

2o48.hackable.c
16D0

Generated by Doxygen 1.8.10

Fri Oct 30 2015 20:09:10

Contents

1	2o48.hackable.c	1
2	File Index	3
2.1	File List	3
3	File Documentation	5
3.1	2048.c File Reference	5
3.1.1	Macro Definition Documentation	7
3.1.1.1	EDOWN	7
3.1.1.2	ELEFT	7
3.1.1.3	ERIGHT	7
3.1.1.4	EUP	7
3.1.1.5	MAX_BOARD_NUM	7
3.1.1.6	MAX_BOARD_SIZE	7
3.1.1.7	MENU_POSITION_X	8
3.1.1.8	MENU_POSITION_Y	8
3.1.1.9	PWD	8
3.1.1.10	PWD_LEN	8
3.1.1.11	WARNING_POSITION_X	8
3.1.1.12	WARNING_POSITION_Y	8
3.1.2	Function Documentation	8
3.1.2.1	AlignCol(int curcol, int direction)	8
3.1.2.2	AlignLine(int curline, int direction)	9
3.1.2.3	c_checksum()	9
3.1.2.4	c_currentStr(bool show)	10
3.1.2.5	c_forceQuit()	10
3.1.2.6	c_loadStr()	11
3.1.2.7	c_loadStr(int iptN, FILE *fp)	11
3.1.2.8	c_readBoard(int from)	11
3.1.2.9	c_readFromDisk(int boards)	12
3.1.2.10	c_saveBoard(int to, bool jmp)	13
3.1.2.11	c_tryQuit()	14

3.1.2.12	c_version()	15
3.1.2.13	c_warning(char *warn)	15
3.1.2.14	c_writeBoardToDisk(char boards)	16
3.1.2.15	CheckEat(char *a, char *b)	17
3.1.2.16	Clrboard(int boardToClr)	17
3.1.2.17	command()	18
3.1.2.18	die()	19
3.1.2.19	Display(char in)	20
3.1.2.20	Eat(bool isH, int direction)	21
3.1.2.21	EatCol(int curcol, int direction)	22
3.1.2.22	EatLine(const int curline, int direction)	23
3.1.2.23	GetRandNums()	24
3.1.2.24	main()	25
3.1.2.25	play()	26
3.1.2.26	Rando(int N)	27
3.1.2.27	settings()	27
3.1.2.28	showBoard(int offy, int offx)	28
3.1.2.29	welcome()	29
3.1.3	Variable Documentation	29
3.1.3.1	board	29
3.1.3.2	boardseed	29
3.1.3.3	boardstr	29
3.1.3.4	col	30
3.1.3.5	cs_pwd	30
3.1.3.6	curs	30
3.1.3.7	display	30
3.1.3.8	eat	30
3.1.3.9	MAX_RANDOMUM	30
3.1.3.10	N	30
3.1.3.11	NA	30
3.1.3.12	P_RANDOMUM	30
3.1.3.13	point	31
3.1.3.14	row	31
3.2	README.md File Reference	31

Chapter 1

2048.hackable.c

An hackable 2048 game written in C

Used ncurses.h to control the input and output

USAGE

Welcome:

Enter the size of board that you want to play. The size must be 1~9, The version number of game executable will be displayed at center of screen;

Play:

Use arrow left/down/up/right or key hjkl to move Type : to input commands;

Command:

```
#### Stash in memory:
##### s | save [n]
    Save current board and continuing current game in a new board
    n (optional): board number to jump to;
##### s! | saveto [n]
    Save current board to a new board and continuing current game in a current board
    n (optional): board number to save to;
##### r | read [n]
    Read from a board saved in memory
    n (optional):board number to read from;

#### Save in file:
    The file name will be 2048[.d].%X.save,
    where %d is the board number and %X is version number in hexadecimal;
##### w | write
    Write current board to file;
##### wb | writeboard [n]
    Write board specfied to file
    n :board number to save;
##### o | open [n]
    Open a board saved in file
    n (optional): board number to read;

####Gaming option
##### q | quit | (ctrl+c)
    Quit the game with a confirming;
##### q! | quit!
    Quit the game without confirming;
##### wq | writequit
```

Write current board to file and quit without confirming

License

The MIT License (MIT)

Copyright (c) 2015 Librazy

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

2048.c	5
----------------------------------	---

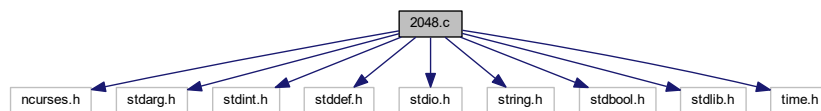
Chapter 3

File Documentation

3.1 2048.c File Reference

```
#include <ncurses.h>
#include <string.h>
#include <stdbool.h>
#include <stdlib.h>
#include <time.h>
```

Include dependency graph for 2048.c:



Macros

- `#define PWD ":2048"`
- `#define PWD_LEN 5`
- `#define MAX_BOARD_NUM 16`
- `#define MAX_BOARD_SIZE 16`
- `#define EUP false,1`
- `#define EDOWN false,-1`
- `#define ELEFT true,1`
- `#define ERIGHT true,-1`
- `#define MENU_POSITION_Y row-1`
- `#define MENU_POSITION_X 0`
- `#define WARNING_POSITION_Y row-2`
- `#define WARNING_POSITION_X 0`

Functions

- void `settings` ()
Set the global settings.
- char `AlignCol` (int curcol, int direction)
Align the vertical direction.

