sprites4curses

0.2.5

Generated by Doxygen 1.9.7

Chapter 1

Contributor Covenant Code of Conduct

1.1 Our Pledge

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone, regardless of age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

We pledge to act and interact in ways that contribute to an open, welcoming, diverse, inclusive, and healthy community.

1.2 Our Standards

Examples of behavior that contributes to a positive environment for our community include:

- Demonstrating empathy and kindness toward other people
- Being respectful of differing opinions, viewpoints, and experiences
- Giving and gracefully accepting constructive feedback
- Accepting responsibility and apologizing to those affected by our mistakes, and learning from the experience
- Focusing on what is best not just for us as individuals, but for the overall community

Examples of unacceptable behavior include:

- The use of sexualized language or imagery, and sexual attention or advances of any kind
- Trolling, insulting or derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or email address, without their explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

1.3 Enforcement Responsibilities

Community leaders are responsible for clarifying and enforcing our standards of acceptable behavior and will take appropriate and fair corrective action in response to any behavior that they deem inappropriate, threatening, offensive, or harmful.

Community leaders have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, and will communicate reasons for moderation decisions when appropriate.

1.4 Scope

This Code of Conduct applies within all community spaces, and also applies when an individual is officially representing the community in public spaces. Examples of representing our community include using an official e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event.

1.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported to the community leaders responsible for enforcement atAll complaints will be reviewed and investigated promptly and fairly.

All community leaders are obligated to respect the privacy and security of the reporter of any incident.

1.6 Enforcement Guidelines

Community leaders will follow these Community Impact Guidelines in determining the consequences for any action they deem in violation of this Code of Conduct:

1.6.1 1. Correction

Community Impact: Use of inappropriate language or other behavior deemed unprofessional or unwelcome in the community.

Consequence: A private, written warning from community leaders, providing clarity around the nature of the violation and an explanation of why the behavior was inappropriate. A public apology may be requested.

1.6.2 2. Warning

Community Impact: A violation through a single incident or series of actions.

Consequence: A warning with consequences for continued behavior. No interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, for a specified period of time. This includes avoiding interactions in community spaces as well as external channels like social media. Violating these terms may lead to a temporary or permanent ban.

1.6.3 3. Temporary Ban

Community Impact: A serious violation of community standards, including sustained inappropriate behavior.

Consequence: A temporary ban from any sort of interaction or public communication with the community for a specified period of time. No public or private interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, is allowed during this period. Violating these terms may lead to a permanent ban.

1.6.4 4. Permanent Ban

Community Impact: Demonstrating a pattern of violation of community standards, including sustained inappropriate behavior, harassment of an individual, or aggression toward or disparagement of classes of individuals.

Consequence: A permanent ban from any sort of public interaction within the community.

1.7 Attribution

This Code of Conduct is adapted from the `Contributor Covenant`, version 2.0, available at `https://www.contributor-covenant.org/version/2/0/code_of_conduct.html`.

Community Impact Guidelines were inspired by `Mozilla's code of conduct enforcement ladder`.

For answers to common questions about this code of conduct, see the FAQ at `https://www.contributor-covenant.org/faq`. Translations are available at `https://www.contributor-covenant.org/translations`.

Chapter 2

Documentation

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

Chapter 3

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 4

sprites4curses

A library of scripts and C functions to deal with sprites in ncurses.

4.1 Scripts

All of them have a dependency on Pillow to process images.

4.1.1 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.

4.1.2 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.

4.1.3 cut_sheet.py

This is a python script that cuts a single PNG spritesheet to a number of sprites, and puts them in the passed directory.

It expects as arguments the spritesheet file name, the output directory 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.

4.1.4 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.

4.2 animate.c and animate.h

This is a C library offering some functions to display an animation read from a formatted text file.

`animate\_sprites()` is useful in a initialised WINDOW, it boxes the window and displays the animation snugly.

`animate\_sprites\_at\_coords()` does the same, but has 2 more parameters to start displaying at any coord in a window big enough to fit the animation.

You can look at the demo.c program to see how you can request the animation after setup. The file format expected is compatible with sprites.py specs.

Since it needs support from terminal capabilities, it may return some errors if your terminal doesn't offer the needed options.

At the moment your solution is to change terminal or help investigate your issues by forcing the check to pass, I may add an unsafe option to do this in a later version.

4.2.1 demo.c

This is a demo program showing how to use the animate library functions. Check out its source code after running it!

