Tymoteusz Bal (nr indeksu 323074)

PROJEKT – Wstęp do programowania w języku C – krótki opis

TETRIS w trybie tekstowym na system Windows

1. Interakcja użytkownika z programem

Użytkownik po włączeniu programu widzi menu, w którym może rozpocząć nową grę, sprawdzić sterowanie, scoreboard lub wyjść z gry. Po wybraniu „START" pojawia się plansza gry ze spadającym pierwszym klockiem, a po prawej stronie widać następny klocek, który się pojawi. Pod planszą mamy informację o wyniku oraz prędkości gry, a także o możliwości wciśnięcia pauzy (ESC). Należy używając klawiatury operować klockiem:

- W – obrót klocka o 90 stopni w prawo;
- S – ruch w dół o 1 pole;
- A – ruch w lewo o 1 pole, D – ruch w prawo o 1 pole;
- ENTER – opuszczenie klocka na sam dół.

Użytkownik widzi cień spadającego klocka na dole. Należy oczywiście zgodnie z zasadami gry TETRIS układać pełne wiersze klocków, za co gra przyznaje punkty: 10 pkt za 1 wiersz, 40 pkt za 2 wiersze, 90 za 3 i 160 za 4 wiersze naraz. Znikaniu wiersza towarzyszy specjalna animacja. Po osiągnięciu odpowiednich progów punktowych prędkość spadania klocka rośnie – jest 5 poziomów prędkości. Gra kończy się, gdy nie ma miejsca na pojawienie się nowego klocka na górze planszy. Wtedy wyświetlana jest informacja „GAME OVER" oraz scoreboard zawierający 10 najlepszych wyników użytkownika na danym komputerze. Można wybrać „RESTART", wyjść do menu lub wyłączyć grę.

2. Moduły

Projekt składa się z 4 modułów:

- main.c – zawiera trzon rozgrywki: inicjalizację planszy, tablicy 7 różnych klocków typu struct Klocek, Klocka Obecny i jego Cienia oraz nieskończoną pętlę, która korzystając z odpowiednich funkcji powoduje wyrysowywanie planszy w oknie konsoli, samoistny spadek klocka w odpowiednim tempie oraz oczekuje wejścia z klawiatury użytkownika (z pomocą biblioteki conio). Oprócz tego do pomiaru czasu użyta jest biblioteka time, a do funkcji ClearScreen i HideCursor biblioteka windows;
- game_state.c i game_state.h – zawiera definicje funkcji odpowiedzialnych za zapisywanie stanu gry, czyli przede wszystkim aktualizowanie zawartości tablicy plansza;
- show.c i show.h – zawiera definicje funkcji odpowiedzialnych za wyświetlanie planszy, czyszczenie ekranu czy wyświetlanie stanów Menu i Pauzy
- blocks.h – zawiera definicje makr (WYS, SZER planszy) oraz struktur:
  - Pkt - zawierająca współrzędne całkowite x, y;
  - Klocek - zawierająca długość boku siatki – lokalnego układu współrzędnych klocka, ilość „zakazanych" pól (które muszą być wolne przy obrocie klocka), a także parametry typu Pkt: środek klocka, 4-elementową tablicę kwadratów, 3-elementową tablicę zakazanych pól oraz tablicę 5x5 – siatkę.

3. Funkcje

Moduł game_state.c składa się z funkcji:

- Inicjalizuj – wstawia na początku gry do tablicy plansza[WYS][SZER] odpowiednie znaki;
- WstawNastepny – wstawia po prawej stronie planszy obraz następnego wylosowanego klocka;
- UstawPredkosc – z każdą iteracją głównej pętli ustawia prędkość w zależności od aktualnego wyniku;
- Losuj – losuje Klocek z 7-elementowej tablicy i zwraca go;
- WstawKlocek – wstawia Klocek Obecny i jego Cien w odpowiednie pola planszy. Zwraca prawdę, jeśli można wstawić Klocek, a fałsz wpp (wtedy uruchamiany jest KoniecGry);

- Spadek, Lewo, Prawo - uruchamiane po wciśnięciu odpowiednich klawiszy i wstawiające Klocek Obecny w odpowiednie miejsce planszy po przemieszczeniu (po sprawdzeniu koniecznych warunków) oraz aktualizujący parametry Klocka (kwadraty, zakazane, srodek, siatka). Operacje wykonywane są z pomocą wskaźnika do struktury Klocek. Zwracają true, jeśli można było wykonać ruch, false wpp;
- ObrotPunktuWzglSrodka i Obrot – druga funkcja z pomocą pierwszej wstawia Klocek Obecny po obrocie w odpowiednie pola planszy (wykonuje obrót punktów srodek oraz punktów z tablic kwadraty, siatka, zakazane). Również wykorzystany jest wskaźnik do struktury Klocek. Obrot zwraca true, jeśli obrót był możliwy, false wpp;
- SprawdzWiersze i UsunWiersz – pierwsza sprawdza, czy odpowiednie wiersze są pełne, jeśli tak, to uruchamia drugą funkcję, która przemieszcza wszystkie klocki powyżej o 1 w dół. SprawdzWiersze zwraca prawdę, gdy usunięto co najmniej jeden wiersz, a fałsz wpp;
- Koniec Gry i Scoreboard – pierwsza funkcja wypisuje na ekran odpowiednie komunikaty oraz scoreboard, korzystając z drugiej funkcji. W nieskończonej pętli oczekuje na wejście z klawiatury użytkownika. Scoreboard natomiast wczytuje top 10 wyników z pliku score.txt (lub mniej, lub tworzy ten plik) do tablicy, wstawia nowy wynik w odpowiednie miejsce w tablicy i przesuwa pozostałe (jeśli wynik mieści się w top 10), a na końcu wypisuje nową listę wyników na ekran oraz do pliku score.txt.

Moduł show.c składa się z funkcji:

- ClearScreen i HideCursor, które pomagają w ładnym wyświetlaniu planszy;
- Menu i Pauza, które wyświetlają odpowiedni tekst na ekran oraz w nieskończonej pętli oczekują na wejście z klawiatury użytkownika;
- Rysuj – wypisuje planszę na ekran.

4. Przykładowe rozszerzenia

- różne opcje w menu, np. zmiana startowej prędkości gry byłaby dość prosta do implementacji (zmiana parametru predkosc, zapisywanie zmian w pliku .txt);
- kolorowe klocki – z pomocą ANSI escape code (choć kod ANSI nie chce mi działać w CodeBlocks, prawdopodobnie to jakiś problem z kompilatorem) lub funkcji napisanej w oparciu o polecenia systemowe (takie jak system(„COLOR 0A”)) – też nie powinno być bardzo trudne, choć może nieco pracochłonne;
- przerobienie gry na grę w interfejsie graficznym, takim jak GTK czy SFML – byłoby trudne, wymagałoby zmiany całej/sporej części struktury kodu.

Pierwsze 2 punkty może spróbuję zaimplementować, choć nie obiecuję niczego, bo muszę zająć się teraz drugim projektem z PWI i nauką do sesji. Ogólnie z racji tego, że gra jest w terminalu, można sobie dostosować wygląd, np. czcionki, kolor tła, kolor znaków. W README repozytorium projektu mam napisane właściwości, przy których otrzymujemy ładnie wyglądającego TETRISa w stylu gier na stare gameboye (przy domyślnych ustawieniach plansza jest zbyt wąska).