

# **SleepingPointers**

Antonio Manuel Santos

**COLLABORATORS**

	<i>TITLE :</i> SleepingPointers		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Antonio Manuel Santos	December 8, 2024	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>SleepingPointers</b>	<b>1</b>
1.1	Sleeping Pointers . . . . .	1
1.2	Distribution of Sleeping Pointers . . . . .	1
1.3	Requirements of Sleeping Pointers . . . . .	3
1.4	Introduction to Sleeping Pointers . . . . .	3
1.5	What can Sleeping Pointers do and not do... . . . .	3
1.6	How is this implemented . . . . .	4
1.7	Bugs ?? oh no !!! . . . . .	5
1.8	Sleeping Pointers V1.0 . . . . .	5
1.9	What to do Next ? . . . . .	5
1.10	How to reach me... . . . .	6
1.11	Good Stuff to registered users. . . . .	6
1.12	How to install SleepingPointers... . . . .	7

---

## Chapter 1

# SleepingPointers

### 1.1 Sleeping Pointers

Sleeping Pointers V1.0  
©1995 António Manuel Santos, All rights reserved  
ShareWare

Distribution        Distribution and registration  
Requirements        What do you need.  
Introduction        What is this Sleep stuff ?  
Installation        How to Install...  
Features            What can Sleeping Pointers do and not do...  
Technical info      How is this implemented ?  
Known Bugs         Bugs ?? oh no !!!  
History            Sleeping Pointers V1.0  
Future             What to do next ?  
Contact            How to reach me...  
Registered Users    Good stuff to registered users.

### 1.2 Distribution of Sleeping Pointers

This program is ShareWare. This means that you can test the program during one month. After this period you must send either:

- I    - A card of your home town to ~my address .
  
  - II   - A source of a program of your own. Can be any thing, but must be of some interesting in programming the Amiga. Needed are:
    - \* Examples of AGA programming:
      - banging directly into the hardware.
      - friendly OS usage of AGA.
  
    - \* Examples of 68040 programming:
      - how to implement virtual memory.
-

- \* Guide of 68040 instructions: if you know some text/guide on internet, please tell-me !

III - A email to my internet account .  
Think of that email as becoming a registered user .  
That way i can receive feedback from SleepingPointers users.

IV - A small contribution in CASH (only CASH) to the author.  
Send the value you think the program deserves.

In return you will receive the latest version available. Also you will receive some more other goodies like:

- \* more animated pointers done by me.
- \* animated pointers done by other SleepingPointers users.
- \* programs that i do to me and may be of your interest.

By doing this contribution, you assure further development of this program and many many more utilities i have in mind. Thanks in advance.

This SHAREWARE version can be freely distributed provided that all the associatedd files are distributed with the program itself. Neither the program nor their associated files must be modified in any way. The complete distribution package can be compressed with LHA, LHarc, LZX or similar for distribution convenience. No charge must be made for use or distribution of this package. You can upload this package to any BBS, include it in a magazine's coverdisk (if you do so, please send-me a copy of the magazine if possible, or tell-me that you have done so), in a Public Domain library or in recompilation CDs of PD/Freeware/Shareware.

If you want to get the lastest version available or contact me for some reason, please send email to:

138058@alfa.ist.utl.pt with subject: SleepingPointers.

The complete distribution package contains:

SleepingPointers-Normal	normal clock with spinning hands (with icon)
SleepingPointers-Cute	cute sand watch (with icon)
SleepingPointers.guide	english guide ( with icon)
SleepingPointers.readme	english aminet .readme
SandClock.iff	How to Draw custom busy pointers (with icon)
NormalClock.iff	The Normal Clock animation (with icon)

This software is provided 'AS IS' without warranty of any kind, either expressed or implied. By using this package you agree to accept the entire risk as to the quality and performance of the program, or even the document you are reading.

The SleepingPointers.guide.info and SleepingPointers.info drawer icons were taken from the Iconographics package available on aminet. If you dont like or think can make improved icons for the program/guide, draw some and send me so that i include them on the next aminet release.

---

## 1.3 Requirements of Sleeping Pointers

SleepingPointers need Workbench 3.0 or plus to work. You do not need AGA, but that will give hi-resolution busy pointers.

Also, PatchPointer should be installed, so that pre 3.x software that sets busy pointers can be patched too. Check PatchPointer on aminet (PatchPointer v1.02 ©1993 by Steve Lemieux).

To have hires pointers on AGA machines, you will need a screen monitor that can display a screenmode of DblPAL, Multiscan, or Super72. In PAL/NTSC, the busy pointer will be in lowres.

## 1.4 Introduction to Sleeping Pointers

What is this Sleep stuff ??

Well, have you ever work on WindowsNT and saw the animated busy pointers?

"Gosh, my Amiga is better than a PeeCe and my Workbench only shows a little-tiny stopped busy pointer!"

You may say: "use PointerX, or LacePointer"

no way : PointerX only shows a ugly busypointer in lowres !!!

You can also say: "use PatchPointer to make the busypointer hires."  
Ok ... try it and you will see the busypointer becomes stopped!  
Damn ...

LacePointer in other way only works in screenmodes of PAL/NTSC.  
If you have a brand new AGA amiga, that is in MULTISCAN or DBLPAL, or something else, no LacePointer.

Those little problems, motivated me to program SleepingPointers.  
After some time, SleepingPointers has born!

Now, animated busy pointers in hires is easy.

If i receive enough feedback, i will do a Prefs Program so that you can load you favourite ANIMATION/ANIMBRUSH of a busy pointer and use it!

