Mastermind

Generated by Doxygen 1.8.8

Fri Apr 17 2015 22:16:54

Contents

Chapter 1

Main Page

Open Source MasterMind implementation for School homework

Chapter 2

DESIGN

Design alternative 1:

```
1 +---+---+---+---+  +-------+
2 |   |   |   |   |  | *   * |
3 | 1 | 2 | 0 | 1 |  |       |
4 |   |   |   |   |  | .     |
5 +---+---+---+---+  +-------+
6 |   |   |   |   |  |       |
7 |   |   |   |   |  |       |
8 |   |   |   |   |  |       |
9 +---+---+---+---+  +-------+
```

Design alternative 2:

```
1 +---+---+---+---+  +-----+-----+-----+
2 | 1 | 2 | 0 | 1 |  | + 2 | = 1 | - 1 |
3 +---+---+---+---+  +-----+-----+-----+
4 |   |   |   |   |  |     |     |     |
5 +---+---+---+---+  +-----+-----+-----+
```

Chapter 3

INSTALLATION

First you need to clone the repo and init its submodules

```
git clone https://github.com/lejenome/mastermind.git
cd mastermind
git submodule update --init # ext/ submodule is need for android, windows builds
```

LINUX

Dependencies: `readline, GNU getopt, libncurses5-dev, sdl2, sdl2_ttf, cmake >= 3.0` To install on Ubuntu, you need ppa that provides a cmake 3.x vez:

```
apt-add-repository ppa:george-edison55/cmake-3.x
apt-get update
apt-get install cmake libreadline-dev libncurses5 libsdl2-2.0 libsdl2-ttf-dev
```

To install on ArchLinux:

```
pacman -S cmake sdl2 sdl2_ttf ncurses readline
```

Next, to build:

```
mkdir -f build && cd build
cmake ..
make
```

To build and run on same dir, you need to set install prefix to build dir until the build instructions are fixed.

```
mkdir -f build && cd build
cmake -DCMAKE_INSTALL_PREFIX=. ..
make
./mastermindcli
./mastermindsdl
```

OS X

Install readline, libncurses, sdl2, sdl2_ttf, cmake from brew

Next, to build:

```
mkdir -f build && cd build
cmake .. -DUSE_LOCALE=OFF
cmake --build .
```

WINDOWS

Build is supported using mingw-w64 on all supported hosts (linux, windows,...), ms visual studio compiler support is still not usable.

```
mkdir build
cd build
cmake -G "MinGW Makefiles" ..
cmake --build . -DUSE_LOCALE=OFF
```

then you need to copy `../ext/mingw/bin<BUILD_ARCH>/*.dll` libs to build dir.

on same linux platforms, cmake builds with mingw integration are provided. Here is a demo of a x86_64 mastermind build on ArchLinux using packages provided on AUR:

```
yaourt -S mingw-w64-sdl2 mingw-w64-sdl2_ttf mingw-w64-readline mingw-w64-pdcurses mingw-w64-cmake mingw-w64-libiconv
mkdir -f build && cdc build
x86_64-w64-mingw32-cmake ..
make
```

ANDROID

First install Android NDK and SDK, and install Ant.

Download SDL2, SDL2_ttf and FreeType lastest sources archives and extract them.

then run:

```
mkdir -p build && cd build
cmake -DUSE_LOCALE=OFF ..
ln -s <PATH_TO_SDL2_SOURCE_DIR> jni/SDL
ln -s <PATH_TO_SDL2_ttf_SOURCE_DIR> jni/SDL_ttf
ln -s <PATH_TO_FreeType_SOURCE_DIR> jni/SDL_ttf/freetype
ndk-build
# you may need to update project target
# e.g: android update project -p . -t android-20
ant release
# gen key and sign apk ile with it, or use your own key
keytool -genkey -v -keystore my-release-key.keystore -alias alias_name -keyalg RSA -keysize 2048 -validity 10000
jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore my-release-key.keystore bin/MasterMind-release-unaligned.apk alias_name
# or just use android debug key at $HOME/.android/debug.keystore with alias androiddebugkey and password android
zipalign -v -f 4 bin/MasterMind-release-unaligned.apk bin/MasterMind.apk
```

the final apk file is bin/MasterMind.apk

NOTE: There is a known issue on SDL2 code caused by the bad GLES 2.0 support. A temporarily solution until it get fixed on SDL2 code, is to disable GLES 2.0 support, you need to modify `jni/SDL/include/SDL_config_android.h` so `SDL_VIDEO_OPENGL_ES2` is set to 0 :

```
#define SDL_VIDEO_OPENGL_ES2 0
```

You may need to set `SDL_VIDEO_RENDER_OGL_ES2` to 0 too :

```
#define SDL_VIDEO_RENDER_OGL_ES2 0
```

then rebuild libs and apk starting from ndk-build instruction.

Chapter 4

TODO

- [x] intl using gettext
- [x] configs
- [x] saving session state
- [x] cli interface
- [x] command auto complete
- [x] subcommand auto complete
- [x] SDL backend
- [x] MM_{GUESSES,COLORS,HOLES}_MAX support
- [x] (do/don't) repeat colors on combination (remise / sans remise)
- [x] saving scores, history
- [x] account support
- [x] cli options support
- [x] color scheme support on sdl
- [x] zsh & bash completion
- [x] windows build/packaging
- [x] Mac OS X build/packaging
- [x] Android build/packaging
- [x] Doxygen HTML and PDF docs
- [x] mastermindcli man page
- [ ] cli colored output
- [ ] ncurses backend
- [ ] build for linux/Mac/Win/Android/iOS/...
- [ ] package it for ubuntu/archlinux/windows/android/tizen
- [ ] multiplayer support
- [ ] cli backend table design themes support
- [ ] compile to NaCL and emscript

Chapter 5

config

- *colors*: Integer: `6`: number of colors
- *holes*: Integer: `4`: number of holes
- *guesses*: Integer: `12`: number of guesses
- *remise* : Boolean: `0`: color repeated only once on combination
- *account*: String: `default`: account name
- *save_on_exit*: Boolean: `0`: save session on exit
- *save_on_play*: Boolean: `0`: save session on every played guess

Chapter 6

learn

Learning sources used on mastermind homework developement.

Online Docs/Tutos:

- Make: 1”
- Readline: 1”
- SDL: 1”
- Gettext: 1”
- NSIS: 1”
- Doxygen: 1"

Commands and tools:

- man command:

```
# examples:
man 0 stdlib.h
man 3 strdup
man 3 SDL_Surface
```

- info command:

```
# example:
info Make
```

- cling (LLVM based interactive C/C++ interpreter): site

Chapter 7

Data Structure Index

7.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 8

File Index

8.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 9

Data Structure Documentation

9.1 cmd_t Struct Reference

command object containing its name, its function and args

```
#include <cli-cmd.h>
```

Data Fields

- char ∗ n

  name of command
- int(∗ e)(const char, const char ∗∗, mm_session ∗)

  function to excute
- char s

  short arg name else 0
- char ∗ l

  long arg name or NULL
- uint8_t a

  nbre of max options args accpeted

9.1.1 Detailed Description

command object containing its name, its function and args

Definition at line 24 of file cli-cmd.h.

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

- cli-cmd.h

9.2 mm_conf_bool_t Struct Reference

configuration option od type boolean (conf.bool.∗)

```
#include <core.h>
```

Data Fields

- uint8_t type

configuration type: MM_CONF_BOOL

- char ∗ name

  configuration name

- uint8_t val

  0 or 1

9.2.1 Detailed Description

configuration option od type boolean (conf.bool.∗)

Definition at line 85 of file core.h.

