# Projekt: Program graficzny pokazujący szybkość działania ASM w stosunku do C++

## Autor: Kornel Domeradzki

### Informatyka I stopnia III sem., 2018/19

# Cel Projektu

Celem projektu było napisanie programu graficznego z użyciem WinAPI oraz OpenGL, który pokaże różnice czasu, w wykonaniu pewnych czynności przy użyciu assembly oraz C++.

# Założenia

1. Program graficzny rysujący wykresy wybranej przez użytkownika z menu funkcji.
2. Wartości funkcji są zawsze obliczane na początku w języku C++ a następnie w ASM (W przypadku windowsa MASM)
3. Wykresy funkcji przedstawiają ile czasu zajmuje programowi obliczenie wartości funkcji *f(x)* dla argumentu *x*
4. Wykresy dla czasów obliczonych przy użyciu C++ są kolorowane kolorem niebieskim a dla ASM kolorem zielonym

# Ograniczenia

Nie ma żadnych ograniczeń co do rozdzielczości ekranu tak długo jak monitor użytkownika posiada rozdzielczość większą niż co najmniej 500x250. Rozdzielczość ekranu wykresu można zmienić poprzez zmianę danych w pliku konfiguracyjnym.

Dostępne jest tylko parę spredefiniowanych przypadków (funkcji) do sprawdzenia. Nie został zaimplementowany parser równań/funkcji, który umożliwiłby narysowanie wszelkich wykresów.

Program ze względu na MASM może operować jedynie na systemie operacyjnym Windows.

# Wstęp

Program po włączeniu wyświetla menu z przyciskami do rysowania wszystkich możliwych funkcji oraz z jednym przyciskiem otwierającym okno z informacją na temat programu.

Kliknięcie jednego przycisku umożliwiającego rysowanie funkcji pewnej czynności od czasu powoduje otworzenie nowego okna na prawo. Okno może być przez pewien czas białe i nie mieć nic nie narysowanego (oznacza to, że wciąż przetwarza pewne informacje). Po wykonaniu obliczeń w lewym górnym rogu powinna się pokazać informacja na temat czasu, ile zajęło obliczenie **wszystkich** obliczeń w danym języku programowania a po środku powinien znajdować się wykres obrazujący ile czasu zostało poświęcone na obliczenie argumentu *x* przez funkcję *f(x)*.

# Implementacja

Przy tworzeniu projektu skorzystano z Windows API w celu utworzenia okna menu oraz należących do okna przycisków i podokien. Do rysowania wykresów użyty został silnik do gier 2D stworzony przeze mnie przy pomocy biblioteki SFML (Binding OpenGL).

Struktura silnika wygląda następująco:

* App Engine
  + App *(Moduł obsługujący całą resztę silnika)*
  + Asset Manager *(Moduł odpowiadający za ładowanie i usuwanie tekstur, dźwięków i wielu innych rzeczy z pamięci)*
  + State Machine *(Moduł odpowiadający za zarządzanie stanami gry lub w tym przypadku aplikacji)*
  + Input Manager *(Moduł umożliwiający sprawdzenie interakcji użytkownika z wyrenderowanymi przedmiotami)*
* App States *(Zawiera stany programu)*
* App Custom Objects *(Zawiera różne przedmioty wykorzystywane w programie np. Customowe obiekty do rysowania na ekranie)*

Wspomniane wyżej *stany programu* są to opisane poprzez kod źródłowy momenty programu – np. Rysowanie grafu jest stanem. Umożliwia nam to podział programu na podprogramy w takim samym stopniu jak funkcje w programowaniu deklaratywnym.

Nowymi obiektami dodanymi do modułu *App Custom Objects* są DiagramGrid (Oś X oraz Y wykresu), Fibonacci (Wykres, który ukazujący jak długo zajmuje obliczenie x-tego argumentu funkcji Fibonacci w C++ lub MASM; Implementuje klasę abstrakcyjną FunctionGraph), NWD (To samo co wcześniej ale dla NWD), SineGraph(To samo co wcześniej ale dla Sinusa), TanGraph(To samo co wcześniej ale dla Tangensa) oraz FunctionGraph (Klasa abstrakcyjna z paroma wirtualnymi funkcjami).

# Przykład działania programu

1. Po odpaleniu programu po prawo możemy zauważyć wykres argumentów od czasu (s).

