Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki

Programowanie Komputerów 2

Koło fortuny

autor	Hanna Podeszwa
prowadzący	mgr inż. Wojciech Dudzik
rok akademicki	2018/2019
kierunek	informatyka
rodzaj studiów	SSI
semestr	2
termin laboratorium	wtorek, 12:00 – 13:30
grupa	7
termin oddania sprawozdania	2019-06-16

1 Treść zadania

Napisać program symulujący grę "Koło fortuny". Program wczytuje słowa z pliku wejściowego. Słowa umieszczane są w dynamicznej tablicy (tablica wskaźników na słowa). Jest dostępna opcja wyboru siły krecenia kołem. Przed grą losowane jest słowo do odgadnięcia. Gracz kreci kołem (losuje punkty/nagrodę). Następnie (jeśli ma możliwość) gracz stara się odgadnąć dane słowo, wpisując kolejne litery (spółgłoski) lub może spróbować odgadnąć całe słowo. Ma też możliwość kupienia samogłoski. Wyniki gry są zapisywane do pliku. W grze może brać udział od jednego do czterech graczy.

Program uruchamiany jest z linii poleceń:

`argv[1]` liczba graczy

2 Analiza zadania

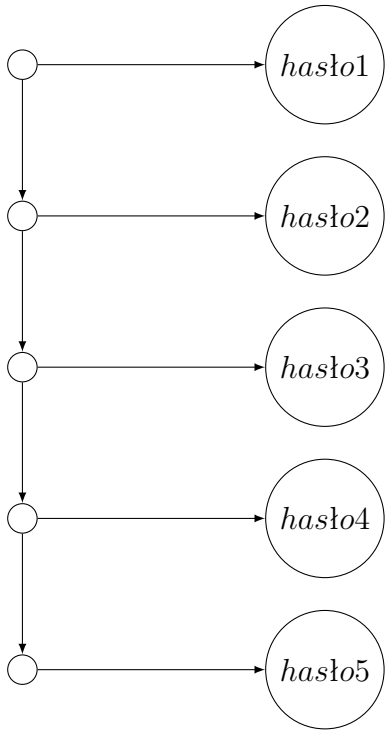
Zagadnienie przedstawia problem zasymulowania gry "Koło fortuny".

2.1 Struktury danych

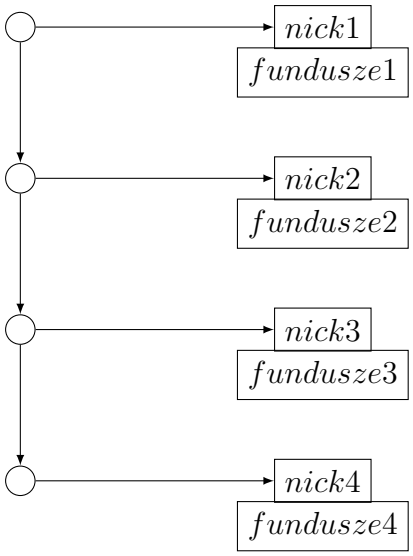
W programie wykorzystano dynamicznie alokowane tablice. Tablica haseł zawiera wszystkie hasła wczytane z pliku. Rysunek 1 przedstawia przykład tablicy haseł. Następna tablica zawiera wylosowane hasło. Kolejna z tablic zawiera struktury z danymi graczy. Rysunek 2 przedstawia przykład tablicy, zawierającej struktury z danymi graczy. Ostatnia tablica zawiera struktury z wynikami. Jest ona zbudowana w ten sam sposób co tablica graczy. Taka struktura danych umożliwia szybki dostęp do danych.

2.2 Algorytmy

Program porównuje wybraną literę ze wszystkimi literami hasła. Jeśli dana litera wystąpi w haśle jest ona odkrywana, a graczowi przyznawana jest nagroda. Niezależnie od tego czy litera wystąpi w haśle, jest ona ustawiana jako niedostępna. Gra toczy się do momentu odgadnięcia hasła lub odkrycia wszystkich liter hasła. Czas wykoniania programu zależy od ilości słów w danym pliku, długości tych słów, długości wylosowanego hasła, a także liczby prób odgadnięcia litery lub hasła podjętych przez gracza przed odgadnięciem hasła. [1].[2].



Rysunek 1: Przykład dynamicznie alkowanej tablicy, przechowującej wszystkie hasła wpisane z pliku



Rysunek 2: Przykład dynamicznie alkowanej tablicy, przechowującej struktury, zawierające dane graczy

3 Specyfikacja zewnętrzna

Program jest uruchamiany z linii poleceń. Należy przekazać do programu liczbę graczy (`argv[1]`),np.:

```
program 1
program 3
```

Liczbę graczy podaje sie cyfrą. Uruchomienie programu z parametrem `h`

```
program h
```

powoduje wyświetlenie krótkiej pomocy. Uruchomienie programu bez żadnego parametru lub z nieprawidłowymi parametrami powoduje wyświetlenie komunikatu:

```
Niepoprawna liczba graczy.
```

.

4 Specyfikacja wewnętrzna

Program został zrealizowany zgodnie z paradygmatem strukturalnym. W programie rozdzielono interfejs (komunikację z użytkownikiem) od logiki aplikacji (np.: sprawdzanie, czy dana litera wystąpiła w haśle).

4.1 Ogólna struktura programu

W funkcji głównej wywołana jest funkcja change_category. Funkcja ta uzmożliwnia wybór kategorii.

Następnie wywoływana jest funkcja run_state . Funkcja ta umożliwia przechodzenie między kolejnymi stanami podczas wykonywania programu. Funkcja ta przyjmuje jako parametry aktualny stan oraz strukturę, zawierającą najważniejsze dane programu oraz zwraca stan następny.

4.2 Szczegółowy opis typów i funkcji

Szczegółowy opis typów i funkcji zawarty jest w załączniku.

5 Testowanie

Program został przetestowany na różnego rodzaju plikach. Pliki wejściowe, zawierające hasła z danej kategorii, nie mogą zawierać błędów. Plik z wynikami nie musi istnieć, ale w takim przypadku nie jest możliwe odczytanie wyników. Mimo to możliwe jest zapisanie wyniku do pliku i tym samym stworzenie go.

Program został sprawdzony pod kątem wycieków pamięci.

6 Wnioski

Program, symulujący grę "Koło fortuny", jest programem prostym, chociaż wymaga samodzielnego zarządzania pamięcią. Najbardziej wymagające okazało się usunięcie wycieków pamięci. Szczególnie trudne było zapewnienie prawidłowego zwolnienia zaalokowanej pamięci na struktury graczy, w przypadku zmiany ich ilości podczas trwania programu. Przygotowanie tego projektu pozwoliło mi lepiej zrozumieć sposób działania wskaźników.

Literatura

[1] Adam Drozdek. C++. algorytmy i struktury danych. 2001.

[2] Jerzy Grębosz. *Symfonia C++*. Oficyna Kallimach, Kraków, 1995.

Dodatek
Szczegółowy opis typów i funkcji

Koło fortuny

Wygenerowano przez Doxygen 1.8.14

Spis treści

Rozdział 1

Indeks struktur danych

1.1 Struktury danych

Tutaj znajdują się struktury danych wraz z ich krótkimi opisami:

Rozdział 2

Indeks plików

2.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

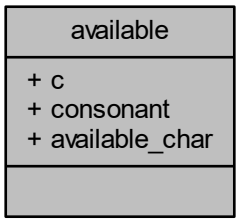
Rozdział 3

Dokumentacja struktur danych

3.1 Dokumentacja struktury available

`#include <Menu.h>`

Diagram współpracy dla available:



Pola danych

- char c
- bool consonant
- bool available_char

3.1.1 Opis szczegółowy

Struktura available - dostepna litera

Parametry

c	dana litera
consonant	true, gdy litera jest spolgloska, false - jest samogloska
available_char	true, gdy litera jest dostepna, false - nie jest dostepna

3.1.2 Dokumentacja pól

3.1.2.1 available_char

```
bool available_char
```

3.1.2.2 c

```
char c
```

3.1.2.3 consonant

```
bool consonant
```

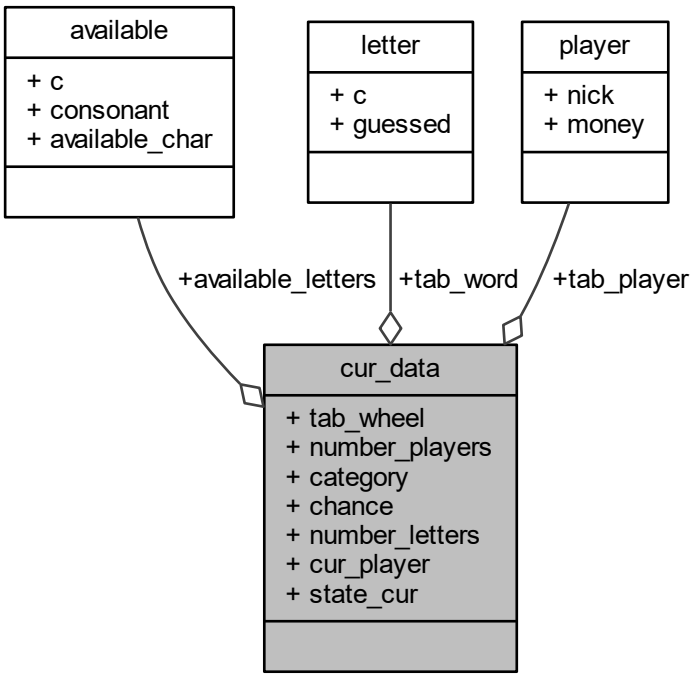
Dokumentacja dla tej struktury została wygenerowana z pliku:

- Menu.h

3.2 Dokumentacja struktury cur_data

```
#include <Menu.h>
```

Diagram współpracy dla cur_data:



Pola danych

- int tab_wheel [25]
- letter * tab_word
- int number_players
- char * category
- int chance
- int number_letters
- available available_letters [number_letters_alphabet]
- player * tab_player
- int cur_player
- int state_cur

3.2.1 Opis szczegółowy

Struktura cur_data - glowne dane

Parametry

tab_wheel	statyczna tablica kola
tab_word	dynamiczna tablica z wylosowanym haslem
number_players	liczba graczy
category	kategoria
chance	numer wylosowanej nagrody z kola
number_letters	liczba litere hasla
available_letters	statyczna tablica dostepnych liter
tab_player	dynamiczna tablica z danymi graczy
cur_player	numer aktualnego gracza
state_cur	aktualny stan

3.2.2 Dokumentacja pól

3.2.2.1 available_letters

```
available available_letters[number_letters_alphabet]
```

3.2.2.2 category

```
char* category
```

3.2.2.3 chance

```
int chance
```

3.2.2.4 cur_player

```
int cur_player
```

3.2.2.5 number_letters

```
int number_letters
```

3.2.2.6 number_players

```
int number_players
```

3.2.2.7 state_cur

```
int state_cur
```

3.2.2.8 tab_player

```
player* tab_player
```

3.2.2.9 tab_wheel

```
int tab_wheel[25]
```

3.2.2.10 tab_word

```
letter* tab_word
```

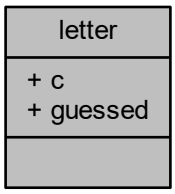
Dokumentacja dla tej struktury została wygenerowana z pliku:

- Menu.h

3.3 Dokumentacja struktury letter

```
#include <Menu.h>
```

Diagram współpracy dla letter:



Pola danych

- char c
- bool guessed

3.3.1 Opis szczegółowy

Struktura letter - litera hasla

Parametry

c	dana litera
guessed	true, gdy litera zostala juz odgadnieta, false - nie zostala odgadnieta

3.3.2 Dokumentacja pól

3.3.2.1 c

```
char c
```

3.3.2.2 guessed

```
bool guessed
```

Dokumentacja dla tej struktury została wygenerowana z pliku:

- Menu.h

3.4 Dokumentacja struktury player

```
#include <Menu.h>
```

Diagram współpracy dla player:

Pola danych

- char * nick
- int money

3.4.1 Opis szczegółowy

Struktura player - gracz

Parametry

nick	nick gracza
money	fundusze gracza

3.4.2 Dokumentacja pól

3.4.2.1 money

```
int money
```

3.4.2.2 nick

```
char* nick
```

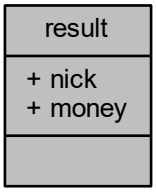
Dokumentacja dla tej struktury została wygenerowana z pliku:

- Menu.h

3.5 Dokumentacja struktury result

```
#include <Results.h>
```

Diagram współpracy dla result:



Pola danych

- char * nick
- int money

3.5.1 Opis szczegółowy

Struktura result - wynik

Parametry

nick	nick gracza
money	fundusze gracza

3.5.2 Dokumentacja pól

3.5.2.1 money

```
int money
```

3.5.2.2 nick

```
char* nick
```

Dokumentacja dla tej struktury została wygenerowana z pliku:

- Results.h

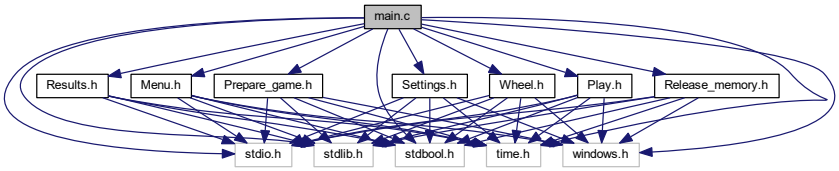
Rozdział 4

Dokumentacja plików

4.1 Dokumentacja pliku main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdbool.h>
#include <windows.h>
#include "Menu.h"
#include "Play.h"
#include "Prepare_game.h"
#include "Results.h"
#include "Settings.h"
#include "Wheel.h"
#include "Release_memory.h"
```

Wykres zależności załączania dla main.c:



Funkcje

- state run_state (state state_cur, cur_data ∗data)
- int main (int argc, char ∗∗argv)

Zmienne

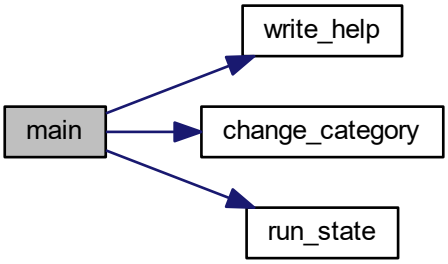
- state(∗ tab [NUMBER])(cur_data ∗data)

4.1.1 Dokumentacja funkcji

4.1.1.1 main()

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

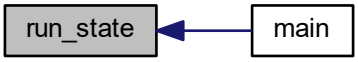
Oto graf wywołań dla tej funkcji:



4.1.1.2 run_state()

```
state run_state (
            state state_cur,
            cur_data * data )
```

Oto graf wywoływań tej funkcji:



4.1.2 Dokumentacja zmiennych

4.1.2.1 tab

```
state(* tab[NUMBER])(cur_data *data)
```

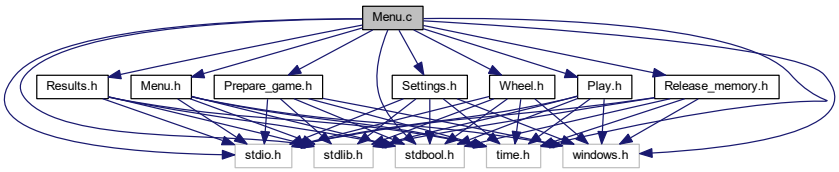
Wartość początkowa:

```
= {release_players, do_prepare_game, do_settings,
      do_results, do_guess_letter,
do_game_menu, do_spin,do_guess_word, do_buy,
      do_win, do_save, do_wheel, do_init, do_menu, NULL}
```

4.2 Dokumentacja pliku Menu.c

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdbool.h>
#include <windows.h>
#include "Menu.h"
#include "Play.h"
#include "Prepare_game.h"
#include "Results.h"
#include "Settings.h"
#include "Wheel.h"
#include "Release_memory.h"
```

Wykres zależności załączania dla Menu.c:



Funkcje

- state do_menu (cur_data *data)
- int write_available_consonant (cur_data data)
- void write_money (cur_data data)
- int write_menu (cur_data data)
- state do_game_menu (cur_data *data)
- void do_write_tab (cur_data data)
- void hit_enter ()
- int count_char (char *tmp)
- void write_help ()

4.2.1 Dokumentacja funkcji

4.2.1.1 count_char()

```
int count_char (
            char * tmp )
```

Funkcja liczy liczbe liter danego slowa

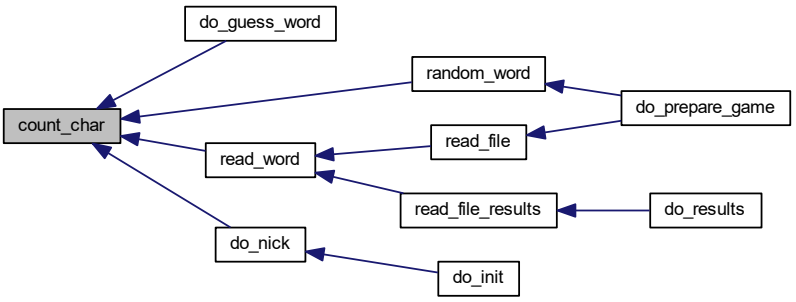
Parametry

tmp	dane slowo

Zwraca

Funckja zwraca liczbe liter danego slowa

Oto graf wywoływań tej funkcji:



4.2.1.2 do_game_menu()

```
state do_game_menu (
            cur_data * data )
```

Funkcja wypisuje menu gry

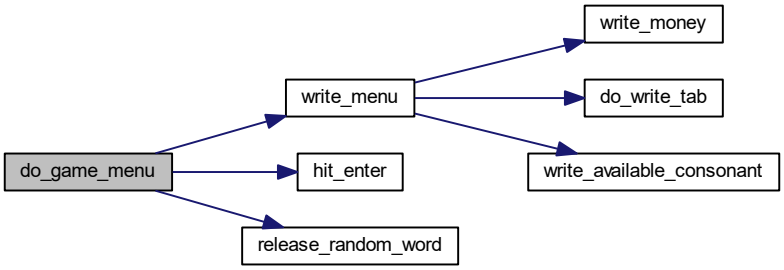
Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

Oto graf wywołań dla tej funkcji:



4.2.1.3 do_menu()

```
state do_menu (
            cur_data * data )
```

Funkcja wypisuje glowne menu

Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

Oto graf wywołań dla tej funkcji:

4.2.1.4 do_write_tab()

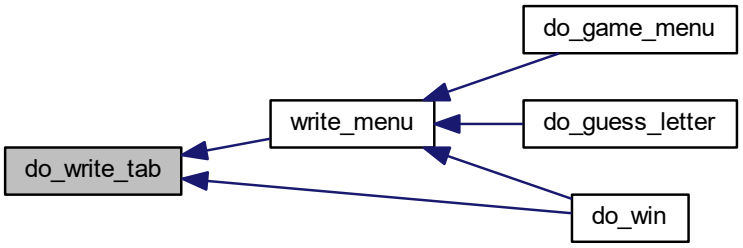
```
void do_write_tab (
            cur_data data )
```