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

- core.h

9.3 mm_conf_int_t Struct Reference

configuration option of type integer (conf.nbre.∗)

```
#include <core.h>
```

Data Fields

- uint8_t type

  configuration type: MM_CONF_INT

- char ∗ name

  configuration name

- int val

  configuration value

- int min

  minimal integer value configuration value acceptes

- int max

  maximal integer value configuration value acceptes

9.3.1 Detailed Description

configuration option of type integer (conf.nbre.∗)

Definition at line 68 of file core.h.

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

- core.h

9.4 mm_conf_str_t Struct Reference

configuration option of type string (conf.str.∗)

```
#include <core.h>
```

Data Fields

- uint8_t type

  configuration typr: MM_CONF_STR
- char ∗ name

  configuration type
- char ∗ val

  configuration value
- uint8_t len

  length of string in value

9.4.1 Detailed Description

configuration option of type string (conf.str.∗)

Definition at line 77 of file core.h.

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

- core.h

9.5 mm_conf_t Union Reference

general configuration type

```
#include <core.h>
```

Data Fields

- uint8_t type

  configuration type, share with all configurations
- mm_conf_str_t str

  string configuration data
- mm_conf_bool_t bool

  boolean configration data
- mm_conf_int_t nbre

  integer configuration data

9.5.1 Detailed Description

general configuration type

Definition at line 92 of file core.h.

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

- core.h

9.6 mm_config Struct Reference

contains current session configuration

```
#include <core.h>
```

Data Fields

- uint8_t guesses

  max guesses on panel
- uint8_t colors

  max nbre of colors
- uint8_t holes

  nbre of holes (items) in a combination
- uint8_t remise

  do/don't repeat color on combination

9.6.1 Detailed Description

contains current session configuration

Definition at line 22 of file core.h.

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

- core.h

9.7 mm_guess Struct Reference

the guess object, containing the combination by the user and its score

```
#include <core.h>
```

Data Fields

- uint8_t ∗ combination

  given combination (guess)
- uint8_t inplace

  nbre of items on right place
- uint8_t insecret

  nbre of items on secret but not inplace

9.7.1 Detailed Description

the guess object, containing the combination by the user and its score

Definition at line 30 of file core.h.

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

- core.h

9.8 mm_score_t Struct Reference

session score, containing score value and account name

```
#include <core.h>
```

Data Fields

- long unsigned score

  score value
- char ∗ account

  account name

9.8.1 Detailed Description

session score, containing score value and account name

Definition at line 55 of file core.h.

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

- core.h

9.9 mm_scores_t Struct Reference

top scores history

```
#include <core.h>
```

Data Fields

- mm_score_t ∗ T

  top scores array
- unsigned max

  max number of top scores to store
- unsigned len

  current number of top scores stores

9.9.1 Detailed Description

top scores history

Definition at line 61 of file core.h.

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

- core.h

9.10 mm_secret Struct Reference

contains the secret generated at the beginning of session and the freq of its colors.

```
#include <core.h>
```

Data Fields

- uint8_t ∗ val

  len: config->holes
- uint8_t ∗ freq

  len: config->colors

9.10.1 Detailed Description

contains the secret generated at the beginning of session and the freq of its colors.

Definition at line 38 of file core.h.

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

- core.h

9.11 mm_session Struct Reference

the session object containing the secret object, the inputed guesses, session configuration and state.

```
#include <core.h>
```

Data Fields

- uint8_t guessed

  nbre of user guessed combination
- uint8_t state

  current state of session
- char ∗ account

  account name or NULL for default account
- mm_secret ∗ secret

  secret combination to guess
- mm_config ∗ config

  session config
- mm_guess ∗ panel

  session panel

9.11.1 Detailed Description

the session object containing the secret object, the inputed guesses, session configuration and state.

Definition at line 45 of file core.h.

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

- core.h

9.12 SDL_Table Struct Reference

table struct that contains table dimensions

Data Fields

- unsigned **x**
- unsigned **y**
- unsigned **w**
- unsigned **h**
- unsigned **rows**
- unsigned **cols**

9.12.1 Detailed Description

table struct that contains table dimensions

Definition at line 35 of file sdl.c.

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

- sdl.c

Chapter 10

File Documentation

10.1 colorscheme/4bit.h File Reference

4bit color scheme color for sdl version

Macros

- #define **bg_color**
- #define **fg_color**
- #define **br_color** fg_color
- #define **fg_black**
- #define **bg_black**
- #define **fg_red**
- #define **bg_red**
- #define **fg_green**
- #define **bg_green**
- #define **fg_yellow**
- #define **bg_yellow**
- #define **fg_blue**
- #define **bg_blue**
- #define **fg_magenta**
- #define **bg_magenta**
- #define **fg_cyan**
- #define **bg_cyan**
- #define **fg_white**
- #define **bg_white**

10.1.1 Detailed Description

4bit color scheme color for sdl version

Author

`http://ciembor.github.io/4bit/`

Definition in file 4bit.h.

10.1.2 Macro Definition Documentation

10.1.2.1 #define bg_black

Value:

```
{                                                                        \
        0x26, 0x26, 0x26, 0xFF                                           \
    }
```

Definition at line 20 of file 4bit.h.

10.1.2.2 #define bg_blue

Value:

```
{                                                                        \
        0xbd, 0xbd, 0xdb, 0xFF                                           \
    }
```

Definition at line 52 of file 4bit.h.

10.1.2.3 #define bg_color

Value:

```
{                                                                        \
        0x0d, 0x19, 0x26, 0xFF                                           \
    }
```

Definition at line 6 of file 4bit.h.

10.1.2.4 #define bg_cyan

Value:

```
{                                                                        \
        0xbd, 0xdb, 0xdb, 0xFF                                           \
    }
```

Definition at line 68 of file 4bit.h.

10.1.2.5 #define bg_green

Value:

```
{                                                                        \
        0xbd, 0xdb, 0xbd, 0xFF                                           \
    }
```

Definition at line 36 of file 4bit.h.

10.1.2.6 #define bg_magenta

Value:

```
{                                                                        \
        0xdb, 0xbd, 0xdb, 0xFF                                           \
    }
```

Definition at line 60 of file 4bit.h.

10.1.2.7 #define bg_red

Value:

```
{                                                                      \
        0xdb, 0xbd, 0xbd, 0xFF                                         \
    }
```

Definition at line 28 of file 4bit.h.

10.1.2.8 #define bg_white

Value:

```
{                                                                      \
        0xff, 0xff, 0xff, 0xFF                                         \
    }
```

Definition at line 76 of file 4bit.h.

10.1.2.9 #define bg_yellow

Value:

```
{                                                                      \
        0xdb, 0xdb, 0xbd, 0xFF                                         \
    }
```

Definition at line 44 of file 4bit.h.

10.1.2.10 #define fg_black

Value:

```
{                                                                      \
        0x00, 0x00, 0x00, 0xFF                                         \
    }
```

Definition at line 16 of file 4bit.h.

10.1.2.11 #define fg_blue

Value:

```
{                                                                      \
        0x7a, 0x7a, 0xb8, 0xFF                                         \
    }
```

Definition at line 48 of file 4bit.h.

10.1.2.12 #define fg_color

Value:

```
{                                                                      \
        0xd9, 0xe6, 0xf2, 0xFF                                         \
    }
```

Definition at line 10 of file 4bit.h.

10.1.2.13 #define fg_cyan

Value:

```
{                                                                      \
        0x7a, 0xb8, 0xb8, 0xFF                                         \
    }
```

Definition at line 64 of file 4bit.h.

10.1.2.14 #define fg_green

Value:

```
{                                                                      \
        0x7a, 0xb8, 0x7a, 0xFF                                         \
    }
```

Definition at line 32 of file 4bit.h.

10.1.2.15 #define fg_magenta

Value:

```
{                                                                      \
        0xb8, 0x7a, 0xb8, 0xFF                                         \
    }
```

Definition at line 56 of file 4bit.h.

