

**GBlankerMod1**

COLLABORATORS
---------------

	TITLE : GBlankerMod1		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		July 22, 2024	

REVISION HISTORY
------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>GBlankerMod1</b>	<b>1</b>
1.1	Garshneblanker modules collection vol.1 . . . . .	1
1.2	Paragraph 1: GBlankerMod1.lha archive . . . . .	1
1.3	Paragraph 2: Installation . . . . .	2
1.4	Paragraph 3: Bugs report and contributions . . . . .	3
1.5	Paragraph 4: About the author . . . . .	4
1.6	Paragraph 5: Acknowledgements . . . . .	4
1.7	Paragraph 6: Distribution . . . . .	4
1.8	Paragraph 7: Disclaimer . . . . .	4
1.9	Crazy Ant blanker . . . . .	5
1.10	Demons blanker . . . . .	6
1.11	Scrawls blanker . . . . .	8

## Chapter 1

# GBlankerMod1

### 1.1 Garshneblanker modules collection vol.1

GarshneBlanker modules collection

-----  
volume I  
-----

version 1.0 (Jan 24 1995)

programs, sources and documentation are  
copyright 1995 Marzio De Biasi  
All Rights Reserved

INDEX

- 
- [1] GBlankerMod1.lha archive
  - [2] Installation
  - [3] Bugs report and contributions <--<< please read this paragraph
  - [4] About the author
  - [5] Acknowledgements
  - [6] Distribution
  - [7] Disclaimer

### 1.2 Paragraph 1: GBlankerMod1.lha archive

Foolish test:

-----  
Do you know what "screen blanker" mean?

~~~~~

|

+--> NO: sorry, you lose!

|

+--> YES: Have you already downloaded M. D. Bayne's great Garshneblanker?

| ~~~~~

|

```
+-----> NO: download it immediately!
|
+-----> YES:  Would you like to install three new blankers?
| ~~~~~
|
+-----> NO:  drop this archive (SIGH!!)-;
|
+-----> YES: I can help you! ;-D
```

In this archive you can find three NEW Garshneblanker modules:

- 1) CrazyAnt blanker : a blanker based on a 2D Turing machine.
- 2) Scrawls blanker : a blanker based on iterated function systems (IFS).
- 3) Demons blanker : a blanker based on a cellular automata.

They are based upon three articles that I've read on "Scientific American" (italian edition) and I really don't know whether they are original blankers or not (I think they are :-).

They have been developed and tested using Garshneblanker version 38.8  
-----  
(Release 36.6) and its source code distribution; if you have an older  
-----  
version of Garshneblanker try 'em as well, but I'm not sure they will run.

You can find their source codes in directories:

```
GSource/blankers/CrazyAnt/
GSource/blankers/Scrawls/
GSource/blankers/Demons/
```

but, in order to re-compile them, you'll need to get the original Garshneblanker source distribution.

If you want to know more about them, you'll find some technical stuff in files:

```
Docs/CrazyAnt.doc
Docs/Scrawls.doc
Docs/Demons.doc
```

## 1.3 Paragraph 2: Installation

---

>>> If you have an Amiga equipped with 68020 or higher CPU:

```
copy all files in directory GBlankerMod1/blankers020 to your
-----
Garshneblanker blankers directory.
-----
```

For example, if you've installed Garshneblanker and its modules in  
SYS:WBStartup/, you can enter this from CLI:

```
CLI> copy GBlankerMod1/blankers020/#? SYS:WBStartup/blankers/
```

>>> If you have an Amiga equipped with 68000 CPU:

```
copy all files in directory GBlankerMod1/blankers to your
-----
Garshneblanker blankers directory.
-----
```

For example, if you've installed Garshneblanker and its modules in  
SYS:WBStartup/, you can enter this from CLI:

```
CLI> copy GBlankerMod1/blankers/#? SYS:WBStartup/blankers/
```

## 1.4 Paragraph 3: Bugs report and contributions

The blankers have been developed and tested on an Amiga 4000/030 + 68882,  
and they all seem to work right.