- char [AlignLine](#) (int curline, int direction)
Align the horizontal direction.
- char [CheckEat](#) (char *a, char *b)
Check while the two number can be eaten.
- void [Clrboard](#) (int boardToClr)
Empty the board specified.
- char * [Display](#) (char in)
Get the grid's display string.
- char [Eat](#) (bool isH, int direction)
Eat the board at the direction specified.
- char [EatCol](#) (int curcol, int direction)
Eat the vertical direction.
- char [EatLine](#) (const int curline, int direction)
Eat the horizontal direction.
- int [GetRandNums](#) ()
Get random grid ranged from 0 to MAX_RANDOM on board.
- unsigned int [Rando](#) (int N)
Generate random num ranged from 0 to N-1.
- void [command](#) ()
Show and handle commands inputed by :
- void [die](#) ()
Handle when no empty grid present.
- void [play](#) ()
Handle for main game.
- void [showBoard](#) (int offy, int offx)
Print the board to screen.
- void [welcome](#) ()
Print welcome message and input the size of the board.
- int [c_checksum](#) ()
Calculate the checksum for saving.
- void [c_currentStr](#) (bool show)
Genetate the string representing current board.
- void [c_forceQuit](#) ()
Quit the game.
- void [c_loadStr](#) ()
- void [c_readBoard](#) (int from)
Read the saved board.
- void [c_readFromDisk](#) (int boards)
Read the saved file.
- void [c_saveBoard](#) (int to, bool jmp)
Save the board in memory.
- void [c_tryQuit](#) ()
Ask player whether to quit.
- int [c_version](#) ()
Calculate the game's version.
- void [c_warning](#) (char *warn)
Print a warning to screen.
- bool [c_writeBoardToDisk](#) (char boards)
Write the board to disk.
- void [c_loadStr](#) (int iptN, FILE *fp)
Load the string representing saved board.
- int [main](#) ()
Main executable.

Variables

- const int `NA` =127
- int `P_RANDOM` =30
- int `MAX_RANDOM` =2
- const char `cs_pwd` [`PWD_LEN`+1] =`PWD`
- char `board` [`MAX_BOARD_NUM`][`MAX_BOARD_SIZE`][`MAX_BOARD_SIZE`]
- char `boardstr` [`MAX_BOARD_SIZE`][`MAX_BOARD_SIZE`][5]
- int `boardseed` [`MAX_BOARD_NUM`]
- unsigned char `curs` =0
- char `eat` [256][256][2]
Eat table(TODO:use eat array in CheckEat)
- char `display` [256][16]
- int `point` [256]
Point table(TODO:use eat array in CheckEat)
- char `N` =5
- int `row`
- int `col`

3.1.1 Macro Definition Documentation

3.1.1.1 `#define EDOWN false,-1`

Eat down

Definition at line 43 of file 2048.c.

3.1.1.2 `#define ELEFT true,1`

Eat left

Definition at line 44 of file 2048.c.

3.1.1.3 `#define ERIGHT true,-1`

Rat right

Definition at line 45 of file 2048.c.

3.1.1.4 `#define EUP false,1`

Eat up

Definition at line 42 of file 2048.c.

3.1.1.5 `#define MAX_BOARD_NUM 16`

The maxium board num (for saving in game)

Definition at line 23 of file 2048.c.

3.1.1.6 `#define MAX_BOARD_SIZE 16`

The maxium board size

Definition at line 27 of file 2048.c.

3.1.1.7 #define MENU_POSITION_X 0

the X position to print menu

Definition at line 58 of file 2048.c.

3.1.1.8 #define MENU_POSITION_Y row-1

the Y position to print menu

Definition at line 57 of file 2048.c.

3.1.1.9 #define PWD ":2048"

The password for the save file and represent the version of the game Should and only be changed when the saved file isn't/shouldn't compatible with others

Definition at line 5 of file 2048.c.

3.1.1.10 #define PWD_LEN 5

The length of PWD

Definition at line 9 of file 2048.c.

3.1.1.11 #define WARNING_POSITION_X 0

the X position to print warning

Definition at line 60 of file 2048.c.

3.1.1.12 #define WARNING_POSITION_Y row-2

the Y position to print warning

Definition at line 59 of file 2048.c.

3.1.2 Function Documentation**3.1.2.1 char AlignCol (int *curcol*, int *direction*)**

Align the vertical direction.

Parameters

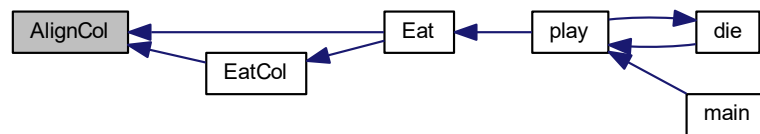
<i>curcol</i>	Current column to align
<i>direction</i>	The direction to align to positive for up and negative for down

Returns

The number of blank grid in the column

Definition at line 173 of file 2048.c.

Here is the caller graph for this function:



3.1.2.2 `char AlignLine (int curline, int direction)`

Align the horizontal direction.

Parameters

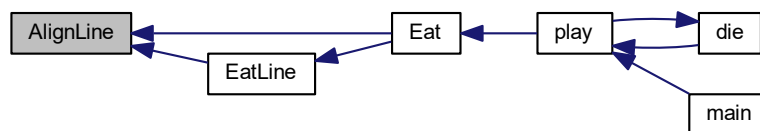
<i>curline</i>	Current line to align
<i>direction</i>	The direction to align to positive for left and negative for right

Returns

The number of blank grid in the column

Definition at line 203 of file 2048.c.

