Atlantis

COLLABORATORS			
	TITLE : Atlantis		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 26, 2024	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

1 Atlantis 1

1.1 Atlantis 1

1.2 Atlantis - Introduction 1

1.3 Atlantis - Disclaimer 3

1.4 Atlantis - Distribution policy and License 5

1.5 Atlantis - Registration 5

1.6 Atlantis - Requirements 7

1.7 Atlantis - Installation 7

1.8 Atlantis - Features 8

1.9 Atlantis - How to use 9

1.10 Atlantis - The Preferences Window 15

1.11 Atlantis - The Arexx Interface 16

1.12 Atlantis - The Arexx Interface - ASKUSER 17

1.13 Atlantis - The Arexx Interface - BOX 18

1.14 Atlantis - The Arexx Interface - CLEARMARK 19

1.15 Atlantis - The Arexx Interface - FILL 20

1.16 Atlantis - The Arexx Interface - GETFREEMACRO 20

1.17 Atlantis - The Arexx Interface - GRABFROMMAP 21

1.18 Atlantis - The Arexx Interface - GRABBLOCKS 22

1.19 Atlantis - The Arexx Interface - HELP 23

1.20 Atlantis - The Arexx Interface - HIDE 24

1.21 Atlantis - The Arexx Interface - INFO 24

1.22 Atlantis - The Arexx Interface - INFORM 25

1.23 Atlantis - The Arexx Interface - KILLMACRO 26

1.24 Atlantis - The Arexx Interface - LINE 27

1.25 Atlantis - The Arexx Interface - MAPHEIGHT 27

1.26 Atlantis - The Arexx Interface - MAPWIDTH 28

1.27 Atlantis - The Arexx Interface - MARKX 29

1.28 Atlantis - The Arexx Interface - MARKY 29

1.29 Atlantis - The Arexx Interface - NEXTSETMARK 30

1.30 Atlantis - The Arexx Interface - NUMSETMARKS 31
1.31 Atlantis - The Arexx Interface - PLOT 32
1.32 Atlantis - The Arexx Interface - QUIT 33
1.33 Atlantis - The Arexx Interface - RECT 33
1.34 Atlantis - The Arexx Interface - SETMARK 34
1.35 Atlantis - The Arexx Interface - SHOW 35
1.36 Atlantis - The Arexx Interface - SLEEP 36
1.37 Atlantis - The Arexx Interface - UPDATE 36
1.38 Atlantis - The Arexx Interface - USEMACRO 37
1.39 Atlantis - The Arexx Interface - Version 38
1.40 Atlantis - The Arexx Interface - WAKEUP 38
1.41 Atlantis - The file format 39
1.42 Atlantis - The history of Atlantis 39
1.43 Atlantis - Known bugs 40
1.44 Atlantis - Next version 40
1.45 Atlantis - Thanks to 41
1.46 Atlantis - Contact Address 41
1.47 Atlantis - Glossary 42
1.48 Atlantis - Alphabetical INDEX 45

Chapter 1

Atlantis

1.1 Atlantis

Atlantis version 1.1
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
© 1997 Staffan Palmroos

Introduction	What does this program do?
Disclaimer	Don’t blame me!
Distribution	and Licence agreements
Registration	This is SHAREWARE!
Requirements	What you need to run Atlantis
Installation	Creating a nice environment
Features	Why should I use this program?
Tutorial	How to use Atlantis
Preferences	Customizing Atlantis
ARexx	The ARexx interface
Files	Description of the map file format
History	Changes & Bugfixes
Bugs	What doesn’t work properly?
Future	What do I get if I register?
Thanks to	Thanks to these people
Author’s address	Who and where am I?
Glossary	explains terms used in this text

> "There is no reason for any individual to have a computer in their home."
> -- Ken Olson, President of DEC, World Future Society Convention, 1977

1.2 Atlantis - Introduction

About Atlantis

Several years ago I started writing a platform game and found myself wanting an editor to create levels for it. I found one, but it was not good at all, but I put up with it for the moment. Later, when I finally got hooked
~~~~~~~~~~~~~~~~~~~~~~~~~~~~

up to the Internet, I searched AmiNET thoroughly and found a few others, but none of them were any good either, so I started writing on my own, Atlantis.

As said above, this program is used to create game levels (I shall refer to them as "maps" in the rest of this document) for platform games. These maps are created from a collection of building blocks stored in a IFF picture file. This technique is called 'Block graphics' and is very commonly used, since it can save a lot of memory.

It works like this:

Every block in the block file is numbered from 0 to N - 1, where N is the number of blocks in the block file. The map is then constructed as a two-dimensional array of these numbers.

Example:

Consider the characters in the quoted string below as 16 x 16 pixel blocks in a block file:

```
  " /\|-*"
```

The blocks in the block file are numbered:

```
  Block  Block number
  -----  ------------
   " "        0
    /         1
    \         2
    |         3
    -         4
    *         5
```

The blocks in the block file are then used to create a map in Atlantis:

```
    //        \   <- The map
   //   ****   \
  ||****----****||
  ||****----****||
   \   ****   //
    \        //
```

The block numbers of the map are saved out to a file. The contents of the file could look like this:

```
  0011000000002200  <- Atlantis output a map file that looks like this.
  0110005555000220
  3355554444555533
  3355554444555533
  0220005555000110
  0022000000001100
```

Now, this looks simple! Couldn't you do this without a program like

Atlantis?

Well, designing maps by typing the numbers directly is not easy at all. In fact, it is almost impossible to create a good map this way. With Atlantis you instantly see what the map is going to look like in the game later on. Also, the powerful tools in Atlantis makes it fast and easy to make enjoyable and tricky levels. Combined with the powerful ARexx interface you can create very advanced maps, like mazes and such.

The primary purpose of the DATA statement is to give names to constants;
instead of referring to pi as 3.141592653589793 at every appearance, the
variable PI can be given that value with a DATA statement and used
instead of the longer form of the constant. This also simplifies
modifying the program, should the value of pi change.
-- FORTRAN manual for Xerox Computers

1.3 Atlantis - Disclaimer

There is no WARRANTY for this program to the extent permitted by applicable law. The program is provided "as is", without warranty of any kind, either expressed or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with the user. Should the program prove defective, the user assume the cost of all necessary servicing, repair or correction, not the author or the copyright holder.

Under no circumstances, unless required by applicable law or agreed to in writing, can the author, the copyright holder or any third party who may redistribute the program be hold liable for any kind of damage, direct or indirect, caused by the use or misuse of the program or any of its associated files.

If you do not agree with this, please delete this program and all it’s associated files. For distribution policies, see the distribution section.

Atlantis was developed on:

* Amiga 1200, KS3.0 (V39), 68030 @ 50 Mhz, 10 Mb Ram.

and also tested on:

* Amiga 1200, KS3.0 (V39), 68020 @ 28 Mhz, 68882, 6 Mb Ram.

...and found working ok. With any other equipment the behaviour and functionality of the program is not known nor guaranteed. See the Requirements section to find out if you have a sufficient setup to run the program. The program has been thoroughly tested with the Enforcer - Mungwall - SegTracker combination, and has been found working ok. This does not however guarantee that it will work properly under all circumstances.

I have also tried to simulate low-memory conditions with 'Eatmem' from the former Commodore, but it is difficult to do that with a MUI application. If the program behave badly in such conditions, please notify me so that I can do something about it.

This application uses

MUI - MagicUserInterface

(c) Copyright 1993-96 by Stefan Stuntz

MUI is a system to generate and maintain graphical user interfaces. With the aid of a preferences program, the user of an application has the ability to customize the outfit according to his personal taste.

MUI is distributed as shareware. To obtain a complete package containing lots of examples and more information about registration please look for a file called "muiXXusr.lha" (XX means the latest version number) on your local bulletin boards or on public domain disks.

If you want to register directly, feel free to send

DM 30.- or US$ 20.-

to

Stefan Stuntz
Eduard-Spranger-Straße 7
80935 München
GERMANY

Support and online registration is available at

http://www.sasg.com/

Atlantis uses the external MUI custom class Icon.mcc (included in the archive) by Russell Leighton (russ@sneezy.lancaster.ca.us). Icon.mcc is © 1996 by Russell Leighton. See the Icon.mcc docs for more information.

The keyfile system of this program is using the RSA Data Security, Inc. MD5 Message Digest algorithm for authentication purposes. The keyfiles are then encrypted with my own RSA Public-Key cryptosystem routines. This makes it practically impossible to forge keyfiles, so it is not even worth trying. There are also additional security checks in the program code. ANY ATTEMPT TO FAKE KEYFILES MAY BE FATAL FOR YOUR EQUIPMENT OR DATA, so don't even try it. You have been warned.

PL/1, "the fatal disease", belongs more to the problem set than to the solution set.
-- E. W. Dijkstra

1.4 Atlantis - Distribution policy and License

Distribution

- You may not in any way modify the archive. If you are to redistribute the archive, for example on cover disks or cd:s, you must include the archive in its original form, and not remove or add anything to it.

- Urban Müller of the AmiNET archive is hereby explicitly granted to redistribute the archive in its original form on the AmiNET CDs.

- If you review this program in any media, such as a computer magazine, please notify me about it, and why not send me a copy of the article? I am always interested in criticism, positive or not.

Licence

- You may not disassemble, decompile, re-source or otherwise reverse engineer the program.

- You agree to cease distributing the program and all it's associated files if requested to do so by the copyright holder.

- By copying, distributing and/or using the program you indicate your acceptance of the terms and conditions stated in this text.

1.5 Atlantis - Registration

Registering Atlantis

Atlantis is SHAREWARE. This means that if you like this program and use it regularly, you are adviced to pay a modest registration fee to the author. When registering you will recieve a special personalized 'keyfile' which will enable all features and disable the restrictions of the program. Note that this keyfile is personal and it is an illegal act to give out copies of the key file to other persons, even to the closest of friends.

Single-user registration.

The single-user registration fee is

SEK	160,-
FIM	120,-
US$	20,-
UKP	15,-
DM	30,-

I believe that at least one of the above stated currencies is available in most countries, so don’t send me any other currency, since I have to pay a high fee at the bank to exchange the money to my own currency.

Payment mehods

Currently you can only use the "money-in-an-envelope" method of payment. In the future I will investigate other payment methods, but for now this method will have to do. I have spoken to other shareware authors about this, and they have ensured me that this method seems quite reliable after all.

To register print out and fill in the supplied registration form and put it in an envelope with the money and send it to me. If you want to register by EMail please contact me first before sending any money.

When I get your registration request and the money I will try to process the registration as fast as possible. Depending on the mail services and the number of registrations the delivery of the keyfile may take up to a month.

Once again please keep in mind that the key file is personal, and spreading it is a crime, even if you just give it to the closest of your friends. The name, address and serial number information is encoded in the key, and it is very easy to find out from where a keyfile has originated. If you do give out copies of your keyfile and I find it out you will lose the right to use the program, and your keyfile will be worthless for all future versions of Atlantis.

When you get your key file you should copy it to either

1) the path described by the KEYPATH environment variable (recommended)
2) S:
3) the program directory

The key file will be searched for in these directories, in that order.

Company registrations.

Software companies can register for a special group license, which allows for all the employees of that company to use the program. The group licenses costs as 10 single-user registrations, see the price list below. This license is valid as long as the user is the employee of the company. If the employee is fired or quits, that employee may not use the same license anymore. Since the license is personal, it is not tied to any particular machine. Instead, the company is considered a "logical" person, and every employee of that company is a "representative" of that "logical" person. This allows the employee of the company to use a copy of the license anywhere, for example at home, as long as he still is an employee of the license-holding company.

The company (group) registration fee is

