

Jacek Giedronowicz

95175 , gr 1

Quoridor

Oświadczenie

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

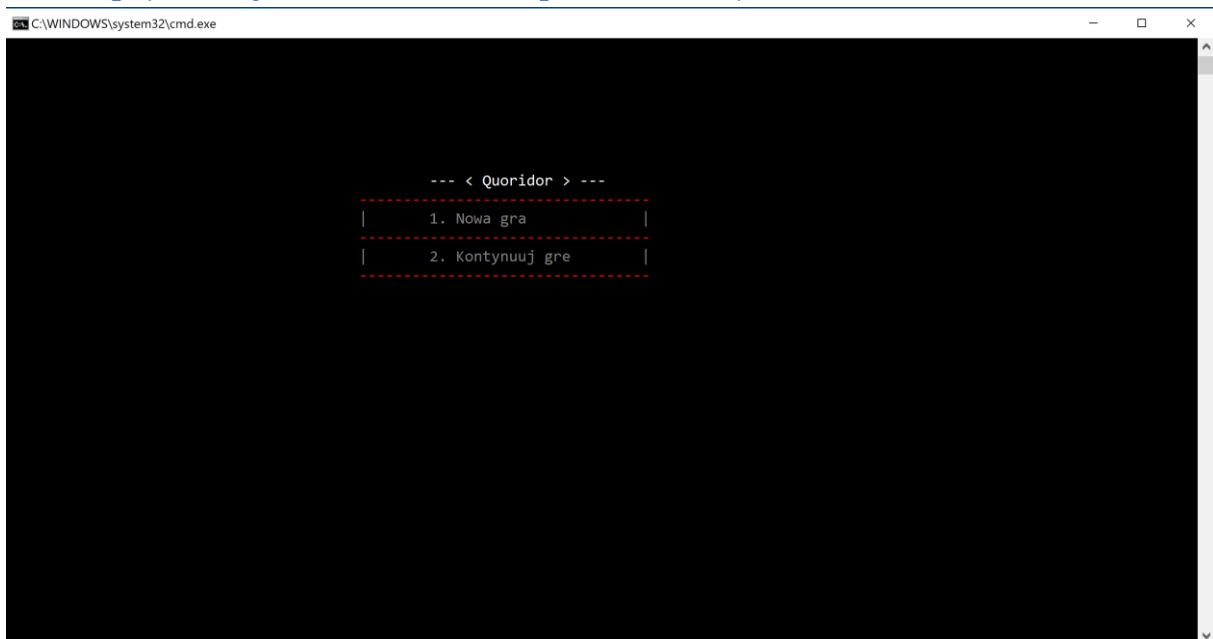
Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego lub zaliczenia przedmiotu z żadnej uczelni ani szkoły.

Manual

Zadanie, które program ma realizować

Program przeprowadza rozgrywkę gry Quoridor w trybie jednego lub dwóch graczy, z możliwością wznowienia poprzedniej rozgrywki.

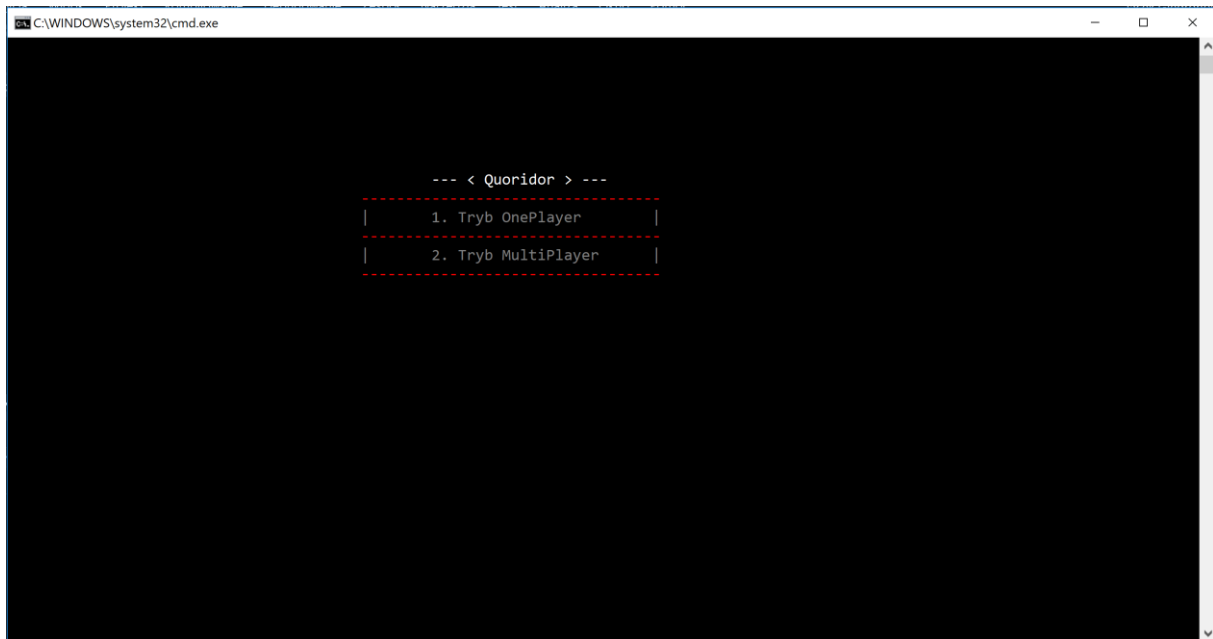
Lista opcji do wyboru z krótkim opisem każdej z nich



Menu.

