

Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki

Programowanie komputerów

Gra w statki

Autor	Maciej Kudela
Prowadzący	Mgr inż. Grzegorz Kwiatkowski
Rok akademicki	2019/2020
Kierunek	Informatyka
Rodzaj studiów	SSI
Semestr	2
Termin oddania sprawozdania	2020-09-10

1 Treść zadania

Należy napisać program, który będzie umożliwiał swobodną grę w statki dla dwóch osób. Gra będzie rozpoczynała się od wyboru pól startowych statków każdego z graczy. Następnie rozpocznie się gra, której celem będzie zniszczenie wszystkich statków przeciwnika. Jeśli gracz trafi w statek przeciwnika, jego tura zostanie przedłużona o jeden ruch, natomiast jeśli spudłuje, następuje tura przeciwnika.

2 Analiza zadania

Najważniejszym aspektem tego zadania jest przyporządkowanie każdemu z graczy planszy, na której umieści swoje statki oraz zintegrowanie jej z aktualnym życiem każdego statku. Każda plansza została zinterpretowana jako tablica struktur (statków). Każdy statek składa się z dwóch masztów, z czego każdy maszt jest na osobnym polu, które musi znajdować się zaraz obok poprzedniego masztu tak, aby oba maszty (cały statek) leżały bezpośrednio obok siebie. Gdy rozpoczyna się gra wszystkie statki posiadają wartość `statek.life = 1`, natomiast jeśli dany maszt zostanie zestrzelony przez przeciwnika wartość życia zmienia się na 0.

3 Instrukcja

Po włączeniu programu należy rozpocząć od wybrania nazw dla obu graczy. Nazwy graczy nie mogą być identyczne. Następnie rozpocznie się procedura dodawania statków dla gracza 1. Statki dodaje się w sposób następujący:

- Wybierz jaki rodzaj statku chcesz dodać (1 = poziomy, 2 = pionowy)

```
Dodawanie statkow gracz 1:

  1  2  3  4  5  6  7  8  9 10
1 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
2 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
3 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
4 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
5 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
6 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
7 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
8 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
9 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
10 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
Jaki statek dodajesz?
Poziomy[1] / Pionowy[2]: _
```

- Następnie należy wybrać kolumnę, w której chcemy, aby nasz statek się znalazł

```
Dodawanie statkow gracz 1:

  1  2  3  4  5  6  7  8  9 10
1 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
2 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
3 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
4 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
5 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
6 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
7 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
8 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
9 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
10 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
Podaj kolumne(1-10): _
```

- Po kolumnie wybieramy wiersz (kolejność może się zmienić w zależności od wybrania poziomego lub pionowego statku)

```
Dodawanie statkow gracz 1:

      1  2  3  4  5  6  7  8  9 10
1  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
2  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
3  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
4  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
5  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
6  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
7  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
8  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
9  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
10 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
Podaj kolumne(1-10): 5
Podaj wiersz(1-10):
```

- Następnie wybieramy pozycję drugiego masztu w naszym statku, w tym momencie pozostają jedynie dwie opcje do wyboru

```
Dodawanie statkow gracz 1:

      1  2  3  4  5  6  7  8  9 10
1  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
2  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
3  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
4  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
5  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
6  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
7  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
8  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
9  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
10 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
Podaj wiersz(4 lub 6): 4
```

- Po wybraniu pozycji dwóch masztów nasz statek zostaje dodany i naszym oczom ukazuje się „martwe pole”, na którym już żaden statek nie może stanąć. Analogicznie wybieramy

pozycje następnych czterech statków.

```
Dodawanie statkow gracz 1:

      1  2  3  4  5  6  7  8  9  10
1  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
2  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
3  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
4  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
5  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
6  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
7  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
8  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
9  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
10 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
Jaki statek dodajesz?
Poziomy[1] / Pionowy[2]:  _
```

- Gdy wybierzemy wszystkie statki przez chwilę pokaże się nam plansza z pozycjami naszych statków.

```
      1  2  3  4  5  6  7  8  9  10
1  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
2  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
3  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
4  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
5  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
6  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
7  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
8  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
9  [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
10 [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
_
```

- Po wybraniu pozycji wszystkich statków przez obu graczy gra się rozpoczyna.

```

Gra rozpoczeta!
Ruch gracza 1!
Twoja plansza:
  1  2  3  4  5  6  7  8  9 10
1  [+][ ][ ][ ][+][ ][ ][ ][+]
2  [+][ ][ ][ ][+][ ][ ][ ][+]
3  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
4  [+][+][ ][ ][ ][ ][ ][ ][ ]
5  [ ][ ][ ][ ][+][ ][ ][ ][ ]
6  [ ][ ][ ][ ][+][ ][ ][ ][ ]
7  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
8  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
9  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
10 [ ][ ][ ][ ][ ][ ][ ][ ][ ]

  Twoja plansza strzalow:
    1  2  3  4  5  6  7  8  9 10
1  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
2  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
3  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
4  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
5  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
6  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
7  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
8  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
9  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
10 [ ][ ][ ][ ][ ][ ][ ][ ][ ]

Podaj wiersz(1-10):

```

- Po trafionym strzale otrzymujemy powiadomienie oraz możemy kontynuować ostrzał wroga

```

Gra rozpoczeta!
Ruch gracza 1!
Twoja plansza:
  1  2  3  4  5  6  7  8  9 10
1  [+][ ][ ][ ][+][ ][ ][ ][+]
2  [+][ ][ ][ ][+][ ][ ][ ][+]
3  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
4  [+][+][ ][ ][ ][ ][ ][ ][ ]
5  [ ][ ][ ][ ][+][ ][ ][ ][ ]
6  [ ][ ][ ][ ][+][ ][ ][ ][ ]
7  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
8  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
9  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
10 [ ][ ][ ][ ][ ][ ][ ][ ][ ]

  Twoja plansza strzalow:
    1  2  3  4  5  6  7  8  9 10
1  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
2  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
3  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
4  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
5  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
6  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
7  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
8  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
9  [ ][ ][ ][ ][ ][ ][ ][ ][ ]
10 [ ][ ][ ][ ][ ][ ][ ][ ][ ]


Podaj wiersz(1-10): 1
Podaj kolumne(1-10): 1
Trafiony!

```

- Gra toczy się, dopóki wszystkie statki jednego z graczy nie zostaną zniszczone

```
Gra zakończona!  
Zwycięza gracz 1!
```

- Po zakończeniu gry możemy sprawdzić tabelę wyników oraz czas zakończenia poprzednich rozgrywek w pliku wyniki.txt

 wyniki — Notatnik

Plik Edycja Format Widok Pomoc

```
Czas zakończenia gry = Thu Sep 10 15:31:44 2020  
zwycięza gracz: maciek
```

```
Czas zakończenia gry = Thu Sep 10 15:33:48 2020  
zwycięza gracz: marcin
```

4 Struktury danych

W programie wykorzystano listy dwukierunkowe, które przechowują wskaźniki na maszty danego statku.

5 Ogólna struktura programu

W funkcji głównej tworzone są najpierw plansze dla obu graczy, oraz tablice przechowujące wskaźniki na ich statki. Następnie dla obu graczy wywołana zostaje funkcja *dodaj_statek*, która umożliwia graczom dodawanie statków na ich plansze. Po dodaniu

wszystkich statków wywołana zostaje funkcja *gra*. Po zakończeniu gry wszystkie listy zostają usunięte przy pomocy funkcji *kasowanie_statków*, by uniknąć wycieków pamięci w programie. Na końcu funkcja *zapis_wyniku* otwiera plik *wyniki.txt* (lub tworzy go, jeśli takowy nie istnieje) oraz dopisuje na jego końcu wyniki z danej gry.

6 Testowanie

Program został przetestowany pod wieloma względami. Podanie nieprawidłowego pola podczas dodawania statku kończy się pojawieniem się błędu: „*Nieprawidłowy wiersz!*” lub „*Nieprawidłowa kolumna!*”. Tak samo podczas oddawania strzału, jeśli podamy nieprawidłowe pole. Program został również sprawdzony pod względem wycieków pamięci. Aby ich uniknąć stworzona została funkcja *kasowanie_statkow*, która usuwa wszystkie elementy list dwukierunkowych. Jeśli wystąpi błąd podczas otwarcia pliku, (na przykład z powodu niewystarczających uprawnień) program zwróci błąd -1, oraz wyświetli komunikat: „*Bład otwarcia pliku*”.

7 Wnioski

Program umożliwiający swobodną grę w statki dla dwóch osób okazał się być dla mnie przeszkodą średniej wielkości. Największe trudności sprawiło prawidłowe tworzenie list dwukierunkowych oraz podpięcie ich pod plansze graczy. Trudnym zadaniem okazało się również stworzenie „martwego pola”, to znaczy pola, w którym nie można było już postawić nowego statku podczas wybierania ich pozycji, jako iż statki nie mogą stać bezpośrednio obok siebie. Wykonanie tego projektu wielokrotnie wskazało mi moje braki i dzięki temu byłem w stanie wiele się nauczyć podczas jego tworzenia.

Projekt PK - gra w statki

Generated by Doxygen 1.8.17

1 Data Structure Index	1
1.1 Data Structures	1
2 File Index	3
2.1 File List	3
3 Data Structure Documentation	5
3.1 statek Struct Reference	5
3.1.1 Detailed Description	5
3.1.2 Field Documentation	5
3.1.2.1 blind	5
3.1.2.2 kolumna	6
3.1.2.3 life	6
3.1.2.4 pNext	6
3.1.2.5 pPrev	6
3.1.2.6 wiersz	6
4 File Documentation	7
4.1 C:/Users/macku/Desktop/Statki 04.09/EI Shipo/EI Shipo/funkcje.h File Reference	7
4.1.1 Function Documentation	8
4.1.1.1 alokacja_masztu()	8
4.1.1.2 dodaj_na_plansze()	8
4.1.1.3 dodaj_statek()	9
4.1.1.4 dodaj_statek_pionowy()	9
4.1.1.5 dodaj_statek_poziomy()	9
4.1.1.6 gra()	10
4.1.1.7 kasowanie_statkow()	10
4.1.1.8 pobierz_kolumne_check()	10
4.1.1.9 pobierz_kolumne_clean()	11
4.1.1.10 pobierz_kolumne_poziomy()	11
4.1.1.11 pobierz_wiersz_check()	11
4.1.1.12 pobierz_wiersz_clean()	11
4.1.1.13 pobierz_wiersz_pionowy()	12
4.1.1.14 ruch_gracza()	12
4.1.1.15 skoryguj()	12
4.1.1.16 sprawdz_statek()	14
4.1.1.17 sprawdz_wszystkie_statki()	14
4.1.1.18 umiesc_martwe_pole_pionowe()	14
4.1.1.19 umiesc_martwe_pole_poziome()	15
4.1.1.20 wypelnij_plansze()	15
4.1.1.21 wyswietl_plansze_game()	15
4.1.1.22 wyswietl_plansze_make()	16
4.1.1.23 wyswietl_plansze_strzalow()	16

4.2 C:/Users/macku/Desktop/Statki 04.09/El Shipo/El Shipo/main.c File Reference	16
4.2.1 Function Documentation	17
4.2.1.1 main()	17

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

statek	5
----------------------------------	---

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/El Shipo/El Shipo/ dodawanie-statku.c	??
C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/El Shipo/El Shipo/ dodawanie-statku.h	??
C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/El Shipo/El Shipo/ main.c	16
C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/El Shipo/El Shipo/ manipulacja-wierszami.c	??
C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/El Shipo/El Shipo/ manipulacja-wierszami.h	??
C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/El Shipo/El Shipo/ Operacje-na-planszach.c	??
C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/El Shipo/El Shipo/ Operacje-na-planszach.h	??
C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/El Shipo/El Shipo/ operacje-na-plikach.c	??
C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/El Shipo/El Shipo/ operacje-na-plikach.h	??
C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/El Shipo/El Shipo/ struktura-statek.h	??
C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/El Shipo/El Shipo/ tok-gry.c	??
C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/El Shipo/El Shipo/ tok-gry.h	??

Chapter 3

Data Structure Documentation

3.1 statek Struct Reference

```
#include <struktura-statek.h>
```

Data Fields

- int [life](#)
- int [blind](#)
- int [wiersz](#)
- int [kolumna](#)
- struct [statek](#) * [pPrev](#)
- struct [statek](#) * [pNext](#)

3.1.1 Detailed Description

podstawowa struktura przedstawiajaca statek, posiada wskaznik zycia, wiersz i kolumne w ktorej sie znajduje oraz wskazniki na poprzedni i nastepny maszt

3.1.2 Field Documentation

3.1.2.1 blind

```
int blind
```

wartosc blindspota, wynosi 1 jesli statek zajmuje pole na planszy

3.1.2.2 kolumna

```
int kolumna
```

przechowuje kolumnę w której znajduje się statek

3.1.2.3 life

```
int life
```

wskaznik życia, jeśli wynosi 0 oznacza to że maszt został zestrzelony

3.1.2.4 pNext

```
struct statek* pNext
```

wskaznik na następny maszt w statku

3.1.2.5 pPrev

```
struct statek* pPrev
```

wskaznik na poprzedni maszt w statku

3.1.2.6 wiersz

```
int wiersz
```

przechowuje wiersz w którym znajduje się statek

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

- C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt P↔
K/EI Shipo/EI Shipo/[struktura-statek.h](#)

Chapter 4

File Documentation

4.1 C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/El Shipo/El Shipo/dodawanie-statku.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <windows.h>
#include "dodawanie-statku.h"
#include "struktura-statek.h"
```

Functions

- struct [statek](#) * [dodaj_statek](#) (int N, struct [statek](#) plansza[10][10], char *gracz)
- struct [statek](#) * [dodaj_statek_pionowy](#) (int N, struct [statek](#) plansza[10][10], char *gracz)
- struct [statek](#) * [dodaj_statek_poziomy](#) (int N, struct [statek](#) plansza[10][10], char *gracz)
- struct [statek](#) * [alokacja_masztu](#) (int wiersz, int kolumna, int i, struct [statek](#) *poprzedni)
- void [skoryguj](#) (struct [statek](#) *statki_przeciwnika[5], int wiersz, int kolumna)

4.1.1 Function Documentation

4.1.1.1 alokacja_masztu()

```
struct statek* alokacja_masztu (
    int wiersz,
    int kolumna,
    int i,
    struct statek * poprzedni )
```

Funkcja alokuje pamiec oraz tworzy liste dwukierunkowa

Parameters

<i>wiersz</i>	wiersz, w którym alokowany jest maszt
<i>kolumna</i>	kolumna, w której alokowany jest maszt
<i>i</i>	licznik potrzebny do odroznienia pierwszego masztu od reszty
<i>poprzedni</i>	wskaznik na poprzedni maszt w danym statku

4.1.1.2 dodaj_statek()

```
struct statek* dodaj_statek (
    int N,
    struct statek plansza[10][10],
    char * gracz )
```

Funkcja dodaje nowy statek

Parameters

<i>N</i>	wskazuje ilosc masztow, jaka posiada dodawany statek
<i>plansza</i>	plansza ze statkami danego gracza
<i>gracz</i>	wskazuje ktory gracz aktualnie dodaje statki

4.1.1.3 dodaj_statek_pionowy()

```
struct statek* dodaj_statek_pionowy (
    int N,
    struct statek plansza[10][10],
    char * gracz )
```

Funkcja dodaje statek pionowy

Parameters

<i>N</i>	ilosc masztow w danym statku
<i>plansza</i>	plansza ze statkami danego gracza
<i>gracz</i>	wskazuje ktory gracz aktualnie dodaje statki

4.1.1.4 dodaj_statek_poziomy()

```
struct statek* dodaj_statek_poziomy (
    int N,
```

```
struct statek plansza[10][10],
char * gracz )
```

Funkcja dodaje statek poziomy

Parameters

<i>N</i>	ilosc masztow w danym statku
<i>plansza</i>	plansza ze statkami danego gracza
<i>gracz</i>	wskazuje ktory gracz aktualnie dodaje statki

4.1.1.5 skoryguj()

```
void skoryguj (
    struct statek * statki_przeciwnika[5],
    int wiersz,
    int kolumna )
```

Funkcja koryguje dane miedzy plansza gracza oraz tablica jego statkow

Parameters

<i>statki_przeciwnika</i>	plansza ze statkami przeciwnika
<i>wiersz</i>	wiersz masztu do korekty
<i>kolumna</i>	kolumna masztu do korekty

4.2 C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/EI Shipo/EI Shipo/dodawanie-statku.h File Reference

Functions

- struct [statek](#) * [dodaj_statek](#) (int N, struct [statek](#) plansza[10][10], char *gracz)
- struct [statek](#) * [dodaj_statek_pionowy](#) (int N, struct [statek](#) plansza[10][10], char *gracz)
- struct [statek](#) * [dodaj_statek_poziomy](#) (int N, struct [statek](#) plansza[10][10], char *gracz)
- struct [statek](#) * [alokacja_masztu](#) (int wiersz, int kolumna, int i, struct [statek](#) *poprzedni)
- void [skoryguj](#) (struct [statek](#) *statki_przeciwnika[5], int wiersz, int kolumna)

4.2.1 Function Documentation

4.2.1.1 alokacja_masztu()

```
struct statek* alokacja_masztu (
    int wiersz,
    int kolumna,
    int i,
    struct statek * poprzedni )
```

Funkcja alokuje pamiec oraz tworzy liste dwukierunkowa

Parameters

<i>wiersz</i>	wiersz, w ktorym alokowany jest maszt
<i>kolumna</i>	kolumna, w ktorej alokowany jest maszt
<i>i</i>	licznik potrzebny do odroznienia pierwszego masztu od reszty
<i>poprzedni</i>	wskaznik na poprzedni maszt w danym statku

4.2.1.2 dodaj_statek()

```
struct statek* dodaj_statek (
    int N,
    struct statek plansza[10][10],
    char * gracz )
```

Funkcja dodaje nowy statek

Parameters

<i>N</i>	wskazuje ilosc masztow, jaka posiada dodawany statek
<i>plansza</i>	plansza ze statkami danego gracza
<i>gracz</i>	wskazuje ktory gracz aktualnie dodaje statki

4.2.1.3 dodaj_statek_pionowy()

```
struct statek* dodaj_statek_pionowy (
    int N,
    struct statek plansza[10][10],
    char * gracz )
```

Funkcja dodaje statek pionowy

Parameters

<i>N</i>	ilosc masztow w danym statku
<i>plansza</i>	plansza ze statkami danego gracza
<i>gracz</i>	wskazuje ktory gracz aktualnie dodaje statki

4.2.1.4 dodaj_statek_poziomy()

```
struct statek* dodaj_statek_poziomy (
    int N,
    struct statek plansza[10][10],
    char * gracz )
```

Funkcja dodaje statek poziomy

Parameters

<i>N</i>	ilosc masztow w danym statku
<i>plansza</i>	plansza ze statkami danego gracza
<i>gracz</i>	wskazuje ktory gracz aktualnie dodaje statki

4.2.1.5 skoryguj()

```
void skoryguj (
    struct statek * statki_przeciwnika[5],
    int wiersz,
    int kolumna )
```

Funkcja koryguje dane miedzy plansza gracza oraz tablica jego statkow

Parameters

<i>statki_przeciwnika</i>	plansza ze statkami przeciwnika
<i>wiersz</i>	wiersz masztu do korekty
<i>kolumna</i>	kolumna masztu do korekty

4.3 C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/El Shipo/El Shipo/main.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <Windows.h>
#include <time.h>
#include "Operacje-na-planszach.h"
#include "struktura-statek.h"
#include "dodawanie-statku.h"
#include "manipulacja-wierszami.h"
```



```
#include "tok-gry.h"
#include "operacje-na-plikach.h"
```

Functions

- int [main](#) ()

4.3.1 Function Documentation

4.3.1.1 main()

```
int main ( )
```

4.4 C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/El Shipo/El Shipo/manipulacja-wierszami.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <windows.h>
#include "manipulacja-wierszami.h"
#include "struktura-statek.h"
```

Functions

- int [pobierz_kolumne_clean](#) ()
Funkcja pobiera kolumne bez obostrzen.
- int [pobierz_wiersz_clean](#) ()
Funkcja pobiera wiersz bez obostrzen.
- int [pobierz_wiersz_check](#) (struct [statek](#) plansza[10][10], int kolumna)
- int [pobierz_kolumne_check](#) (struct [statek](#) plansza[10][10], int wiersz)
- int [pobierz_wiersz_pionowy](#) (struct [statek](#) plansza[10][10], int kolumna, int min, int max)
- int [pobierz_kolumne_poziomy](#) (struct [statek](#) plansza[10][10], int wiersz, int min, int max)

4.4.1 Function Documentation

4.4.1.1 pobierz_kolumne_check()

```
int pobierz_kolumne_check (
    struct statek plansza[10][10],
    int wiersz )
```

Funkcja pobiera kolumne i sprawdza czy dane miejsce jest wolne

Parameters

<i>plansza</i>	plansza ze statkami danego gracza
<i>wiersz</i>	wiersz w ktorym bedzie dodany statek

4.4.1.2 pobierz_kolumne_clean()

```
int pobierz_kolumne_clean ( )
```

Funkcja pobiera kolumne bez obostrzen.

4.4.1.3 pobierz_kolumne_poziomy()

```
int pobierz_kolumne_poziomy (
    struct statek plansza[10][10],
    int wiersz,
    int min,
    int max )
```

Funkcja pobiera kolumne podczas dalszej czesci dodawania poziomego statku

Parameters

<i>plansza</i>	plansza ze statkami danego gracza
<i>wiersz</i>	wiersz w ktorym bedzie dodany statek
<i>min</i>	wartosc najmniejsza potrzebna pozniej do wyznaczenia martwego pola
<i>max</i>	wartosc największa potrzebna pozniej do wyznaczenia martwego pola

4.4.1.4 pobierz_wiersz_check()

```
int pobierz_wiersz_check (
    struct statek plansza[10][10],
    int kolumna )
```

Funkcja pobiera wiersz i sprawdza czy dane miejsce jest wolne

Parameters

<i>plansza</i>	plansza ze statkami danego gracza
<i>kolumna</i>	kolumna w ktorej bedzie dodany statek

4.4.1.5 pobierz_wiersz_clean()

```
int pobierz_wiersz_clean ( )
```

Funkcja pobiera wiersz bez obostrzen.

4.4.1.6 pobierz_wiersz_pionowy()

```
int pobierz_wiersz_pionowy (
    struct statek plansza[10][10],
    int kolumna,
    int min,
    int max )
```

Funkcja pobiera wiersz podczas dalszej czesci dodawania pionowego statku

Parameters

<i>plansza</i>	plansza ze statkami danego gracza
<i>kolumna</i>	kolumna w ktorej bedzie dodany statek
<i>min</i>	wartosc najmniejsza potrzebna pozniej do wyznaczenia martwego pola
<i>max</i>	wartosc największa potrzebna pozniej do wyznaczenia martwego pola

4.5 C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/El Shipo/El Shipo/manipulacja-wierszami.h File Reference

Functions

- int [pobierz_kolumne_clean](#) ()
Funkcja pobiera kolumne bez obostrzen.
- int [pobierz_wiersz_clean](#) ()
Funkcja pobiera wiersz bez obostrzen.
- int [pobierz_wiersz_check](#) (struct [statek](#) plansza[10][10], int kolumna)
- int [pobierz_kolumne_check](#) (struct [statek](#) plansza[10][10], int wiersz)
- int [pobierz_wiersz_pionowy](#) (struct [statek](#) plansza[10][10], int kolumna, int min, int max)
- int [pobierz_kolumne_poziomy](#) (struct [statek](#) plansza[10][10], int wiersz, int min, int max)

4.5.1 Function Documentation

4.5.1.1 pobierz_kolumne_check()

```
int pobierz_kolumne_check (
    struct statek plansza[10][10],
    int wiersz )
```

Funkcja pobiera kolumne i sprawdza czy dane miejsce jest wolne

Parameters

<i>plansza</i>	plansza ze statkami danego gracza
<i>wiersz</i>	wiersz w ktorym bedzie dodany statek

4.5.1.2 pobierz_kolumne_clean()

```
int pobierz_kolumne_clean ( )
```

Funkcja pobiera kolumne bez obostrzen.

4.5.1.3 pobierz_kolumne_poziomy()

```
int pobierz_kolumne_poziomy (
    struct statek plansza[10][10],
    int wiersz,
    int min,
    int max )
```

Funkcja pobiera kolumne podczas dalszej czesci dodawania poziomego statku

Parameters

<i>plansza</i>	plansza ze statkami danego gracza
<i>wiersz</i>	wiersz w ktorym bedzie dodany statek
<i>min</i>	wartosc najmniejsza potrzebna pozniej do wyznaczenia martwego pola
<i>max</i>	wartosc największa potrzebna pozniej do wyznaczenia martwego pola

4.5.1.4 pobierz_wiersz_check()

```
int pobierz_wiersz_check (
    struct statek plansza[10][10],
    int kolumna )
```

Funkcja pobiera wiersz i sprawdza czy dane miejsce jest wolne

Parameters

<i>plansza</i>	plansza ze statkami danego gracza
<i>kolumna</i>	kolumna w ktorej bedzie dodany statek

4.5.1.5 pobierz_wiersz_clean()

```
int pobierz_wiersz_clean ( )
```

Funkcja pobiera wiersz bez obostrzen.

4.5.1.6 pobierz_wiersz_pionowy()

```
int pobierz_wiersz_pionowy (
    struct statek plansza[10][10],
    int kolumna,
    int min,
    int max )
```

Funkcja pobiera wiersz podczas dalszej czesci dodawania pionowego statku

Parameters

<i>plansza</i>	plansza ze statkami danego gracza
<i>kolumna</i>	kolumna w ktorej bedzie dodany statek
<i>min</i>	wartosc najmniejsza potrzebna pozniej do wyznaczenia martwego pola
<i>max</i>	wartosc największa potrzebna pozniej do wyznaczenia martwego pola

4.6 C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/El Shipo/El Shipo/Operacje-na-planszach.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <windows.h>
#include "Operacje-na-planszach.h"
#include "struktura-statek.h"
```

Functions

- void [wyswietl_plansze_make](#) (struct [statek](#) plansza[10][10], char *gracz)
- void [wyswietl_plansze_game](#) (struct [statek](#) plansza[10][10])
- void [wyswietl_plansze_strzalow](#) (int tab[10][10])
- void [dodaj_na_plansze](#) (struct [statek](#) *statki[5], struct [statek](#) plansza[10][10])
- void [wypelnij_plansze](#) (struct [statek](#) plansza[10][10])
- void [umiesc_martwe_pole_pionowe](#) (struct [statek](#) plansza[10][10], int min, int max, int kolumna)
- void [umiesc_martwe_pole_poziome](#) (struct [statek](#) plansza[10][10], int min, int max, int wiersz)

4.6.1 Function Documentation

4.6.1.1 dodaj_na_plansze()

```
void dodaj_na_plansze (
    struct statek * statki[5],
    struct statek plansza[10][10] )
```

Funkcja umieszcza statki na planszy

Parameters

<i>statki</i>	tablica zawierająca wskaźniki na wszystkie statki danego gracza
<i>plansza</i>	plansza ze statkami danego gracza

4.6.1.2 umieśc_martwe_pole_pionowe()

```
void umieśc_martwe_pole_pionowe (
    struct statek plansza[10][10],
    int min,
    int max,
    int kolumna )
```

Funkcja umieszcza pole, przez które na danym miejscu nie można postawić nowego statku

Parameters

<i>plansza</i>	plansza ze statkami danego gracza
<i>min</i>	wartość najmniejsza podczas dodawania martwego pola
<i>max</i>	wartość największa podczas dodawania martwego pola
<i>kolumna</i>	wskazuje kolumnę, w której będzie dodawane martwe pole

4.6.1.3 umieśc_martwe_pole_poziome()

```
void umieśc_martwe_pole_poziome (
    struct statek plansza[10][10],
    int min,
    int max,
    int wiersz )
```

Funkcja umieszcza pole, przez które na danym miejscu nie można postawić nowego statku

Parameters

<i>plansza</i>	plansza ze statkami danego gracza
<i>min</i>	wartosc najmniejsza podczas dodawania martwego pola
<i>max</i>	wartosc największa podczas dodawania martwego pola
<i>wiersz</i>	wskazuje wiersz w którym bedzie dodawane martwe pole

4.6.1.4 wypelnij_plansze()

```
void wypelnij_plansze (
    struct statek plansza[10][10] )
```

Funkcja wypelnia pusta plansze "blindspotami"

Parameters

<i>plansza</i>	plansza ze statkami danego gracza
----------------	-----------------------------------

4.6.1.5 wyswietl_plansze_game()

```
void wyswietl_plansze_game (
    struct statek plansza[10][10] )
```

Funkcja wyswietla plansze podczas gry

Parameters

<i>plansza</i>	plansza ze statkami danego gracza
----------------	-----------------------------------

4.6.1.6 wyswietl_plansze_make()

```
void wyswietl_plansze_make (
    struct statek plansza[10][10],
    char * gracz )
```

Funkcja wyswietla plansze podczas dodawania statkow

Parameters

<i>plansza</i>	plansza ze statkami danego gracza
<i>gracz</i>	wskazuje ktory gracz aktualnie dodaje statki

4.6.1.7 `wyswietl_plansze_strzalow()`

```
void wyswietl_plansze_strzalow (
    int tab[10][10] )
```

Funkcja wyświetla plansze poprzednich strzałów podczas gry

Parameters

<i>tab</i>	plansza z poprzednimi strzałami
------------	---------------------------------

4.7 C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/EI Shipo/EI Shipo/Operacje-na-planszach.h File Reference

Functions

- void [wyswietl_plansze_make](#) (struct [statek](#) plansza[10][10], char *gracz)
- void [wyswietl_plansze_game](#) (struct [statek](#) plansza[10][10])
- void [wyswietl_plansze_strzalow](#) (int tab[10][10])
- void [dodaj_na_plansze](#) (struct [statek](#) *statki[5], struct [statek](#) plansza[10][10])
- void [wypelnij_plansze](#) (struct [statek](#) plansza[10][10])
- void [umiesc_martwe_pole_pionowe](#) (struct [statek](#) plansza[10][10], int min, int max, int kolumna)
- void [umiesc_martwe_pole_poziome](#) (struct [statek](#) plansza[10][10], int min, int max, int wiersz)

4.7.1 Function Documentation

4.7.1.1 `dodaj_na_plansze()`

```
void dodaj_na_plansze (
    struct statek * statki[5],
    struct statek plansza[10][10] )
```

Funkcja umieszcza statki na planszy

Parameters

<i>statki</i>	tablica zawierająca wskaźniki na wszystkie statki danego gracza
<i>plansza</i>	plansza ze statkami danego gracza

4.7.1.2 umieśc_martwe_pole_pionowe()

```
void umieśc_martwe_pole_pionowe (
    struct statek plansza[10][10],
    int min,
    int max,
    int kolumna )
```

Funkcja umieszcza pole, przez które na danym miejscu nie można postawić nowego statku

Parameters

<i>plansza</i>	plansza ze statkami danego gracza
<i>min</i>	wartość najmniejsza podczas dodawania martwego pola
<i>max</i>	wartość największa podczas dodawania martwego pola
<i>kolumna</i>	wskazuje kolumnę, w której będzie dodawane martwe pole

4.7.1.3 umieśc_martwe_pole_poziome()

```
void umieśc_martwe_pole_poziome (
    struct statek plansza[10][10],
    int min,
    int max,
    int wiersz )
```

Funkcja umieszcza pole, przez które na danym miejscu nie można postawić nowego statku

Parameters

<i>plansza</i>	plansza ze statkami danego gracza
<i>min</i>	wartość najmniejsza podczas dodawania martwego pola
<i>max</i>	wartość największa podczas dodawania martwego pola
<i>wiersz</i>	wskazuje wiersz w którym będzie dodawane martwe pole

4.7.1.4 wypełnij_plansze()

```
void wypełnij_plansze (
    struct statek plansza[10][10] )
```

Funkcja wypełnia pustą planszę "blindspotami"

Parameters

<i>plansza</i>	plansza ze statkami danego gracza
----------------	-----------------------------------

4.7.1.5 wyswietl_plansze_game()

```
void wyswietl_plansze_game (
    struct statek plansza[10][10] )
```

Funkcja wyswietla plansze podczas gry

Parameters

<i>plansza</i>	plansza ze statkami danego gracza
----------------	-----------------------------------

4.7.1.6 wyswietl_plansze_make()

```
void wyswietl_plansze_make (
    struct statek plansza[10][10],
    char * gracz )
```

Funkcja wyswietla plansze podczas dodawania statkow

Parameters

<i>plansza</i>	plansza ze statkami danego gracza
<i>gracz</i>	wskazuje ktory gracz aktualnie dodaje statki

4.7.1.7 wyswietl_plansze_strzalow()

```
void wyswietl_plansze_strzalow (
    int tab[10][10] )
```

Funkcja wyswietla plansze poprzednich strzalow podczas gry

Parameters

<i>tab</i>	plansza z poprzednimi strzalami
------------	---------------------------------

4.8 C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/EI Shipo/EI Shipo/operacje-na-plikach.c File Reference

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

Functions

- void [zapis_wyniku](#) (char *gracz)

4.8.1 Function Documentation

4.8.1.1 zapis_wyniku()

```
void zapis_wyniku (
    char * gracz )
```

Funkcja zapisuje do pliku .txt dane dotyczące zakończonej rozgrywki, zapisuje zwycięzce oraz czas, w której rozgrywka się zakończyła

Parameters

<i>gracz</i>	przechowuje nazwe zwyciezcy
--------------	-----------------------------

4.9 C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/EI Shipo/EI Shipo/operacje-na-plikach.h File Reference

Functions

- void [zapis_wyniku](#) (char *gracz)

4.9.1 Function Documentation

4.9.1.1 zapis_wyniku()

```
void zapis_wyniku (
    char * gracz )
```

Funkcja zapisuje do pliku .txt dane dotyczące zakończonej rozgrywki, zapisuje zwycięzce oraz czas, w której rozgrywka się zakończyła

Parameters

<code>gracz</code>	przechowuje nazwe zwyciezcy
--------------------	-----------------------------

4.10 C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/EI Shipo/EI Shipo/struktura-statek.h File Reference

Data Structures

- struct [statek](#)

4.11 C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/EI Shipo/EI Shipo/tok-gry.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <windows.h>
#include "tok-gry.h"
#include "struktura-statek.h"
#include "operacje-na-plikach.h"
```

Functions

- bool [ruch_gracza](#) (struct [statek](#) plansza[10][10], struct [statek](#) plansza_przeciwnika[10][10], int strzaly[10][10], struct [statek](#) *statki_przeciwnika[5])
- bool [sprawdz_statek](#) (struct [statek](#) plansza_przeciwnika[10][10], int wiersz, int kolumna)
- bool [sprawdz_wszystkie_statki](#) (struct [statek](#) *statki[5])
- void [gra](#) (struct [statek](#) plansza_gracz1[10][10], struct [statek](#) plansza_gracz2[10][10], struct [statek](#) *statki←_gracz1[5], struct [statek](#) *statki_gracz2[5], int strzalyg1[10][10], int strzalyg2[10][10], char *gracz1, char *gracz2)
- void [kasowanie_statkow](#) (struct [statek](#) *statki[5])

4.11.1 Function Documentation

4.11.1.1 gra()

```
void gra (
    struct statek plansza_gracz1[10][10],
    struct statek plansza_gracz2[10][10],
    struct statek * statki_gracz1[5],
    struct statek * statki_gracz2[5],
    int strzalyg1[10][10],
    int strzalyg2[10][10],
    char * gracz1,
    char * gracz2 )
```

Funkcja rozpoczyna faktyczna czesc gry

Parameters

<i>plansza_gracz1</i>	plansza ze statkami pierwszego gracza
<i>plansza_gracz2</i>	plansza ze statkami drugiego gracza
<i>statki_gracz1</i>	tablica ze wskaznikami na statki pierwszego gracza
<i>statki_gracz2</i>	tablica ze wskaznikami na statki drugiego gracza
<i>strzalyg1</i>	plansza z poprzednimi strzałami pierwszego gracza
<i>strzalyg2</i>	plansza z poprzednimi strzałami drugiego gracza
<i>gracz1</i>	przechowuje nazwe gracza 1
<i>gracz2</i>	przechowuje nazwe gracza 2

4.11.1.2 kasowanie_statkow()

```
void kasowanie_statkow (
    struct statek * statki[5] )
```

Funkcja usuwa zalegle statki w pamieci

Parameters

<i>statki</i>	tablica ze wskaznikami na wszystkie statki danego gracza
---------------	--

4.11.1.3 ruch_gracza()

