

Politechnika Śląska  
Wydział Matematyk Stosowanej  
Kierunek Informatyka  
Gliwice, 30.01.2024

Programowanie I  
**projekt zaliczeniowy**

***"Test prędkości pisania"***

**Marta Załuska gr. lab. 3/6**

## 1. Opis projektu

Program testuje prędkość oraz poprawność przepisywania podanego tekstu przez użytkownika oraz wyświetla statystyki jego podejścia.

## 2. Funkcjonalności

- Możliwość wyboru czasu trwania (20 lub 60 sekund)
- Możliwość wyboru rodzaju tekstu do wpisania (czy ma zawierać polskie znaki diakrytyczne, czy nie)
- Podświetlanie wpisywanych znaków na zielono lub czerwono, zgodnie z ich poprawnością
- Sygnał dźwiękowy informujący użytkownika o popełnionym błędzie
- Obliczanie poprawności, ilości wpisanych słów i prędkości pisanie (znaków na minutę)
- Ekran z podsumowaniem i statystykami
- Informacja o dotychczasowym rekordzie i ewentualnie jego pobiciu

## 3. Przebieg realizacji

Proste menu użytkownik obsługuje strzałkami. Przy poruszaniu się po menu program odtwarza dźwięki w stylu starych konsoli do gier. Użytkownik za pomocą menu ma możliwość wyświetlenia instrukcji oraz dostosowania ustawień – czasu trwania testu oraz typu tekstu do przepisania (czy ma zawierać polskie znaki). Tekst do przepisania składa się z losowanych unikatowych słów z bazy 10000 najpopularniejszych słów w języku polskim (link do bazy znajduje się w opisie pliku poniżej), bądź, w wariantcie bez znaków polskich, z 5301 z tych słów, które nie zawierają polskich znaków. Formatowanie pliku tak, aby nie zawierał słów z polskimi znakami przeprowadziłam w Excelu. Wyświetlanie polskich znaków odbywa się dzięki funkcji `setlocale`, której przekazywany jest parametr „Polish”.

Podczas testu użytkownik widzi zegar odliczający czas trwania, który wybrał. Zegar rozpoczyna się po pierwszym naciśnięciu dowolnej litery (a-z) – program zapisuje czas z funkcji `clock` jako czas startowy i odmierza daną ilość czasu.

Wczytywanie klawisza wciśniętego przez użytkownika odbywa się dzięki funkcji `GetAsyncKeyState`, dzięki czemu program nie musi czekać na znak od użytkownika, więc może dynamicznie wyświetlać pozostały czas.

Podobne zadania spełniałaby funkcja `_kbhit` z biblioteki `conio.h`, ale czasem ma ona problemy z obsługą polskich znaków. `GetAsyncKeyState` sprawdza, czy dany klawisz jest naciśnięty, więc aby obsłużyć polskie znaki sprawdza ona również, czy podczas naciśnięcia litery wciśnięty jest również klawisz `Alt`. Nie sprawdza ona również żadnych znaków poza literami (a-z) oraz `Alt`, `Backspace` i `Escape`, dzięki czemu nie trzeba specjalnie oczyszczać strumienia z niechcianych znaków.

Aby zminimalizować miganie konsoli, wielokrotnie używam zmieniania koordynatów kursora. Dzięki temu możliwe jest również postawienie kursora w określonym miejscu, aby użytkownik widział, który znak wpisuje. Cursor ma określoną pozycję docelową – na początku znajduje się po lewej stronie ekranu, na lewym końcu paska z tekstem do wpisania, a kiedy dojdzie do ustalonej pozycji zatrzymuje się, a przesuwać zaczyna się pasek z tekstem. Dzięki temu użytkownik może skupić wzrok na jednym miejscu.

Poprawność wpisywanych znaków jest sygnalizowana za pomocą kolorów (błędnie wpisany znak wyświetlany jest na czerwono, poprawnie – na zielono) oraz sygnału dźwiękowego w momencie popełnienia błędu. Użytkownik może cofać wpisany tekst klawiszem Backspace.

Po zakończeniu czasu program oblicza poprawność, ilość wpisanych wyrazów (jest ona obliczana na podstawie ilości wyrazów w tekście do wpisania, a nie wpisanego przez użytkownika, aby w razie pomyłki w miejscu spacji złączone dwa wyrazy były liczone osobno), oraz ilość znaków na minutę. Wyświetlany jest również rekord, który wczytywany jest z pliku highscore.txt, a w razie jego pobicia zastępowany jest nowym rekordem.