Funkcja wypisuje tabele z losowym slowem

Parametry

data	struktura z glownymi danymi

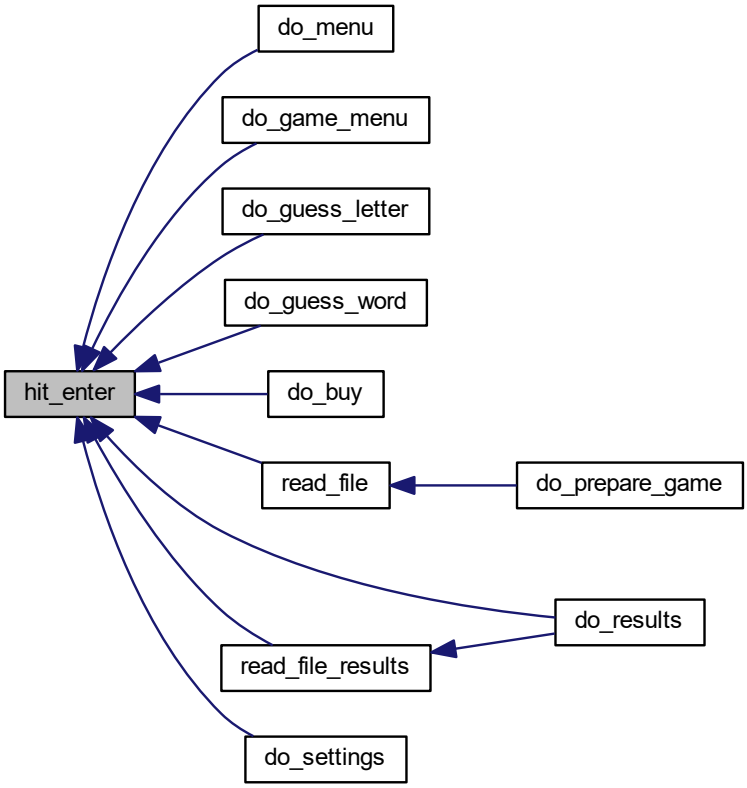
Oto graf wywoływań tej funkcji:



4.2.1.5 hit_enter()

```
void hit_enter ( )
```

Funkcja wypisuje komunikat i czeka na podanie entera Oto graf wywoływań tej funkcji:



4.2.1.6 write_available_consonant()

```
int write_available_consonant (
            cur_data data )
```

Funkcja wypisuje dostepne spolgloski

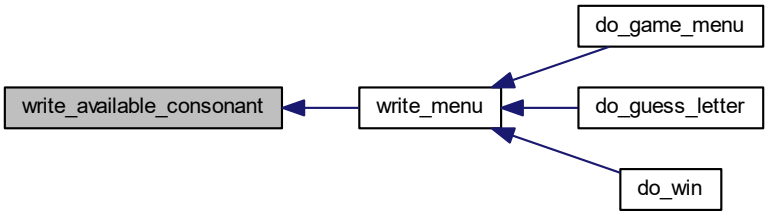
Parametry

data	struktura z glownymi danymi

Zwraca

Funckja zwraca liczbe wystapien danej litery w hasle

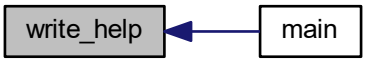
Oto graf wywoływań tej funkcji:



4.2.1.7 write_help()

```
void write_help ( )
```

Funkcja wypisuje help Oto graf wywoływań tej funkcji:



4.2.1.8 write_menu()

```
int write_menu (
            cur_data data )
```

Funkcja wypisuje tekst menu gry

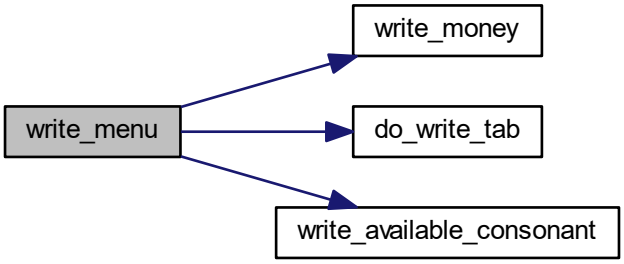
Parametry

data	struktura z glownymi danymi

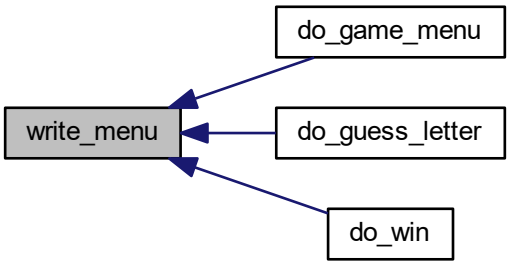
Zwraca

Funckja zwraca liczbe wystapien danej litery w hasle

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.2.1.9 write_money()

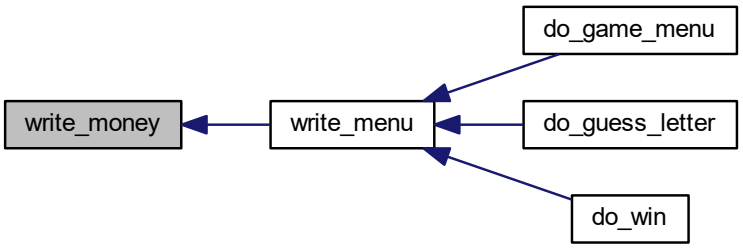
```
void write_money (
            cur_data data )
```

Funkcja wypisuje dostepne fundusze

Parametry

data	struktura z glownymi danymi

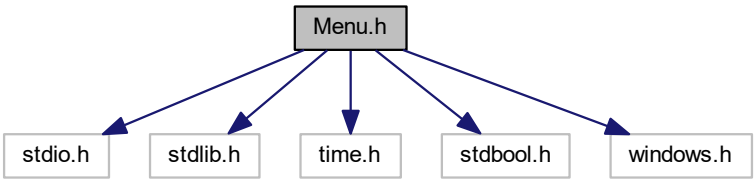
Oto graf wywoływań tej funkcji:



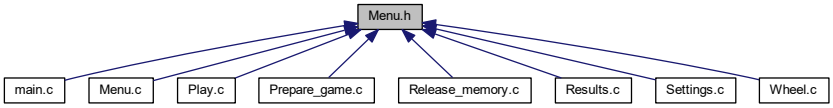
4.3 Dokumentacja pliku Menu.h

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdbool.h>
#include <windows.h>
```

Wykres zależności załączania dla Menu.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Struktury danych

- struct letter
- struct available
- struct player
- struct cur_data

Wyliczenia

- enum state {
  TURN_OFF, PREPARE_GAME, SETTINGS, RESULTS,
  GUESS_LETTER, GAME_MENU, SPIN, GUESS_WORD,
  BUY, WIN, SAVE, WHEEL,
  INIT, MENU, STOP, NUMBER }
- enum constant { space =32, enter =13, max_size =100, number_letters_alphabet =26 }

Funkcje

- state do_menu (cur_data *data)
- state do_game_menu (cur_data *data)
- void do_write_tab (cur_data data)
- int write_available_consonant (cur_data data)
- void write_money (cur_data data)
- int write_menu (cur_data data)
- void hit_enter ()
- int count_char (char *tmp)
- void write_help ()

4.3.1 Dokumentacja typów wyliczanych

4.3.1.1 constant

`enum constant`

Wartości wyliczeń

space	
enter	
max_size	
number_letters_alphabet	

4.3.1.2 state

`enum state`

Wartości wyliczeń

TURN_OFF	
PREPARE_GAME	
SETTINGS	
RESULTS	

Wartości wyliczeń

GUESS_LETTER	
GAME_MENU	
SPIN	
GUESS_WORD	
BUY	
WIN	
SAVE	
WHEEL	
INIT	
MENU	
STOP	
NUMBER	

4.3.2 Dokumentacja funkcji

4.3.2.1 count_char()

```
int count_char (
            char * tmp )
```

Funkcja liczy liczbe liter danego slowa

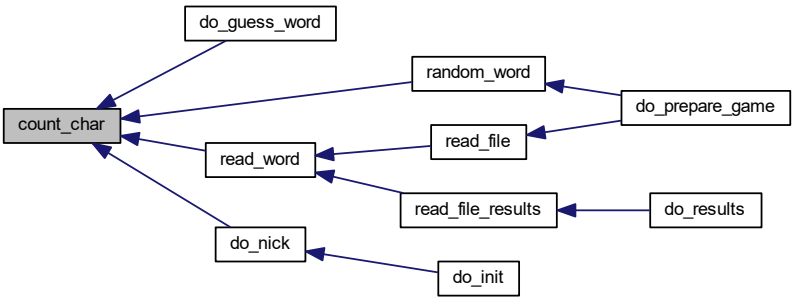
Parametry

tmp	dane slowo

Zwraca

Funckja zwraca liczbe liter danego slowa

Oto graf wywoływań tej funkcji:



4.3.2.2 do_game_menu()

```
state do_game_menu (
            cur_data * data )
```

Funkcja wypisuje menu gry

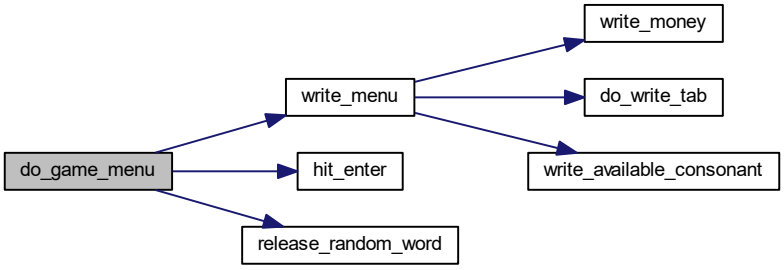
Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

Oto graf wywołań dla tej funkcji:



4.3.2.3 do_menu()