Here is the caller graph for this function:



3.1.2.3 `int c_checksum ()`

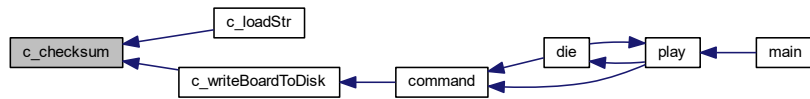
Calculate the checksum for saving.

Returns

The checksum for current board

Definition at line 497 of file 2048.c.

Here is the caller graph for this function:



3.1.2.4 void c_currentStr (bool show)

Genetate the string representing current board.

Parameters

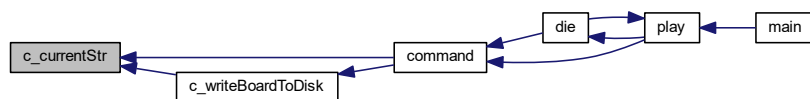
<i>show</i>	If need to print the string to screen
-------------	---------------------------------------

Returns

void

Definition at line 522 of file 2048.c.

Here is the caller graph for this function:



3.1.2.5 void c_forceQuit ()

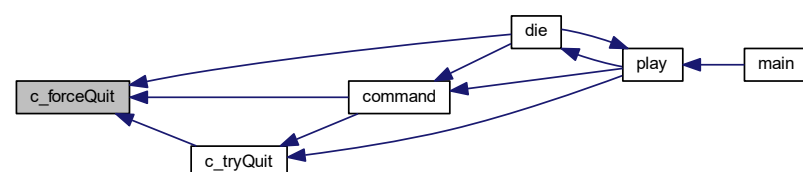
Quit the game.

Returns

void

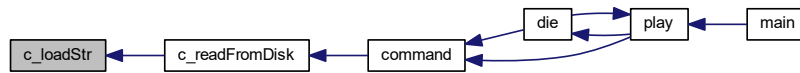
Definition at line 551 of file 2048.c.

Here is the caller graph for this function:



3.1.2.6 void c_loadStr ()

Here is the caller graph for this function:

3.1.2.7 void c_loadStr (int *iptN*, FILE * *fp*)

Load the string representing saved board.

Parameters

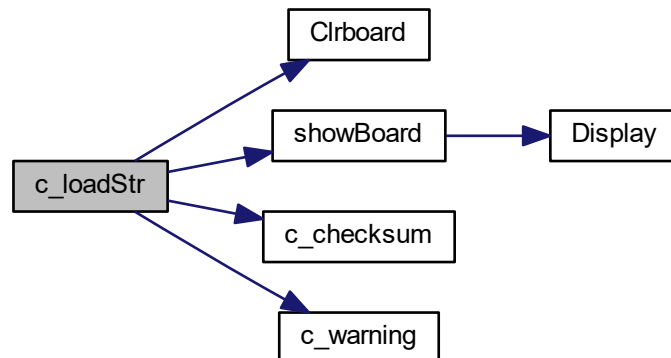
<i>iptN</i>	The N in the saved game
<i>fp</i>	The file stream to read from

Returns

void

Definition at line 560 of file 2048.c.

Here is the call graph for this function:

3.1.2.8 void c_readBoard (int *from*)

Read the saved board.

Parameters

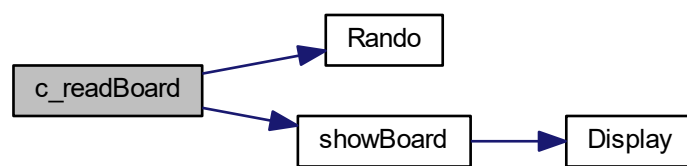
<i>from</i>	The number of board to read from
-------------	----------------------------------

Returns

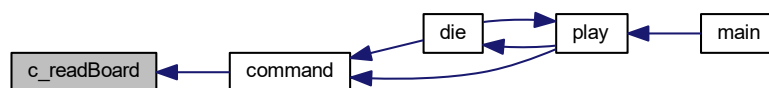
void

Definition at line 607 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:



3.1.2.9 void c_readFromDisk (int boards)

Read the saved file.

Parameters

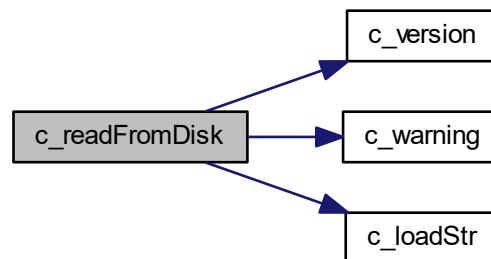
<i>boards</i>	The number of the saved board.NA for not to use
---------------	---

Returns

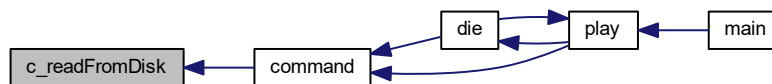
void

Definition at line 627 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:



3.1.2.10 void c_saveBoard (int to, bool jmp)

Save the board in memory.

Parameters

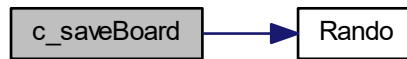
<i>to</i>	The number of the board to save to.NA for auto find nnext
<i>jmp</i>	If should jump to new board

Returns

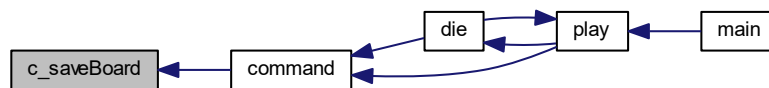
void

Definition at line 663 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:

**3.1.2.11 void c_tryQuit ()**

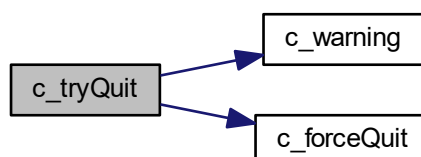
Ask player whether to quit.