10.1.2.16 #define fg_red

Value:

```
{                                                                      \
        0xb8, 0x7a, 0x7a, 0xFF                                         \
    }
```

Definition at line 24 of file 4bit.h.

10.1.2.17 #define fg_white

Value:

```
{                                                                      \
        0xd9, 0xd9, 0xd9, 0xFF                                         \
    }
```

Definition at line 72 of file 4bit.h.

10.1.2.18 #define fg_yellow

Value:

```
{                                                                      \
        0xb8, 0xb8, 0x7a, 0xFF                                         \
    }
```

Definition at line 40 of file 4bit.h.

10.2 colorscheme/jellybeans.h File Reference

jellybeans color scheme for sdl

Macros

- #define **fg_black**
- #define **bg_black**
- #define **fg_red**
- #define **bg_red**
- #define **fg_green**
- #define **bg_green**
- #define **fg_yellow**
- #define **bg_yellow**
- #define **fg_blue**
- #define **bg_blue**
- #define **fg_magenta**
- #define **bg_magenta**
- #define **fg_cyan**
- #define **bg_cyan**
- #define **fg_white**
- #define **bg_white**
- #define **bg_color**
- #define **fg_color**
- #define **br_color** fg_color

10.2.1 Detailed Description

jellybeans color scheme for sdl

Definition in file jellybeans.h.

10.2.2 Macro Definition Documentation

10.2.2.1 #define bg_black

Value:

```
{                                                                   \
        0x40, 0x40, 0x40, 0xFF                                      \
    }
```

Definition at line 9 of file jellybeans.h.

10.2.2.2 #define bg_blue

Value:

```
{                                                                   \
        0x81, 0x97, 0xBF, 0xFF                                      \
    }
```

Definition at line 41 of file jellybeans.h.

10.2.2.3 #define bg_color

Value:

```
{                                                                       \
        0x15, 0x15, 0x15, 0xFF                                          \
    }
```

Definition at line 69 of file jellybeans.h.

10.2.2.4 #define bg_cyan

Value:

```
{                                                                       \
        0x8F, 0xBF, 0xDC, 0xFF                                          \
    }
```

Definition at line 57 of file jellybeans.h.

10.2.2.5 #define bg_green

Value:

```
{                                                                       \
        0x99, 0xAD, 0x6A, 0xFF                                          \
    }
```

Definition at line 25 of file jellybeans.h.

10.2.2.6 #define bg_magenta

Value:

```
{                                                                       \
        0xC6, 0xB6, 0xEE, 0xFF                                          \
    }
```

Definition at line 49 of file jellybeans.h.

10.2.2.7 #define bg_red

Value:

```
{                                                                       \
        0xCF, 0x6A, 0x4C, 0xFF                                          \
    }
```

Definition at line 17 of file jellybeans.h.

10.2.2.8 #define bg_white

Value:

```
{                                                                       \
        0xE8, 0xE8, 0xD3, 0xFF                                          \
    }
```

Definition at line 65 of file jellybeans.h.

10.2.2.9 #define bg_yellow

Value:

```
{                                                              \
        0xFA, 0xD0, 0x7A, 0xFF                                 \
    }
```

Definition at line 33 of file jellybeans.h.

10.2.2.10 #define fg_black

Value:

```
{                                                              \
        0x1C, 0x1C, 0x1C, 0xFF                                 \
    }
```

Definition at line 5 of file jellybeans.h.

10.2.2.11 #define fg_blue

Value:

```
{                                                              \
        0x66, 0x78, 0x99, 0xFF                                 \
    }
```

Definition at line 37 of file jellybeans.h.

10.2.2.12 #define fg_color

Value:

```
{                                                              \
        0x88, 0x88, 0x88, 0xFF                                 \
    }
```

Definition at line 73 of file jellybeans.h.

10.2.2.13 #define fg_cyan

Value:

```
{                                                              \
        0x66, 0x87, 0x99, 0xFF                                 \
    }
```

Definition at line 53 of file jellybeans.h.

10.2.2.14 #define fg_green

Value:

```
{                                                              \
        0x79, 0x9D, 0x6A, 0xFF                                 \
    }
```

Definition at line 21 of file jellybeans.h.

10.2.2.15 #define fg_magenta

Value:

```
{                                                                  \
        0x87, 0x87, 0xAF, 0xFF                                     \
    }
```

Definition at line 45 of file jellybeans.h.

10.2.2.16 #define fg_red

Value:

```
{                                                                  \
        0xB8, 0x53, 0x35, 0xFF                                     \
    }
```

Definition at line 13 of file jellybeans.h.

10.2.2.17 #define fg_white

Value:

```
{                                                                  \
        0x88, 0x88, 0x88, 0xFF                                     \
    }
```

Definition at line 61 of file jellybeans.h.

10.2.2.18 #define fg_yellow

Value:

```
{                                                                  \
        0xFF, 0xB9, 0x64, 0xFF                                     \
    }
```

Definition at line 29 of file jellybeans.h.

10.3 colorscheme/rainbow_simple.h File Reference

simple_rainbow color scheme for sdl version

Macros

- #define **fg_black**
- #define **bg_black**
- #define **fg_red**
- #define **bg_red**
- #define **fg_green**
- #define **bg_green**
- #define **fg_yellow**
- #define **bg_yellow**
- #define **fg_blue**

- #define **bg_blue**
- #define **fg_magenta**
- #define **bg_mgenta**
- #define **fg_cyan**
- #define **bg_cyan**
- #define **fg_white**
- #define **bg_white**
- #define **bg_color**
- #define **fg_color**
- #define **br_color** fg_color

10.3.1 Detailed Description

simple_rainbow color scheme for sdl version

Definition in file rainbow_simple.h.

10.3.2 Macro Definition Documentation

10.3.2.1 #define bg_black

Value:

```
{                                                                          \
        0x88, 0x88, 0x88, 0xFF                                             \
    }
```

Definition at line 9 of file rainbow_simple.h.

10.3.2.2 #define bg_blue

Value:

```
{                                                                          \
        0x77, 0xbe, 0xe0, 0xFF                                             \
    }
```

Definition at line 41 of file rainbow_simple.h.

10.3.2.3 #define bg_color

Value:

```
{                                                                          \
        0x57, 0x57, 0x57, 0xFF                                             \
    }
```

Definition at line 70 of file rainbow_simple.h.

10.3.2.4 #define bg_cyan

Value:

```
{                                                                          \
        0xff, 0xc1, 0x78, 0xFF                                             \
    }
```

Definition at line 57 of file rainbow_simple.h.

10.3.2.5 #define bg_green

Value:

```
{                                                             \
        0xbd, 0xe0, 0x77, 0xFF                                \
    }
```

Definition at line 25 of file rainbow_simple.h.

10.3.2.6 #define bg_mgenta

Value:

```
{                                                             \
        0xdd, 0x91, 0xf3, 0xFF                                \
    }
```

Definition at line 49 of file rainbow_simple.h.

10.3.2.7 #define bg_red

Value:

```
{                                                             \
        0xff, 0x82, 0x78, 0xFF                                \
    }
```

Definition at line 17 of file rainbow_simple.h.

10.3.2.8 #define bg_white

Value:

```
{                                                             \
        0xcc, 0xcc, 0xcc, 0xFF                                \
    }
```

Definition at line 65 of file rainbow_simple.h.

10.3.2.9 #define bg_yellow

Value:

```
{                                                             \
        0xea, 0xdc, 0x84, 0xFF                                \
    }
```

Definition at line 33 of file rainbow_simple.h.

10.3.2.10 #define fg_black

Value:

```
{                                                             \
        0x66, 0x66, 0x66, 0xFF                                \
    }
```

Definition at line 5 of file rainbow_simple.h.

10.3.2.11 #define fg_blue

Value:

```
{                                                               \
        0x77, 0xbe, 0xe0, 0xFF                                  \
    }
```

Definition at line 37 of file rainbow_simple.h.

