

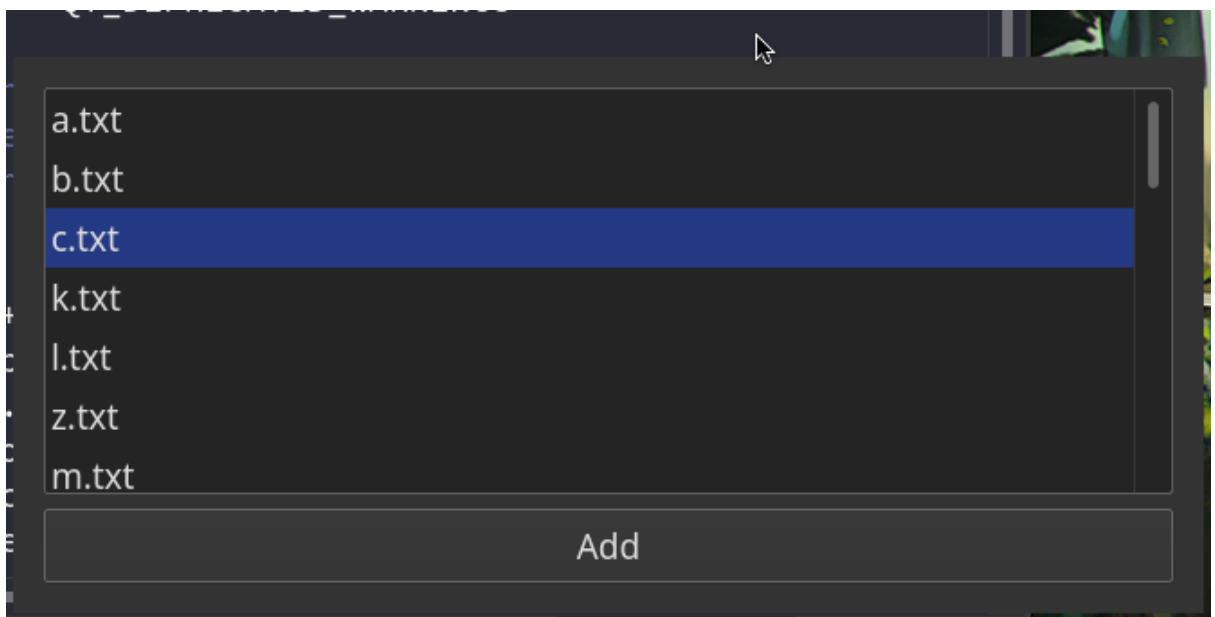
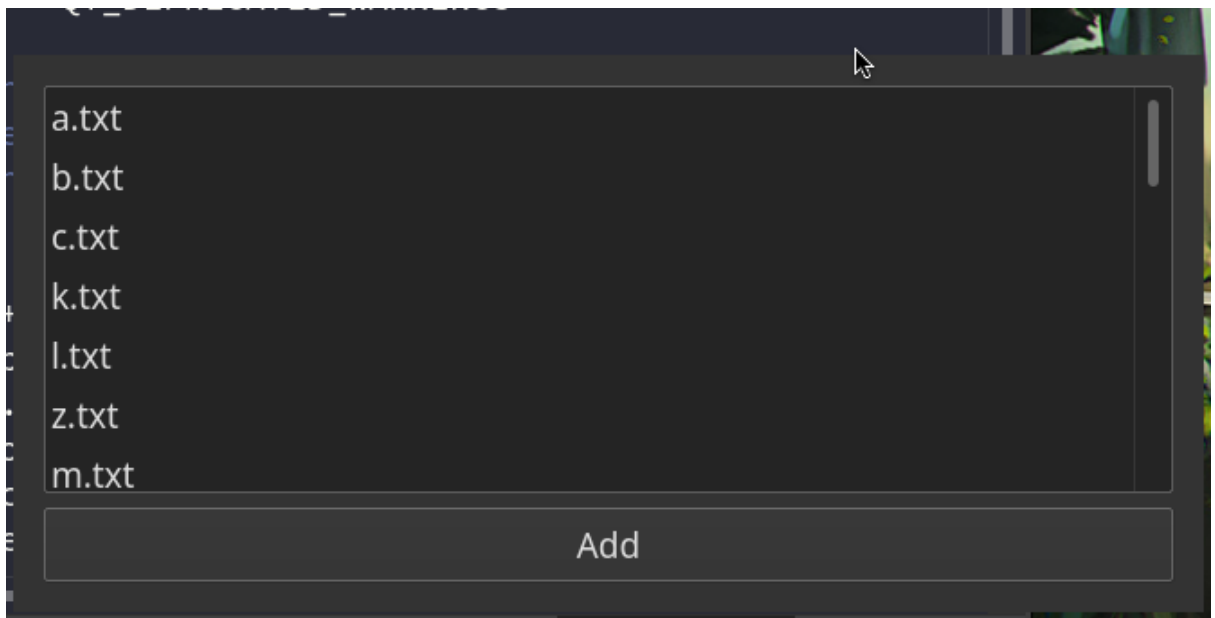
## 1. Temat