- To run the C demo program, do:

4.2.1.1 The demo is meant to run with the provided file.

4.2.1.2 <tt>make; ./demo demofile.txt</tt>

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

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

4.3 Scripts usage

To use the python scripts you need to install Pillow:

4.3.0.0.1 <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:

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

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

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

4.3.0.0.4 <tt>python sheet_converter.py <sheet file> <sprite width> <sprite height> <separator thickness> <first sprite left corner X> <first sprite LC Y> > file.txt</tt>

- To run the sheet cutter script, give the sheet png file and the output directory:

4.3.0.0.5 <tt>python cut_sheet.py <sheet file> <output dir> <sprite width> <sprite height> <separator thickness> <first sprite left corner X> <first sprite LC Y></tt>

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

4.3.0.0.6 <tt>python png_resize.py <sprites directory> <sprite width> <sprite height></tt>

4.3.0.1 This overwrites the source pngs, so be careful.

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

4.3.1 palette.gpl

This is a GIMP palette file. It's used by the library to initialise the color pairs for curses to display the sprites. It's also useful in the first place for exporting PNG with the correct color alignment. Info on how to use it are in the palette-README.md file.

Chapter 5

Namespace Index

5.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 6

Data Structure Index

6.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 7

File Index

7.1 File List

Here is a list of all files with brief descriptions:

Chapter 8

Namespace Documentation

8.1 cut_sheet Namespace Reference

Program that cut a passed png spritesheet, and saves the sprites to a passed directory.

Functions

- usage ()

  Prints correct invocation.
- color_distance (c1, c2)

  Calculates the distance in color between two rgb tuples.
- cut_spritesheet (filename, output_directory, spriteSizeX, spriteSizeY, separatorSize, startX, startY)

  Converts a spritesheet to a set of individual sprite images.
- main (argv)

  Main program entry.

8.1.1 Detailed Description

Program that cut a passed png spritesheet, and saves the sprites to a passed directory.

8.1.2 Function Documentation

8.1.2.1 color_distance()

```
cut_sheet.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.

8.1.2.2 cut_spritesheet()

```
cut_sheet.cut_spritesheet (
              filename,
              output_directory,
              spriteSizeX,
              spriteSizeY,
              separatorSize,
              startX,
              startY )
```

Converts a spritesheet to a set of individual sprite images.

Parameters

filename	The input spritesheet file.
output_directory	The directory where output images will be saved.
spriteSizeX	The sprite width.
spriteSizeY	The sprite height.
separatorSize	Thickess of separator pixels.
startX	X coord of left corner of first sprite.
startY	Y coord of left corner of first sprite.

8.1.2.3 main()

```
cut_sheet.main (
              argv )
```

Main program entry.

8.1.2.4 usage()

```
cut_sheet.usage ( )
```

Prints correct invocation.

8.2 png_resize Namespace Reference

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

Functions

- usage ()

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

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

  Main program entry.

8.2.1 Detailed Description

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

8.2.2 Function Documentation

8.2.2.1 main()

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

Main program entry.

8.2.2.2 resize_sprites()

```
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.

8.2.2.3 usage()

```
png_resize.usage ( )
```

Prints correct invocation.

8.3 sheet_converter Namespace Reference

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

Functions

- usage ()

  Prints correct invocation.
- color_distance (c1, c2)

  Calculates the distance in color between two rgb tuples.
- convert_spritesheet (filename, spriteSizeX, spriteSizeY, separatorSize, startX, startY)

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

  Main program entry.

Variables

- str FILE_VERSION = "0.1.2"

  The file format version.

8.3.1 Detailed Description

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

8.3.2 Function Documentation

8.3.2.1 color_distance()

```
sheet_converter.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.

8.3.2.2 convert_spritesheet()

```
sheet_converter.convert_spritesheet (
              filename,
              spriteSizeX,
              spriteSizeY,
              separatorSize,
              startX,
              startY )
```

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.
startX	X coord of left corner of first sprite.
startY	Y coord of left corner of first sprite.

8.3.2.3 main()

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

Main program entry.

8.3.2.4 usage()

```
sheet_converter.usage ( )
```

Prints correct invocation.

8.3.3 Variable Documentation

8.3.3.1 FILE_VERSION

```
str sheet_converter.FILE_VERSION = "0.1.2"
```

The file format version.

8.4 sprites Namespace Reference

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

Functions

- usage ()

  Prints correct invocation.
- color_distance (c1, c2)

  Calculates the distance in color between two rgb tuples.
- convert_sprite (file)

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

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

  Main program entry.

Variables

- str FILE_VERSION = "0.1.2"

  The file format version.

8.4.1 Detailed Description

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

8.4.2 Function Documentation

8.4.2.1 color_distance()

```
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.

8.4.2.2 convert_sprite()

```
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.

8.4.2.3 main()

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

Main program entry.

8.4.2.4 print_converted_sprites()

```
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.

8.4.2.5 usage()

```
sprites.usage ( )
```

Prints correct invocation.