```
      SEK 1600,-
      FIM 1200,-
      US$  200,-
      UKP  150,-
      DM   300,-
```

```
Aquadextrous, adj.:
    Possessing the ability to turn the bathtub faucet on and off
    with your toes.
                -- Rich Hall, "Sniglets"
```

1.6 Atlantis - Requirements

Requirements

To install and run Atlantis you should have (at least)

* 512 Kb free ram (mostly because MUI uses a lot of memory)
* AmigaOS 2.04 (V37)
* A few hundred k:s of harddrive space
* MUI version 3.3 (not included)
* A positive attitude.

There is also a special 68020 optimized version of the program included in the archive. Naturally, this version requires an 68020 or better processor.

The program also needs the MUI external custom class Icon.mcc (included) by Russell Leighton, see the Disclaimer section for more information.

```
Bagdikian’s Observation:
    Trying to be a first-rate reporter on the average American
    newspaper is like trying to play Bach’s "St. Matthew Passion"
    on a ukelele.
```

1.7 Atlantis - Installation

Installing Atlantis

Installation is very easy. The package contains a standard Installer-script that creates a program drawer called "Atlantis" and copies all files there. You will be asked which version of the program to install. The script will also add a Atlantis: assign to S:User-Startup.

Military intelligence is a contradiction in terms.
GROUCHO MARX

1.8 Atlantis - Features

This is Atlantis

* Only uses operating system routines for the graphic rendering, so it should be future-compatible
* Unlimited number of editing windows
* Uses MUI for a flexible and powerful GUI
* Overview mode
* Display cache that speeds up the display (not in Overview mode)
* True patterned flood-fill
* Unlimited map size (registered users only!)
* Supports VMM for very large levels (I have tried a 8000 x 4000 blocks level and it worked just fine!)
* Variable block sizes from 8x8 to 64x64 pixels (tell me if you need more)
* Adapts to low-memory situations by disabling various features
* "Fast cursor" option for slower machines
* User selectable screenmode (V38+)
* Localized (V38+)
* Extensive ARexx port for ’smart’-scripts (for making mazes etc.)
* Quickly jump to important places using marks
* Multiple Undo:s
* Uses file-notify on the block file: If you modify the blocks with some drawing program, Atlantis immediately updates its own representation of the blocks.
* Commodities support: Iconify, Disable, Enable and Remove Atlantis with the Exchange program.

The restrictions of the unregistered version is:

* You can not edit bigger levels than 80 x 40 blocks
* You can’t use project files, which can be very helpful for remembering the dependancies and sizes of the files in the project.
* An annoying requester pops up every 10 minutes asking you to register

```
Command, n.:
    Statement presented by a human and accepted by a computer in
    such a manner as to make the human feel as if he is in control.
```

1.9 Atlantis - How to use

How to use Atlantis

Atlantis can be started from either Workbench or CLI. It uses no tooltypes or arguments. When you start the program you are presented with the project requester:

```
 +-----------------------------------------------------------------+---+---+
 |Atlantis game map editor ©1997 Staffan Palmroos                  |   |   |
 +-----------------------------------------------------------------+---+---+
 | \ /I|D +----------------------------------------+----------------------+|
 |  V o|  |             Registered to              |        Atlantis      ||
 |   L"/  |            Staffan Palmroos            |          V1.1        ||
 |  /|'|. |            Serial Nr 000001            |  © Staffan Palmroos  ||
 | `-' `- +----------------------------------------+----------------------+|
 | .---------------------------- Files ----------------------------------. |
 ||               +-------------------------------------------------+---+ ||
 || Project Name: |                                                 | F | ||
 ||               +-------------------------------------------------+---+ ||
 ||               +-------------------------------------------------+---+ ||
 ||     Map file: |                                                 | F | ||
 ||               +-------------------------------------------------+---+ ||
 ||               +-------------------------------------------------+---+ ||
 ||   Block file: |                                                 | F | ||
 ||               +-------------------------------------------------+---+ ||
 | `---------------------------------------------------------------------' |
 | .----------- Map size ------------.  .---------- Block size ----------. |
 ||   +------------+   +------------+ ||   +-----------+    +-----------+ ||
 || X:|            | Y:|            | || X:|           |  Y:|           | ||
 ||   +------------+   +------------+ ||   +-----------+    +-----------+ ||
 | `---------------------------------'  `--------------------------------' |
 | ----------------------------------------------------------------------- |
 | +----------------------+                       +----------------------+ |
 | |          Go          |                       |        Cancel        | |
 | +----------------------+                       +----------------------+ |
 +----------------------------------------------------------------------+--+
 +----------------------------------------------------------------------+--+
```

Picture

In the top left corner you see a symbol describing the current registration status of Atlantis. In this case the symbol is a King, which indicates a VIP registration. VIP registrations are special registrations which you can not buy. You can only earn a VIP registration, for example by translating the program to your language. Contact me for more information. There is however no difference between a Normal and a VIP registration in terms of performance, it is just more honourable to have a VIP registration.

A normal registration is indicated with a famous cartoon character, and a company (group) registration is illustrated with a dollar bill. If you are not registered at all the word "DEMO" will be shown here. For more information on registering look in the Registration section.

Below the registration symbol there are some string gadgets that should be filled in:

Project name: This feature is only available to registered users. Here you can enter the name of an project file. Project files are used for remembering the size of the map, the size of the building blocks, which block file to use for the map and the set marks (see later). In later versions of Atlantis you will be able to work with several maps at the same time, then the project files will become handy to remember which block file the different maps uses. If you type a filename that cannot be found the program will ask you if you want to create a new project. If your answer is yes the name you typed in will be used for the new project.

Map file: Here you enter the name of the map you want to edit. This field will be filled in automatically if you selected a valid project file in the field above. If you type a filename that cannot be found the program will ask you if you want to create a new map. If your answer is yes the name typed in will be used for the name of the new map.

Block file: In this field you tell the program the name of the IFF picture file that contains the blocks for the map. This field must of course be filled in correctly. If you selected a project file in the corresponding field this field will be set automatically.

Note that you can use the small gadgets marked with an ’F’ in the picture to get a standard system file requester where you can click on a filename using the mouse.

When you have filled in the filenames you should fill in the size of the map and the blocks. The minimun map size is 2 x 2 blocks, and in the unregistered version the maximum size is 80 x 40 blocks. The size of the blocks range from 8x8 to 64x64 pixels. I think this should suffice, but in case anyone needs more (or less) mail me and I’ll change it. The most common sizes used are 16x16 and 32x32.

Now, when you have filled in all the fields correctly press "Go" to go to the main screen.

