

2o48.hackable.c

16D0

Generated by Doxygen 1.8.10

Wed Oct 28 2015 14:18:44

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	8
3.1.1.6	MAX_BOARD_SIZE	8
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)	12
3.1.2.9	c_readFromDisk(int boards)	13
3.1.2.10	c_saveBoard(int to, bool jmp)	13
3.1.2.11	c_tryQuit()	14

3.1.2.12	c_warning(char *warn)	15
3.1.2.13	c_writeBoardToDisk(char boards)	15
3.1.2.14	CheckEat(char *a, char *b)	16
3.1.2.15	Clrboard(int boardToClr)	17
3.1.2.16	command()	17
3.1.2.17	die()	18
3.1.2.18	Display(char in)	19
3.1.2.19	Eat(bool isH, int direction)	20
3.1.2.20	EatCol(int curcol, int direction)	21
3.1.2.21	EatLine(const int curline, int direction)	22
3.1.2.22	GetRandNums()	23
3.1.2.23	main()	23
3.1.2.24	play()	24
3.1.2.25	Rando(int N)	26
3.1.2.26	settings()	27
3.1.2.27	showBoard(int offy, int offx)	27
3.1.2.28	welcome()	28
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	29
3.1.3.5	cs_pwd	29
3.1.3.6	curs	29
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	30
3.1.3.14	row	30
3.2	README.md File Reference	30

Chapter 1

2048.hackable.c

An hackable 2048 game written in C

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

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

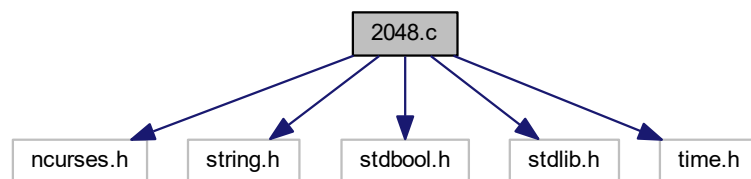
Chapter 3

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 `ELLEFT` `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.
- 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_RANDOMUM` =30
- int `MAX_RANDOMUM` =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 40 of file 2048.c.

3.1.1.2 #define ELEFT true,1

Eat left

Definition at line 41 of file 2048.c.

3.1.1.3 #define ERIGHT true,-1

Rat right

Definition at line 42 of file 2048.c.

3.1.1.4 #define EUP false,1

Eat up

Definition at line 39 of file 2048.c.

3.1.1.5 `#define MAX_BOARD_NUM 16`

The maxium board num (for saving in game)

Definition at line 20 of file 2048.c.

3.1.1.6 `#define MAX_BOARD_SIZE 16`

The maxium board size

Definition at line 24 of file 2048.c.

3.1.1.7 `#define MENU_POSITION_X 0`

the X position to print menu

Definition at line 55 of file 2048.c.

3.1.1.8 `#define MENU_POSITION_Y row-1`

the Y position to print menu

Definition at line 54 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 57 of file 2048.c.

3.1.1.12 `#define WARNING_POSITION_Y row-2`

the Y position to print warning

Definition at line 56 of file 2048.c.

3.1.2 Function Documentation

3.1.2.1 `char AlignCol (int curcol, int direction)`

Align the vertical direction.

Parameters

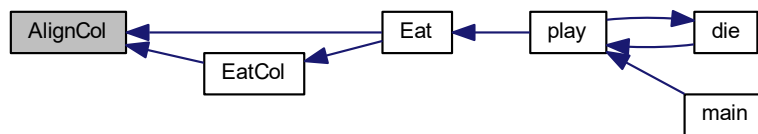
<i>curcol</i>	Current column to align
<i>direction</i>	The direction to align to positive for up and negative for down

Returns

The number of blank grid in the column

Definition at line 169 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

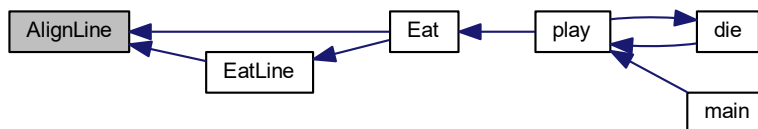
<i>curline</i>	Current line to align
<i>direction</i>	The direction to align to positive for left and negative for right

Returns

The number of blank grid in the column

Definition at line 198 of file 2048.c.

