

Statki Ultimate

Konrad Sikorski

Wygenerowano przez Doxygen 1.8.4
Wt, 21 sty 2014 22:48:34

Contents

1 Indeks klas	1
1.1 Lista klas	1
2 Indeks plików	2
2.1 Lista plików	2
3 Dokumentacja klas	3
3.1 Dokumentacja struktury BoardSetter	3
3.1.1 Opis szczegółowy	4
3.2 Dokumentacja struktury BoardViewer	4
3.2.1 Opis szczegółowy	5
3.3 Dokumentacja struktury Brain	5
3.3.1 Opis szczegółowy	6
3.4 Dokumentacja struktury Logger	6
3.4.1 Opis szczegółowy	7
3.5 Dokumentacja struktury LogMove	7
3.5.1 Opis szczegółowy	7
3.6 Dokumentacja struktury Menu	7
3.6.1 Opis szczegółowy	8
3.7 Dokumentacja struktury NumGet	8
3.7.1 Opis szczegółowy	8
3.8 Dokumentacja struktury Plansza	9
3.8.1 Opis szczegółowy	9
3.8.2 Dokumentacja atrybutów składowych	9
3.9 Dokumentacja struktury Pole	10
3.9.1 Opis szczegółowy	10
3.10 Dokumentacja struktury Statek	10
3.10.1 Opis szczegółowy	10
3.11 Dokumentacja struktury StringGet	10
3.11.1 Opis szczegółowy	11
4 Dokumentacja plików	11
4.1 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/BoardSetter.c	11
4.1.1 Opis szczegółowy	12
4.1.2 Dokumentacja funkcji	13
4.2 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/BoardSetter.h	15
4.2.1 Opis szczegółowy	16
4.2.2 Dokumentacja funkcji	16
4.3 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/BoardViewer.c	18

4.3.1	Opis szczegółowy	19
4.4	Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/BoardViewer.h	20
4.4.1	Opis szczegółowy	21
4.5	Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Brain.c	21
4.5.1	Opis szczegółowy	22
4.6	Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Brain.h	22
4.6.1	Opis szczegółowy	23
4.7	Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Logger.c	23
4.7.1	Opis szczegółowy	24
4.7.2	Dokumentacja funkcji	24
4.8	Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Logger.h	24
4.8.1	Opis szczegółowy	26
4.8.2	Dokumentacja funkcji	26
4.9	Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/main.c	26
4.9.1	Opis szczegółowy	31
4.9.2	Dokumentacja funkcji	31
4.10	Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Menu.c	36
4.10.1	Opis szczegółowy	36
4.10.2	Dokumentacja funkcji	36
4.11	Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Menu.h	40
4.11.1	Opis szczegółowy	41
4.11.2	Dokumentacja funkcji	41
4.12	Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/NumGet.c	45
4.12.1	Opis szczegółowy	46
4.13	Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/NumGet.h	46
4.13.1	Opis szczegółowy	48
4.14	Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/ResHan.c	48
4.14.1	Opis szczegółowy	49
4.14.2	Dokumentacja funkcji	49
4.15	Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/ResHan.h	50
4.15.1	Opis szczegółowy	51
4.15.2	Dokumentacja funkcji	51
4.16	Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Screen.c	52
4.16.1	Opis szczegółowy	53
4.16.2	Dokumentacja funkcji	53
4.17	Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Screen.h	54
4.17.1	Opis szczegółowy	55
4.17.2	Dokumentacja funkcji	55
4.18	Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Statki.c	56
4.18.1	Opis szczegółowy	57

4.18.2 Dokumentacja funkcji	57
4.19 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Statki.h	59
4.19.1 Opis szczegółowy	61
4.19.2 Dokumentacja funkcji	61
4.20 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/StringGet.c	63
4.20.1 Opis szczegółowy	64
4.20.2 Dokumentacja funkcji	64
4.21 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/StringGet.h	65
4.21.1 Opis szczegółowy	66
4.21.2 Dokumentacja funkcji	66
4.22 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Util.c	67
4.22.1 Opis szczegółowy	67
4.23 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Util.h	67
4.23.1 Opis szczegółowy	68

Indeks

69

1 Indeks klas

1.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

BoardSetter Struktura reprezentująca ustawiacz statków	3
BoardViewer Struktura reprezentująca "wyświetlacz" planszy	4
Brain Struktura reprezentująca jednostkę sztucznej inteligencji	5
Logger Struktura reprezentująca moduł logujący przebieg rozgrywki	6
LogMove Struktura reprezentująca pojedynczy wpis do modułu logującego przebieg rozgrywki	7
Menu Struktura reprezentująca menu	7
NumGet Struktura reprezentująca pobieracz rozmiaru	8
Plansza Struktura reprezentująca planszę	9
Pole Struktura reprezentująca pojedyncze pole (współrzędne na planszy)	10

Statek

Struktura reprezentująca statek (używana podczas ustawiania planszy)	10
--	----

StringGet

Struktura reprezentująca pobieracz łańcucha znakowego	10
---	----

2 Indeks plików

2.1 Lista plików

Tutaj znajduje się lista wszystkich udokumentowanych plików z ich krótkimi opisami:

/home/konrad/Pulpit/pri/Statki/ BoardSetter.c	11
Plik źródłowy ustawiacza statków	
/home/konrad/Pulpit/pri/Statki/ BoardSetter.h	15
Plik nagłówkowy ustawiacza statków	
/home/konrad/Pulpit/pri/Statki/ BoardViewer.c	18
Plik źródłowy obsługi wyświetlania planszy	
/home/konrad/Pulpit/pri/Statki/ BoardViewer.h	20
Plik nagłówkowy obsługi wyświetlania planszy	
/home/konrad/Pulpit/pri/Statki/ Brain.c	21
Plik źródłowy obsługi sztucznej inteligencji	
/home/konrad/Pulpit/pri/Statki/ Brain.h	22
Plik nagłówkowy obsługi sztucznej inteligencji	
/home/konrad/Pulpit/pri/Statki/ Logger.c	23
Plik źródłowy obsługi modułu zapisującego przebieg rozgrywki	
/home/konrad/Pulpit/pri/Statki/ Logger.h	24
Plik nagłówkowy obsługi modułu zapisującego przebieg rozgrywki	
/home/konrad/Pulpit/pri/Statki/ main.c	26
Plik główny	
/home/konrad/Pulpit/pri/Statki/ Menu.c	36
Plik źródłowy obsługi menu	
/home/konrad/Pulpit/pri/Statki/ Menu.h	40
Plik nagłówkowy obsługi menu	
/home/konrad/Pulpit/pri/Statki/ NumGet.c	45
Plik źródłowy obsługi pobieracza rozmiaru	
/home/konrad/Pulpit/pri/Statki/ NumGet.h	46
Plik nagłówkowy obsługi pobieracza rozmiaru planszy	
/home/konrad/Pulpit/pri/Statki/ ResHan.c	48
Plik źródłowy funkcji przerw (zmiana rozmiaru terminala)	
/home/konrad/Pulpit/pri/Statki/ ResHan.h	50
Plik nagłówkowy funkcji przerw (zmiana rozmiaru terminala)	
/home/konrad/Pulpit/pri/Statki/ Screen.c	52
Plik źródłowy obsługi ekranu	

/home/konrad/Pulpit/pri/Statki/Screen.h	
Plik nagłówkowy obsługi ekranu	54
/home/konrad/Pulpit/pri/Statki/Statki.c	
Plik źródłowy obsługi planszy	56
/home/konrad/Pulpit/pri/Statki/Statki.h	
Plik nagłówkowy obsługi planszy	59
/home/konrad/Pulpit/pri/Statki/StringGet.c	
Plik źródłowy obsługi pobieracza łańcucha znakowego	63
/home/konrad/Pulpit/pri/Statki/StringGet.h	
Plik nagłówkowy obsługi pobieracza łańcucha znakowego	65
/home/konrad/Pulpit/pri/Statki/Util.c	
Plik źródłowy funkcji pomocniczych	67
/home/konrad/Pulpit/pri/Statki/Util.h	
Plik nagłówkowy funkcji pomocniczych	67

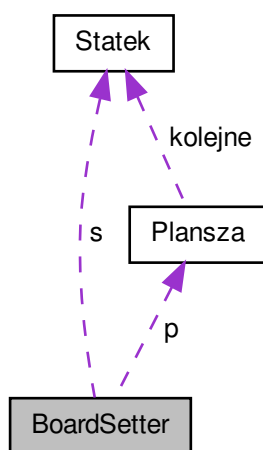
3 Dokumentacja klas

3.1 Dokumentacja struktury BoardSetter

Struktura reprezentująca ustawiacz statków.

```
#include <BoardSetter.h>
```

Diagram współpracy dla BoardSetter:



Atrybuty publiczne

- WINDOW * **win**
Wskaźnik na okno, w którym ma być rysowany ustawiacz (ncurses)

- **Plansza** * **p**
Wskaźnik na ustawianą planszę
- **Statek** **s** [10]
Tablica kolejno ustawianych statków (od najdłuższego do najkrótszego)
- int **akt**
*Indeks w tablicy **s** aktualnie ustawianego statku.*
- int **vpl**
- int **vpc**

3.1.1 Opis szczegółowy

Struktura reprezentująca ustawiacz statków.

Dokumentacja dla tej struktury została wygenerowana z pliku:

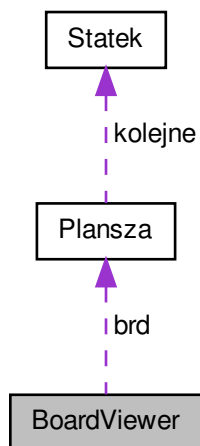
- `/home/konrad/Pulpit/pri/Statki/BoardSetter.h`

3.2 Dokumentacja struktury BoardViewer

Struktura reprezentująca "wyświetlacz" planszy.

```
#include <BoardViewer.h>
```

Diagram współpracy dla BoardViewer:



Atrybuty publiczne

- int **cur**
Wartość logiczna, oznaczająca widoczność kursora wyboru pola na planszy.
- int **cy**
Liczba, oznaczająca linię aktualnie wyświetlanego obszaru (względna), w której znajduje się pole wskazywane przez kursor.
- int **cx**

Liczba, oznaczająca kolumnę aktualnie wyświetlanego obszaru (względna), w której znajduje się pole wskazywane przez kursor.

- **MODE mode**

Zmienna, oznaczająca tryb wyświetlania planszy.

- **PAN pan**

Zmienna, oznaczająca stronę ekranu, po której ma być narysowany wyświetlacz.

- **Plansza * brd**

Wskaźnik na rysowaną planszę

- **WINDOW * win**

Wskaźnik na okno, w którym ma być rysowany wyświetlacz (ncurses)

- **int vpl**

Współrzędna oznaczająca linię lewego górnego rogu aktualnie wyświetlanego obszaru (viewport)

- **int vpc**

Współrzędna oznaczająca kolumnę lewego górnego rogu aktualnie wyświetlanego obszaru (viewport)

3.2.1 Opis szczegółowy

Struktura reprezentująca "wyświetlacz" planszy.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- `/home/konrad/Pulpit/pri/Statki/BoardViewer.h`

3.3 Dokumentacja struktury Brain

Struktura reprezentująca jednostkę sztucznej inteligencji.

```
#include <Brain.h>
```

Diagram współpracy dla Brain:



Atrybuty publiczne

- **Pole ok [20]**

Tablica z polami, na których znajdują się segmenty statków.

- **int oki**

Indeks tablicy ok, który wskazuje na pierwsze niewykorzystane z niej pole.

- Pole * `bad`

Tablica z polami, na których nie ma statków.

- int `badi`

Indeks tablicy `bad`, który wskazuje na pierwsze niewykorzystane z niej pole.

- int `prop`

Prawdopodobieństwo (w promilach) wybrania pola z segmentem statku.

- int `prevbrdsize`

Rozmiar poprzedniej planszy, dla której AI było inicjowane.

3.3.1 Opis szczegółowy

Struktura reprezentująca jednostkę sztucznej inteligencji.

Jej schemat działania jest następujący: przy inicjalizacji do tablicy `ok` wpisywane są pola planszy przeciwnika, na których znajdują się segmenty statków, a do tablicy `bad` pola, na których ich nie ma. Podczas rozgrywki AI podejmuje decyzję, czy wybrać pole ze statkiem, czy nie (z określonym prawdopodobieństwem: 10%/20%/40% odpowiednio dla łatwego/średniego/trudnego poziomu trudności), a następnie wybiera kolejne pole z odpowiedniej tablicy. Jeśli wszystkie pola w tablicy zostały wykorzystane (jest to mało prawdopodobne, ale jednak), wtedy wybierane jest pole z tablicy, w której jeszcze są pola.

Dokumentacja dla tej struktury została wygenerowana z pliku:

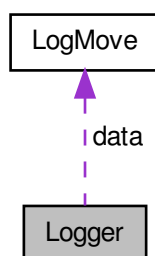
- `/home/konrad/Pulpit/pri/Statki/Brain.h`

3.4 Dokumentacja struktury Logger

Struktura reprezentująca moduł logujący przebieg rozgrywki.

```
#include <Logger.h>
```

Diagram współpracy dla Logger:



Atrybuty publiczne

- time_t `begin`

Czas rozpoczęcia rozgrywki.

- time_t `end`

Czas zakończenia rozgrywki.

- int `reason`

- Powód zakończenia rozgrywki.*
- `LogMove * data`
Tablica kolejnych ruchów.
- `int size`
*Rozmiar tablicy kolejnych ruchów (2*rozmiar*rozmiar)*
- `int wsk`
Liczba ruchów dotychczas już wykonanych.
- `int width`
Szerokość rysowanego okna.
- `int height`
Wysokość rysowanego okna, liczba ostatnich ruchów do wypisania.
- `WINDOW * win`
Wskaźnik na okno (z ncurses), którym logger ma być rysowany.

3.4.1 Opis szczegółowy

Struktura reprezentująca moduł logujący przebieg rozgrywki.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- `/home/konrad/Pulpit/pri/Statki/Logger.h`

3.5 Dokumentacja struktury LogMove

Struktura reprezentująca pojedynczy wpis do modułu logującego przebieg rozgrywki.

```
#include <Logger.h>
```

Atrybuty publiczne

- `int l`
Linia.
- `int c`
Kolumna.
- `int pid`
Id gracza.
- `int res`
Wynik ruchu (trafiony/pudło/zatopiony)

3.5.1 Opis szczegółowy

Struktura reprezentująca pojedynczy wpis do modułu logującego przebieg rozgrywki.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- `/home/konrad/Pulpit/pri/Statki/Logger.h`

3.6 Dokumentacja struktury Menu

Struktura reprezentująca menu.

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

Atrybuty publiczne

- WINDOW * [win](#)
Wskaźnik na okno (z ncurses), którym menu ma być rysowane.
- int [w](#)
Szerokość menu (w znakach)
- int [n](#)
Liczba elementów w menu.
- char [choices](#) [10][20]
Tablica napisów do wyświetlenia w menu.
- int [now](#)
Numer aktualnie wybranego menu.

3.6.1 Opis szczegółowy

Struktura reprezentująca menu.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- [/home/konrad/Pulpit/pri/Statki/Menu.h](#)

3.7 Dokumentacja struktury NumGet

Struktura reprezentująca pobieracz rozmiaru.

```
#include <NumGet.h>
```

Atrybuty publiczne

- WINDOW * [win](#)
Wskaźnik na okno (z ncurses), którym pobieracz ma być rysowany.
- int [w](#)
Szerokość pobieracza (w znakach)
- int [minn](#)
Dolne ograniczenie.
- int [maxx](#)
Górne ograniczenie.
- int [now](#)
Aktualna wartość

3.7.1 Opis szczegółowy

Struktura reprezentująca pobieracz rozmiaru.