10.3.2.12 #define fg_color

Value:

```
{                                                               \
        0xdc, 0xdc, 0xcc, 0xFF                                  \
    }
```

Definition at line 74 of file rainbow_simple.h.

10.3.2.13 #define fg_cyan

Value:

```
{                                                               \
        0xff, 0xc1, 0x78, 0xFF                                  \
    }
```

Definition at line 53 of file rainbow_simple.h.

10.3.2.14 #define fg_green

Value:

```
{                                                               \
        0xbd, 0xe0, 0x77, 0xFF                                  \
    }
```

Definition at line 21 of file rainbow_simple.h.

10.3.2.15 #define fg_magenta

Value:

```
{                                                               \
        0xdd, 0x91, 0xf3, 0xFF                                  \
    }
```

Definition at line 45 of file rainbow_simple.h.

10.3.2.16 #define fg_red

Value:

```
{                                                               \
        0xff, 0x82, 0x78, 0xFF                                  \
    }
```

Definition at line 13 of file rainbow_simple.h.

10.3.2.17 #define fg_white

Value:

```
{                                                                          \
        0xdd, 0xdd, 0xdd, 0xFF                                             \
    }
```

Definition at line 61 of file rainbow_simple.h.

10.3.2.18 #define fg_yellow

Value:

```
{                                                                          \
        0xea, 0xdc, 0x84, 0xFF                                             \
    }
```

Definition at line 29 of file rainbow_simple.h.

10.4 colorscheme/rcg_term.h File Reference

rcg_term color scheme for sdl version

Macros

- #define **fg_color**
- #define **bg_color**
- #define **br_color** fg_color
- #define **fg_black**
- #define **fg_red**
- #define **fg_green**
- #define **fg_yellow**
- #define **fg_blue**
- #define **fg_magenta**
- #define **fg_cyan**
- #define **fg_white**
- #define **bg_black**
- #define **bg_red**
- #define **bg_green**
- #define **bg_yellow**
- #define **bg_blue**
- #define **bg_magenta**
- #define **bg_cyan**
- #define **bg_white**

10.4.1 Detailed Description

rcg_term color scheme for sdl version

Definition in file rcg_term.h.

10.4.2 Macro Definition Documentation

10.4.2.1 #define bg_black

Value:

```
{                                                                     \
        0x74, 0x74, 0x74, 0xFF                                        \
    }
```

Definition at line 48 of file rcg_term.h.

10.4.2.2 #define bg_blue

Value:

```
{                                                                     \
        0xb6, 0xde, 0xfb, 0xFF                                        \
    }
```

Definition at line 64 of file rcg_term.h.

10.4.2.3 #define bg_color

Value:

```
{                                                                     \
        0x4B, 0x5D, 0x7E, 0xFF                                        \
    }
```

Definition at line 9 of file rcg_term.h.

10.4.2.4 #define bg_cyan

Value:

```
{                                                                     \
        0xd7, 0xd9, 0xfc, 0xFF                                        \
    }
```

Definition at line 72 of file rcg_term.h.

10.4.2.5 #define bg_green

Value:

```
{                                                                     \
        0xc3, 0xf7, 0x86, 0xFF                                        \
    }
```

Definition at line 56 of file rcg_term.h.

10.4.2.6 #define bg_magenta

Value:

```
{                                                                     \
        0xfb, 0xa1, 0xfb, 0xFF                                        \
    }
```

Definition at line 68 of file rcg_term.h.

10.4.2.7 #define bg_red

Value:

```
{                                                                      \
        0xf9, 0x92, 0x86, 0xFF                                         \
    }
```

Definition at line 52 of file rcg_term.h.

10.4.2.8 #define bg_white

Value:

```
{                                                                      \
        0xe2, 0xe2, 0xe2, 0xFF                                         \
    }
```

Definition at line 76 of file rcg_term.h.

10.4.2.9 #define bg_yellow

Value:

```
{                                                                      \
        0xfc, 0xfb, 0xcc, 0xFF                                         \
    }
```

Definition at line 60 of file rcg_term.h.

10.4.2.10 #define fg_black

Value:

```
{                                                                      \
        0x6c, 0x6c, 0x6c, 0xFF                                         \
    }
```

Definition at line 15 of file rcg_term.h.

10.4.2.11 #define fg_blue

Value:

```
{                                                                      \
        0xa9, 0xcd, 0xeb, 0xFF                                         \
    }
```

Definition at line 31 of file rcg_term.h.

10.4.2.12 #define fg_color

Value:

```
{                                                                      \
        0xCC, 0xCC, 0xCC, 0xFF                                         \
    }
```

Definition at line 5 of file rcg_term.h.

10.4.2.13 #define fg_cyan

Value:

```
{                                                                   \
        0xc9, 0xca, 0xec, 0xFF                                      \
    }
```

Definition at line 39 of file rcg_term.h.

10.4.2.14 #define fg_green

Value:

```
{                                                                   \
        0xb6, 0xe7, 0x7d, 0xFF                                      \
    }
```

Definition at line 23 of file rcg_term.h.

10.4.2.15 #define fg_magenta

Value:

```
{                                                                   \
        0xea, 0x96, 0xeb, 0xFF                                      \
    }
```

Definition at line 35 of file rcg_term.h.

10.4.2.16 #define fg_red

Value:

```
{                                                                   \
        0xe9, 0x89, 0x7c, 0xFF                                      \
    }
```

Definition at line 19 of file rcg_term.h.

10.4.2.17 #define fg_white

Value:

```
{                                                                   \
        0xCC, 0xCC, 0xCC, 0xFF                                      \
    }
```

Definition at line 43 of file rcg_term.h.

10.4.2.18 #define fg_yellow

Value:

```
{                                                                   \
        0xec, 0xeb, 0xbe, 0xFF                                      \
    }
```

Definition at line 27 of file rcg_term.h.

10.5 colorscheme/solarized_dark.h File Reference

solarized dark color scheme for sdl version

Macros

- #define **fg_black**
- #define **fg_red**
- #define **fg_green**
- #define **fg_yellow**
- #define **fg_blue**
- #define **fg_magenta**
- #define **fg_cyan**
- #define **fg_white**
- #define **bg_black**
- #define **bg_red**
- #define **bg_green**
- #define **bg_yellow**
- #define **bg_blue**
- #define **bg_magenta**
- #define **bg_cyan**
- #define **bg_white**
- #define **bg_color** bg_black
- #define **fg_color** fg_white
- #define **br_color** fg_white

10.5.1 Detailed Description

solarized dark color scheme for sdl version

Definition in file solarized_dark.h.

10.5.2 Macro Definition Documentation

10.5.2.1 #define bg_black

Value:

```
{                                                                    \
        0x00, 0x2b, 0x36, 0xff                                       \
    }
```

Definition at line 37 of file solarized_dark.h.

10.5.2.2 #define bg_blue

Value:

```
{                                                                    \
        0x83, 0x94, 0x96, 0xff                                       \
    }
```

Definition at line 53 of file solarized_dark.h.

10.5.2.3 #define bg_cyan

Value:

```
{                                                                      \
        0x93, 0xa1, 0xa1, 0xff                                         \
    }
```

Definition at line 61 of file solarized_dark.h.

10.5.2.4 #define bg_green

Value:

```
{                                                                      \
        0x58, 0x6e, 0x75, 0xff                                         \
    }
```

Definition at line 45 of file solarized_dark.h.

10.5.2.5 #define bg_magenta

Value:

```
{                                                                      \
        0x6c, 0x71, 0xc4, 0xff                                         \
    }
```

Definition at line 57 of file solarized_dark.h.

10.5.2.6 #define bg_red

Value:

```
{                                                                      \
        0xcb, 0x4b, 0x16, 0xff                                         \
    }
```

Definition at line 41 of file solarized_dark.h.

