Projekt nr 018
Przebiegi Czasowe

Autorzy:
Błażej Czaicki:
- Napisanie programu tworzącego dane “generator.exe”
- Wczytywanie danych z pliku

Bartosz Gawron:
- Pisanie dokumentacji
- Konwersja wczytanych danych

Jakub Smuga:
- Stworzenie GUI
- Implementacja metod rysujących wykresy

Opis projektu

W wielu dziedzinach nauki i techniki występuje potrzeba obserwacji przebiegów czasowych. Najprostszym, powszechnie znanym przykładem jest oscyloskop. Czasami zdarza się, że aparatura pomiarowa zapisuje w regularnych odstępach czasu interesujące nas dane do pliku. Celem projektu będzie napisanie programu do wizualizacji przebiegu zmienności pojedynczego parametru w czasie.

Założenia wstępne przyjęte w realizacji projektu

- Został stworzony program tworzący kolejne dane w odstępach 5 sekundowych i zapisujący je do pliku. Zostaje on automatycznie uruchomiony razem z programem głównym.
- Program na bieżąco wczytuje dane z pliku, a następnie je wyświetla w odpowiedni sposób.
- Program posiada przyciski ustalające sposób wyświetlania wykresu.
- Program rysuje osie oraz ma możliwość wyświetlanie na nich osi.

Analiza projektu

1. Specyfikacja danych wejściowych

Dane wczytywane z pliku mają formę:
- Każdy wiersz składa się z dwóch kolumn
- Kolumny oddzielone są spacjami
- Pierwsza kolumna zawiera czas wyrażony w sekundach
- Druga kolumna wartość parametru w danej chwili

2. **Opis oczekiwanych danych wyjściowych**

Dane powinny być móc wyświetlane w następujące sposoby:

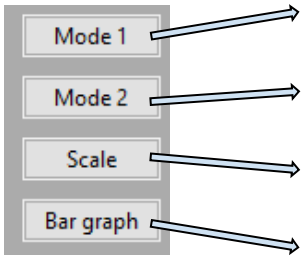
- Wykres liniowy ostatnio wczytanych danych
- Wykres liniowy ostatnio wczytanych danych oraz przedostatnio wczytanych danych
- Wykres słupkowy ostatnio wczytanych danych
- Wykres słupkowy ostatnio wczytanych danych oraz przedostatnio wczytanych danych
- Istnieje możliwość wyświetlenia podziałki na tle wykresu

3. **Zdefiniowanie struktur danych**

Dane wczytane w pliku przechowywane są w *std::vector*

4. **Specyfikacja interfejsu użytkownika**

Okno podzielone jest na dwa obszary. Na jednym wyświetlany jest wykres, na drugim znajdują się przyciski którymi użytkownik ma możliwość zmiany wyświetlanego wykresu. Kolejne przyciski zapewniają funkcjonalności:



Wyświetlanie tylko ostatnio wczytanych danych

Wyświetlanie ostatnio oraz przedostanio wczytanych danych

Wyświetlenie podziałki na osiach

Zmiana wykresu z liniowego na słupkowy oraz odwrotnie

5. **Wyodrębnienie i zdefiniowanie zadań**

- Napisanie programu generującego dane
- Wczytywanie danych z pliku
- Stworzenie struktury przechowującej wczytane dane
- Konwersja danych do wyświetlenia
- Wyświetlanie Osi
- Wyświetlanie podziałki na osiach
- Obsługa przycisków
- Implementacja funkcji rysującej
- Implementacja funkcji tworzących odpowiednie wykresy

6. **Decyzja o wyborze narzędzi programistycznych**

Do projektu użyliśmy biblioteki wxWidgets. Zdecydowaliśmy się na tą bibliotekę, ponieważ przy pomocy aplikacji wxFormBuilder w bardzo wygodny sposób mogliśmy zaprojektować interfejs. W zadaniu używaliśmy standardu C++14. Projekt został przygotowany w programie Visual Studio 2022.