8.4.3 Variable Documentation

8.4.3.1 FILE_VERSION

```
str sprites.FILE_VERSION = "0.1.2"
```

The file format version.

Chapter 9

Data Structure Documentation

9.1 animate_args Struct Reference

```
#include <animate.h>
```

Data Fields

- int stop_thread

  Stops the thread when false.
- WINDOW ∗ win

  WINDOW to animate to.
- char sprites [MAXFRAMES][MAXROWS][MAXCOLS]

  Array for the animation.
- int frametime

  How many ms a frame will stay on screen after drawing.
- int num_frames

  How many frames the animation has.
- int frameheight

  Height of the frames.
- int framewidth

  Width of the frames.
- int startX

  Starting X value to print at.
- int startY

  Starting Y value to print at.

9.1.1 Field Documentation

9.1.1.1 frameheight

```
int animate_args::frameheight
```

Height of the frames.

9.1.1.2 frametime

```
int animate_args::frametime
```

How many ms a frame will stay on screen after drawing.

9.1.1.3 framewidth

```
int animate_args::framewidth
```

Width of the frames.

9.1.1.4 num_frames

```
int animate_args::num_frames
```

How many frames the animation has.

9.1.1.5 sprites

```
char animate_args::sprites[MAXFRAMES][MAXROWS][MAXCOLS]
```

Array for the animation.

9.1.1.6 startX

```
int animate_args::startX
```

Starting X value to print at.

9.1.1.7 startY

```
int animate_args::startY
```

Starting Y value to print at.

9.1.1.8 stop_thread

```
int animate_args::stop_thread
```

Stops the thread when false.

9.1.1.9 win

```
WINDOW* animate_args::win
```

WINDOW to animate to.

The documentation for this struct was generated from the following file:

- sprites4curses/animate_src/animate.h

Chapter 10

File Documentation

10.1 sprites4curses/animate_src/animate.c File Reference

```
#include "animate.h"
```

Functions

- void s4c_printVersionToFile (FILE ∗f)

  Prints animate version.
- void init_s4c_color_pairs (FILE ∗palette)

  Initialises all the needed color pairs for animate, from the palette file.
- int load_sprites (char sprites[MAXFRAMES][MAXROWS][MAXCOLS], FILE ∗f, int rows, int columns)

  Takes an empty 3D char array (frame, height, width) and a file to read the sprites from.
- int animate_sprites (char sprites[MAXFRAMES][MAXROWS][MAXCOLS], WINDOW ∗w, int repetitions, int frametime, int num_frames, int frameheight, int framewidth)

  Calls animate_sprites_at_coords() with 0,0 as starting coordinates.
- int animate_sprites_at_coords (char sprites[MAXFRAMES][MAXROWS][MAXCOLS], WINDOW ∗w, int repetitions, int frametime, int num_frames, int frameheight, int framewidth, int startX, int startY)

  Takes a WINDOW pointer to print into and a string for the file passed.
- void ∗ animate_sprites_thread_at (void ∗args_ptr)

  Takes a void pointer, to be cast to animate_args∗, containing parameters to animate a sprite in a WINDOW, using a separate thread.
- int animate_rangeof_sprites_at_coords (char sprites[MAXFRAMES][MAXROWS][MAXCOLS], WINDOW ∗w, int fromFrame, int toFrame, int repetitions, int frametime, int num_frames, int frameheight, int framewidth, int startX, int startY)

  Takes a WINDOW pointer to print into and a string for the file passed.

10.1.1 Function Documentation

10.1.1.1 animate_rangeof_sprites_at_coords()

```
int animate_rangeof_sprites_at_coords (
            char sprites[MAXFRAMES][MAXROWS][MAXCOLS],
            WINDOW * w,
            int fromFrame,
            int toFrame,
            int repetitions,
            int frametime,
            int num_frames,
            int frameheight,
            int framewidth,
            int startX,
            int startY )
```

Takes a WINDOW pointer to print into and a string for the file passed.

Loads sprites from the file and displays a range of them in the passed window if it is big enough. File format should have a sprite line on each line, or be a valid array definition. Color-character map is define in print_spriteline(). Sets all the frames to the passed array.

See also

print_spriteline()

Parameters

sprites	The sprites array.
w	The window to print into.
repetition	The number of times the animation will be cycled through.
frametime	How many mseconds each frame is displayed.
num_frames	How many frames the animation will have.
frameheight	Height of the frame.
framewidth	Width of the frame.
startY	Y coord of the window to start printing to.
startY	X coord of the window to start printing to.

See also

S4C_ERR_CURSOR

S4C_ERR_SMALL_WIN

Returns

1 if successful, a negative value for errors.