## 1.5 What can Sleeping Pointers do and not do...

- On AGA machines, display a hires animated busy pointer, if the screen mode selected is not one of :

PAL  
NTSC

---

- In use with PatchPointer, make Pre 3.x software that asks for busy pointers, display an animated one.
- If you do not use PatchPointer, Sleeping Pointers, cannot make pre 3.x software show busy pointers in hires.
- Cannot load user ANIMATIONS. See Registered Users .
- Cannot change the pointer preferences color. Use the supplied from Workbench distribution disk.
- Only uses 3 colors busy pointers. Some time later i will make 16 colors busy pointers.
- Cannot have the pointer colors cycle in a specified way. Some time later i will make that feature also.
- Has not a preferences program. See Registered Users .

## 1.6 How is this implemented

I implemented Sleeping Pointers in the most friendly way as possible i could. As i have a A4000/040, all the software must behave the most friendly as possible, if not, BANG ... \*GURU\*. This is also true for any other AMIGA computer (Thank God ← !!!)

First, i have patched the following routines from intuition.library:

```
OpenWindow();
OpenWindowTagList();
CloseWindow();
SetPointer();
ClearPointer();
SetWindowPointerA();
```

For every Window Opened, i add a client to a private list on SleepingPointers so that i can walk that list from time to time to change the busy pointer. When a busy pointer is requested, i will not let intuition.library know that. I put a flag on my client saying : yep ... i wanna a busy pointer (but animated one !). In the main loop of my program, i use timer.device to awake from a Wait() from time to time. In each invocation, i increment a counter of the current frame of my busy pointer, walk through my private list and if a busy pointer is requested, i just call the original SetWindowPointerA() with the current frame requested. Note that all the frames of the animation, are allocated with NewObject(NULL, "pointerclass", ...); so that i can request hi resolution busy pointers.

The client arbitration is done with Semaphores so that no garbage during setting/add/rem clients is done.

The busy pointers were drawn by me on Brilliance.  
The program is in C, compiled with SASC 6.55  
Used DevPac to include the pointer animations in the resulting executable.

source size is about 1178 lines in C, but many comments and blank lines. Also has some stuff that is still not implemented like the Preferences GUI, etc etc.

## 1.7 Bugs ?? oh no !!!

If you know any bug other than the ones listed here, please tell me so. For that, @{"Contact me " link "Contact" 0}.

BUGS

version 1.0 (23 Jul 1995)

- There is a known bug to me that i still could not trace:  
If you start a new SleepingPointers, it will \*GURU\* your machine.  
If you need to end SleepingPointers, just select the Exchange Window and remove SleepingPointers.

## 1.8 Sleeping Pointers V1.0

- AWSM-09beta.lzh

Pre-Release of SleepingPointers.  
Please delete this file. It's just too bad !!! :^)

- SleepingPointers.lha - SleepingPointers V1.0 (23 Jul 1995)

First Aminet Release. Hope receive enough feedback.  
Also this is my first Aminet Release of any program.

Done a Pre-Release that will \*GURU\* any Amiga. It's called AWSM-09beta.lzh and is on /util/cdity/AWSM-09beta.lzh

If you see that file, Please delete it. SleepingPointers its very stable and AWSM-09beta not.

## 1.9 What to do Next ?

- Normal pointer also animated ... could be cool and its very easy to do.
  - Implement a Preferences Program, so that users can load ANIM/ANIMBRUSHES of custom busy pointers.
  - 16 colors normal and busy pointers.
  - more than 64 pixels wide pointers. Tell-me if you really need this feature.
-

- custom definition of color cycling in pointer colors. This is for example, your pointer is dark blue, and have cycle like this: darkblue, blue, violete, pink, red, darkred, red, pink, violet, blue, darkblue, etc etc ...

Could cause some groovy effects :)

- What you think it will cool to add to SleepingPointers ? please tell me that!

## 1.10 How to reach me...

You can reach me on :

\* Internet as l38058@alfa.ist.utl.pt (note the ele [L] in username)

\* SnailMail:

António Manuel Tavares de Oliveira Santos  
Rua do Zaire, 5 1 dto  
1170 Lisbon, Portugal

\* IRC:  
as 'psys' on channels:

#amiga  
#amigadev  
#amigascne  
#amigafr

i will be mostly on #amiga

Every one that contacts me receives a answer.

## 1.11 Good Stuff to registered users.

If you register, you will get support and SleepingPointers will get bigger and better. Also if you want some Workbench Utility, just tell me so.

If you register:

\* I will send any new updates of SleepingPointers to you by email.

In the next TODO list of updates is a Prefs Program to support loading of ANIMS to use as busy pointers. Also there is a idea of making normal pointers animated.

\* I will send any new busy pointers i do or i receive from registered users to you by email.

\* until i get no prefs program, you can send me your animations and i will

---

make a custom SleepingPointers to you. Some time later, i will send you the custom SleepingPointers by email.

If you want to send some normal/busy animated pointers, please uuencode them, or use pine.

If you have some busy pointers drawing/sketches/animations, please send me so that i can send those to all SleepingPointers registered users.

Thanks in advance to all registered users that i will get.

If you want to register, please send a email to me stating:

- You System configuration: computer, processor, harddisk, monitor, ram, etc etc
- Your preferable contact.
- Your registration, just take a look at Distribution
- Any other things you may want.

## 1.12 How to install SleepingPointers...

Easy. Just copy the executable you want to use,  
SleepingPointers-Normal  
SleepingPointers-Normal.info

or

SleepingPointers-Cute  
SleepingPointers-Cute.info

to your WBStartup Drawer.

If you want to set a different animation speed or a diferent task priority than the default (15) you can start SleepingPointers from your file s:user-startup ↔  
:

```
SleepingPointers 1000 10
                  ^^^^^^ ^^
                  |   \_ task pri (default: 15)
                  |
                  \_____ tick delay in micro seconds (default: 10000)

                  1000 will animate the busy pointer very fast
```