



AKADEMIA GÓRNICZO-HUTNICZA

Dokumentacja do projektu

Quiz Maker

z przedmiotu

Języki programowania obiektowego

Elektronika i Telekomunikacja, III rok

Mikołaj Telec

poniedziałek 14:40

prowadzący: Rafał Frączek

11.01.2021

1. Opis projektu

Aplikacja służy do generowania Quizu na podstawie pytań pobranych z pliku tekstowego QuizQuestions.txt . Pytania zostają przemieszane i tylko określona przez użytkownika ilość zostaje użyta. Quiz może zostać uruchomiony w trybie „EASY MODE” lub „HARD MODE”. Różnią się one jedynie systemem punktacji. Udzielone odpowiedzi są sprawdzane z kluczem odpowiedzi.

Punktacja:

- prawidłowa odpowiedź: +1 punkt,
- nieprawidłowa odpowiedź:
 - 0 punktów w trybie „EASY MODE”,
 - -1 punkt w trybie „HARD MODE”.
- pominięta odpowiedź: 0 punktów.

Po zakończeniu Quizu zostaje wyświetlony wynik użytkownika oraz informacje statystyczne: ilość prawidłowych, złych i pominiętych odpowiedzi. W przypadku wystąpienia złych odpowiedzi lub pominiętych pytań, wyświetlane są prawidłowe rozwiązania. Informacje na temat Quizu (tryb, data), imię użytkownika i jego wynik zostają zapisane w pliku tekstowym QuizResults.txt .

Aplikacja może być używana do sprawdzania wiedzy uczniów/studentów lub celów rozrywkowych.

2. Project description

The main goal of application is to generate Quiz based on questions loaded from QuizQuestions.txt. Questions get mixed up and only specified amount of them is used. Quiz may be started in two modes: “EASY MODE” or “HARD MODE”. They differ only in the scoring system. User answers are checked with answer key.

Scoring:

- correct answer: +1 point,
- wrong answer:
 - 0 points in „EASY MODE”,
 - -1 point in „HARD MODE”.
- skipped answer: 0 points.

After Quiz is finished, final result with statistical information are displayed. If wrong or skipped answers occurred, also correct answers are shown. Quiz information (mode and date), user name and his result are saved in QuizResults.txt.

Application may be used to test Students knowledge or for entertainment purpose.

3. Instrukcja użytkownika

W trakcie korzystania z aplikacji, użytkownik powinien podawać symbole odpowiadające danej opcji lub liczby z podanego zakresu. Użytkownik ma możliwość podjęcia ponownego wyboru w przypadku podania nieodpowiedniej wartości wejściowej.

Strukturę programu można podzielić na dwie części: Menu główne i Quiz. Obie części są opisane poniżej

I. Menu główne

Ta część pojawia się po uruchomieniu aplikacji. Użytkownik ma następujące opcje:

1. Quiz in EASY MODE

- a. Po wybraniu tej opcji mamy dwie możliwości:
 - i. Potwierdzenie decyzji
 1. Po potwierdzeniu decyzji mamy dwie możliwości:
 - a. Podanie ilości pytań, która ma zostać dodana do Quizu. Nie może przekraczać ilości pytań w pliku tekstowym. Gdy podamy ilość pytań, Quiz zostaje uruchomiony i nie mamy możliwości powrotu do menu głównego, dopóki nie odpowiemy na wszystkie pytania.
 - b. Powrót do wcześniejszego ekranu.
 - ii. Powrót do menu głównego.

2. Quiz in HARD MODE

- a. Schemat postępowania wygląda identycznie jak w przypadku opcji 1.

3. Exit

- a. Wybranie tej opcji powoduje zakończenie działania aplikacji.

II. Quiz

W tej części użytkownik na samym początku podaje swoje imię. Może to być dowolna sekwencja znaków. Następnie zostaje uruchomiony Quiz. Użytkownik odpowiada na wyświetlane pytania przy użyciu następujących znaków:

- „A”, „B”, „C”, „D” – odpowiadają one konkretnym odpowiedziom na pytanie,
- „0” – znak ten oznacza pominięcie pytania.

W przypadku podania nieodpowiedniej wartości wejściowej, użytkownik ma możliwość ponownego wyboru.