Dokumentacja dla tej struktury została wygenerowana z pliku:

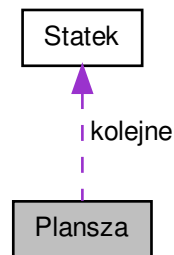
- [/home/konrad/Pulpit/pri/Statki/NumGet.h](#)

3.8 Dokumentacja struktury Plansza

Struktura reprezentująca planszę

```
#include <Statki.h>
```

Diagram współpracy dla Plansza:



Atrybuty publiczne

- [Statek kolejne](#) [10]
Tablica statków, w której przechowywane są kolejno wstawiane na planszę statki, aby móc cofnąć wstawiony już statek podczas ustawiania.
- [int ile](#)
Aktualna liczba statków na planszy (podczas wstawiania statków na planszę)
- [pole ** tab](#)
Tablica pól planszy (z dodatkowymi kolumnami granicznymi)
- [int shot](#)
Liczba aktualnie zestrzelonych segmentów statków (do sprawdzania, czy gra już się skończyła)
- [int sizebefore](#)
Poprzednio ustawiony rozmiar planszy.

3.8.1 Opis szczegółowy

Struktura reprezentująca planszę

3.8.2 Dokumentacja atrybutów składowych

3.8.2.1 `pole** Plansza::tab`

Tablica pól planszy (z dodatkowymi kolumnami granicznymi)

Zawartość prawdziwej planszy znajduje się od pola o indeksach (1,1) do (brdsize,brdsize) włącznie. Zostało to tak zaprojektowane, aby ułatwić sprawdzanie sąsiadujących do statka pól.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- `/home/konrad/Pulpit/pri/Statki/Statki.h`

3.9 Dokumentacja struktury Pole

Struktura reprezentująca pojedyncze pole (współrzędne na planszy)

```
#include <Brain.h>
```

Atrybuty publiczne

- int **x**
- int **y**

3.9.1 Opis szczegółowy

Struktura reprezentująca pojedyncze pole (współrzędne na planszy)

Dokumentacja dla tej struktury została wygenerowana z pliku:

- /home/konrad/Pulpit/pri/Statki/[Brain.h](#)

3.10 Dokumentacja struktury Statek

Struktura reprezentująca statek (używana podczas ustawiania planszy)

```
#include <Statki.h>
```

Atrybuty publiczne

- int [lin](#)
Linia w której znajduje się lewy/górny segment statku.
- int [col](#)
Kolumna w której znajduje się lewy/górny segment statku.
- [orient ori](#)
Orientacja statku (pionowa/pozioma)
- int [len](#)
Długość statku.

3.10.1 Opis szczegółowy

Struktura reprezentująca statek (używana podczas ustawiania planszy)

Dokumentacja dla tej struktury została wygenerowana z pliku:

- /home/konrad/Pulpit/pri/Statki/[Statki.h](#)

3.11 Dokumentacja struktury StringGet

Struktura reprezentująca pobieracz łańcucha znakowego.

```
#include <StringGet.h>
```

Atrybuty publiczne

- char * [data](#)
Bufor danych.
- int [maxlen](#)
Maksymalna długość przechowywanego łańcucha znakowego.
- int [akt](#)
Liczba aktualnie wczytanych znaków.
- WINDOW * [win](#)
Wskaźnik na okno (z ncurses), którym pobieracz ma być rysowany.
- int [w](#)
Szerokość pola do wprowadzania łańcucha.

3.11.1 Opis szczegółowy

Struktura reprezentująca pobieracz łańcucha znakowego.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- [/home/konrad/Pulpit/pri/Statki/StringGet.h](#)

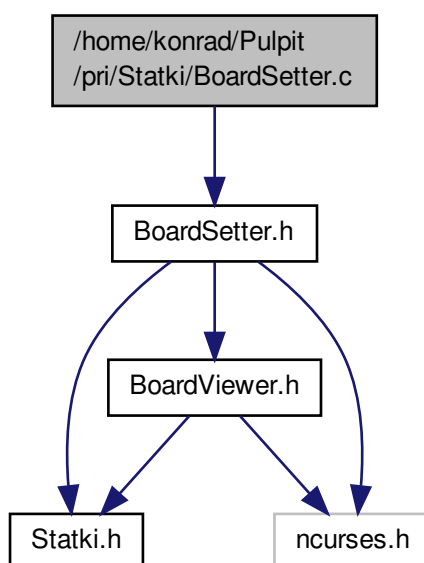
4 Dokumentacja plików

4.1 Dokumentacja pliku [/home/konrad/Pulpit/pri/Statki/BoardSetter.c](#)

Plik źródłowy ustawiacza statków.

```
#include "BoardSetter.h"
```

Wykres zależności załączania dla BoardSetter.c:



Wyliczenia

- enum {
 ST =0, **EM**, **NW**, **OKO**,
 KON }

Enumeracja używana w oznaczeniu stanu pola przy ustawianiu planszy (segment statku znajdujący się na planszy / puste pole / aktualnie wstawiany statek / jego okolica / konfliktowe pole - pole, które powoduje, że dane ustawienie jest nieprawidłowe)

Funkcje

- void **newbs** (**BoardSetter** *bs, **Plansza** *p)
Funkcja inicjująca nowy ustawiacz.
- int **nextstat** (**BoardSetter** *bs)
Funkcja akceptująca ustawienie aktualnie ustawianego statku.
- int **prevstat** (**BoardSetter** *bs)
Funkcja cofająca ustawienie ostatnio ustawionego statku.
- void **drawbs** (**BoardSetter** *bs)
Funkcja rysująca ustawiacz.
- void **updatebs** (**BoardSetter** *bs)
Funkcja uaktualniająca ustawiacz, aby nie tworzyć nowego okna za każdym razem, gdy użytkownik wciśnie klawisz.
- void **bsup** (**BoardSetter** *bs)
Funkcja przesuwająca aktualnie ustawiany statek w górę
- void **bsdown** (**BoardSetter** *bs)
Funkcja przesuwająca aktualnie ustawiany statek w dół
- void **bsleft** (**BoardSetter** *bs)
Funkcja przesuwająca aktualnie ustawiany statek w lewo.
- void **bsright** (**BoardSetter** *bs)
Funkcja przesuwająca aktualnie ustawiany statek w prawo.
- void **bsrot** (**BoardSetter** *bs)
Funkcja obracająca (jeśli się da) aktualnie ustawiany statek.

Zmienne

- const int **WIDTH**
Szerokość wyświetlanego pola.
- const int **HEIGHT**
Wysokość wyświetlanego pola.
- int **brdsize**
- int **brdt**
Wybrany typ planszy.

4.1.1 Opis szczegółowy

Plik źródłowy ustawiacza statków.

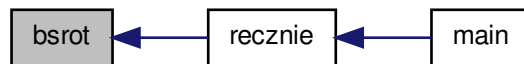
4.1.2 Dokumentacja funkcji

4.1.2.1 void bsrot (BoardSetter * bs)

Funkcja obracająca (jeśli się da) aktualnie ustawiany statek.

Statek jest obracany o 90 stopni względem jego skrajnie lewego/górnego segmentu. Jeśli obrót spowoduje wystawienie statku poza planszę, obrót nie jest wykonywany

Oto graf wywoływań tej funkcji:



4.1.2.2 void newbs (BoardSetter * bs, Plansza * p)

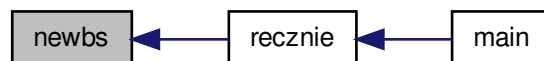
Funkcja inicjująca nowy ustawiacz.

Funkcja wypełnia tablicę kolejnych statków do wstawienia od najdłuższego do najkrótszego

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.1.2.3 int nextstat (BoardSetter * bs)

Funkcja akceptująca ustawienie aktualnie ustawianego statku.

Funkcja zwraca:

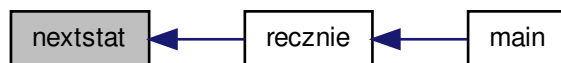
- -1, gdy ustawienie jest zabronione

- 0, gdy wstawiła statek na planszę
- 1, gdy wstawiła ostatni statek na planszę

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.1.2.4 int prevstat (BoardSetter * bs)

Funkcja cofająca ustawienie ostatnio ustawionego statku.

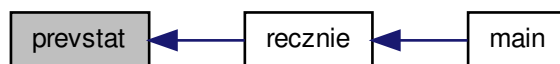
Funkcja zwraca:

- -1, gdy na planszy nie znajduje się jeszcze żaden statek
- 0, gdy ostatnio wstawiony statek został usunięty z planszy

Oto graf wywołań dla tej funkcji:



Oto graf wywołań tej funkcji:

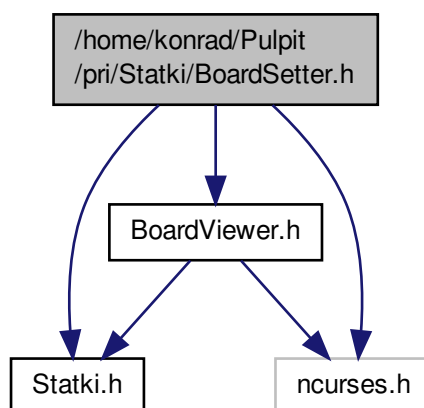


4.2 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/BoardSetter.h

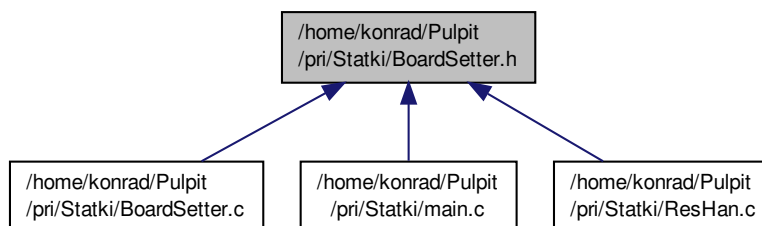
Plik nagłówkowy ustawia statków.

```
#include "Statki.h"
#include <ncurses.h>
#include "BoardViewer.h"
```

Wykres zależności załączania dla BoardSetter.h:



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



Komponenty

- struct `BoardSetter`
Struktura reprezentująca ustawiacz statków.

Funkcje

- void `newbs (BoardSetter *, Plansza *)`
Funkcja inicjująca nowy ustawiacz.
- void `drawbs (BoardSetter *)`
Funkcja rysująca ustawiacz.
- void `updatebs (BoardSetter *)`
Funkcja uaktualniająca ustawiacz, aby nie tworzyć nowego okna za każdym razem, gdy użytkownik wciśnie klawisz.
- void `bsup (BoardSetter *)`
Funkcja przesuwająca aktualnie ustawiany statek w górę
- void `bsdown (BoardSetter *)`
Funkcja przesuwająca aktualnie ustawiany statek w dół
- void `bsleft (BoardSetter *)`
Funkcja przesuwająca aktualnie ustawiany statek w lewo.
- void `bsright (BoardSetter *)`
Funkcja przesuwająca aktualnie ustawiany statek w prawo.
- void `bsrot (BoardSetter *)`
Funkcja obracająca (jeśli się da) aktualnie ustawiany statek.
- int `nextstat (BoardSetter *)`
Funkcja akceptująca ustawienie aktualnie ustawianego statku.
- int `prevstat (BoardSetter *)`
Funkcja cofająca ustawienie ostatnio ustawionego statku.

4.2.1 Opis szczegółowy

Plik nagłówkowy ustawiacza statków.

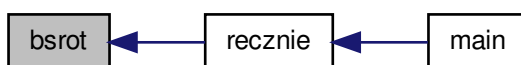
4.2.2 Dokumentacja funkcji

4.2.2.1 void `bsrot (BoardSetter * bs)`

Funkcja obracająca (jeśli się da) aktualnie ustawiany statek.

Statek jest obracany o 90 stopni względem jego skrajnie lewego/górnego segmentu. Jeśli obrót spowoduje wystawienie statku poza planszę, obrót nie jest wykonywany

Oto graf wywoływań tej funkcji:

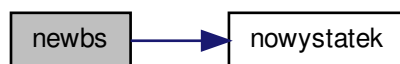


4.2.2.2 void newbs (BoardSetter * bs, Plansza * p)

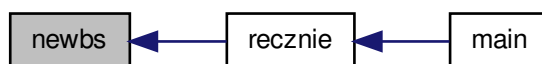
Funkcja inicjująca nowy ustawiacz.

Funkcja wypełnia tablicę kolejnych statków do wstawienia od najdłuższego do najkrótszego

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.2.2.3 int nextstat (BoardSetter * bs)

Funkcja akceptująca ustawienie aktualnie ustawianego statku.

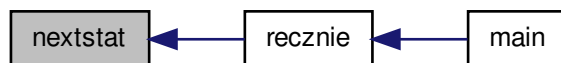
Funkcja zwraca:

- -1, gdy ustawienie jest zabronione
- 0, gdy wstawiła statek na planszę
- 1, gdy wstawiła ostatni statek na planszę

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.2.2.4 int prevstat (BoardSetter * bs)

Funkcja cofająca ustawienie ostatnio ustawionego statku.

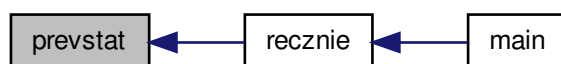
Funkcja zwraca:

- -1, gdy na planszy nie znajduje się jeszcze żaden statek
- 0, gdy ostatnio wstawiony statek został usunięty z planszy

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

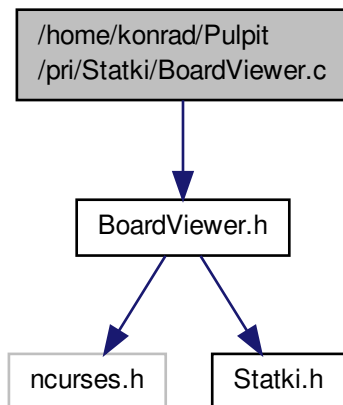


4.3 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/BoardViewer.c

Plik źródłowy obsługi wyświetlania planszy.

```
#include "BoardViewer.h"
```

Wykres zależności załączania dla BoardViewer.c:



Funkcje

- void **newbv** (BoardViewer *bv, Plansza *p, int curs, int mode, int pan)
Funkcja inicjująca nowy wyświetlacz.
- void **drawbv** (BoardViewer *bv)
Funkcja rysująca wyświetlacz.
- void **updatebv** (BoardViewer *bv)
Funkcja uaktualniająca wyświetlacz.
- void **bvup** (BoardViewer *bv)
Funkcja przesuwająca kursor w górę
- void **bvdown** (BoardViewer *bv)
Funkcja przesuwająca kursor w dół
- void **bvleft** (BoardViewer *bv)
Funkcja przesuwająca kursor w lewo.
- void **bvright** (BoardViewer *bv)
Funkcja przesuwająca kursor w prawo.

Zmienne

- const int **WIDTH** =4
Szerokość wyświetlanego pola.
- const int **HEIGHT** =2
Wysokość wyświetlanego pola.
- int **brdsize**
- int **brdt**
Wybrany typ planszy.

4.3.1 Opis szczegółowy

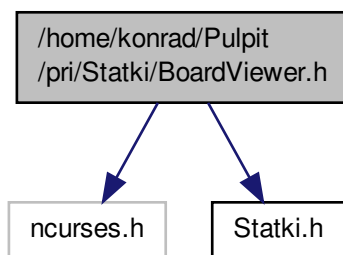
Plik źródłowy obsługi wyświetlania planszy.