Returns

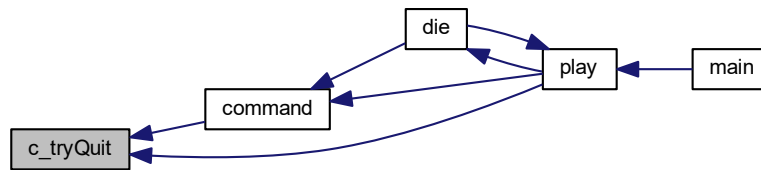
void

Definition at line 691 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:



3.1.2.12 int c_version ()

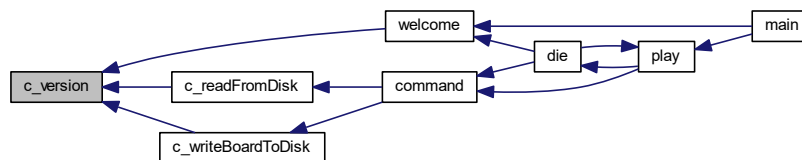
Calculate the game's version.

Returns

Version

Definition at line 682 of file 2048.c.

Here is the caller graph for this function:



3.1.2.13 void c_warning (char * warn)

Print a warning to screen.

Parameters

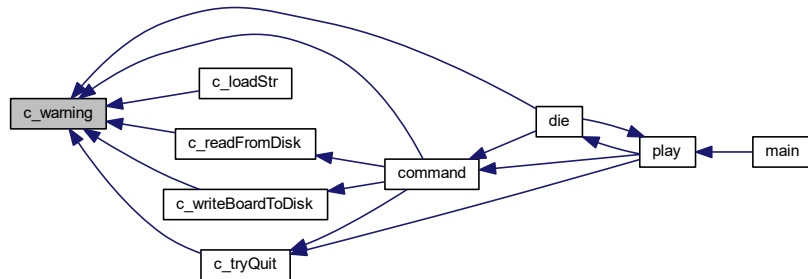
<i>warn</i>	The string to print
-------------	---------------------

Returns

void

Definition at line 767 of file 2048.c.

Here is the caller graph for this function:

**3.1.2.14 bool c_writeBoardToDisk (char *boards*)**

Write the board to disk.

Parameters

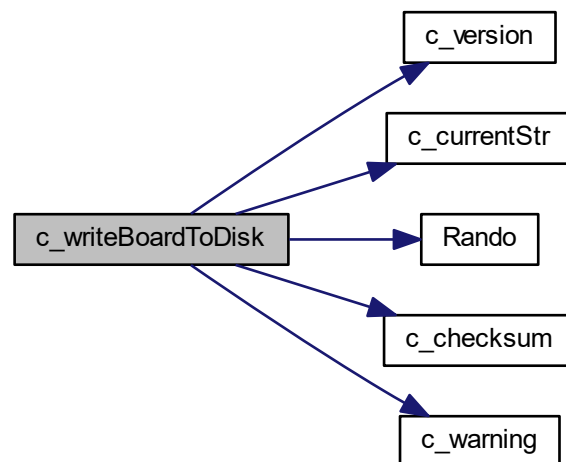
<i>boards</i>	The number of board to save
---------------	-----------------------------

Returns

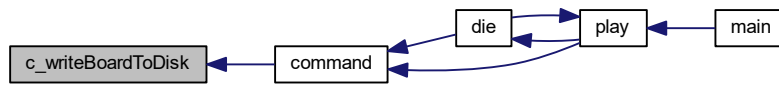
Whether the file is saved successfully

Definition at line 714 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:



3.1.2.15 char CheckEat (char * a, char * b)

Check while the two number can be eaten.

Parameters

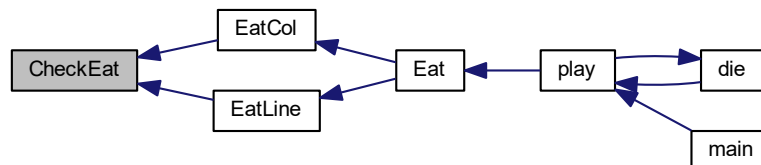
<i>a</i>	Value a
<i>b</i>	Value b

Returns

The final value of a

Definition at line 231 of file 2048.c.

Here is the caller graph for this function:



3.1.2.16 void Clrboard (int boardToClr)

Empty the board specified.

Parameters

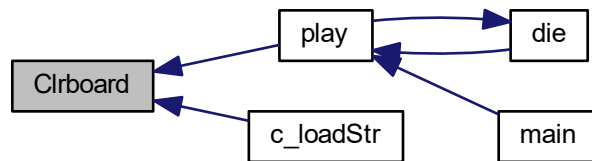
<i>boardToClr</i>	The board to empty
-------------------	--------------------

Returns

void

Definition at line 241 of file 2048.c.