```
 +---+--------------------------------------------+---+---+-+---+----------+
 |   | DemoMap.Map                                |   |   |_|   |Plot      |
 +---+--------------------------------------------+---++--+ +---++----+----+
 |                                                     |##| | P  | R  | B  |
 |                                                     |##| |    |    |    |
 |                                                     |##| +----+----+----+
 |                                                     |  | | L  | C  | F  |
 |                                                     |  | |    |    |    |
 |                                                     |  | +----+----+----+
 |                                                     |  | | SM | CM | U  |
 |                                                     |  | |    |    |    |
 |                                                     |  | +--+-++--++-+--+
 |                                                     +--+ | 0| 1| 2| 3| 4|
 |                                                     |/\| +--+--+--+--+--+
 |                                                     +--+ | 5| 6| 7| 8| 9|
 |                                                     |\/| +--+--+--+--+--+
 +---+---+-------------------------------------+---+---+--+                |
 | V | O |########                             | < | > |  |                |
 +---+---+-------------------------------------+---+---+--+                |
 |                                                                         |
 +---+---------------------------------------------------------+---+---+   |
 |   |DemoBlocks.iff                                           |   |   |   |
 +---+---------------------------------------------------------+---+---+   |
 |                                                                  |##|   |
 |                                                                  |##|   |
 |                                                                  |  |   |
 |                                                                  |  |   |
 |                                                                  |  |   |
 |                                                                  +--+   |
 |                                                                  |/\|   |
 |                                                                  +--+   |
 |                                                                  |\/|   |
 +----------------------------------------------------------+---+---+--+   |
 |###################################################       | < | > |  |   |
 +----------------------------------------------------------+---+---+--+---+
```

When the main screen opens three windows initially opens on it. On the top left you see a "map window", on the top right you see the tool window. On the lower half of the screen you see the block window.

Map windows

The map windows are considered as views of the map. You can have any number of map windows, but there must be at least one. When you close the last map window the program will exit (if the map has changed you will be asked first).

This kind of window have two special border gadgets, in the picture above marked as V and O. The gadget marked V is used to open up a new map window, that can show another part of the map. When you press this gadget you will get (if memory permits) a new map window showing the same view of the map as the old window. All map windows are independant of each other, so you can close the old window, or make it view another part of the map.

The gadget marked O is used to get the map window in Overview mode. The window will then be redrawn with a set of blocks 1/16:th in size, showing a larger part of the map. This is why you can't use blocks with size less than 8x8 pixels: the overview-blocks will then be 2x2 pixels big, and if you make them smaller (1x1) you can't see what your map looks like (it is hard enough with 2x2 blocks). If you press the V gadget in a window that is in overview mode you won't get another window in overview mode. Instead, your cursor turns into a fixed size square representing the boundaries of the new window. With this cursor you select an area of the map that the new window will show. This makes it possible to "zoom in" on an area: You can have one window in the background in overview mode, and then open new map windows of different parts of the map directly from the first one.

The overview mode is rendered with a separate set of blocks, which are generated after the block file has been read. If there isn't enough memory for the overview blocks the overview button in all map windows will be disabled.

The block window

On the lower half of the screen you see the block window. This window contains the palette of blocks you use to create the map. This window is a direct view of the block file. In the picture you can see that one of the blocks in the block window has a square around it. This square indicates the contents of macro 0. This macro is reserved for the block window; macro 0 is always equivalent to the blocks indicated in the block window. You can select several blocks in the block window by pressing and holding the left mouse button and then drag the mouse pointer around. There must always be a block window on the screen. If you click on the close gadget in the block window the program will exit (if the map has been changed you will be asked first).

To move within the map and block windows you can use the scroll bars on the window borders or use the cursor keys. If you combine the cursor keys with a Shift key you move one window at a time, and if you hold down an Alt key with the cursor keys you move to the start/end of the map/blocks.

The tool window

In the top right corner of the screen you see the tool window. The title bar of the tool window shows you which paint mode is currently active, in the picture it is the "Plot" mode. Below the title bar you have 9 large gadgets. You can edit the graphics for these gadgets yourself if you want to. The graphics for these gadgets are taken directly from the icons with the corresponding names in the graphics directory. The nine large gadgets in the tool window have the following functions:

P - Plot — This is the default paint mode. Just press and hold the left mouse button to draw. This and the fill tool are the only drawing tools that uses the whole macro. The other drawing tools only use the top left block of the macro.

R - Rectangle — With this tool you draw rectangles. Press and hold the left mouse button to select the first corner of the wanted rectangle. Then drag the mouse to the position of the last corner and release the left mouse button. The first corner of the rectangle doesn't have to be the top left corner, you can draw the rectangle any way you want.

B - Box — This is almost the same as the rectangle tool, except that you with this tool draw a filled rectangle. Just as with the rectangle you select the first corner by pressing the left mouse button and holding it down while moving the mouse to the position of the last corner, where you release the button.

L - Line — This is the line tool, with which you draw lines of blocks. Press and hold the left mousebutton to mark the first end of the line, then drag the mouse to the other end of the line and release the button.

C - Copy

With this function you can copy parts of the map to the brushes, or "macros" as they are called in Atlantis. You have 10 macros to draw with. Two of them have special properties, see below. If you try to copy from a map window to macro 0, the current macro will change to the first unused macro (which is macro 9 if all the others are used.)

F - Fill

This is a pattern-flood-fill tool. You can outline an area with the other drawing tools and then fill it with this one. The current macro will be used for the fill pattern. This and the plot tool is the only drawing tools that use the whole macro. The other drawing tools only use the top left block of the macro. Note that this function clears the undo-buffer, so you can not undo changes after a fill operation. See the text about the undo-gadget below for more information.

SM - Set Mark

With the help of "marks" you can point out important places in the map. Select this tool and click on a block in the map where you want a mark. Unless you have already used up all your 9 marks yo u should see a number super-superimposed on the block you clicked on. This is the number of the mark. When you have placed your marks where you want them you can use them for the functions in the menus. In fact, the functions in the menus currently ONLY works with marks, but this will probably change in the future. Note that you can not choose which mark to set. The first free mark will be used instead, until there are no free (unset) marks left. Note also that the marks are not shown in overview-mode. This is because the blocks in overview-mode are (sometimes much) smaller than the font used to indicate the marks.

CM - Clear Mark

This function is used to remove the marks you have set with the function above. Just click on a block with a mark set and the mark will be removed.

U - Undo

This function is used to take back changes you have done recently. You can use this function many times. The number of undo:s you can make depends on a special undo buffer; the larger the buffer the more undo:s you can do. The size of the undo buffer can be set in the preferences window. The function of this buffer is to remember the changes you make in the map. When you draw a block in the map somewhere, the previous block on that position is put in the buffer along with its coordinates. If you draw with a macro consisting of several blocks, all the blocks that will be changed is saved to this buffer as a group. They can then be retrieved back in one go. Note that the undo buffer will be cleared when you make a flood-fill, since the buffer in most cases will not be large enough to remember all changes the flood-fill does.

With the 10 gadgets below the tool gadgets marked 0 - 9 you select which macro to draw with. You can also select a macro with the numeric keys. Two of these macros have special properties. Macro 0 is reserved for the blocks marked in the block window. When you select block(s) in the block window macro 0 will become the current macro, unless you have the copy tool mentioned above activated.

Macro 9 is considered a 'scratch' macro. This means that it will be selected whenever a free macro is required (for example in an ARexx script) but all macros are used.

The menus

In the menus you have some additional useful functions.

Project

```
* New           - Start all over with a new project.

* Save...       - Save the map. A file requester will pop up asking you to
                  select a file name,  or use the old name.  Currently the
                  maps are saved uncompressed,  but in the future you will
                  be able to use some compression method.

* Change Dim.   - Change the dimensions of the  map.  The  map will not be
                  lost, but it will be  cropped  if  the  new  map size is
                  smaller than the original.

* About         - Shows version & copyright information and free memory.

* About MUI     - Brings up the MUI systems own about-requester.

* Quit          - Quits Atlantis (but you don't want to do that, do you?)
```

Misc.

```
* Move Area     - Move an area of blocks to another place. You place marks
                  at the significant positions  (the upper left corner and
                  the lower right corner of the area to move,  and the new
                  position of the  upper  left  corner)  and fill in which
                  marks to use in the requester.

* Jump to Mark  - This is a good way to  move  within a large map: Place a
                  mark in the map and  select this function from the menu.
                  This will bring up a  requester  where  you can select a
                  mark to jump to.  The  gadgets for the marks that is not
                  set will be disabled,  so  you can only choose set marks
                  or 'cancel'.

* Preferences   - Brings up the preferences window. The preferences window
                  is described in the Preferences section.