Po zapoznaniu się z wyświetlonymi statystykami możliwy jest powrót do menu i ponowna rozgrywka nielimitowaną ilość razy, aż do zakończenia programu klikając Escape w menu.

Na projekt składają się pliki:

- main.cpp – zawierający funkcję main i główne funkcje obsługujące program
- display.cpp – zawierający funkcje służące do wyświetlania interfejsu
- functions.cpp – zawierający funkcje pomocnicze, np. do obsługi pozycji kursora w konsoli, do mierzenia czasu, czy czytania naciśniętego znaku
- header.h – plik nagłówkowy
- słowa.txt, słowa2.txt – pliki zawierające słowa z bazy 10000 najpopularniejszych słów w języku polskim ([https://pl.wiktionary.org/wiki/Indeks:Polski - Najpopularniejsze s%C5%82owa 1-10000 wersja Jerzego Kazojcia](https://pl.wiktionary.org/wiki/Indeks:Polski_-_Najpopularniejsze_s%C5%82owa_1-10000_wersja_Jerzego_Kazojcia)); słowa2.txt zawiera tylko słowa bez polskich znaków diakrytycznych
- highscore.txt – plik tekstowy służący do zapisywania rekordu użytkownika
- wrong\_sound.wav, next\_sound.wav, select\_sound.wav – pliki z dźwiękami, które odtwarzane są podczas działania programu

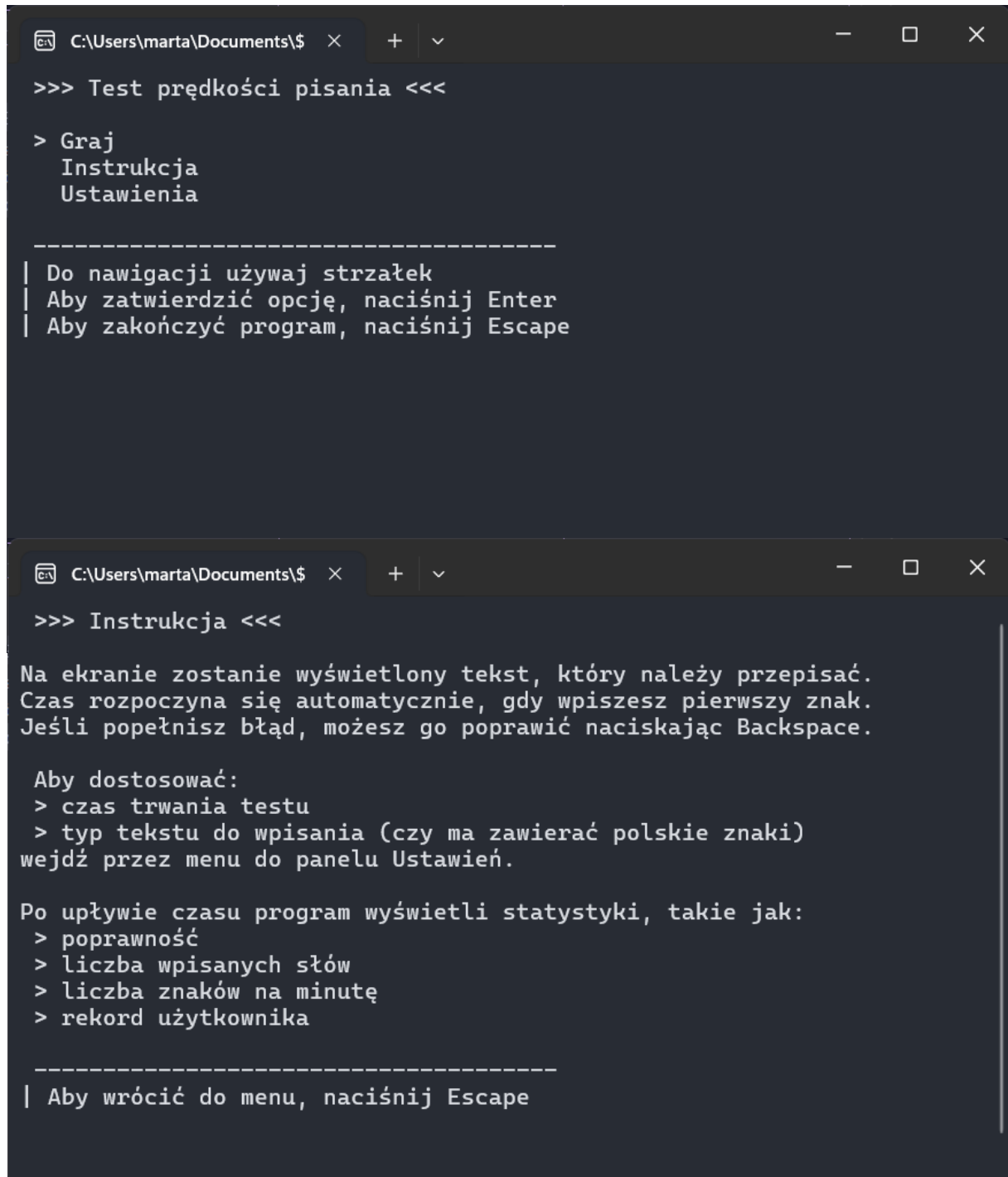
W projekcie zostały wykorzystane biblioteki:

- iostream – do obsługi strumienia wejścia/wyjścia
- fstream – do obsługi odczytu plików
- conio.h – \_getch(), \_kbhit()
- windows.h – PlaySound(), SetConsoleTextAttribute(), Sleep(), GetConsoleCursorInfo(), SetConsoleCursorPosition(), GetAsyncKeyState(), GetStdHandle()
- string – getline()

- iomanip – setprecision()
- ctime – do obsługi zegara (funkcja clock())
- locale.h – setlocale()
- winmm.lib – do obsługi dźwięków

#### 4. Instrukcja użytkownika

Obsługa menu odbywa się za pomocą strzałek na klawiaturze i klawiszy Enter (wybierz opcję) oraz Escape (w menu – wyjdź z programu, w pozostałych przypadkach – wróć do menu).



```

C:\Users\marta\Documents\$ >>> Test prędkości pisania <<<

> Graj
  Instrukcja
  Ustawienia

-----
| Do nawigacji używaj strzałek
| Aby zatwierdzić opcję, naciśnij Enter
| Aby zakończyć program, naciśnij Escape

C:\Users\marta\Documents\$ >>> Instrukcja <<<

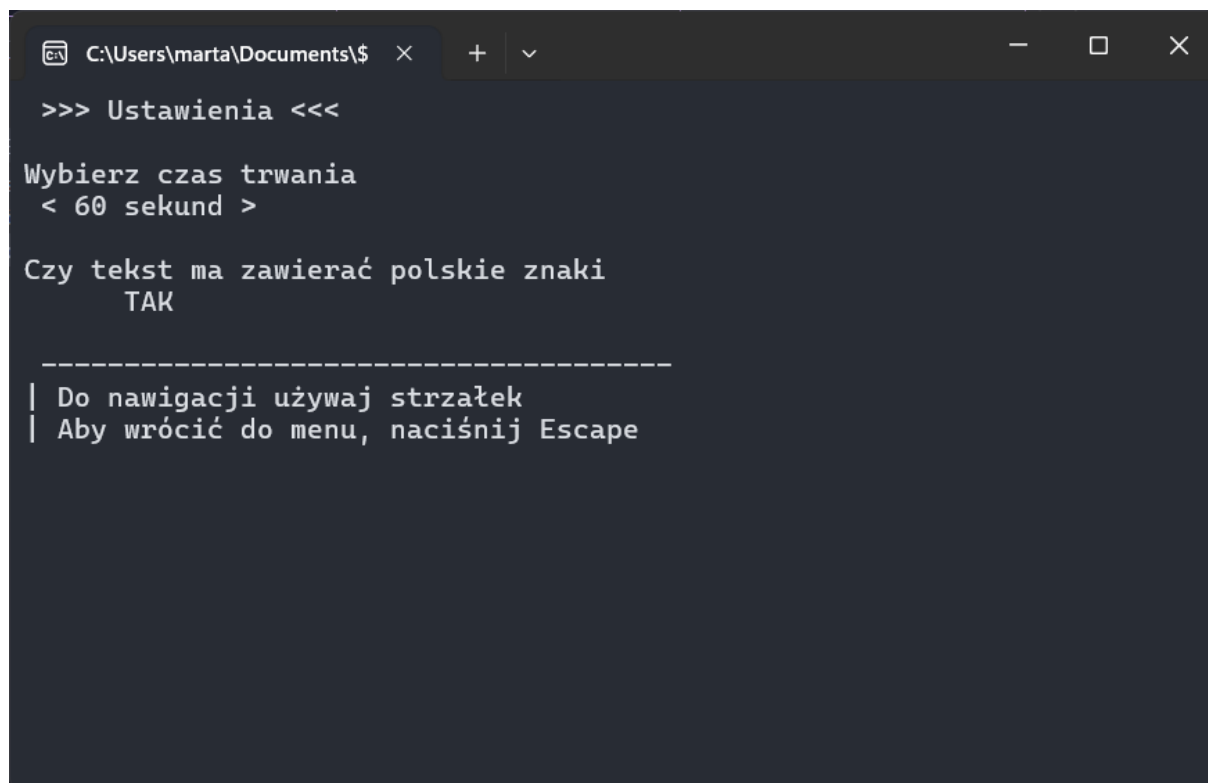
Na ekranie zostanie wyświetlony tekst, który należy przepisać.
Czas rozpoczyna się automatycznie, gdy wpiszesz pierwszy znak.
Jeśli popełnisz błąd, możesz go poprawić naciskając Backspace.

Aby dostosować:
> czas trwania testu
> typ tekstu do wpisania (czy ma zawierać polskie znaki)
wejdź przez menu do panelu Ustawień.

Po upływie czasu program wyświetli statystyki, takie jak:
> poprawność
> liczba wpisanych słów
> liczba znaków na minutę
> rekord użytkownika

-----
| Aby wrócić do menu, naciśnij Escape
  
```

Aby dostosować czas trwania testu oraz typ tekstu do wpisania (czy ma zawierać polskie znaki) wejdź przez menu do panelu Ustawień.

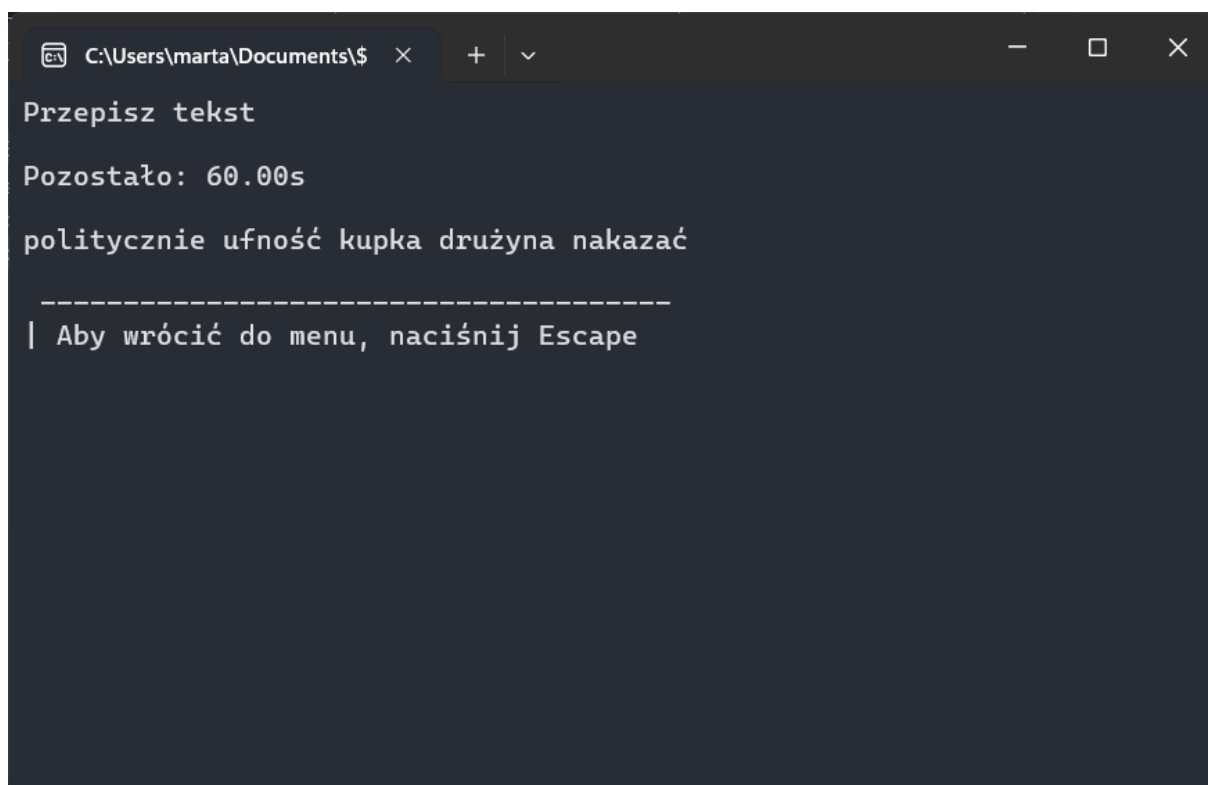


```
C:\Users\marta\Documents\$ x + v - □ X
>>> Ustawienia <<<

Wybierz czas trwania
< 60 sekund >

Czy tekst ma zawierać polskie znaki
TAK

-----
| Do nawigacji używaj strzałek
| Aby wrócić do menu, naciśnij Escape
```



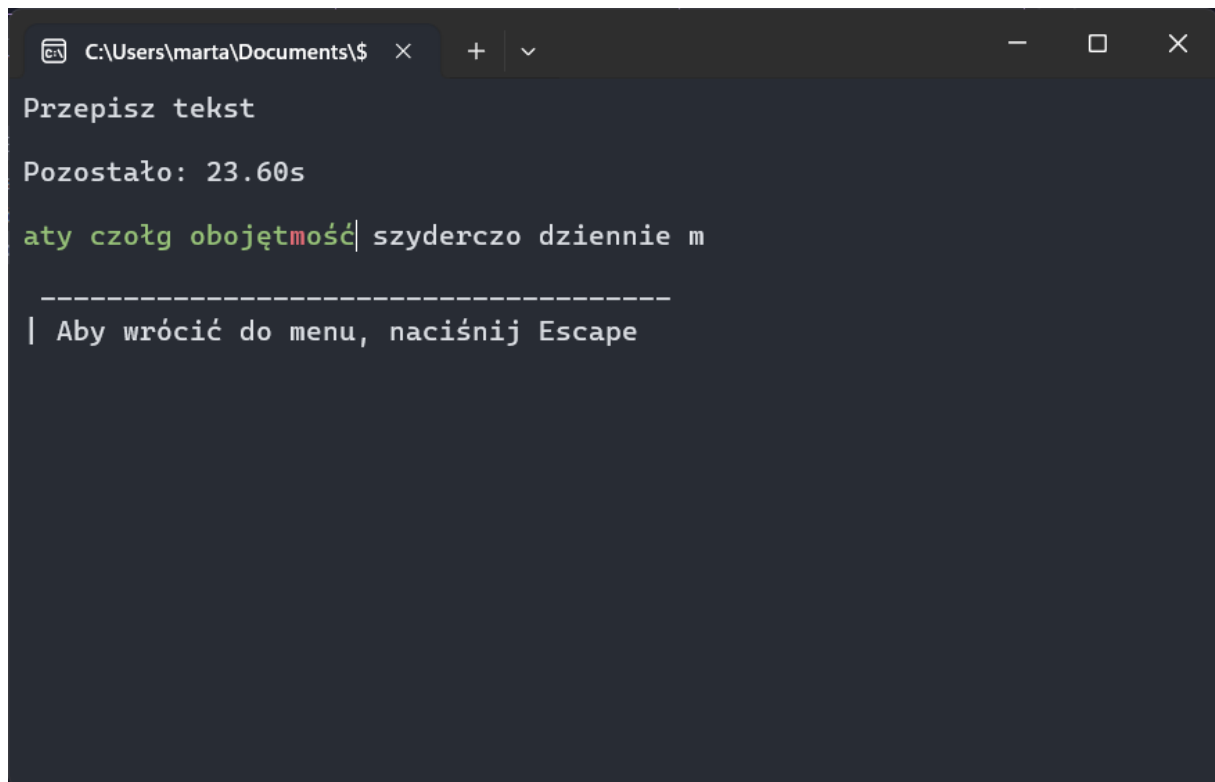
```
C:\Users\marta\Documents\$ x + v - □ X
Przepisz tekst

Pozostało: 60.00s

politycznie ufność kupka drużyna nakazać

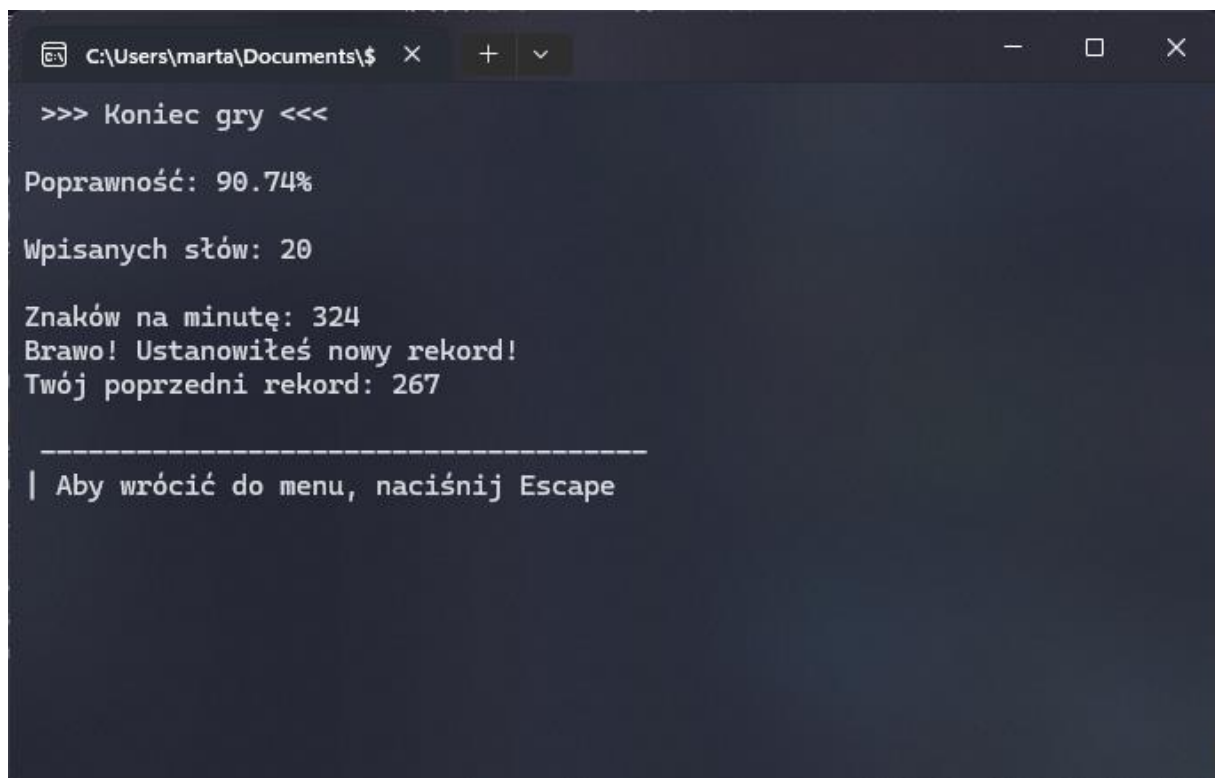
-----
| Aby wrócić do menu, naciśnij Escape
```

Na ekranie zostanie wyświetlony tekst, który należy przepisać. Czas rozpoczyna się automatycznie, gdy wpiszesz pierwszy znak. Jeśli popełnisz błąd, możesz go poprawić naciskając Backspace.



```
C:\Users\marta\Documents\> Przepisz tekst
Pozostało: 23.60s
aty czołg obojętmość| szyderczo dziennie m
-----
| Aby wrócić do menu, naciśnij Escape
```

Po upływie czasu program wyświetli statystyki, takie jak poprawność, liczba wpisanych słów, liczba znaków na minutę, ustanowiony rekord.



```
C:\Users\marta\Documents\> >>> Koniec gry <<<
Poprawność: 90.74%
Wpisanych słów: 20
Znaków na minutę: 324
Brawo! Ustanowiłeś nowy rekord!
Twój poprzedni rekord: 267
-----
| Aby wrócić do menu, naciśnij Escape
```

## 5. Podsumowanie i wnioski

Zrealizowałam wszystkie zaplanowane zadania, które realizują inne tego typu programy (zwykle w formie stron internetowych). Później dodałam również dodatkowe funkcjonalności, jak możliwość wyboru ustawień, dźwięki oraz rekord użytkownika.

Największym problemem okazało się optymalizowanie wyświetlania, w tym obliczanie pozycji kursora oraz sposób poruszania się paska z tekstem.

Ewentualne dalsze rozbudowanie programu może polegać np. na dodaniu opcji automatycznego pomijania Spacji, aby użytkownik mógł wpisywać same litery. Można by również zmusić program do zmniejszenia rozmiaru i pozycji okienka konsoli, aby zwiększyć estetykę projektu.