Here is the caller graph for this function:



3.1.2.3 int c_checksum ()

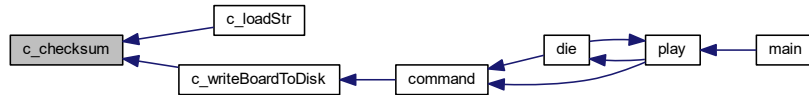
Calculate the checksum for saving.

Returns

The checksum for current board

Definition at line 492 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

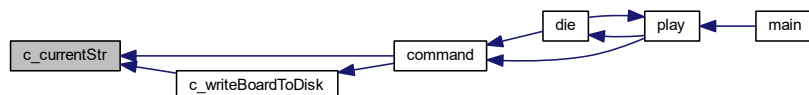
<i>show</i>	If need to print the string to screen
-------------	---------------------------------------

Returns

void

Definition at line 517 of file 2048.c.

Here is the caller graph for this function:

**3.1.2.5 void c_forceQuit ()**

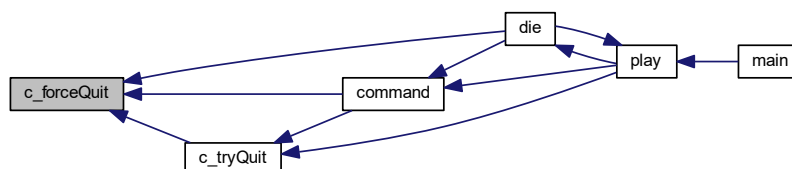
Quit the game.

Returns

void

Definition at line 546 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

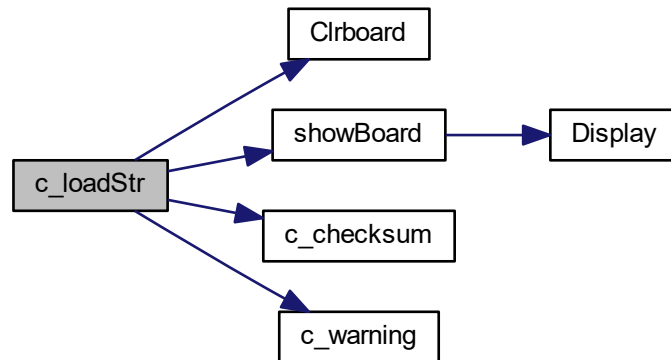
<i>iptN</i>	The N in the saved game
<i>fp</i>	The file stream to read from

Returns

void

Definition at line 555 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

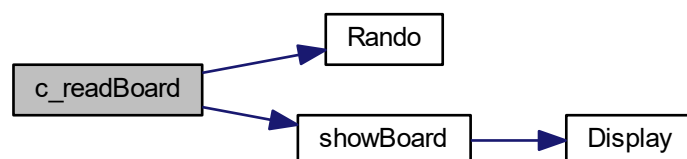
<i>from</i>	The number of board to read from
-------------	----------------------------------

Returns

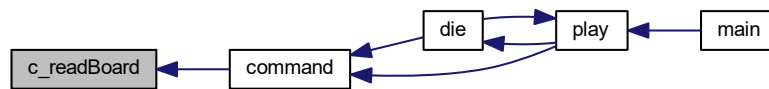
void

Definition at line 602 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

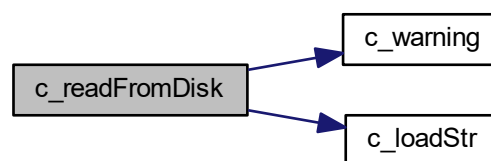
<i>boards</i>	The number of the saved board.NA for not to use
---------------	---