4.4 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/BoardViewer.h

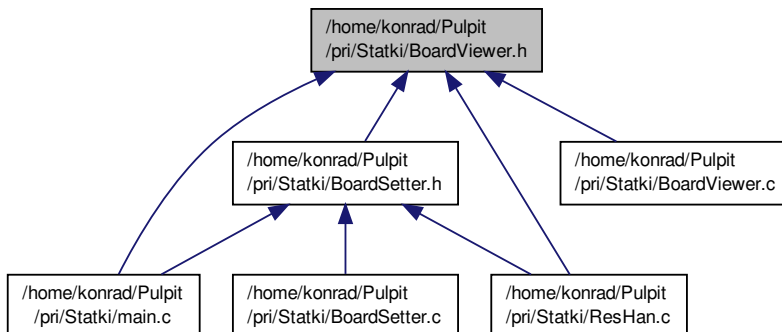
Plik nagłówkowy obsługi wyświetlania planszy.

```
#include <ncurses.h>
#include "Statki.h"
```

Wykres zależności załączania dla BoardViewer.h:



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



Komponenty

- struct [BoardViewer](#)

Struktura reprezentująca "wyświetlacz" planszy.

Wyliczenia

- enum [MODE](#) { **SELF** =0, **OPP** =1 }

Enumeracja oznaczająca tryb wyświetlania planszy (czy niesprawdzone dotąd pola mają być widoczne)

- enum [PAN](#) { **LEFT**, **RIGHT** }

Enumeracja oznaczająca stronę ekranu, po której ma być narysowany wyświetlacz.

Funkcje

- void `newbv` (`BoardViewer *`, `Plansza *`, int, int, int)

Funkcja inicjująca nowy wyświetlacz.

- void `drawbv` (`BoardViewer *`)

Funkcja rysująca wyświetlacz.

- void `updatebv` (`BoardViewer *`)

Funkcja uaktualniająca wyświetlacz.

- void `bvup` (`BoardViewer *`)

Funkcja przesuwająca kursor w górę

- void `bvdown` (`BoardViewer *`)

Funkcja przesuwająca kursor w dół

- void `bvleft` (`BoardViewer *`)

Funkcja przesuwająca kursor w lewo.

- void `bvright` (`BoardViewer *`)

Funkcja przesuwająca kursor w prawo.

4.4.1 Opis szczegółowy

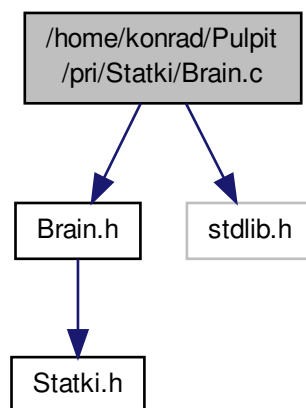
Plik nagłówkowy obsługi wyświetlania planszy.

4.5 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Brain.c

Plik źródłowy obsługi sztucznej inteligencji.

```
#include "Brain.h"
#include <stdlib.h>
```

Wykres zależności załączania dla Brain.c:



Funkcje

- void `newbrain` (`Brain *br`, `Plansza *p`, int `trud`)

Funkcja inicjująca AI dla danej planszy.

- void `getmove` (`Brain *br`, int *y, int *x)
Funkcja pobierająca kolejny ruch AI.

Zmienne

- int `brdsize`
- int `brdt`
Wybrany typ planszy.

4.5.1 Opis szczegółowy

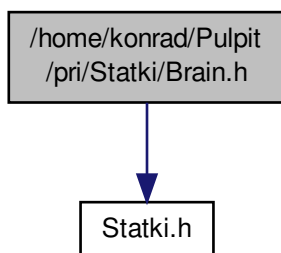
Plik źródłowy obsługi sztucznej inteligencji.

4.6 Dokumentacja pliku `/home/konrad/Pulpit/pri/Statki/Brain.h`

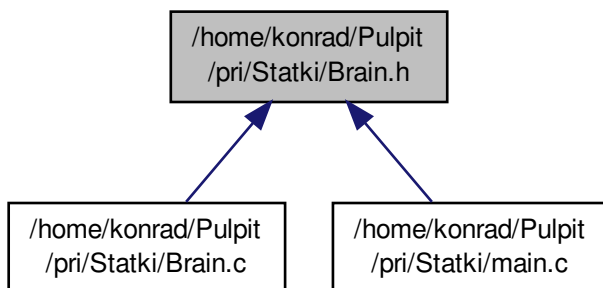
Plik nagłówkowy obsługi sztucznej inteligencji.

```
#include "Statki.h"
```

Wykres zależności załączania dla `Brain.h`:



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



Komponenty

- struct **Pole**
Struktura reprezentująca pojedyncze pole (współrzędne na planszy)
- struct **Brain**
Struktura reprezentująca jednostkę sztucznej inteligencji.

Funkcje

- void **newbrain** (Brain *, Plansza *, int)
Funkcja inicjująca AI dla danej planszy.
- void **getmove** (Brain *, int *, int *)
Funkcja pobierająca kolejny ruch AI.

4.6.1 Opis szczegółowy

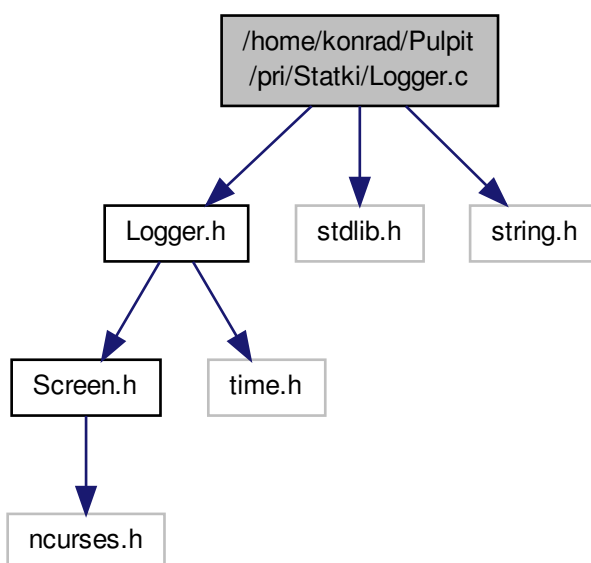
Plik nagłówkowy obsługi sztucznej inteligencji.

4.7 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Logger.c

Plik źródłowy obsługi modułu zapisującego przebieg rozgrywki.

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

Wykres zależności załączania dla Logger.c:



Funkcje

- void **newlm** (**LogMove** *lm, int l, int c, int pid, int res)
Funkcja inicjująca nowy wpis.
- void **newlog** (**Logger** *lg, int size, int width, int height)
*Funkcja inicjująca nowy **Logger** na podstawie wartości przekazanych poprzez parametry.*
- void **addmove** (**Logger** *lg, int l, int c, int pid, int res)
Funkcja dodająca nowy wpis do Loggera na podstawie wartości przekazanych poprzez parametry.
- void **endgame** (**Logger** *lg, int reason)
Funkcja kończąca rozgrywkę
- int **exportlog** (**Logger** *lg)
Funkcja eksportująca zapis rozgrywki do pliku (sama generuje nazwę pliku)
- void **drawlog** (**Logger** *lg)
*Funkcja rysująca **Logger**.*
- void **updatelog** (**Logger** *lg)
*Funkcja uaktualniająca **Logger**.*

Zmienne

- int **brdt**
Wybrany typ planszy.
- int **brdsize**

4.7.1 Opis szczegółowy

Plik źródłowy obsługi modułu zapisującego przebieg rozgrywki.

4.7.2 Dokumentacja funkcji

4.7.2.1 int exportlog (**Logger** *)

Funkcja eksportująca zapis rozgrywki do pliku (sama generuje nazwę pliku)

Zwraca 0, gdy wyeksportowano pomyślnie

Oto graf wywoływań tej funkcji:

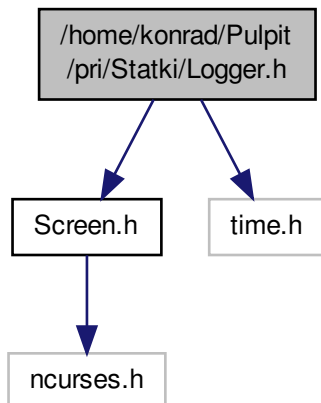


4.8 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Logger.h

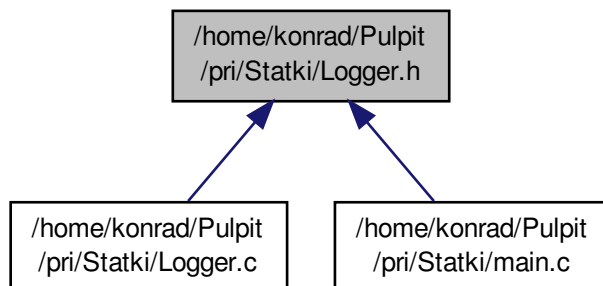
Plik nagłówkowy obsługi modułu zapisującego przebieg rozgrywki.

```
#include "Screen.h"  
#include <time.h>
```

Wykres zależności załączania dla Logger.h:



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



Komponenty

- struct [LogMove](#)
Struktura reprezentująca pojedynczy wpis do modułu logującego przebieg rozgrywki.
- struct [Logger](#)
Struktura reprezentująca moduł logujący przebieg rozgrywki.

Wyliczenia

- enum {
 WIN, SHOT, DESTROYED, MISS,
 FORB }

Enumeracja oznaczająca odpowiedzi po wykonaniu ruchu.

- enum { P1, P2, IRR }

Enumeracja oznaczająca powód zakończenia rozgrywki (Gracz 1 / Gracz 2 / Przerwane)

Funkcje

- void newlm (LogMove *, int, int, int, int)

Funkcja inicjująca nowy wpis.

- void newlog (Logger *, int, int, int)

Funkcja inicjująca nowy Logger na podstawie wartości przekazanych poprzez parametry.

- void addmove (Logger *, int, int, int, int)

Funkcja dodająca nowy wpis do Loggera na podstawie wartości przekazanych poprzez parametry.

- int exportlog (Logger *)

Funkcja eksportująca zapis rozgrywki do pliku (sama generuje nazwę pliku)

- void endgame (Logger *, int)

Funkcja kończąca rozgrywkę

- void drawlog (Logger *)

Funkcja rysująca Logger.

- void updatelog (Logger *)

Funkcja uaktualniająca Logger.

4.8.1 Opis szczegółowy

Plik nagłówkowy obsługi modułu zapisującego przebieg rozgrywki.

4.8.2 Dokumentacja funkcji

4.8.2.1 int exportlog (Logger *)

Funkcja eksportująca zapis rozgrywki do pliku (sama generuje nazwę pliku)

Zwraca 0, gdy wyeksportowano pomyślnie

Oto graf wywołań tej funkcji:



4.9 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/main.c

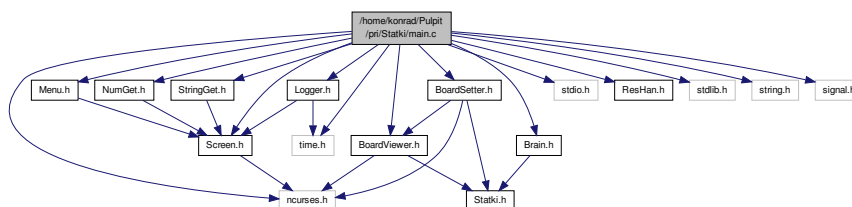
Plik główny.

```

#include <ncurses.h>
#include <stdio.h>
#include "Screen.h"
#include "Menu.h"
#include "ResHan.h"
#include "BoardViewer.h"
#include "Brain.h"
#include "BoardSetter.h"
#include "NumGet.h"
#include "StringGet.h"
#include "Logger.h"
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <signal.h>

```

Wykres zależności załączania dla main.c:



Wyliczenia

- enum {
BYE, MENU, QUIT, LEVEL,
GEN, GRA, BLAD, OPPMENU,
TORMENU, METAST, RECZNIE, LOSOWO,
ZPLIKU, GSIZE }

Aktualny stan programu (co jest aktualnie wyświetlane na ekranie)

- enum { **LATWY** =0, **SREDNI** =1, **TRUDNY** =2 }

Wybrany poziom trudności.

- enum { **VSCPU, VSP2** }

Wybrany przeciwnik.

- enum loader {
NOERR =0, **SYNERR, SHIPERR, READERR,**
SIZERR }

Błędy możliwe do zwrócenia przez funkcję wczytującą planszę z pliku.

Funkcje

- void **genmenus** ()

Funkcja inicjalizująca wszystkie menu, które będą wyświetlane w programie.

- int **quit** ()

Funkcja reprezentująca stan programu przed wyjściem (QUIT)

- int **suref** ()

Funkcja wyświetlająca menu "Czy jesteś pewien?".

- int **sure** ()

- Funkcja odpowiadająca za obsługę wyboru użytkownika w menu "Czy jesteś pewien?" (uruchamiana podczas rozgrywki)*

 - int **sure2** ()
- Funkcja odpowiadająca za obsługę wyboru użytkownika w menu "Czy jesteś pewien?" (uruchamiana podczas ustawiania planszy)*

 - int **ruch** (**Plansza** *p, int l, int c)

Funkcja obsługująca pojedynczy ruch na zadanej planszy.
- Funkcja obsługująca przebieg tury.*

 - int **tura** (**Plansza** *p, int l, int c, **Logger** *lg, int pid)
- Funkcja obsługująca wyświetlanie ekranu zmiany gracza.*

 - void **chscreen** (int id)
- Funkcja obsługująca wyświetlanie ekranu końca rozgrywki.*

 - void **winscreen** (int id)
- Funkcja wyświetlająca menu "Czy chcesz wyeksportować przebieg rozgrywki?".*

 - int **logexpf** ()
- Funkcja obsługująca zapytanie użytkownika o eksport przebiegu rozgrywki.*

 - void **logexp** (**Logger** *lg)
- Funkcja reprezentująca stan programu, oznaczający rozgrywkę (GRA)*

 - int **gra** ()
- Funkcja reprezentująca stan programu, oznaczający ręczne ustawianie statków (RECZNIE)*

 - int **recznie** ()
- Funkcja wczytująca planszę z (wcześniej otwartego) pliku.*

 - int **loadboard** (**Plansza** *p, FILE *f)
- Funkcja reprezentująca stan programu, oznaczający wczytywanie planszy z pliku.*

 - int **zpliku** ()
- Funkcja reprezentująca stan programu, oznaczający losowe rozstawienie statków na planszy użytkownika (LOSOWO)*

 - int **losowo** ()
- Funkcja reprezentująca stan programu, oznaczający menu wyboru sposobu generowania planszy.*

 - int **gen** ()
- Funkcja reprezentująca stan programu, oznaczający menu wyboru typu planszy (kwadrat/torus)*

 - int **tormenu** ()
- Funkcja reprezentująca stan programu, obsługujący niuanse związane z opcją wyboru drugiego gracza.*

 - int **metaf** ()
- Funkcja reprezentująca stan programu, oznaczający menu główne.*

 - int **menu** ()
- Funkcja reprezentująca stan programu, oznaczający menu wyboru przeciwnika.*

 - int **oppm** ()
- Funkcja reprezentująca stan programu, oznaczający menu wyboru poziomu trudności.*

 - int **level** ()
- Funkcja reprezentująca stan programu, oznaczający ekran pobierania od użytkownika wielkości planszy.*

 - int **gsize** ()
- Główna funkcja.*

 - int **main** ()