```
state do_menu (
            cur_data * data )
```

Funkcja wypisuje glowne menu

Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

Oto graf wywołań dla tej funkcji:

4.3.2.4 do_write_tab()

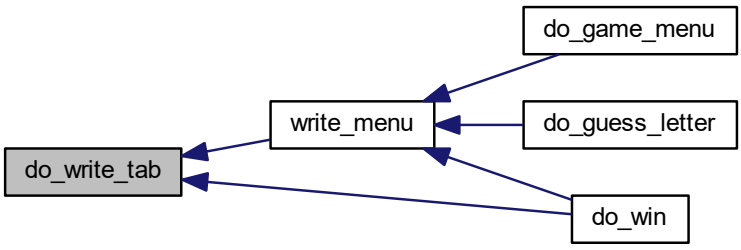
```
void do_write_tab (
            cur_data data )
```

Funkcja wypisuje tabele z losowym slowem

Parametry

data	struktura z glownymi danymi

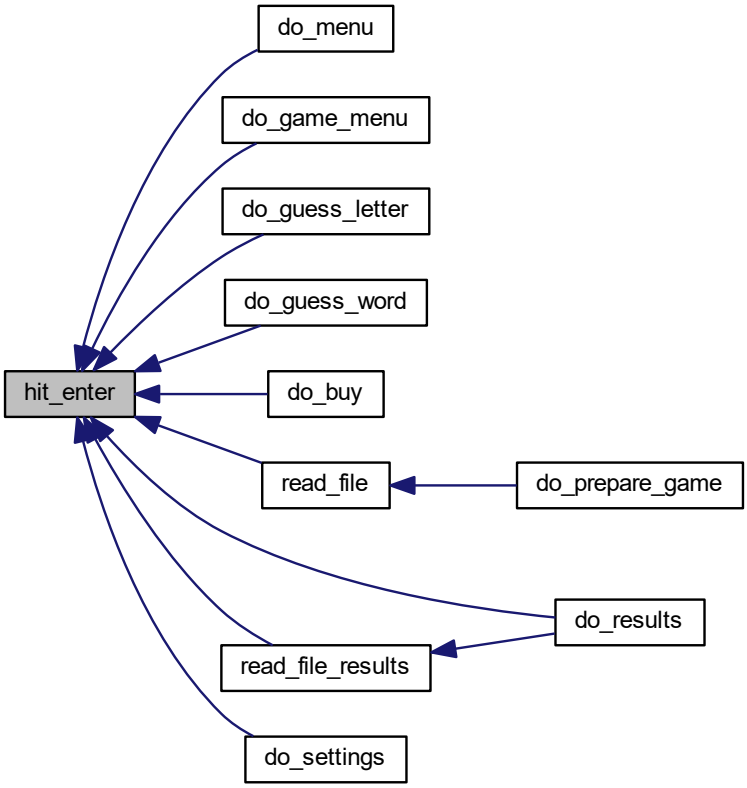
Oto graf wywoływań tej funkcji:



4.3.2.5 hit_enter()

```
void hit_enter ( )
```

Funkcja wypisuje komunikat i czeka na podanie entera Oto graf wywoływań tej funkcji:



4.3.2.6 write_available_consonant()

```
int write_available_consonant (
            cur_data data )
```

Funkcja wypisuje dostepne spolgloski

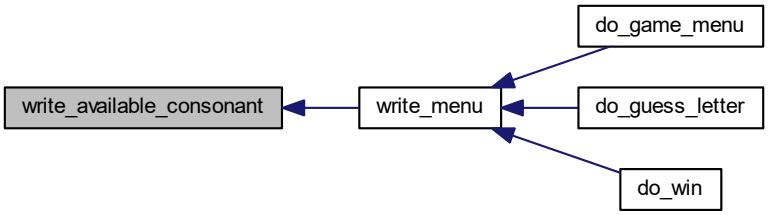
Parametry

data	struktura z glownymi danymi

Zwraca

Funckja zwraca liczbe wystapien danej litery w hasle

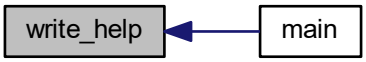
Oto graf wywoływań tej funkcji:



4.3.2.7 write_help()

```
void write_help ( )
```

Funkcja wypisuje help Oto graf wywoływań tej funkcji:



4.3.2.8 write_menu()

```
int write_menu (
            cur_data data )
```

Funkcja wypisuje tekst menu gry

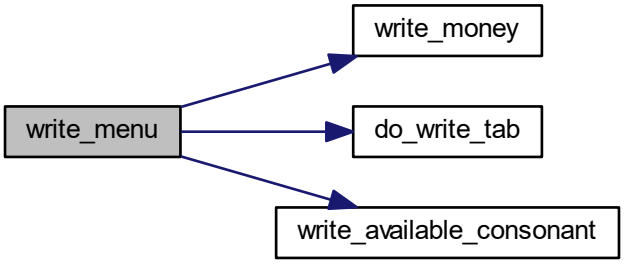
Parametry

data	struktura z glownymi danymi

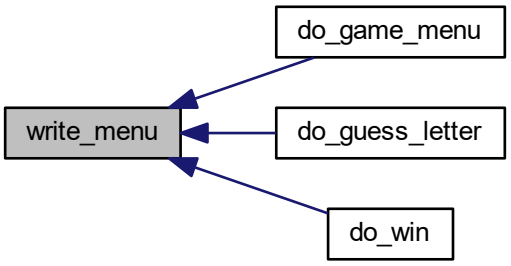
Zwraca

Funckja zwraca liczbe wystapien danej litery w hasle

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.3.2.9 write_money()

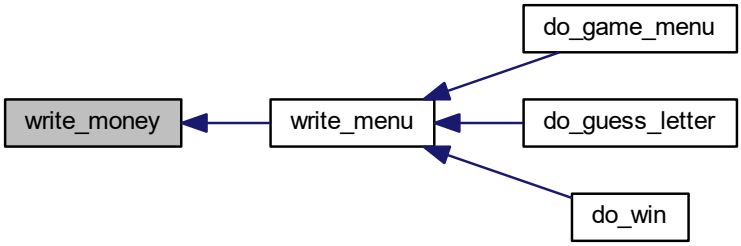
```
void write_money (
            cur_data data )
```

Funkcja wypisuje dostepne fundusze

Parametry

data	struktura z glownymi danymi

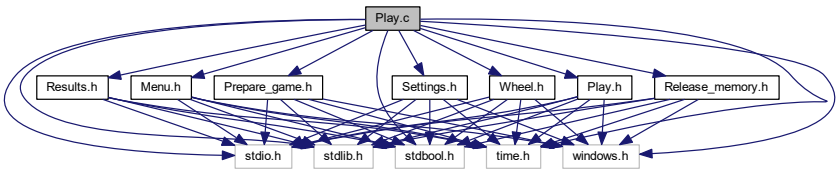
Oto graf wywoływań tej funkcji:



4.4 Dokumentacja pliku Play.c

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdbool.h>
#include <windows.h>
#include "Menu.h"
#include "Play.h"
#include "Prepare_game.h"
#include "Results.h"
#include "Settings.h"
#include "Wheel.h"
#include "Release_memory.h"
```

Wykres zależności załączania dla Play.c:



Funkcje

- int check_letter (cur_data *data, char letter_c)
- state do_guess_letter (cur_data *data)
- state do_guess_word (cur_data *data)
- state do_buy (cur_data *data)
- state do_win (cur_data *data)

4.4.1 Dokumentacja funkcji

4.4.1.1 check_letter()

```
int check_letter (
            cur_data * data,
            char letter_c )
```

Funkcja sprawdza czy podana litera wystepuje w hasle

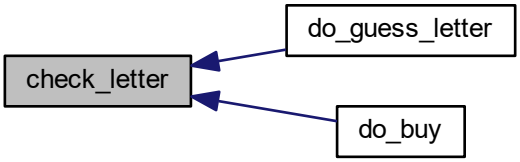
Parametry

data	struktura z glownymi danymi
letter_c	wybrana litera

Zwraca

Funkcja zwraca liczbe wystapien danej litery w hasle

Oto graf wywoływań tej funkcji:



4.4.1.2 do_buy()

```
state do_buy (
            cur_data * data )
```

Funkcja umozliwia kupienie samogloski

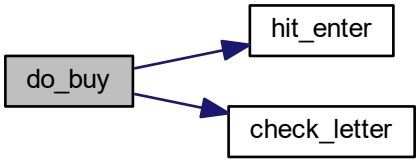
Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

Oto graf wywołań dla tej funkcji:



4.4.1.3 do_guess_letter()

```
state do_guess_letter (
            cur_data * data )
```

Funkcja umozliwia wybranie litery

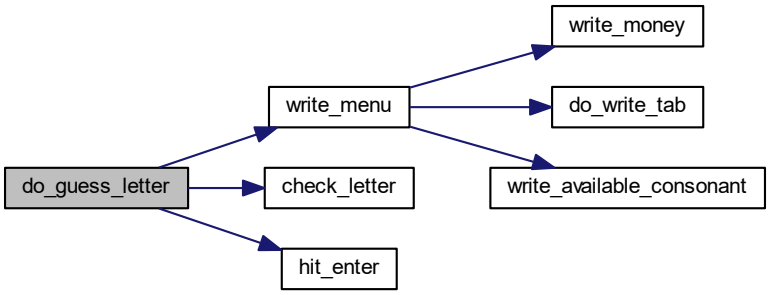
Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

Oto graf wywołań dla tej funkcji:



4.4.1.4 do_guess_word()

```
state do_guess_word (
            cur_data * data )
```

Funkcja umozliwia odgadniecie hasla

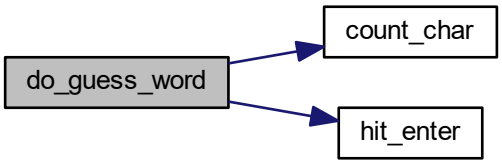
Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