10.5.2.7 #define bg_white

Value:

```
{                                                                      \
        0xfd, 0xf6, 0xe3, 0xff                                         \
    }
```

Definition at line 65 of file solarized_dark.h.

10.5.2.8 #define bg_yellow

Value:

```
{                                                                      \
        0x65, 0x7b, 0x83, 0xff                                         \
    }
```

Definition at line 49 of file solarized_dark.h.

10.5.2.9 #define fg_black

Value:

```
{                                                                    \
        0x07, 0x36, 0x42, 0xff                                       \
    }
```

Definition at line 5 of file solarized_dark.h.

10.5.2.10 #define fg_blue

Value:

```
{                                                                    \
        0x26, 0x8b, 0xd2, 0xff                                       \
    }
```

Definition at line 21 of file solarized_dark.h.

10.5.2.11 #define fg_cyan

Value:

```
{                                                                    \
        0x2a, 0xa1, 0x98, 0xff                                       \
    }
```

Definition at line 29 of file solarized_dark.h.

10.5.2.12 #define fg_green

Value:

```
{                                                                    \
        0x85, 0x99, 0x00, 0xff                                       \
    }
```

Definition at line 13 of file solarized_dark.h.

10.5.2.13 #define fg_magenta

Value:

```
{                                                                    \
        0xd3, 0x36, 0x82, 0xff                                       \
    }
```

Definition at line 25 of file solarized_dark.h.

10.5.2.14 #define fg_red

Value:

```
{                                                                    \
        0xdc, 0x32, 0x2f, 0xff                                       \
    }
```

Definition at line 9 of file solarized_dark.h.

10.5.2.15 #define fg_white

Value:

```
{                                                                              \
        0xee, 0xe8, 0xd5, 0xff                                                 \
    }
```

Definition at line 33 of file solarized_dark.h.

10.5.2.16 #define fg_yellow

Value:

```
{                                                                              \
        0xb5, 0x89, 0x00, 0xff                                                 \
    }
```

Definition at line 17 of file solarized_dark.h.

10.6 colorscheme/solarized_light.h File Reference

solarized light color scheme for sdl version

Macros

- #define **fg_black**
- #define **fg_red**
- #define **fg_green**
- #define **fg_yellow**
- #define **fg_blue**
- #define **fg_magenta**
- #define **fg_cyan**
- #define **fg_white**
- #define **bg_black**
- #define **bg_red**
- #define **bg_green**
- #define **bg_yellow**
- #define **bg_blue**
- #define **bg_magenta**
- #define **bg_cyan**
- #define **bg_white**
- #define **bg_color** bg_white
- #define **fg_color** fg_black
- #define **br_color** fg_black

10.6.1 Detailed Description

solarized light color scheme for sdl version

Definition in file solarized_light.h.

10.6.2 Macro Definition Documentation

10.6.2.1 #define bg_black

Value:

```
{                                                                      \
        0x00, 0x2b, 0x36, 0xff                                         \
    }
```

Definition at line 37 of file solarized_light.h.

10.6.2.2 #define bg_blue

Value:

```
{                                                                      \
        0x83, 0x94, 0x96, 0xff                                         \
    }
```

Definition at line 53 of file solarized_light.h.

10.6.2.3 #define bg_cyan

Value:

```
{                                                                      \
        0x93, 0xa1, 0xa1, 0xff                                         \
    }
```

Definition at line 61 of file solarized_light.h.

10.6.2.4 #define bg_green

Value:

```
{                                                                      \
        0x58, 0x6e, 0x75, 0xff                                         \
    }
```

Definition at line 45 of file solarized_light.h.

10.6.2.5 #define bg_magenta

Value:

```
{                                                                      \
        0x6c, 0x71, 0xc4, 0xff                                         \
    }
```

Definition at line 57 of file solarized_light.h.

10.6.2.6 #define bg_red

Value:

```
{                                                                      \
        0xcb, 0x4b, 0x16, 0xff                                         \
    }
```

Definition at line 41 of file solarized_light.h.

10.6.2.7 #define bg_white

Value:

```
{                                                                  \
        0xfd, 0xf6, 0xe3, 0xff                                     \
    }
```

Definition at line 65 of file solarized_light.h.

10.6.2.8 #define bg_yellow

Value:

```
{                                                                  \
        0x65, 0x7b, 0x83, 0xff                                     \
    }
```

Definition at line 49 of file solarized_light.h.

10.6.2.9 #define fg_black

Value:

```
{                                                                  \
        0x07, 0x36, 0x42, 0xff                                     \
    }
```

Definition at line 5 of file solarized_light.h.

10.6.2.10 #define fg_blue

Value:

```
{                                                                  \
        0x26, 0x8b, 0xd2, 0xff                                     \
    }
```

Definition at line 21 of file solarized_light.h.

10.6.2.11 #define fg_cyan

Value:

```
{                                                                  \
        0x2a, 0xa1, 0x98, 0xff                                     \
    }
```

Definition at line 29 of file solarized_light.h.

10.6.2.12 #define fg_green

Value:

```
{                                                                  \
        0x85, 0x99, 0x00, 0xff                                     \
    }
```

Definition at line 13 of file solarized_light.h.

10.6.2.13 #define fg_magenta

Value:

```
{                                                                    \
        0xd3, 0x36, 0x82, 0xff                                       \
    }
```

Definition at line 25 of file solarized_light.h.

10.6.2.14 #define fg_red

Value:

```
{                                                                    \
        0xdc, 0x32, 0x2f, 0xff                                       \
    }
```

Definition at line 9 of file solarized_light.h.

10.6.2.15 #define fg_white

Value:

```
{                                                                    \
        0xee, 0xe8, 0xd5, 0xff                                       \
    }
```

Definition at line 33 of file solarized_light.h.

10.6.2.16 #define fg_yellow

Value:

```
{                                                                    \
        0xb5, 0x89, 0x00, 0xff                                       \
    }
```

Definition at line 17 of file solarized_light.h.

10.7 config.h File Reference

mastermind core build time configuration options

```
#include "colorscheme/solarized_light.h"
```

Macros

- #define PACKAGE "mastermind"

  package name
- #define PROGRAM_NAME "MasterMind"

  user firendly program name
- #define PROGRAM_VERSION "0.1.0"

  verion

- #define PROGRAM_URL "https://github.com/lejenome/mastermind"

  website
- #define MM_HOLES 4

  default holes number
- #define MM_COLORS 6

  default colors number
- #define MM_GUESSES 10

  default guesses number
- #define MM_HOLES_MAX 8

  max number of holes mastermind can accepts
- #define MM_COLORS_MAX 12

  max number of colors mastermind can accepts
- #define MM_GUESSES_MAX 20

  max number of guesses mastermind can accepts
- #define **FONTSDIR** "./fonts/"
- #define **ICONSDIR** "./icons/"
- #define **LOCALEDIR** "locale/"

10.7.1 Detailed Description

mastermind core build time configuration options

Definition in file config.h.