Here is the caller graph for this function:



3.1.2.17 void command ()

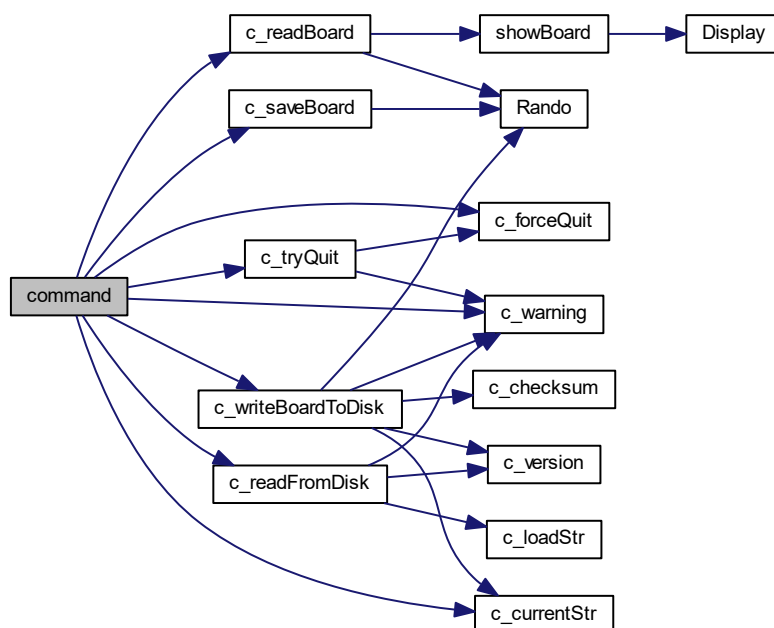
Show and handle commands inputed by :

Returns

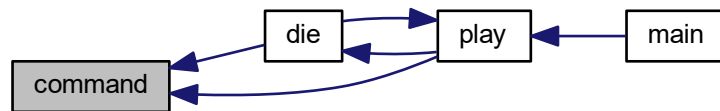
void

Definition at line 346 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:



3.1.2.18 void die ()

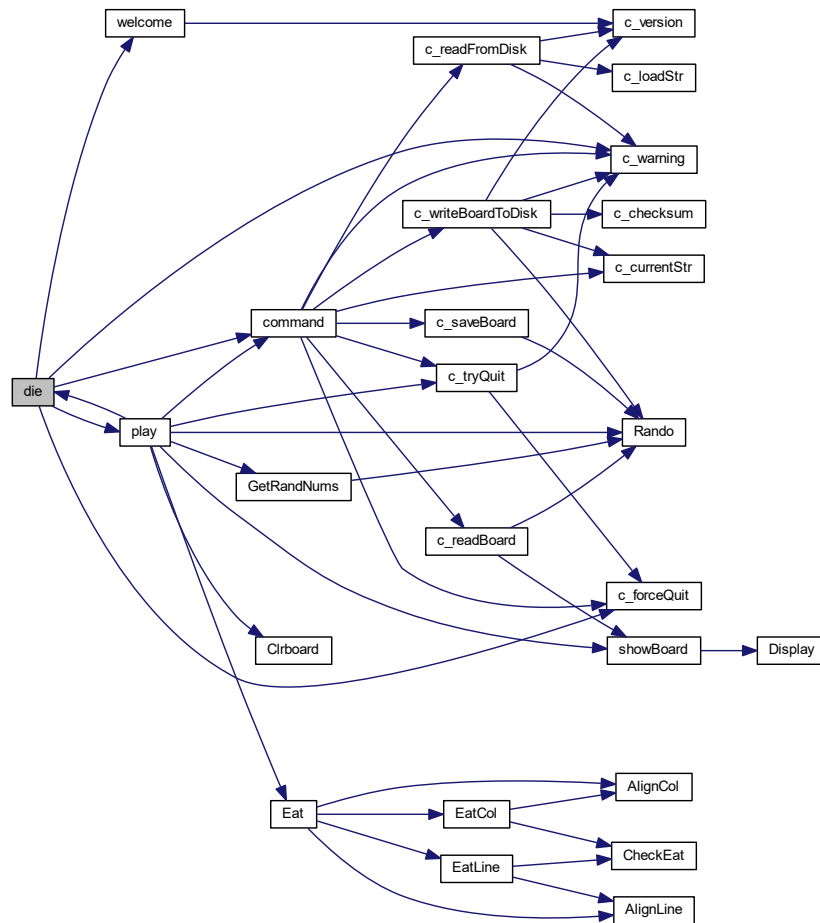
Handle when no empty grid present.

Returns

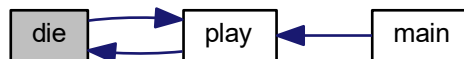
void

Definition at line 388 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:



3.1.2.19 char * Display (char in)

Get the grid's display string.

Parameters

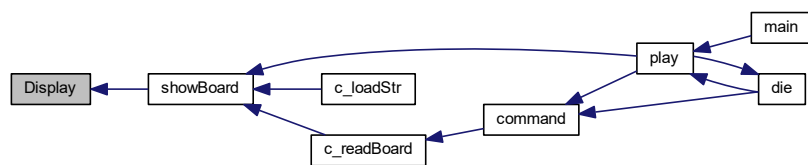
<i>in</i>	The grid's value
-----------	------------------

Returns

The string to display

Definition at line 247 of file 2048.c.

Here is the caller graph for this function:

3.1.2.20 char Eat (bool *isH*, int *direction*)

Eat the board at the direction specified.

