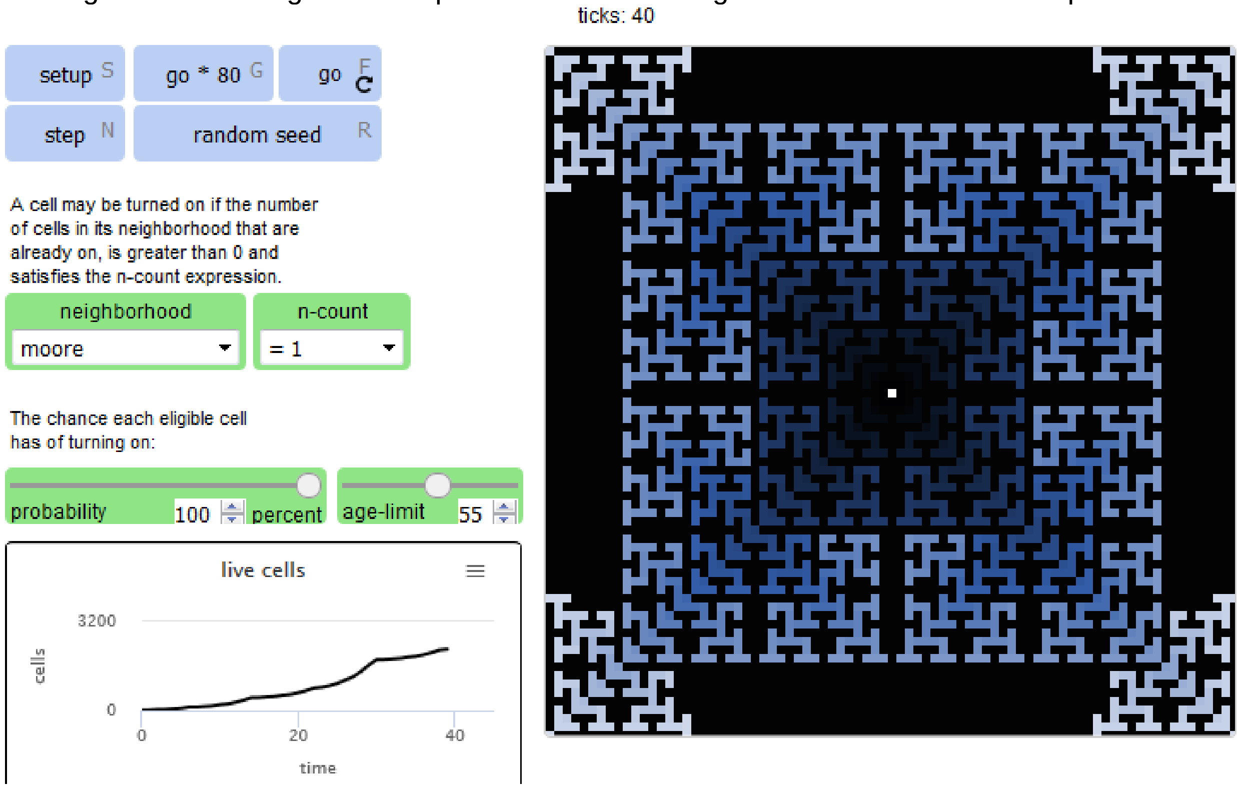


After 40 time steps:



Athanasios Votsis
(athanasios.votsis@helsinki.fi)

1.9 Using the Moore neighborhood produces the following outcome after 40 time steps:



A cell may be turned on if the number of cells in its neighborhood that are already on, is greater than 0 and satisfies the n-count expression.