Returns

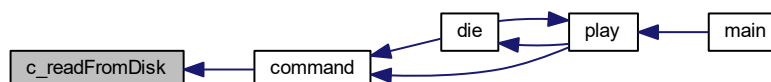
void

Definition at line 619 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

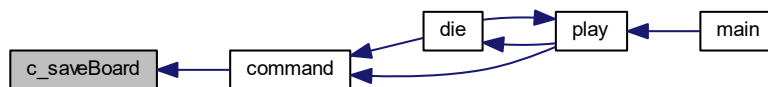
void

Definition at line 657 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 ()

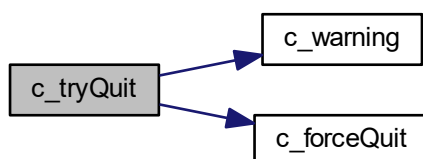
Ask player whether to quit.

Returns

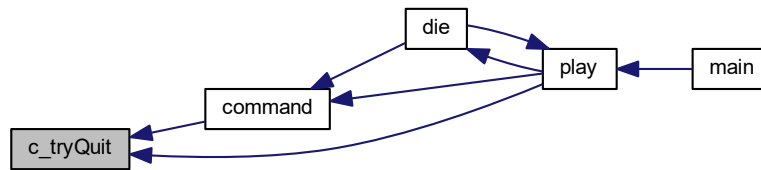
void

Definition at line 671 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:



3.1.2.12 void c_warning (char * warn)

Print a warning to screen.

Parameters

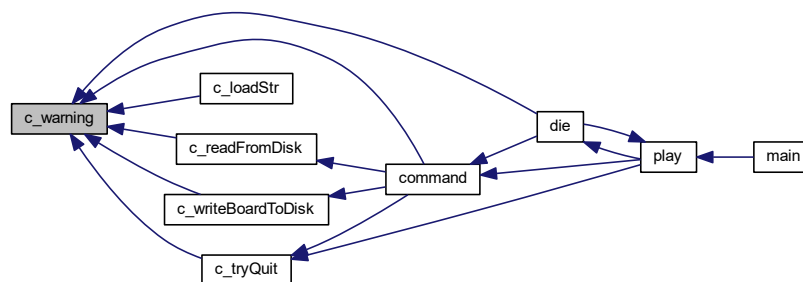
<i>warn</i>	The string to print
-------------	---------------------

Returns

void

Definition at line 746 of file 2048.c.

Here is the caller graph for this function:



3.1.2.13 bool c_writeBoardToDisk (char boards)

Write the board to disk.

Parameters

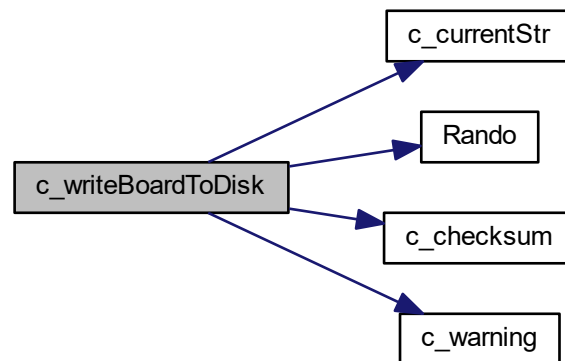
<i>boards</i>	The number of board to save
---------------	-----------------------------

Returns

Whether the file is saved successfully

Definition at line 694 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:



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

Check while the two number can be eaten.

Parameters

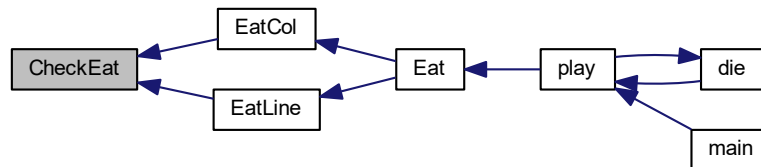
<i>a</i>	Value a
<i>b</i>	Value b

Returns

The final value of a

Definition at line 226 of file 2048.c.

Here is the caller graph for this function:

**3.1.2.15 void Clrboard (int *boardToClr*)**

Empty the board specified.

