sprites4curses

0.1.1

Generated by Doxygen 1.9.6

Chapter 1

Documentation

These folders contain documentation on the project, generated using Doxygen's latex output to pdf.

Chapter 2

palette.gpl

If your image does not have a palette of 256 colors, you can convert it to 8-bit indexed color mode with a custom palette in GIMP.

```
+ Open the image in GIMP.
+ Select Image > Mode > Indexed.

+ In the Indexed Color Conversion dialog, choose "Generate optimum palette" as the conversion type.
+ Under the "Maximum number of colors" option, enter "256".

+ Check the "Use custom palette" checkbox.
+ Click the "Edit palette" button.
+ In the Palette Editor dialog, click the "Import Palette" button.
+ Select "From file" and choose the palette file (palette.gpl).
+ Click "OK" to close the Palette Editor dialog.

+ Click "Convert" in the Indexed Color Conversion dialog to convert the image to indexed color mode with the c
+ Export the image in PNG format.
```

Chapter 3

sprites4curses

A library of scripts to deal with sprites in ncurses.

3.1 sprites.py

This is a python script that converts PNG's to a char representation. The output text should be a valid C declaration for a 3D char array.

It expects as arguments a directory with the images to convert. There's a dependency on Pillow to do the image conversion.

3.2 sheet-converter.py

This is a python script that converts a single PNG spritesheet to a char representation. The output text should be a valid C declaration for a 3D char array.

It expects as arguments the spritesheet file name, the sprite width, the sprite height, the thickness of the separator between sprites, and the start coordinate or the first sprite's left corner. There's a dependency on Pillow to do the image conversion.

3.3 png-resize.py

This is a python script that resizess PNG's to a desired size.

It expects as arguments a directory with the images to resize, and two ints for width and height of the resulting PNGs. There's a dependency on Pillow to do the image conversion.

3.4 palette.gpl

This is a GIMP palette file, useful for exporting PNG with the correct color alignment. Info on how to use it are in the palette-Readme.md file.

3.5 animate.c and animate.h

This is a C program to display an animation read from a formatted text file. The format expected is compatible with sprites.py specs.

3.6 demo.c

This is a demo program that calls directly in its main, the demo() function from the animate library.

3.7 Usage

To use the python scripts you need to install Pillow:

3.7.0.0.1 <tt>pip install Pillow</tt>

- To run the sprites script and redirect output on "file.txt", give a directory to get the png's from:

3.7.0.0.2 File names in the directory should follow a imageX.png, imageX+1.png pattern.

3.7.0.0.3 <tt>python sprites.py <directory> > file.txt</tt>

- To run the sheet converter script and redirect output on "file.txt", give all required arguments:

3.7.0.0.4 <tt>python sheet_converter.py <sheet file> <sprite width> <sprite height> <separator thickness> <starting coordinate> > file.txt</tt>

- To run the png resize script, give all required arguments:

3.7.0.0.5 <tt>python png-resize.py <sprites directory> <sprite width> <sprite height></tt>

3.7.0.1 This overwrites the source pngs, so be careful.

- To run the C demo program, you do:

3.7.0.1.1 File format should be a valid char array declaration, or one sprite row per line.

3.7.0.1.2 <tt>make; ./demo file.txt</tt>

- To be fancy you can use process substitution in bash to give the python output directly as an argument:

3.7.0.1.3 <tt>make; ./demo <(python sprites.py <directory>)</tt>

3.7.0.1.4 Possible animation glitches if the frame rate is too high, add in-between frames as needed.

Chapter 4

Namespace Index

4.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

Chapter 6

Namespace Documentation

6.1 png_resize Namespace Reference

Program that parses pngs from a passed spritesheet, to encode their color to a char per pixel.

Functions

- def usage ()
  Prints correct invocation.
- def resize_sprites (directory, targetSizeX, targetSizeY)
  Resizes all png files in the passed directory to the specified size.
- def main (argv)

6.1.1 Detailed Description

Program that parses pngs from a passed spritesheet, to encode their color to a char per pixel.

6.1.2 Function Documentation

6.1.2.1 main()

```
def png_resize.main (
              argv )
```

6.1.2.2 resize_sprites()

```
def png_resize.resize_sprites (
              directory,
              targetSizeX,
              targetSizeY )
```

Resizes all png files in the passed directory to the specified size.