10.8 cli-cmd.c File Reference

commands unctions handlers

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "lib.h"
#include "core.h"
#include "cli-cmd.h"
#include <getopt.h>
```

Functions

- int execArgs (int argc, char ∗argv[ ], cmd_t ∗cmds, size_t len, mm_session ∗session)
- int **cmd_quit** (const char argc, const char ∗∗argv, mm_session ∗session)
- int **cmd_savegame** (const char argc, const char ∗∗argv, mm_session ∗session)
- int **cmd_set** (const char argc, const char ∗∗argv, mm_session ∗session)
- int **cmd_restart** (const char argc, const char ∗∗argv, mm_session ∗session)
- int **cmd_help** (const char argc, const char ∗∗argv, mm_session ∗session)
- int **cmd_score** (const char argc, const char ∗∗argv, mm_session ∗session)
- int **cmd_account** (const char argc, const char ∗∗argv, mm_session ∗session)
- int **cmd_version** (const char argc, const char ∗∗argv, mm_session ∗session)

Variables

- uint8_t **mm_cmd_mode** = MM_CMD_MODE_OPT

10.8.1 Detailed Description

commands unctions handlers

Definition in file cli-cmd.c.

10.8.2 Function Documentation

10.8.2.1 int execArgs (int *argc*, char ∗ *argv[ ]*, cmd_t ∗ *cmds*, size_t *len*, mm_session ∗ *session*)

parse and exec command line arguments

Parameters

argc	main function argc param
argv	main function argv param
cmds	list of commands objects
len	length of cmds array

Returns

excuted commands return values

Definition at line 24 of file cli-cmd.c.

10.9 cli-cmd.h File Reference

commands functions handlers

```
#include "core.h"
```

Data Structures

- struct cmd_t

  command object containing its name, its function and args

Macros

- #define MM_CMD_SUCCESS 0x00

  cmds return values
- #define MM_CMD_ERROR 0x01

  their was an error
- #define MM_CMD_NEW_SESSION 0x02

  new session needed
- #define MM_CMD_REDESIGN 0x04

  redesign the panel is needed
- #define MM_CMD_OPT_EXIT 0x08

  just exit the program if cmd executed from option mode cmds are executed on (to modify the output style)
- #define MM_CMD_MODE_OPT 0

  command executed on option mode
- #define MM_CMD_MODE_CLI 1

  command executed on interactive cli mode
- #define MM_CMD_MODE_GUI 2

  command executed on GUI interface mode

Functions

- int **cmd_quit** (const char, const char ∗∗, mm_session ∗)
- int **cmd_savegame** (const char, const char ∗∗, mm_session ∗)
- int **cmd_set** (const char, const char ∗∗, mm_session ∗)
- int **cmd_restart** (const char, const char ∗∗, mm_session ∗)
- int **cmd_score** (const char, const char ∗∗, mm_session ∗)
- int **cmd_help** (const char, const char ∗∗, mm_session ∗)
- int **cmd_account** (const char, const char ∗∗, mm_session ∗)
- int **cmd_version** (const char, const char ∗∗, mm_session ∗)
- int execArgs (int argc, char ∗argv[ ], cmd_t ∗cmds, size_t len, mm_session ∗)

Variables

- uint8_t **mm_cmd_mode**

10.9.1 Detailed Description

commands functions handlers

Definition in file cli-cmd.h.

10.9.2 Macro Definition Documentation

10.9.2.1 #define MM_CMD_SUCCESS 0x00

cmds return values

cmd succeded

Definition at line 11 of file cli-cmd.h.

10.9.3 Function Documentation

10.9.3.1 int execArgs (int *argc,* char ∗ *argv[ ],* cmd_t ∗ *cmds,* size_t *len,* mm_session ∗ *session*)

parse and exec command line arguments

Parameters

argc	main function argc param
argv	main function argv param
cmds	list of commands objects
len	length of cmds array

Returns

excuted commands return values

Definition at line 24 of file cli-cmd.c.

10.10 cli.c File Reference

command line interface implemetation of mastermind

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdint.h>
#include <readline/readline.h>
#include <readline/history.h>
#include "lib.h"
#include "cli-cmd.h"
#include "core.h"
```

Functions

- void printPanel ()

  draw session panel
- char ∗∗ parseBuf (char ∗buf, unsigned ∗argc)
- int getCombination (uint8_t ∗T)
- int **main** (int argc, char ∗argv[ ])

Variables

- mm_session ∗ **session**
- cmd_t **cmds** [ ]

10.10.1 Detailed Description

command line interface implemetation of mastermind

Definition in file cli.c.

10.10.2 Function Documentation

10.10.2.1 int getCombination (uint8_t ∗ *T*)

get guessed combination and handle input buffer commands

Returns

-1 : input error, redo (do not redraw table) 0 : seccess input, redo if mm_play(T) does not success (do not redraw table) or next (redraw table) 1 : cmd input, redo (do not redo table) 2 : cmd input, next (redraw table)

Definition at line 236 of file cli.c.

10.10.2.2 char∗∗ parseBuf (char ∗ *buf,* unsigned ∗ *argc*)

parse buffer and get arguments from it

Parameters

buf	buffer to parse
argc	poiter to where to store arguments count

Returns

arguments array or NULL if buf is invalid

Definition at line 68 of file cli.c.

10.10.3 Variable Documentation

10.10.3.1 cmd_t cmds[ ]

Initial value:

```
= {
    {.n = "quit", .e = cmd_quit, .s = 0},
    {.n = "set", .e = cmd_set, .s = 's', .a = 2},
    {.n = "restart", .e = cmd_restart, .s = 0},
    {.n = "savegame", .e = cmd_savegame, .s = 0},
    {.n = "score", .e = cmd_score, .s = 'c', .a = 0},
    {.n = "help", .e = cmd_help, .s = 'h', .a = 1},
    {.n = "account", .e = cmd_account, .s = 'a', .a = 1},
    {.n = "version", .e = cmd_version, .s = 'v', .a = 0},
}
```

Definition at line 21 of file cli.c.

10.11 core.c File Reference

mastermind core funtions and types definition

```
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include <string.h>
#include <assert.h>
#include <stdint.h>
#include <errno.h>
#include "lib.h"
#include "core.h"
#include <windows.h>
```

Macros

- #define **MM_POS_GUESSES** 0
- #define **MM_POS_COLORS** 1
- #define **MM_POS_HOLES** 2
- #define **MM_POS_REMISE** 3
- #define **MM_POS_ACCOUNT** 4
- #define **MM_POS_SAVE_EXIT** 5
- #define **MM_POS_SAVE_PLAY** 6

Functions

- mm_session * mm_session_new ()
- void mm_config_load ()
- mm_config * mm_config_new ()
- void mm_config_save ()
- unsigned mm_config_set (const char *name, const char *value)
- mm_secret * mm_secret_new (mm_config *conf)
- void mm_scores_load ()
- const mm_scores_t * mm_scores_get ()
- void mm_scores_save (mm_session *session)
- unsigned mm_play (mm_session *session, uint8_t *T)

- long unsigned mm_score (mm_session ∗session)
- mm_guess mm_play_last (mm_session ∗session)
- void mm_init (const char ∗data_dir)
- void mm_session_free (mm_session ∗session)
- void mm_session_exit (mm_session ∗session)
- unsigned int mm_session_save (mm_session ∗session)
- mm_session ∗ mm_session_restore ()

Variables

- mm_scores_t **mm_scores** = {.T = NULL, .max = 20, .len = 0}
- char ∗ **mm_config_path** = NULL
- char ∗ **mm_score_path** = NULL
- char ∗ **mm_store_path** = NULL
- mm_conf_t **mm_confs** [7]

10.11.1 Detailed Description

mastermind core funtions and types definition

Definition in file core.c.

10.11.2 Function Documentation

10.11.2.1 void mm_config_load ( )

load global config from config file and create new session config based on global config

Returns

new session config object

Definition at line 83 of file core.c.

10.11.2.2 mm_config∗ mm_config_new ( )

create new session config

Returns

new session config

Definition at line 141 of file core.c.

10.11.2.3 void mm_config_save ( )

save global config on the config file

Definition at line 153 of file core.c.

10.11.2.4 unsigned mm_config_set (const char ∗ *name,* const char ∗ *value*)

change global config with name to value then save to config file

Parameters

name	name of global config to change
value	the new value of global config name

Returns

> 0 on success , 1 if conf option not found, 2 if conf value is not valid

Definition at line 182 of file core.c.

