

RDTextures

COLLABORATORS

	<i>TITLE :</i> RDTextures		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 5, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	RDTextures	1
1.1	RDTextures.guide	1
1.2	Introduction	2
1.3	History	2
1.4	Terminology	2
1.5	Requirements	2
1.6	Installation	3
1.7	PBM PGM PPM	3
1.8	Gui4Cli: Limitations and Rationale	3
1.9	palgen	3
1.10	cpgm	4
1.11	texture	4
1.12	Problems	7
1.13	Future	7
1.14	Author	8
1.15	Acknowledgements	8
1.16	Appendix	8

Chapter 1

RDTextures

1.1 RDTextures.guide

```
unrealaudio v1.0
```

Dodgy docs and dodgy program written by
Michael Cheng

Introduction

History

Terminology

Requirements

Installation

PBM PGM PPM

The GUI

The Programs:

palgen

cpgm

texture

Problems

Future

Acknowledgements

Author

Appendix

1.2 Introduction

I wanted to play with seamless textures and pretty colours so I wrote the following CLI programs:

texture - seamless texture generator

palgen - palette generator (for specifying gradient palettes)

cpgm - colours a pgm file using a palgen palette

I've written little Gui4Cli gui's for the programs so that it's reasonably simple to play with the numbers to get different output. (Beware the limitations of the

GUI

.

What I'd really like is a palette manipulator like XV and a texture generator like Kai's Power Tools. *sigh*

1.3 History

v1.0

1 September 1996

- initial release

1.4 Terminology

seamless texture - one that can be tile mapped without showing a seam

reaction-diffusion - an algorithm used to describe some biological processes

pbm - portable bit map (?). Generic name for these utils. (gfx/pbm)

pgm - portable grey map (?). Grey scale picture format.

ppm - portable picture map (?). 24bit colour picture format.

1.5 Requirements

- a 68000+

- ixemul43+ (I should recompile with libnix)

- stack 20000

- Gui4Cli installed

- some of the binaries from the PBM distribution
+ ppmtolbm

1.6 Installation

1. Copy the programs in the rdttextures/c directory to somewhere in your path
2. Ensure that you have ixemul43+ installed, as well as Gui4Cli
3. Start 'texture.gc' by either double clicking it's icon, or by starting gui4cli and then loading it in.

1.7 PBM PGM PPM

Why use PBM format images?

- (a) because I originally wrote this program on a unix machine
- (b) PBM/PGM/PPM is a damn simple format from a programming standpoint
- (c) I would much rather spend time playing with algorithms than file I/O in the 'correct' format

1.8 Gui4Cli: Limitations and Rationale

Why use Gui4Cli instead of a real GUI?

- (a) I'm a hopeless programmer
- (b) I originally wrote these programs on a unix machine
- (c) I'd much rather spend time on algorithms than a UI

Limitations of my Gui4Cli gui

- there's no progress indicator. Just be aware that button clicks within the GUI while it's calculated are buffered, and not executed until after the previous button click completes its task. *TIP* Texture generation can take quite a while. click 'generate texture' and 'view texture' and then wait. The 'view texture' will not execute until the texture generation is complete, which means that when it does complete, the texture will be shown.
- No stdout or stderr output
Get around this by running the Gui4Cli program 'gui' within a CLI and then picking the .gc file from the requestor. Any stdout,stderr output from the GUI gets sent to the CLI from where 'gui' was started

1.9 palgen

This is a really simple palette generator. It generates 256 color ↔ palettes which are used to map the 256 grey levels in a PGM file to 24bit rgb colour in a PPM file. (The conversion from PGM to PPM is done using cp_{gm})

Syntax

```
palgen -p <palfile> -t <level> <red> <green> <blue> ...
```

palgen assigns the 24bit colour given by the rgb triplet to the grey level <level>. It then iterates the values between different levels.

(The first level given is forced to be level 0 regardless of what is actually there)

eg

```
palgen -t 0 0 0 0 -t 255 255 0 0
```

This will make a colour scale from black to bright red over 256 levels. Palette is saved in default "test.pal"

```
palgen -t 0 255 255 255 -t 50 150 150 0 -t 150 100 10 10 -t 255 0 0 0
```

This will make a colour scale from white to yellow to red to black. (a pretty easy sort of 'fire' palette)

palgen outputs

```
A Palette file for use with
    cpgm
    . This defaults to
    "test.pal"
```

A ppm file "test.ppm", which is a picture file showing the palette

1.10 cpgm

'cpgm' converts a PGM file to a PPM file using a palette created by palgen link "palgen"} to convert from greyscale to colour.

syntax

```
cpgm -i <pgmfile> -o <ppmfile> -p <palfile>
```

[I realise this offers the same functionality as the PBM program 'pgmtoppm' but I was having troubles with that, and this wasn't that hard to program and it is tailored to my needs]

Defaults:

```
pgmfile "test.pgm"
ppmfile "test.ppm"
palfile "test.pal"
```

1.11 texture

'texture' synthesises textures using a reaction diffusion algorithm. There are three internal buffers on which it operates

- 0 - SLUR
- 1 - MAIN

MAIN holds the information we are interested in and is the buffer that is output as the actual texture.