Parameters

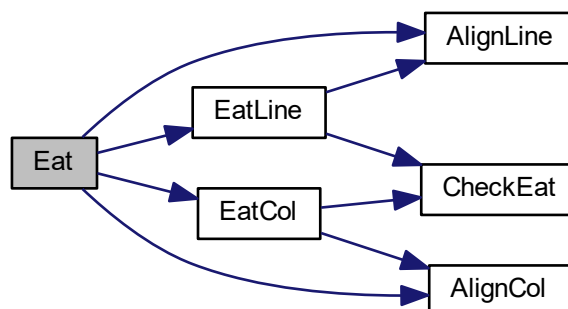
<i>isH</i>	Is horizontal true for horizontal and false for vertical
<i>direction</i>	The direction to eat to positive for left/up and negative for right/down

Returns

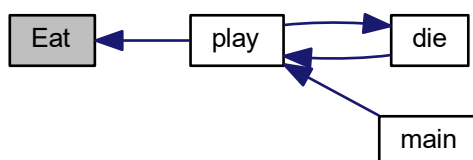
The number of empty grids

Definition at line 259 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:



3.1.2.21 char EatCol (int *curcol*, int *direction*)

Eat the vertical direction.

Parameters

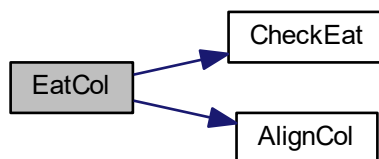
<i>curcol</i>	Current column to eat
<i>direction</i>	The direction to eat positive for up and negative for down

Returns

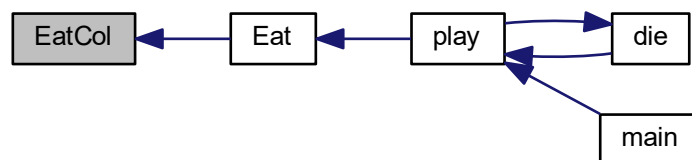
The number of blank grid in the column

Definition at line 280 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:



3.1.2.22 char EatLine (const int *curline*, int *direction*)

Eat the horizontal direction.

Parameters

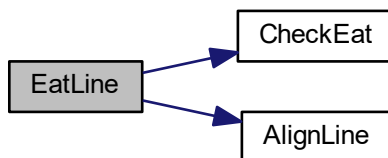
<i>curline</i>	Current line to eat
<i>direction</i>	The direction to eat positive for left and negative for right

Returns

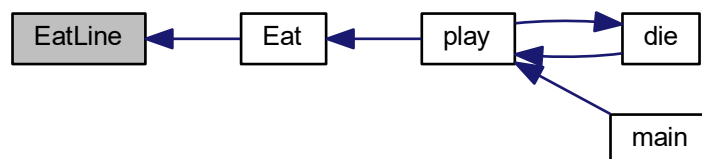
The number of blank grid in the column

Definition at line 295 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:

**3.1.2.23 int GetRandNums ()**

Get random grid ranged from 0 to MAX_RANDOMUM on board.

Returns

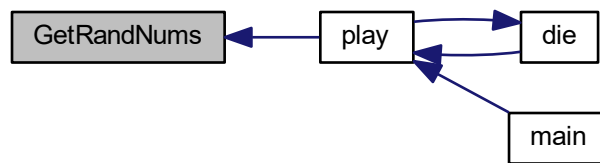
The number of random grid generated

Definition at line 306 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:



3.1.2.24 int main ()

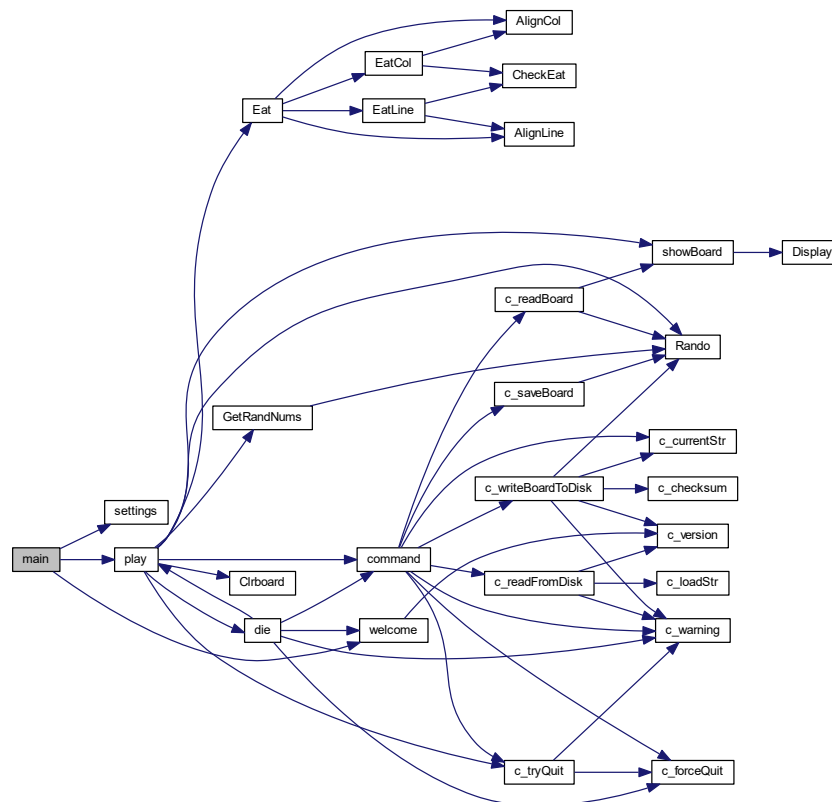
Main executable.

Returns

0

Definition at line 775 of file 2048.c.