10.1.1.2 animate_sprites()

```
int animate_sprites (
            char sprites[MAXFRAMES][MAXROWS][MAXCOLS],
            WINDOW * w,
            int repetitions,
            int frametime,
            int num_frames,
            int frameheight,
            int framewidth )
```

Calls animate_sprites_at_coords() with 0,0 as starting coordinates.

See also

animate_sprites_at_coords()

Parameters

sprites	The sprites array.
w	The window to print into.
repetition	The number of times the animation will be cycled through.
frametime	How many mseconds each frame is displayed.
num_frames	How many frames the animation will have.
frameheight	Height of the frame.
framewidth	Width of the frame.

See also

S4C_ERR_CURSOR

S4C_ERR_SMALL_WIN

Returns

1 if successful, a negative value for errors.

10.1.1.3 animate_sprites_at_coords()

```
int animate_sprites_at_coords (
            char sprites[MAXFRAMES][MAXROWS][MAXCOLS],
            WINDOW * w,
            int repetitions,
            int frametime,
            int num_frames,
            int frameheight,
            int framewidth,
            int startX,
            int startY )
```

Takes a WINDOW pointer to print into and a string for the file passed.

Loads sprites from the file and displays them in the passed window if it is big enough. File format should have a sprite line on each line, or be a valid array definition. Color-character map is define in print_spriteline(). Sets all the frames to the passed array.

See also

print_spriteline()

Parameters

sprites	The sprites array.
w	The window to print into.
repetition	The number of times the animation will be cycled through.
frametime	How many mseconds each frame is displayed.
num_frames	How many frames the animation will have.
frameheight	Height of the frame.
framewidth	Width of the frame.
startY	Y coord of the window to start printing to.
startY	X coord of the window to start printing to.

See also

S4C_ERR_CURSOR

S4C_ERR_SMALL_WIN

Returns

1 if successful, a negative value for errors.

10.1.1.4 animate_sprites_thread_at()

```
void * animate_sprites_thread_at (
            void * args_ptr )
```

Takes a void pointer, to be cast to animate_args*, containing parameters to animate a sprite in a WINDOW, using a separate thread.

See also

animate_args

10.1.1.5 init_s4c_color_pairs()

```
void init_s4c_color_pairs (
            FILE * palette )
```

Initialises all the needed color pairs for animate, from the palette file.

Parameters

palette	The palette file to read the colors from.

10.1.1.6 load_sprites()

```
int load_sprites (
            char sprites[MAXFRAMES][MAXROWS][MAXCOLS],
            FILE * f,
            int rows,
            int columns )
```

Takes an empty 3D char array (frame, height, width) and a file to read the sprites from.

Checks if the file version is compatible with the current reader version, otherwise returns a negative error value. Closes file pointer before returning. File format should have a sprite line on each line. Sets all the frames to the passed array.

Parameters

sprites	The char array to fill with all the frames.
f	The file to read the sprites from.
rows	The number of rows in each sprite.
columns	The number of columns in each sprite.

See also

S4C_ERR_FILEVERSION

S4C_ERR_LOADSPRITES

Returns

A negative error value if loading fails or the number of sprites read.

10.1.1.7 s4c_printVersionToFile()

```
void s4c_printVersionToFile (
            FILE * f )
```

Prints animate version.

