main

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

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

main

1.1 MapMaster C documentation Guide

```
MapMaster C functions Version 1.0
Hello welcome to the documentation for MapMaster C Functions.
```

```
These Functions are FreeWare and may be used in your
own programs if you include my name in the credits

If you are a shareware author
email me a keyfile for your program as payment. :-).

If you have any questions as to how to use these Functions
or suggestions as to what features you would like to
see just E-Mail.

E-Mail- samel@telusplanet.net
my WebPage featuring-
Emulators my programs and animations
and various other stuff is at
www.telusplanet.net/public/samel/index.html

          MapMaster ©1997 Kelly Samel.
```

1.2 Introduction

```
Introduction
```

This is a C language Include file that has Functions
in it to use maps generated by MapMaster. I created these
Functions to help make it easier to use the maps and as an example
of how to make your own Functions.

1.3 Requirements

Requirements

You will need an Amiga computer with a decent C compiler
and a bit of programming knowledge. You will also require
the include files for at least v37.

1.4 Installation

Installation

Copy the mapmaster drawer to your include dir.
Copy the iff.library to libs.

Thats all!

1.5 Usage

Usage

This will explain the parameters that each of the functions take
and how to use them.

First of all make sure you install the mapmaster drawer into your
include drawer and copy iff.library into your libs drawer.
You may want to look at the Example included, its much easier
to follow than it looks here. Its in the MapExample drawer
in binary and c source forms.

1. How to use these Functions

The First thing you must do is define a global array of int sized
variables to hold your map data.
like this-int mapdata[horizblocks*vertblocks+1];
This array must be called mapdata[]; and be large enough to
hold your map.

Secondly you must #include <mapmaster/mapfunctions.h>
this file holds the actual functions and may be changed
to suit your needs.

Once you do this you should declare a BitMap like this

```
Struct BitMap *blocks;
to hold the picture that contains your blocks.
You can name it anything you like.

lets review what we must have to use this.
1. #include <mapmaster/mapfunctions.h>
2. int mapdata[241]; /* must be global */
3. Struct BitMap *mybitmap;

After you have performed these three simple steps you
are ready to use the functions ...

Note- These functions have very little error checking
      you must make sure the data fits into the bitmap
      and that the map you paste down fits into the
      Destination BitMap perfectly.
      eg. horizblocks*blockwidth=widthofbitmap
          vertblocks*blockheight=heightofbitmap

2. Parameters and description of each function.

These functions are written in Sas/c V6.3 and should
be compatible on any C compiler with a bit of tweaking.

These are the common parameters that I will explain.

int horizblocks- This refers to the actual Horizontal # of blocks
                 in your actual map.
int vertblocks-  This refers to the actual Vertical # of blocks
                 in your map.
int blockwidth-  This refers to the width of the blocks in pixels
int blockheight- This refers to the Height of the blocks in pixels
int blockgap-    This refers to the size of the gap between blocks
                 in your pic in pixels.

Here is a listing of all Functions and how to use them.

List Of All Functions
LoadMap()- Loads in the mapdata[]
GetBlocks()-Gets the iff containing your blocks into a bitmap
GetBlocksPalette()-Gets the color palette of your blocks
FreeBlocks()- Frees the memory stolen by GetBlocks
PasteBlocks()-Pastes your map to any BitMap including the screens.

Detailed Description
----------------------------------------------------------------------------
NAME
LoadMap

SYNOPSIS
void LoadMap(char filename[],int horizblocks,int vertblocks);

Function
LoadMap is used to load in the mapdata from a mapmaster
map file for use by the other Functions.

INPUTS
```

```
filename[]- name of the map to load
int horizblocks- This refers to the actual Horizontal # of blocks
                 in your actual map.
int vertblocks-  This refers to the actual Vertical # of blocks
                 in your map.

----------------------------------------------------------------------------
----------------------------------------------------------------------------
NAME
GetBlocks

SYNOPSIS
void GetBlocks(char blocksfile[],struct BitMap *tempbitmap,int depth,
               int bmwidth,int bmheight);

Function
This loads the Iff picture data containing your blocks into
a bitmap that will be used by the pasting function.
This Function automatically initializes and allocates
memory for the BitMap supplied.

INPUTS
blocksfile[]- Name of the iff file containing your blocks
*tempbitmap-  A pointer to a unintialized bitmap structure that will be
              used to store the blocks.
depth - The depth of the iff containing your blocks.
bmwidth- Width of the iff file containing your blocks. eg.320
bmheight- Height of the iff file containing your blocks. eg.200

----------------------------------------------------------------------------
----------------------------------------------------------------------------
NAME
GetBlocksPalette

SYNOPSIS
void GetBlocksPalette(char blocksfile[],struct Screen *screen);

Function
This loads the palette from the specified Iff file
and loads it into a screen of your choice.Its meant
to be used to get the palette of your blocks.

INPUTS
blocksfile[]- Name of the iff file containing your blocks
*screen- Pointer to the screen you would like the colors to
         be loaded into.
----------------------------------------------------------------------------
----------------------------------------------------------------------------
NAME
FreeBlocks

SYNOPSIS
void FreeBlocks(struct BitMap *tempbitmap,int depth);

Function
This frees the memory allocated by GetBlocks.
```

```
INPUTS
*tempbitmap-  A pointer to the bitmap structure previously
              initialized by GetBlocks.
depth - The depth of the BitMap containing your blocks.

----------------------------------------------------------------------------
----------------------------------------------------------------------------
NAME
PasteBlocks

SYNOPSIS
void PasteBlocks(int horizblocks,int vertblocks,
                 int blockwidth,int blockheight,int blockgap,
                 int xoffset,int yoffset,
                 struct BitMap *source,struct BitMap *dest);

Function
This Pastes your blocks to another BitMap using the mapdata[].
It pastes from the upper left corner to the bottom right corner
in a left to right fashion.It can paste to a screen with &screen->BitMap
as a *dest parameter.

INPUTS
horizblocks- This refers to the actual Horizontal # of blocks
                 in your actual map.
vertblocks-  This refers to the actual Vertical # of blocks
                 in your map.
blockwidth-  This refers to the width of the blocks in pixels
blockheight- This refers to the Height of the blocks in pixels
blockgap-    This refers to the size of the gap between blocks
                 in your pic in pixels.
xoffset- Will offset the x pasting by however many pixels
         you give it.
yoffset- Will offset the y pasting by however many pixels
         you give it.
*source- The source BitMap to get your blocks from.
*dest-   The Destination BitMap to paste the blocks to. eg.&screen->BitMap

----------------------------------------------------------------------------
----------------------------------------------------------------------------
```

