

# **AUTOMATA for ART 3.1**

**How to register?**

**Extended version**

**Applications**

**Mouse use**

**Keyboard shortcuts**

**Commands**

**What is new in the current version?**

**Disclaimer of warranty**

**Support**

**Configuration**

**Installation**

**Files**

All marks mentioned in this documentation are trademarks.  
This documentation is copyright (c) 1992-1998 Paul FRANCESCHI.

# COMMANDS

The *Automata for Art* software contains the features listed below:

## File

New

Save initialisation file

Quit

## Edit

Copy status bar informations

Copy bitmap window

Copy mosaïc window

Copy array window

Save bitmap as

Save array as

Save mosaïc as

## Display

Increment cell dimensions

Decrement cell dimensions

## Initialization

Null

Random

Special patterns

Array fill

## Selection

Save

Load

Copy

Paste

0-State

Horizontal symmetry

Vertical symmetry

## Options

Step

Array

Array32x32

Array16x16

Mosaïc

Automata

Random

Colors

Language

Transition rules

Neighborhood parameters

Preferences

Statistics

## Process

Next

**Level**  
**Continuous**  
**Continuous until**

**Window**  
**Informations**  
**Array window**  
**Mosaic**  
**Multi-bitmap**

**Help**  
**Index**  
**Registration code**  
**About**

## HOW TO REGISTER?

This section only concerns the limited shareware version:

The *Automata for Art 3.1* application is a shareware program, and is provided at no charge to the user for evaluation. Feel free to share it with the others, but do not give it away altered or as part of another system.

If you find this program useful, and continue to use the *Automata for Art* software, after a reasonable trial period, you must make a registration payment of \$19. You will receive a personal registration code, giving you access to the version ***Automata for Art 3.1 extended***.

How to register?

Please first execute the *Register* application (REGISTER.EXE) included in the package. Fill in the form and print it. Then send it to Kagi, with your payment of \$19, at the following address:

Mail: **Kagi**  
**1442-A Walnut Street #392-NIF**  
**Berkeley, CA 94709-1405**  
**USA**  
Fax: **+1 510 652 6589**  
Email: **kagi@kagi.com**

Kagi will inform me by email of your registration and I will send you (by email, fax or classic mail) your personal registration code, which will give you access to the **extended version** of *Automata for Art 3.1*.

-----  
The registration will license one copy for use on any one computer at any one time. Commercial users of the *Automata for Art* software must register and pay for their copies within 30 days of first use. Site-License arrangements may be made.

Commercial users of *Automata for Art* must register and pay their copies within 30 days of first use or their license is withdrawn.

## EXTENDED VERSION

The advantages of registering your version, are the following:

1. The **extended** version contains the following additional functionalities:

- option **Options/Transition rules**: this option gives access to  $16^9$  (68719476736) cellular automata

- option **Options/Neighborhood parameters**: for each cellular automaton, one can modify the 9 neighborhood parameters, attributing them specific values, giving hereby access to  $16^{18}$  cases (4722366482869645213696) of cellular automata

2. The license of the software is attributed to you. The software license concerns a single copy, to use with a single computer.

3. You can directly contact me directly. Any comments or suggestions are welcome. Please specify your email address, if you have one.

My email address is the following: [Paul.Franceschi@wanadoo.fr](mailto:Paul.Franceschi@wanadoo.fr)

## APPLICATIONS

The *Automata for Art* application is a **graphical software**, allowing the creation of bitmaps, using the graphical properties of cellular automata.

You can select pre-defined automata with the **Options/Automata**, whose graphical properties are already remarkable, or define your own automata, using the **Options/Parameters**, which allows you to modify the parameters of the current automaton.

You may also change the colors corresponding to each state (from 0 to 15); you will use the **Options/Colors** option in the main menu, and access to the default colors modification. This possibility is very important for graphical creation, and you may define your own colors, corresponding to your particular scopes. Another available and interesting mode, is to modify the colors, several times, when using a specified automaton. Colors can be changed at any time, according to the effects desired.

When you are satisfied of your bitmap, you can transfer and export it toward another graphical software, or another program. The bitmaps created with the *Automata for Art* application are bitmap format (.BMP), and are compatible with other softwares. Simply copy your bitmap in the clipboard, using the **Edit/Copy bitmap** option, and transfer the bitmap in the other application, with the **Edit/Paste** or equivalent command of the destination software.