10.11.2.5 void mm_init (const char ∗ *data_dir*)

This function initialize data && config && store files path using system and core default standard or passed dir path

Note

> you do not need to call this function only if you want to use custom dir

Parameters

data_dir	path to dir that will contain the files or NULL to use system default/standard paths

Definition at line 393 of file core.c.

10.11.2.6 unsigned mm_play (mm_session ∗ *session,* uint8_t ∗ *T*)

This function is the most important function in the code this function accept new guess combination , add it to the session if it's not ended and calculed the score of the current guess then update session status

Parameters

session	current session
t	the new guess combination

Returns

> 0 on success 1 on failure (session already ended, combination is not valid)

Definition at line 314 of file core.c.

10.11.2.7 mm_guess mm_play_last (mm_session ∗ *session*)

get last gusess objet

Parameters

session	current session

Returns

> last guess object

Definition at line 381 of file core.c.

10.11.2.8 long unsigned mm_score (mm_session ∗ *session*)

geenrate session score

Parameters

session	session which to generate score

Returns

session score

Definition at line 358 of file core.c.

10.11.2.9 const mm_scores_t* mm_scores_get ( )

return pointer to scores object

Returns

pointer to score object

Definition at line 257 of file core.c.

10.11.2.10 void mm_scores_load ( )

load scores from file

Definition at line 238 of file core.c.

10.11.2.11 void mm_scores_save (mm_session * *session*)

generate score of session and save it to score object/file if it's on top 20

Parameters

session	session which to save score

Definition at line 266 of file core.c.

10.11.2.12 mm_secret* mm_secret_new (mm_config * *conf*)

create the secret part of mastermind using session config this fuction use random and save it on mm_secret->val && save freq of every color on mm_secret->freq

Parameters

conf	config of current session

Returns

secret objet for this session

Definition at line 221 of file core.c.

10.11.2.13 void mm_session_exit (mm_session * *session*)

save session if not ended && save_on_exit = 1 then free object

Parameters

session	session to check and free before exit

Note

if your are not exiting the program use mm_session_free instead as mm_session_exit may store the session

Definition at line 486 of file core.c.

10.11.2.14 void mm_session_free (mm_session ∗ *session*)

free session object

Parameters

session	session to free

Definition at line 470 of file core.c.

10.11.2.15 mm_session∗ mm_session_new ( )

create new mastermind session and initialize viables && config

Returns

new session object

Definition at line 67 of file core.c.

10.11.2.16 mm_session∗ mm_session_restore ( )

this function restore session object from mm_store_path file

Returns

NULL on failure , session object pointeur on success

Definition at line 538 of file core.c.

10.11.2.17 unsigned int mm_session_save (mm_session ∗ *session*)

save session object on mm_store_path file

Parameters

session	current session object

Returns

0 on success , 1 on failure

Definition at line 497 of file core.c.

10.11.3 Variable Documentation

10.11.3.1 mm_conf_t mm_confs[7]

Initial value:

```
= {
    [MM_POS_GUESSES] = {.nbre = {.name = "guesses",
                        .type = MM_CONF_INT,
                        .val = MM_GUESSES,
                        .min = 2,
                        .max = MM_GUESSES_MAX}},
    [MM_POS_COLORS] = {.nbre = {.type = MM_CONF_INT,
                       .name = "colors",
                       .val = MM_COLORS,
                       .min = 2,
                       .max = MM_COLORS_MAX}},
    [MM_POS_HOLES] = {.nbre = {.type = MM_CONF_INT,
                      .name = "holes",
                      .val = MM_HOLES,
                      .min = 2,
                      .max = MM_HOLES_MAX}},
    [MM_POS_REMISE] = {.bool = {.type = MM_CONF_BOOL,
                       .name = "remise",
                       .val = 0}},
    [MM_POS_ACCOUNT] = {.str = {.type = MM_CONF_STR,
                       .name = "account",
                       .val = "default"}},
    [MM_POS_SAVE_EXIT] = {.bool = {.type = MM_CONF_BOOL,
                          .name = "save_on_exit",
                          .val = 0}},
    [MM_POS_SAVE_PLAY] = {.bool = {.type = MM_CONF_BOOL,
                          .name = "save_on_play",
                          .val = 0}},
}
```

Definition at line 35 of file core.c.

10.12 core.h File Reference

mastermind core funtions and types definition

```
#include <stdint.h>
#include "../config.h"
```

Data Structures

- struct mm_config

  contains current session configuration
- struct mm_guess

  the guess object, containing the combination by the user and its score
- struct mm_secret

  contains the secret generated at the beginning of session and the freq of its colors.
- struct mm_session

  the session object containing the secret object, the inputed guesses, session configuration and state.
- struct mm_score_t

  session score, containing score value and account name
- struct mm_scores_t

  top scores history
- struct mm_conf_int_t

  configuration option of type integer (conf.nbre.∗)
- struct mm_conf_str_t

configuration option of type string (conf.str.∗)

- struct mm_conf_bool_t

  configuration option od type boolean (conf.bool.∗)
- union mm_conf_t

  general configuration type

Macros

- #define **MM_NEW** 0
- #define **MM_PLAYING** 1
- #define **MM_SUCCESS** 2
- #define **MM_FAIL** 4
- #define **MM_CONF_INT** 0
- #define **MM_CONF_STR** 1
- #define **MM_CONF_BOOL** 2

Functions

- mm_session ∗ mm_session_new ()
- mm_session ∗ mm_session_restore ()
- unsigned int mm_session_save (mm_session ∗)
- void mm_session_free (mm_session ∗)
- void mm_session_exit (mm_session ∗)
- mm_config ∗ mm_config_new ()
- void mm_config_load ()
- void mm_config_save ()
- unsigned mm_config_set (const char ∗, const char ∗)
- const mm_scores_t ∗ mm_scores_get ()
- long unsigned mm_score (mm_session ∗session)
- mm_secret ∗ mm_secret_new (mm_config ∗)
- unsigned mm_play (mm_session ∗, uint8_t ∗)
- void mm_init (const char ∗)

Variables

- mm_conf_t **mm_confs** [7]

10.12.1 Detailed Description

mastermind core funtions and types definition

Definition in file core.h.

10.12.2 Function Documentation

10.12.2.1 void mm_config_load ( )

load global config from config file and create new session config based on global config

Returns

new session config object

Definition at line 83 of file core.c.

10.12.2.2 mm_config∗ mm_config_new ( )

create new session config

Returns

new session config

Definition at line 141 of file core.c.

10.12.2.3 void mm_config_save ( )

save global config on the config file

Definition at line 153 of file core.c.

10.12.2.4 unsigned mm_config_set (const char ∗ *name,* const char ∗ *value*)

change global config with name to value then save to config file

Parameters

name	name of global config to change
value	the new value of global config name

Returns

0 on success , 1 if conf option not found, 2 if conf value is not valid

Definition at line 182 of file core.c.

10.12.2.5 void mm_init (const char ∗ *data_dir*)

This function initialize data && config && store files path using system and core default standard or passed dir path

Note

you do not need to call this function only if you want to use custom dir

Parameters

data_dir	path to dir that will contain the files or NULL to use system default/standard paths

Definition at line 393 of file core.c.

10.12.2.6 unsigned mm_play (mm_session ∗ *session,* uint8_t ∗ *T*)

This function is the most important function in the code this function accept new guess combination , add it to the session if it's not ended and calculed the score of the current guess then update session status

Parameters

session	current session
t	the new guess combination

Returns

0 on success 1 on failure (session already ended, combination is not valid)

Definition at line 314 of file core.c.

10.12.2.7 long unsigned mm_score (mm_session ∗ *session*)

geenrate session score

Parameters

session	session which to generate score

Returns

session score

Definition at line 358 of file core.c.

10.12.2.8 const mm_scores_t∗ mm_scores_get ( )

return pointer to scores object

Returns

pointer to score object

Definition at line 257 of file core.c.