1.6 The Block Format

```
"this describes the format used by MapMaster you may use
 more colors ect."

The Block Format

The blocks you use to build up your map must be rectangular
shaped and all the same width and height. They can be in a 2
to 32 color Iff file in the standard screen dimensions of
320x200 for Lores use or 640x200 for Hires use. They must
be evenly spaced and put from left to right as far as they will
fit evenly. Look at the example blocks included. There may
be a gap in between each block as long as it is even all
```

```
around them. They do not have to fill the whole page but
must go from left to right as far as possible without
cutting them off on the right side.

The blocks will be cut from left to right and the first
block on the left will be block 1 and each block after
that will go up by 1. Eg. 1  2  3  4  5  6
                          7  8  9  10 11 12
                          13 14 15 16 17 18
```

1.7 The Map Format

```
The Map Format

The map format used by MapMaster is very simple. Its a plain
Text file that has a listing of all blocks used in your map
by there number. Each listing is moved down 1 line by hitting
the newline key (return) this is done to seperate the numbers
and make them easy to read with any programming language.
Your program must handle getting the blocks and pasting
them down in the correct places this is fairly easy
to set up in a loop. There is some example code for AmosPro
in the Amos drawer for doing this. I am still working on
C versions of the example code. If you know what would
be the best way to do this in C then E-Mail me.
Its best to store the map data in an array of some kind.
In Amos- Dim map(width*Height)
In C-    int mapdata[width*Height+1];

Example.

w= block 1
o= block 2

wwwwwwwwww
wwoooooww
woooooooow
woooooooow
wwoooooww
wwwwwwwwww

if this is your map and you save it this is
the way the save file would look. If you
would like another save format with more
info or stored differently E-Mail and I'll
see what I can do.

1
1
1
1
1
1
1
1
```

```
1
1
1
1
2
2
2
2
2
2
1
1
Ect...
```

1.8 Bugs

Bugs

I think there might be a few little bugs yet but its pretty
good. E-Mail me if you have problems and I will try to fix it.

Email- samel@telusplanet.net

1.9 Future

Future

If you have any ideas/Suggestions for future versions of these
functions E-Mail them to me and I will try to implement them.

E-Mail: samel@telusplanet.net

1.10 Author

Author

Hi my name is Kelly Samel and I have been using the amiga for
almost 3 years now and will be continuing to use it for many
years to come. I like to do Raytracing in Imagine programming
in AmosPro and now C as well as play games and various other
things like painting going on the net ect.

Feel free to e-mail me and tell me if you liked these Functions.
Or if you have any docs or knowledge that would help a new guy get better
at programming in C, mail me as well.

my email is- samel@telusplanet.net
my WebPage featuring-
Emulators my programs and animations
and various other stuff is at

www.telusplanet.net/public/samel/index.html