Parameters

directory	The input directory with the pngs.
targetSizeX	The target width.
targetSizeY	The target height.

6.1.2.3 usage()

```
def png_resize.usage ( )
```

Prints correct invocation.

6.2 sheet_converter Namespace Reference

Program that parses pngs from a passed spritesheet, to encode their color to a char per pixel.

Functions

- def usage ()

  Prints correct invocation.
- def convert_spritesheet (filename, spriteSizeX, spriteSizeY, separatorSize, startCoords)

  Converts a spritesheet to a 3D char array representation of pixel color and then prints it with the needed brackets and commas.
- def main (argv)

  Main program entry.

6.2.1 Detailed Description

Program that parses pngs from a passed spritesheet, to encode their color to a char per pixel.

6.2.2 Function Documentation

6.2.2.1 convert_spritesheet()

```
def sheet_converter.convert_spritesheet (
              filename,
              spriteSizeX,
              spriteSizeY,
              separatorSize,
              startCoords )
```

Converts a spritesheet to a 3D char array representation of pixel color and then prints it with the needed brackets and commas.

Parameters

filename	The input spritesheet file.
spriteSizeX	The sprite width.
spriteSizeY	The sprite height.
separatorSize	Thickess of separator pixels.
startCoords	Coords (a,a) of left corner of first sprite.

6.2.2.2 main()

```
def sheet_converter.main (
              argv )
```

Main program entry.

6.2.2.3 usage()

```
def sheet_converter.usage ( )
```

Prints correct invocation.

6.3 sprites Namespace Reference

Program that parses pngs from a passed directory, to encode their color to a char per pixel.

Functions

- def usage ()

  Prints correct invocation.
- def color_distance (c1, c2)

  Calculates the distance in color between two rgb tuples.
- def create_palette (colors)

  Creates a palette by adding the colors passed as an array, then pads the palette.
- def convert_sprite (file)

  Takes a image file and converts each pixel to a char representation of its color (closest match to CHAR_MAP).
- def print_converted_sprites (direc)

  Takes a directory containing image file and calls convert_sprite on each one.
- def main (argv)

  Main program entry.

6.3.1 Detailed Description

Program that parses pngs from a passed directory, to encode their color to a char per pixel.

6.3.2 Function Documentation

6.3.2.1 color_distance()

```
def sprites.color_distance (
              c1,
              c2 )
```

Calculates the distance in color between two rgb tuples.

Parameters

c1	The first input color to measure.
c2	The second input color to measure.

Returns

The color distance between the two.

6.3.2.2 convert_sprite()

```
def sprites.convert_sprite (
              file )
```

Takes a image file and converts each pixel to a char representation of its color (closest match to CHAR_MAP).

Parameters

file	The image file to convert.

Returns

The converted sprite as a char matrix.

6.3.2.3 create_palette()

```
def sprites.create_palette (
              colors )
```

Creates a palette by adding the colors passed as an array, then pads the palette.

Parameters

colors	The colors to put in the palette.

Returns

The filled and padded palette.

6.3.2.4 main()

```
def sprites.main (
              argv )
```

Main program entry.

6.3.2.5 print_converted_sprites()

```
def sprites.print_converted_sprites (
              direc )
```

Takes a directory containing image file and calls convert_sprite on each one.

Then it outputs all the converted sprites to stdout, including the necessary brackets to have a valid C array declaration.

Parameters

direc	The directory of image files to convert and print.

6.3.2.6 usage()

```
def sprites.usage ( )
```

Prints correct invocation.

Chapter 7

File Documentation

7.1 sprites4curses/animate.c File Reference

```
#include <ncurses.h>
#include <string.h>
#include <ctype.h>
#include <stdlib.h>
#include "animate.h"
```

Functions

- void print_spriteline (char ∗line, int line_num)
- char ∗ trim (char ∗str)
- void load_sprites (char sprites[NUM_FRAMES][ROWS][COLS], const char ∗filename)
- void usage (char ∗progname)
- int demo (int argc, char ∗∗argv)

7.1.1 Function Documentation

7.1.1.1 demo()

```
int demo (
          int argc,
          char ** argv )
```

7.1.1.2 load_sprites()

```
void load_sprites (
          char sprites[NUM_FRAMES][ROWS][COLS],
          const char * filename )
```

7.1.1.3 print_spriteline()

```
void print_spriteline (
             char * line,
             int line_num )
```

