

Katarzyna Mitrus

99269, grupa 6

Zadanie nr 2, Gra Gomoku

Oświadczenie

Świadomy odpowiedzialności prawnej oświadczam, że niniejszy projekt został napisany przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego lub zaliczenia przedmiotu z żadnej uczelni ani szkoły.

Manual

Zadanie, które program ma realizować

Program umożliwia rozgrywkę partii w Gomoku. Jest to gra podobna do słynnego „Kółko i krzyżyk”. Posiada jednak planszę 15x15, natomiast by wygrać należy ułożyć dokładnie 5 pionków w jednej linii (poziomo/pionowo lub po skosie), ułożenie większej ilości pionków nie przynosi wygranej, jest to tzw. „overline”.

Możliwy jest również remis w przypadku zapełnienia całej planszy pionkami obydwu graczy, bez wcześniejszego wyłonienia zwycięzcy.

Lista opcji do wyboru z krótkim opisem każdej z nich

Menu startowe pozwala nam na wybór jednej z 4 opcji:

```
NOWA GRA wcisnij (n)
WCZYTAJ OSTATNIO ZAPISANA GRE wcisnij (w)
POKAZ STATYSTYKI wcisnij (s)
WYJSCIE Z GRY wcisnij (0)
```

Wprowadzenie przez użytkownika znaku:

- ‘w’ oznacza wczytanie ostatnio przerwanej gry, niezbędne informacje do wznowienia rozgrywki wczytywane są z pliku,
- ‘s’ powoduje wyświetlenie listy dotychczasowych zwycięzców oraz ich przeciwników, a następnie powraca do menu głównego pozwalając zakończyć program lub rozpocząć grę,
- ‘0’ kończy działanie programu,
- ‘n’ oznacza wybór nowej gry, co prowadzi do wyświetlenia menu wyboru trybu:

Wybierz tryb gry:

```
Wcisnij (1) by grac [z komputerem]
Wcisnij (2) by grac [z drugim graczem]
Aby zakonczyc program wcisnij (0)
```

Wprowadzenie przez użytkownika znaku:

- ‘1’ oznacza rozpoczęcie gry z komputerem, program prosi o podanie dowolnej nazwy jednego gracza,
- ‘2’ oznacza rozpoczęcie gry z drugim człowiekiem (ruch na zmianę), program prosi o podanie dowolnej nazwy dla dwóch graczy,
- ‘0’ kończy działanie programu.

Użytkownik w każdej chwili może przerwać grę podając jako wybór wiersza ‘0’, o czym jest informowany na początku rozgrywki.

Program napisany jest tak, by radził sobie ze wszystkimi danymi wprowadzonymi od użytkownika. W przypadku nieprawidłowości prosi o ponowny wybór z dostępnych opcji.

ROZGRYWKA

Po podaniu nazwy gracza, program informuje użytkownika o przypisaniu pionków.

Przypomina także, że w każdej chwili może przerwać grę podając '0' jako wiersz.

Następnie wyświetla planszę i prosi o ruch jednego z graczy.

Poprawne dane dla wiersza to liczby z przedziału <0,15> , a dla kolumny litery z przedziału <a,o>.

W przypadku nieprawidłowości program, przypomina jakich danych oczekuje i pozwala na ponowne wprowadzenie.

Podaj swój nick: Katarzyna

Katarzyna Twoje pionki to: O
KOMPUTER Twoje pionki to: X

W kazdej chwili mozesz przerwac gre podajac '0' jako wiersz

| | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | | | | |
| 11 | | | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | |

Teraz ruch ma: Katarzyna

Podaj wiersz:
e

Musisz podac cyfre z przedzialu <1-15> ! Sprobuj jeszcze raz.
Podaj wiersz: 1

Podaj kolumne:
w

Musisz podac mala litere alfabetu <od 'a' do 'o'> ! Sprobuj jeszcze raz.
Podaj kolumne: a