## DISCLAIMER OF WARRANTY

*Automata for Art* AND ITS DOCUMENTATION ARE SOLD "AS IS", AND WITHOUT WARRANTIES AS TO PERFORMANCE OF MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED, BECAUSE OF THE VARIOUS HARDWARE AND SOFTWARE ENVIRONMENTS INTO WHICH THIS PROGRAM MAY BE PUT, NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED. ANY LIABILITY OF THE SELLER WILL BE LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT OR REFUND OF PURCHASE PRICE. THE AUTHOR ASSUMES NO LIABILITY FOR DAMAGES, DIRECT OR CONSEQUENTIAL, WHICH MAY RESULT FROM THE USE OF *Automata for Art*.

## SUPPORT

You can contact me at

Paul FRANCESCHI  
Résidence la Pietrina  
Avenue de la Grande Armée  
20000 Ajaccio  
FRANCE

or by email

my email address is: [Paul.Franceschi@wanadoo.fr](mailto:Paul.Franceschi@wanadoo.fr)

A mail or email support is provided during a period of 3 months, from the date of registration. If you have an electronic adress, I can provide you assistance by email.

Elsewhere, your suggestions, remarks and propositions are greatly encouraged and accepted. Any suggestion concerning new options or fonctionnalités are welcome. Your feedback, if your are a professionnall user, is important, too.

In order to make support faster, please specify, if possible, your email address.



## WHAT IS NEW IN THE CURRENT VERSION?

### In version 3.1:

This version has been entirely rewritten, for Windows 95 (tm), and includes many changes:

- the interface has been modified
- a status bar replaces the Information window
- an icon bar has been added, including the most common functionalities
- the following functionalities have been added:
  - Display / Increment cell dimensions
  - Display / Decrement cell dimensions
  - Edition / Save bitmap window
  - Edition / Save array window
  - Edition / Save mosaic window
  - Options / Array 16x16
  - Options / Array 32x32
- the extended version is now directly accessible with the help of an individual registration code

### In version 3.0:

- this version is a 32 version, specifically implemented for Windows 95 (TM).
- the presentation of the dialog boxes and windows is improved
- the presentation of the help file is improved
- the software installation is now simplified and improved

### In version 2.2:

- this new version is entirely bilingual: *english* and *french*. Simply commute from english to french using **Option/Language** allowing selection between the english and french version.

### In version 2.1:

- this section is now included
- the windows system is modified. The ergonomoy of the **Bitmap**, **Informations**, **Array**, **Multi-Bitmap**, and **Mosaic** windows is improved. The **Window/Zoom** is no longer implemented, because the **Window/Array** does the same task. An option **Window/Initialize** is added.
- you are asked to mention your eventual electronic adress. If possible, I can send you the extended version by email, as an attached file.
- my email adress is now included in the software. This permits fastest assistance, by electronic mail. It is also more reliable, if you want to adress your suggestions.

## CONFIGURATION

The system requirements for *Automata for Art 3.1* are the following,:

- an IBM-PC compatible computer: at least 80486
- WINDOWS 95 (TM) or further
- 2 megas RAM
- a mouse
- a color monitor

## INSTALLATION

Microsoft Windows 95 (TM) or further must be installed and working, before you install *Automata for Art*. To install *Automata for Art*, simply copy all files provided with the *Automata for Art* application in a directory of your hard drive. For example, you can create a directory named AFA and copy all the files in it, using the following command: `COPY *.* C:\AFA\`  
The executable file is: AFA.EXE.

## FILES

*Automata for Art* includes the following files:

- AFA.EXE: the executable file
- ARTUS.HLP: the english help file
- ARTFR.HLP: the french help file
- FILE\_ID.DIZ: a short text file describing the *Automata for Art* software
- REGISTER.EXE: the program to use for registration
- REGISTER.HLP: the help file of the program to use for registration

## MOUSE

**Left button:** with the left button, in the **Array** window, you can select a *field*, to use with the **Selection** options (**Copy**, **0-state**, **Increment**, **Decrement**, etc.).

## KEYBOARD SHORTCUTS

Several keyboard shortcuts are available, in order to optimize the options access for the experimented user:

Ctrl-N **File/New**  
Ctrl-Q **File/Quit**  
Ctrl-R **Initialization/At random**  
Ctrl-P **Initialization/Point**  
Ctrl-S **Execution/Next**  
Ctrl-C **Execution/Continuous**

## **NEW**

Clears the whole array elements. Cells are initialized with the standard initial configuration of the Life Game. All options and preferences previously defined are cancelled. The options, preferences, and parameters of the currently saved configuration (included in AFA31.INI file) are active.