10.12.2.9 mm_secret∗ mm_secret_new (mm_config ∗ *conf*)

create the secret part of mastermind using session config this fuction use random and save it on mm_secret->val && save freq of every color on mm_secret->freq

Parameters

conf	config of current session

Returns

secret objet for this session

Definition at line 221 of file core.c.

10.12.2.10 void mm_session_exit (mm_session ∗ *session*)

save session if not ended && save_on_exit = 1 then free object

Parameters

session	session to check and free before exit

Note

if your are not exiting the program use mm_session_free instead as mm_session_exit may store the session

Definition at line 486 of file core.c.

10.12.2.11 void mm_session_free (mm_session ∗ *session*)

free session object

Parameters

session	session to free

Definition at line 470 of file core.c.

10.12.2.12 mm_session∗ mm_session_new ( )

create new mastermind session and initialize viables && config

Returns

new session object

Definition at line 67 of file core.c.

10.12.2.13 mm_session∗ mm_session_restore ( )

this function restore session object from mm_store_path file

Returns

NULL on failure , session object pointeur on success

Definition at line 538 of file core.c.

10.12.2.14 unsigned int mm_session_save (mm_session ∗ *session*)

save session object on mm_store_path file

Parameters

session	current session object

Returns

0 on success , 1 on failure

Definition at line 497 of file core.c.

10.13 lib.c File Reference

missing or core/iterface independent functions definition

```
#include <stdlib.h>
#include <string.h>
#include "lib.h"
```

Functions

- char ∗ **strndup** (const char ∗buf, size_t len)

10.13.1 Detailed Description

missing or core/iterface independent functions definition

Definition in file lib.c.

10.14 lib.h File Reference

core/interface independent functions/types definition and fixing systems incompatibility problems

```
#include "../config.h"
#include <locale.h>
#include <libintl.h>
```

Macros

- #define **LEN**(a) (sizeof(a) / sizeof(a[0]))
- #define **srandom**(var) srand(var)
- #define **random**() rand()
- #define **snprintf**(s, n, fmt,...) sprintf_s(s, n, fmt, __VA_ARGS__)
- #define _(str) gettext(str)

Functions

- char ∗ **strndup** (const char ∗buf, size_t len)

10.14.1 Detailed Description

core/interface independent functions/types definition and fixing systems incompatibility problems

Definition in file lib.h.

10.15 sdl.c File Reference

SDL interface implementation of mastermind.

```
#include <stdio.h>
#include <stdlib.h>
#include <SDL.h>
#include <SDL_ttf.h>
#include <string.h>
#include "lib.h"
#include "core.h"
```

Data Structures

- struct SDL_Table

    table struct that contains table dimensions

Macros

- #define **drawSecret**() drawCombination(session->secret->val, session->config->guesses, 0)
- #define **drawGuess**(p) drawCombination(session->panel[p].combination, p, 1)
- #define **sdl_print_center**(s, x, y, color) sdl_print(s, x, y, color, 0)
- #define **sdl_print_left**(s, x, y, color) sdl_print(s, x, y, color, -1)
- #define **TAB_GAME** (uint8_t)0
- #define **TAB_SETTINGS** (uint8_t)1

Functions

- void init_sdl ()
- void clean ()
- unsigned sdl_print (char ∗s, int x, int y, SDL_Color ∗color, int align)
- unsigned sdl_print_icon (uint16_t c, int x, int y, SDL_Color ∗color)
- void setBg ()
- void initTables ()
- void initColors ()
- int drawTableBottom (SDL_Table ∗T)
- void drawTableTop (SDL_Table ∗T)
- void drawCombination (uint8_t ∗G, unsigned p, unsigned drawState)
- void drawSelector ()
- void redraw_settings ()
- void redraw_game ()
- void redraw ()
- int onMouseUp (SDL_MouseButtonEvent e)
- int getGuess ()
- void iter ()
- int **main** (int argc, char ∗argv[ ])

Variables

- int **SCREEN_HEIGHT** = 640
- int **SCREEN_WIDTH** = 480
- int **SCREEN_HEIGHT_MIN** = 420
- int **SCREEN_WIDTH_MIN** = 360
- SDL_Window ∗ **win** = NULL
- SDL_Renderer ∗ **rend** = NULL
- TTF_Font ∗ **font** = NULL
- TTF_Font ∗ **icons** = NULL
- mm_session ∗ **session** = NULL
- uint8_t ∗ **curGuess** = NULL
- unsigned **curPos** = 0
- SDL_Table **panel**
- SDL_Table **state**
- SDL_Table **control**
- SDL_Table **play**
- unsigned **case_w**
- unsigned **case_h**
- unsigned **button_w**
- SDL_Color ∗ **colors** = NULL
- uint8_t **curTab** = TAB_GAME

10.15.1 Detailed Description

SDL interface implementation of mastermind.

Definition in file sdl.c.

10.15.2 Function Documentation

10.15.2.1 void clean ( )

close sdl subsystems and free memory

Definition at line 96 of file sdl.c.

10.15.2.2 void drawCombination (uint8_t ∗ *G*, unsigned *p*, unsigned *drawState*)

draw given combination on panel table and its score on state table

Parameters

G	combination array
p	position on panel
drawState	draw combination score from current session

Definition at line 320 of file sdl.c.

10.15.2.3 void drawSelector ( )

draw selector icons on panel on current guess position of current session

Definition at line 360 of file sdl.c.

10.15.2.4 int drawTableBottom (SDL_Table ∗ *T*)

draw borders of bottom tables

Definition at line 285 of file sdl.c.

10.15.2.5 void drawTableTop (SDL_Table ∗ *T*)

draw borders of top tables with double case for selector if session still not ended

Definition at line 301 of file sdl.c.

10.15.2.6 int getGuess ( )

handle all available events on events pipe

Returns

-1 to restart the session 0 to play current guess 1 if nothing to do

Definition at line 553 of file sdl.c.

10.15.2.7 void init_sdl ( )

Init SDL subsystem, create window and load fonts

Definition at line 57 of file sdl.c.

10.15.2.8 void initColors ( )

create colors array with current session colors as length

Definition at line 269 of file sdl.c.

10.15.2.9 void initTables ( )

recalcualte tables elements dimensions/values using current session settings

Definition at line 238 of file sdl.c.

10.15.2.10 void iter ( )

one iteration on main loop, handle available events and exec requested action (play guess, restart session) and redraw if needed

Definition at line 616 of file sdl.c.

10.15.2.11 int onMouseUp (SDL_MouseButtonEvent *e*)

mouse button up event handler

Returns

-1 to reset session 0 to play current guess 1 to keep listing to events

Definition at line 468 of file sdl.c.

10.15.2.12 void redraw ( )

clean renderer and redraw current tab

Definition at line 449 of file sdl.c.

10.15.2.13 void redraw_game ( )

draw game tab

Definition at line 421 of file sdl.c.

10.15.2.14 void redraw_settings ( )

draw settings tab

Definition at line 376 of file sdl.c.

10.15.2.15 unsigned sdl_print (char ∗ *s,* int *x,* int *y,* SDL_Color ∗ *color,* int *align*)

printf text with deined position and color

Parameters

s	text to print
x	x coord
y	y coord
color	pointer to fourground color to use or NULL for default
align	text align to provided position: -1: left align 0: center align 1: right align

Returns

printed text width

Definition at line 121 of file sdl.c.

10.15.2.16 unsigned sdl_print_icon (uint16_t *c*, int *x*, int *y*, SDL_Color * *color*)

print icon

Parameters

c	unicode o icon to print on icons font
x	x coord
y	y coord
color	poiter to fourground color or NULL to use default color

Returns

printed icon width

Definition at line 166 of file sdl.c.

10.15.2.17 void setBg ( )

draw background color

Definition at line 228 of file sdl.c.