* Tool Window   - Shows or hides the tool window.
```

THE LESSER-KNOWN PROGRAMMING LANGUAGES #12: LITHP

This otherwise unremarkable language is distinguished by the absence of an "S" in its character set; users must substitute "TH". LITHP is said to be useful in protheththing lithtth.

1.10 Atlantis - The Preferences Window

The Preferences Window

The preferences window lets you customize the behaviour of Atlantis. The name of the preferences file is "Atlantis.prefs". If you press ’use’ the preferences file is saved in the ENV: directory, and if you press ’save’ the file is saved both in ENV: and ENVARC:. Below is a brief description of the gadgets in the preferences window.

At the top there are some flags:

Query Autosave
If this flag is activated, the program will when it is time to save ask you if you want to do it.

Keep Backup
if this flag is activated the program will keep a backup of the map when saving. The old file will be renamed with the extension ".bak". If you have the storage space this flag is very recommended.

Create Icon
If this flag is activated the program will create icons for the project and map files. Atlantis will search for default icons in ENV:Sys/def_AtlantisProject.info and ENV:Sys/def_AtlantisMap.info. If Atlantis can’t find these icons, the default system project icons will be used instead.

Fast Cursor
Normally the block cursor is drawn with the current macro for "WYSIWYG" behaviour. However, if you have a slow machine you might want to activate this flag to make the cursor be a rectangle representing the macro. This is a lot faster since it just involves drawing four lines instead of plotting the whole macro.

Display Cache
The display cache is a buffer used for speeding up the redrawing of the map windows. If you are short of memory you might want to turn this feature off to save some.

In the middle of the window there are some sliders:

Autosave
Here you set the autosave delay. The delay is measured in minutes, and range from 1 to 60 minutes. If you don’t want the auto-save feature set the delay to 0.

Undo buffer

Here you can set the size of the undo buffer. The size is measured in number of blocks that can be remembered. Each block in the undo buffer requires 6 bytes. The smallest buffer is 256 blocks (~1.5 Kbytes) and the maximum size is 8192 blocks (~50 Kbytes).

In the bottom of the window you can set the screenmode you want to work in. This option is only available to users with AmigaOS 2.1 or later:

Use the block file screenmode

This option makes Atlantis use the screenmode that was used when the block file was created. This is the only possible choice if you have AmigaOS 2.04.

Screenmode

Here you can override the block file screenmode. Press the gadget to the right to get a standard screenmode-requester where you can select one of the available screenmodes.

The trouble with doing something right the first time is that nobody appreciates how difficult it was.

1.11 Atlantis - The Arexx Interface

The ARexx Interface

Atlantis has an extensive ARexx interface. With it you can make your own advanced functions, like for example a maze-generator. Note that the syntax for the ARexx commands might change in the future, since I am not satisfied with the way MUI handles ARexx. Because of this I have included the VERSION command which returns the version of the ARexx interface so that you can make version dependant Arexx scripts.

Here is a list of Arexx commands that Atlantis recognizes:

```
ASKUSER
BOX
CLEARMARK
FILL
GETFREEMACRO
GRABFROMMAP
GRABBLOCKS
HELP           *
HIDE           *
INFO           *
INFORM
KILLMACRO
LINE
```

```
   MAPHEIGHT
   MAPWIDTH
   MARKX
   MARKY
   NEXTSETMARK
   NUMSETMARKS
   PLOT
   QUIT           *
   RECT
   SETMARK
   SHOW           *
   SLEEP
   UPDATE
   USEMACRO
   VERSION
   WAKEUP
```

Commands marked with an asterisk (*) are built-in commands in MUI, see the disclaimer section.

Unless otherwise stated in the individual descriptions of the commands, the commands must not be called before the main screen is opened! If you do call a command that needs the main screen the command will fail with the code RC_FATAL (20) in the ARexx variable RC.

Note also that the descriptions of the commands does not cover the errors generated by MUI. For more information of those errors refer to the MUI documentation.

That's the thing about people who think they hate computers. What they really hate is lousy programmers.
        - Larry Niven and Jerry Pournelle in "Oath of Fealty"

1.12 Atlantis - The Arexx Interface - ASKUSER

NAME
  ASKUSER -- Ask the user a question (V1)

USAGE
  ASKUSER <answers> <question>

DESCRIPTION
  This command is used to ask the user a question. The <answers> string contains the possible answers the user can choose from, separated by vertical bars ('|'), and the <question> string contains the actual question. Note that if you want to have spaces in the <answers> string you need to enclose the string in both single and double quotes. The reason for this is a combination of ARexx and MUI:

When ARexx sends the command string to MUI for parsing, it concatenates the two argument strings into one string. Thus, the call

```
  ASKUSER "ONE TWO" "THREE"
```

becomes ASKUSER ’ONE TWO THREE’ when ARexx passes it to MUI. When MUI gets the argument string MUI passes it on to dos.library/ReadArgs() for the actual argument parsing. ReadArgs() will then see the first space in the argument string as the separator for the two substrings. To prevent this, you must somehow pass on the double quotes to ReadArgs(). This can be done by enclosing the first string in single quotes. The call

```
  ASKUSER ’"ONE TWO"’ "THREE"
```

becomes ASKUSER ’"ONE TWO" THREE’, and the problem is solved!

Note also the argument numbering: The first argument is number 1, the next is number 2 and so on until the last argument which becomes argument 0. This behavior is inhereted from intuition.library/EasyRequest(). You can have a maximum of 10 arguments in the <answer> string. If you have more, the result will always be 0.

INPUTS

```
  <answers>  A string  containing  the  possible  answers  to  the  question,
             separated by vertical bar (’|’) characters.
  <question> A string containing the question asked.
```

RESULT

The special ARexx variable RESULT is set with the argument number that the user selected, or 0 if the number of arguments exceeded 10.

EXAMPLE

```
  /**/

  OPTIONS RESULTS
  ADDRESS ATLANTIS

  ASKUSER ’"Excellent|Very Good|Not so good|Bad|Terrible"’,
           "How are you feeling today?"

  IF RESULT=1 INFORM "That’s great!"
  IF RESULT=2 INFORM "That’s nice"
  IF RESULT=3 INFORM "Oh?"
  IF RESULT=4 INFORM "I’m sorry to hear that!"
  IF RESULT=0 INFORM "It serves you right!"
```

SEE ALSO

dos.library/ReadArgs(), intuition.library/EasyRequest(), INFORM

NOTES

This command can be used before the main screen is opened.

1.13 Atlantis - The Arexx Interface - BOX

NAME

BOX -- Draw a filled rectangle (V1)

```
USAGE
  BOX <startX> <startY> <endX> <endY>

DESCRIPTION
  This command draws a  filled  rectangle  from  <startX>,<startY> to <endX>,
<endY> using the top left block  of  the  current  macro. If you want the box
filled with a pattern, combine this command with the FILL command.

INPUTS
  <startX>, <startY> The start coordinate of the area to be the box. Does not
                     have to be the top left corner of the area.

  <endX>, <endY>     The end coordinate  of  the area to be the box. Does not
                     have to be the lower right corner of the area.

RESULT
  This command does not return a value.  If  the  coordinates of the area are
outside the map  the  command  will  fail with the code RC_ERROR in the ARexx
variable RC.

EXAMPLE
  /**/

  ADDRESS ATLANTIS

  USEMACRO      /* Use the same macro as the user */
  BOX 1 1 4 4
  UPDATE        /* Refresh the display */

SEE ALSO
  FILL, RECT
```

1.14 Atlantis - The Arexx Interface - CLEARMARK

```
NAME
  CLEARMARK -- Remove a mark from the map (V1)

USAGE
  CLEARMARK <mark>

DESCRIPTION
  Removes a mark previously set with SETMARK  or the SetMark tool. If <mark>
is not set nothing happens.

INPUTS
  <mark> The number of the mark that should be removed.  Must be in the range
         1..9.

RESULT
  This command does not return a value.  If <mark> is outside the valid range
the command will fail with the code RC_ERROR in the ARexx variable RC.

EXAMPLE
  /**/
  ADDRESS ATLANTIS

  CLEARMARK 4  /* Remove mark 4 */
  UPDATE        /* Refresh the display */
```

SEE ALSO
  SETMARK

1.15 Atlantis - The Arexx Interface - FILL

NAME
  FILL -- Flood fill an area with pattern (V1)

USAGE
  FILL <Xpos> <Ypos>

DESCRIPTION

This command flood-fills an area previously outlined with the other tools. The command uses the current macro for the fill pattern. The implementation of the fill function uses a separate buffer which is relatively big. If Atlantis can't allocate memory for the fill buffer it will disable the fill function and go on. A requester will pop up in this case to tell you what's going on.

INPUTS
  <Xpos>, <Ypos> The position in the map to start the fill.

RESULT

This command does not return a value. If the coordinates of the start position are outside the map the command will fail with the code RC_ERROR in the ARexx variable RC.

EXAMPLE

```
  /**/

  ADDRESS ATLANTIS

  USEMACRO      /* Use the same macro as the user */
  FILL 1 2      /* Start a fill operation at position (1,2) */
  UPDATE        /* Refresh the display */
```

SEE ALSO
  RECT, USEMACRO, UPDATE

NOTES

This command clears the undo buffer, so the user can not undo the fill operation. If there is no fill buffer, this command will fail with the code RC_WARN in the ARexx variable RC.

1.16 Atlantis - The Arexx Interface - GETFREEMACRO

NAME
GETFREEMACRO -- Return the first free macro (V1)

USAGE