Parameters

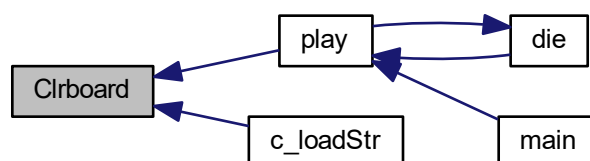
<i>boardToClr</i>	The board to empty
-------------------	--------------------

Returns

void

Definition at line 236 of file 2048.c.

Here is the caller graph for this function:

**3.1.2.16 void command ()**

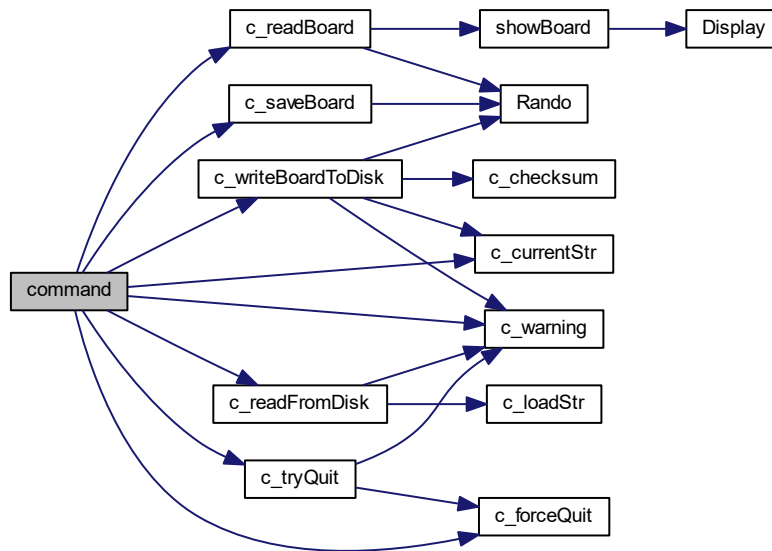
Show and handle commands inputed by :

Returns

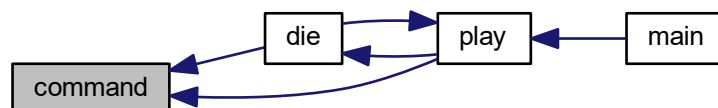
void

Definition at line 336 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:

**3.1.2.17 void die ()**

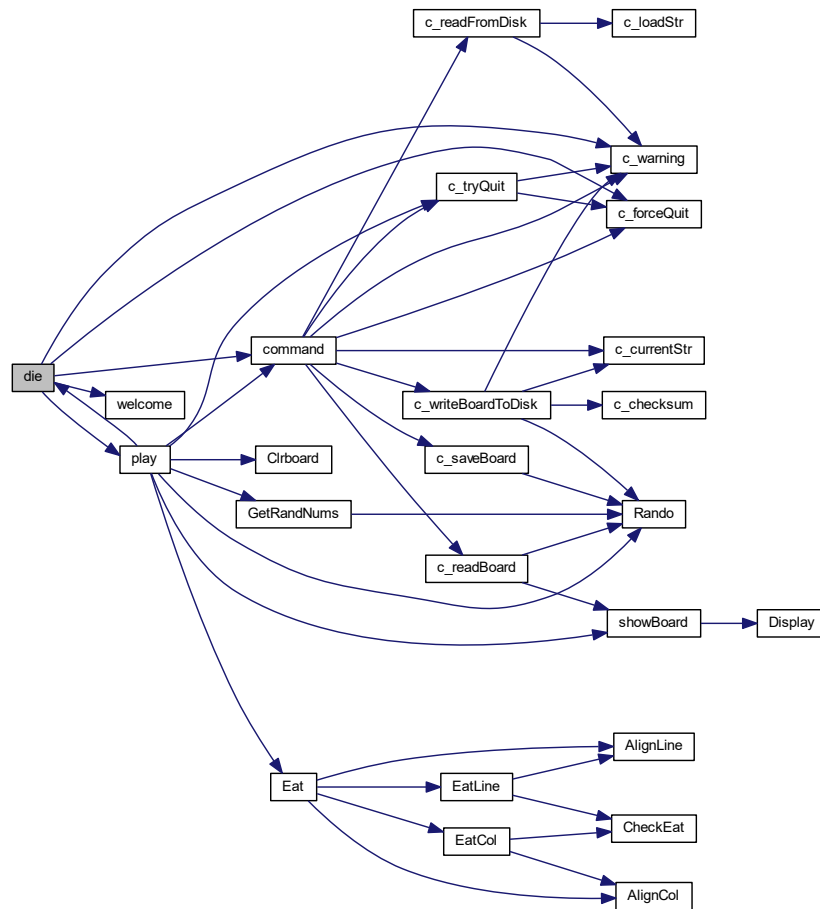
Handle when no empty grid present.

