MiniLasca

Generated by Doxygen 1.9.0

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Data Structure Documentation

3.1 campo Struct Reference

`#include <supporto.h>`

Data Fields

- char ** mat
- unsigned int r
- unsigned int c

3.1.1 Field Documentation

3.1.1.1 c

`unsigned int c`

3.1.1.2 mat

`char** mat`

3.1.1.3 r

`unsigned int r`

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

- supporto.h

3.2 pedina Struct Reference

`#include <supporto.h>`

Data Fields

- char * et
- unsigned int dim
- unsigned int cima
- unsigned int grado
- unsigned int r
- unsigned int c
- unsigned int isPromoted

3.2.1 Field Documentation

3.2.1.1 c

`unsigned int c`

3.2.1.2 cima

`unsigned int cima`

3.2.1.3 dim

`unsigned int dim`

3.2.1.4 et

`char* et`

3.2.1.5 grado

`unsigned int grado`

3.2.1.6 isPromoted

```
unsigned int isPromoted
```

3.2.1.7 r

```
unsigned int r
```

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

- supporto.h

3.3 player Struct Reference

```
#include <supporto.h>
```

Data Fields

- char colore
- tpedina * arr
- unsigned int dim

3.3.1 Field Documentation

3.3.1.1 arr

```
tpedina* arr
```

3.3.1.2 colore

```
char colore
```

3.3.1.3 dim

```
unsigned int dim
```

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

- supporto.h

Chapter 4

File Documentation

4.1 main.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "supporto.h"
```

Functions

- int main ()

4.1.1 Function Documentation

4.1.1.1 main()

```
int main ( )
```

4.2 menu.c File Reference

```
#include <stdio.h>
#include <ncurses.h>
#include <string.h>
#include <menu.h>
#include <stdlib.h>
```

Functions

- void fail (char *msg)
- int main ()

4.2.1 Function Documentation

4.2.1.1 fail()

```
void fail (
            char * msg )
```

4.2.1.2 main()

```
int main ( )
```

4.3 supporto.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "supporto.h"
#include <string.h>
#include <time.h>
#include <math.h>
#include "src/colors/colors.h"
```

Functions

- tcampo * create_board (unsigned int r, unsigned int col, unsigned int cifre)

  Creazione del campo di gioco.
- void initialize_board (tcampo *t, unsigned int cifre)
- void print_board (tcampo t, unsigned int cifre, unsigned npl)
- void print_directions (unsigned int *arr, unsigned int dim, unsigned int np)
- tplayer * create_pawns (unsigned int n, char ped, unsigned int np, unsigned int cifre, tcampo t)
- void print_player (tplayer p)
- void update_board (tcampo *t, tplayer p1, tplayer p2)
- unsigned int check_spot (tcampo t, unsigned int r, unsigned int c, unsigned int cifre)
- unsigned int is_selected (tplayer p1, tplayer p2, unsigned int np, unsigned int npl)
- void remove_pawn (tcampo *t, unsigned int r, unsigned int c, unsigned cifre)
- void pawn_promotion (tplayer *p, unsigned int np, unsigned int numpl, unsigned int meta)
- unsigned int check_player (tplayer p1, tplayer p2, unsigned int x, unsigned int y)
- unsigned int char_converter (tcampo t, unsigned int r, unsigned int c, unsigned int dim, unsigned int cifre)
- unsigned int is_in (int r, int c, tcampo t)
- unsigned int check_directions (unsigned int *arr, unsigned int dim, char *str)
- unsigned int * must_eat (tplayer p1, tplayer p2, tcampo t, unsigned int np, unsigned int npl)
- int can_eat (tplayer *p1, unsigned int np, char *str, tcampo *t, tplayer *p2, unsigned int pl)
- unsigned int move_noeat (tplayer *p1, unsigned int np, char *str, tcampo *t, tplayer *p2, unsigned int pl)
- unsigned int eat (tplayer *p1, tplayer *p2, char *str, unsigned int np, tcampo t, unsigned int num, unsigned int npl)