Podział pracy i analiza czasowa

Większość projektu zrobiliśmy wspólnie. Na początku stworzyliśmy projekt w wxFormBuilder. Dodaliśmy wszystkie potrzebne przyciski i suwaki. Następnie zajęliśmy się stworzeniem programu generującego dane. Posiadając już taki program zajęliśmy się

wczytaniem danych z pliku oraz ich analizą. Kolejnym krokiem było stworzenie funkcji rysujących wykres na podstawie przygotowanych danych oraz obsługa przycisków.

Opracowanie i opis niezbędnych algorytmów

Algorytm wczytywania danych opierał się o wczytywanie danych z pliku “dane.dat” w interwale czasu równym 1 sekundzie. Sprawdzamy, czy wczytane dane różnią się od przechowywanych obecnie w pamięci, jeśli tak to wywołujemy metodę Repaint odpowiedzialną za przerysowanie wykresu.

Algorytm wyliczania pixeli, odpowiadających następnemu punktowi na wykresie. Opiera się o maksymalną liczbę danych oraz obecną szerokość i wysokość okna.

Kodowanie

Opis Klas:

- **MyApp** niezbędna klasa do uruchomienia programu
- **RenderTimer** klasa odpowiadająca za wczytywanie danych w odpowiednim odstępie czasu
- **DataAnalyzer** klasa wczytująca dane z pliku
- **Data** klasa przechowująca dane
- **GUI** klasa generowana przez program wxFormBuilder, odpowiada za interfejs użytkownika
- **MyProject1MyFrame1** klasa odpowiadająca za rysowanie wykresów

Opis Zmiennych:

- **DataAnalyzer::newData** przetrzymuje ostatnio wczytane dane
- **DataAnalyzer::oldData** przetrzymuje przedostatnio wczytane dane
- **DataAnalyzer::DATA_SIZE** zmienna ustalająca ilość wczytywanych danych
- **Data::dataX** wektor przechowujący czas pomiaru
- **Data::dataY** wektor przechowujący wartość pomiaru

Opis funkcji:

- **DataAnalyzer::ReadData m**etoda wczytująca dane z pliku
- **Data::getMin** zwraca minimalną wartość dataY
- **Data::getMax** zwraca maksymalną wartość dataY
- **Data::getPixelY** zwraca wartość na osi Y, na której powinien być narysowany punkt
- **MyProject1MyFrame1::DrawScale** rysuje podziałkę na osiach
- **MyProject1MyFrame1::DrawAxes** rysuje osie
- **MyProject1MyFrame1::DrawChart** rysuje wykres liniowy
- **MyProject1MyFrame1::BarGraphDrawChart** rysuje wykres słupkowy
- **MyProject1MyFrame1::Repaint()** wywołuje inne funkcje rysujące, wywoływana za każdym razem gdy zaistnieje konieczność przerysowania
- **RenderTimer::Notify** wywołuje metodę Repaint
- **RenderTimer::start** włącza odliczanie czasu do kolejnego wywołania Notify

Testowanie

Działanie naszego programu testowaliśmy manualnie, testując różne możliwe scenariusze, w postaci m. in. kombinacji ustawień przycisków.

Wdrożenie, raport i wnioski

W programie udało nam się wykonać wszystkie wymagania podstawowe zatem dane są wczytywane z pliku w regularnych odstępach czasu (1s), a generowane co (5s). Dostępne są dwa tryby pracy wyświetlające ostatni przebieg danych lub ostatni i przedostatni. Jest również możliwość wyświetlenia skali na osiach.

Z wymagań rozszerzonych dodaliśmy możliwość wyświetlania wykresu słupkowego, wyświetlenie całej historii przebiegu nie zostało przez nas zrealizowane ze względu na niejasność w sposobie przedstawienia całego przebiegu na wykresie, w jednym momencie.