Values in the SLUR buffer are used as the localised diffusion directions in the MAIN buffer

ONLY USE BUFFERS 0 & 1

Methodology

- (1) Create a pattern in the SLUR buffer
- (2) Create a pattern in the MAIN buffer
- (3) Do Reaction-Diffusion on the MAIN buffer using the SLUR buffer for local diffusion direction

The program is told how to do all this through its command line arguments. There are two types

- (a) position sensitive
- (b) position insensitive

Position Insensitive Arguments

These can come anywhere in the command line, but appear only once:

-a <float> <float>

The directional diffusion factors. `alpha1` and `alpha2` determine how quickly diffusion takes place parallel and perpendicular (respectively) to the local orientation

DEFAULTS: 1.1 , 0.1

-b <float>

rate at which values in MAIN decay

DEFAULT: 0.001

-g <int>

Number of iterations of the reaction diffusion algorithm
(This number is directly proportional to run time)

DEFAULT: 10

Position sensitive

The order and positioning of these arguments will lead to different results.

-d <int> <int>

The size of the generated texture. (At present limited to 255x255)
It defaults to 100x100

*IF THIS ARGUMENT APPEARS AT ALL IT MUST APPEAR BEFORE ANY OTHER
POSITION SENSITIVE ARGUMENT*

-s <int>

Seed the random number generator with this number

-l <bufnum> <num> <angle>

Draw 'num' lines in buffer 'bufnum' at an angle 'angle'. If 'angle'
is negative then lines are drawn at random angles. Lines start at
a random point in the buffer, and are of a random length.

-l 0 10 30

Draws 10 lines at angle 30 in the SLUR buffer

-l 1 5 -1

Draws 5 randomly oriented lines in the MAIN buffer

-c <bufnum> <num> <radius>

Draw 'num' circles in the buffer 'bufnum' with radius 'radius'. If
'radius' is negative then radius is a random value. Circle centres
are at a random point in the buffer.

-c 0 5 20

Draws 5 circles of radius 20 in the SLUR buffer

-f <bufnum> <int num>

fills the buffer 'bufnum' with random values up to 'num'

-f 1 255

Fills the MAIN buffer with random noise between 0 and 255

-n <bufnum> <lower> <upper>

Normalize the values in buffer 'bufnum' to between 'lower' and
'upper'

-n 0 0.0 128.0

Rescales all the values in the SLUR buffer to between 0.0
and 128.0

-u <bufnum> <num>

Performs a simple blur on the contents of buffer 'bufnum'. Does the blur 'num' times.

```
-u 1 3
```

Does a simple blur of the MAIN buffer 3 times.

To create the texture "examples/example.ilbm" I used

```
texture -c 0 10 -1 -1 0 10 0 -1 0 10 30 -u 0 2 -f 1 250 -u 1 1
```

1.12 Problems

- Click a button but nothing seems to happen
 - (a) texture generation can take a while. Don't keep clicking buttons as gui4cli queues them up while it is busy.
 - (b) quit all 'multiview' windows and try again (can't redirect into a file that multiview is showing, thus stopping file conversions from 'ppm' to 'ilbm')
 - (c) quit Gui4Cli totally. restart it from a CLI by typing 'gui'. restart the gui. stderr and stdout should be redirected to the CLI. Should be able to figure out the problem by looking at the output
 - (d) it may be an option that doesn't produce any output. (eg 'generate palette' doesn't produce any noticeable output. click 'view palette' to see what it produced)
- Final Texture is blank
 - Be careful when changing the options. The default values definitely produce a picture. Changing these numbers too drastically may lead to a blank image. *TIP* Try little changes rather than big ones.

1.13 Future

Possibilities

- more save options. ie specify how often to save. (could make anims?)
 - stdout,stderr into a con?
 - could probably do quite a lot with palgen if i thought about it
 - + independant red green and blue control points
 - + control points on a spline (like XV)
-

- texture
 - + add in extra buffers.
 - + command "addbuf" to be accessible
 - + command "outputbuf" to be accessible
 - + command "eggtiler" to be accessible
 - + algorithm to seed MAIN with large non-regular features to offset all the small-scale stuff
- recompile with libnix instead of ixemul
- Feel free to re-write the gui4cli gui

1.14 Author

Mail me, Mail me, Mail me. Send me ideas, money, chocolate or a life.

Michael Cheng
memfc@alinga.newcastle.edu.au
<http://joffre.newcastle.edu.au/>
Cstar on #amiga

1.15 Acknowledgements

This little project wouldn't have been possible without the following people who write such useful stuff for the amiga.

Author of Gui4Cli
Author 24bit PicDT

1.16 Appendix

Do an altavista search for "texture synthesis"