## SAVE INITIALIZATION FILE

This options saves the configuration file, including the options and preferences declared. At next use of the software, the options and preferences previously saved, will be automatically loaded. A file called **AFA31.INI** is automatically written in the C:\WINDOWS repertory, including all selected preferences and parameters.



## ABOUT

Shows Copyright informations about *Automata for Art*. The version number of the software also appears on this box.

## QUIT

Quits *Automata for Art*, and returns to WINDOWS (TM) interface.

## **COPY STATUS BAR**

Copy the informations included in the status bar. These informations (automaton number, current level, etc.), are copied in the clipboard. This option is useful to note all interesting informations about a specific automaton, in order to re-use it. All the informations copied, can be transfered to another WINDOWS(TM) application, such as a word processor.

## COPY BITMAP

Copy the current **Bitmap** window to the clipboard, in order to transfer it to another WINDOWS application, such as a graphical software, or a word processor. The current bitmap is exported, via the clipboard to other softwares. This option is fundamental for professional use of the *Automata for Art* software. Sophisticated, esthetical bitmaps can be created with specific automata, and exported to other applications.

If you use the *Automata for Art* software as an *icon drawer*, you may choose a 32x32 array, and transfer the resulting bitmap with the **Edit/Copy bitmap** option, to another application.

## COPY MOSAIC

Using this option, you copy the bitmap resulting of the **Mosaic** window option in the clipboard. The background window is created by the juxtaposition of the current bitmap, representing the current state of the automaton. The background window can be compared to the **Mosaic** option of the WINDOWS(TM) interface.

## COPY ARRAY IMAGE

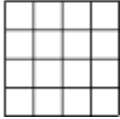
Copy the contents of the current array. The image of the **Array** window is transferred to the clipboard.

## NULL

Initializes the array with no elements. All cells are set to 0-state. This option may be used to configure the initial array, as desired. You can add cells using filed selection, and **Increment cell state**, or **Decrement cell state**.

If some preferences or parameters have been defined, they are still active.

The following 4x4 array, is initialized with the **NULL** command:



## RANDOM

Initializes the array with random elements. Using this option, initialization is always different.

If some preferences or parameters have been defined, they are still active.

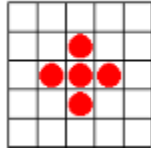
The count of elements randomly determined for each specific state are defined with the **Options/Random** option.



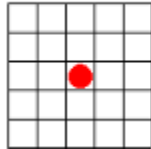
## SPECIAL PATTERNS

Initial figures are very important, when you use a specific automaton; the result mainly depends on the initial configuration. The following special patterns, for initialisation of a selected automaton are available:

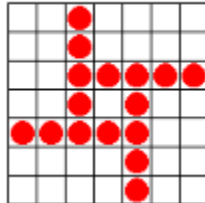
- cross



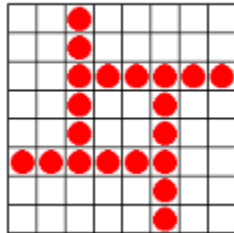
- point (central point)



- screw (7x7)



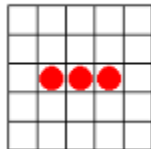
- screw (8x8)



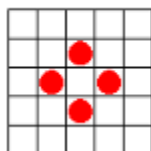
- square



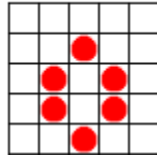
- line (3 aligned and central points)



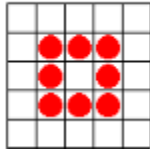
- rhomb



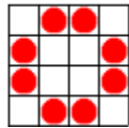
- hexagon



- square



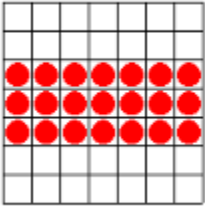
- octagon



All the initial figures central and constituted of state-1 (red, by default) points.

## ARRAY FILL

Fills the current array with a central rectangle (1-state cells) whose coordinates are parametrable.  
For example, the following array results from the use of the **Array Fill** option, 7x3 parameters whose width and height are respectively 7 and 3:



## STEP

Defines the step selected, for each iteration.  
At loading time, this value is set to 1 (single step).