10.2 sprites4curses/animate_src/animate.h File Reference

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <stdlib.h>
#include <ncurses.h>
#include <pthread.h>
```

Data Structures

- struct animate_args

Macros

- #define S4C_ANIMATE_VERSION "0.2.5"
- #define S4C_PRINTVERSION() s4c_printVersionToFile(stdout);
- #define MAX_COLORS 256
- #define MAX_COLOR_NAME_LEN 256

  Defines max size for the name strings in palette.gpl.
- #define S4C_BLACK 9
- #define S4C_RED 10
- #define S4C_BRIGHT_GREEN 11
- #define S4C_BRIGHT_YELLOW 12
- #define S4C_BLUE 13
- #define S4C_MAGENTA 14
- #define S4C_CYAN 15
- #define S4C_WHITE 16
- #define S4C_ORANGE 17
- #define S4C_LIGHT_BROWN 18
- #define S4C_DARK_BROWN 19
- #define S4C_PURPLE 20
- #define S4C_DARK_GREEN 21
- #define S4C_LIGHT_YELLOW 22
- #define S4C_LIGHT_BLUE 23
- #define MAX_LINE_LENGTH 1024
- #define MAXFRAMES 121

  Defines the maximum number of sprites.
- #define MAXROWS 26

  Defines the maximum number of rows per sprite.
- #define MAXCOLS 84

  Defines the maximum number of colums per sprite.
- #define S4C_ERR_SMALL_WIN -1

  Defines the error value for a window too small for the animation.
- #define S4C_ERR_FILEVERSION -2

  Defines the error value for a file version mismatch.
- #define S4C_ERR_LOADSPRITES -3

  Defines the error value for a failure while loading sprites.
- #define S4C_ERR_TERMCOLOR -4

  Defines the error value for when the terminal doesn't support colors.

- #define S4C_ERR_TERMCHANGECOLOR -5

  Defines the error value for when the terminal doesn't support changing colors.
- #define S4C_ERR_CURSOR -6

  Defines the error value for when the terminal doesn't support changing cursor visibility.
- #define S4C_ERR_RANGE -7

  Defines the error value for invalid range requests for animate_rangeof_sprites_at_coords().

Typedefs

- typedef struct animate_args animate_args

Functions

- void s4c_printVersionToFile (FILE ∗f)

  Prints animate version.
- void init_s4c_color_pairs (FILE ∗palette_file)

  Initialises all the needed color pairs for animate, from the palette file.
- int load_sprites (char sprites[MAXFRAMES][MAXROWS][MAXCOLS], FILE ∗file, int rows, int columns)

  Takes an empty 3D char array (frame, height, width) and a file to read the sprites from.
- int animate_sprites (char sprites[MAXFRAMES][MAXROWS][MAXCOLS], WINDOW ∗w, int repetitions, int frametime, int num_frames, int frameheight, int framewidth)

  Calls animate_sprites_at_coords() with 0,0 as starting coordinates.
- void ∗ animate_sprites_thread_at (void ∗animate_args)

  Takes a void pointer, to be cast to animate_args∗, containing parameters to animate a sprite in a WINDOW, using a separate thread.
- int animate_sprites_at_coords (char sprites[MAXFRAMES][MAXROWS][MAXCOLS], WINDOW ∗w, int repetitions, int frametime, int num_frames, int frameheight, int framewidth, int startX, int startY)

  Takes a WINDOW pointer to print into and a string for the file passed.
- int animate_rangeof_sprites_at_coords (char sprites[MAXFRAMES][MAXROWS][MAXCOLS], WINDOW ∗w, int fromFrame, int toFrame, int repetitions, int frametime, int num_frames, int frameheight, int framewidth, int startX, int startY)

  Takes a WINDOW pointer to print into and a string for the file passed.

10.2.1 Macro Definition Documentation

10.2.1.1 MAX_COLOR_NAME_LEN

```
#define MAX_COLOR_NAME_LEN 256
```

Defines max size for the name strings in palette.gpl.

10.2.1.2 MAX_COLORS

```
#define MAX_COLORS 256
```

10.2.1.3 MAX_LINE_LENGTH

```
#define MAX_LINE_LENGTH 1024
```

10.2.1.4 MAXCOLS

```
#define MAXCOLS 84
```

Defines the maximum number of colums per sprite.

10.2.1.5 MAXFRAMES

```
#define MAXFRAMES 121
```

Defines the maximum number of sprites.

10.2.1.6 MAXROWS

```
#define MAXROWS 26
```

Defines the maximum number of rows per sprite.

10.2.1.7 S4C_ANIMATE_VERSION

```
#define S4C_ANIMATE_VERSION "0.2.5"
```

10.2.1.8 S4C_BLACK

```
#define S4C_BLACK 9
```

10.2.1.9 S4C_BLUE

```
#define S4C_BLUE 13
```

10.2.1.10 S4C_BRIGHT_GREEN

```
#define S4C_BRIGHT_GREEN 11
```

10.2.1.11 S4C_BRIGHT_YELLOW

```
#define S4C_BRIGHT_YELLOW 12
```

10.2.1.12 S4C_CYAN

```
#define S4C_CYAN 15
```

10.2.1.13 S4C_DARK_BROWN

```
#define S4C_DARK_BROWN 19
```

10.2.1.14 S4C_DARK_GREEN

```
#define S4C_DARK_GREEN 21
```

10.2.1.15 S4C_ERR_CURSOR

```
#define S4C_ERR_CURSOR -6
```

Defines the error value for when the terminal doesn't support changing cursor visibility.

10.2.1.16 S4C_ERR_FILEVERSION

```
#define S4C_ERR_FILEVERSION -2
```

Defines the error value for a file version mismatch.

10.2.1.17 S4C_ERR_LOADSPRITES

```
#define S4C_ERR_LOADSPRITES -3
```

Defines the error value for a failure while loading sprites.

10.2.1.18 S4C_ERR_RANGE

```
#define S4C_ERR_RANGE -7
```

Defines the error value for invalid range requests for animate_rangeof_sprites_at_coords().

10.2.1.19 S4C_ERR_SMALL_WIN

```
#define S4C_ERR_SMALL_WIN -1
```

Defines the error value for a window too small for the animation.

10.2.1.20 S4C_ERR_TERMCHANGECOLOR