- unsigned int move_p1 (tplayer *p1, unsigned int np, char *str, tcampo *t, tplayer *p2, unsigned int pl)
- unsigned int move_p2 (tplayer *p2, unsigned int np, char *str, tcampo *t, tplayer *p1)
- unsigned int all_blocked (tplayer p1, tplayer p2, tcampo t, unsigned int npl)
- unsigned int is_victory (tplayer p1, tplayer p2, tcampo t)
- unsigned int round_choice ()
- unsigned int round_player (tplayer *p1, tplayer *p2, tcampo *t, unsigned int npl)
- int is_empty (tplayer p)
- unsigned int max_pawns (unsigned int r, unsigned int c)
- tcampo * copy_board (tcampo t, tcampo *new)
- tplayer * player_copy (tplayer p, tplayer *n, unsigned int cifre)
- unsigned int is_notstuck (tplayer p1, tplayer p2, tcampo t, unsigned int nped, unsigned int npl)
- char int_converter (unsigned int num, unsigned int index)
- unsigned int add_pawn (tplayer *p1, tplayer *p2, unsigned int np)
- int player_vs_player (unsigned int x)
- unsigned int round_ia (tplayer *p1, tplayer *ia, tcampo *t, unsigned int npl)
- void player_vs_ia ()
- void destroy_player (tplayer *p)
- void destroy_board (tcampo *t)

4.3.1 Function Documentation

4.3.1.1 add_pawn()

```
unsigned int add_pawn (
            tplayer * p1,
            tplayer * p2,
            unsigned int np )
```

4.3.1.2 all_blocked()

```
unsigned int all_blocked (
            tplayer p1,
            tplayer p2,
            tcampo t,
            unsigned int npl )
```

4.3.1.3 can_eat()

```
int can_eat (
            tplayer * p1,
            unsigned int np,
            char * str,
            tcampo * t,
            tplayer * p2,
            unsigned int pl )
```

4.3.1.4 char_converter()

```
unsigned int char_converter (
             tcampo t,
             unsigned int r,
             unsigned int c,
             unsigned int dim,
             unsigned int cifre )
```

4.3.1.5 check_directions()

```
unsigned int check_directions (
             unsigned int * arr,
             unsigned int dim,
             char * str )
```

4.3.1.6 check_player()

```
unsigned int check_player (
             tplayer p1,
             tplayer p2,
             unsigned int x,
             unsigned int y )
```

4.3.1.7 check_spot()

```
unsigned int check_spot (
             tcampo t,
             unsigned int r,
             unsigned int c,
             unsigned int cifre )
```

4.3.1.8 copy_board()

```
tcampo* copy_board (
             tcampo t,
             tcampo * new )
```

4.3.1.9 create_board()

```
tcampo* create_board (
            unsigned int r,
            unsigned int c,
            unsigned int cifre )
```

Creazione del campo di gioco.

Crea una matrice r∗c e per ogni "cella" alloca abbastanza spazio da rappresentare correttamente la pedina. Lo spazio è dato da "altezza massima della torre(3)" + "spazio per la promozione(1)" + "spazio per rappresentare le cifre(variabile)"

Parameters

r	Numero di righe della schacchiera.
c	Numero di colonne della scacchiera.
cifre	Numero di cifre da allocare in modo da rappresentare il numero delle pedine nella scacchiera.

Returns

tcampo* Ritorna il puntatore al campo.

4.3.1.10 create_pawns()

```
tplayer* create_pawns (
            unsigned int n,
            char ped,
            unsigned int np,
            unsigned int cifre,
            tcampo t )
```

4.3.1.11 destroy_board()

```
void destroy_board (
            tcampo * t )
```

4.3.1.12 destroy_player()

```
void destroy_player (
            tplayer * p )
```

4.3.1.13 eat()

```
unsigned int eat (
            tplayer * p1,
            tplayer * p2,
            char * str,
            unsigned int np,
            tcampo t,
            unsigned int num,
            unsigned int npl )
```

4.3.1.14 initialize_board()

```
void initialize_board (
            tcampo * t,
            unsigned int cifre )
```

4.3.1.15 int_converter()

```
char int_converter (
            unsigned int num,
            unsigned int index )
```

4.3.1.16 is_empty()

```
int is_empty (
            tplayer p )
```

4.3.1.17 is_in()

```
unsigned int is_in (
            int r,
            int c,
            tcampo t )
```

4.3.1.18 is_notstuck()

```
unsigned int is_notstuck (
            tplayer p1,
            tplayer p2,
            tcampo t,
            unsigned int nped,
            unsigned int npl )
```