Zmienne

- [Menu mainmenu](#)
Obiekt reprezentujący menu główne.
- [Menu levelmenu](#)
Obiekt reprezentujący menu wyboru poziomu trudności.
- [Menu genmenu](#)
Obiekt reprezentujący menu wyboru sposobu generowania planszy.
- [Menu suremenu](#)
Obiekt reprezentujący menu "Czy jesteś pewien?".
- [Menu logexpmenu](#)
Obiekt reprezentujący menu "Czy chcesz wyeksportować przebieg rozgrywki?".
- [Menu torusmenu](#)
Obiekt reprezentujący menu wyboru typu planszy (kwadrat/torus)
- [Menu oppmenu](#)
Obiekt reprezentujący menu wyboru przeciwnika (człowiek/komputer)
- [NumGet sizeng](#)
Obiekt reprezentujący ekran wprowadzania rozmiaru planszy.
- [StringGet sg](#)
Obiekt reprezentujący ekran wprowadzania nazwy pliku.
- [Logger lg](#)
Obiekt reprezentujący moduł zapisujący przebieg rozgrywki.
- [Menu * aktmenu](#)
Wskaźnik na aktualnie wyświetlane menu.
- `enum { ... } stan`
Aktualny stan programu (co jest aktualnie wyświetlane na ekranie)
- `enum { ... } trud`
Wybrany poziom trudności.
- `enum { ... } opp`
Wybrany przeciwnik.
- `int menuplayer`
Numer aktualnie obsługiwanego gracza w menu.
- [VIEWT brdt](#)
Wybrany typ planszy.
- `int brdsize =10`
- `char * naglowek`
Wskaźnik na napis, który ma być wyświetlony w górnej belce.
- `char * stopka`
Wskaźnik na napis, który ma być wyświetlony w dolnej belce.
- [Plansza plansza1](#)
Obiekt reprezentujący planszę użytkownika (Gracza 1)
- [Plansza plansza2](#)
Obiekt reprezentujący planszę komputera / gracza drugiego.
- [Plansza * aktplansza](#)
Wskaźnik na aktualnie ustawianą planszę
- [BoardViewer lewa1](#)
Obiekt reprezentujący wyświetlacz planszy gracza (dla Gracza 1)
- [BoardViewer prawa1](#)
Obiekt reprezentujący wyświetlacz planszy przeciwnika (dla Gracza 1)
- [BoardViewer lewa2](#)
Obiekt reprezentujący wyświetlacz planszy gracza (dla Gracza 2)

- **BoardViewer prawa2**
Obiekt reprezentujący wyświetlacz planszy przeciwnika (dla Gracza 2)
- **BoardViewer * aktlewa**
Wskaźnik na aktualnie wyświetlany wyświetlacz po lewej stronie.
- **BoardViewer * aktprawa**
Wskaźnik na aktualnie wyświetlany wyświetlacz po prawej stronie.
- **BoardSetter ustawiacz**
Obiekt reprezentujący narzędzie do ustawiania statków.
- **Brain mozg**
Obiekt reprezentujący sztuczną inteligencję komputera.
- char * **maincontent** [3] = {"Rozpocznij gre", "Wyjście"}
Napisy do wyświetlenia w głównym menu.
- char * **levelcontent** [3] = {"Łatwy", "Średni", "Trudny"}
Napisy do wyświetlenia w menu wyboru poziomu trudności.
- char * **gencontent** [3] = {"Recznie", "Losowo", "Z pliku"}
Napisy do wyświetlenia w menu wyboru sposobu generowania planszy.
- char * **surecontent** [2] = {"Nie, wracam do gry", "Tak, chce zakonczyc gre"}
Napisy do wyświetlenia w menu "Czy jesteś pewien?".
- char * **logexpcontent** [2] = {"Nie, dziekuje", "Tak, poprosze"}
Napisy do wyświetlenia w menu "Czy chcesz wyeksportować przebieg rozgrywki?".
- char * **toruscontent** [2] = {"Kwadrat", "Torus"}
Napisy do wyświetlenia w menu wyboru typu planszy (kwadrat/torus)
- char * **oppcontent** [2] = {"Komputer", "Człowiek"}
Napisy do wyświetlenia w menu wyboru przeciwnika (człowiek/komputer)

Napisy

używane w górnym i dolnym pasku

- char * **mennag** = "Statki ULTIMATE, Konrad Sikorski 2014"
- char * **levnag** = "Wybierz poziom trudności:"
- char * **gennag** = "Wybierz sposob generowania planszy:"
- char * **oppnag** = "Wybierz przeciwnika:"
- char * **tornag** = "Wybierz typ planszy:"
- char * **surenag** = "Czy na pewno chcesz zakonczyc gre?"
- char * **logexpnag** = "Czy chcesz wyeksportowac zapis przebiegu rozgrywki?"
- char * **err** = "Wystapil blad."
- char * **exported** = "Wyeksportowano pomyslnie!"
- char * **ngnag** = "Wybierz rozmiar planszy:"
- char * **ngstop** = "Uzyj strzalek aby zwiekszyć lub zmniejszyć wartość."
- char * **pname1** = "Gracz 1"
- char * **pname2** = "Gracz 2"
- char * **forbid** = "Ruch zabroniony."
- char * **usermiss** = "Spudlowales!"
- char * **usershot** = "Trafiony!"
- char * **userdest** = "Trafiony-zatopiony!"
- char * **userwin** = "Wygrales!"
- char * **cpumiss** = "Na szczescie, komputer spudlowal!"
- char * **cpushot** = "Doznales obrazen!"
- char * **cpuwin** = "Przegrales!"
- char * **author** = "Konrad Sikorski"
- char * **pusty** = ""
- char * **menstop** = "Poruszanie sie po menu: strzałki / ENTER / BACKSPACE"
- char * **steering** = "Uzyj strzalek, aby wybrac **pole** do zaatakowania."
- char * **steering2** = "Wcisnij ENTER, aby zaatakowac."
- char * **anybut** = "Nacisnij dowolny klawisz, aby kontynuowac."
- char * **niema** = "Nie ma statkow na planszy."
- char * **notable** = "Nie mozna tu umiescic statku."
- char * **steering3** = "Uzyj strzalek, aby wybrac miejsce."
- char * **steering4** = "Spacja - obrot / Enter - akceptacja"

4.9.1 Opis szczegółowy

Plik główny. W tym pliku zaimplementowany jest główny mechanizm gry:

- główna pętla gry
- funkcje wywoływane po nastąpieniu danego stanu
- obsługa mechanizmu turowego

4.9.2 Dokumentacja funkcji

4.9.2.1 `int loadboard (Plansza * p, FILE * f)`

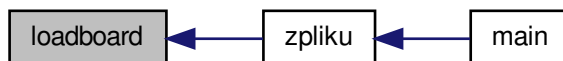
Funkcja wczytująca planszę z (wcześniej otwartego) pliku.

Funkcja nie zamyka pliku, trzeba to obsłużyć poza funkcją.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

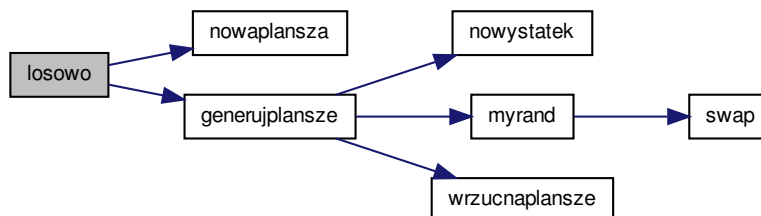


4.9.2.2 `int losowo ()`

Funkcja reprezentująca stan programu, oznaczający losowe rozstawienie statków na planszy użytkownika (LOSO-WO)

W przeciwieństwie do innych stanów nie wchodzi w interakcję z użytkownikiem, ani nic nie wyświetla.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

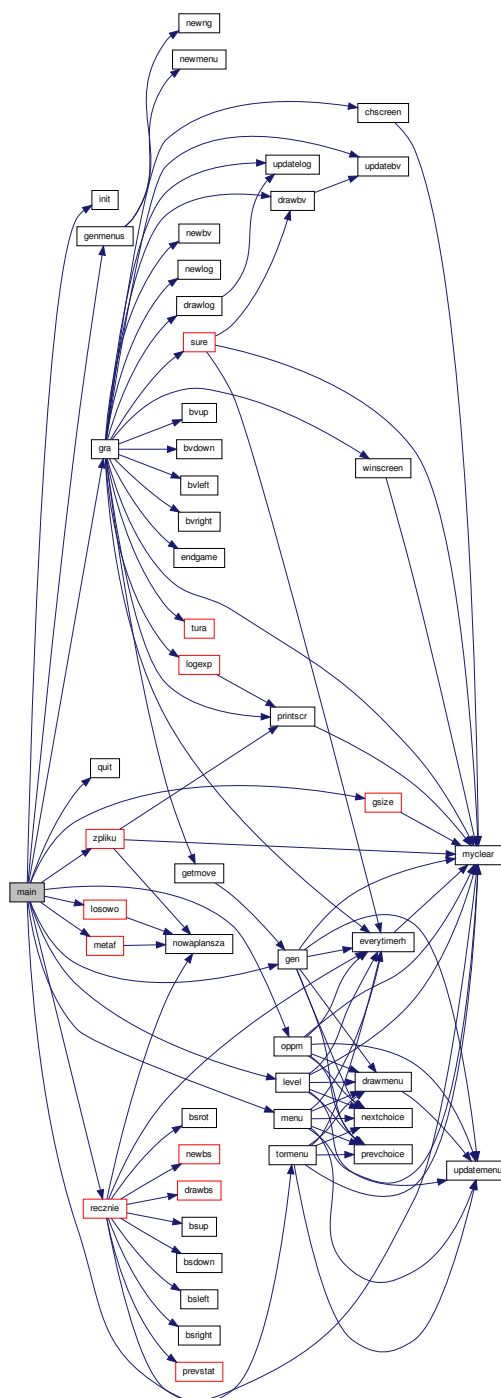


4.9.2.3 `int main ()`

Główna funkcja.

Zawiera inicjalizację `ncurses`, kolorów, wszystkich menu oraz napisu w dolnej belce, a następnie pętlę gry. Mechanizm przechodzenia pomiędzy różnymi stanami gry odbywa się następująco: w głównej pętli gry zapamiętywany jest aktualny stan gry. Następnie uruchamiana jest funkcja reprezentująca dany stan. Wartość przez nią zwrócona określa stan, do którego należy przejść.

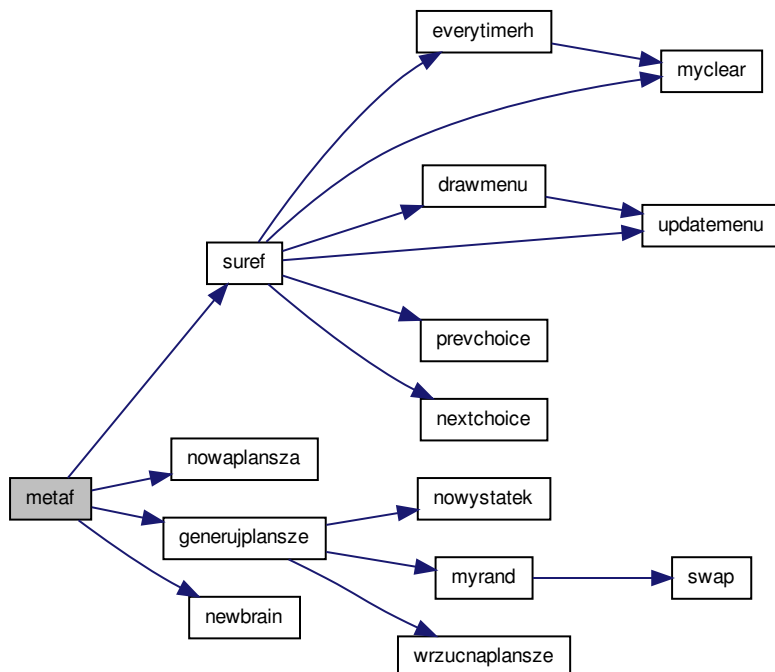
Oto graf wywołań dla tej funkcji:



4.9.2.4 int metaf ()

Funkcja reprezentująca stan programu, obsługujący niuanse związane z opcją wyboru drugiego gracza. W przeciwieństwie do innych stanów nie wchodzi w interakcję z użytkownikiem, ani nic nie wyświetla.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

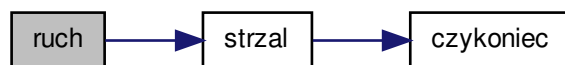


4.9.2.5 int ruch (Plansza * p, int l, int c)

Funkcja obsługująca pojedynczy ruch na zadanej planszy.

Zwraca stan gry po wykonaniu ruchu

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

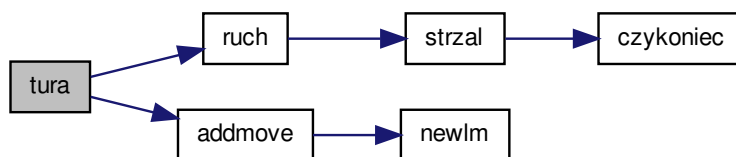


4.9.2.6 int tura (Plansza * p, int l, int c, Logger * lg, int pid)

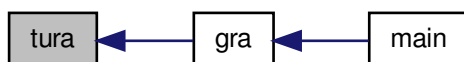
Funkcja obsługująca przebieg tury.

Zwraca wartość logiczną, oznaczającą, czy po wykonaniu danej tury rozgrywka ma być nadal kontynuowana

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

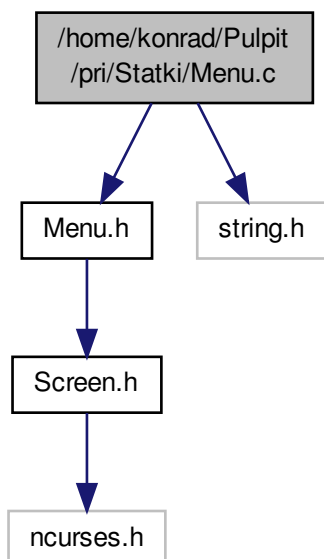


4.10 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Menu.c

Plik źródłowy obsługi menu.

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

Wykres zależności załączania dla Menu.c:



Funkcje

- void **newmenu** (Menu *m, int n, char *choices[], int w)
Funkcja inicjująca menu wartościami przekazanymi w parametrach.
- void **nextchoice** (Menu *m)
Funkcja realizująca zmianę aktualnie wybranej opcji na następną
- void **prevchoice** (Menu *m)
Funkcja realizująca zmianę aktualnie wybranej opcji na poprzednią
- void **drawmenu** (Menu *m)
Funkcja wykonująca ustalenie pozycji menu oraz rysowanie go.
- void **updatemenu** (Menu *m)
Funkcja uaktualniająca wyświetlanie menu.

4.10.1 Opis szczegółowy

Plik źródłowy obsługi menu.

