Statki

Dokumentacja projektu

autor: Karolina Kołek

1. Instrukcja gry

Celem gry jest zatopienie wszystkich statków należących do przeciwnika.

Każdy z graczy zaznacza swoją flotę za pomocą znaków `'X'` w przeznaczonym do tego pliku (pierwszy z graczy w pliku `flota1.txt`, natomiast drugi gracz - w pliku `flota2.txt`). Do dyspozycji jest 100 pól oznaczonych kropkami. Na tych polach powinny się znaleźć 4 jednomasztowce, 3 dwumasztowce, 2 trójmasztowce i 1 czteromasztowiec. Statki można zaznaczać w pionie lub poziomie, mogą one również w dowolny sposób przylegać do krawędzi pola gry, jednak nie mogą się stykać z innymi statkami-ani bokami, ani rogami. Rysunek 1 przedstawia plik gotowy do naniesienia floty przez gracza, zaś rysunek 2 to przykład poprawnie naniesionej floty.

Rozgrywkę rozpoczyna gracz, który nanosił swoją flotę w pliku `flota1.txt`. Podaje on najpierw adres kolumny, a później numer wiersza wybranego przez siebie pola. Jeśli na danym polu znajduje się statek przeciwnika, na ekranie pojawia się informacja o trafieniu i gracz może strzelać kolejny raz. Jeżeli gracz spudłuje, kolejka przechodzi na przeciwnika. Zwycięzcą zostaje ten, kto pierwszy zatopi wszystkie okręty przeciwnika.

```
*ABCDEFGHIJ
0..........
1..........
2..........
3..........
4..........
5..........
6..........
7..........
8..........
9..........
```

Rysunek 1: plik gotowy do naniesienia floty

```
*ABCDEFGHIJ
0x........x
1..xx......
2.......xxx
3..........
4...xx....x
5.........x
6....xx...x
7..........
8...xxxx...
9x........x
```

Rysunek 2: przykład poprawnie naniesionej floty

2. Analiza zadania

2.1. Struktury danych

W programie wykorzystano listę jednokierunkową do przechowywania danych ze statystyk. Na dane te składa się imię gracza oraz liczba rozgrywek wygranych przez niego.

2.2. Algorytmy

Wyświetlanie statystyk:

Plik ze statystykami napisany jest tak, że najpierw podana jest nazwa gracza, a linijkę niżej liczba wygranych przez niego rozgrywek. Po otwarciu pliku ze statystykami, uruchamiana jest funkcja `Przygotuj_statystyki`, która do zmiennej `linia` pobiera linię tekstu. Jeśli jest to linia o numerze parzystym, jej zawartość zapisywana jest do zmiennej `nazwa`. W przeciwnym przypadku zawartość linii za pomocą funkcji `strtol` konwertowana jest na liczbę typu `int` i zapisywana do zmiennej `wygrane`. Następnie zmienne `nazwa` i `wygrane` są wysyłane do funkcji `Statystyki`, która odpowiedzialna jest za ułożenie listy jednokierunkowej. W kolejnym kroku za pomocą funkcji `Wyswietl_statystyki` na ekran drukowane są dane ze statystyk.

Przygotowanie do gry:

Przed rozpoczęciem gry gracze są proszeni o przedstawienie się. Następnie pojawia się możliwość wyświetlenia zasad gry. Jeśli użytkownik wybierze tę opcję, uruchamiana jest funkcja `Zasady_gry`, która drukuje na ekran reguły gry. W kolejnym kroku otwierany jest plik zawierający flotę naniesioną przez pierwszego gracza. Funkcja `Dobry_rozmiar` sprawdza, czy zmodyfikowany przez gracza plik ma prawidłowe wymiary. Funkcja `Czy_dobre_znaki` sprawdza, czy w pliku zostały użyte dopuszczalne znaki (gracz nie powinien używać znaków innych niż `'X'` lub `'x'`). Jeśli któraś z tych funkcji zwróci wartość `0`, wyświetla się komunikat informujący o błędnie naniesionej flocie i program kończy swoje działanie. Jeśli obydwie funkcje zwrócą wartość `1`, zostaje zaalokowana dynamicznie tablica o nazwie `flota1` (tablica z flotą naniesioną przez drugiego gracza nazywa się `flota2`), która za pomocą funkcji `Uzupelnij_flote`, wypełniana jest znakami z pliku tak, aby nowo utworzona flota była taka sama, jak flota z pliku. Do zmiennej `licznik` przypisana zostaje liczba znaków w pliku. Następnie zostaje zaalokowana dynamicznie tablica `tab` o rozmiarze określonym przez zmienną `licznik`. W kolejnym kroku wszystkie znajdujące się w tablicy spacje zostają zamienione na puste znaki, a wszystkie znaki `'\n'` zostają zastąpione `'\0'`. Do tablicy `flota1` zapisywane są kolejne znaki z `tab`. Gdy `flota1` jest uzupełniona, zwalniana jest pamięć przeznaczona na `tab`. Później funkcja `Sprawdz_poprawnosc` określa, czy flota została prawidłowo naniesiona, tzn. czy gracz naniósł odpowiednią liczbę statków oraz czy statki nie stykają się ze sobą. Jeśli funkcja zwróci wartość `1`, cała procedura zostaje powtórzona dla floty drugiego gracza. Jeśli któryś z graczy niepoprawnie naniósł swoją flotę, wyświetla się informujący o tym komunikati program kończy swoje działanie. W przypadku, gdy obie floty zostały naniesione poprawnie, pojawia się komunikat o rozpoczęciu gry.

Gra:

Następuje przejście do funkcji `Gra`, która odpowiedzialna jest za przeprowadzenie całej rozgrywki. Na początku uruchamiana jest funkcja `Przygotuj`, która modyfikuje tablice o nazwach `strzaly1` i `strzaly2` tak, by w pierwszym wierszu podane były adresy kolumn, a w pierwszej kolumnie- numery wierszy. Reszta tablicy wypełniona jest spacjami. Gra przekazana zostaje pierwszemu graczowi. Wyświetlana jest tablica `strzaly1`/`strzaly2` (w zależności od tego, czyja kolej), w której odnotowane są jego dotychczasowe strzały. Gracz podaje współrzędne miejsca, do którego chce strzelić. Funkcje `Czy_dobra_kolumna` i `Czy_dobry_wiersz` sprawdzają, czy podane przez użytkownika współrzędne są poprawne. Jeśli któraś ze współrzędnych będzie nieprawidłowa, pojawia się informacja o błędnym podaniu wiersza/kolumny i gracz zobowiązany jest do podania współrzędnych kolejny raz. Jeśli gracz poda poprawne współrzędne, następuje sprawdzenie, czy w tablicy `flota1`/`flota2`, w miejscu o podanych współrzędnych znajduje się statek. Jeśli tak, wyświetla się komunikat `Trafiony!`, a w tablicy `strzaly1`/`strzaly2` w miejscu o podanych współrzędnych pojawia się `'X'`. Funkcja `Czy_zatopiony` sprawdza, czy trafiony statek został zatopiony. Jeżeli tak, zostaje to zakomunikowane graczowi. Jeśli gracz nie trafił, pojawia się komunikat `Pudło!`, a w tablicy ze strzałami w miejscu o podanych współrzędnych pojawia się `'-'`. W przypadku, gdy gracz strzelał w miejsce, w którym znajduje się trafiony wcześniej statek, wyświetlona zostaje informacja, że gracz już tam strzelał oraz następuje utrata kolejki. Jeśli gracz nie trafi, kolejka przechodzi na przeciwnika i cała procedura powtarza się do momentu, gdy któryś z graczy zatopi wszystkie statki przeciwnika. Gdy to się stanie, wyświetlony zostaje komunikat informujący o zwycięstwie. Funkcja `Uaktualnij_statystyki` zwiększa o jeden liczbę gier wygranych przez danego gracza lub (jeśli gracz ten jeszcze nigdy nie wygrał), dodaje do listy jego nazwę wraz z liczbą wygranych gier równą 1.

Zakończenie:

Przed wyłączeniem się gry, wywoływana jest funkcja `Uaktualnij_statystyki`, która zwiększa o jeden liczbę wygranych przez zwycięzcę rozgrywek lub (jeśli gracz wygrał po raz pierwszy) zapisuje imię zwycięzcy do listy ze statystykami i ustawia liczbę wygranych gier na 1. Następnie za pomocą funkcji `Usun_flote`, pamięć przeznaczona na tablice `flota1` i `flota2` zostaje zwolniona. Uruchomiona zostaje funkcja `Zakoncz`, która zastępuje pliki `flota1.txt` i `flota2.txt` plikami, do których zapisane są tablice wypełnione za pomocą funkcji `Przygotuj` (pliki te wygląda jak na rysunku 1). Dzięki temu pliki są przygotowane do nowej rozgrywki. W kolejnym kroku plik ze statystykami zastępowany jest przez nowy plik wypełniony danymi ze zmodyfikowanej wcześniej listy. Funkcja `Usun_statystyki` zwalnia pamięć przeznaczoną na listę. Program kończy swoje działanie.

3. Specyfikacja zewnętrzna

Program uruchamiany jest z linii poleceń. Należy przekazać do programu nazwy plików zawierających kolejno: statystyki, flotę naniesioną przez pierwszego gracza oraz flotę naniesioną przez drugiego gracza po odpowiednich przełącznikach (odpowiednio `-s` dla pliku ze statystykami, `-i` dla pliku z flotą naniesioną przez pierwszego gracza, `-j` dla pliku z flotą naniesioną przez drugiego gracza), np.:

```
statki.exe -s statystyki.txt -i flota1.txt -j flota2.txt
```

Pliki są plikami tekstowymi. Przełączniki mogą być podane w dowolnej kolejności. Uruchomienie programu bez żadnego parametru, z nieprawidłową liczbą parametrów lub ze złymi parametrami spowoduje wyświetlenie komunikatu:

```
Podano zla liczbe parametrow.
```

Podanie nieprawidłowej nazwy pliku spowoduje wyświetlenie komunikatu:

```
Nie odnaleziono pliku o nazwie <nazwa.txt>.
```

4. Specyfikacja wewnętrzna

Program został zrealizowany zgodnie z paradygmatem strukturalnym. W programie rozdzielono interfejs (komunikację z użytkownikiem) od logiki aplikacji.

4.1. Ogólna struktura programu

W funkcji głównej, poprzez funkcje `Przygotuj_statystyki` i `Wyswietl_statystyki`, na ekran drukowane są nazwy graczy wraz z liczbą wygranych przez nich rozgrywek. Następnie, na życzenie gracza, uruchamiana jest funkcja `Zasady_gry`, wyświetlająca reguły gry w statki. W kolejnym kroku, za pomocą funkcji `Dobry_rozmiar`, `Czy_dobre_znaki`, `Uzupelnij_flote` i `Sprawdz_poprawnosc` sprawdzane jest, czy floty obu graczy zostały prawidłowo naniesione. Jeśli tak, funkcja `Przygotuj` uzupełnia tablice, w których będą zapisywane strzały graczy. Przed każdym strzałem gracze widzą, w które miejsca już celowali. W trakcie podawania współrzędnych miejsca, do którego gracz chce strzelić, funkcje `Czy_dobra_kolumna` i `Czy_dobry_wiersz` sprawdzają, czy podane współrzędne są poprawne. Jeżeli tak, sprawdzane jest, czy gracz trafił w statek przeciwnika. Jeżeli nie, gra przekazywana jest przeciwnikowi. W przeciwnym przypadku uruchamiana jest funkcja `Czy_zatopiony` sprawdzająca, czy dany okręt zatonął. Następnie sprawdzana jest liczba trafień gracza. Jeśli wynosi ona 20, oznacza to, że gracz zatopił całą flotę przeciwnika. Gra kończy się komunikatem o zwycięstwie. Uruchamiana jest funkcja `Uaktualnij_statystyki`, która modyfikuje listę ze statystykami (zwiększa o jeden liczbę wygranych przez zwycięzcę rozgrywek lub w przypadku, gdy gracz wygrał po raz pierwszy, dopisuje jego nazwę do listy i ustawia liczbę wygranych gier na 1). Następnie za pomocą funkcji `Usun_flote` zwalniana jest pamięć przeznaczona na tablice zawierające floty graczy. Na końcu uruchamiana jest funkcja `Zakoncz`, która przygotowuje pliki `flota1.txt` i `flota2.txt` do nowej

gry, zapisuje do pliku ze statystykami aktualne wyniki oraz zwalnia pamięć przeznaczoną na listę ze statystykami.

4.2. Szczegółowy opis typów i funkcji

Szczegółowy opis typów i funkcji zawarty jest w załączniku.

5. Testowanie

Program został przetestowany na różnego rodzaju plikach. Pliki puste, składające się z niepoprawnych znaków lub zawierające niepoprawnie naniesioną flotę, powodują wyświetlenie komunikatu:

```
<imie_gracza>, Twoja flota nie została poprawnie naniesiona. Prosze, popraw bledy i ponownie uruchom gre.
```

Następnie program kończy swoje działanie.

Jeżeli plik ze statystykami nie istnieje, zostanie wyświetlony komunikat:

```
Plik ze statystykami nie istnieje.
```

Jeżeli plik `flota1.txt` lub `flota2.txt` nie istnieje, wyświetlony zostanie komunikat:

```
Nie udalo sie otworzyc pliku <nazwa_pliku>.
```

i program zakończy swoje działanie.

Jeśli floty obu graczy zostały prawidłowo naniesione, rozpoczyna się gra.

W przypadku, kiedy gracz poda niepoprawny adres kolumny, wyświetlony zostaje komunikat:

```
Podano zla kolumne. Prosze o ponowne podanie wspolrzednych.
```

W przypadku, gdy gracz poda niepoprawny adres wiersza, wyświetlony zostaje komunikat:

```
Podano zly wiersz. Prosze o ponowne podanie wspolrzednych.
```

(w tym miejscu warto wspomnieć o tym, że jeżeli gracz poda liczbę większą od 9, program nie zinterpretuje tego jako źle podanego adresu wiersza. Pobrana zostanie tylko pierwsza cyfra z podanej liczby. To samo dotyczy kolumn- jeśli gracz poda ciąg znaków zaczynających się od litery z przedziału od `a` do `j`, pobrany zostanie tylko pierwszy znak, bez możliwości poprawienia błędu). Następnie gracz ma możliwość wpisania poprawnych współrzędnych. Jeśli gracz poda prawidłowe współrzędne i trafi w statek przeciwnika, wyświetlony zostaje komunikat: `Trafiony!` i gracz ma możliwość

strzelenia kolejny raz. Jeśli gracz zatopił statek przeciwnika, zostaje wyświetlony komunikat: `Zatopiony!` Jeżeli gracz zatopił całą flotę przeciwnika, zostaje wyświetlony komunikat: `Gratulacje! <nazwa_gracza> wygrywa!` oraz następuje uaktualnienie statystyk i program kończy swoje działanie. W przypadku, gdy gracz trafi w puste pole, wyświetlony zostaje komunikat: `Pudlo!` Jeśli gracz usiłował strzelić w miejsce, w którym znajduje się trafiony wcześniej statek, zostaje wyświetlony komunikat:

```
Juz tutaj strzelano! Tracisz kolejke!
```

Program został sprawdzony pod kątem wycieków pamięci.

Załącznik

Szczegółowy opis typów i funkcji

Dokumentacja gry w statki

Generated by Doxygen 1.8.18

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Class Documentation

3.1 element Struct Reference

`#include <funkcje.h>`

Public Attributes

- char nazwa [30]
- int wygrane
- struct element * pNastepny

3.1.1 Detailed Description

Definition at line 8 of file funkcje.h.

3.1.2 Member Data Documentation

3.1.2.1 nazwa

`char element::nazwa[30]`

Definition at line 10 of file funkcje.h.

3.1.2.2 pNastepny

`struct element* element::pNastepny`

Definition at line 12 of file funkcje.h.

3.1.2.3 wygrane

`int element::wygrane`

Definition at line 11 of file funkcje.h.

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

- funkcje.h

Chapter 4

File Documentation

4.1 flota1.txt File Reference

4.2 flota2.txt File Reference

4.3 funkcje.c File Reference

```
#include "funkcje.h"
```

Functions

- void Statystyki (struct element ∗poczatek, char ∗nazwa, int wygrane)
- void Przygotuj_statystyki (FILE ∗fp, struct element ∗poczatek)
- void Wyswietl_statystyki (struct element ∗poczatek)
- void Zasady_gry ()
- void Czekaj (int x)
- int Dobry_rozmiar (FILE ∗fp)
- int Czy_dobre_znaki (FILE ∗fp)
- int Sprawdz_poprawnosc (char ∗∗flota, int wiersze, int kolumny)
- void Uzupelnij_flote (FILE ∗fp, char ∗∗flota, int wiersze, int kolumny)
- void Przygotuj (char flota[11][11], char znak)
- int Gra (char ∗∗flota1, char ∗∗flota2, char ∗imie1, char ∗imie2, int wiersze1, int wiersze2, struct element ∗pHead)
- int Czy_dobra_kolumna (char kolumna)
- int Czy_dobry_wiersz (char wiersz)
- int Czy_zatopiony (char ∗∗flota, int wiersz, int kolumna)
- void Ususn_flote (char ∗∗flota, int wiersze)
- void Zapisz_do_statystyk (FILE ∗fp, struct element ∗poczatek)
- void Uaktualnij_statystyki (char ∗nazwa, struct element ∗poczatek)
- void Usun_statystyki (struct element ∗pHead)
- void Zakoncz (struct element ∗poczatek)

4.3.1 Function Documentation

4.3.1.1 Czekaj()

```
void Czekaj (
            int x )
```

Funkcja zatrzymuje na pewien czas system, dzieki czemu komunikaty na ekranie pojawiaja sie w okreslonych odstepach czasowych

Parameters

x	czas w milisekundach

Definition at line 99 of file funkcje.c.

4.3.1.2 Czy_dobra_kolumna()

```
int Czy_dobra_kolumna (
            char kolumna )
```

Funkcja sprawdza, czy adres kolumny podany przez gracza jest prawidlowy.

Parameters

kolumna	adres kolumny podany przez gracza

Returns

1, jezeli adres kolumny jest poprawny, w przeciwnym przypadku zwracana jest wartosc 0

Definition at line 674 of file funkcje.c.

4.3.1.3 Czy_dobre_znaki()

```
int Czy_dobre_znaki (
            FILE * fp )
```

Funkcja sprawdza, czy naniesiona w pliku flota sklada sie z dozwolonych znakow.

Parameters

fp	wskaznik do pliku

Returns

1, jezeli w pliku zostaly uzyte dozwolone znaki, w przeciwnym przypadku zwracana jest wartosc 0

Definition at line 137 of file funkcje.c.

4.3.1.4 Czy_dobry_wiersz()

```
int Czy_dobry_wiersz (
            char wiersz )
```

Funkcja sprawdza, czy adres wiersza podany przez gracza jest prawidlowy.

Parameters

wiersz	adres wiersza podany przez gracza

Returns

1, jezeli wiersz jest poprawny, w przeciwnym przypadku zwracana jest wartosc 0

Definition at line 683 of file funkcje.c.

4.3.1.5 Czy_zatopiony()

```
int Czy_zatopiony (
            char ** flota,
            int wiersz,
            int kolumna )
```

Funkcja sprawdza, czy trafiony statek zostal zatopiony.

Parameters

flota	tablica zawierajaca flote przeciwnika
wiersz	adres wiersza, do ktorego strzelano
kolumna	adres kolumny, do ktorej strzelano

Returns

1, jesli statek zostal zatopiony, jezeli nie, zwracana jest wartosc 0

Definition at line 691 of file funkcje.c.

4.3.1.6 Dobry_rozmiar()

```
int Dobry_rozmiar (
            FILE * fp )
```

Funkcja sprawdza, czy naniesiona w pliku flota ma prawidlowy rozmiar.

Parameters

fp	wskaznik do pliku

Returns

1, jezeli flota ma poprawny rozmiar, w przeciwnym przypadku zwracana jest wartosc 0

Definition at line 106 of file funkcje.c.

4.3.1.7 Gra()

```
int Gra (
            char ** flota1,
            char ** flota2,
            char * imie1,
            char * imie2,
            int wiersze1,
            int wiersze2,
            struct element * pHead )
```

Funkcja odpowiedzialna jest za przeprowadzenie gry.

Parameters

flota1	flota pierwszego gracza
flota2	flota drugiego gracza
imie1	imie pierwszego gracza
imie2	imie drugiego gracza
wiersze1	liczba wierszy tablicy flota1
wiersze2	liczba wierszy tablicy flota2
pHead	wskaznik na glowe listy

Returns

0, co oznacza koniec gry

Definition at line 472 of file funkcje.c.

4.3.1.8 Przygotuj()

```
void Przygotuj (
            char flota[11][11],
            char znak )
```

Funkcja wypelnia tablice tak, aby wiersz zerowy skladal sie z liter A-J, a zerowa kolumna z liczb 0-9. Wspolrzednej (0,0) przypisany jest znak '∗', a reszta tablicy wypelniona jest okreslonym znakiem (' ' w trakcie gry, '.', gdy tablica zapisywana jest do pliku). Dzieki tej funkcji gracz moze zobaczyc, w ktore miejsca juz strzelal, a po zakonczeniu gry funkcja ta pozwala na wyczyszczenie pliku z flota i przygotowanie jej do nastepnej gry.

Parameters

flota	tablica, ktora przygotowuje sie do gry lub do zapisu do pliku
znak	znak, ktorym tablica ma zostac wypelniona

Definition at line 450 of file funkcje.c.

4.3.1.9 Przygotuj_statystyki()

```
void Przygotuj_statystyki (
            FILE * fp,
            struct element * poczatek )
```

Funkcja pobiera z pliku kolejne linie, dzieli je na nazwy graczy i liczbe wygranych przez nich gier oraz wysyla otrzymane dane do funkcji "Statystyki".

Parameters

fp	wskaznik do pliku ze statystykami
poczatek	wskaznik na glowe listy

Definition at line 27 of file funkcje.c.

4.3.1.10 Sprawdz_poprawnosc()

```
int Sprawdz_poprawnosc (
            char ** flota,
```

```
            int wiersze,
            int kolumny )
```

Funkcja sprawdza, czy flota zostala naniesione w poprawny sposob.

Parameters

flota	wskaznik do tablicy, w ktorej naniesiona jest flota
wiersze	liczba wierszy tablicy
kolumny	liczba kolumn tablicy

Returns

1, jezeli flota zostala poprawnie naniesiona, w przeciwnym przypadku zwracana jest wartosc 0

Definition at line 154 of file funkcje.c.

4.3.1.11 Statystyki()

```
void Statystyki (
            struct element * poczatek,
            char * nazwa,
            int wygrane )
```

Funkcja uklada z przeslanych do niej danych liste jednokierunkowa.

Parameters

poczatek	wskaznik na glowe listy
nazwa	nazwa gracza
wygrane	liczba wygranych przez danego gracza rozgrywek

Definition at line 3 of file funkcje.c.

4.3.1.12 Uaktualnij_statystyki()

```
void Uaktualnij_statystyki (
            char * nazwa,
            struct element * poczatek )
```

Funkcja uaktualnia statystyki, tzn. sprawdza, czy nazwa zwyciezcy zostala juz kiedys zapisana w statystykach. Jesli tak, liczba wygranych przez niego gier zostaje zwiekszona o jeden. W przeciwnym przypadku nazwa gracza dodawana jest do listy przechowujacej dane ze statystykami.

Parameters

nazwa	nazwa gracza
poczatek	wskaznik na glowe listy

Definition at line 785 of file funkcje.c.

4.3.1.13 Usun_statystyki()

```
void Usun_statystyki (
            struct element * poczatek )
```

Funkcja usuwa liste ze statystykami.

Parameters

poczatek	wskaznik na glowe listy

Definition at line 818 of file funkcje.c.

4.3.1.14 Ususn_flote()

```
void Ususn_flote (
            char ** flota,
            int wiersze )
```

Funkcja usuwa tablice z naniesiona flota.

Parameters

flota	tablica, w ktorej naniesiona jest flota
wiersz	liczba wierszy tablicy

Definition at line 763 of file funkcje.c.

4.3.1.15 Uzupelnij_flote()

```
void Uzupelnij_flote (
            FILE * fp,
            char ** flota,
            int wiersze,
            int kolumny )
```

Funkcja uzupelnia wczesniej zaalokowana dynamicznie tablice znakami z pliku.

Parameters

fp	wskaznik do pliku
wiersze	liczba wierszy tablicy
kolumny	liczba kolumn tablicy

Definition at line 401 of file funkcje.c.

4.3.1.16 Wyswietl_statystyki()

```
void Wyswietl_statystyki (
            struct element * poczatek )
```

Funkcja wyswietla statystyki.

Parameters

poczatek	wskaznik na glowe listy

Definition at line 52 of file funkcje.c.

4.3.1.17 Zakoncz()

```
void Zakoncz (
            struct element * poczatek )
```

Funkcja usuwa pliki "flota1.txt" i "flota2.txt"oraz zastepuje je nowymi plikami o tych samych nazwach, ktore sa przygotowane do nowej gry. Nastepuje zaktualizowanie pliku ze statystykami oraz usuniecie listy, w ktorej dotychczas przechowywane byly statystyki.

Parameters

poczatek	wskaznik na glowe listy

Definition at line 834 of file funkcje.c.

4.3.1.18 Zapisz_do_statystyk()

```
void Zapisz_do_statystyk (
            FILE * fp,
            struct element * poczatek )
```

Funkcja aktualizuje statystyki po zakonczonej grze.

Parameters

fp	wskaznik do pliku ze statystykami
poczatek	wskaznik na glowe listy
nazwa	nazwa gracza, ktory wygral

Definition at line 773 of file funkcje.c.

4.3.1.19 Zasady_gry()

```
void Zasady_gry ( )
```

Funkcja wyswietla zasady gry

Definition at line 64 of file funkcje.c.

4.4 funkcje.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <ctype.h>
#include <string.h>
#include <math.h>
```

Classes

- struct element

Functions

- void Statystyki (struct element ∗poczatek, char ∗nazwa, int wygrane)
- void Przygotuj_statystyki (FILE ∗fp, struct element ∗poczatek)
- void Wyswietl_statystyki (struct element ∗poczatek)
- void Zasady_gry ()
- void Czekaj (int x)
- int Dobry_rozmiar (FILE ∗fp)
- int Czy_dobre_znaki (FILE ∗fp)
- void Uzupelnij_flote (FILE ∗fp, char ∗∗flota, int wiersze, int kolumny)
- int Sprawdz_poprawnosc (char ∗∗flota, int wiersze, int kolumny)
- void Przygotuj (char flota[11][11], char znak)
- int Gra (char ∗∗flota1, char ∗∗flota2, char ∗imie1, char ∗imie2, int wiersze1, int wiersze2, struct element ∗pHead)
- int Czy_dobra_kolumna (char kolumna)
- int Czy_dobry_wiersz (char wiersz)
- int Czy_zatopiony (char ∗∗flota, int wiersz, int kolumna)
- void Ususn_flote (char ∗∗flota, int wiersze)
- void Uaktualnij_statystyki (char ∗nazwa, struct element ∗poczatek)
- void Usun_statystyki (struct element ∗poczatek)
- void Zapisz_do_statystyk (FILE ∗fp, struct element ∗poczatek)
- void Zakoncz (struct element ∗poczatek)

Variables

- struct element element
- struct element * pHead

4.4.1 Function Documentation

4.4.1.1 Czekaj()

```
void Czekaj (
            int x )
```

Funkcja zatrzymuje na pewien czas system, dzieki czemu komunikaty na ekranie pojawiaja sie w okreslonych odstepach czasowych

Parameters

x	czas w milisekundach

Definition at line 99 of file funkcje.c.

4.4.1.2 Czy_dobra_kolumna()

```
int Czy_dobra_kolumna (
            char kolumna )
```

Funkcja sprawdza, czy adres kolumny podany przez gracza jest prawidlowy.

Parameters

kolumna	adres kolumny podany przez gracza

Returns

1, jezeli adres kolumny jest poprawny, w przeciwnym przypadku zwracana jest wartosc 0

Definition at line 674 of file funkcje.c.

4.4.1.3 Czy_dobre_znaki()

```
int Czy_dobre_znaki (
            FILE * fp )
```

Funkcja sprawdza, czy naniesiona w pliku flota sklada sie z dozwolonych znakow.

Parameters

fp	wskaznik do pliku

Returns

1, jezeli w pliku zostaly uzyte dozwolone znaki, w przeciwnym przypadku zwracana jest wartosc 0

Definition at line 137 of file funkcje.c.

4.4.1.4 Czy_dobry_wiersz()

```
int Czy_dobry_wiersz (
            char wiersz )
```

Funkcja sprawdza, czy adres wiersza podany przez gracza jest prawidlowy.

Parameters

wiersz	adres wiersza podany przez gracza

Returns

1, jezeli wiersz jest poprawny, w przeciwnym przypadku zwracana jest wartosc 0

Definition at line 683 of file funkcje.c.

4.4.1.5 Czy_zatopiony()

```
int Czy_zatopiony (
            char ** flota,
            int wiersz,
            int kolumna )
```

Funkcja sprawdza, czy trafiony statek zostal zatopiony.

Parameters

flota	tablica zawierajaca flote przeciwnika
wiersz	adres wiersza, do ktorego strzelano
kolumna	adres kolumny, do ktorej strzelano

Returns

1, jesli statek zostal zatopiony, jezeli nie, zwracana jest wartosc 0

Definition at line 691 of file funkcje.c.

4.4.1.6 Dobry_rozmiar()

```
int Dobry_rozmiar (
            FILE * fp )
```

Funkcja sprawdza, czy naniesiona w pliku flota ma prawidlowy rozmiar.

Parameters

fp	wskaznik do pliku

Returns

1, jezeli flota ma poprawny rozmiar, w przeciwnym przypadku zwracana jest wartosc 0

Definition at line 106 of file funkcje.c.

4.4.1.7 Gra()

```
int Gra (
            char ** flota1,
            char ** flota2,
            char * imie1,
            char * imie2,
            int wiersze1,
            int wiersze2,
            struct element * pHead )
```

Funkcja odpowiedzialna jest za przeprowadzenie gry.

Parameters

flota1	flota pierwszego gracza
flota2	flota drugiego gracza
imie1	imie pierwszego gracza
imie2	imie drugiego gracza
wiersze1	liczba wierszy tablicy flota1
wiersze2	liczba wierszy tablicy flota2
pHead	wskaznik na glowe listy

Returns

0, co oznacza koniec gry

Definition at line 472 of file funkcje.c.

4.4.1.8 Przygotuj()

```
void Przygotuj (
            char flota[11][11],
            char znak )
```

Funkcja wypelnia tablice tak, aby wiersz zerowy skladal sie z liter A-J, a zerowa kolumna z liczb 0-9. Wspolrzednej (0,0) przypisany jest znak '∗', a reszta tablicy wypelniona jest okreslonym znakiem (' ' w trakcie gry, '.', gdy tablica zapisywana jest do pliku). Dzieki tej funkcji gracz moze zobaczyc, w ktore miejsca juz strzelal, a po zakonczeniu gry funkcja ta pozwala na wyczyszczenie pliku z flota i przygotowanie jej do nastepnej gry.

Parameters

flota	tablica, ktora przygotowuje sie do gry lub do zapisu do pliku
znak	znak, ktorym tablica ma zostac wypelniona

Definition at line 450 of file funkcje.c.

4.4.1.9 Przygotuj_statystyki()

```
void Przygotuj_statystyki (
            FILE * fp,
            struct element * poczatek )
```

Funkcja pobiera z pliku kolejne linie, dzieli je na nazwy graczy i liczbe wygranych przez nich gier oraz wysyla otrzymane dane do funkcji "Statystyki".

Parameters

fp	wskaznik do pliku ze statystykami
poczatek	wskaznik na glowe listy

Definition at line 27 of file funkcje.c.

4.4.1.10 Sprawdz_poprawnosc()

```
int Sprawdz_poprawnosc (
            char ** flota,
```

```
            int wiersze,
            int kolumny )
```

Funkcja sprawdza, czy flota zostala naniesione w poprawny sposob.

Parameters

flota	wskaznik do tablicy, w ktorej naniesiona jest flota
wiersze	liczba wierszy tablicy
kolumny	liczba kolumn tablicy

Returns

1, jezeli flota zostala poprawnie naniesiona, w przeciwnym przypadku zwracana jest wartosc 0

Definition at line 154 of file funkcje.c.

4.4.1.11 Statystyki()

```
void Statystyki (
            struct element * poczatek,
            char * nazwa,
            int wygrane )
```

Funkcja uklada z przeslanych do niej danych liste jednokierunkowa.

Parameters

poczatek	wskaznik na glowe listy
nazwa	nazwa gracza
wygrane	liczba wygranych przez danego gracza rozgrywek

Definition at line 3 of file funkcje.c.

4.4.1.12 Uaktualnij_statystyki()

```
void Uaktualnij_statystyki (
            char * nazwa,
            struct element * poczatek )
```

Funkcja uaktualnia statystyki, tzn. sprawdza, czy nazwa zwyciezcy zostala juz kiedys zapisana w statystykach. Jesli tak, liczba wygranych przez niego gier zostaje zwiekszona o jeden. W przeciwnym przypadku nazwa gracza dodawana jest do listy przechowujacej dane ze statystykami.

Parameters

nazwa	nazwa gracza
poczatek	wskaznik na glowe listy

Definition at line 785 of file funkcje.c.

4.4.1.13 Usun_statystyki()

```
void Usun_statystyki (
            struct element * poczatek )
```

Funkcja usuwa liste ze statystykami.

Parameters

poczatek	wskaznik na glowe listy

Definition at line 818 of file funkcje.c.

4.4.1.14 Ususn_flote()

```
void Ususn_flote (
            char ** flota,
            int wiersze )
```

Funkcja usuwa tablice z naniesiona flota.

Parameters

flota	tablica, w ktorej naniesiona jest flota
wiersz	liczba wierszy tablicy

Definition at line 763 of file funkcje.c.

4.4.1.15 Uzupelnij_flote()

```
void Uzupelnij_flote (
            FILE * fp,
            char ** flota,
            int wiersze,
            int kolumny )
```

Funkcja uzupelnia wczesniej zaalokowana dynamicznie tablice znakami z pliku.

Parameters

fp	wskaznik do pliku
wiersze	liczba wierszy tablicy
kolumny	liczba kolumn tablicy

Definition at line 401 of file funkcje.c.

4.4.1.16 Wyswietl_statystyki()

```
void Wyswietl_statystyki (
            struct element * poczatek )
```

Funkcja wyswietla statystyki.

Parameters

poczatek	wskaznik na glowe listy

Definition at line 52 of file funkcje.c.

4.4.1.17 Zakoncz()

```
void Zakoncz (
            struct element * poczatek )
```

Funkcja usuwa pliki "flota1.txt" i "flota2.txt"oraz zastepuje je nowymi plikami o tych samych nazwach, ktore sa przygotowane do nowej gry. Nastepuje zaktualizowanie pliku ze statystykami oraz usuniecie listy, w ktorej dotychczas przechowywane byly statystyki.

Parameters

poczatek	wskaznik na glowe listy

Definition at line 834 of file funkcje.c.

4.4.1.18 Zapisz_do_statystyk()

```
void Zapisz_do_statystyk (
            FILE * fp,
            struct element * poczatek )
```

Funkcja aktualizuje statystyki po zakonczonej grze.

Parameters

fp	wskaznik do pliku ze statystykami
poczatek	wskaznik na glowe listy
nazwa	nazwa gracza, ktory wygral

Definition at line 773 of file funkcje.c.

4.4.1.19 Zasady_gry()

```
void Zasady_gry ( )
```

Funkcja wyswietla zasady gry

Definition at line 64 of file funkcje.c.

4.4.2 Variable Documentation

4.4.2.1 element

```
struct element element
```

4.4.2.2 pHead

```
struct element* pHead
```

Definition at line 16 of file funkcje.h.

4.5 main.c File Reference

```
#include "funkcje.h"
```

Functions

- int main (int argc, char *argv[ ])

4.5.1 Function Documentation

4.5.1.1 main()

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

Definition at line 3 of file main.c.

4.6 statystyki.txt File Reference

Index