Oto graf wywołań dla tej funkcji:



4.4.1.5 do_win()

```
state do_win (
            cur_data * data )
```

Funkcja obsluguje stan po wygranej

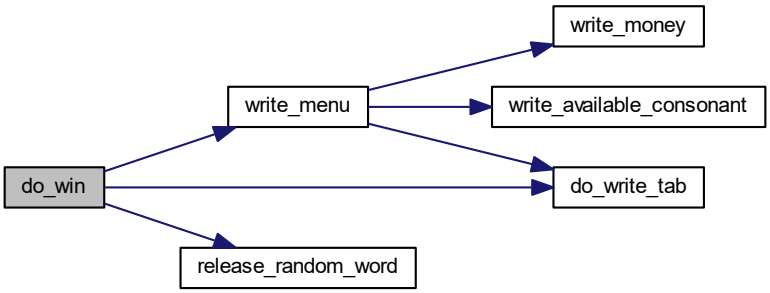
Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

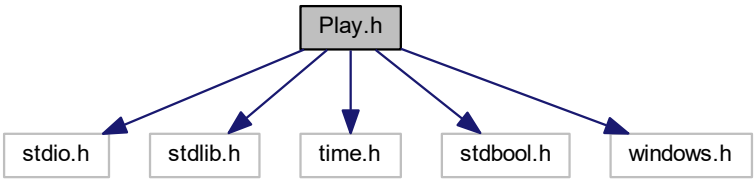
Oto graf wywołań dla tej funkcji:



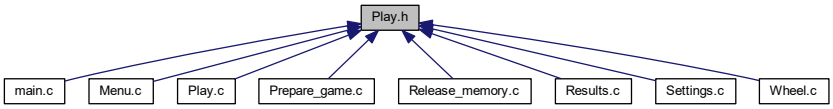
4.5 Dokumentacja pliku Play.h

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdbool.h>
#include <windows.h>
```

Wykres zależności załączania dla Play.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Funkcje

- state do_guess_letter (cur_data *data)

- int check_letter (cur_data ∗data, char letter_c)
- state do_guess_word (cur_data ∗data)
- state do_buy (cur_data ∗data)
- state do_win (cur_data ∗data)

4.5.1 Dokumentacja funkcji

4.5.1.1 check_letter()

```
int check_letter (
            cur_data * data,
            char letter_c )
```

Funkcja sprawdza czy podana litera wystepuje w hasle

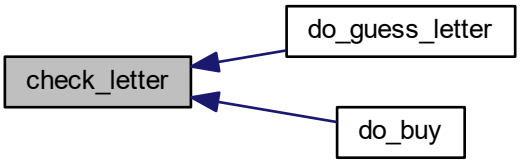
Parametry

data	struktura z glownymi danymi
letter_c	wybrana litera

Zwraca

Funkcja zwraca liczbe wystapien danej litery w hasle

Oto graf wywoływań tej funkcji:



4.5.1.2 do_buy()

```
state do_buy (
            cur_data * data )
```

Funkcja umozliwia kupienie samogloski

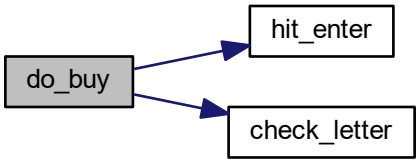
Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

Oto graf wywołań dla tej funkcji:



4.5.1.3 do_guess_letter()

```
state do_guess_letter (
            cur_data * data )
```

Funkcja umozliwia wybranie litery

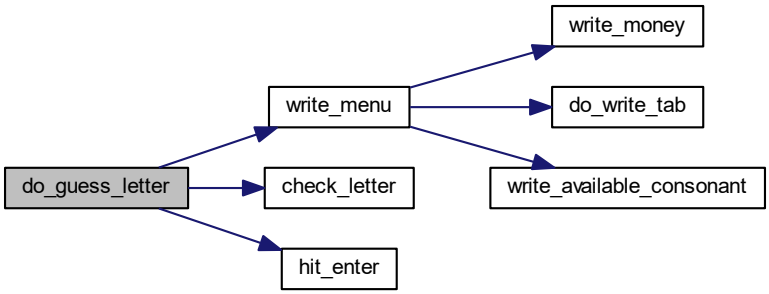
Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

Oto graf wywołań dla tej funkcji:



4.5.1.4 do_guess_word()

```
state do_guess_word (
            cur_data * data )
```

Funkcja umozliwia odgadniecie hasla

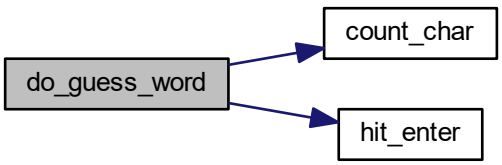
Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

Oto graf wywołań dla tej funkcji:



4.5.1.5 do_win()

```
state do_win (
            cur_data * data )
```

Funkcja obsluguje stan po wygranej

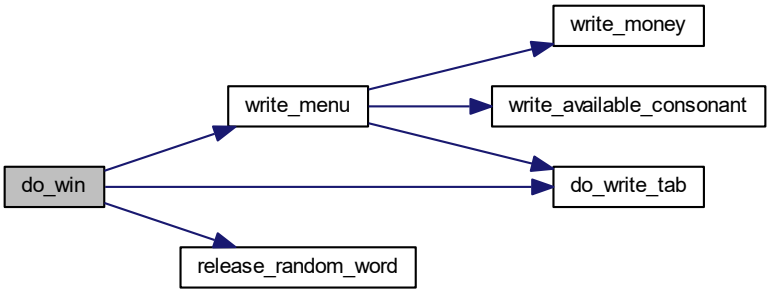
Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

Oto graf wywołań dla tej funkcji:

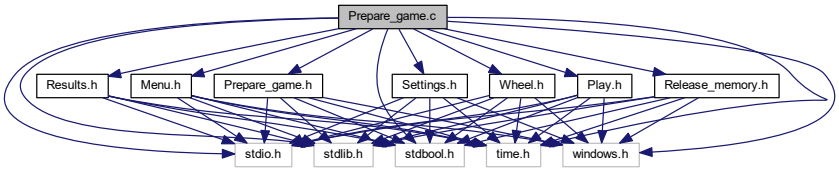


4.6 Dokumentacja pliku Prepare_game.c

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdbool.h>
#include <windows.h>
#include "Menu.h"
#include "Play.h"
#include "Prepare_game.h"
#include "Results.h"
#include "Settings.h"
#include "Wheel.h"
```

```
#include "Release_memory.h"
```

Wykres zależności załączania dla Prepare_game.c:



Funkcje

- state do_prepare_game (cur_data ∗data)
- void do_available_letters (cur_data ∗data)
- void random_word (cur_data ∗data, char ∗∗tab_words, int j)
- char ∗ read_word (FILE ∗file)
- char ∗∗ read_file (cur_data data, int ∗j)
- state do_init (cur_data ∗data)
- char ∗ do_nick ()

4.6.1 Dokumentacja funkcji

4.6.1.1 do_available_letters()

```
void do_available_letters (
            cur_data * data )
```

Funkcja przygotowuje tablice dostepnych liter

Parametry

data	struktura z glownymi danymi

Oto graf wywoływań tej funkcji:

4.6.1.2 do_init()

```
state do_init (
            cur_data * data )
```

Funkcja tworzy tablice graczy

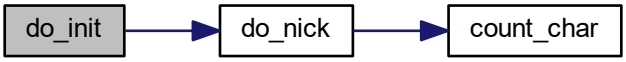
Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

Oto graf wywołań dla tej funkcji:



4.6.1.3 do_nick()

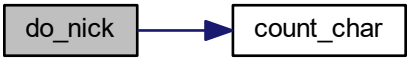
```
char* do_nick ( )
```

Funkcja wpisuje nick do dynamicznej tablicy

Zwraca

Funkcja zwraca wpisany nick

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

4.6.1.4 do_prepare_game()

```
state do_prepare_game (
            cur_data * data )
```

Funkcja przygotowuje gre

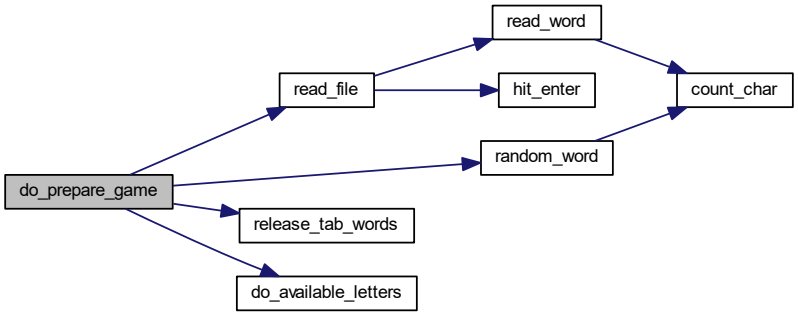
Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

Oto graf wywołań dla tej funkcji:



4.6.1.5 random_word()

```
void random_word (
            cur_data * data,
            char ** tab_words,
            int j )
```

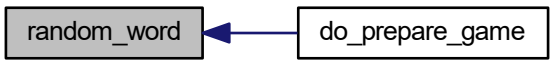
Funkcja losuje haslo

Parametry

data	struktura z glownymi danymi
tab_words	tablica slow z pliku liczba slow z pliku

Oto graf wywołań dla tej funkcji:

Oto graf wywoływań tej funkcji:



4.6.1.6 read_file()

```
char** read_file (
            cur_data data,
            int * j )
```