```
IF YOU ENCOUNTER SOME PROBLEMS INSTALLING THEM, FIND OUT SOME BUGS OR HAVE
-----
SOME SUGGESTIONS (ABOUT THEM OR ABOUT NEW BLANKERS YOU WOULD LIKE TO HAVE)
-----
PLEASE CONTACT ME!!!!
-----
```

```
+-----+
|
| IF YOU INSTALL THEM AND LIKE THEM (OR HATE THEM) PLEASE SEND ME AN
| -----
| E-MAIL: REALIZING THAT SOMEONE USE MY MODULES, WILL BOOST MY
| -----
| BLANKERS-PRODUCTION !!! :-)
| -----
|
+-----+
```

## 1.5 Paragraph 4: About the author

If you want to contact me:

-----  
De Biasi Marzio  
via Borgo Simoi, 34  
31029 Vittorio Veneto (TV)  
Italy            E-Mail: debiasi@dimi.uniud.it  
-----

## 1.6 Paragraph 5: Acknowledgements

THANKS TO Micheal D. Bayne FOR HIS GREAT BLANKER.  
-----

and to Martin Huttenloher for his beautiful MagicWB icons.

## 1.7 Paragraph 6: Distribution

"Garshneblanker Modules Collection Volume I" package is FREeware.  
You may freely distribute it as long as all of its files are included  
in their original form without additions, deletions, or modifications,  
and only a nominal fee is charged for its distribution. It cannot be  
distributed as a part of commercial product without author's written  
permission.

- \* This program may be included in freeware collections, providing that the  
previous conditions are respected.
- \* TO SOFTWARE REVIEWS: if you include this program in one of your  
disks, please send me a copy of the review, thanks :-)

## 1.8 Paragraph 7: Disclaimer

No responsibility or liability will be accepted for any damage that may  
appear to have resulted from use of this program. All use is at your own  
risk. The software is provided "as is" without any warranty implied or  
otherwise to the fitness or accuracy of the software and documentation.  
The author reserves the right to update the software and/or documentation  
without notice.

---

## 1.9 Crazy Ant blanker

```
GarshneBlanker modules collection
-----
                                volume I
-----

                                version 1.0 (Jan 24 1995)

programs, sources and documentation are
copyright 1995 Marzio De Biasi
All Rights Reserved

CrazyAnt blanker
*****
                                v1.0
technical information
```

### Description

-----

The screen represents a 2D finite grid; at first, each cell of the grid is black (few of them can be randomly coloured).

A crazy ant (the white cell) is placed in the center of the grid, then the ant starts its long trip: at each tick, it changes the color of the cell where it is, turns itself 90 degrees left or right (the choice is made by its crazy-brain) and move forward one step.

The ant dies when it gets out of the screen, but a new ant is suddenly set free.

The random-ant produces nice configurations that seems not to be so random!!!

### Garshneblanker preference window

-----

You can set the following parameters:

Ant size : the size of the grid-cells (and of the ant).

-----

Initial blobs : number of cells that are randomly coloured before the  
----- ant is set free; the configurations will be very  
interesting even though there are no initial blobs.



Delay : ant speed; lower values correspond to a faster ant.

-----

#### Algorithm description

-----

NC : screen colors (NC = 2 ^ Screen\_depth)  
 GRID\_W : grid width (GRID\_W = Screen\_width / ant\_size)  
 GRID\_H : grid height (GRID\_H = Screen\_height / ant\_size)  
 GRID(x,y) : color of cell at (x,y)  
 BRAIN : ant's brain, a vector of NC bits  
 (ANT\_X, ANT\_Y): ant's position  
 DIRECTION : ant's direction (0 = N, 1 = E, 2 = S, 3 = W)

- while (ANT\_X, ANT\_Y) is inside screen region repeats these steps
- c = GRID(ANT\_X, ANT\_Y)
- if ( BRAIN[c] == 1)
  - then DIRECTION = (DIRECTION + 1) mod 4
  - else DIRECTION = (DIRECTION + 3) mod 4
- GRID(ANT\_X, ANT\_Y) = (c + 1) mod NC
- moves ant one step in direction DIRECTION
- redraws new-coloured cell and ant

## 1.10 Demons blanker

GarshneBlanker modules collection

-----

volume I

-----

version 1.0 (Jan 24 1995)

programs, sources and documentation are  
 copyright 1995 Marzio De Biasi  
 All Rights Reserved

Demons blanker  
 \*\*\*\*\*  
 v1.0  
 technical information

#### Description

-----

This blanker is a N-states cellular automata, which follow this simple rule:

"at each generation, if a cell in state X is surrounded by at least one cell in state X+1, then it will change its state to X+1"

The blanker begins with a chaotic configuration of cells ("rubble" phase); the state of a cell is represented by its color.

The first generations keep a chaotic aspect except for small "blobs" made of color waves.

Whit time these blobs slowly grow and wrap up the chaotic cells ("drops" phase).

Some generations after, the grid of cells has become a field of beautiful spirals that slowly expand themself; only a few chaotic spots survive: the DEMONS ("defects" phase).

#### Garshneblanker preference window

You can set the following parameters:

Cells size : the size of the grid cells

-----

Colors : number of colors to be used; fewer colors usually  
 ----- means that the automata will reach the "defect" phase  
 in fewer generations.

Width : grid width

\_\_\_\_\_

Height : grid height

-----

NOTE: in order to obtain a fast refresh, you must use small values for Width and Height, but, if you are patient, try the highest resolution, set the cells size to 1, and wait for the last phase.

#### Algorithm description

-----

NC : screen colors (NC =  $2^{\text{Screen\_depth}}$ )

COLOR(x,y) : color of cell at (x,y)

COLOR'(x,y) : next color of cell at (x,y)

GRID\_W : grid width

GRID\_H : grid height