7.1.1.4 trim()

```
char * trim (
             char * str )
```

7.1.1.5 usage()

```
void usage (
             char * progname )
```

7.2 sprites4curses/animate.h File Reference

Macros

- #define RED 1
- #define YELLOW 2
- #define GREEN 3
- #define CYAN 4
- #define BLUE 5
- #define MAGENTA 6
- #define BLACK 7
- #define WHITE 8
- #define NUM_FRAMES 31

  Defines the number of sprites, +1.
- #define ROWS 18

  Defines the number of rows per sprite, +1.
- #define COLS 18

  Defines the number of colums per sprite, +1.
- #define FRAMETIME 67

  Defines for how many millisecs a sprite should stay on screen.

Functions

- void print_spriteline (char ∗line, int line_num)
- char ∗ trim (char ∗str)
- void load_sprites (char sprites[NUM_FRAMES][ROWS][COLS], const char ∗filename)
- void usage (char ∗progname)
- int demo (int argc, char ∗∗argv)

7.2.1 Macro Definition Documentation

7.2.1.1 BLACK

```
#define BLACK 7
```

7.2.1.2 BLUE

```
#define BLUE 5
```

7.2.1.3 COLS

```
#define COLS 18
```

Defines the number of colums per sprite, +1.

7.2.1.4 CYAN

```
#define CYAN 4
```

7.2.1.5 FRAMETIME

```
#define FRAMETIME 67
```

Defines for how many millisecs a sprite should stay on screen.

7.2.1.6 GREEN

```
#define GREEN 3
```

7.2.1.7 MAGENTA

```
#define MAGENTA 6
```

7.2.1.8 NUM_FRAMES

```
#define NUM_FRAMES 31
```

Defines the number of sprites, +1.

7.2.1.9 RED

```
#define RED 1
```

7.2.1.10 ROWS

```
#define ROWS 18
```

Defines the number of rows per sprite, +1.

7.2.1.11 WHITE

```
#define WHITE 8
```

7.2.1.12 YELLOW

```
#define YELLOW 2
```

7.2.2 Function Documentation

7.2.2.1 demo()

```
int demo (
            int argc,
            char ** argv )
```

7.2.2.2 load_sprites()

```
void load_sprites (
            char sprites[NUM_FRAMES][ROWS][COLS],
            const char * filename )
```

7.2.2.3 print_spriteline()

```
void print_spriteline (
            char * line,
            int line_num )
```

7.2.2.4 trim()

```
char * trim (
            char * str )
```

7.2.2.5 usage()

```
void usage (
            char * progname )
```

7.3 animate.h

Go to the documentation of this file.

```
//These define the colors for init_pair() without an order
#define RED 1
#define YELLOW 2
#define GREEN 3
#define CYAN 4
#define BLUE 5
#define MAGENTA 6
#define BLACK 7
#define WHITE 8

#define NUM_FRAMES 31
#define ROWS 18
#define COLS 18
#define FRAMETIME 67
void print_spriteline(char* line, int line_num);
char *trim(char *str);
void load_sprites(char sprites[NUM_FRAMES][ROWS][COLS], const char *filename);
void usage(char* progname);
int demo(int argc, char** argv);
```

7.4 sprites4curses/demo.c File Reference

```
#include <ncurses.h>
#include <string.h>
#include <ctype.h>
#include <stdlib.h>
#include "animate.h"
```

Functions

- void main (int argc, char **argv)

7.4.1 Function Documentation

7.4.1.1 main()

```
void main (
             int argc,
             char ** argv )
```

7.5 sprites4curses/documentation/README.md File Reference

7.6 sprites4curses/README.md File Reference

7.7 sprites4curses/palette-README.md File Reference

7.8 sprites4curses/png_resize.py File Reference

Program that resizes pngs to a desired size and overwrites them.

Namespaces

- namespace png_resize

  Program that parses pngs from a passed spritesheet, to encode their color to a char per pixel.

Functions

- def png_resize.usage ()

  Prints correct invocation.
- def png_resize.resize_sprites (directory, targetSizeX, targetSizeY)

  Resizes all png files in the passed directory to the specified size.
- def png_resize.main (argv)

7.8.1 Detailed Description

Program that resizes pngs to a desired size and overwrites them.

7.8.2 Description

The program overwrites the passed pngs with the resized version.