Program reaguje również, kiedy pole wybrane przez użytkownika jest już zajęte pozwalając na ponowny wybór obu współrzędnych.

```

 9: | | | | | | | | | | | | | | | |
--|-----|-----|-----|-----|
10: | | | | | | | | | | | | | | | |
--|-----|-----|-----|-----|
11: | | | | | | | | | | | | | | | |
--|-----|-----|-----|-----|
12: | | | | | | | | | | | | | | | |
--|-----|-----|-----|-----|
13: | | | | | | | | | | | | | | | |
--|-----|-----|-----|-----|
14: | | | | | | | | | | | | | | | |
--|-----|-----|-----|-----|
15: | | | | | | | | | | | | | | | |
--|-----|-----|-----|-----|

Teraz ruch ma: Katarzyna
Podaj wiersz:
10
Podaj kolumne:
d
To pole jest zajete !
Wybierz jeszcze raz:
Podaj wiersz:
-

```

W przypadku wczytywania zapisu z pliku dane przetwarzane są przez seter posiadający odpowiednie warunki, co zapobiega wystartowaniu gry z niepoprawnymi danymi.

Nietypowe zachowania programu

Program w procesie testowania nie wykazał nietypowych zachowań.

Nie odnotowano także przypadku niekontrolowanego przerwania swojej pracy lub błędnego działania z powodu niewłaściwych danych. Zastosowano szereg warunków uniemożliwiających rozpoczęcie rozgrywki w przypadku niepoprawnych zmian zawartości plików wejściowych.

Wszystkie wyjścia z programu są kontrolowane, uprzedzone mechanizmem zwalniania pamięci oraz opatrzone odpowiednim komunikatem dla użytkownika.

Niezgodności z założeniami przekazanymi w treści zadania

Projekt tworzony był ściśle z listą wymagań. Spełnia on wszystkie założenia z treści zadania.

W programie zastosowane są następujące techniki i technologie obiektowe:

- ✓ kapsułkowanie
- ✓ dziedziczenie
- ✓ lista inicjalizatorów konstruktora
- ✓ polimorfizm
- ✓ szablony funkcji
- ✓ strumienie
- ✓ podział kodu programu pliki

Składnia danych wejściowych i wyjściowych

Struktura plików, typy danych przechowywane w poszczególnych polach pliku

Plik **statystyki.txt** przechowuje dane na temat rozegranych gier:

Katarzyna wygrywa z KOMPUTER 22.05.2016 14:43:50

Nazwy to pola (string) klasy gracz, a czas pobierany jest z bufora pomocniczego (tablica typu char).

Linie wczytywane są funkcją `getline()` przy użyciu klasy `string`.

Plik **zapis.txt** przechowuje dane na temat ostatnio przerwanej gry:

| | | |
|-----------|----------|---------------------------|
| Katarzyna | (string) | - nazwa pierwszego gracza |
| O | (char) | - pionek |
| 3 | (int) | - liczba ruchów |
| 4 | (int) | - wiersz |
| h | (char) | - kolumna |
| 5 | (int) | - wiersz |
| g | (char) | - kolumna |
| KOMPUTER | (string) | - nazwa drugiego gracza |
| X | (char) | - pionek |
| 3 | (int) | - liczba ruchów |
| 10 | (int) | - wiersz |
| d | (char) | - kolumna |
| 15 | (int) | - wiersz |
| e | (char) | - kolumna |

Historia ruchów przechowywana jest w oddzielnej liście dynamicznej dla każdego z graczy, po czym odpowiednio zapisywana do pliku. Przy wczytywaniu danych następuje jej ponowne zapełnienie, tak by dane te nie zostały utracone.

Struktura katalogów niezbędna dla poprawnego działania programu



Opis kodu

Lista plików z kodem źródłowym wchodzących w skład programu

Headers Files:

czlowiek.h
funkcje.h
gracz.h
komputer.h
rozmieszczenie.h

Source Files:

czlowiek.cpp
funkcje.cpp
gracz.cpp
komputer.cpp
main.cpp

Diagram klas