4.10.2 Dokumentacja funkcji

4.10.2.1 void drawmenu (Menu * m)

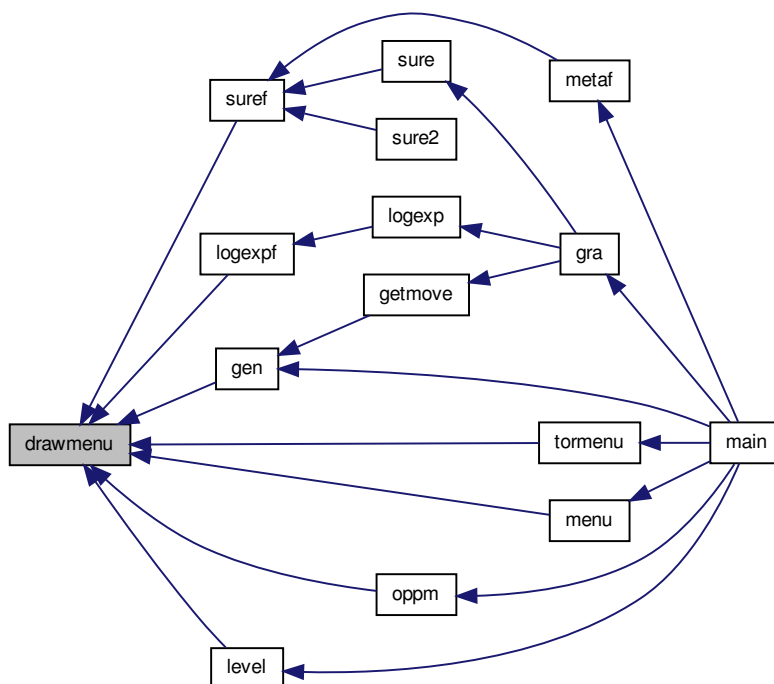
Funkcja wykonująca ustalenie pozycji menu oraz rysowanie go.

Funkcja jest wywoływana, gdy menu jest otwierane (zmiana stanu) gry oraz za każdym razem, gdy zmieniony zostanie rozmiar terminala. Tworzy okno (ncurses), w którym będzie rysowane menu - na środku ekranu zarówno w pionie, jak i w poziomie. Następnie wypisuje opcje menu oraz wywołuje funkcję, która podświetli aktualnie wybraną opcję i odświeży okno.

Oto graf wywołań dla tej funkcji:



Oto graf wywołań tej funkcji:

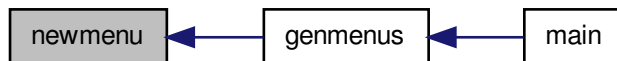


4.10.2.2 void newmenu (Menu * m, int n, char * choices[], int w)

Funkcja inicjująca menu wartościami przekazanymi w parametrach.

Funkcja inicjuje liczbę wyborów, napisy oznaczające wybory oraz szerokość menu

Oto graf wywoływań tej funkcji:

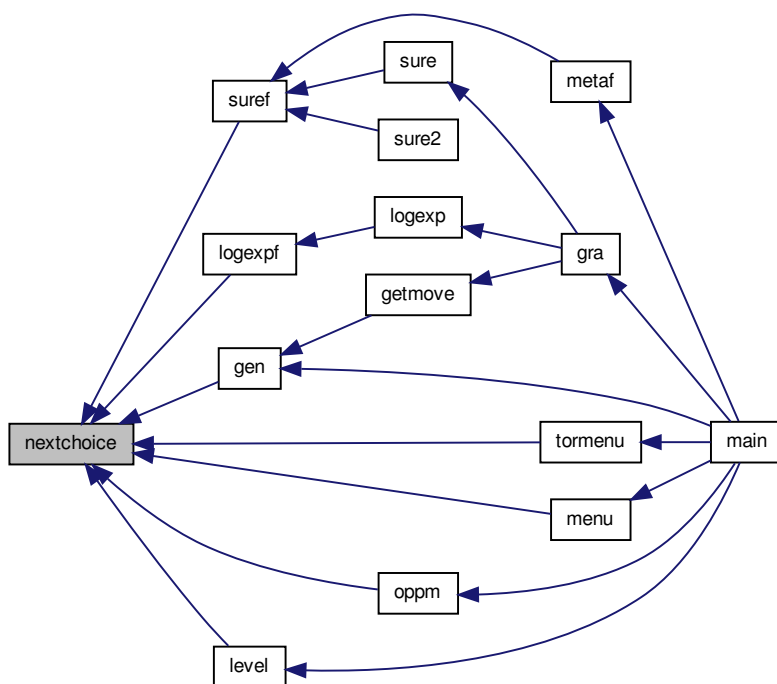


4.10.2.3 void nextchoice (Menu * m)

Funkcja realizująca zmianę aktualnie wybranej opcji na następną

Gdy osiągnięty zostanie koniec menu, aktualnie wybraną staje się pierwsza opcja

Oto graf wywoływań tej funkcji:

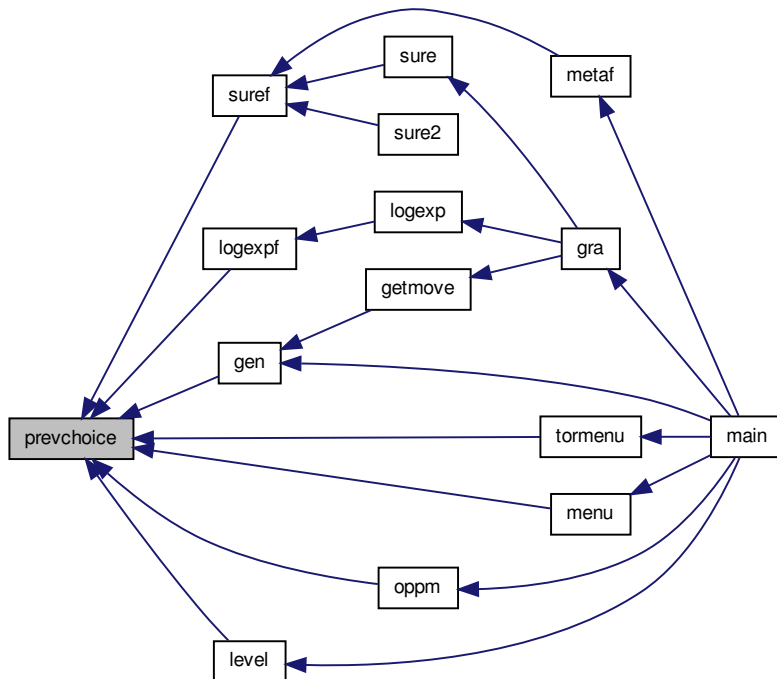


4.10.2.4 void prevchoice (Menu * m)

Funkcja realizująca zmianę aktualnie wybranej opcji na poprzednią

Gdy osiągnięty zostanie początek menu, aktualnie wybraną staje się ostatnia opcja

Oto graf wywoływań tej funkcji:

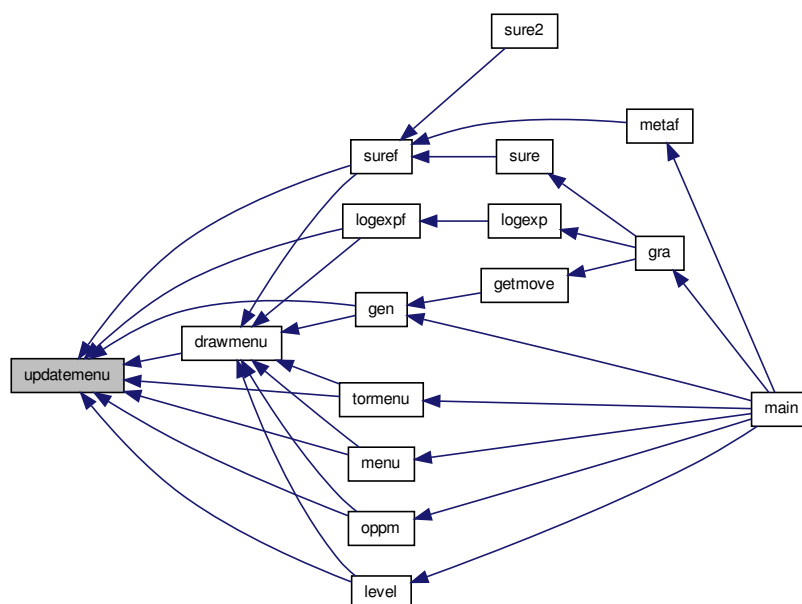


4.10.2.5 void updatemenu (Menu * m)

Funkcja uaktualniająca wyświetlanie menu.

Funkcja jest wywoływana, gdy zmieni się aktualny wybór w danym menu lub gdy menu jest przerysowywane. Stworzona, aby nie przerysowywać całego menu za każdym razem, gdy użytkownik zmieni aktualnie wybraną opcję

Oto graf wywoływań tej funkcji:

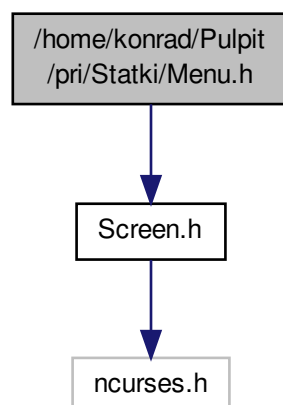


4.11 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Menu.h

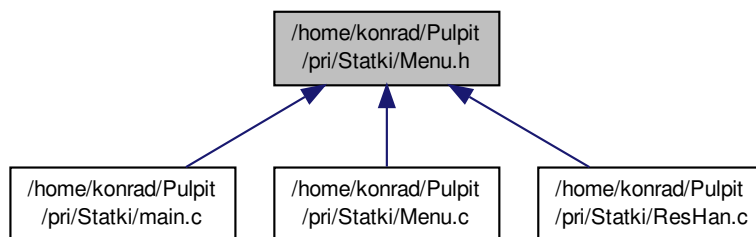
Plik nagłówkowy obsługi menu.

```
#include "Screen.h"
```

Wykres zależności załączania dla Menu.h:



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



Komponenty

- struct [Menu](#)

Struktura reprezentująca menu.

Funkcje

- void [newmenu](#) ([Menu](#) *, int, char *[], int)

Funkcja inicjująca menu wartościami przekazanymi w parametrach.

- void [nextchoice](#) ([Menu](#) *)

Funkcja realizująca zmianę aktualnie wybranej opcji na następną

- void [prevchoice](#) ([Menu](#) *)

Funkcja realizująca zmianę aktualnie wybranej opcji na poprzednią

- void [drawmenu](#) ([Menu](#) *)

Funkcja wykonująca ustalenie pozycji menu oraz rysowanie go.

- void [updatemenu](#) ([Menu](#) *)

Funkcja uaktualniająca wyświetlanie menu.

4.11.1 Opis szczegółowy

Plik nagłówkowy obsługi menu.

4.11.2 Dokumentacja funkcji

4.11.2.1 void drawmenu ([Menu](#) * *m*)

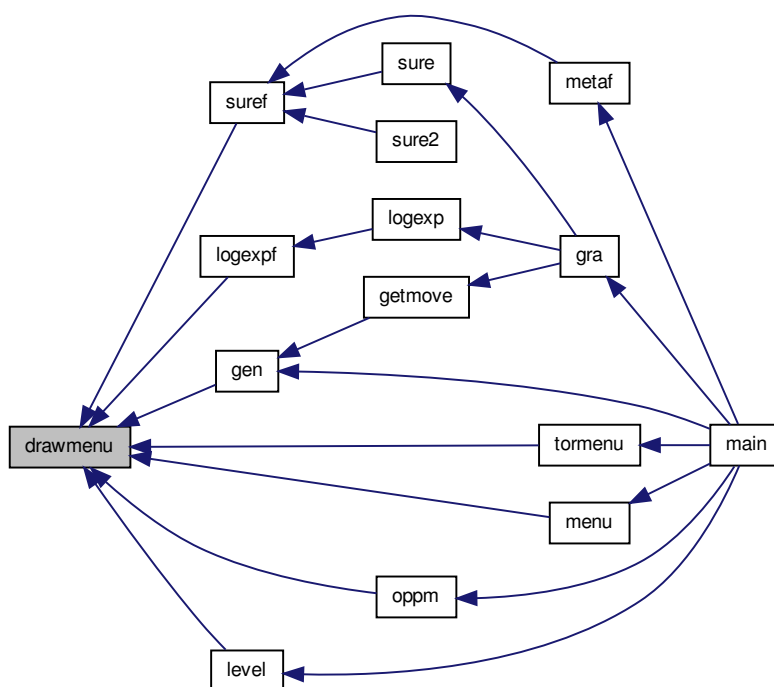
Funkcja wykonująca ustalenie pozycji menu oraz rysowanie go.

Funkcja jest wywoływana, gdy menu jest otwierane (zmiana stanu) gry oraz za każdym razem, gdy zmieniony zostanie rozmiar terminala. Tworzy okno (ncurses), w którym będzie rysowane menu - na środku ekranu zarówno w pionie, jak i w poziomie. Następnie wypisuje opcje menu oraz wywołuje funkcję, która podświetli aktualnie wybraną opcję i odświeży okno.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

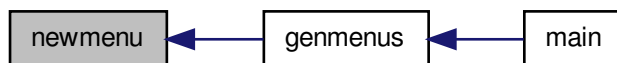


4.11.2.2 void newmenu (Menu * m, int n, char * choices[], int w)

Funkcja inicjująca menu wartościami przekazanymi w parametrach.

Funkcja inicjuje liczbę wyborów, napisy oznaczające wybory oraz szerokość menu

Oto graf wywoływań tej funkcji:

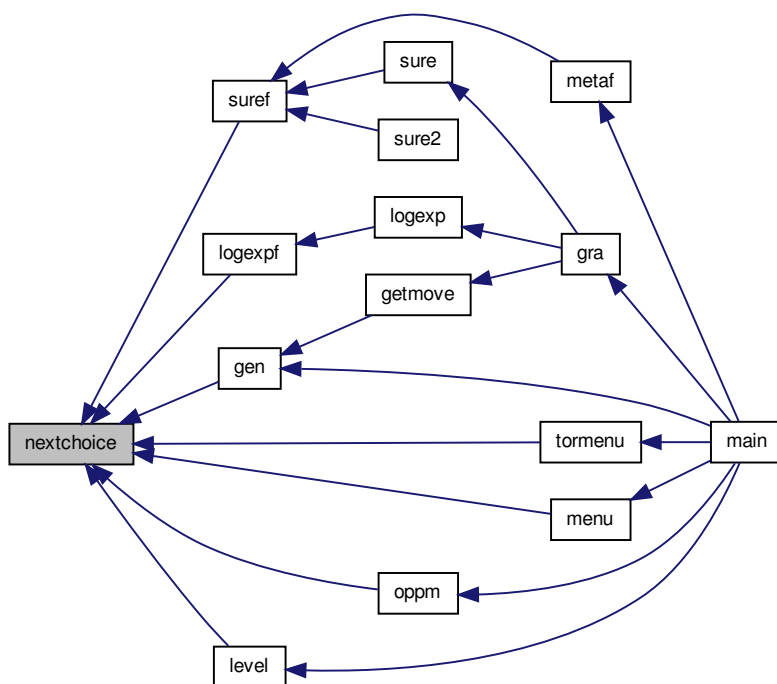


4.11.2.3 void nextchoice (Menu * m)

Funkcja realizująca zmianę aktualnie wybranej opcji na następną

Gdy osiągnięty zostanie koniec menu, aktualnie wybraną staje się pierwsza opcja

Oto graf wywoływań tej funkcji:

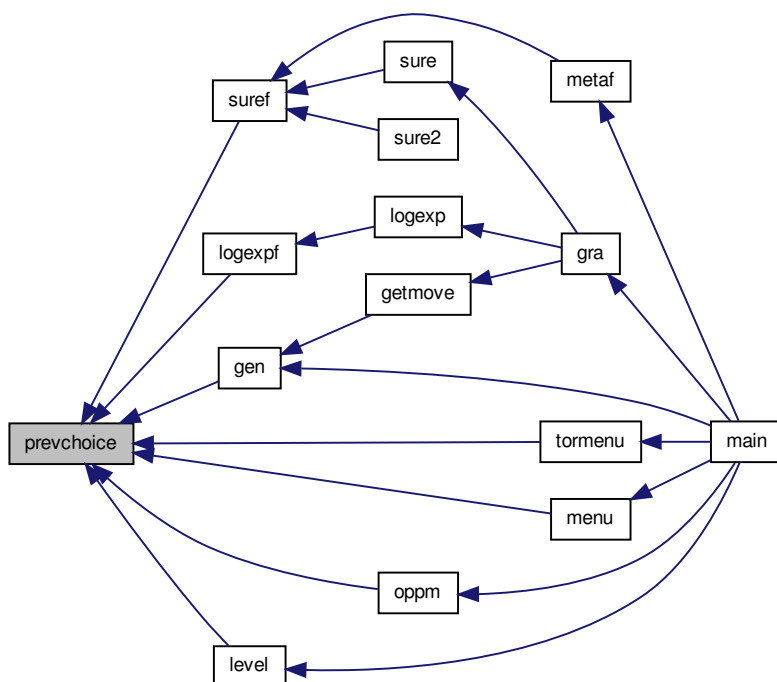


4.11.2.4 void prevchoice (Menu * m)

Funkcja realizująca zmianę aktualnie wybranej opcji na poprzednią

Gdy osiągnięty zostanie początek menu, aktualnie wybraną staje się ostatnia opcja

Oto graf wywołań tej funkcji:

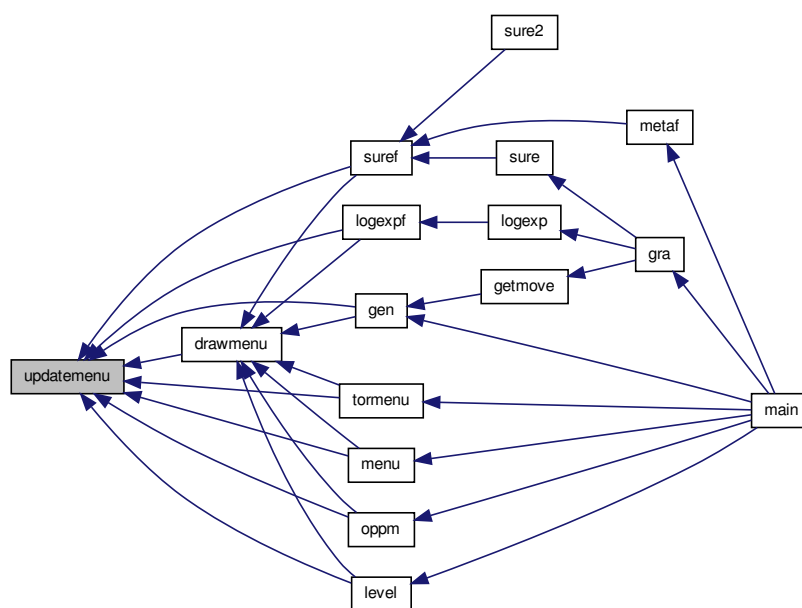


4.11.2.5 void updatemenu (Menu * m)

Funkcja uaktualniająca wyświetlanie menu.

Funkcja jest wywoływana, gdy zmieni się aktualny wybór w danym menu lub gdy menu jest przerysowywane. Stworzona, aby nie przerysowywać całego menu za każdym razem, gdy użytkownik zmieni aktualnie wybraną opcję

Oto graf wywoływań tej funkcji:

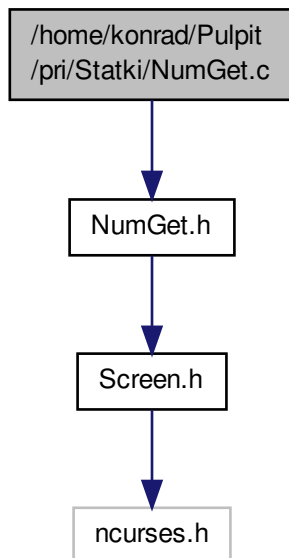


4.12 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/NumGet.c

Plik źródłowy obsługi pobieracza rozmiaru.


```
#include "NumGet.h"
```

Wykres zależności załączania dla NumGet.c:



Funkcje

- void **newng** (NumGet *ng, int now, int minn, int maxx, int w)
Funkcja inicjująca pobieracz wartościami przekazanymi w parametrach.
- void **incrng** (NumGet *ng)
Funkcja powiększająca aktualną wartość o 1.
- void **decrng** (NumGet *ng)
Funkcja zmniejszająca aktualną wartość o 1.
- void **incr10ng** (NumGet *ng)
Funkcja powiększająca aktualną wartość o 10.
- void **decr10ng** (NumGet *ng)
Funkcja zmniejszająca aktualną wartość o 10.
- void **drawng** (NumGet *ng)
Funkcja wykonująca ustalenie pozycji pobieracza oraz rysowanie go.
- void **updateng** (NumGet *ng)
Funkcja uaktualniająca wyświetlanie pobieracza.

4.12.1 Opis szczegółowy

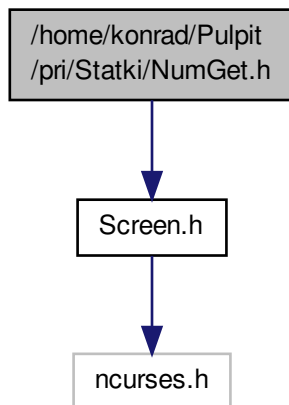
Plik źródłowy obsługi pobieracza rozmiaru.

4.13 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/NumGet.h

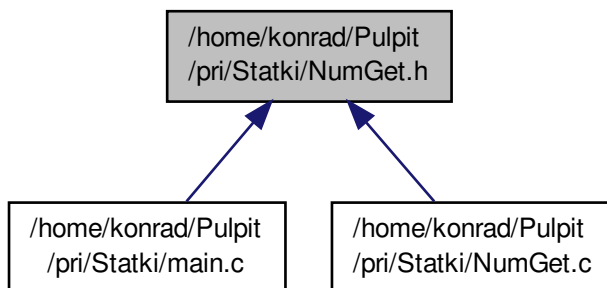
Plik nagłówkowy obsługi pobieracza rozmiaru planszy.

```
#include "Screen.h"
```

Wykres zależności załączania dla NumGet.h:



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



Komponenty

- struct `NumGet`

Struktura reprezentująca pobieracz rozmiaru.

Funkcje

- void `newng (NumGet *, int, int, int, int)`

Funkcja inicjująca pobieracz wartościami przekazanymi w parametrach.

- void `incrng (NumGet *)`

Funkcja powiększająca aktualną wartość o 1.

- void `decrng (NumGet *)`

Funkcja zmniejszająca aktualną wartość o 1.

- void `incr10ng` (NumGet *)

Funkcja powiększająca aktualną wartość o 10.

- void `decr10ng` (NumGet *)

Funkcja zmniejszająca aktualną wartość o 10.

- void `drawng` (NumGet *)

Funkcja wykonująca ustalanie pozycji pobieracza oraz rysowanie go.

- void `updateng` (NumGet *)

Funkcja uaktualniająca wyświetlanie pobieracza.

4.13.1 Opis szczegółowy

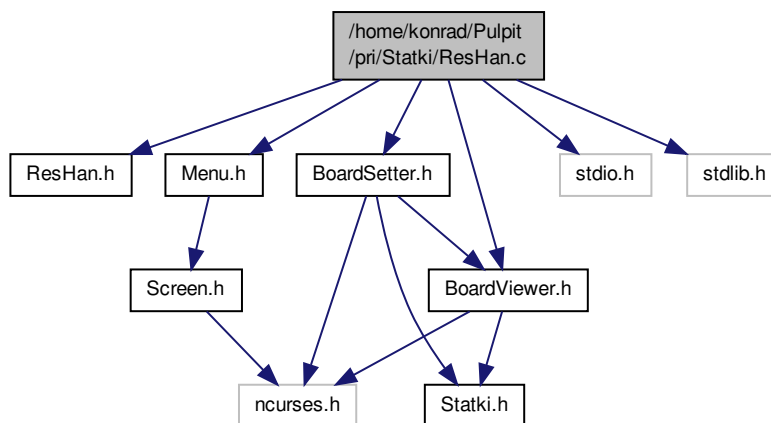
Plik nagłówkowy obsługi pobieracza rozmiaru planszy.

4.14 Dokumentacja pliku `/home/konrad/Pulpit/pri/Statki/ResHan.c`

Plik źródłowy funkcji przerwań (zmiana rozmiaru terminala)

```
#include "ResHan.h"
#include "Menu.h"
#include "BoardViewer.h"
#include "BoardSetter.h"
#include <stdio.h>
#include <stdlib.h>
```

Wykres zależności załączania dla `ResHan.c`:



Funkcje

- void `everytimerh` ()

Funkcja wykonywana przy zmianie rozmiaru terminala niezależnie od aktualnego stanu programu.

Zmienne

- `Menu` * `aktmenu`

Wskaźnik na aktualnie wyświetlane menu.

- [BoardViewer](#) * [aktlewa](#)

Wskaźnik na aktualnie wyświetlany wyświetlacz po lewej stronie.

- [BoardViewer](#) * [aktprawa](#)

Wskaźnik na aktualnie wyświetlany wyświetlacz po prawej stronie.

- [BoardSetter](#) [ustawiacz](#)

Obiekt reprezentujący narzędzie do ustawiania statków.

4.14.1 Opis szczegółowy

Plik źródłowy funkcji przerwań (zmiana rozmiaru terminala)

4.14.2 Dokumentacja funkcji

4.14.2.1 void everytimerh ()

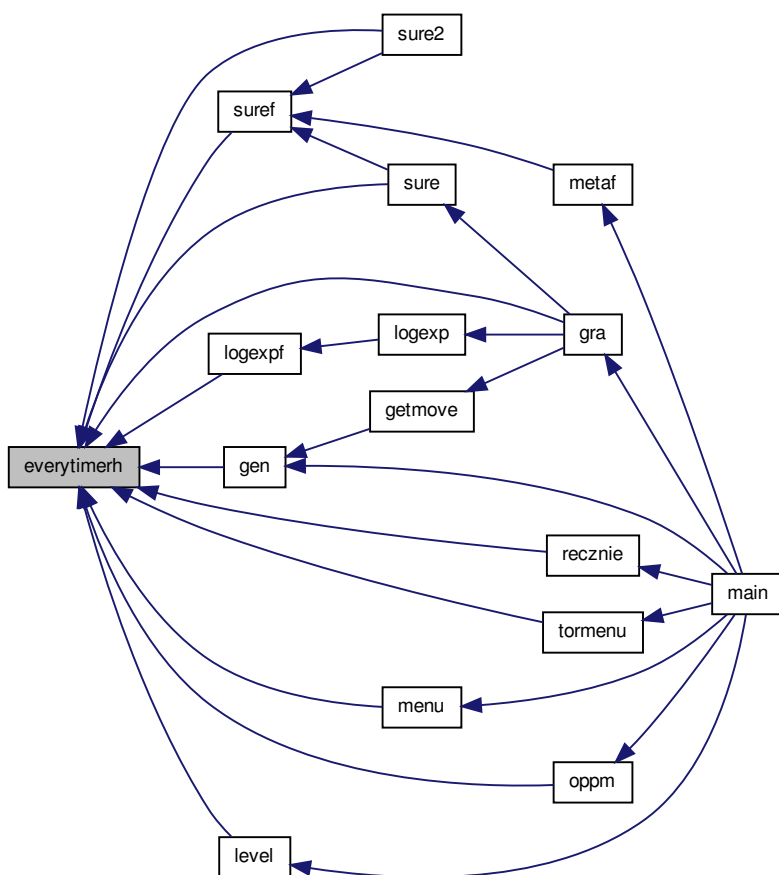
Funkcja wykonywana przy zmianie rozmiaru terminala niezależnie od aktualnego stanu programu.

Sprawdza, czy rozmiar terminala nie jest zbyt mały i inicjuje ekran od nowa. Jeśli terminal jest zbyt mały, kończy program.

Oto graf wywołań dla tej funkcji:



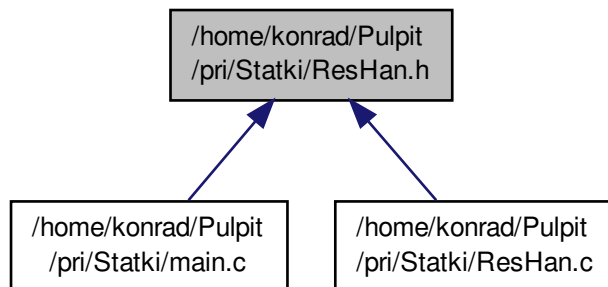
Oto graf wywoływań tej funkcji:



4.15 Dokumentacja pliku `/home/konrad/Pulpit/pri/Statki/ResHan.h`

Plik nagłówkowy funkcji przerwań (zmiana rozmiaru terminala)

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



Funkcje

- void `everytimerh` ()

Funkcja wykonywana przy zmianie rozmiaru terminala niezależnie od aktualnego stanu programu.

4.15.1 Opis szczegółowy

Plik nagłówkowy funkcji przerwań (zmiana rozmiaru terminala)

4.15.2 Dokumentacja funkcji

4.15.2.1 void `everytimerh` ()

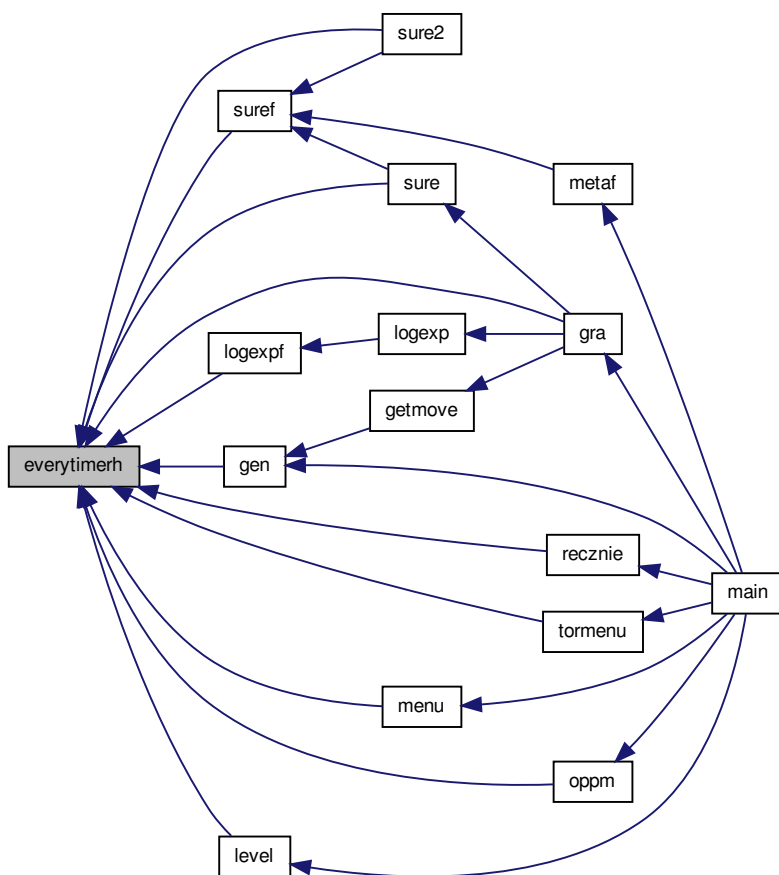
Funkcja wykonywana przy zmianie rozmiaru terminala niezależnie od aktualnego stanu programu.