```
#define S4C_ERR_TERMCHANGECOLOR -5
```

Defines the error value for when the terminal doesn't support changing colors.

10.2.1.21 S4C_ERR_TERMCOLOR

```
#define S4C_ERR_TERMCOLOR -4
```

Defines the error value for when the terminal doesn't support colors.

10.2.1.22 S4C_LIGHT_BLUE

```
#define S4C_LIGHT_BLUE 23
```

10.2.1.23 S4C_LIGHT_BROWN

```
#define S4C_LIGHT_BROWN 18
```

10.2.1.24 S4C_LIGHT_YELLOW

```
#define S4C_LIGHT_YELLOW 22
```

10.2.1.25 S4C_MAGENTA

```
#define S4C_MAGENTA 14
```

10.2.1.26 S4C_ORANGE

```
#define S4C_ORANGE 17
```

10.2.1.27 S4C_PRINTVERSION

```
#define S4C_PRINTVERSION( ) s4c_printVersionToFile(stdout);
```

10.2.1.28 S4C_PURPLE

```
#define S4C_PURPLE 20
```

10.2.1.29 S4C_RED

```
#define S4C_RED 10
```

10.2.1.30 S4C_WHITE

```
#define S4C_WHITE 16
```

10.2.2 Typedef Documentation

10.2.2.1 animate_args

```
typedef struct animate_args animate_args
```

10.2.3 Function Documentation

10.2.3.1 animate_rangeof_sprites_at_coords()

```
int animate_rangeof_sprites_at_coords (
            char sprites[MAXFRAMES][MAXROWS][MAXCOLS],
            WINDOW * w,
            int fromFrame,
            int toFrame,
            int repetitions,
            int frametime,
            int num_frames,
            int frameheight,
            int framewidth,
            int startX,
            int startY )
```

Takes a WINDOW pointer to print into and a string for the file passed.

Loads sprites from the file and displays a range of them in the passed window if it is big enough. File format should have a sprite line on each line, or be a valid array definition. Color-character map is define in print_spriteline(). Sets all the frames to the passed array.

See also

print_spriteline()

Parameters

sprites	The sprites array.
w	The window to print into.
repetition	The number of times the animation will be cycled through.
frametime	How many mseconds each frame is displayed.
num_frames	How many frames the animation will have.
frameheight	Height of the frame.
framewidth	Width of the frame.
startY	Y coord of the window to start printing to.
startY	X coord of the window to start printing to.

See also

S4C_ERR_CURSOR

S4C_ERR_SMALL_WIN

Returns

1 if successful, a negative value for errors.

10.2.3.2 animate_sprites()

```
int animate_sprites (
            char sprites[MAXFRAMES][MAXROWS][MAXCOLS],
            WINDOW * w,
            int repetitions,
            int frametime,
            int num_frames,
            int frameheight,
            int framewidth )
```

Calls animate_sprites_at_coords() with 0,0 as starting coordinates.

See also

animate_sprites_at_coords()

Parameters

sprites	The sprites array.
w	The window to print into.
repetition	The number of times the animation will be cycled through.
frametime	How many mseconds each frame is displayed.
num_frames	How many frames the animation will have.
frameheight	Height of the frame.
framewidth	Width of the frame.

See also

S4C_ERR_CURSOR

S4C_ERR_SMALL_WIN

Returns

1 if successful, a negative value for errors.

10.2.3.3 animate_sprites_at_coords()

```
int animate_sprites_at_coords (
            char sprites[MAXFRAMES][MAXROWS][MAXCOLS],
            WINDOW * w,
            int repetitions,
            int frametime,
            int num_frames,
            int frameheight,
            int framewidth,
            int startX,
            int startY )
```

Takes a WINDOW pointer to print into and a string for the file passed.

Loads sprites from the file and displays them in the passed window if it is big enough. File format should have a sprite line on each line, or be a valid array definition. Color-character map is define in print_spriteline(). Sets all the frames to the passed array.

See also

print_spriteline()

Parameters

sprites	The sprites array.
w	The window to print into.
repetition	The number of times the animation will be cycled through.
frametime	How many mseconds each frame is displayed.
num_frames	How many frames the animation will have.
frameheight	Height of the frame.
framewidth	Width of the frame.
startY	Y coord of the window to start printing to.
startY	X coord of the window to start printing to.

See also

S4C_ERR_CURSOR
S4C_ERR_SMALL_WIN

Returns

1 if successful, a negative value for errors.

10.2.3.4 animate_sprites_thread_at()

```
void * animate_sprites_thread_at (
            void * args_ptr )
```

Takes a void pointer, to be cast to animate_args*, containing parameters to animate a sprite in a WINDOW, using a separate thread.

See also

animate_args

10.2.3.5 init_s4c_color_pairs()