```
  GETFREEMACRO
```

DESCRIPTION
This command returns the number of the first unused macro. If all macros are in use this command will return 9, which is considered a ’scratch’ macro and therefore always unused. This command is good when your script need a temporary macro. When you don’t need the temporary macro anymore you should call the KILLMACRO function to make the macro available to the GETFREEMACRO command again.

INPUTS
None.

RESULT
The special ARexx variable RESULT is set with the unused macro. If no other macros are unused, the RESULT variable will be 9.

EXAMPLE

```
  /**/

  OPTIONS RESULTS
  ADDRESS ATLANTIS

  GETFREEMACRO      /* Request a temporary macro */
  SAY 'Got macro ' RESULT

  USEMACRO RESULT      /* Use my temporary macro */
  GRABFROMMAP 0 0 1 1  /* Get 4 blocks from the map */

  PLOT 2 2             /* Plot them somewhere */
  UPDATE               /* Refresh the display */

  KILLMACRO RESULT     /* The macro is free for temporary use again */
```

SEE ALSO
KILLMACRO, USEMACRO

NOTES
The macro is actually not allocated in any way; if you call this command two times in a row, both calls will return the same macro.

1.17 Atlantis - The Arexx Interface - GRABFROMMAP

NAME
GRABFROMMAP -- Copy an area from the map to a macro (V1)

USAGE

```
  GRABFROMMAP  <startX> <startY> <endX> <endY>
```

DESCRIPTION

With this command you can copy a part of the map to the current macro for use later. That is great when composing complex objects that is used in several places.

INPUTS

<startX>, <startY> The start coordinate of the area to be copied. Does not have to be the top left corner of the area.

<endX>, <endY> The end coordinate of the area to be copied. Does not have to be the lower right corner of the area.

RESULT

This command does not return a value. If the coordinates of the area are outside the map the command will fail with the code RC_ERROR in the ARexx variable RC.

EXAMPLE

```
  /**/

  OPTIONS RESULTS
  ADDRESS ATLANTIS

  GETFREEMACRO         /* Get a temporary macro */
  USEMACRO RESULT      /* Use it */

  GRABFROMMAP 1 1 4 4  /* Get 9 blocks from the map */
  PLOT 5 5             /* Plot them somwhere
  UPDATE               /* Refresh the display */

  KILLMACRO RESULT     /* Make macro free for temporary use again */
```

SEE ALSO

GRABBLOCKS

NOTES

The current macro must not be macro 0, since macro 0 is reserved for the block window. If the current macro is macro 0, the command will fail with RC_ERROR in the ARexx variable RC.

1.18 Atlantis - The Arexx Interface - GRABBLOCKS

NAME

GRABBLOCKS -- Copy an area from the block window to a macro (V1)

USAGE

```
  GRABBLOCKS <startblock> <endblock>
```

DESCRIPTION

With this command you copy an area of blocks from the block file to the current macro. You enter the block numbers of two opposite corners of the area. The blocks in the block file are numbered from left to right, from top to bottom, making the top left block number 0, the block next to the left number 1 and so on until the end of the first row of blocks. The next number is then the first block on the second row.

INPUTS

```
  <startblock> The number of the block in the first corner  of the area. Does
               not have to be the top left corner.

  <endblock>   The number of the block in the second corner of the area. Does
               not have to be the lower right corner.
```

RESULT

This command does not return a value. If the block numbers are outside the valid range (0 .. Number of blocks in block file) the command will fail with the code RC_ERROR in the ARexx variable RC.

EXAMPLE

```
  /**/

  OPTIONS RESULTS
  ADDRESS ATLANTIS

  /* Assume there are 10 blocks per row in the block file */

  GETFREEMACRO         /* Get a temporary macro */
  USEMACRO RESULT      /* Use it */

  GRABBLOCKS 0 11      /* Get the 4 top left blocks from the block file */
  PLOT 5 5             /* Plot them somwhere
  UPDATE               /* Refresh the display */

  KILLMACRO RESULT     /* Make macro free for temporary use again */
```

SEE ALSO

GRABFROMMAP

NOTES

The current macro must not be macro 0, since macro 0 is reserved for the block window. If the current macro is macro 0, the command will fail with RC_ERROR in the ARexx variable RC.

1.19 Atlantis - The Arexx Interface - HELP

NAME

HELP -- Write a list of ARexx commands to a file (MUI)

USAGE

```
  HELP <file>
```

DESCRIPTION

This command writes a list of all ARexx commands to <file>. This is a built-in command in the MUI system. Consult the documentation for MUI for more information.

INPUTS

<file> The name of the file to create.

RESULT

This command does not return a value.

```
EXAMPLE
  /**/

  ADDRESS ATLANTIS

  HELP Atlantis:Arexx_Commands.txt
```

1.20 Atlantis - The Arexx Interface - HIDE

```
NAME
  HIDE -- Iconifies Atlantis (MUI)

USAGE
  HIDE

DESCRIPTION
  This command iconifies Atlantis. The main screen closes and an icon appears
on the Workbench.  Double-clicking on the icon or executing the SHOW command
brings back the main screen.

  This is a built-in command in the MUI system. Consult the documentation for
MUI for more information.

INPUTS
  None.

RESULT
  This command does not return a value.

EXAMPLE
  /**/

  ADDRESS ATLANTIS

  HIDE  /* Close the main screen */
  SHOW  /* Open it again */

SEE ALSO
  SHOW
```

1.21 Atlantis - The Arexx Interface - INFO

```
NAME
  INFO -- Return a string of information about Atlantis (MUI)

USAGE
  INFO <item>

DESCRIPTION
  INFO returns a string of information about the program according to <item>.
Valid values for <item> are:
```

TITLE returns the title of the program, in this case "ATLANTIS"

AUTHOR returns the name of the author of the program.

COPYRIGHT returns copyright information.

DESCRIPTION returns a short description of the program.

VERSION returns a version string.

BASE returns the name of the ARexx port the program uses. This is a quite useless parameter, since you have to know the name of the ARexx port to be able to make this call in the first place!

SCREEN normally returns the name of the public screen. In Atlantis this parameter shouldn't return anything, since Atlantis uses its own screen which is not public.

This is a built-in command in the MUI system. Consult the documentation for MUI for more information.

INPUTS
<item> Describes what the ARexx script wants to know.

RESULT
The special ARexx variable RESULT is filled with the requested information.

EXAMPLE

```
  /**/

  OPTIONS RESULTS
  ADDRESS ATLANTIS

  INFO AUTHOR

  SAY 'Atlantis was programmed by' RESULT
```

1.22 Atlantis - The Arexx Interface - INFORM

NAME
INFORM -- Inform the user of an event (V1)

USAGE
INFORM <text>

DESCRIPTION
This command is used to inform the user about something. The command brings up an requester containing <text> and an OK button. The ARexx script will be halted until the user clicks on the button. Calling this command is almost the same as calling

```
  ASKUSER "OK" <text>
```

except that this command doesn't return a value.

```
INPUTS
  <text> A string containing the message to the user

RESULT
  This command does not return a value.

EXAMPLE
  /**/

  ADDRESS ATLANTIS

  INFORM "I'm done!"

SEE ALSO
  ASKUSER

NOTES
  This command can be used before the main screen is opened.
```

1.23 Atlantis - The Arexx Interface - KILLMACRO

```
NAME
  KILLMACRO -- Free the memory used by a macro and mark it unused (V1)

USAGE
  KILLMACRO <macro>

DESCRIPTION
  This command is used to  free  the  memory a macro occupies. The macro will
also be marked as unused so that the GETFREEMACRO command will recognize this
macro. If your script uses a macro  for  temporary  use  you should call this
command before the  script  terminates,  so  that  it can be used by the next
ARexx script that the user invokes.  You  should not however kill macros that
the user has defined; this will probably  be  very annoying to the user. This
command is  currently  the  only  way  to  free the resources used by a macro
(except for quitting the program, of course).

INPUTS
  <macro> The macro to free. Must be in the range 1..9

RESULT
  This command does not return a value.  If  <macro>  is  out  of  range, the
command will fail with the code RC_ERROR in the ARexx variable RC.

EXAMPLE
  /**/

  OPTIONS RESULTS
  ADDRESS ATLANTIS

  GETFREEMACRO         /* Get a temporary macro */
  USEMACRO RESULT      /* Use it */

  GRABFROMMAP 0 0 1 1  /* Get the 4 top left blocks from the block file */
  PLOT 5 5              /* Plot them somwhere
  UPDATE                /* Refresh the display */

  KILLMACRO RESULT      /* Make macro free for temporary use again */
```

SEE ALSO
GETFREEMACRO

1.24 Atlantis - The Arexx Interface - LINE

NAME
LINE -- Draw a line of blocks (V1)

USAGE

```
  LINE <startX> <startY> <endX> <endY>
```

DESCRIPTION

This command draws a line from <startx> , <starty> to <endx> , <endy> using the top left block of the current macro.

INPUTS

```
  <startX>, <startY> The start coordinate of the  line.  Does  not have to be
                     the top left coordinate.