Sprawdza, czy rozmiar terminala nie jest zbyt mały i inicjuje ekran od nowa. Jeśli terminal jest zbyt mały, kończy program.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

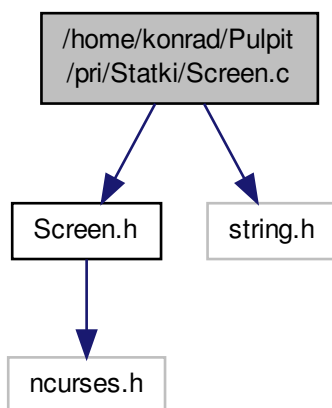


4.16 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Screen.c

Plik źródłowy obsługi ekranu.

```
#include "Screen.h"
#include <string.h>
```

Wykres zależności załączania dla Screen.c:



Funkcje

- void `init` ()
Funkcja inicjująca obsługę ekranu.
- void `myclear` ()
Funkcja czyszcząca ekran i wyświetlająca belki.
- void `printscr` (char *p1, char *p2)
Funkcja wyświetlająca pusty ekran z zadanymi przez argument napisami ustawianymi w nagłówku i stopce.

Zmienne

- char * `naglowek`
Wskaźnik na napis, który ma być wyświetlony w górnej belce.
- char * `stopka`
Wskaźnik na napis, który ma być wyświetlony w dolnej belce.
- char * `mennag`

4.16.1 Opis szczegółowy

Plik źródłowy obsługi ekranu.

4.16.2 Dokumentacja funkcji

4.16.2.1 void `printscr` (char *, char *)

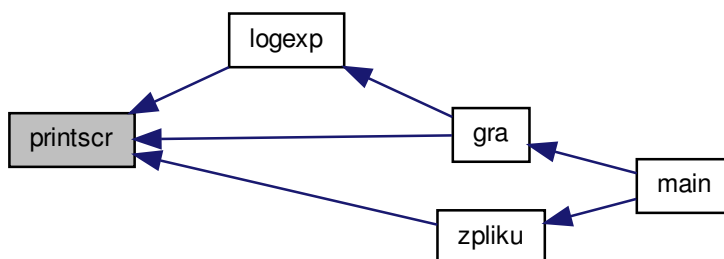
Funkcja wyświetlająca pusty ekran z zadanymi przez argument napisami ustawianymi w nagłówku i stopce.

Ekran znika gdy użytkownik wciśnie dowolny klawisz

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

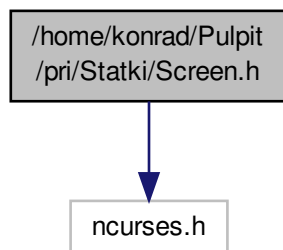


4.17 Dokumentacja pliku `/home/konrad/Pulpit/pri/Statki/Screen.h`

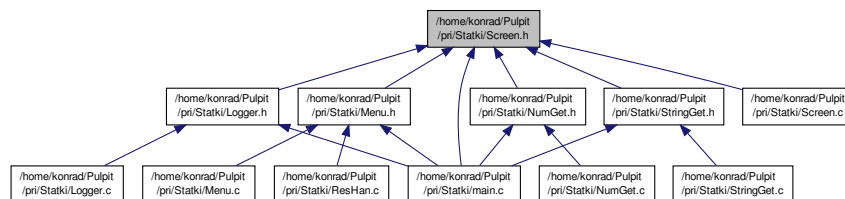
Plik nagłówkowy obsługi ekranu.

```
#include <ncurses.h>
```

Wykres zależności załączania dla `Screen.h`:



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



Funkcje

- void `init` ()

Funkcja inicjująca obsługę ekranu.

- void `myclear` ()

Funkcja czyszcząca ekran i wyświetlająca belki.

- void `printscr` (char *, char *)

Funkcja wyświetlająca pusty ekran z zadanymi przez argument napisami ustawianymi w nagłówku i stopce.

4.17.1 Opis szczegółowy

Plik nagłówkowy obsługi ekranu.

4.17.2 Dokumentacja funkcji

4.17.2.1 void `printscr` (char *, char *)

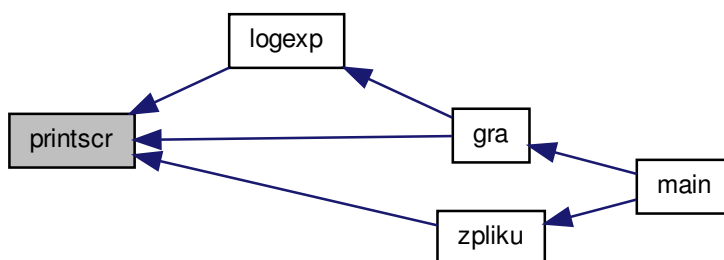
Funkcja wyświetlająca pusty ekran z zadanymi przez argument napisami ustawianymi w nagłówku i stopce.

Ekran znika gdy użytkownik wciśnie dowolny klawisz

Oto graf wywołań dla tej funkcji:



Oto graf wywołań tej funkcji:

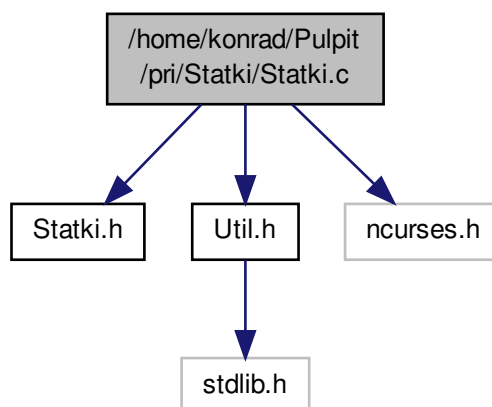


4.18 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Statki.c

Plik źródłowy obsługi planszy.

```
#include "Statki.h"
#include "Util.h"
#include <ncurses.h>
```

Wykres zależności załączania dla Statki.c:



Funkcje

- int **nowystatek** (**Statek** *s, int l, int k, int len, **orient** o)
Funkcja inicjująca statek.
- int **nowaplansza** (**Plansza** *p)
Funkcja inicjująca planszę
- int **wrzucnaplansze** (**Statek** *s, **Plansza** *p)
Funkcja wstawiająca statek na planszę

- int `cofnijzplanszy` (`Plansza *p`)

Funkcja usuwająca z planszy ostatni wstawiony na nią statek.

- int `strzał` (`Plansza *p`, int `l`, int `c`)

Funkcja realizująca strzał w dane pole na planszy.

- int `czykoniec` (`Plansza *p`)

Funkcja zwracająca wartość logiczną, czy na danej planszy gra się już skończyła (nie ma już segmentów statków do zestrzelenia)

- int `generujplansze` (`Plansza *p`, int `seed`)

Funkcja generująca planszę na podstawie podanej wartości.

Zmienne

- int `brdsize`

- int `brdt`

Wybrany typ planszy.

4.18.1 Opis szczegółowy

Plik źródłowy obsługi planszy.

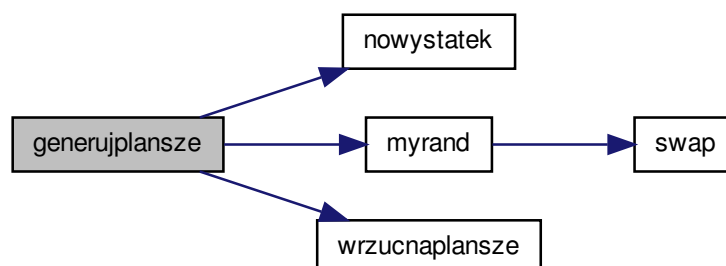
4.18.2 Dokumentacja funkcji

4.18.2.1 int generujplansze (Plansza * p, int seed)

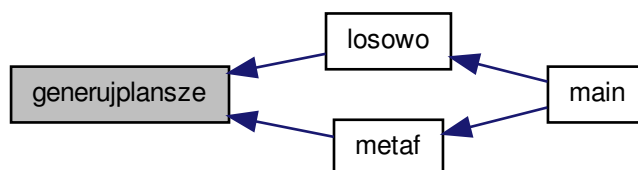
Funkcja generująca planszę na podstawie podanej wartości.

Funkcja ustawia seed do generatora liczb pseudolosowych jako liczbę podaną w parametrze. Następnie, dla każdej długości statku (w kolejności: 4-masztowiec, 2 3-masztowce, 3 2-masztowce, 4 1-masztowce) losuje jego orientację, później pozycję na planszy i wykonuje te czynności dopóki funkcja wstawiająca statek na planszę nie potwierdzi jego wstawienia (dopóki miejsce, na które statek miał zostać wstawiony, będzie zajęte).

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

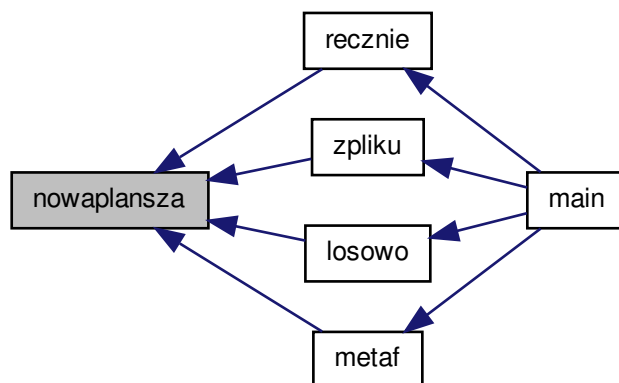


4.18.2.2 `int nowapiansza (Plansza * p)`

Funkcja inicjująca planszę

Funkcja ustawia stan wszystkich pól na puste.

Oto graf wywoływań tej funkcji:



4.18.2.3 `int strzal (Plansza * p, int l, int c)`

Funkcja realizująca strzał w dane pole na planszy.

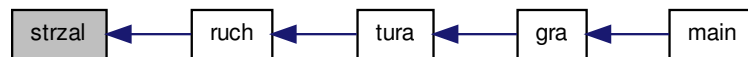
Funkcja zwraca:

- -1, gdy strzał nastąpił w miejsce, które już było ostrzeliwane
- 0, gdy w puste
- 1, gdy trafia w segment
- 2, gdy trafia w segment i zatapia statek
- 3, gdy po wykonaniu strzału na planszy nie został żaden segment

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



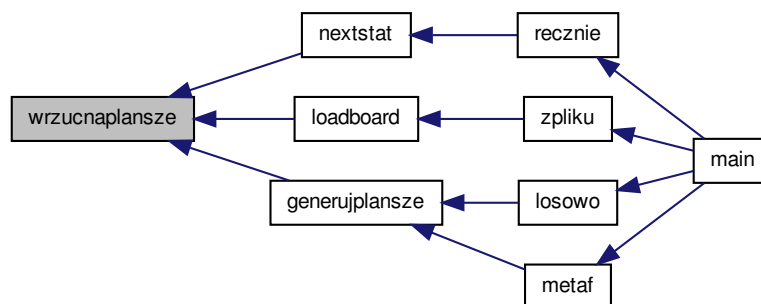
4.18.2.4 int wrzucnaplansze (Statek * s, Plansza * p)

Funkcja wstawiająca statek na planszę

Funkcja zakłada, że podany przez argument statek mieści się na planszy. Funkcja sprawdza, czy dany statek można ustawić w danym miejscu - sprawdza, czy na polach zajmowanych przez niego i sąsiadujących do nich (również po skosie) nie stoi żaden segment statku. Funkcja zwraca:

- -1, jeśli statek nie może zostać wstawiony na planszę
- 0, gdy statek spełnia warunki wstawienia na planszę (oraz wstawia statek na planszę)

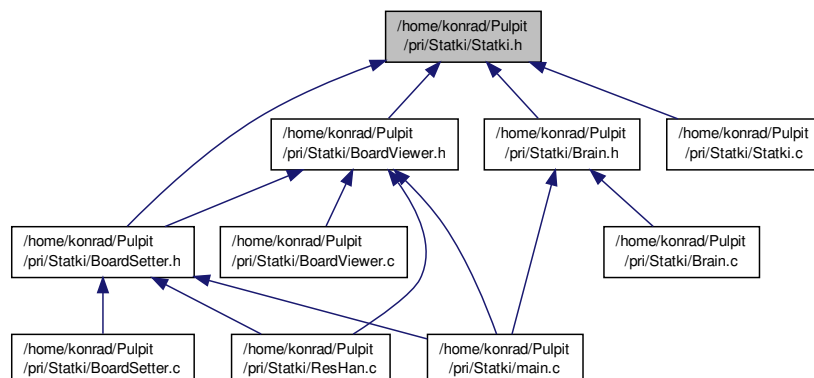
Oto graf wywoływań tej funkcji:



4.19 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Statki.h

Plik nagłówkowy obsługi planszy.

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



Komponenty

- struct **Statek**
Struktura reprezentująca statek (używana podczas ustawiania planszy)
- struct **Plansza**
Struktura reprezentująca planszę

Wyliczenia

- enum **orient** { **VERT** =0, **HOR** =1 }
Enumeracja reprezentująca orientację statku (pionowa/pozioma)
- enum **pole** { **NONE** =0, **STAT** =1, **NOCH** =2, **STCH** =3 }
Enumeracja reprezentująca stan pola planszy (puste / segment statku / puste sprawdzone / zniszczony segment statku)
- enum **VIEWT** { **TORUS**, **KWADRAT** }
Enumeracja oznaczająca tryb wyświetlania planszy, w zależności od tego, czy rozgrywka odbywa się na zwykłej planszy, czy na torusie.

Funkcje

- int **nowystatek** (**Statek** *, int, int, int, **orient**)
Funkcja inicjująca statek.
- int **nowaplansza** (**Plansza** *)
Funkcja inicjująca planszę
- int **wrzucnplansze** (**Statek** *, **Plansza** *)
Funkcja wstawiająca statek na planszę
- int **cofnijzplanszy** (**Plansza** *)
Funkcja usuwająca z planszy ostatni wstawiony na nią statek.
- int **strzał** (**Plansza** *, int, int)
Funkcja realizująca strzał w dane pole na planszy.
- int **czykoniec** (**Plansza** *)
Funkcja zwracająca wartość logiczną, czy na danej planszy gra się już skończyła (nie ma już segmentów statków do zestrzelenia)
- int **generujplansze** (**Plansza** *, int)
Funkcja generująca planszę na podstawie podanej wartości.

4.19.1 Opis szczegółowy

Plik nagłówkowy obsługi planszy.

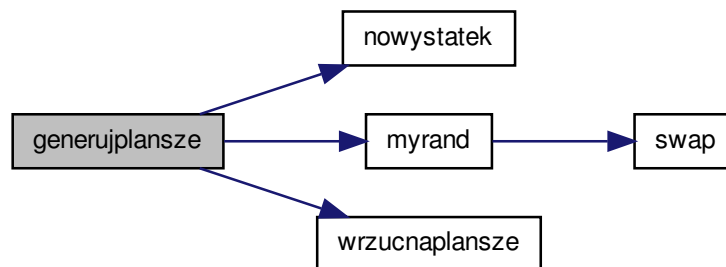
4.19.2 Dokumentacja funkcji

4.19.2.1 `int generujplansze (Plansza * p, int seed)`

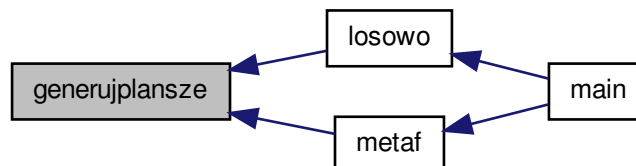
Funkcja generująca planszę na podstawie podanej wartości.