```
bool ruch_gracza (
    struct statek plansza[10][10],
    struct statek plansza_przeciwnika[10][10],
    int strzaly[10][10],
    struct statek * statki_przeciwnika[5] )
```

Funkcja umożliwia ruch, przez co gracz może oddać strzał, po czym sprawdza czy był trafiony, oraz czy gra może zostać zakończona

Parameters

<i>plansza</i>	plansza ze statkami danego gracza
<i>plansza_przeciwnika</i>	plansza ze statkami przeciwnika
<i>strzaly</i>	plansza z poprzednimi strzałami danego gracza
<i>statki_przeciwnika</i>	plansza ze statkami przeciwnika

4.11.1.4 sprawdz_statek()

```
bool sprawdz_statek (
    struct statek plansza_przeciwnika[10][10],
    int wiersz,
    int kolumna )
```

Funkcja sprawdza czy dany statek został w całości zniszczony

Parameters

<i>plansza_przeciwnika</i>	plansza ze statkami przeciwnika
<i>wiersz</i>	wiersz w którym znajduje się zestrzelony statek
<i>kolumna</i>	kolumna w której znajduje się zestrzelony statek

4.11.1.5 sprawdz_wszystkie_statki()

```
bool sprawdz_wszystkie_statki (
    struct statek * statki[5] )
```

Funkcja sprawdza czy graczowi zostały jeszcze jakieś statki

Parameters

<i>statki</i>	tablica ze wskaźnikami na wszystkie statki danego gracza
---------------	--

4.12 C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/EI Shipo/EI Shipo/tok-gry.h File Reference

Functions

- bool `ruch_gracza` (struct `statek` plansza[10][10], struct `statek` plansza_przeciwnika[10][10], int strzaly[10][10], struct `statek` *statki_przeciwnika[5])

- bool `sprawdz_statek` (struct `statek` plansza_przeciwnika[10][10], int wiersz, int kolumna)
- bool `sprawdz_wszystkie_statki` (struct `statek` *statki[5])
- void `gra` (struct `statek` plansza_gracz1[10][10], struct `statek` plansza_gracz2[10][10], struct `statek` *statki↔_gracz1[5], struct `statek` *statki_gracz2[5], int strzalyg1[10][10], int strzalyg2[10][10], char *gracz1, char *gracz2)
- void `kasowanie_statkow` (struct `statek` *statki[5])

4.12.1 Function Documentation

4.12.1.1 gra()

```
void gra (  
    struct statek plansza_gracz1[10][10],  
    struct statek plansza_gracz2[10][10],  
    struct statek * statki_gracz1[5],  
    struct statek * statki_gracz2[5],  
    int strzalyg1[10][10],  
    int strzalyg2[10][10],  
    char * gracz1,  
    char * gracz2 )
```

Funkcja rozpoczyna faktyczna czesc gry

Parameters

<i>plansza_gracz1</i>	plansza ze statkami pierwszego gracza
<i>plansza_gracz2</i>	plansza ze statkami drugiego gracza
<i>statki_gracz1</i>	tablica ze wskaznikami na statki pierwszego gracza
<i>statki_gracz2</i>	tablica ze wskaznikami na statki drugiego gracza
<i>strzalyg1</i>	plansza z poprzednimi strzałami pierwszego gracza
<i>strzalyg2</i>	plansza z poprzednimi strzałami drugiego gracza
<i>gracz1</i>	przechowuje nazwe gracza 1
<i>gracz2</i>	przechowuje nazwe gracza 2

4.12.1.2 kasowanie_statkow()

```
void kasowanie_statkow (  
    struct statek * statki[5] )
```

Funkcja usuwa zalegle statki w pamieci

Parameters

<i>statki</i>	tablica ze wskaznikami na wszystkie statki danego gracza
---------------	--

4.12.1.3 ruch_gracza()

```
bool ruch_gracza (
    struct statek plansza[10][10],
    struct statek plansza_przeciwnika[10][10],
    int strzaly[10][10],
    struct statek * statki_przeciwnika[5] )
```

Funkcja umożliwia ruch, przez co gracz może oddać strzał, po czym sprawdza czy był trafiony, oraz czy gra może zostać zakończona

Parameters

<i>plansza</i>	plansza ze statkami danego gracza
<i>plansza_przeciwnika</i>	plansza ze statkami przeciwnika
<i>strzaly</i>	plansza z poprzednimi strzałami danego gracza
<i>statki_przeciwnika</i>	plansza ze statkami przeciwnika

4.12.1.4 sprawdz_statek()

```
bool sprawdz_statek (
    struct statek plansza_przeciwnika[10][10],
    int wiersz,
    int kolumna )
```

Funkcja sprawdza czy dany statek został w całości zniszczony

Parameters

<i>plansza_przeciwnika</i>	plansza ze statkami przeciwnika
<i>wiersz</i>	wiersz w którym znajduje się zestrzelony statek
<i>kolumna</i>	kolumna w której znajduje się zestrzelony statek

4.12.1.5 sprawdz_wszystkie_statki()

```
bool sprawdz_wszystkie_statki (
    struct statek * statki[5] )
```

Funkcja sprawdza czy graczowi zostały jeszcze jakieś statki

Parameters

<i>statki</i>	tablica ze wskaźnikami na wszystkie statki danego gracza
---------------	--

4.13 C:/Users/macku/Desktop/Projekt PK (Semestr 2) 10.09/Maciej Kudela Projekt PK/Maciej Kudela Projekt PK/El Shipo/El Shipo/wyniki.txt File Reference