```
void init_s4c_color_pairs (
             FILE * palette )
```

Initialises all the needed color pairs for animate, from the palette file.

Parameters

palette	The palette file to read the colors from.

10.2.3.6 load_sprites()

```
int load_sprites (
             char sprites[MAXFRAMES][MAXROWS][MAXCOLS],
             FILE * f,
             int rows,
             int columns )
```

Takes an empty 3D char array (frame, height, width) and a file to read the sprites from.

Checks if the file version is compatible with the current reader version, otherwise returns a negative error value. Closes file pointer before returning. File format should have a sprite line on each line. Sets all the frames to the passed array.

Parameters

sprites	The char array to fill with all the frames.
f	The file to read the sprites from.
rows	The number of rows in each sprite.
columns	The number of columns in each sprite.

See also

S4C_ERR_FILEVERSION

S4C_ERR_LOADSPRITES

Returns

A negative error value if loading fails or the number of sprites read.

10.2.3.7 s4c_printVersionToFile()

```
void s4c_printVersionToFile (
              FILE * f )
```

Prints animate version.

10.3 animate.h

Go to the documentation of this file.

```c
#ifndef S4C_ANIMATE_H
#define S4C_ANIMATE_H
#include <stdio.h>

#define S4C_ANIMATE_VERSION "0.2.5"
void s4c_printVersionToFile(FILE* f);

#define S4C_PRINTVERSION() s4c_printVersionToFile(stdout); //Prints version to stdout

#include <string.h>
#include <ctype.h>
#include <stdlib.h>
#include <ncurses.h>
#include <pthread.h>

#define MAX_COLORS 256
#define MAX_COLOR_NAME_LEN 256
// These define constants for the colors prepared by init_s4c_color_pairs().
#define S4C_BLACK 9
#define S4C_RED 10
#define S4C_BRIGHT_GREEN 11
#define S4C_BRIGHT_YELLOW 12
#define S4C_BLUE 13
#define S4C_MAGENTA 14
#define S4C_CYAN 15
#define S4C_WHITE 16
#define S4C_ORANGE 17
#define S4C_LIGHT_BROWN 18
#define S4C_DARK_BROWN 19
#define S4C_PURPLE 20
#define S4C_DARK_GREEN 21
#define S4C_LIGHT_YELLOW 22
#define S4C_LIGHT_BLUE 23

#define MAX_LINE_LENGTH 1024

#define MAXFRAMES 121
#define MAXROWS 26
#define MAXCOLS 84
#define S4C_ERR_SMALL_WIN -1
#define S4C_ERR_FILEVERSION -2
#define S4C_ERR_LOADSPRITES -3
#define S4C_ERR_TERMCOLOR -4
#define S4C_ERR_TERMCHANGECOLOR -5
#define S4C_ERR_CURSOR -6
#define S4C_ERR_RANGE -7
/*
 * Holds arguments for a call to animate_sprites_thread_at().
 * WIP.
 */
typedef struct animate_args {
    int stop_thread;
    WINDOW* win;
    char sprites[MAXFRAMES][MAXROWS][MAXCOLS];
    int frametime;
    int num_frames;
    int frameheight;
    int framewidth;
    int startX;
    int startY;
} animate_args;

void init_s4c_color_pairs(FILE* palette_file);
static void print_spriteline(WINDOW* win, char* line, int curr_line_num, int line_length, int startX);
int load_sprites(char sprites[MAXFRAMES][MAXROWS][MAXCOLS], FILE* file, int rows, int columns);
int animate_sprites(char sprites[MAXFRAMES][MAXROWS][MAXCOLS], WINDOW* w, int repetitions, int frametime, int num_frames, int frameheight, int framewidth);
void *animate_sprites_thread_at(void *animate_args);
```

```
int animate_sprites_at_coords(char sprites[MAXFRAMES][MAXROWS][MAXCOLS], WINDOW* w, int repetitions, int frametime, int num_frames, int frameheight, int framewidth, int startX, int startY);
int animate_rangeof_sprites_at_coords(char sprites[MAXFRAMES][MAXROWS][MAXCOLS], WINDOW* w, int fromFrame, int toFrame, int repetitions, int frametime, int num_frames, int frameheight, int framewidth, int startX, int startY);

#endif
```

10.4 sprites4curses/CODE_OF_CONDUCT.md File Reference

10.5 sprites4curses/demo_src/demo.c File Reference

```
#include <unistd.h>
#include <stdlib.h>
#include <locale.h>
#include "../animate_src/animate.h"
```

Macros

- #define DEMOFRAMES 30

  Defines the number of sprites in the demo.
- #define DEMOROWS 18

  Defines the maximum number of rows per sprite.
- #define DEMOCOLS 18

  Defines the maximum number of colums per sprite.
- #define DEMOFRAMETIME 67

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

Functions

- void usage (char ∗progname)
- int demo (FILE ∗mainthread_file, FILE ∗newthread_file)
- int main (int argc, char ∗∗argv)

10.5.1 Macro Definition Documentation

10.5.1.1 DEMOCOLS