4.3.1.19 is_selected()

```
unsigned int is_selected (
            tplayer p1,
            tplayer p2,
            unsigned int np,
            unsigned int npl )
```

4.3.1.20 is_victory()

```
unsigned int is_victory (
            tplayer p1,
            tplayer p2,
            tcampo t )
```

4.3.1.21 max_pawns()

```
unsigned int max_pawns (
            unsigned int r,
            unsigned int c )
```

4.3.1.22 move_noeat()

```
unsigned int move_noeat (
            tplayer * p1,
            unsigned int np,
            char * str,
            tcampo * t,
            tplayer * p2,
            unsigned int pl )
```

4.3.1.23 move_p1()

```
unsigned int move_p1 (
            tplayer * p1,
            unsigned int np,
            char * str,
            tcampo * t,
            tplayer * p2,
            unsigned int pl )
```

4.3.1.24 move_p2()

```
unsigned int move_p2 (
            tplayer * p2,
            unsigned int np,
            char * str,
            tcampo * t,
            tplayer * p1 )
```

4.3.1.25 must_eat()

```
unsigned int* must_eat (
            tplayer p1,
            tplayer p2,
            tcampo t,
            unsigned int np,
            unsigned int npl )
```

4.3.1.26 pawn_promotion()

```
void pawn_promotion (
            tplayer * p,
            unsigned int np,
            unsigned int numpl,
            unsigned int meta )
```

4.3.1.27 player_copy()

```
tplayer* player_copy (
            tplayer p,
            tplayer * n,
            unsigned int cifre )
```

4.3.1.28 player_vs_ia()

```
void player_vs_ia ( )
```

4.3.1.29 player_vs_player()

```
int player_vs_player (
            unsigned int x )
```

4.3.1.30 print_board()

```
void print_board (
            tcampo t,
            unsigned int cifre,
            unsigned npl )
```

4.3.1.31 print_directions()

```
void print_directions (
            unsigned int * arr,
            unsigned int dim,
            unsigned int np )
```

4.3.1.32 print_player()

```
void print_player (
            tplayer p )
```

4.3.1.33 remove_pawn()

```
void remove_pawn (
            tcampo * t,
            unsigned int r,
            unsigned int c,
            unsigned cifre )
```

4.3.1.34 round_choice()

```
unsigned int round_choice ( )
```

4.3.1.35 round_ia()

```
unsigned int round_ia (
            tplayer * p1,
            tplayer * ia,
            tcampo * t,
            unsigned int npl )
```

4.3.1.36 round_player()

```
unsigned int round_player (
            tplayer * p1,
            tplayer * p2,
            tcampo * t,
            unsigned int npl )
```

4.3.1.37 update_board()

```
void update_board (
            tcampo * t,
            tplayer p1,
            tplayer p2 )
```

4.4 supporto.h File Reference

Data Structures

- struct campo
- struct pedina
- struct player

Typedefs

- typedef struct campo tcampo
- typedef struct pedina tpedina
- typedef struct player tplayer

Functions

- tcampo ∗ create_board (unsigned int r, unsigned int c, unsigned int cifre)