  <endX>, <endY>     The end coordinate of the line.  Does not have to be the
                     lower right coordinate.
```

RESULT

This command does not return a value. If the coordinates of the line are outside the map the command will fail with the code RC_ERROR in the ARexx variable RC.

EXAMPLE

```
  /**/

  ADDRESS ATLANTIS

  USEMACRO      /* Use the same macro as the user */
  LINE 1 1 4 4
  UPDATE        /* Refresh the display */
```

SEE ALSO
USEMACRO, UPDATE

1.25 Atlantis - The Arexx Interface - MAPHEIGHT

NAME
MAPHEIGHT -- Get the height of the map (V1)

USAGE

```
  MAPHEIGHT
```

```
DESCRIPTION
  This command and the MAPWIDTH command are used to ask Atlantis the size of
the map. This can be handy for  making  sure  that the script will not try to
draw outside the map causing run-time errors.  If the main screen is not open
the command returns 0. This can be used to test if the main screen is open.

INPUTS
  None.

RESULT
  The height of the map is returned in the ARexx variable RESULT.

EXAMPLE
  /**/

  OPTIONS RESULTS
  ADDRESS ATLANTIS

  MAPWIDTH; W = RESULT
  MAPHEIGHT; H = RESULT

  SAY 'The size of the map is' W 'x' H

SEE ALSO
  MAPWIDTH

NOTES
  This command can be used  before  the  main screen is opened. In this case,
the command returns 0.
```

1.26 Atlantis - The Arexx Interface - MAPWIDTH

```
NAME
  MAPWIDTH -- Get the width of the map (V1)

USAGE
  MAPWIDTH

DESCRIPTION
  This command and the MAPHEIGHT command  are  used to ask Atlantis the size
of the map. This can be handy for making sure that the script will not try to
draw outside the map causing run-time errors.  If the main screen is not open
the command returns 0. This can be used to test if the main screen is open.

INPUTS
  None.

RESULT
  The width of the map is returned in the ARexx variable RESULT.

EXAMPLE
  /**/

  OPTIONS RESULTS
  ADDRESS ATLANTIS
  MAPWIDTH; W = RESULT
  MAPHEIGHT; H = RESULT

  SAY 'The size of the map is' W 'x' H
```

```
SEE ALSO
  MAPHEIGHT

NOTES
  This command can be used  before  the  main screen is opened. In this case,
the command returns 0.
```

1.27 Atlantis - The Arexx Interface - MARKX

```
NAME
  MARKX -- Get the X coordinate of a mark (V1)

USAGE
  MARKX <mark>

DESCRIPTION
  This command returns the X coordinate  of  <mark>. If <mark> is not set, -1
is returned.

INPUTS
  <mark> The number of the mark who's X coordinate you want to know.  Must be
         in the range 1..9.

RESULT
  The ARexx variable RESULT is set with the X coordinate of <mark>. If <mark>
is outside the valid range  the  command  will fail with the code RC_ERROR in
the ARexx variable RC.

EXAMPLE
  /**/

  OPTIONS RESULTS
  ADDRESS ATLANTIS

  MARKX 1; X = RESULT
  MARKY 1; Y = RESULT

  SAY 'Mark 1 is at position' X ',' Y

SEE ALSO
  MARKY
```

1.28 Atlantis - The Arexx Interface - MARKY

```
NAME
  MARKY -- Get the Y coordinate of a mark (V1)
```

USAGE

```
  MARKY <mark>
```

DESCRIPTION

This command returns the Y coordinate of <mark>. If <mark> is not set, -1 is returned.

INPUTS

```
  <mark> The number of the mark who’s Y coordinate you want to know.  Must be
         in the range 1..9.
```

RESULT

The ARexx variable RESULT is set with the Y coordinate of <mark>. If <mark> is outside the valid range the command will fail with the code RC_ERROR in the ARexx variable RC.

EXAMPLE

```
  /**/

  OPTIONS RESULTS
  ADDRESS ATLANTIS

  MARKX 1; X = RESULT
  MARKY 1; Y = RESULT

  SAY 'Mark 1 is at position' X ',' Y
```

SEE ALSO

MARKX

1.29 Atlantis - The Arexx Interface - NEXTSETMARK

NAME

NEXTSETMARK -- Get the number of the next set mark (V1)

USAGE

```
  NEXTSETMARK [<mark>]
```

DESCRIPTION

NEXTSETMARK returns the number of the next set mark after <mark>. If there are no more set marks after <mark>, NEXTSETMARK returns 0. The parameter is optional, NEXTSETMARK without the parameter returns the first set mark.

This command can be used for scripts that require the user to have set some marks before invoking the script. This could for example be a script that generates a room by drawing a line between the set marks.

INPUTS

```
  <mark> Start looking for set marks after this one. Optional.
```

RESULT

The ARexx variable RESULT is set with the number of the first set mark after <mark>, or 0 if there are no more set marks after <mark>.

```
EXAMPLE
  /**/

  OPTIONS RESULTS
  ADDRESS ATLANTIS

  /* A program that draws lines between the set marks */

  SLEEP                            /* Don't let the user interfere */
  NUMSETMARKS; marks = RESULT      /* How many marks are set? */

  NEXTSETMARK; frommark = RESULT; /* Get the first set mark */
  firstmark = frommark

  MARKX frommark; xs = RESULT      /* Get the X coord. for the first mark */
  MARKY frommark; ys = RESULT      /* Get the Y coord. for the first mark */

  DO FOR marks
    NEXTSETMARK frommark           /* Get the next set mark */
    tomark = RESULT

    MARKX tomark; xe = RESULT      /* Get the X coord. */
    MARKY tomark; ye = RESULT      /* Get the Y coord. */

    LINE xs ys xe ye               /* Draw the line */

    xs = xe                        /* The current end mark... */
    ys = ye                        /* ...will be the next...   */
    frommark = tomark              /* ...start mark            */
  END

  MARKX firstmark;   xe = RESULT  /* Finish the loop */
  MARKY firstmark;   ye = RESULT

  LINE xs ys xe ye

  WAKEUP                           /* Give control to the user again */

SEE ALSO
  NUMSETMARKS

NOTES
  <Mark> itself does not have to be set.
```

1.30 Atlantis - The Arexx Interface - NUMSETMARKS

```
NAME
  NUMSETMARKS -- Get the number of set marks (V1)

USAGE
  NUMSETMARKS

DESCRIPTION
  This command  returns  the  number  of  set  marks. This can be useful when
iterating through the set marks, or just for making sure that there still are
```

unset marks.

INPUTS
  None.

RESULT
  The number of set marks is returned in the ARexx variable RESULT. Currently the maximum number of (set) marks is 9.

EXAMPLE

```
  /**/

  OPTIONS RESULTS
  ADDRESS ATLANTIS

  NUMSETMARKS               /* Get the number of set marks */
  IF (RESULT < 9) THEN DO   /* There are still unset marks */
    SETMARK 1 1             /* Place a mark somewhere */
    UPDATE                  /* Refresh the display */
    SAY 'Mark' RESULT 'set at position 1,1'
  END
  ELSE                      /* All marks are used up */
    SAY 'No unset marks left!'
```

SEE ALSO
  NEXTSETMARK, SETMARK

1.31 Atlantis - The Arexx Interface - PLOT

NAME
  PLOT -- Paste the contents of a macro on the map (V1)

USAGE
  PLOT <Xpos> <Ypos>

DESCRIPTION
  This command is the ARexx equivalent of the freehand Plot tool in the tool window. It pastes the current macro on the map with the top left corner of the macro on position <Xpos>, <Ypos>. If the macro will not fit entirely within the map it will be cropped, but if <Xpos>, <Ypos> is outside the map an error is issued. This means that you can over-step the map boundaries on the right and bottom side of the map, but not on the top or left side.

INPUTS
  <Xpos>, <Ypos> The position to place the top left corner of the macro.

RESULT
  This command does not return a value. If the position <Xpos>, <Ypos> is outside the map the command will fail with the code RC_ERROR in the ARexx variable RC.

EXAMPLE

```
  /**/

  ADDRESS ATLANTIS
  USEMACRO      /* Use the same macro as the user */
  PLOT 7 7      /* Plot the macro somewhere */
  UPDATE        /* Refresh the display */

SEE ALSO
  USEMACRO, UPDATE
```

1.32 Atlantis - The Arexx Interface - QUIT

```
NAME
  QUIT -- Quit Atlantis (MUI)

USAGE
  QUIT

DESCRIPTION
  This command makes Atlantis quit.  If  the map has changed the user will be
asked first. This is  a  built-in  command  in  the  MUI  system. Consult the
documentation for MUI for more information.

INPUTS
  None.

RESULT
  This command does not return a value.

EXAMPLE
  /**/

  ADDRESS ATLANTIS

  QUIT
```

1.33 Atlantis - The Arexx Interface - RECT

```
NAME
  RECT -- Draw a rectangle (V1)

USAGE
  RECT <startX> <startY> <endX> <endY>

DESCRIPTION
  This command  draws  a  rectangle from <startX>, <startY> to <endX>, <endY>
using the top left block of the current macro.  This command can successfully
be combined with the BOX and  the FILL commands to generate rooms in a map,
see the example.

INPUTS
  <startX>, <startY> The first coordinate  of  the  area to be the rectangle.
                     Does not have to be the top left corner of the area.
  <endX>, <endY>     The second coordinate  of  the area to be the rectangle.
                     Does not have to be the lower right corner of the area.