Okno pojawia się tylko wtedy gdy program wykryje, że poprzednia rozgrywka nie została zakończona. Użytkownik ma dwie możliwości:

1. „Nowa gra” – zacząć grę od początku. Program resetuje planszę.
2. „Kontynuuj gre” – kontynuować grę od momentu jej przerwania.



Okno pojawia się zawsze, niezależnie od pojawienia się poprzedniego okna lub wyboru użytkownika. W tym oknie ma on dwie możliwości:

1. „Tryb OnePlayer” – gra w którym użytkownik gra z komputerem.
2. „tryb MultiPlayer” – gra w której przeciwko sobie gra dwóch użytkowników.

W obu oknach Menu program sprawdza poprawność wprowadzanych danych (odpowiednio „1” lub „2”) i w razie potrzeby prosi o powtórzenie.



Plansza:

Od lewej została wyświetlona plansza gry. 9x9 pól z przestrzenią pomiędzy na wstawienie ściany. Po prawej od planszy zostało połączone: tytuł gry, autor oraz krótka instrukcja możliwych ruchów wraz z przykładowymi postawionymi ścianami H12 i P12.

Pod planszą znajduje się informacja o aktualnej turze takie jak : czyj ruch , jego kolor , ile pozostało ścian do postawienia.

Zostało wprowadzone intuicyjne sterowanie pionkami: w , a , s , d. Odpowiednio góra , lewo , dół , prawo. Aby postawić ścianę należy podać 3-literową kombinację gdzie pierwsza oznacza kierunek (H – poziomo, P – pionowo) druga oznacza kolumnę (1-8) trzecia oznacza wiersz (1-8).

W wyborze drugiego i trzeciego parametru (kolumna i wiersz) nie prawidłową wartością było by 9 z powodu, że ściany zajmują dwa pola, oznaczałoby to, że ściana wyszła by po za planszę.

Program rozpoznaje jaki rodzaj ruchu chciał wykonać gracz (poruszyć pionkiem czy postawić ścianę) oraz sprawdza poprawność wprowadzonych danych.

Nietypowe zachowania programu

W przypadku gry OnePlayer komputer symuluje grę gracza 2 z odpowiednim prawdopodobieństwem wykonanych ruchów:

z prawdopodobieństwem $\frac{1}{3}$ komputer postawi ścianę.

z prawdopodobieństwem $\frac{2}{3}$ komputer wykona ruch pionkiem, gdzie

z prawdopodobieństwem $\frac{5}{10}$ poruszy się do dołu w kierunku mety.

z prawdopodobieństwem $\frac{2}{10}$ poruszy się w lewo oraz $\frac{2}{10}$ poruszy się w prawo.

z prawdopodobieństwem $\frac{1}{10}$ poruszy się do góry.

Z powodu niewielkiego prawdopodobieństwa poruszenia się do góry, w niektórych sytuacjach użytkownik może odczuwać powolny ruch komputera, który w pętli wciąż szuka dozwolonego ruchu.

Nie zgodności z założeniami przekazanymi w treści zadania

Program nie sprawdza czy istnieje droga, którą pionek mógłby przejść i wygrać.

Brak szablonów.

Składnia danych wejściowych i wyjściowych

Struktura plików, typy danych przechowywane w poszczególnych polach pliku

stan-gry.txt – służy do odczytania aktualnego stanu planszy zapisanego za pomocą oznaczeń :

„p” – pole , niezajęte przez żadnego gracza,

„s” – spacja, przerwa między polami, gdzie nie ma żadnej ściany ,

„g1” – gracz 1,

„g2” – gracz 2,

„-” – ściana pozioma,

„|” – ściana pionowa.

parametry.txt – przechowują postawione ściany.

Pierwszy wiersz – ściany poziome gracza 1.

Drugi wiersz – ściany pionowe gracza 1.

Trzeci wiersz – ściany poziome gracza 2.

Czwarty wiersz – ściany pionowe gracza 2.

Struktura katalogów niezbędna dla poprawnego działania programu

Pliki z kodem źródłowym (.cpp oraz .h), oraz pliki txt (stan-gry.txt oraz parametry.txt) powinny znajdować się w jednym, tym samym folderze.

Opis kodu

Lista plików z kodem źródłowym wchodzących w skład programu

biblioteki.h	Quoridor.cpp
człowiek.h	człowiek.cpp
gra.h	gra.cpp
gracz.h	gracz.cpp
komputer.h	komputer.cpp
menu.h	menu.cpp
plansza.h	plansza.cpp

Diagram klas