neighborhood: moore

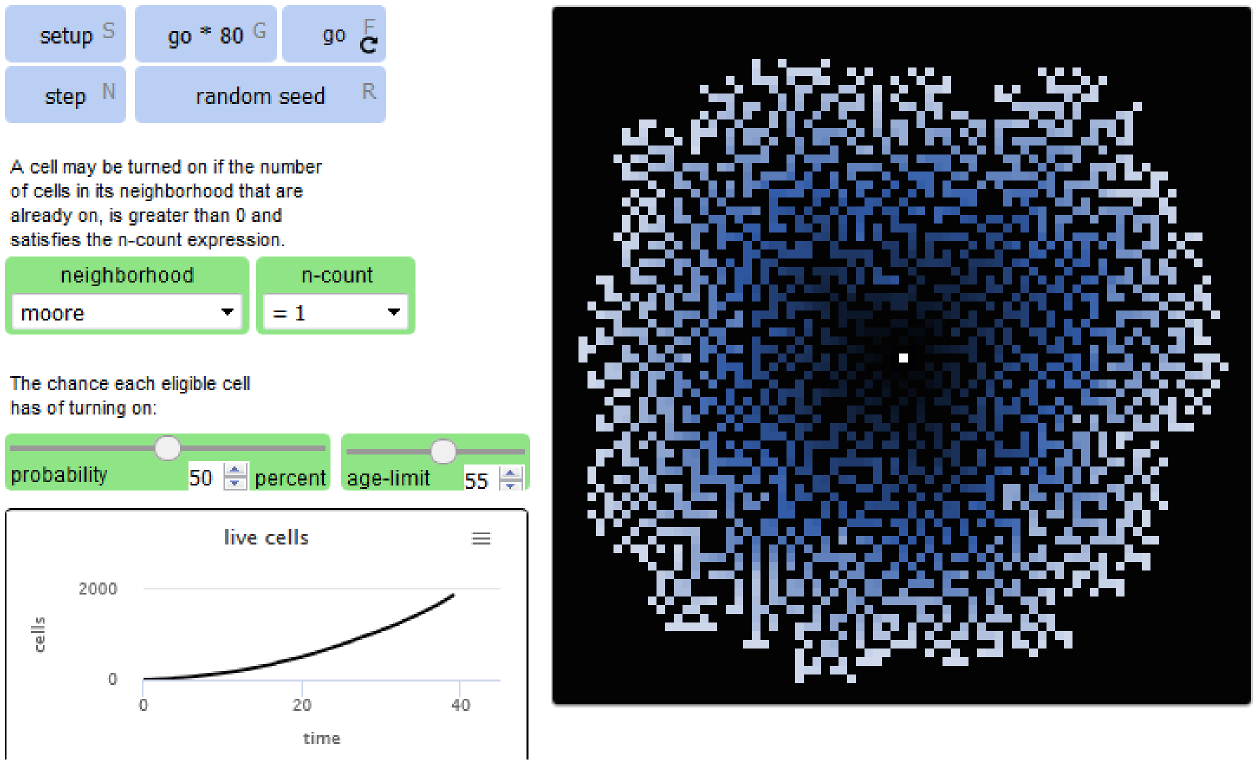
n-count: = 1

The chance each eligible cell has of turning on:

probability 100 percent

age-limit 55

1.10 And using again the Moore neighborhood, but now with only 50% of the cells that fulfil the condition for state transition actually making the transition (i.e. “probability” = 50%), produces the following outcome after 40 time steps:



A cell may be turned on if the number of cells in its neighborhood that are already on, is greater than 0 and satisfies the n-count expression.

neighborhood: moore

n-count: = 1

The chance each eligible cell has of turning on:

probability 50 percent

age-limit 55

Athanasios Votsis
(athanasios.votsis@helsinki.fi)