  Creazione del campo di gioco.
- void initialize_board (tcampo ∗t, unsigned int cifre)
- void print_board (tcampo t, unsigned int cifre, unsigned int npl)
- tplayer ∗ create_pawns (unsigned int n, char ped, unsigned int np, unsigned int cifre, tcampo t)
- void print_player (tplayer p)
- void print_directions (unsigned int ∗arr, unsigned int dim, unsigned int np)
- void update_board (tcampo ∗t, tplayer p1, tplayer p2)
- int can_eat (tplayer ∗p1, unsigned int np, char ∗str, tcampo ∗t, tplayer ∗p2, unsigned int pl)
- unsigned int ∗ must_eat (tplayer p1, tplayer p2, tcampo t, unsigned int np, unsigned int npl)
- unsigned int check_directions (unsigned int ∗arr, unsigned int dim, char ∗str)
- unsigned int move_noeat (tplayer ∗p1, unsigned int np, char ∗str, tcampo ∗t, tplayer ∗p2, unsigned int pl)
- unsigned int is_in (int r, int c, tcampo t)
- unsigned int check_spot (tcampo t, unsigned int r, unsigned int c, unsigned int cifre)
- int is_empty (tplayer p)
- unsigned int is_selected (tplayer p1, tplayer p2, unsigned int np, unsigned int npl)
- unsigned int max_pawns (unsigned int r, unsigned int c)
- unsigned int is_notstuck (tplayer p1, tplayer p2, tcampo t, unsigned int nped, unsigned int npl)
- tcampo ∗ copy_board (tcampo t, tcampo ∗new)
- tplayer ∗ player_copy (tplayer p, tplayer ∗n, unsigned int cifre)
- unsigned int add_pawn (tplayer ∗p1, tplayer ∗p2, unsigned int np)
- unsigned int move_p1 (tplayer ∗p1, unsigned int np, char ∗str, tcampo ∗t, tplayer ∗p2, unsigned int pl)
- unsigned int move_p2 (tplayer ∗p2, unsigned int np, char ∗str, tcampo ∗t, tplayer ∗p1)
- void remove_pawn (tcampo ∗t, unsigned int r, unsigned int c, unsigned int cifre)
- void pawn_promotion (tplayer ∗p, unsigned int np, unsigned int numpl, unsigned int meta)
- unsigned int eat (tplayer ∗p1, tplayer ∗p2, char ∗str, unsigned int np, tcampo t, unsigned int num, unsigned int npl)

- unsigned int char_converter (tcampo t, unsigned int r, unsigned int c, unsigned int dim, unsigned int cifre)
- char int_converter (unsigned int num, unsigned int index)
- unsigned int check_player (tplayer p1, tplayer p2, unsigned int x, unsigned int y)
- unsigned int is_victory (tplayer p1, tplayer p2, tcampo t)
- unsigned int all_blocked (tplayer p1, tplayer p2, tcampo t, unsigned int npl)
- unsigned int round_choice ()
- unsigned int round_player (tplayer *p1, tplayer *p2, tcampo *t, unsigned int npl)
- int player_vs_player (unsigned int x)
- void player_vs_ia ()
- unsigned int round_ia (tplayer *p1, tplayer *ia, tcampo *t, unsigned int npl)
- int minimax ()
- void destroy_player (tplayer *p)
- void destroy_board (tcampo *t)

4.4.1 Typedef Documentation

4.4.1.1 tcampo

```
typedef struct campo tcampo
```

4.4.1.2 tpedina

```
typedef struct pedina tpedina
```

4.4.1.3 tplayer

```
typedef struct player tplayer
```

4.4.2 Function Documentation

4.4.2.1 add_pawn()

```
unsigned int add_pawn (
            tplayer * p1,
            tplayer * p2,
            unsigned int np )
```

4.4.2.2 all_blocked()

```
unsigned int all_blocked (
            tplayer p1,
            tplayer p2,
            tcampo t,
            unsigned int npl )
```

4.4.2.3 can_eat()

```
int can_eat (
            tplayer * p1,
            unsigned int np,
            char * str,
            tcampo * t,
            tplayer * p2,
            unsigned int pl )
```

4.4.2.4 char_converter()

```
unsigned int char_converter (
            tcampo t,
            unsigned int r,
            unsigned int c,
            unsigned int dim,
            unsigned int cifre )
```

4.4.2.5 check_directions()

```
unsigned int check_directions (
            unsigned int * arr,
            unsigned int dim,
            char * str )
```

4.4.2.6 check_player()

```
unsigned int check_player (
            tplayer p1,
            tplayer p2,
            unsigned int x,
            unsigned int y )
```

4.4.2.7 check_spot()

```
unsigned int check_spot (
            tcampo t,
            unsigned int r,
            unsigned int c,
            unsigned int cifre )
```

4.4.2.8 copy_board()

```
tcampo* copy_board (
            tcampo t,
            tcampo * new )
```

4.4.2.9 create_board()

```
tcampo* create_board (
            unsigned int r,
            unsigned int c,
            unsigned int cifre )
```

Creazione del campo di gioco.

Crea una matrice r∗c e per ogni "cella" alloca abbastanza spazio da rappresentare correttamente la pedina. Lo spazio è dato da "altezza massima della torre(3)" + "spazio per la promozione(1)" + "spazio per rappresentare le cifre(variabile)"

Parameters

r	Numero di righe della schacchiera.
c	Numero di colonne della scacchiera.
cifre	Numero di cifre da allocare in modo da rappresentare il numero delle pedine nella scacchiera.