“Aplikacja do nauki pisania bezwzrokowego”

## 2. Interfejs Aplikacji

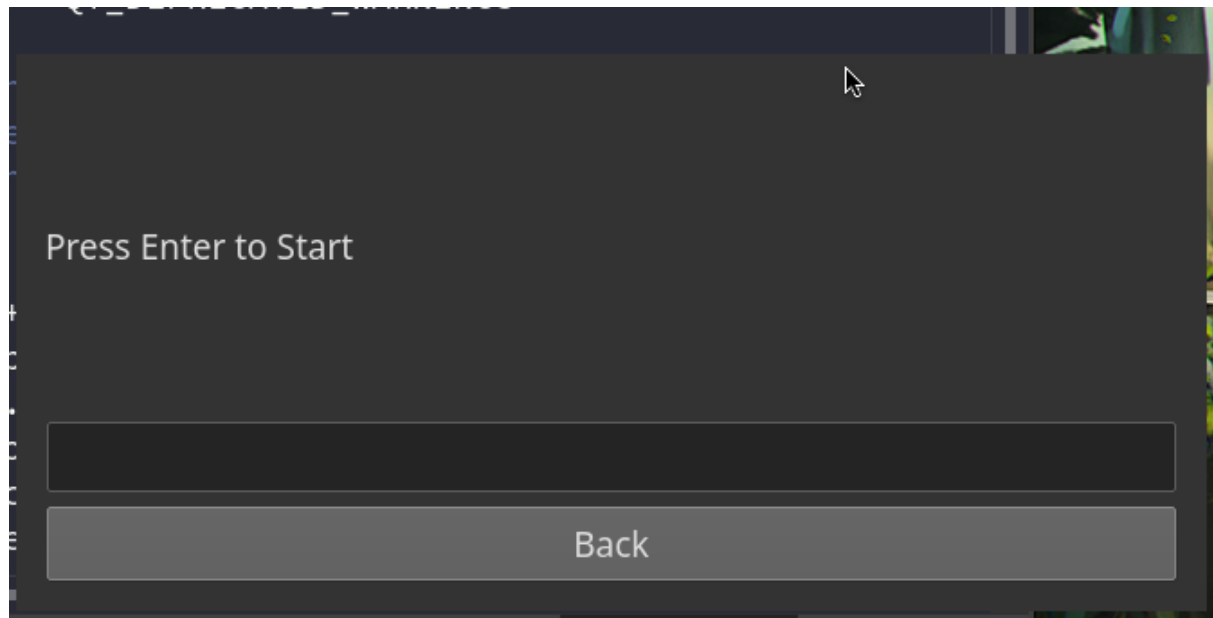
Gui ( przy pomocy Qtcreator )

- a. Możliwość wybrania przepisywanego tekstu w menu poprzez podwójne naciśnięcie wybranego tekstu.