Funkcja czyta z pliku wszystkie slowa

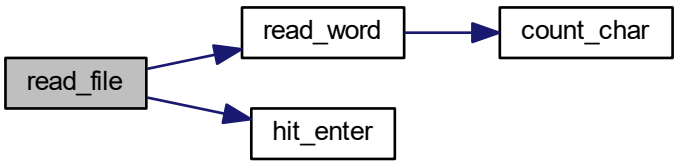
Parametry

data	struktura z glownymi danymi
j	liczba slow w pliku

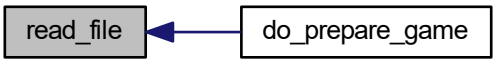
Zwraca

Funkcja zwraca tablice wszystkich slow z pliku

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.6.1.7 read_word()

```
char* read_word (
            FILE * file )
```

Funkcja czyta z pliku slowa i wpisuje je do dynamicznej tablicy

Parametry

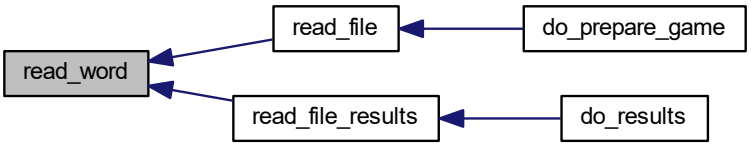
file	plik wejsciowy

Zwraca

Funkcja zwraca wczytane slowo

Oto graf wywołań dla tej funkcji:

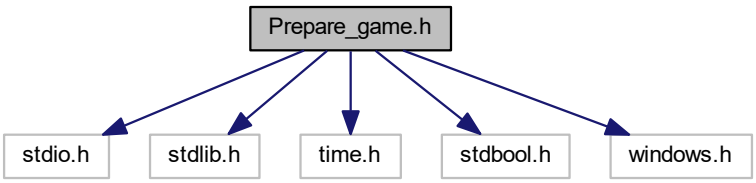
Oto graf wywoływań tej funkcji:



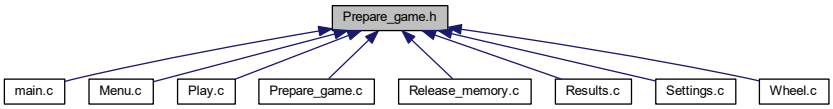
4.7 Dokumentacja pliku Prepare_game.h

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdbool.h>
#include <windows.h>
```

Wykres zależności załączania dla Prepare_game.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Definicje

- #define preapre_game_H

Funkcje

- state do_prepare_game (cur_data ∗data)
- void random_word (cur_data ∗data, char ∗∗tab_words, int j)
- char ∗ read_word (FILE ∗file)
- char ∗∗ read_file (cur_data data, int ∗j)
- void do_available_letters (cur_data ∗data)
- state do_init (cur_data ∗data)
- char ∗ do_nick ()

4.7.1 Dokumentacja definicji

4.7.1.1 preapre_game_H

```
#define preapre_game_H
```

4.7.2 Dokumentacja funkcji

4.7.2.1 do_available_letters()

```
void do_available_letters (
            cur_data * data )
```

Funkcja przygotowuje tablice dostepnych liter

Parametry

data	struktura z glownymi danymi

Oto graf wywoływań tej funkcji:

4.7.2.2 do_init()

```
state do_init (
            cur_data * data )
```

Funkcja tworzy tablice graczy

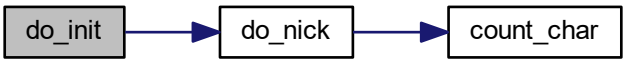
Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

Oto graf wywołań dla tej funkcji:



4.7.2.3 do_nick()

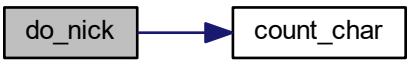
```
char* do_nick ( )
```

Funkcja wpisuje nick do dynamicznej tablicy

Zwraca

Funkcja zwraca wpisany nick

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

4.7.2.4 do_prepare_game()

```
state do_prepare_game (
            cur_data * data )
```

Funkcja przygotowuje gre

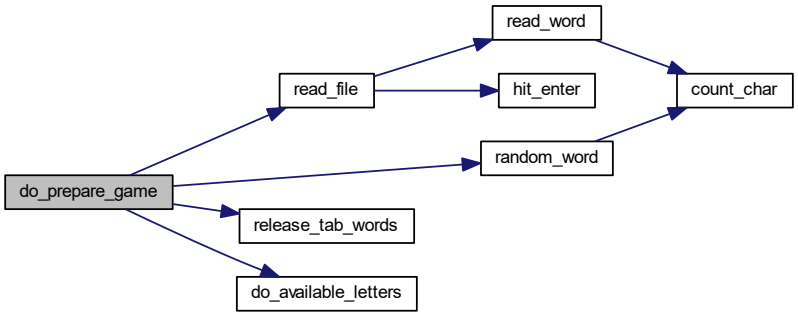
Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

Oto graf wywołań dla tej funkcji:



4.7.2.5 random_word()

```
void random_word (
            cur_data * data,
            char ** tab_words,
            int j )
```

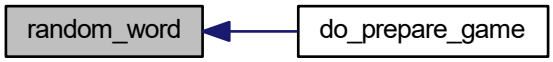
Funkcja losuje haslo

Parametry

data	struktura z glownymi danymi
tab_words	tablica slow z pliku liczba slow z pliku

Oto graf wywołań dla tej funkcji:

Oto graf wywoływań tej funkcji:



4.7.2.6 read_file()

```
char** read_file (
            cur_data data,
            int * j )
```

Funkcja czyta z pliku wszystkie slowa

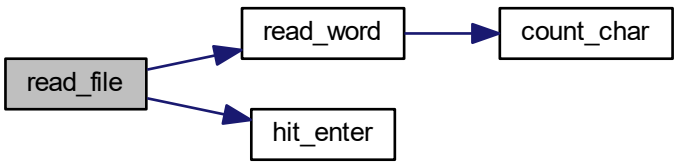
Parametry

data	struktura z glownymi danymi
j	liczba slow w pliku

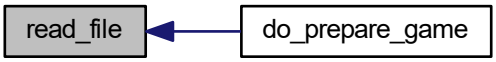
Zwraca

Funkcja zwraca tablice wszystkich slow z pliku

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.7.2.7 read_word()

```
char* read_word (
            FILE * file )
```

Funkcja czyta z pliku slowa i wpisuje je do dynamicznej tablicy

Parametry

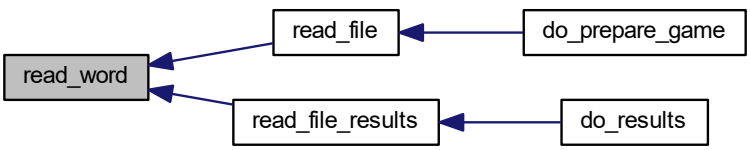
file	plik wejsciowy

Zwraca

Funkcja zwraca wczytane slowo

Oto graf wywołań dla tej funkcji:

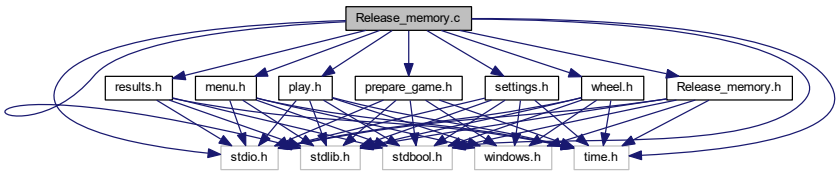
Oto graf wywoływań tej funkcji:



4.8 Dokumentacja pliku Release_memory.c

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdbool.h>
#include "menu.h"
#include "play.h"
#include "prepare_game.h"
#include "results.h"
#include "settings.h"
#include "wheel.h"
#include "Release_memory.h"
```

Wykres zależności załączania dla Release_memory.c:



Funkcje

- void release_results (result **tab, int j)
- state release_players (cur_data *data)
- void release_tab_words (char ***tab, int j)
- void release_random_word (cur_data *data)

4.8.1 Dokumentacja funkcji

4.8.1.1 release_players()

```
state release_players (
            cur_data * data )
```

Funkcja zwalnia pamiec zaalokowana na graczy

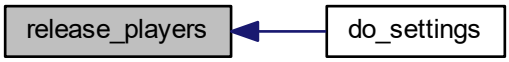
Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

Oto graf wywoływań tej funkcji:



4.8.1.2 release_random_word()

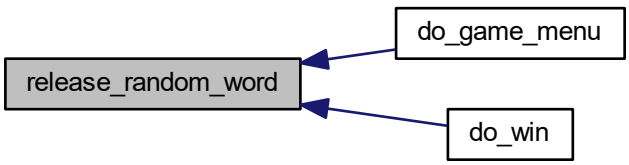
```
void release_random_word (
            cur_data * data )
```

Funkcja zwalnia pamiec zaalokowana na haslo

Parametry

data	struktura z glownymi danymi

Oto graf wywoływań tej funkcji:



4.8.1.3 release_results()

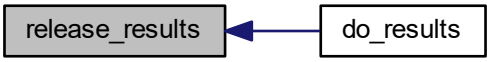
```
void release_results (
            result ** tab,
            int j )
```

Funkcja zwalnia pamiec zaalokowana na wyniki

Parametry

tab	tablica wynikow
j	liczba wynikow

Oto graf wywoływań tej funkcji:



4.8.1.4 release_tab_words()

```
void release_tab_words (
            char *** tab,
            int j )
```

Funkcja zwalnia pamiec zaalokowana na slowa z pliku

Parametry

tab	tablica slow z pliku

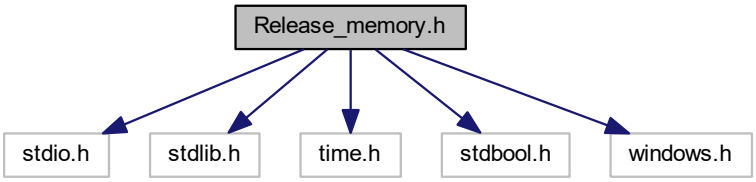
Oto graf wywoływań tej funkcji:

4.9 Dokumentacja pliku Release_memory.h

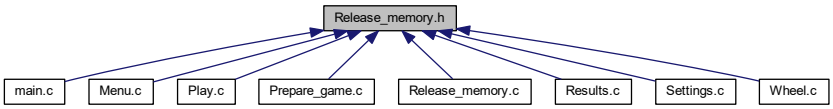
```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdbool.h>
```

```
#include <windows.h>
```

Wykres zależności załączania dla Release_memory.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Funkcje

- void release_results (result **tab, int j)
- state release_players (cur_data *data)
- void release_tab_words (char ***tab, int j)
- void release_random_word (cur_data *data)

4.9.1 Dokumentacja funkcji

4.9.1.1 release_players()

```
state release_players (
            cur_data * data )
```

Funkcja zwalnia pamiec zaalokowana na graczy

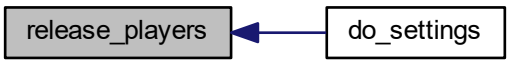
Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

Oto graf wywoływań tej funkcji:



4.9.1.2 release_random_word()

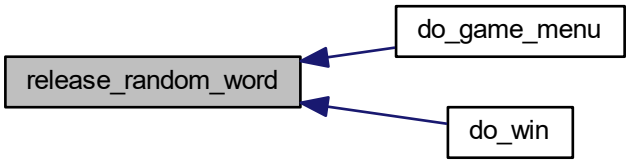
```
void release_random_word (
            cur_data * data )
```

Funkcja zwalnia pamiec zaalokowana na haslo

Parametry

data	struktura z glownymi danymi

Oto graf wywoływań tej funkcji:



4.9.1.3 release_results()

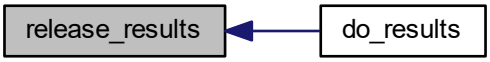
```
void release_results (
            result ** tab,
            int j )
```

Funkcja zwalnia pamiec zaalokowana na wyniki

Parametry

tab	tablica wynikow
j	liczba wynikow

Oto graf wywoływań tej funkcji:



4.9.1.4 release_tab_words()

```
void release_tab_words (
             char *** tab,
             int j )
```

Funkcja zwalnia pamiec zaalokowana na slowa z pliku

Parametry

tab	tablica slow z pliku

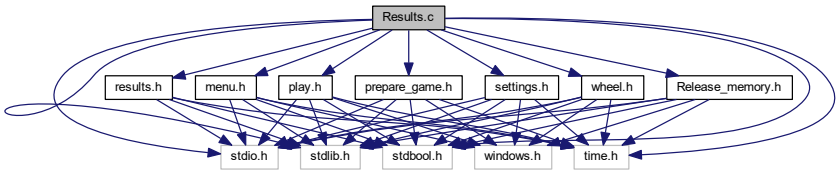
Oto graf wywoływań tej funkcji:

4.10 Dokumentacja pliku Results.c

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdbool.h>
```

```
#include "menu.h"
#include "play.h"
#include "prepare_game.h"
#include "results.h"
#include "settings.h"
#include "wheel.h"
#include "Release_memory.h"
```

Wykres zależności załączania dla Results.c:



Funkcje

- int sort_nick (const void ∗a, const void ∗b)
- int sort_money (const void ∗a, const void ∗b)
- state do_results (cur_data ∗data)
- state do_save (cur_data ∗data)
- result ∗ read_file_results (int ∗j)

4.10.1 Dokumentacja funkcji

4.10.1.1 do_results()

```
state do_results (
            cur_data * data )
```

Funkcja obsluguje stan wyniki

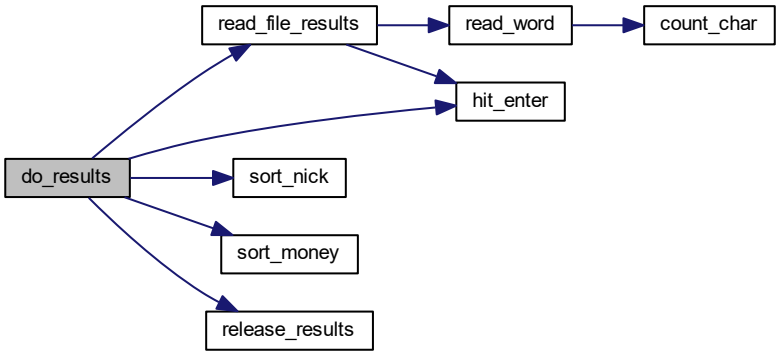
Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

Oto graf wywołań dla tej funkcji:



4.10.1.2 do_save()

```
state do_save (
            cur_data * data )
```

Funkcja zapisuje wynik

Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

4.10.1.3 read_file_results()

```
result* read_file_results (
            int * j )
```

Funkcja wpisuje do tablicy wyniki z pliku

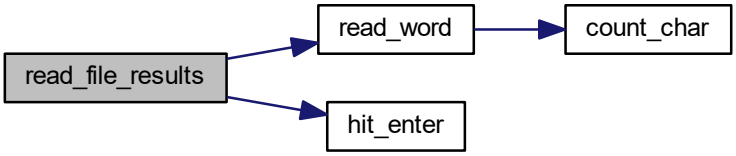
Parametry

j	liczba wynikow

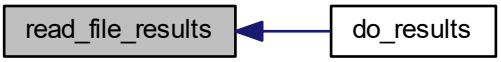
Zwraca

Funkcja zwraca tablice wynikow

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.10.1.4 sort_money()

```
int sort_money (
            const void * a,
            const void * b )
```

Funkcja porownuje elementy tablicy po funduszach

Parametry

a	pierwszy element tablicy
b	kolejny element tablicy

Zwraca

Funkcja zwraca 1 - b>a, -1 - b<a, 0 - a=b

Oto graf wywoływań tej funkcji:

4.10.1.5 sort_nick()

```
int sort_nick (
            const void * a,
            const void * b )
```

Funkcja porownuje elementy tablicy po nicku

Parametry

a	pierwszy element tablicy
b	kolejny element tablicy

Zwraca

Funkcja zwraca 1 - a>b, -1 - a<b, 0 - a=b

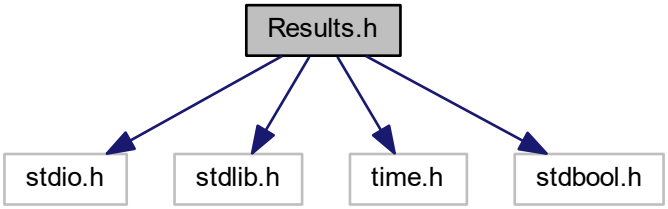
Oto graf wywoływań tej funkcji:

4.11 Dokumentacja pliku Results.h

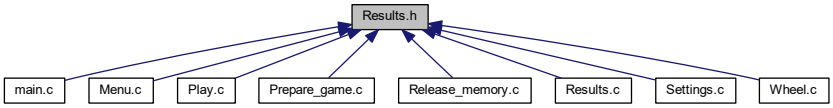
```
#include <stdio.h>
#include <stdlib.h>
```

```
#include <time.h>
#include <stdbool.h>
```

Wykres zależności załączania dla Results.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Struktury danych

- struct result

Funkcje

- state do_results (cur_data ∗data)
- state do_save (cur_data ∗data)
- result ∗ read_file_results (int ∗j)
- int sort_nick (const void ∗a, const void ∗b)
- int sort_money (const void ∗a, const void ∗b)

4.11.1 Dokumentacja funkcji

4.11.1.1 do_results()

```
state do_results (
            cur_data * data )
```

Funkcja obsługuje stan wyniki

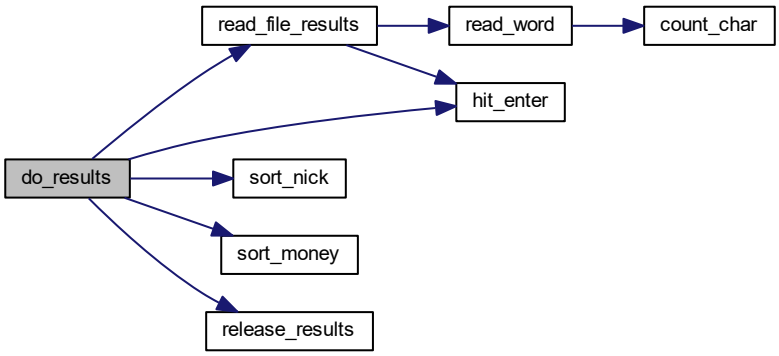
Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

Oto graf wywołań dla tej funkcji:



4.11.1.2 do_save()

```
state do_save (
            cur_data * data )
```

Funkcja zapisuje wynik

Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

4.11.1.3 read_file_results()

```
result* read_file_results (
            int * j )
```

Funkcja wpisuje do tablicy wyniki z pliku

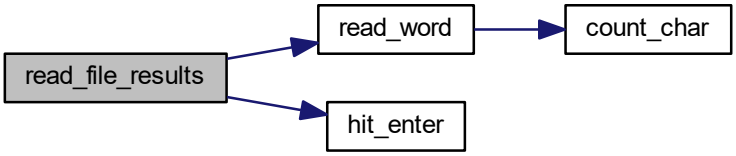
Parametry

j	liczba wynikow

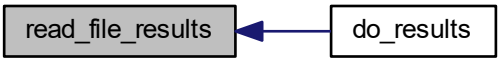
Zwraca

Funkcja zwraca tablice wynikow

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.11.1.4 sort_money()