Returns

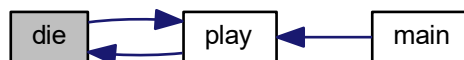
void

Definition at line 380 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:



3.1.2.18 char * Display (char in)

Get the grid's display string.

Parameters

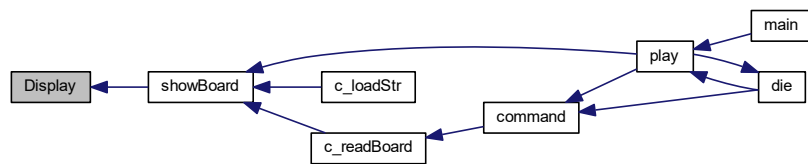
<i>in</i>	The grid's value
-----------	------------------

Returns

The string to display

Definition at line 242 of file 2048.c.

Here is the caller graph for this function:

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

Eat the board at the direction specified.

Parameters

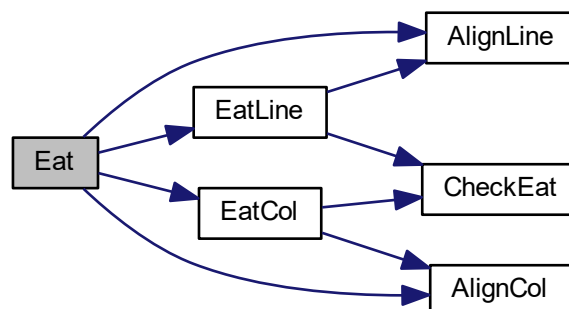
<i>isH</i>	Is horizontal true for horizontal and false for vertical
<i>direction</i>	The direction to eat to positive for left/up and negative for right/down

Returns

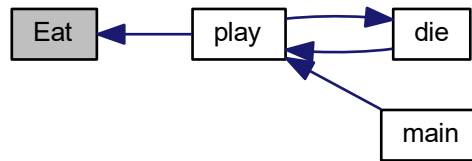
The number of empty grids

Definition at line 252 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:



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

Eat the vertical direction.

Parameters

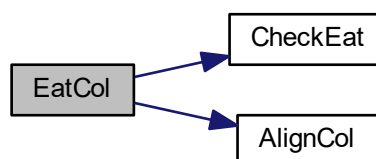
<i>curcol</i>	Current column to eat
<i>direction</i>	The direction to eat positive for up and negative for down

Returns

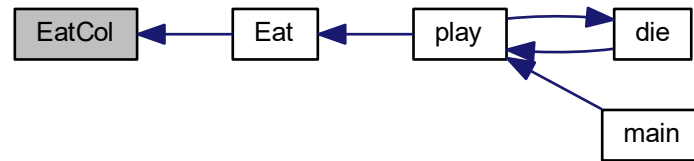
The number of blank grid in the column

Definition at line 272 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 EatLine (const int *curline*, int *direction*)

Eat the horizontal direction.

Parameters

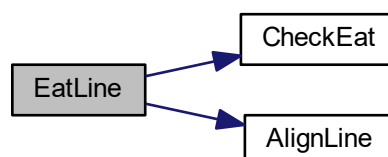
<i>curline</i>	Current line to eat
<i>direction</i>	The direction to eat positive for left and negative for right

Returns

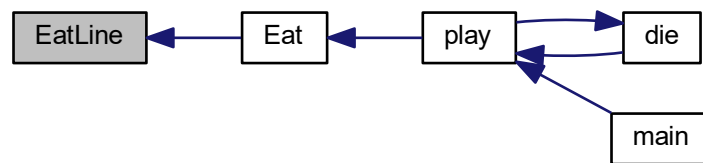
The number of blank grid in the column

Definition at line 286 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:



3.1.2.22 int GetRandNums ()

Get random grid ranged from 0 to MAX_RANDOMUM on board.

Returns

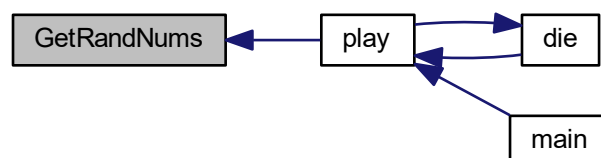
The number of random grid generated

Definition at line 297 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 main ()

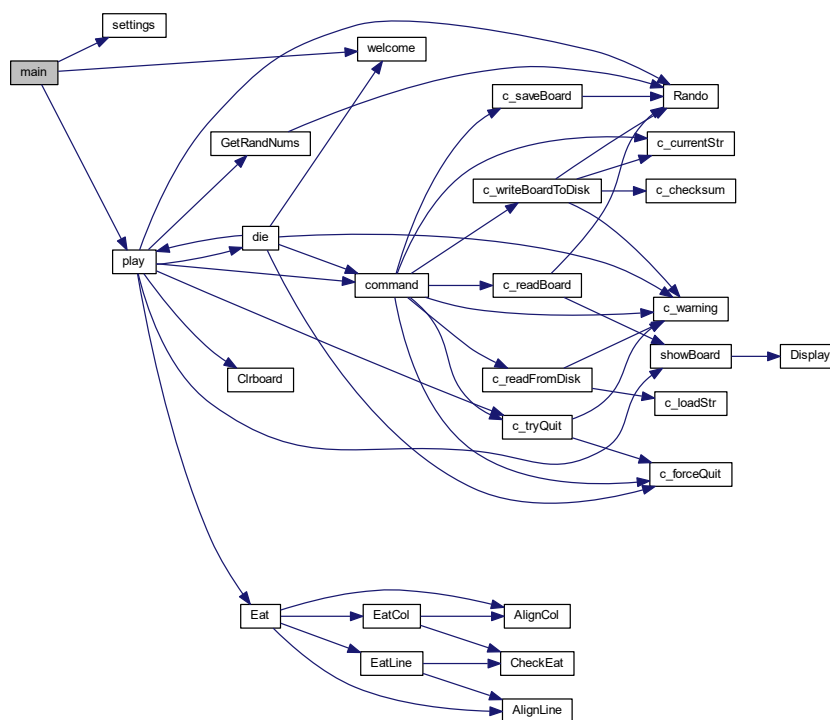
Main executable.

Returns

0

Definition at line 754 of file 2048.c.

Here is the call graph for this function:



3.1.2.24 void play ()

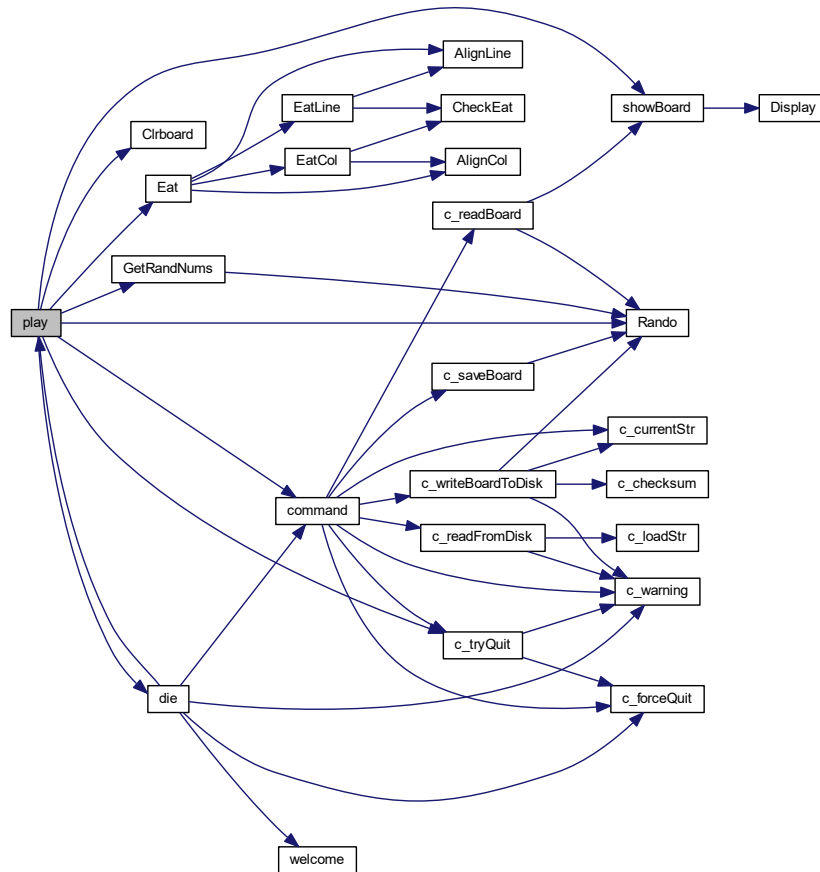
Handle for main game.