2 – Producing “urban morphologies” consistent with observed urban morphologies

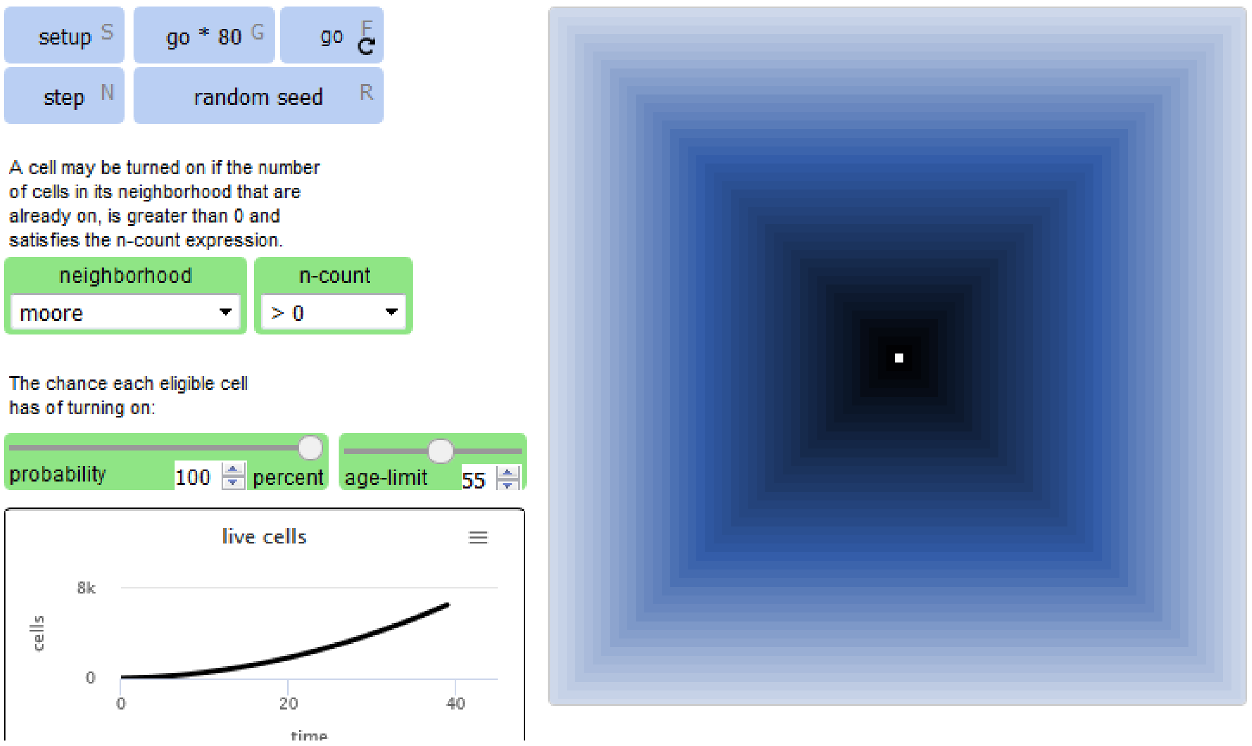
2.1 The three outcomes that we have produced above in steps 1.8 - 1.10 replicate Batty’s (2007) demonstration that we can start thinking about cities or urban growth processes in terms of two fundamental elements:

1. simple neighborhood rules
2. growth constraints

2.2 While neighborhood rules can have an impact, in the end it does not matter that much which particular version is active – almost just like we saw in spatial statistics. The reason is that one way or another a growth process will affect all of the system (here: all of the cells) over time, sooner or later.

2.3 The defining feature is *putting constraints on growth*. This means that it is not enough that you establish neighborhood rules so that a growth process can spread across geographical space. You also need to limit how this happens, much like in the real world where there are constraints like, for instance, limited resources or uneven competition.

In our case this has been achieved by telling the cells to switch state not if *any* cell in the defined neighborhood is “developed”, but if *exactly one* cell in the defined neighborhood is developed. To see the difference use the following settings and compare the results with step 1.9:



A cell may be turned on if the number of cells in its neighborhood that are already on, is greater than 0 and satisfies the n-count expression.

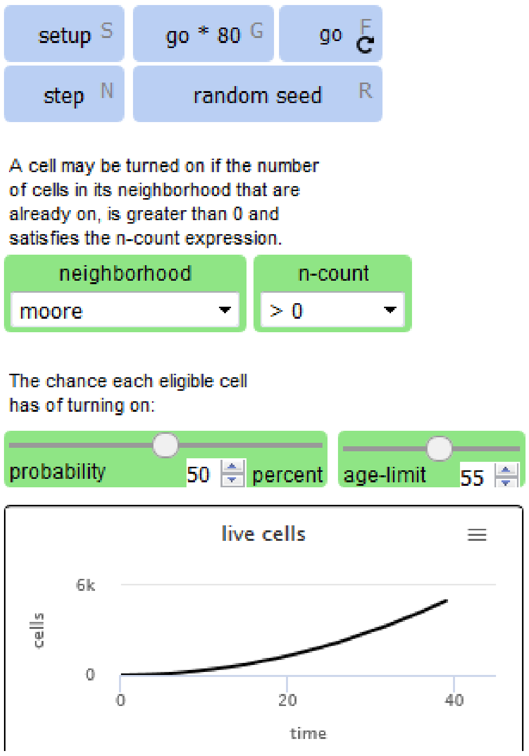
neighborhood: moore
n-count: > 0

The chance each eligible cell has of turning on:

probability 100 percent
age-limit 55

You’ll see that within 40 time steps simply all of the cells have switched state and for this reason no morphology exists. The colors just denote the time step in which of a cell has switched state.

2.4 Now add to the settings of step 2.3 above a constraint: a 50% probability of development, re-initialize and re-run. The outcome after 40 time steps will be as follows:



A cell may be turned on if the number of cells in its neighborhood that are already on, is greater than 0 and satisfies the n-count expression.

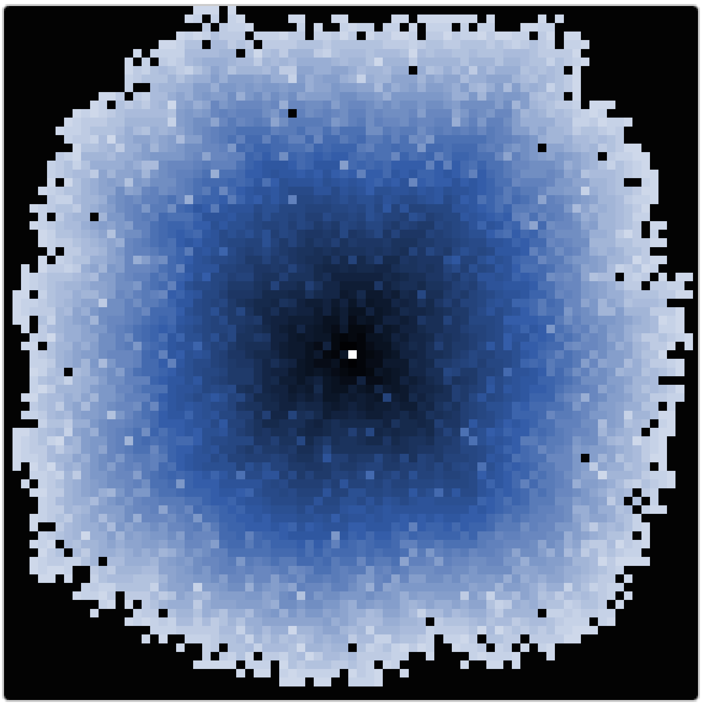
neighborhood: moore

n-count: > 0

The chance each eligible cell has of turning on:

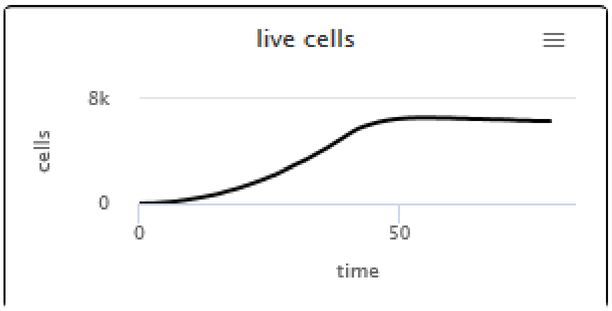
probability 50 percent

age-limit 55



Structure that resembles some observed urban morphology appears, and you can even discern familiar linear features in the older parts of the system.

2.5 Lastly, press one more time the `go*80` button and observe the corresponding growth curve:



More specifically, S-shaped growth curves are typical of constrained growth processes. At the early stages of growth, when there are a lot of resources to "be consumed", growth follows an exponential (or other similar) form with high growth rates (time = 0 to approx. 40 in the above graph). As resources –in this case available cells– deplete, growth rate slows down and the growth curve flattens out.