RESULT
  This command does not return a value.  If  the  coordinates of the area are
outside the map  the  command  will  fail with the code RC_ERROR in the ARexx
variable RC.

EXAMPLE
  /**/

  /* Make a room in the map */

  ADDRESS ATLANTIS

  /* Assume that macro 1 is block 0
   * Assume further that macro 2 contains the boundaries of the room, for
     example a block of bricks
   * Assume also that macro 3 contains the background pattern in the room
   */

  SLEEP               /* Don't let the user interfere */

  USEMACRO 1
  BOX 0 0 10 10       /* Clear a square */

  USEMACRO 2
  RECT 0 0 10 10      /* Make the boundaries of the room */

  USEMACRO 3
  FILL 1 1            /* Fill with the background pattern */

  WAKEUP              /* Give control to the user */

SEE ALSO
  BOX, FILL

NOTES
```

1.34 Atlantis - The Arexx Interface - SETMARK

```
NAME
  SETMARK -- Place a mark in the map (V1)

USAGE
  SETMARK <Xpos> <Ypos>

DESCRIPTION
  This command tries  to  place  a mark in the map. As you can see, you can't
select which mark to set. This is so that a script can not change a mark that
the user has set.  If there already are a mark at the position, the number of
that mark is returned.  If not, a new mark will be placed on the location and
the number of that mark  is  returned.  If there are no free marks left, 0 is
returned and no mark is set.  You  can  test if there are any free marks left
with the NUMSETMARKS command.
```

```
INPUTS
  <Xpos>, <Ypos> The position in the map to place a mark.

RESULT
  If successful the ARexx variable RESULT  is set with the number of the mark
at the position. If not, RESULT is set to  0.  If the position is outside the
map the command will fail with the code RC_ERROR in the ARexx variable RC.

EXAMPLE
  /**/

  OPTIONS RESULTS
  ADDRESS ATLANTIS

  NUMSETMARKS               /* Get the number of set marks */
  IF (RESULT < 9) THEN DO   /* There are still unset marks */
    SETMARK 1 1             /* Place a mark somewhere */
    UPDATE                  /* Refresh the display */
    SAY 'Mark' RESULT 'set at position 1,1'
  END
  ELSE                      /* All marks are used up */
    SAY 'No unset marks left!'

SEE ALSO
  CLEARMARK, NUMSETMARKS
```

1.35 Atlantis - The Arexx Interface - SHOW

```
NAME
  SHOW -- Uniconifies Atlantis (MUI)

USAGE
  SHOW

DESCRIPTION
  This command undos the HIDE command.  The  main screen opens again and the
icon disappears  from  the  Workbench  screen.  If  Atlantis is not iconified
nothing happens. This is  a  built-in  command in the MUI system. Consult the
documentation for MUI for more information.

INPUTS
  None.

RESULT
  This command does not return a value.

EXAMPLE
  /**/

  ADDRESS ATLANTIS

  HIDE  /* Close the main screen */
  SHOW  /* Open it again */
```

```
SEE ALSO
  HIDE
```

1.36 Atlantis - The Arexx Interface - SLEEP

```
NAME
  SLEEP -- Disable the GUI to prevent the user to interfere (V1)

USAGE
  SLEEP

DESCRIPTION
  This command disables Atlantis' GUI  to  prevent  the user from interfering
with the script.  All  windows  are disabled and the mouse pointer turns into
the busy-pointer.  Be careful when using this command, and make sure that the
GUI always is enabled again  before  the  script  exits. Therefore you should
trap all run-time errors when using this  function.  Read the ARexx manual to
find out how to trap run-time erros.

INPUTS
  None.

RESULT
  This command returns no value.

EXAMPLE
  /**/

  ADDRESS ATLANTIS

  SLEEP  /* Prevent the user from interfering */

  /* Do your script stuff */

  WAKEUP /* Give the user control again */

SEE ALSO
  WAKEUP
```

1.37 Atlantis - The Arexx Interface - UPDATE

```
NAME
  UPDATE -- Refresh the display (V1)

USAGE
  UPDATE

DESCRIPTION
  This command redraws all map windows in the GUI to make the changes done in
the script visible to the user.  If your script doesn't use the SLEEP/WAKEUP
pair it should call  this  command  before  it exits (the WAKEUP command also
updates the GUI, so there is no need to call both UPDATE and WAKEUP).
```

```
INPUTS
  None.

RESULT
  This command does not return a value.

EXAMPLE
  /**/

  ADDRESS ATLANTIS

  USEMACRO   /* Use the same macro as the user */
  PLOT 1 1   /* Plot it somewhere */
  UPDATE     /* Refresh the display before exiting */

SEE ALSO
  SLEEP, WAKEUP
```

1.38 Atlantis - The Arexx Interface - USEMACRO

```
NAME
  USEMACRO -- Set the "current" macro (V1)

USAGE
  USEMACRO [<macro>]

DESCRIPTION
  This command selects which macro  to use in the subsequent ARexx calls. The
ARexx 'current' macro and the users 'current' macro is NOT the same macro. If
called without the <macro>  parameter  the ARexx current macro will be set to
the same as the users current macro. Do not assume anything about which macro
is the current.  The  current  macro  is not guaranteed to be a certain macro
until this command is called.  The  current macro is macro 0 when Atlantis is
started, but is most probably changed every time an ARexx script is called.

INPUTS
  <macro> The macro to use in the subsequent ARexx calls. Optional.

RESULT
  This command returns no value.

EXAMPLE
  /**/

  ADDRESS ATLANTIS

  USEMACRO     /* Use the same macro as the user */
  PLOT 1 1     /* Plot it somewhere */
  UPDATE       /* Refresh the display */

SEE ALSO
  GETFREEMACRO

NOTES
```

All ARexx scripts should call this function before doing any drawing operations.

1.39 Atlantis - The Arexx Interface - Version

NAME
  VERSION -- Return the version of the ARexx interface (V1)

USAGE
  VERSION

DESCRIPTION
  This command returns the version of the ARexx interface. With this command you can make sure that the ARexx port supports the commands used in the script.

INPUTS
  None.

RESULT
  The ARexx variable RESULT is set to the version of the ARexx interface.

EXAMPLE

```
  /**/

  OPTIONS RESULTS
  ADDRESS ATLANTIS

  VERSION     /* Get the version */

  IF (RESULT ~= 1) THEN DO
    SAY 'Sorry, need version 1 of the ARexx interface'
    EXIT
  END
```

1.40 Atlantis - The Arexx Interface - WAKEUP

NAME
  WAKEUP -- Enable the GUI and refresh the display (V1)

USAGE
  WAKEUP

DESCRIPTION
  This command activates the GUI after a SLEEP command. Always remember to call this command if you have put the GUI to sleep earlier, otherwise the user will not be able to do anything. This command also updates all windows so there is no need to call the UPDATE command.

INPUTS
  None.

RESULT
  This command returns no value.

EXAMPLE

```
  /**/

  ADDRESS ATLANTIS

  SLEEP  /* Prevent the user from interfering */

  /* Do your script stuff */

  WAKEUP /* Give the user control again */
```

SEE ALSO
  SLEEP, UPDATE

1.41 Atlantis - The file format

The File Format

The file format of the maps is very simple, it could be described in pseudo-code like this:

```
  struct AtlantisMap {
    ULONG FileID;
    UWORD Width;
    UWORD Height;
    WORD Data[Width * Height];
  }
```

The map file starts with the identification tag 'MAP1' in the FileID field above. Then comes the map width and height in two 16 bit unsigned words. Then comes the map data in the form of Width * Height signed words. They are signed words because Atlantis uses negative values internally for special purposes.

It is not difficult to see that this format is not very fault-tolerant. Because of this I would like to develop an IFF MAP format, but I would want you, the users, opinion of what the form should cover and what it should not. Therefore, please e-mail me for comments about this.

It is Texas law that

"... when two trains meet each other at a railroad crossing, each shall come to a full stop, and neither shall proceed until the other has gone."

1.42 Atlantis - The history of Atlantis

```
23-Apr-97
   * Version 1.0
   * First official release.