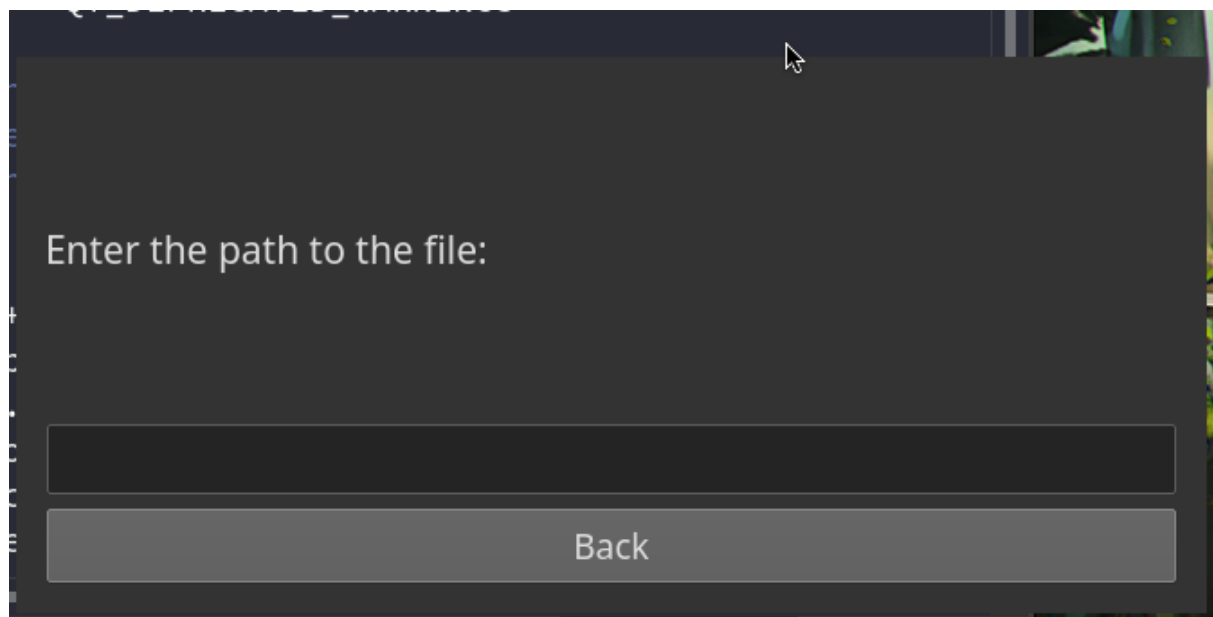


( zaznaczenie tekstu nie wywołuje żadnej reakcji, należy kliknąć dwukrotnie na wybraną pozycję menu)

- b. Możliwość przejścia do okna przepisywania tekstu



- c. Po zakończeniu pisania powinno wyświetlić się okno z informacją o uzyskanym wyniku w WPM.
- d. Możliwość przejście do okna dodawania pliku poprzez wciśnięcie Add



- e. W momencie wpisania błędnej ścieżki powinno wyskoć okno informujące o błędzie

### 3. Oczekiwany wynik działania programu

- a. przepisywanie tekstu, funkcjonalność reakcji programu na działania użytkownika typu jeśli napiszę słowo dobrze przejdź dalej itp.
- b. dodawanie plików do bazy danych, obsługa plików tekstowych, poprzez podanie ścieżki
- c. obsługa wyjątku spowodowanego podaniem błędnej ścieżki
- d. funkcjonalność czasomierza mierzącego czas pisania.

#### 4. Analiza obiektów występujących w trakcie działania aplikacji

Gui (Qt) :

dialog - klasa zawierająca w ciele elementy UI, zawiera informacje o prezentacji wyniku

Rzeczywista funkcjonalność programu (C++):

typingApp - obiekt która jedynie wywołuje inne funkcje, centrum sterowania aplikacją.

database - obiekt zarządzający oraz składający pliki w wektorze.

timer - obiekt reprezentujący funkcjonalność czasomierza

text - obiekt klasy typingApp reprezentujący wybrany przez użytkownika tekst przy pomocy Reader, zostanie wczytane z pliku do wektora.

reader - obiekt reprezentujący funkcjonalność czytnika danych

saver - obiekt reprezentujący funkcjonalność zapisu plików do bazy danych.

myexception - obsługa wyjątków, poprzez przeciążenie what

#### 5. Przykłady użycia

- a. wybranie tekstu, przepisanie go i uzyskania wyników prędkości pisania bezwzrokowego
- b. dodawanie plików do bazy

#### 6. Statyczna analiza typów i interfejsów poszczególnych modułów

typingApp - "wrapper" zarządza innymi metodami

Kompozycja:

database

text

timer

database

Agregacja:

saver (w metodzie)

text

Agregacja:

reader (w metodzie)

(Ważna uwaga saver jest wywoływany w database a reader w text ponieważ database ma własnego reader w postaci refresh database o kompletnie innej funkcjonalności)

myException

Dziedziczenie:

po std::exception