Here is the call graph for this function:



3.1.2.25 void play ()

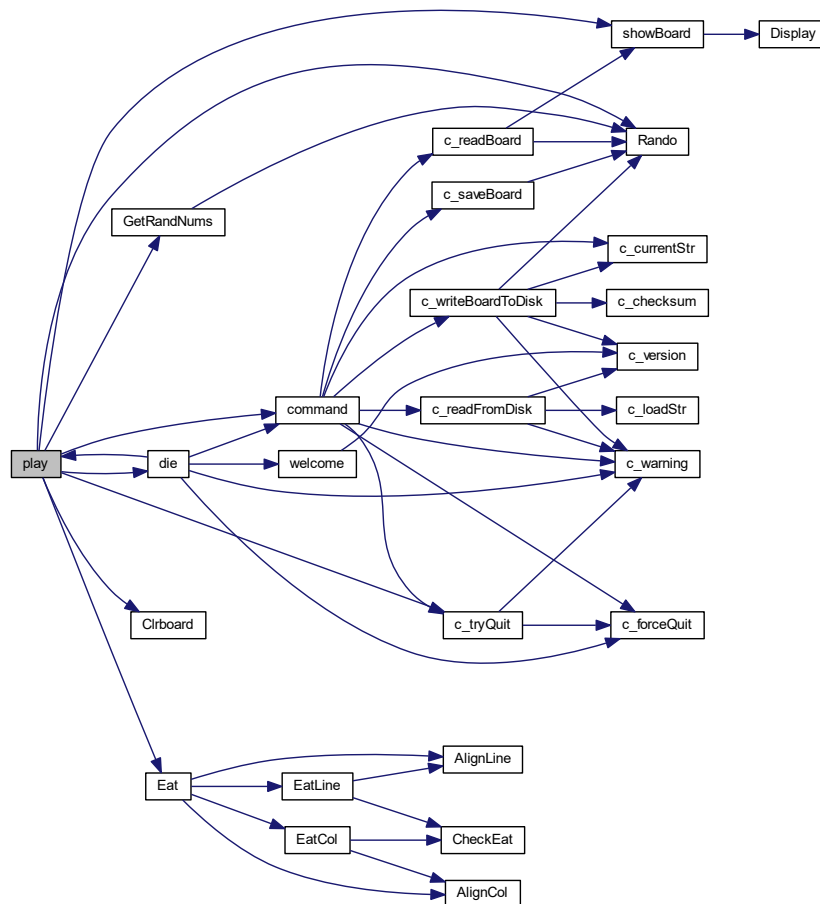
Handle for main game.

Returns

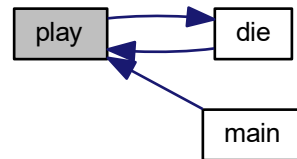
void

Definition at line 410 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:



3.1.2.26 unsigned int Rando (int *N*)

Generate random num ranged from 0 to *N*-1.

Parameters

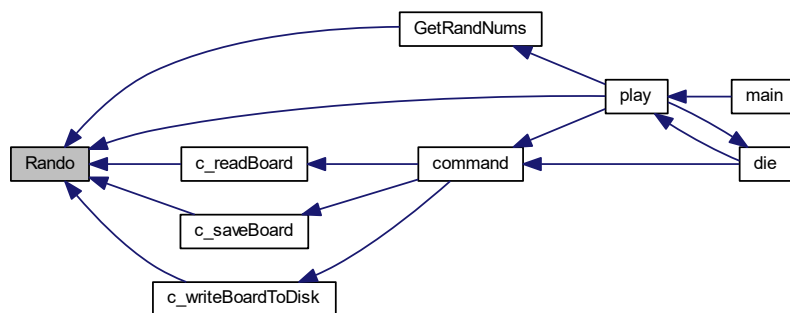
<i>N</i>	The upper bound of the random number
----------	--------------------------------------

Returns

The random number

Definition at line 318 of file 2048.c.

Here is the caller graph for this function:



3.1.2.27 void settings ()

Set the global settings.

TODO:use a ini instead?

Returns

void

Definition at line 95 of file 2048.c.

Here is the caller graph for this function:

**3.1.2.28 void showBoard (int *offy*, int *offx*)**

Print the board to screen.

Parameters

<i>offy</i>	The y position for the left-up corner
<i>offx</i>	The x position for the left-up corner

Returns

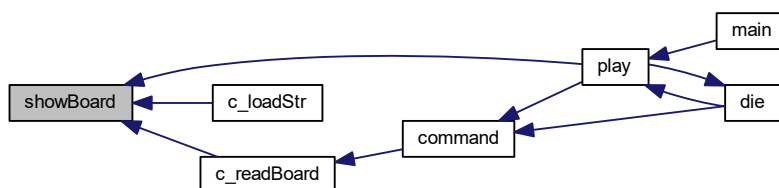
void

Definition at line 451 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:



3.1.2.29 void welcome ()

Print welcome message and input the size of the board.

Returns

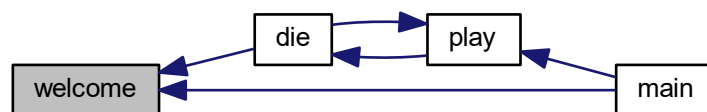
void

Definition at line 478 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:



3.1.3 Variable Documentation

3.1.3.1 char board[MAX_BOARD_NUM][MAX_BOARD_SIZE][MAX_BOARD_SIZE]

The boards to storage game progress

Definition at line 64 of file 2048.c.

3.1.3.2 int boardseed[MAX_BOARD_NUM]

The random seed of the boards

Definition at line 68 of file 2048.c.

3.1.3.3 char boardstr[MAX_BOARD_SIZE][MAX_BOARD_SIZE][5]

The output string

Definition at line 66 of file 2048.c.

3.1.3.4 int col

Definition at line 89 of file 2048.c.

3.1.3.5 const char cs_pwd[PWD_LEN+1] =PWD

The password when generating the checksum

Definition at line 62 of file 2048.c.

