



AKADEMIA GÓRNICZO-HUTNICZA

Dokumentacja do projektu

Othello

z przedmiotu

Języki programowania obiektowego

Elektronika i Telekomunikacja III rok

Paweł Urban

Grupa wtorek 15:00

prowadzący: Rafał Frączek

20.11.2024

1. Opis projektu

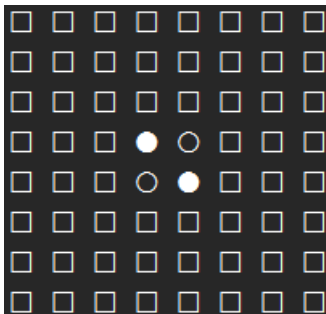
Othello to dwuosobowa gra planszowa, w której gracze naprzemiennie stawiają krążki na planszy 8 na 8. Gra wyświetla się w terminalu, ruchy wprowadzane są za pomocą klawiatury. Aplikacja zawiera menu główne w którym można zakończyć program lub przejść do jednego z kilku submenu - zasady, o autorze, gra. Postępy w grze są zapisywane, gracz przy ponownym włączeniu aplikacji zostaje zapytany czy wczytać ostatnią grę czy zacząć nową.

2. Project description

Othello is a two player turn-based board game. During the turn player places a disc on a 8x8 board. The game runs in terminal, players input moves using keyboard. Application contains main menu in which you can terminate the program or go to a few submenus – rules, author, game. Progress made in the game is saved. Upon entering the game the player is asked whether to continue or to start new game.

3. Instrukcja użytkownika

Gra odbywa się na planszy 8 na 8 pól. Pozycja startowa :



Zaczyna czarny, podczas ruchu gracz stawia krążek swojego koloru na którymś z wolnych pól. Musi położyć go w taki sposób by „otoczyć” przynajmniej jeden krążek przeciwnika. Otoczyć to znaczy położyć w taki sposób by rząd (bądź kolumna) krążków przeciwnika zaczynała się i kończyła krążkiem twojego koloru. Wówczas wszystkie otoczone krążki zmieniają kolor na kolor gracza który je otoczył. Gracze wykonują ruchy na przemian. Gdy w rzadkiej sytuacji jakiś gracz nie może wykonać ruchu który otacza przeciwnika pomija swoją turę. Gra kończy się gdy plansza zostanie zapełniona (lub żaden gracz nie może wykonać ruchu). Wygrywa ten który ma więcej krążków na planszy.

4. Kompilacja

Program Othello.cpp zawiera bardziej przejrzyste symbole, jednak trzeba go uruchomić w debugerze Visual Studio Code. Symbole te nie są zsynchronizowane ze standardem konsoli. Drugą opcją jest skompilowanie i włączenie programu OthelloSymbols.cpp w którym zostały użyte litery zamiast krążków.

5. Pliki źródłowe

Projekt składa się z następujących plików źródłowych:

- *Othello.cpp* – Plik główny w którym zaprogramowane są menu, submenu oraz sama gra.
- *Grid8x8.hpp* – Plik typu header w którym jest klasa Grid8x8. Jest ona odpowiedzialna za planszę oraz wszystkie operacje z nią związane.
- *OthelloSymbols.cpp* – Plik analogiczny do Othello.cpp używający pliku Grid8x8Symbols.hpp
- *Grid8x8Symbols.hpp* – Plik analogiczny do Grid8x8.hpp używający jako symboli 'B' 'W' 'O'

6. Zależności

W projekcie wykorzystano następujące dodatkowe biblioteki:

- Brak

7. Opis klas

W projekcie utworzono następujące klasy:

- **Grid8x8** – reprezentuje planszę na której rozgrywa się gra
 - `void Render()` – wyświetla w konsoli aktualny stan planszy,
 - `bool WhiteTurn(int RowNumber, int ColumnNumber, bool MakeMove)` – dla `MakeMove` równego `true` wykonuje ruch białego na polu o współrzędnych `[RowNumber, ColumnNumber]` pod warunkiem, że zasady na to pozwalają (nie wykona się np. gdy pole jest już zajęte) zwraca `true` gdy ruch się wykonał i `false` gdy się nie wykonał. Dla `MakeMove` równego `false` działa tak samo ale nie wykonuje ruchu, innymi słowy informuje czy ruch jest możliwy do wykonania ale go nie wykonuje.
 - `bool BlackTurn(int RowNumber, int ColumnNumber, bool MakeMove)` – wykonuje ruch czarnego, metoda analogiczna do `WhiteTurn`,
 - `void CheckWinner()` – Sprawdza czy gra się zakończyła (plansza jest zapełniona) i jeżeli tak to ogłasza kto wygrał oraz zakańcza działanie programu,
 - `void save()` – zapisuje obecny stan planszy w pliku `saveFile.txt`,
 - `void load()` – wczytuje stan planszy z pliku `saveFile.txt`,
 - `bool CheckMoveWhite()` – sprawdza czy istnieje jakiś legalny ruch dla białego gracza. Zwraca `true` jeśli istnieje i `false` gdy nie istnieje. Wykorzystuje funkcję `WhiteTurn` z argumentem `MakeMove` równym `false`.
 - `bool CheckMoveBlack()` – sprawdza czy istnieje jakiś legalny ruch dla czarnego gracza. Zwraca `true` jeśli istnieje i `false` gdy nie istnieje. Wykorzystuje funkcję `WhiteTurn` z argumentem `MakeMove` równym `false`.

8. Zasoby

W projekcie wykorzystywane są następujące pliki zasobów:

- *saveFile.txt* – plik w którym przechowywany jest stan planszy, oraz z którego jest wczytywana plansza gdy gracz chce kontynuować zapisaną rozgrywkę. Struktura pliku:
 - 8 linijek
 - Każda linijka składa się z 8 symboli oddzielonych spacjami
- *rules.txt* – plik w którym przechowywany jest spis zasad w j.angielskim. Struktura pliku:
 - Piętnaście linijek tekstu.

9. Dalszy rozwój i ulepszenia

Kod z pewnością mógłby zostać zoptymalizowany. Ponadto funkcjonalnością wartą dodania byłyby profile graczy, które zawierałyby informacje na temat zwycięstw i porażek z innymi graczami. Dalszym krokiem mogłoby być dodanie UI i możliwości wprowadzania ruchów za pomocą myszki.

10. Inne

Brak