A message error is issued if the step, added to current iteration level, is greater than the **Iteration limit** (500 by default).

## **SELECTION / SAVE**

Saves the selected field in a file in .TXT format.

The .TXT format can be directly read by a word processor, or a spreadsheet. The state cell are numbers from 0 to 15.

This option may be useful, in order to compute sophisticated statistics concerning a specific array, within - for example - a spreadsheet.

## SELECTION / LOAD

Load a configuration into the destination area. The load file must be written in .TXT format. The .TXT format can be directly created by a word processor, or a spreadsheet.

This property allows the computation of a specific array with a spreadsheet, and its capture in the Automata software, using the **Selection / Load** option.

This option may be useful to compute a specific array with formulas - for example - in a spreadsheet, save in in .TXT format, and load the corresponding configuration in the *Automata for Art* application.

## SELECTION / COPY

Performs a copy of the selected field. To use with the **Selection/Paste** option.

## SELECTION / PASTE

Pastes the previously selected field with the **Selection/Copy** option, into the destination area.



## **SELECTION / 0-STATE**

Sets to 0 state the selected field.

## **HORIZONTAL SYMMETRY**

Computes an horizontal symmetry for the selected field.

## **VERTICAL SYMMETRY**

Computes a vertical symmetry for the selected field.

## ARRAY

Defines the array dimensions. Horizontal and vertical values are declared. New width and height of cells, are automatically computed and displayed, if the **Dynamic cell** option has been selected (See **Options/Preferences**).

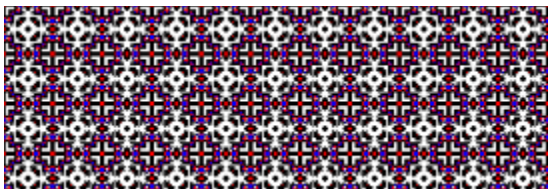
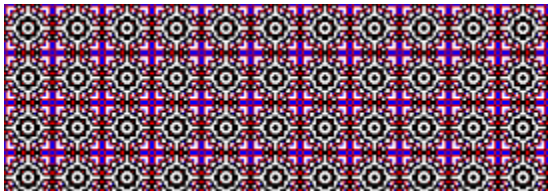
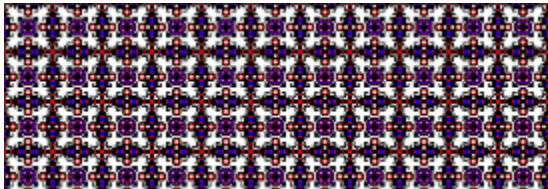
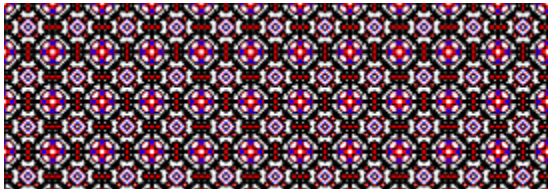
An error message is issued, if **Horizontal maximum**, or **Vertical maximum**, exceed certain limits: respectively 100 and 100.

## MOSAIC

This option displays a new window, composed of juxtapositions of the current bitmap, considered as a repetitive pattern.

It is useful to visualize and see the effect produced by the repetition of the current bitmap.

The following images results of the use of this option:



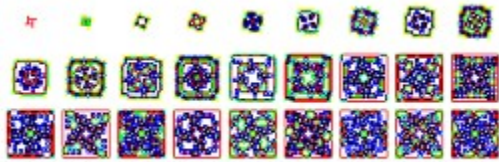
## MULTI-BITMAP

This option displays a new window, including the following states of the current automaton. The successive steps are displayed.

The following are examples of the resulting window, when this option is selected:



Here are the 27 first iterations of the 257-32-8-128-2-16-0-64 automaton, with a 32x32 array, initialized with a 8x8 screw:



# AUTOMATA

The following automata are available:

- **Life Game:** 499-8



- 387-8-0-4-0-0-0-112



- 387-8-0-0-16-0-0-100



- 289-64-0-128-16-4-0-10



- 289-72-0-128-16-4-2



- 257-32-8-128-2-16-0-64



- 297-64-16-132-0-0-0-10



- 337-0-128-4-2-0-0-40



- **Sierpinski:** 32-4 (only available on extended 4.1 version)



These automata are remarkable, and you may test their astonishing properties. But this list is non-exhaustive, and I recommend you to explore and experiment other automata.