Returns

tcampo∗ Ritorna il puntatore al campo.

4.4.2.10 create_pawns()

```
tplayer* create_pawns (
            unsigned int n,
            char ped,
            unsigned int np,
            unsigned int cifre,
            tcampo t )
```

4.4.2.11 destroy_board()

```
void destroy_board (
            tcampo * t )
```

4.4.2.12 destroy_player()

```
void destroy_player (
            tplayer * p )
```

4.4.2.13 eat()

```
unsigned int eat (
            tplayer * p1,
            tplayer * p2,
            char * str,
            unsigned int np,
            tcampo t,
            unsigned int num,
            unsigned int npl )
```

4.4.2.14 initialize_board()

```
void initialize_board (
            tcampo * t,
            unsigned int cifre )
```

4.4.2.15 int_converter()

```
char int_converter (
            unsigned int num,
            unsigned int index )
```

4.4.2.16 is_empty()

```
int is_empty (
            tplayer p )
```

4.4.2.17 is_in()

```
unsigned int is_in (
            int r,
            int c,
            tcampo t )
```

4.4.2.18 is_notstuck()

```
unsigned int is_notstuck (
            tplayer p1,
            tplayer p2,
            tcampo t,
            unsigned int nped,
            unsigned int npl )
```

4.4.2.19 is_selected()

```
unsigned int is_selected (
            tplayer p1,
            tplayer p2,
            unsigned int np,
            unsigned int npl )
```

4.4.2.20 is_victory()

```
unsigned int is_victory (
            tplayer p1,
            tplayer p2,
            tcampo t )
```

4.4.2.21 max_pawns()

```
unsigned int max_pawns (
            unsigned int r,
            unsigned int c )
```

4.4.2.22 minimax()

```
int minimax ( )
```

4.4.2.23 move_noeat()

```
unsigned int move_noeat (
            tplayer * p1,
            unsigned int np,
            char * str,
            tcampo * t,
            tplayer * p2,
            unsigned int pl )
```

4.4.2.24 move_p1()

```
unsigned int move_p1 (
            tplayer * p1,
            unsigned int np,
            char * str,
            tcampo * t,
            tplayer * p2,
            unsigned int pl )
```

4.4.2.25 move_p2()

```
unsigned int move_p2 (
            tplayer * p2,
            unsigned int np,
            char * str,
            tcampo * t,
            tplayer * p1 )
```

4.4.2.26 must_eat()

```
unsigned int* must_eat (
            tplayer p1,
            tplayer p2,
            tcampo t,
            unsigned int np,
            unsigned int npl )
```

4.4.2.27 pawn_promotion()

```
void pawn_promotion (
            tplayer * p,
            unsigned int np,
            unsigned int numpl,
            unsigned int meta )
```

4.4.2.28 player_copy()

```
tplayer* player_copy (
            tplayer p,
            tplayer * n,
            unsigned int cifre )
```

4.4.2.29 player_vs_ia()

```
void player_vs_ia ( )
```

4.4.2.30 player_vs_player()

```
int player_vs_player (
            unsigned int x )
```

4.4.2.31 print_board()

```
void print_board (
            tcampo t,
            unsigned int cifre,
            unsigned int npl )
```

4.4.2.32 print_directions()

```
void print_directions (
            unsigned int * arr,
            unsigned int dim,
            unsigned int np )
```

4.4.2.33 print_player()

```
void print_player (
            tplayer p )
```

4.4.2.34 remove_pawn()

```
void remove_pawn (
            tcampo * t,
            unsigned int r,
            unsigned int c,
            unsigned int cifre )
```

4.4.2.35 round_choice()

```
unsigned int round_choice ( )
```

4.4.2.36 round_ia()

```
unsigned int round_ia (
            tplayer * p1,
            tplayer * ia,
            tcampo * t,
            unsigned int npl )
```

4.4.2.37 round_player()

```
unsigned int round_player (
            tplayer * p1,
            tplayer * p2,
            tcampo * t,
            unsigned int npl )
```

4.4.2.38 update_board()

```
void update_board (
            tcampo * t,
            tplayer p1,
            tplayer p2 )
```

Index