```

- for each generation do the following steps
    - for each cell at (i,j) do the following steps
- c1 = COLOR((i + 1) mod GRID_W, j)
- c2 = COLOR((i - 1) mod GRID_W, j)
- c3 = COLOR(i, (j + 1) mod GRID_H)
- c4 = COLOR(i, (j - 1) mod GRID_H)
- if at least one among c1,c2,c3 and c4 equals (COLOR(i,j)+1) mod NC
  then COLOR'(i,j) = (COLOR(i,j) + 1) mod NC
  else COLOR'(i,j) = COLOR(i,j)
    - for each cell at (i,j) do the following steps
- COLOR(i,j) = COLOR'(i,j)
- redraw cell at (i,j)

```

## 1.11 Scrawls blanker

```

                        GarshneBlanker modules collection
-----
                        volume I
-----

                        version 1.0 (Jan 24 1995)

programs, sources and documentation are
copyright 1995 Marzio De Biasi
All Rights Reserved

Scrawls blanker
*****
v1.0
technical information

```

### Description

-----

This blanker calculates the attractor of a random generated iterated function system (IFS).

An IFS is a set of maps  $f_1, f_2, \dots, f_N$  of the plane to itself:

$f_i =$

```

{ x' = a * x + b * y + c
{
{ y' = d * x + e * y + f

```

which, under particular conditions, has an unique attractor (invariant set), that is a set A that satisfies the following equation:

$$A = f_1[A] \cup f_2[A] \cup \dots \cup f_N[A]$$

Given a point  $P$  that belongs to  $A$ , the points  $f_1[P]$ ,  $f_2[P]$ , ...,  $f_N[P]$  belong to  $A$  too.

The attractor  $A$  can be approximated recursively applying one of the  $f_1 \dots f_N$  (randomly chosen) to a starting point  $P$  of  $A$ .

The attractor is a fractal that CAN look very good on a computer monitor; unlikely, there is no way (I think) to establish whether an IFS will generate a beautiful attractor or will only generate strange scrawls.

However, observe this blanker for some minutes, and you'll probably see some good stuff.

#### Garshneblanker preference window

-----

You can set the following parameters:

Equations : number of functions of the IFS.

----- You can obtain good results with only three equations, but if you use a lot of equations and you are VERY VERY LUCKY you can get something that looks like a tree, a mountain, a human face or a Picasso's painting.

Factor : the greater absolute value that will assume  $c$  and  $f$  parameters in IFS equations.

Iterations : number of iterations used for drawing an attractor: ----- higher values correspond to a better defined attractor.

Zoom : if you obtain only small attractors, try increasing ---- zoom value.

Delay : drawing speed; lower values correspond to a greater speed. -----

#### Algorithm description

-----

NITER : number of iterations

NF : number of functions

FUN(N) : Nth function, its parameters are indicated with FUN(N).a, FUN(N).b, ..., FUN(N).f

FUN(N).prob : probability assigned to function N

FUN(1).prob + FUN(2).prob + ... + FUN(NF-1).prob = 1

COLOR(x,y) : color of pixel at (x,y)

- for each  $0 \leq I < NF$  randomly set FUN(I).a, FUN(I).b, ..., FUN(I).prob

- 
- randomly choose a point  $P=(x,y)$
  - execute the following steps 20 times
  - randomly choose a function  $I$
  - calculate  $P' = \text{FUN}(I)[P]$
  - set  $P = P'$
  - execute the following steps NITER times
  - randomly choose a number  $Z$  between 0 and 1
  - find the first function  $I$  that satisfies the following
$$(\text{FUN}(1).\text{prob} + \dots + \text{FUN}(I).\text{prob}) - Z \leq 0$$
  - calculate  $P' = \text{FUN}(I)[P]$
  - set  $P = P'$
  - set  $\text{COLOR}(x,y) = \text{COLOR}(x,y) + 1$
-