Returns

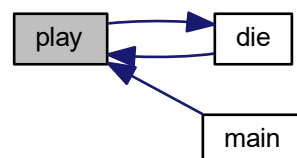
void

Definition at line 402 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:



3.1.2.25 unsigned int Rando (int *N*)

Generate random num ranged from 0 to N-1.

Parameters

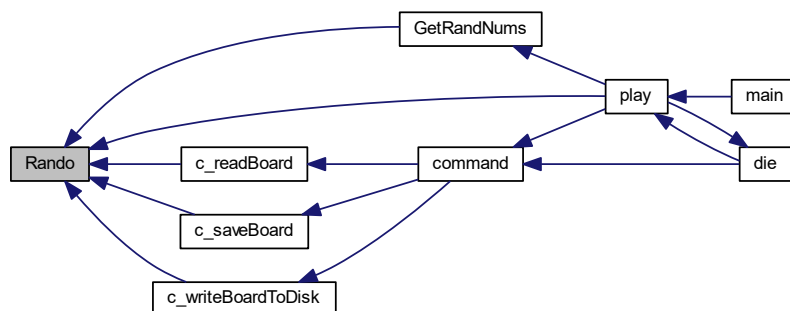
<i>N</i>	The upper bound of the random number
----------	--------------------------------------

Returns

The random number

Definition at line 309 of file 2048.c.

Here is the caller graph for this function:



3.1.2.26 void settings ()

Set the global settings.

TODO:use a ini instead?

Returns

void

Definition at line 92 of file 2048.c.

Here is the caller graph for this function:



3.1.2.27 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

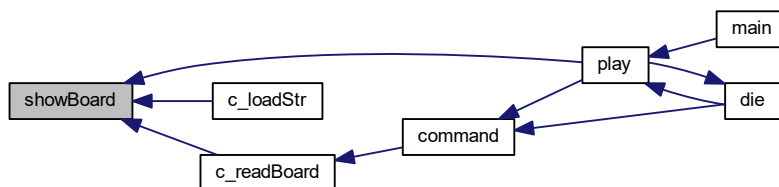
void

Definition at line 443 of file 2048.c.

Here is the call graph for this function:



Here is the caller graph for this function:



3.1.2.28 void welcome ()

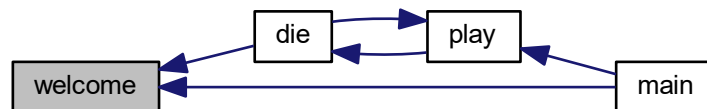
Print welcome message and input the size of the board.

Returns

void

Definition at line 470 of file 2048.c.

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 61 of file 2048.c.

3.1.3.2 `int boardseed[MAX_BOARD_NUM]`

The random seed of the boards

Definition at line 65 of file 2048.c.

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

The output string

Definition at line 63 of file 2048.c.

3.1.3.4 `int col`

Definition at line 86 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 59 of file 2048.c.

3.1.3.6 `unsigned char curs=0`

Current board

Definition at line 67 of file 2048.c.

3.1.3.7 char display[256][16]

Display table

Will display a as string display[a]

Definition at line 77 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 73 of file 2048.c.

3.1.3.9 int MAX_RANDOM =2

The maxium level of filling an grid.

Definition at line 26 of file 2048.c.

3.1.3.10 char N =5

The size of the board

Definition at line 83 of file 2048.c.

3.1.3.11 const int NA =127

Stand for invalid grid.

Definition at line 16 of file 2048.c.

3.1.3.12 int P_RANDOM =30

The probability of an empty grid becoming filled

Definition at line 25 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 81 of file 2048.c.

3.1.3.14 int row

The size of the screen

Definition at line 86 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, [11](#)

c_readBoard, [12](#)

c_readFromDisk, [13](#)

c_saveBoard, [13](#)

c_tryQuit, [14](#)

c_warning, [15](#)

c_writeBoardToDisk, [15](#)

CheckEat, [16](#)

Clrboard, [17](#)

col, [29](#)

command, [17](#)

cs_pwd, [29](#)

curs, [29](#)

die, [18](#)

Display, [19](#)

display, [29](#)

EDOWN, [7](#)

ELEFT, [7](#)

ERIGHT, [7](#)

EUP, [7](#)

Eat, [20](#)

eat, [30](#)

EatCol, [21](#)

EatLine, [22](#)

GetRandNums, [23](#)

MAX_BOARD_NUM, [7](#)

MAX_BOARD_SIZE, [8](#)

MAX_RANDOM, [30](#)

MENU_POSITION_X, [8](#)

MENU_POSITION_Y, [8](#)

main, [23](#)

N, [30](#)

NA, [30](#)

P_RANDOM, [30](#)

PWD, [8](#)

PWD_LEN, [8](#)

play, [24](#)

point, [30](#)

Rando, [25](#)

row, [30](#)

settings, [27](#)

showBoard, [27](#)

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, 11](#)

c_readBoard

[2048.c, 12](#)

c_readFromDisk

[2048.c, 13](#)

c_saveBoard

[2048.c, 13](#)

c_tryQuit

[2048.c, 14](#)

c_warning

[2048.c, 15](#)

c_writeBoardToDisk

[2048.c, 15](#)

CheckEat

[2048.c, 16](#)

Clrboard

[2048.c, 17](#)

col

[2048.c, 29](#)

command

[2048.c, 17](#)

cs_pwd

[2048.c, 29](#)

curs

[2048.c, 29](#)

die
 2048.c, [18](#)

Display
 2048.c, [19](#)

display
 2048.c, [29](#)

EDOWN
 2048.c, [7](#)

ELEFT
 2048.c, [7](#)

ERIGHT
 2048.c, [7](#)

EUP
 2048.c, [7](#)

Eat
 2048.c, [20](#)

eat
 2048.c, [30](#)

EatCol
 2048.c, [21](#)

EatLine
 2048.c, [22](#)

GetRandNums
 2048.c, [23](#)

MAX_BOARD_NUM
 2048.c, [7](#)

MAX_BOARD_SIZE
 2048.c, [8](#)

MAX_RANDOMUM
 2048.c, [30](#)

MENU_POSITION_X
 2048.c, [8](#)

MENU_POSITION_Y
 2048.c, [8](#)

main
 2048.c, [23](#)

N
 2048.c, [30](#)

NA
 2048.c, [30](#)

P_RANDOMUM
 2048.c, [30](#)

PWD
 2048.c, [8](#)

PWD_LEN
 2048.c, [8](#)

play
 2048.c, [24](#)

point
 2048.c, [30](#)

README.md, [30](#)

Rando
 2048.c, [25](#)

row
 2048.c, [30](#)

settings
 2048.c, [27](#)

showBoard
 2048.c, [27](#)

WARNING_POSITION_X
 2048.c, [8](#)

WARNING_POSITION_Y
 2048.c, [8](#)

welcome
 2048.c, [28](#)