Funkcja ustawia seed do generatora liczb pseudolosowych jako liczbę podaną w parametrze. Następnie, dla każdej długości statku (w kolejności: 4-masztowiec, 2 3-masztowce, 3 2-masztowce, 4 1-masztowce) losuje jego orientację, później pozycję na planszy i wykonuje te czynności dopóki funkcja wstawiająca statek na planszę nie potwierdzi jego wstawienia (dopóki miejsce, na które statek miał zostać wstawiony, będzie zajęte).

Oto graf wywołań dla tej funkcji:



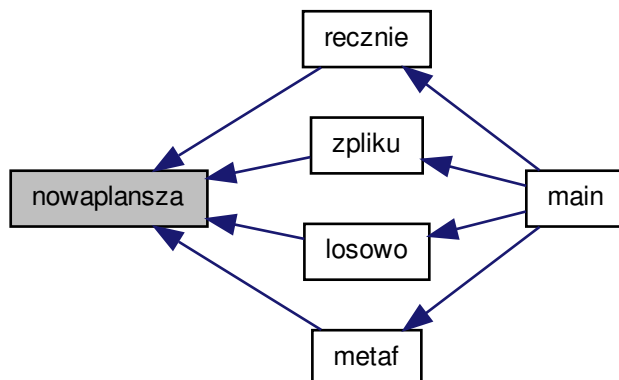
Oto graf wywoływań tej funkcji:

4.19.2.2 `int nowaplansza (Plansza * p)`

Funkcja inicjująca planszę

Funkcja ustawia stan wszystkich pól na puste.

Oto graf wywołań tej funkcji:



4.19.2.3 int strzal (Plansza * p, int l, int c)

Funkcja realizująca strzał w dane pole na planszy.

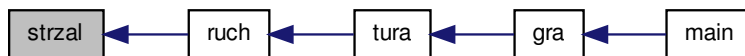
Funkcja zwraca:

- -1, gdy strzał nastąpił w miejsce, które już było ostrzelywane
- 0, gdy w puste
- 1, gdy trafia w segment
- 2, gdy trafia w segment i zatapia statek
- 3, gdy po wykonaniu strzału na planszy nie został żaden segment

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



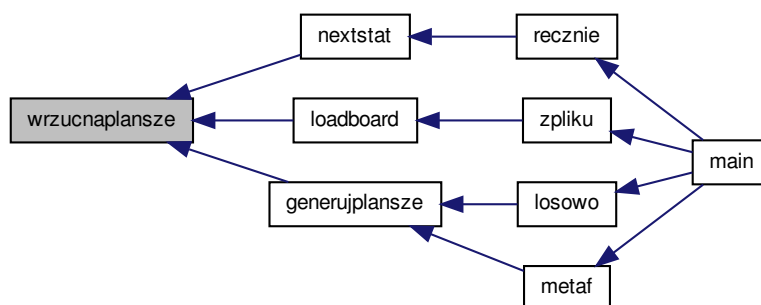
4.19.2.4 int wrzucnapiansze (Statek * s, Plansza * p)

Funkcja wstawiająca statek na planszę

Funkcja zakłada, że podany przez argument statek mieści się na planszy. Funkcja sprawdza, czy dany statek można ustawić w danym miejscu - sprawdza, czy na polach zajmowanych przez niego i sąsiadujących do nich (również po skosie) nie stoi żaden segment statku. Funkcja zwraca:

- -1, jeśli statek nie może zostać wstawiony na planszę
- 0, gdy statek spełnia warunki wstawienia na planszę (oraz wstawia statek na planszę)

Oto graf wywoływań tej funkcji:

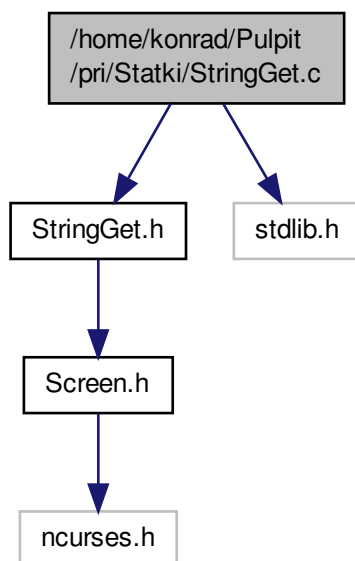


4.20 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/StringGet.c

Plik źródłowy obsługi pobieracza łańcucha znakowego.

```
#include "StringGet.h"
#include <stdlib.h>
```

Wykres zależności załączania dla StringGet.c:



Funkcje

- void `newsg (StringGet *sg, int maxlen, int w)`
Funkcja inicjująca nowy pobieracz znaków.
- void `keyhandler (StringGet *sg, chtype c)`
Funkcja obsługująca wciśnięcia klawiszy.
- void `drawsg (StringGet *sg)`
Funkcja rysująca pobieracz.
- void `updatesg (StringGet *sg)`
Funkcja aktualizująca pobieracz.

4.20.1 Opis szczegółowy

Plik źródłowy obsługi pobieracza łańcucha znakowego.

4.20.2 Dokumentacja funkcji

4.20.2.1 void `updatesg (StringGet * sg)`

Funkcja aktualizująca pobieracz.

Funkcja jest wywoływana, gdy zmieni się aktualny wybór w danym menu lub gdy menu jest przerysowywane. Stworzona, aby nie przerysowywać całego menu za każdym razem, gdy użytkownik zmieni aktualnie wybraną opcję

Oto graf wywoływań tej funkcji:

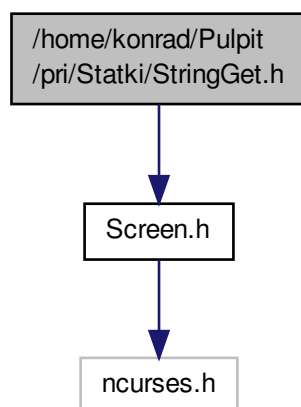


4.21 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/StringGet.h

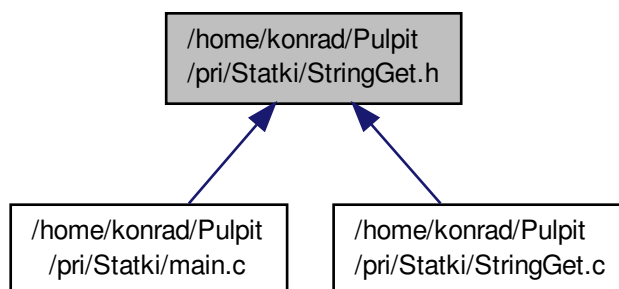
Plik nagłówkowy obsługi pobieracza łańcucha znakowego.

```
#include "Screen.h"
```

Wykres zależności załączania dla StringGet.h:



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



Komponenty

- struct [StringGet](#)

Struktura reprezentująca pobieracz łańcucha znakowego.

Funkcje

- void [newsg](#) ([StringGet](#) *, int, int)

Funkcja inicjująca nowy pobieracz znaków.

- void [keyhandler](#) ([StringGet](#) *, chtype)

Funkcja obsługująca wciśnięcia klawiszy.

- void [drawsg](#) ([StringGet](#) *)

Funkcja rysująca pobieracz.

- void [updatesg](#) ([StringGet](#) *)

Funkcja aktualizująca pobieracz.

4.21.1 Opis szczegółowy

Plik nagłówkowy obsługi pobieracza łańcucha znakowego.

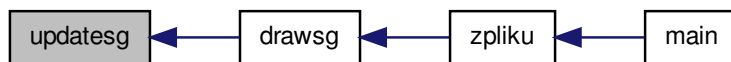
4.21.2 Dokumentacja funkcji

4.21.2.1 void [updatesg](#) ([StringGet](#) * *sg*)

Funkcja aktualizująca pobieracz.

Funkcja jest wywoływana, gdy zmieni się aktualny wybór w danym menu lub gdy menu jest przerysowywane. Stworzona, aby nie przerysowywać całego menu za każdym razem, gdy użytkownik zmieni aktualnie wybraną opcję

Oto graf wywoływań tej funkcji:

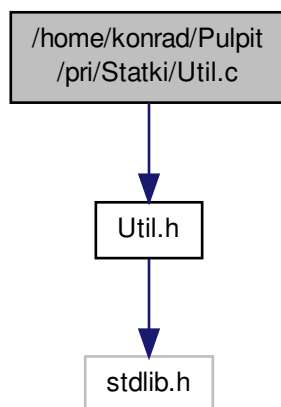


4.22 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Util.c

Plik źródłowy funkcji pomocniczych.

```
#include "Util.h"
```

Wykres zależności załączania dla Util.c:



Funkcje

- void **swap** (int *a, int *b)
Funkcja zamieniająca zawartość miejsc w pamięci wskazywanych przez wskaźniki przekazane w parametrach.
- int **myrand** (int a, int b)
Funkcja generująca pseudolosową liczbę z podanego w parametrach przedziału zamkniętego.

4.22.1 Opis szczegółowy

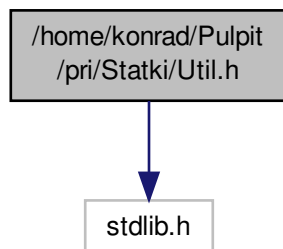
Plik źródłowy funkcji pomocniczych.

4.23 Dokumentacja pliku /home/konrad/Pulpit/pri/Statki/Util.h

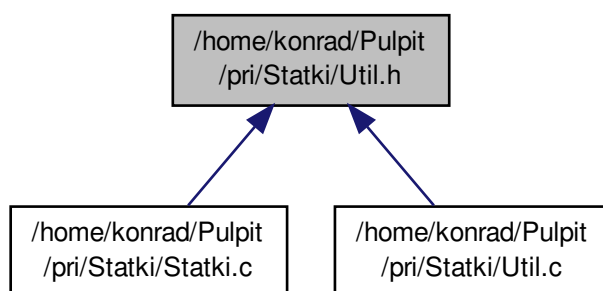
Plik nagłówkowy funkcji pomocniczych.

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

Wykres zależności załączania dla Util.h:



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



Funkcje

- void `swap` (int *, int *)
Funkcja zamieniająca zawartość miejsc w pamięci wskazywanych przez wskaźniki przekazane w parametrach.
- int `myrand` (int, int)
Funkcja generująca pseudolosową liczbę z podanego w parametrach przedziału zamkniętego.

4.23.1 Opis szczegółowy

Plik nagłówkowy funkcji pomocniczych.

Index

/home/konrad/Pulpit/pri/Statki/BoardSetter.c, [11](#)
/home/konrad/Pulpit/pri/Statki/BoardSetter.h, [15](#)
/home/konrad/Pulpit/pri/Statki/BoardViewer.c, [18](#)
/home/konrad/Pulpit/pri/Statki/BoardViewer.h, [20](#)
/home/konrad/Pulpit/pri/Statki/Brain.c, [21](#)
/home/konrad/Pulpit/pri/Statki/Brain.h, [22](#)
/home/konrad/Pulpit/pri/Statki/Logger.c, [23](#)
/home/konrad/Pulpit/pri/Statki/Logger.h, [24](#)
/home/konrad/Pulpit/pri/Statki/Menu.c, [36](#)
/home/konrad/Pulpit/pri/Statki/Menu.h, [40](#)
/home/konrad/Pulpit/pri/Statki/NumGet.c, [45](#)
/home/konrad/Pulpit/pri/Statki/NumGet.h, [46](#)
/home/konrad/Pulpit/pri/Statki/ResHan.c, [48](#)
/home/konrad/Pulpit/pri/Statki/ResHan.h, [50](#)
/home/konrad/Pulpit/pri/Statki/Screen.c, [52](#)
/home/konrad/Pulpit/pri/Statki/Screen.h, [54](#)
/home/konrad/Pulpit/pri/Statki/Statki.c, [56](#)
/home/konrad/Pulpit/pri/Statki/Statki.h, [59](#)
/home/konrad/Pulpit/pri/Statki/StringGet.c, [63](#)
/home/konrad/Pulpit/pri/Statki/StringGet.h, [65](#)
/home/konrad/Pulpit/pri/Statki/Util.c, [67](#)
/home/konrad/Pulpit/pri/Statki/Util.h, [67](#)
/home/konrad/Pulpit/pri/Statki/main.c, [26](#)

BoardSetter, [3](#)
BoardSetter.c
 bsrot, [13](#)
 newbs, [13](#)
 nextstat, [13](#)
 prevstat, [14](#)
BoardSetter.h
 bsrot, [16](#)
 newbs, [16](#)
 nextstat, [17](#)
 prevstat, [18](#)
BoardViewer, [4](#)
Brain, [5](#)
bsrot
 BoardSetter.c, [13](#)
 BoardSetter.h, [16](#)

drawmenu
 Menu.c, [36](#)
 Menu.h, [41](#)

everytimerh
 ResHan.c, [49](#)
 ResHan.h, [51](#)

exportlog
 Logger.c, [24](#)
 Logger.h, [26](#)

generujplansze
 Statki.c, [57](#)
 Statki.h, [61](#)

loadboard
 main.c, [31](#)

LogMove, [7](#)
Logger, [6](#)
Logger.c
 exportlog, [24](#)
Logger.h
 exportlog, [26](#)
losowo
 main.c, [31](#)

main
 main.c, [32](#)
main.c
 loadboard, [31](#)
 losowo, [31](#)
 main, [32](#)
 metaf, [33](#)
 ruch, [34](#)
 tura, [35](#)
Menu, [7](#)
Menu.c
 drawmenu, [36](#)
 newmenu, [37](#)
 nextchoice, [38](#)
 prevchoice, [38](#)
 updatemenu, [39](#)
Menu.h
 drawmenu, [41](#)
 newmenu, [42](#)
 nextchoice, [43](#)
 prevchoice, [43](#)
 updatemenu, [44](#)
metaf
 main.c, [33](#)

newbs
 BoardSetter.c, [13](#)
 BoardSetter.h, [16](#)
newmenu
 Menu.c, [37](#)
 Menu.h, [42](#)
nextchoice
 Menu.c, [38](#)
 Menu.h, [43](#)
nextstat
 BoardSetter.c, [13](#)
 BoardSetter.h, [17](#)
nowaplansza
 Statki.c, [58](#)
 Statki.h, [61](#)
NumGet, [8](#)

Plansza, [9](#)
 tab, [9](#)
Pole, [10](#)
prevchoice

- Menu.c, [38](#)
- Menu.h, [43](#)
- prevstat
 - BoardSetter.c, [14](#)
 - BoardSetter.h, [18](#)
- printscr
 - Screen.c, [53](#)
 - Screen.h, [55](#)
- ResHan.c
 - everytimerh, [49](#)
- ResHan.h
 - everytimerh, [51](#)
- ruch
 - main.c, [34](#)
- Screen.c
 - printscr, [53](#)
- Screen.h
 - printscr, [55](#)
- Statek, [10](#)
- Statki.c
 - generujplansze, [57](#)
 - nowaplansza, [58](#)
 - strzal, [58](#)
 - wrzucnaplansze, [59](#)
- Statki.h
 - generujplansze, [61](#)
 - nowaplansza, [61](#)
 - strzal, [62](#)
 - wrzucnaplansze, [63](#)
- StringGet, [10](#)
- StringGet.c
 - updatesg, [64](#)
- StringGet.h
 - updatesg, [66](#)
- strzal
 - Statki.c, [58](#)
 - Statki.h, [62](#)
- tab
 - Plansza, [9](#)
- tura
 - main.c, [35](#)
- updatemenu
 - Menu.c, [39](#)
 - Menu.h, [44](#)
- updatesg
 - StringGet.c, [64](#)
 - StringGet.h, [66](#)
- wrzucnaplansze
 - Statki.c, [59](#)
 - Statki.h, [63](#)