```
int sort_money (
            const void * a,
            const void * b )
```

Funkcja porownuje elementy tablicy po funduszach

Parametry

a	pierwszy element tablicy
b	kolejny element tablicy

Zwraca

Funkcja zwraca 1 - b>a, -1 - b<a, 0 - a=b

Oto graf wywoływań tej funkcji:

4.11.1.5 sort_nick()

```
int sort_nick (
            const void * a,
            const void * b )
```

Funkcja porownuje elementy tablicy po nicku

Parametry

a	pierwszy element tablicy
b	kolejny element tablicy

Zwraca

Funkcja zwraca 1 - a>b, -1 - a<b, 0 - a=b

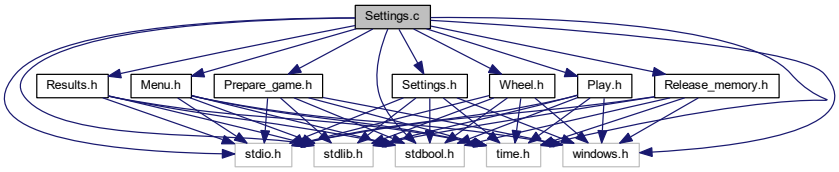
Oto graf wywoływań tej funkcji:

4.12 Dokumentacja pliku Settings.c

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

```
#include <time.h>
#include <stdbool.h>
#include <windows.h>
#include "Menu.h"
#include "Play.h"
#include "Prepare_game.h"
#include "Results.h"
#include "Settings.h"
#include "Wheel.h"
#include "Release_memory.h"
```

Wykres zależności załączania dla Settings.c:



Funkcje

- state do_settings (cur_data *data)
- void change_category (char **category)
- void change_number_players (int *number)

4.12.1 Dokumentacja funkcji

4.12.1.1 change_category()

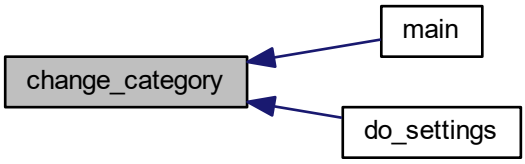
```
void change_category (
            char ** category )
```

Funkcja umozliwia zmiane kategorii

Parametry

category	nazwa kategorii

Oto graf wywoływań tej funkcji:



4.12.1.2 change_number_players()

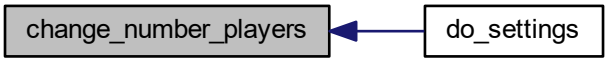
```
void change_number_players (
            int * number )
```

Funkcja umozliwia zmiane ilosci graczy

Parametry

number	liczba graczy

Oto graf wywoływań tej funkcji:



4.12.1.3 do_settings()

```
state do_settings (
            cur_data * data )
```

Funkcja obsluguje stan ustawienia

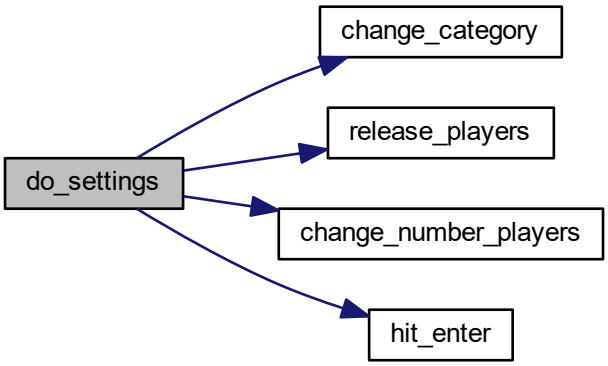
Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

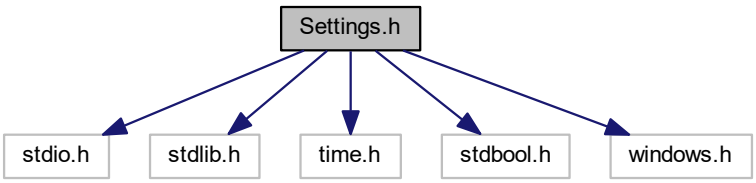
Oto graf wywołań dla tej funkcji:



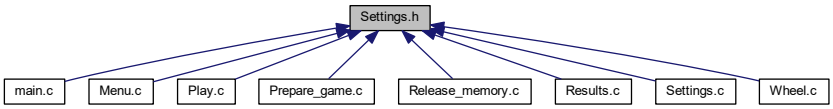
4.13 Dokumentacja pliku Settings.h

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdbool.h>
#include <windows.h>
```

Wykres zależności załączania dla Settings.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Funkcje

- state do_settings (cur_data ∗data)
- void change_category (char ∗∗category)
- void change_number_players (int ∗number)

4.13.1 Dokumentacja funkcji

4.13.1.1 change_category()

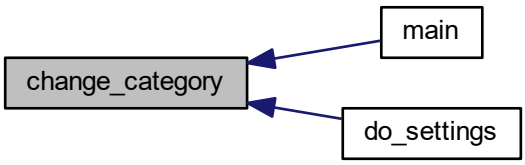
```
void change_category (
            char ** category )
```

Funkcja umozliwia zmiane kategorii

Parametry

category	nazwa kategorii

Oto graf wywoływań tej funkcji:



4.13.1.2 change_number_players()

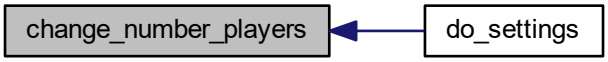
```
void change_number_players (
            int * number )
```

Funkcja umozliwia zmiane ilosci graczy

Parametry

number	liczba graczy

Oto graf wywoływań tej funkcji:



4.13.1.3 do_settings()

```
state do_settings (
            cur_data * data )
```

Funkcja obsluguje stan ustawienia

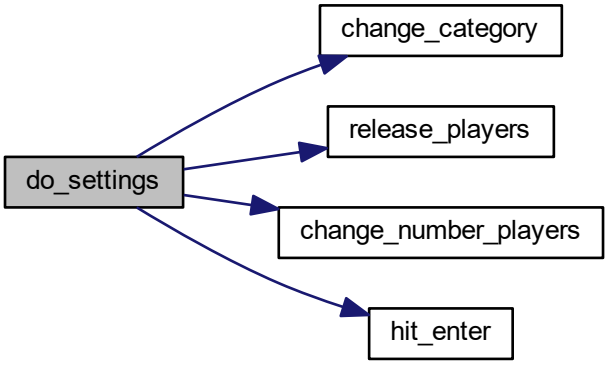
Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

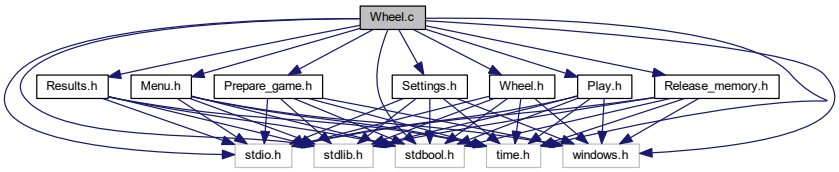
Oto graf wywołań dla tej funkcji:



4.14 Dokumentacja pliku Wheel.c

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdbool.h>
#include <windows.h>
#include "Menu.h"
#include "Play.h"
#include "Prepare_game.h"
#include "Results.h"
#include "Settings.h"
#include "Wheel.h"
#include "Release_memory.h"
```

Wykres zależności załączania dla Wheel.c:



Funkcje

- state do_wheel (cur_data *data)
- int power ()
- state do_spin (cur_data *data)

4.14.1 Dokumentacja funkcji

4.14.1.1 do_spin()

```
state do_spin (
            cur_data * data )
```

Funkcja obsluguje krecenie kolem

Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

Oto graf wywołań dla tej funkcji:

4.14.1.2 do_wheel()

```
state do_wheel (
            cur_data * data )
```

Funkcja czyta z pliku wartosci na kole i wpisuje je do tablicy

Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

4.14.1.3 power()

```
int power ( )
```

Funkcja wyznacza mocy krecenia kolem

Zwraca

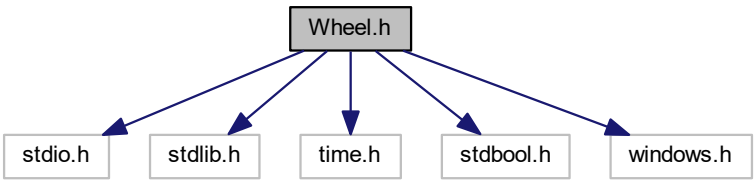
Funkcja zwraca liczbe oznaczajaca moc krecenia kolem

Oto graf wywoływań tej funkcji:

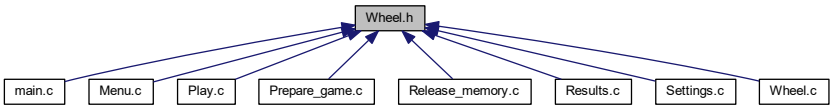
4.15 Dokumentacja pliku Wheel.h

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdbool.h>
#include <windows.h>
```

Wykres zależności załączania dla Wheel.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Funkcje

- state do_wheel (cur_data ∗data)
- state do_spin (cur_data ∗data)
- int power ()

4.15.1 Dokumentacja funkcji

4.15.1.1 do_spin()

```
state do_spin (
            cur_data * data )
```

Funkcja obsluguje krecenie kolem

Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

Oto graf wywołań dla tej funkcji:

4.15.1.2 do_wheel()

```
state do_wheel (
            cur_data * data )
```

Funkcja czyta z pliku wartosci na kole i wpisuje je do tablicy

Parametry

data	struktura z glownymi danymi

Zwraca

Funkcja zwraca nastepny stan

4.15.1.3 power()

```
int power ( )
```

Funkcja wyznacza mocy krecenia kolem

Zwraca

Funkcja zwraca liczbe oznaczajaca moc krecenia kolem

Oto graf wywoływań tej funkcji: