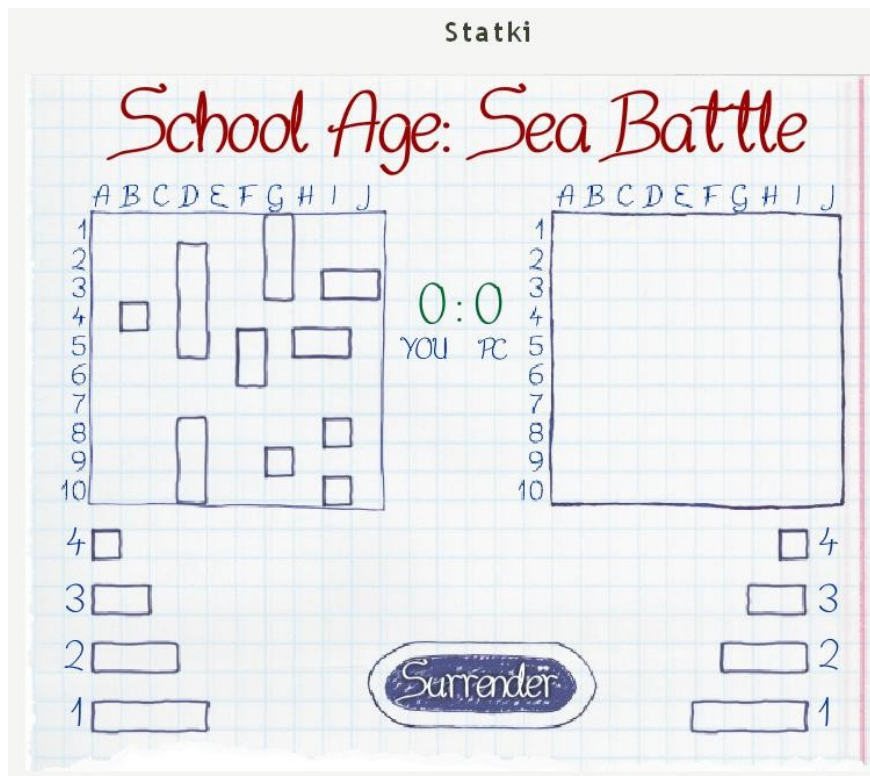


Podstawy programowania, Poniedziałek 5.06.2017

Projekt (zajęcia kontrolne 3), część 1

1. Zadanie

Projekt polega na stworzeniu logicznej gry komputerowej działającej w trybie tekstowym o nazwie „Statki”.



2. Cele

Celem projektu jest napisanie bardziej rozbudowanego programu niż było to możliwe na pojedynczych zajęciach i przećwiczenie wcześniej nauczonych mechanizmów języka C. Ponieważ jest to projekt, wymagana jest również własna kreatywność oraz inwencja – nie wszystko w treści zadania będzie napisane krok po kroku jak to zrobić.

Uwaga! Zalecane jest szerokie stosowanie zewnętrznych funkcji – w dobrym projekcie funkcja `main` powinna być jak najkrótsza.

3. Menu główne

Od razu po uruchomieniu program powinien wyświetlić menu główne, w którym znajdować się będzie opis możliwych do wyboru opcji. Menu główne powinno wyglądać mniej więcej tak:

```
wfpw@meyrinMac: /mnt/c/Users/Lukasz/Dropbox/Nasze_pPb/Uczelnia/PP/2017/lgraczyk/projekt/2017/p1/Statki
File Edit View Search Terminal Tabs Help
wfpw@meyrinMac: ~/Dropbox/Nasze_pPb/Uczelnia/PP/2017/lgraczyk/projekt/2017/p1
wfpw@meyrinMac: /mnt/c/Users/Lukasz/Dropbox/Nasze_pPb/Uczelnia/PP/2017/lgraczyk/projekt/2017/p1/Statki$
wfpw@meyrinMac: /mnt/c/Users/Lukasz/Dropbox/Nasze_pPb/Uczelnia/PP/2017/lgraczyk/projekt/2017/p1/Statki$ ./main

WITAJ W GRZE... STATKI!!!!
Co chcesz zrobic?
1. Nowa gra
0. Wyjście
Wybieram: 1
```

Menu działa w nieskończonej pętli `while`, której przerwanie i wyjście z programu następuje po wpisaniu opcji „0”. Wybór każdej z opcji jest dokonywany poprzez instrukcję `switch-case`.

2. Nowa Gra

Główną częścią pierwszych zajęć projektowych jest stworzenie funkcji `void NowaGra()`, w której ustawimy parametry jednego gracza (każdego z Was) – czyli statki na planszy.

Szczegóły implementacji

Głównym zadaniem jest implementacja funkcji `void NowaGra()`, która jest wywoływana po wybraniu w menu opcji 1. W funkcji tej tworzymy dwie plansze (naszych statków oraz trafień w plansze przeciwnika) oraz dodajemy statki do naszej planszy.

Struktura Gracz:

Tworzymy strukturę `Gracz`, która zawiera trzy pola: imię gracza (tablica znaków), oraz dwie tablice dwuwymiarowe (statyczne) typu `int` o rozmiarze 10x10 każda (jedna do przechowywania planszy z naszymi statkami, druga do przechowywania planszy z trafieniami w statki przeciwnika).