## **RANDOM**

This option is necessary to adjust the random options parameters. A dialog box appears, allowing you to select the amount of elements initially affected to a selected automaton. For each state, the count of elements can be adjusted.



## **COLORS**

Using this option, a dialog box appears, allowing you to select a specific color, for each state. The default colors can be modified, to create special effects, and specific graphical patterns.

## LANGUAGE

This option allows the selection between the english or the french version of the software. *Automata for Art* is a bilingual version (**english** and **french**).

## TRANSITION RULES

Defines the **transition rules** for the *Automata for Art* application.

Please carefully read this section, which is very important, for an efficient use of *Automata for Art*.

Modifying parameters, and testing *Automata for Art* is highly recommended; many kinds of AUTOMATA for ART are astonishing, and their properties are very different.

The standard parameters are initialized with the automaton used to create the Life Game (John CONWAY ).

For 16-state and 9-neighbours automata, that are implemented, it is necessary to have an efficient **numbering method** for precise identification and individualisation of automata, in order to recognize without ambiguity whatever automaton has been selected. An amount of  $8^9$  different automata is **theoretically** available, corresponding to 8 states and 9 possibilities of transition for each state, according to the total count of neighbours for each cell: this count can be equal to 0, 1, 2, ..., 7, or 8, corresponding to 9 different possibilities.

In a perspective of **ascending comptability**, whose specifications are increase of available states and colors, with the same numbering method, we assign a number to every set of transition rules determining every state change: this allows the complete identification of a specific automaton with 8 numbers. If new states and colors, or additionnal neighbours, are subsequently added, it will not be necessary to modify the numbering system. Accordingly, we can consider our rules array of 8 states and 9 neighbours, as the top left part of an illimited and extensible array, consisting of m columns (neighbours) and n rows (states).

Consequently, the **Life Game automaton** corrisponds in this numbering system, to the automaton referenced 499-8-0-0-0-0-0-0, that is to say the **automaton 499-8**, because the 6 following states or colors are set to null and not used. We also remark another advantage of this numbering system: it is not necessary to specify the maximum count of available colors and states.

For automata of this family, the **n numbers** are coding the **whole set of transition rules to state n**, from 0 to  $2^9 - 1$  (that is to say from 0 to 511); the first number is coding the commutation rule to state 0. For the Life Game automaton, 499 corrisponds to  $2^0 + 2^1 + 2^4 + 2^5 + 2^6 + 2^7 + 2^8$ , because the transition to state 0 is made when the amount of life neighbours reaches 0, 1, 4, 5, 6, 7 ou 8. Similarly, 8, the second number identifying the automaton, is coding the transition rule to state 1, and corrisponds to  $2^3$ , because the only case where a cell becomes life, is when the total of its life neighbours reaches 3.

## NEIGHBORHOOD PARAMETERS

Neighborhood is parametrable, in order to allow the implementation of many automata. You can also give a coefficient  $\mathbf{b}$  to each of the 8 neighboring cells of a central site, and the central cell itself. So 9 total parameters of neighborhood are available.

For 1-dimensional automata, only 3 cells are eventually used: the central site  $\mathbf{a(0)}$  itself, and its left  $\mathbf{a(-1)}$  and right neighbors  $\mathbf{a(1)}$ .

For 2-dimensional automata, only 9 cells are eventually used: the central site  $\mathbf{a(0, 0)}$  itself, its left  $\mathbf{a(-1, 0)}$  and right neighbors  $\mathbf{a(1, 0)}$ , and the upper  $\mathbf{a(-1, -1)}$ ,  $\mathbf{a(-1, 0)}$ ,  $\mathbf{a(-1, 1)}$ , and lower sites  $\mathbf{a(1, -1)}$ ,  $\mathbf{a(1, 0)}$ ,  $\mathbf{a(1, 1)}$ .

When the sum  $\mathbf{a(i, j)*b(i, j)}$  is computed, a modulo is applied, depending on the modulo option selected. The  $\mathbf{b(i, j)}$  values are the coefficients for each of the 9 available neighbors.

With the parameter **Homogeneous neighbor mode** set to 1, the value  $\mathbf{a(i, j)}$  is:

- set to 1 if the current state of the cell is different from 0
- set to 0 if the current site is a 0-state cell

If the **Homogeneous neighbor mode** is set to 0, the value  $\mathbf{a(i, j)}$  is equal to the current value of the cellstate.


The calculus of the modulo applied to the cellular automaton may result from different techniques:

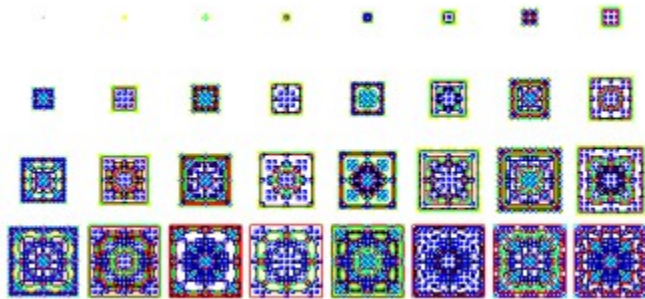
- 1) by default, this value is set to 10 (the maximum neighbors + 1)
- 2) this value can be user-defined
- 3) this value may be the maximum + 1 neighbors used in the **Transition rules** option
- 4) this value may be the maximum + 1 neighbors used in the **Neighborhood Parameters** option

A few neighborhood parameters are pre-defined:

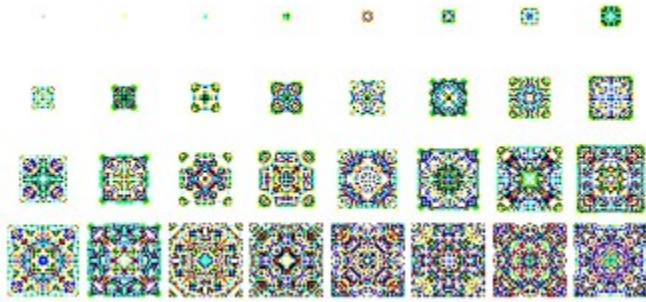
- **1-dimensional**: 0-0-0-1-1-1-0-0-0 from top left to bottom right
- **2-dimensional**: 1-1-1-1-0-1-1-1-1 from top left to bottom right
- **5-neighbors**: 0-1-0-1-1-1-0-1-0 from top left to bottom right
- **9-dimensional**: 1-1-1-1-1-1-1-1-1 from top left to bottom right

The available automata are called *Totalistic automata*.

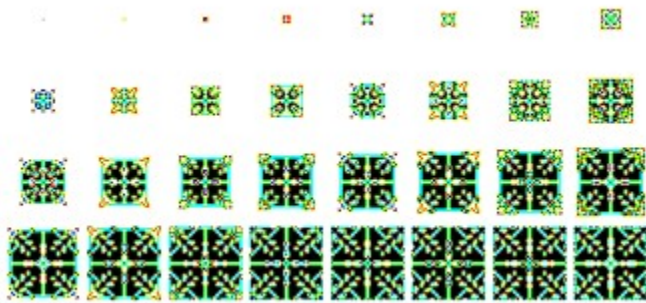
With the  automaton, a 49x49 array, a single seed initialization, **Homogeneous neighbor mode** = 1, and a standard 2-dimensional neighborhood, the 32 first iterations are the following:



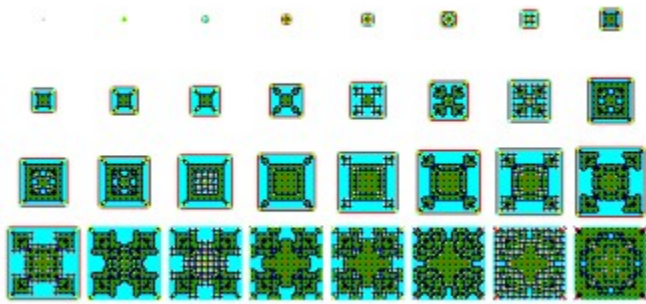
With the same automaton, but with a neighborhood parameters defined as (from left top to right bottom) 1-2-1-2-0-2-1-2-1 we obtain:



With the same automaton, but with a neighborhood parameters defined as (from left top to right bottom) 1-2-1-2-4-2-1-2-1 we obtain:



With the same automaton, but with a neighborhood parameters defined as (from left top to right bottom) 1-3-1-3-3-3-1-3-1 we obtain:



## PREFERENCES

Defines the user preferences:

**DYNAMIC CELL DIMENSIONS:** this option, when selected, induces automatic computation of cell size, adapted to current window dimensions. At the beginning, this option is non-active: the cell height and width are set to a value of 16 pixels.

**CELL WIDTH:** defines the cell width, to use with the **Array window**.

**CELL HEIGHT:** defines the cell height, to use with the **Array window**.

**CELL WIDTH INCREMENT:** defines the cell width increment value, to use with the **Increment cell dimensions** and **Decrement cell dimensions** options.

**ITERATION LIMIT:** this value is initially set to 500; it is used by the **Process/Until Stable** option. If no stable state is found for the current automaton, iterations are continued until an amount of 500 iterations will be reached. An error message is emitted if this value is less than the current **Level** value.

**ITERATION LIMIT INCREMENT:** this value is initially set to 250; it is used by the **Execution/Continuous** option. If the iteration limit is reached, a dialog box appears; if the user prefers to continue the iterations, the **iteration increment** is added to the initial **iteration limit**. For example, with the default parameters, the original iteration limit is 500. If this limit is reached, and the user wishes to continue, 250 is added, so the new value for the iteration limit is 750.

**CONTINUOUS DELAY:** defines the value in milliseconds (default value is 0), in order to generate a temporization, when using the **Execution / Continuous** option.

**GRID LINE WIDTH:** defines the grid line width, to use with the **Array window**. If a value of 0 is selected, no grid appears.

**ELLIPSE / RECTANGLE:** determines the current shape of each site, in the array window.

**PLANE / TORE:** selects a geometry mode, for the computation of the cellular automata. The **tore** selection is made to avoid difficulties generated by border conditions. The **tore** model is used to simulate an infinite array.

## STATISTICS

Display several statistics concerning the current state of the automaton.

## NEXT

Computes and displays next step of the current automaton.

An error message appears, if the level value of next iteration is greater than the **Iteration Limit**.



## **LEVEL**

Using this option, you can select a specific level, for the current automaton. The new state of the automaton is showed. For complex, large arrays, and high levels, a delay may be necessary.

## CONTINUOUS

Computes the iteration process, continuously. The iterative process is halted:

- if the user presses a key
- or if the iteration limit is reached

You can stop the process using the **Esc** key.

## CONTINUOUS UNTIL

Computes the iteration process, continuously, until the specified level is reached. The iterative process is halted:

- if the user presses a key
- or if the **Iteration limit** is reached

## STATUS BAR

A status bar displaying several informations appears on the bottom of the screen, when you select this option. The iteration current level, the amount of modified elements, the count of non-state-0 elements, are displayed on the screen, in an independent window.

An example of the informations displayed in the status bar, is shown below:

```
Level: 26  
Count: 708  
Changes: 0  
Automaton: 387 8 0 0 16 0 0 100 0 0 0 0 0 0  
0 0
```

## ARRAY WINDOW

Selecting this option activates a resizable **Array window**, representing the whole automaton current array. This option is useful for scientific study of the automata cells evolution.

## MOSAIC WINDOW

This option selects a resizable Background window, corresponding to the option specified with the **Options/Mosaic** option.

## HELP INDEX

*Automata for Art* is a self-documented software. This option load the Help file, named ARTUS.HLP.This option displays the index of the help file.

## REGISTRATION CODE

The limited version of *Automata for Art* is a shareware software. Once you have sent your payment, a registration code is sent to you. Please then enter this code in the corresponding dialog box. Once you have entered this code, you will automatically have access to the extended version of *Automata for Art*.



## INCREMENT CELL DIMENSIONS

Adds one pixel (by default) to the current width and length of the current *Array* window cells.

## DECREMENT CELL DIMENSIONS

Subtracts one pixel (by default) to the current width and length of the current *Array* window cells.

## **SAVE BITMAP AS**

Saves the image of the *Bitmap* window in a file whose format is.BMP.

## **SAVE ARRAY AS**

Saves the image of the *Array* window in a file whose format is .BMP.

## **SAVE MOSAIC AS**

Saves the image of the *Mosaic* window in a file whose format is.BMP.

## **ARRAY 16 x 16**

Selects an array whose lines and rows count is equal to 16. This format can notably be useful when one wants to create an icon whose dimensions are 16 x 16.

## **ARRAY 32 x 32**

Selects an array whose lines and rows count is equal to 32. This format can notably be useful when one wants to create an icon whose dimensions are 32 x 32.