Gdy Quiz zostaje zakończony oraz wyniki zostaną wyświetlone, użytkownik wraca do miejsca w menu głównym, w którym był proszony o podanie ilości pytań.

4. Kompilacja

Do stworzenia projektu wystarczy standardowa kompilacja.

5. Pliki źródłowe

Projekt składa się z następujących plików źródłowych:

- *main.cpp*
 - główny plik źródłowy, w którym stworzone zostało menu główne aplikacji,
 - oprócz implementacji menu zawiera trzy funkcje:
 - *int question_file_test(void)* – sprawdza czy QuizQuestions.txt istnieje i zawiera odpowiednią ilość linii. Gdy plik będzie zawierał odpowiednią ilość linii zostanie zwrócona ich ilość. W przeciwnym przypadku zostanie zwrócona wartość 0, która spowoduje wyłączenie aplikacji.
 - *void mode_selection(string mode, int max_range)* – jest to część menu głównego. Używana do konfiguracji Quiz-u oraz stworzenia instancji klasy QuizMaker. Jako argumenty przyjmuje tryb: *mode* i ilość pytań w pliku: *max_range*.
 - *bool is_proper_value(string value, int max_range)* – sprawdza poprawność wartości wejściowej. Funkcja ta jest użyta w *mode_selection()*. Jako argumenty przyjmuje znak z klawiatury podany przez użytkownika: *value* i ilość pytań w pliku: *max_range*.
- *QuizMaker.h*, *QuizMaker.cpp* – deklaracja oraz implementacja klasy QuizMaker.

6. Zależności

Brak.

7. Opis klas

W projekcie utworzono następującą klasę:

1. QuizMaker – reprezentuje Quiz. Pozwala na pobranie pytań z pliku, wygenerowanie Quiz-u, jego przeprowadzenie oraz zapisanie wyniku w pliku tekstowym. Wszystkie atrybuty klasy są prywatne, a metody publiczne.
 - *QuizMaker(const string& quiz_mode)* – konstruktor klasy, który jako argument przyjmuje tryb Quiz-u: *quiz_mode*. Pobiera wszystkie pytania wraz z odpowiedziami z pliku QuizQuestions.txt.
 - *void quiz_generator(int question_amount)* – publiczna metoda klasy, która miesza pytania oraz wybiera z nich tylko ilość podaną przez użytkownika: *question_amount*. Jest ona podawana jako argument metody.
 - *void run(void)* – publiczna metoda klasy, która pobiera imię użytkownika, przeprowadza Quiz oraz wyświetla wynik końcowy.
 - *void save_results(void)* – publiczna metoda klasy, która zapisuje wynik użytkownika oraz informację na temat Quiz-u (tryb, data zakończenia) w pliku QuizResults.txt.
 - *bool valid_input(string input)* – publiczna metoda klasy, która sprawdza wartość wejściową *string input* podaną przez użytkownika podczas rozwiązywania Quiz-u. Gdy podany znak jest prawidłowy zwracana jest wartość *true*.

8. Zasoby

W projekcie wykorzystywane są następujące pliki zasobów:

- `QuizQuestions.txt` – plik tekstowy zawierający pytania oraz odpowiedzi. Aby program działał prawidłowo, struktura pliku powinna być następująca:
 - pierwsza linia: pytanie,
 - druga linia:
 - przykładowe odpowiedzi zapisana w formacie: „A. ... B. ... C. ... D. ...” ,
 - prawidłowa odpowiedź na pytanie zapisana w formacie: „CorrectAnswer:...”. Musi znajdować się za przykładowymi odpowiedziami.
 - ilość linii w pliku powinna być parzysta (dwie linie tworzą jedno pytanie). Niespełnienie tego warunku spowoduje wyłączenie aplikacji na samym początku.
- `QuizResults.txt` – plik tekstowy, w którym zapisywane będą wyniki. Plik jest tworzony automatycznie, więc jego wcześniejsze stworzenie jest opcjonalne.

9. Dalszy rozwój i ulepszenia

Projekt mógłbym ulepszyć poprzez:

- implementację mechanizmu losowania kolejności odpowiedzi,
- dodanie limitu czasowego,
- możliwość powrotu do pytań i poprawienie odpowiedzi,
- interfejs graficzny.

10. Inne

- Przygotowany przeze mnie plik z pytaniami „QuizQuestions.txt” zawiera jedynie 20 pytań.