 2-May-97
   * Version 1.1
   * Fixed a serious bug in the  save-routine.  Atlantis  only saved half the
     map if you were not registered!
   * Fixed some graphic bugs in the block window
   * Added a skeleton rexx script.  Start  from this one when making your own
     scripts.
   * Atlantis no longer complains if it can't find the preferences file.
```

1.43 Atlantis - Known bugs

Known bugs

If you use odd block sizes the overview mode will look strange. This is due to integer rounding errors. To avoid this, use block sizes that are multiples of 4 (8, 12, 16, 20 ... 56, 60, 64 pixels)

When you press and hold the mouse button over the arrow gadgets and then moves the mouse inside the drawing area the cursor will be drawn. This happens because there is no way of telling if the user has released a boopsi boolean gadget outside the gadget's bounding box. This problem can be solved by creating a custom boopsi class for the arrow gadgets, so I will probably do that in the future.

If you find any other bugs, don't hesitate to contact me

Man invented language to satisfy his deep need to complain.
        -- Lily Tomlin

1.44 Atlantis - Next version

In the future

The following features are planned for the future of Atlantis:

- Stencils
- Multiple layers
- Composite blocks
- "Three dimensional" blocks. That is blocks that can partly cover other blocks.
- Multiple levels editing

- Printing of levels (I don’t have a printer yet :( )
- Block editing
- Block attributes
- Macro editing, loading & saving

Ideas are of course welcome. If you have a good idea or missing some function don’t hesitate to drop me a line.

It has taken me more than a year to complete this program, mainly because this also was a project to learn C-programming. I have rewritten large parts of the code several times, and I think there is still more to do. Therefore I promise you that there will be several new versions after this one. I have a big TODO list with many more features to implement. If you have any suggestions or complaints, don’t hesitate to write to me.

The future arrives before we expect it, but is seldom what we expect...
 - Arnold Glasow

1.45 Atlantis - Thanks to

Thanks to the following people:

- Toni Brandt for testing
- Stefan Stuntz for MUI
- Russell Leighton for Icon.mcc
- Urban Müller for his work with AmiNET
- Roman Patzner for his icons
- Maik Solf for bugreport and ideas
- All people that is about to register very soon ;)
- All people who still believes in the Amiga!

It has just been discovered that research causes cancer in rats.

1.46 Atlantis - Contact Address

Contact Address

Send registrations, comments, suggestions and flames to:

Staffan Palmroos
Rydsvagen 250 C:17
58434 Linkoping

Sweden

Note: During the summer (June, July and August) I can be hard to reach, since I then live at another place. If you send any mail in that period it may take a while for me to reply.

```
  EMAIL: crush@lysator.liu.se
         d93stapa@isy.liu.se
         d93stapa@und.ida.liu.se
```

EMail is preferred.

Visit the official homepage of Atlantis:

http://www.lysator.liu.se/~crush/atlantis.html

(path is case-sensitive)

THE LESSER-KNOWN PROGRAMMING LANGUAGES #18 -- FIFTH

FIFTH is a precision mathematical language in which the data types refer to quantity. The data types range from CC, OUNCE, SHOT, and JIGGER to FIFTH (hence the name of the language), LITER, MAGNUM and BLOTTO. Commands refer to ingredients such as CHABLIS, CHARDONNAY, CABERNET, GIN, VERMOUTH, VODKA, SCOTCH, and WHATEVERSAROUND.

The many versions of the FIFTH language reflect the sophistication and financial status of its users. Commands in the ELITE dialect include VSOP and LAFITE, while commands in the GUTTER dialect include HOOTCH and RIPPLE. The latter is a favorite of frustrated FORTH programmers who end up using this language.

1.47 Atlantis - Glossary

Explanations of some terms used in this text

Block

A block is a rectangular piece of graphics that the maps are built with. The blocks is drawn in a bitmap drawing program of your choice, and saved in an IFF picture file. In Atlantis the blocks can be as small as 8 x 8 pixels, but not larger than 64 x 64 pixels. All the blocks in the block file must have the same size.

Block file

The IFF picture file containing the blocks. If the file system supports it, Atlantis will automatically reload the block file whenever it is modified by another program, so that you don’t have to quit and restart Atlantis if you want to change the blocks.

Commodities

The commodities system is new to version 2.0 of AmigaOS. It lets the user control certain properties of an application. The commodities system lets you iconify, disable, enable or quit Atlantis with the help of the 'Exchange' program, which itself is a 'commodity' program.

Display Cache

All map windows can have a display cache to speed up the redrawing of the window. The display cache is a buffer in memory that remembers which blocks has been drawn in the map previously. If a block is to be drawn in the window at a position already containing the block, the block is not drawn. Since the display cache uses extra memory, this feature can be turned off. The effects of the display cache is very noticeable when scrolling a map window.

EatMem

A program used to simulate low-memory situations.

Enforcer

A program used when developing applications to catch illegal memory accesses. Requires a processor with an MMU.

Environment variable

Environment variables are global system parameters used to describe your setup and preferred settings. You set environment variables with the cli command SetENV. Atlantis looks for one environment variable: KEYPATH. The KEYPATH variable should be set to the path where you keep your key files. I keep all my keyfiles in the S:KeyFiles/ directory, so to set my KEYPATH variable I write

```
SetENV KEYPATH S:KeyFiles
```

Note that this does not set the variable permanently. If you want the variable to be set permanently you should copy the variable to the ENVARC: directory, like this:

```
Copy ENV:KEYPATH ENVARC:
```

Filenotify

From version 2.0 of AmigaDOS the filesystem can notify an application that a file has been changed. Not all filesystems support this feature though, but if the filesystem you are using supports it, Atlantis will automatically reload the block file every time it changes. This lets you modify the blocks in the block file without having to restart Atlantis to load in the new blocks.

Installer

The Installer is a program with the sole purpose of installing other programs. The author of a program creates an installer script that the user then uses to install the program. The Installer program normally reside in the Utilities drawer on the SYS: volume (the startup volume).

Keyfile

A keyfile is a special file you will get when registering that will enable the disabled features of Atlantis. This keyfile should preferably be put in the drawer described by the environment variable KEYPATH. It can also be put in S: and the main program directory, but note that it is an criminal act to give out your keyfile to someone else. Your name and address is encoded in the keyfile so it is very easy to see from where a keyfile has originated.

Localization
A feature of AmigaOS 2.1 and later which enables programs to come in different languages by providing a separate file containing the programs strings in a different language.

Macro
The drawing brushes in Atlantis is called macros. You have 10 macros to draw with, but two of them have special properties. Macro 0 is reserved for the block window, and macro 9 is a scratch macro.

Mark
A mark is simply a marker in the map that is used to mark out positions in the map. The marks can be used for a variety of things, for example to send coordinates to ARexx scripts. All functions in the menu uses marks instead of coordinates.

Map
A two-dimensional array of pointers to the blocks.

Map file
The file containing the map.

MUI
Short for Magic User Interface. A system to create nice GUI's (graphical user interfaces). See the disclaimer section for more info.

Mungwall
A program used developing applications to track memory allocations.

Overview mode
The overview mode lets you view a larger part of the map by using a set of blocks 1/16:th in size. If there is no memory for the overview blocks the overview mode will be disabled.

Public Domain
A Public Domain package is a package that can be redistributed and copied given some restrictions. The package must be complete and not modified in any way. Public Domain programs usually comes with no guarantee and it if something nasty happens when running the program the author(s) can not be held liable.

Rexx
A special scripting language used to control other applications. The Amiga version (ARexx) is © William S. Hawes.

Segtracker
A program to log which memory blocks that belongs to different programs. In combination with Enforcer it can tell which program is causing the error and even what part of the program.

The current macro
The current macro is the macro you are currently drawing with. The ARexx interface has its own current macro. If the script calls the USEMACRO

command without a parameter the ARexx current macro will become the same as the users current macro.

Virtual Memory

Virtual memory is a technique to use a part of a hard disk partition for memory. To be able to do this you have to have a MMU (Memory Management Unit) in the system. This unit could either be built-in in the CPU or be a separate 68551 chip. If you think you have a MMU check out the program VMM (available for example on AmiNET). Atlantis fully supports VMM, which lets you make very large maps.

Reporter: Mr. Gandhi, what do you think of Western Civilization?
Gandhi: I think it would be a good idea.

1.48 Atlantis - Alphabetical INDEX

A

- Address
- ARexx
  - ASKUSER
  - BOX
  - CLEARMARK
  - FILL
  - GETFREEMACRO
  - GRABFROMMAP
  - GRABBLOCKS
  - HELP
  - HIDE
  - INFO
  - INFORM
  - KILLMACRO
  - LINE
  - MAPHEIGHT
  - MAPWIDTH
  - MARKX
  - MARKY
  - NEXTSETMARK
  - NUMSETMARKS
  - PLOT
  - QUIT
  - RECT
  - SETMARK
  - SHOW
  - SLEEP
  - UPDATE
  - USEMACRO
  - VERSION
  - WAKEUP
- Author

B

- Block

Block file
Block graphics
Block size
Block window
Bugs

C
Commodities

D
Disclaimer
Display Cache
Distribution

E
EatMem
EMail address
Enforcer

F
Fast Cursor
File format
Filenotify
Future

I
Icons
  default
Icon.mcc
Installation

K
Keyfile

L
Licence

M
Macro
Map
Map file
Map windows
Mark
Menus
  About
  About MUI
  Change Dim.
  Jump to Mark
  Move Area
  New
  Preferences
  Save
  Tool Window
  Quit
MUI
Mungwall

```
O
 Overview mode

P
 Preferences
   Autosave
   Create Icon
   Display Cache
   Fast cursor
   Keep Backup
   Query Autosave
   Screenmode
   Undo buffer
 Project name

R
 Registering
   company
   status
   VIP
 Requirements

S
 SegTracker
 Shareware fee
   payment

T
 Thanks
 Tool window
   Box
   Clear Mark
   Copy
   Fill
   Line
   Plot
   Rectangle
   Set Mark
   Undo
 Tutorial

U
 Undo buffer

V
Virtual Memory

W
 Warranty
```