



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

Dokumentacja do projektu

Connect Four

z przedmiotu

Języki programowania

Kierunek: Elektronika I

Kacper Filipek

czwartek 14:40

Prowadzący: Rafał Frączek

Kraków, 1 czerwca 2022

1. Opis projektu

Projekt jest realizacją gry w "Czwórki" (ang. "Connect Four") w C++, z interfejsem TUI (t.j. interfejs używający znaków specjalnych i kolorów do rysowania interfejsu użytkownika w konsoli).

2. Project description

The project is a realisation of the game "Connect Four" made in C++ with TUI interface (an interface using special characters for drawing UI in the terminal)

3. Instrukcja użytkownika

Po włączeniu programu wyświetla się menu główne z następującymi opcjami:

- Play - rozpoczyna grę w bieżącymi ustawieniami
- Load game - umożliwia wczytanie pliku z zapisem gry
- Options - otwiera menu umożliwiające zmianę ustawień gry
- Quit game - wychodzi z programu

Wciśnięcie klawisza Escape spowoduje pauzę w grze i wyświetlenie się menu z następującymi opcjami:

- Resume - wznawia przebieg gry (alternatywnie można ponownie wcisnąć Escape)
- Save game - zapisuje stan gry do pliku
- Main menu - kończy grę i przechodzi do głównego menu bez zapisu

Wybranie "Load game" wyświetla użytkownikowi pole tekstowe, do którego należy wpisać nazwę pliku z zapisem gry. W przypadku błędu w odczycie pliku, wyświetli się komunikat powiadamiający o wystąpieniu błędu.

W menu "Options" można zmienić następujące parametry:

- Board width - szerokość planszy do gry (domyślnie 7)
- Board height - wysokość planszy do gry (domyślnie 5)
- Victory condition - liczba żetonów w sekwencji potrzebna do wygrania gry (domyślnie 4)

4. Kompilacja

Program został napisany na systemy operacyjne z rodziny Linux, chociaż powinien on działać na Windowsie, ponieważ kod nie używa żadnych zależnych od platformy plików nagłówkowych. Program używa systemu CMake do budowania projektu. Można skompilować go na dwa sposoby:

- Sposób 1:
 1. Wejść do folderu build/
 2. Wykonać polecenie `cmake ..`
- Sposób 2:
 1. Z katalogu głównego wywołać skrypt `./bld.sh`. Ze względu na fakt, że skrypt wywołuje program `make`, może on nie działać na Windowsie.

Po zbudowaniu powinien się plik wykonywalny o `./build/connect-four`. Z uwagi na fakt, że ścieżki do zasobów są wpisane w programie relatywnie do głównego katalogu, to program wykonywalny powinien z niego wywoływany.

5. Pliki źródłowe

W tym punkcie należy opisać wszystkie pliki źródłowe (.cpp, .h) w projekcie. Należy podać nazwę każdego pliku oraz informację o tym co się w nim znajduje. Na przykład: Projekt składa się z następujących plików źródłowych:

- board.h, board.cpp – deklaracja oraz implementacja klasy Board,
- game.h, game.cpp – deklaracja oraz implementacja klasy Game,
- menu.h, menu.cpp – deklaracja oraz implementacja klasy Menu.
- extras.h, extras.cpp – deklaracja oraz implementacja funkcji pomocniczych.
- color.h – definicje procesora nazw kolorów do użycia w funkcjach biblioteki ncurses.
- main.cpp – główny plik z implementacją funkcji main.

6. Zależności

W projekcie wykorzystano następujące dodatkowe biblioteki:

- ncurses – biblioteka do interakcji z emulatorem terminala, pozwala tworzyć zaawansowane interfejsy konsolowe:
<https://invisible-island.net/ncurses/>

7. Opis klas

W projekcie utworzono następujące klasy:

- Menu - klasa reprezentująca menu główne programu
 - Menu()
 - GameParameters Start() - rozpoczyna pętlę klasy Menu i zwraca strukturę z parametrami gry po jej zakończeniu
- Game - klasa reprezentująca stan i właściwy przebieg rogowki
 - Game() - domyślny konstruktor klasy Game
 - void Start() - ustawia parametry i rozpoczyna pętlę gry
 - void set_parameters(GameParameters parameters) - ustawia parametry gry na podstawie podanego argumentu
 - void load_game(const std::string& path) - wczytuje zapis gry z pliku podanego przez path
- Board - klasa reprezentująca planszę do gry
 - Board() - domyślny konstruktor, ustawiający domyślne parametry
 - Board(uint8_t board_rows, uint8_t board_columns) - konstruktor parametryczny ustawia rozmiar planszy na board_rows × board_columns
 - uint16_t get_columns() - zwraca liczbę kolumn planszy
 - uint16_t get_rows() - zwraca liczbę wierszy planszy
 - uint16_t get_win_condition() - zwraca warunek zwycięstwa
 - void set_win_condition() - ustawia warunek zwycięstwa
 - void set_dimensions(uint16_t rows, uint16_t columns) - ustawia wymiary planszy
 - uint8_t check_victory() - sprawdza czy któryś z graczy wygrał

8. Zasoby

W projekcie wykorzystywane są następujące pliki zasobów:

- assets/ – katalog zawierający dodatkowe zasoby do gry.
Struktura katalogu:
 - logo1.txt, logo2.txt – pliki zawierające tekstowe logo pojawiające się na ekranie startowym.

9. Dalszy rozwój i ulepszenia

10. Inne

10.1. Format zapisu

Stan gry jest zapisywany do pliku w formie binarnej. Poniżej podane są po kolei bajty oznaczające konkretne parametry gry oraz ich rozmiary w pliku

1. Liczba graczy - 1 bajt (unsigned)
2. Aktualny gracz - 1 bajt (unsigned)
3. Wymiary planszy - 4 bajty (unsigned)
 - 3.1. liczba wierszy w planszy - 2 bajty (unsigned)
 - 3.2. liczba kolumn w planszy - 2 bajty (unsigned)
4. Warunek zwycięstwa - 2 bajty (unsigned)
5. zawartość planszy - każdej komórce odpowiada jeden bajt, więc rozmiar tej sekcji w bajtach, to liczba kolumn \times liczba wierszy (unsigned)

Parametry większe niż jeden bajt są zapisywane w formacie little endian (najmniej znaczący bajt na początku).