3.1.3.6 unsigned char curs =0

Current board

Definition at line 70 of file 2048.c.

3.1.3.7 char display[256][16]

Display table

Will display a as string display[a]

Definition at line 80 of file 2048.c.

3.1.3.8 char eat[256][256][2]

Eat table(TODO:use eat array in CheckEat)

Will set a=eat[a][b][0] and b=eat[a][b][1] when eating a and b

Definition at line 76 of file 2048.c.

3.1.3.9 int MAX_RANDOM =2

The maxium level of filling an grid.

Definition at line 29 of file 2048.c.

3.1.3.10 char N =5

The size of the board

Definition at line 86 of file 2048.c.

3.1.3.11 const int NA =127

Stand for invalid grid.

Definition at line 19 of file 2048.c.

3.1.3.12 int P_RANDOM =30

The probability of an empty grid becoming filled

Definition at line 28 of file 2048.c.

3.1.3.13 int point[256]

Point table(TODO:use eat array in CheckEat)

Will count a by point[a] when adding up the points

Definition at line 84 of file 2048.c.

3.1.3.14 int row

The size of the screen

Definition at line 89 of file 2048.c.

3.2 README.md File Reference

Index

2048.c, [5](#)

AlignCol, [8](#)

AlignLine, [9](#)

board, [29](#)

boardseed, [29](#)

boardstr, [29](#)

c_checksum, [9](#)

c_currentStr, [10](#)

c_forceQuit, [10](#)

c_loadStr, [10](#), [11](#)

c_readBoard, [11](#)

c_readFromDisk, [12](#)

c_saveBoard, [13](#)

c_tryQuit, [14](#)

c_version, [15](#)

c_warning, [15](#)

c_writeBoardToDisk, [16](#)

CheckEat, [17](#)

Clrboard, [17](#)

col, [29](#)

command, [18](#)

cs_pwd, [30](#)

curs, [30](#)

die, [19](#)

Display, [20](#)

display, [30](#)

EDOWN, [7](#)

ELLEFT, [7](#)

ERIGHT, [7](#)

EUP, [7](#)

Eat, [21](#)

eat, [30](#)

EatCol, [22](#)

EatLine, [23](#)

GetRandNums, [24](#)

MAX_BOARD_NUM, [7](#)

MAX_BOARD_SIZE, [7](#)

MAX_RANDOM, [30](#)

MENU_POSITION_X, [7](#)

MENU_POSITION_Y, [8](#)

main, [25](#)

N, [30](#)

NA, [30](#)

P_RANDOM, [30](#)

PWD, [8](#)

PWD_LEN, [8](#)

play, [25](#)

point, [30](#)

Rando, [27](#)

row, [31](#)

settings, [27](#)

showBoard, [28](#)

WARNING_POSITION_X, [8](#)

WARNING_POSITION_Y, [8](#)

welcome, [28](#)

AlignCol

2048.c, [8](#)

AlignLine

2048.c, [9](#)

board

2048.c, [29](#)

boardseed

2048.c, [29](#)

boardstr

2048.c, [29](#)

c_checksum

2048.c, [9](#)

c_currentStr

2048.c, [10](#)

c_forceQuit

2048.c, [10](#)

c_loadStr

2048.c, [10](#), [11](#)

c_readBoard

2048.c, [11](#)

c_readFromDisk

2048.c, [12](#)

c_saveBoard

2048.c, [13](#)

c_tryQuit

2048.c, [14](#)

c_version

2048.c, [15](#)

c_warning

2048.c, [15](#)

c_writeBoardToDisk

2048.c, [16](#)

CheckEat

2048.c, [17](#)

Clrboard

2048.c, [17](#)

col

2048.c, [29](#)

command

2048.c, [18](#)

cs_pwd

- 2048.c, [30](#)
- curs
 - 2048.c, [30](#)
- die
 - 2048.c, [19](#)
- Display
 - 2048.c, [20](#)
- display
 - 2048.c, [30](#)
- EDOWN
 - 2048.c, [7](#)
- ELEFT
 - 2048.c, [7](#)
- ERIGHT
 - 2048.c, [7](#)
- EUP
 - 2048.c, [7](#)
- Eat
 - 2048.c, [21](#)
- eat
 - 2048.c, [30](#)
- EatCol
 - 2048.c, [22](#)
- EatLine
 - 2048.c, [23](#)
- GetRandNums
 - 2048.c, [24](#)
- MAX_BOARD_NUM
 - 2048.c, [7](#)
- MAX_BOARD_SIZE
 - 2048.c, [7](#)
- MAX_RANDOM
 - 2048.c, [30](#)
- MENU_POSITION_X
 - 2048.c, [7](#)
- MENU_POSITION_Y
 - 2048.c, [8](#)
- main
 - 2048.c, [25](#)
- N
 - 2048.c, [30](#)
- NA
 - 2048.c, [30](#)
- P_RANDOM
 - 2048.c, [30](#)
- PWD
 - 2048.c, [8](#)
- PWD_LEN
 - 2048.c, [8](#)
- play
 - 2048.c, [25](#)
- point
 - 2048.c, [30](#)
- README.md, [31](#)
- Rando
 - 2048.c, [27](#)
- row
 - 2048.c, [31](#)
- settings
 - 2048.c, [27](#)
- showBoard
 - 2048.c, [28](#)
- WARNING_POSITION_X
 - 2048.c, [8](#)
- WARNING_POSITION_Y
 - 2048.c, [8](#)
- welcome
 - 2048.c, [28](#)