```
#define DEMOCOLS 18
```

Defines the maximum number of colums per sprite.

10.5.1.2 DEMOFRAMES

```
#define DEMOFRAMES 30
```

Defines the number of sprites in the demo.

10.5.1.3 DEMOFRAMETIME

```
#define DEMOFRAMETIME 67
```

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

10.5.1.4 DEMOROWS

```
#define DEMOROWS 18
```

Defines the maximum number of rows per sprite.

10.5.2 Function Documentation

10.5.2.1 demo()

```
int demo (
             FILE * mainthread_file,
             FILE * newthread_file )
```

10.5.2.2 main()

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

10.5.2.3 usage()

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

10.6 sprites4curses/documentation/README.md File Reference

10.7 sprites4curses/README.md File Reference

10.8 sprites4curses/palette-README.md File Reference

10.9 sprites4curses/scripts/cut_sheet.py File Reference

Program that cut a passed png spritesheet, and saves the sprites to a passed directory.

Namespaces

- namespace cut_sheet

  Program that cut a passed png spritesheet, and saves the sprites to a passed directory.

Functions

- cut_sheet.usage ()

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

  Calculates the distance in color between two rgb tuples.
- cut_sheet.cut_spritesheet (filename, output_directory, spriteSizeX, spriteSizeY, separatorSize, startX, startY)

  Converts a spritesheet to a set of individual sprite images.
- cut_sheet.main (argv)

  Main program entry.

10.9.1 Detailed Description

Program that cut a passed png spritesheet, and saves the sprites to a passed directory.

10.9.2 Description

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, then the output directory, the sprite width, the sprite height, separator size (thickness), X coord of left corner of the first sprite (0 if sheet has no edge separator), Y coord of left corner of the first sprite (0 if sheet has no edge separator).

10.9.3 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.
- math standard library (https://docs.python.org/3/library/math.html)
  - Access to sqrt.

10.9.4 Notes

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

10.9.5 TODO

- Check if the encoded value is exceeding latin literals.

10.9.6 Author(s)

- Created by jgabaut on 04/03/2023.
- Modified by jgabaut on 09/03/2023.

10.10 sprites4curses/scripts/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

- png_resize.usage ()

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

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

  Main program entry.

10.10.1 Detailed Description

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

10.10.2 Description

The program overwrites the passed pngs with the resized version.

Program expects the spritesheet filename as first argument, then the sprite width, the sprite heigth.

10.10.3 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.
- os standard library (`https://docs.python.org/3/library/os.html`)
  - Access to program name.

10.10.4 Notes

- The pngs are overwritten by default.

10.10.5 TODO

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

10.10.6 Author(s)

- Created by jgabaut on 24/02/2023.
- Modified by jgabaut on 04/03/2023.

10.11 sprites4curses/scripts/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

- sheet_converter.usage ()

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

  Calculates the distance in color between two rgb tuples.
- sheet_converter.convert_spritesheet (filename, spriteSizeX, spriteSizeY, separatorSize, startX, startY)

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

  Main program entry.

Variables

- str sheet_converter.FILE_VERSION = "0.1.2"

  The file format version.

10.11.1 Detailed Description

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

10.11.2 Description

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, then the sprite width, the sprite height, separator size (thickness), X coord of left corner of the first sprite (0 if sheet has no edge separator), Y coord of left corner of the first sprite (0 if sheet has no edge separator).

10.11.3 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.
- math standard library (`https://docs.python.org/3/library/math.html`)
  - Access to sqrt.

10.11.4 Notes

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

10.11.5 TODO

- Check if the encoded value is exceeding latin literals.

10.11.6 Author(s)

- Created by jgabaut on 24/02/2023.
- Modified by jgabaut on 04/03/2023.

10.12 sprites4curses/scripts/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

- sprites.usage ()

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

  Calculates the distance in color between two rgb tuples.
- sprites.convert_sprite (file)

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

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

  Main program entry.

Variables

- str sprites.FILE_VERSION = "0.1.2"

  The file format version.

10.12.1 Detailed Description

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

10.12.2 Description

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.

10.12.3 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.
- math standard library (https://docs.python.org/3/library/math.html)
  - Access to sqrt.

10.12.4 Notes

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

10.12.5 TODO

- Check if the encoded value is exceeding latin literals.

10.12.6 Author(s)

- Created by jgabaut on 24/02/2023.
- Modified by jgabaut on 09/03/2023.

Index