Program expects the spritesheet filename as first argument, the sprite width as second arg, the sprite heigth as third arg.

7.8.3 Libraries/Moodules

- Pillow (`https://pillow.readthedocs.io/en/stable/`)
  - Access to image manipulation functions.
- sys standard library (`https://docs.python.org/3/library/sys.html`)
  - Access to command line arguments.
- os standard library (`https://docs.python.org/3/library/os.html`)
  - Access to program name.

7.8.4 Notes

- The pngs are overwritten by default.

7.8.5 TODO

- Offer option to output to new files and not overwrite.

7.8.6 Author(s)

- Created by jgabaut on 24/02/2022.
- Modified by jgabaut on 24/02/2022.

7.9 sprites4curses/sheet_converter.py File Reference

Program that parses pngs from a passed spritesheet, to encode their color to a char per pixel.

Namespaces

- namespace sheet_converter

  Program that parses pngs from a passed spritesheet, to encode their color to a char per pixel.

Functions

- def sheet_converter.usage ()

  Prints correct invocation.
- def sheet_converter.convert_spritesheet (filename, spriteSizeX, spriteSizeY, separatorSize, startCoords)

  Converts a spritesheet to a 3D char array representation of pixel color and then prints it with the needed brackets and commas.
- def sheet_converter.main (argv)

  Main program entry.

7.9.1 Detailed Description

Program that parses pngs from a passed spritesheet, to encode their color to a char per pixel.

7.9.2 Description

The program supports 8 colors at the moment. The png parsing uses Pillow, and the mapping is done against a preset color list. The list is described in palette.gpl to aid in exporting images with the correct color indexing.

Program expects the spritesheet filename as first argument, the sprite width as second arg, the sprite height as third, separator size (thickness) as fourth, a 0 or 1 for starting coords of the first sprite (0 if sheet has no edge separator) as fifht argument.

7.9.3 Libraries/Moodules

- Pillow (`https://pillow.readthedocs.io/en/stable/`)
  - Access to image manipulation functions.
- sys standard library (`https://docs.python.org/3/library/sys.html`)
  - Access to command line arguments.
- os standard library (`https://docs.python.org/3/library/os.html`)
  - Access to program name.

7.9.4 Notes

- Color map should have the same order as the palette used to index the sprites.

7.9.5 TODO

- The limitation to 8 colors will be overcome soon.

7.9.6 Author(s)

- Created by jgabaut on 24/02/2022.
- Modified by jgabaut on 24/02/2022.

7.10 sprites4curses/sprites.py File Reference

Program that parses pngs from a passed directory, to encode their color to a char per pixel.

Namespaces

- namespace sprites

  Program that parses pngs from a passed directory, to encode their color to a char per pixel.

Functions

- def sprites.usage ()

  Prints correct invocation.
- def sprites.color_distance (c1, c2)

  Calculates the distance in color between two rgb tuples.
- def sprites.create_palette (colors)

  Creates a palette by adding the colors passed as an array, then pads the palette.
- def sprites.convert_sprite (file)

  Takes a image file and converts each pixel to a char representation of its color (closest match to CHAR_MAP).
- def sprites.print_converted_sprites (direc)

  Takes a directory containing image file and calls convert_sprite on each one.
- def sprites.main (argv)

  Main program entry.

7.10.1 Detailed Description

Program that parses pngs from a passed directory, to encode their color to a char per pixel.

7.10.2 Description

The program supports 8 colors at the moment. The png parsing uses Pillow, and the mapping is done against a preset color list. The list is described in palette.gpl to aid in exporting images with the correct color indexing.

7.10.3 Libraries/Moodules

- Pillow (`https://pillow.readthedocs.io/en/stable/`)
  - Access to image manipulation functions.
- sys standard library (`https://docs.python.org/3/library/sys.html`)
  - Access to command line arguments.
- glob standard library (`https://docs.python.org/3/library/glob.html`)
  - Access to pattern expansion.
- re standard library (`https://docs.python.org/3/library/re.html`)
  - Access to regular expressions.
- os standard library (`https://docs.python.org/3/library/os.html`)
  - Access to program name.

7.10.4 Notes

- Color map should have the same order as the palette used to index the sprites.

7.10.5 TODO

- The limitation to 8 colors will be overcome soon.

7.10.6 Author(s)

- Created by jgabaut on 24/02/2022.
- Modified by jgabaut on 24/02/2022.

Index