Do „obsługi” gracza tworzymy trzy funkcje pomocnicze:

- `void InicjalizujGracza(Gracz *gracz)` – przyjmuje zmienną typu `Gracz` i pyta użytkownika o podanie z klawiatury imienia gracza oraz ustawia wszystkie wartości pól obu tablic na 0

- `void UstawStatek(Gracz *gracz, int typ)` – przyjmuje zmienną typu `Gracz` i ustawia na naszej własnej planszy statek danego typu. Typ odpowiada danemu statkowi oraz jego rozmiarowi:

- Typ 5 – lotniskowiec – 5 kratek
- Typ 4 – pancernik – 4 kratki
- Typ 3 – krążownik – 3 kratki
- Typ 2 – niszczyciel – 2 kratki

Funkcja pyta się użytkownika o podanie z klawiatury pozycji (x,y) każdej kratki statku i ustawia odpowiednie pola na wartość odpowiadającą danemu typowi (np. dla lotniskowca ustawiamy 5 kratek na wartość 5). Przykładowo:

```
wfpw@meyrinMac: /mnt/c/Users/Lukasz/Dro
File Edit View Search Terminal Tab
wfpw@meyrinMac: ~/Dropb
WITAJ W GRZE... STATKI!!!!
Co chcesz zrobic?
1. Nowa gra
0. Wyjscie
Wybieram: 1
---===STATKI===---
Rozpoczynamy nowa gre

Gracz 1 - podaj swoje imie: Lukasz

Gracz 1 ustawia swoje 5 statkow!
Typ 5 - lotniskowiec
Podaj wspolrzedne
Podaj wspolrzedna 1 punktu statku
Podaj wspolrzedna x: 0
Podaj wspolrzedna y: 0

Podaj wspolrzedna 2 punktu statku
Podaj wspolrzedna x: 1
Podaj wspolrzedna y: 0

Podaj wspolrzedna 3 punktu statku
Podaj wspolrzedna x: 2
Podaj wspolrzedna y: 0

Podaj wspolrzedna 4 punktu statku
Podaj wspolrzedna x: 3
Podaj wspolrzedna y: 0

Podaj wspolrzedna 5 punktu statku
Podaj wspolrzedna x: 4
Podaj wspolrzedna y: 0

Statek typu 5 ustawiony!
```

- void WypiszPlansza(Gracz *gracz, int typPlanszy) – przyjmuje również zmienną typu Gracz oraz zmienną typu int określającą typ planszy, którą chcemy wypisać (0 – własnych statków, 1 – planszę trafień). Przykład użycia:

```
wfpw@meyrinMac: /mnt/c/Users/Lukasz/Dropbox/Nasze_pPb/Uczel
File Edit View Search Terminal Tabs Help
wfpw@meyrinMac: ~/Dropbox/Nasze_pPb/Uczel

Gracz 1 - poczatkowe plansze!
Wypisuje Twoja plansze statkow
  0 1 2 3 4 5 6 7 8 9 x
  = = = = = = = = =
0|5 5 5 5 5 0 0 0 0 0
1|0 0 0 0 0 0 0 0 0 0
2|4 4 4 4 0 0 0 0 0 0
3|0 0 0 0 0 0 0 0 0 0
4|3 3 3 0 0 0 0 0 0 0
5|0 0 0 0 0 0 0 0 0 0
6|3 3 3 0 0 0 0 0 0 0
7|0 0 0 0 0 0 0 0 0 0
8|2 2 0 0 0 0 0 0 0 0
9|0 0 0 0 0 0 0 0 0 0
y

Wypisuje Twaja plansze trafien statkow przeciwnika
  0 1 2 3 4 5 6 7 8 9 x
  = = = = = = = = =
0|5 5 5 5 5 0 0 0 0 0
1|0 0 0 0 0 0 0 0 0 0
2|4 4 4 4 0 0 0 0 0 0
3|0 0 0 0 0 0 0 0 0 0
4|3 3 3 0 0 0 0 0 0 0
5|0 0 0 0 0 0 0 0 0 0
6|3 3 3 0 0 0 0 0 0 0
7|0 0 0 0 0 0 0 0 0 0
8|2 2 0 0 0 0 0 0 0 0
9|0 0 0 0 0 0 0 0 0 0
y
```

Funkcje te wykorzystujemy do inicjalizacji statków gracza oraz wypisania jego plansz (np. w funkcji `NowaGra`)

W grze ustawiamy każdemu graczowi 5 statków:

- 1 lotniskowiec
- 1 pancernik
- 2 krążowniki
- 1 niszczyciel

Uwaga 1! Oczywiście, w idealnym świecie powinniśmy założyć ograniczenia na ustawianie pól statków (np. pola mogą być tylko obok siebie, statek musi być w całości położony pionowo lub poziomo na planszy, statki nie mogą się stykać i przekrywać, itp.) . W naszym prostym programie możemy pominąć tego typu zabezpieczenia. Oczywiście jeśli komuś starczy czasu to można próbować to implementować.

Uwaga 2! Na stronie znajduje się przykładowy skompilowany program (bez wyżej wspomnianych zabezpieczeń).

Punktacja:

1. Stworzenie menu głównego wraz z opcją wyjścia (**1 p.**)
2. Obsługa opcji „Nowa gra” wraz ze stworzeniem struktury `Punkt` i narysowaniem planszy (**8 p.**)
3. Makefile (**1 p.**)