

# Pole przyspieszeń

Jakub Mikusek  
Kacper Tracz  
Marcelin Raczek

28 czerwca 2023

*Wszystkie załączone grafiki dostępne są w folderze DOC*

## 1 Opis projektu

Przedstawiony projekt jest programem służącym do zobrazowania pola przyspieszeń cząstki na podstawie funkcji ukształtowania terenu. Funkcja jest podawana w postaci  $h(x,y,t)$  gdzie  $x,y$  to współrzędne funkcji ukształtowania terenu a  $t$  to czas.

## 2 Założenia

We wprowadzanej funkcji mogą występować tylko zmienne  $x$ ,  $y$  oraz  $t$ . Wprowadzane funkcje muszą posiadać znane funkcje matematyczne

## 3 Analiza projektu

### 3.1 Dane wejściowe

Użytkownik wprowadza łańcuch znaków reprezentujący funkcję ukształtowania terenu. Domyślnie funkcja ta to  $x^2 + y^2$ .

### 3.2 Dane wyjściowe

Program rysuje w oknie pole wektorowe bądź konturowe w postaci bitmapy w zależności od wyboru użytkownika. Dodatkowo, po kliknięciu odpowiedniego przycisku, przygotowuje plik z rozszerzeniem `.bmp` zawierającym oba te pola.

### 3.3 Struktury danych

W programie użyta została struktura bitmapy w celu przechowywania obrazów pola wektorowego i konturowego. Użyte zostały również tablice np. w postaci typu `string` będącego tablicą znaków.

### 3.4 UI

Uruchamiając program użytkownik widzi okno, w którym ma możliwość:

- wpisania funkcji ukształtowania terenu
- podania wartości czasu (t)
- wyboru pomiędzy widokiem mapy konturowej oraz mapy wektorowej
- zapisu map do pliku

### 3.5 Wyodrębnienie i zdefiniowanie zadań

- Wygenerowanie kodu źródłowego odpowiedzialnego za okno oraz sizery przy użyciu wxFormBuilder
- Obsługa inputu z użyciem biblioteki tinyexpr w postaci łańcucha znaków reprezentującego funkcję ukształtowania terenu
- Rysowanie ukształtowania terenu na podstawie funkcji
- Rysowanie pola wektorowego
- Przygotowanie pliku w formacie .bmp do zapisu

### 3.6 Narzędzia programistyczne

Podczas projektu użyte zostały następujące narzędzia:

- IDE: Visual Studio przez wzgląd na prostsze zarządzanie wxWidgets.
- Biblioteki: wxWidgets jako domyślne narzędzie wymagane do stworzenia projektu oraz tinyexpr jako biblioteka implementująca zamianę łańcucha znaków na funkcję.
- Dodatkowe narzędzia: wxFormBuilder w celu wygenerowania kodu odpowiedzialnego za okno, Doxygen w celu wygenerowania dokumentacji.

## 4 Podział pracy i estymata czasowa

wygenerowanie okna oraz rysowanie ukształtowania terenu - Jakub Mikusek  
rysowanie pola wektorowego, obsługa tinyexpr - Kacper Tracz  
dokumentacja - Marcelin Raczek

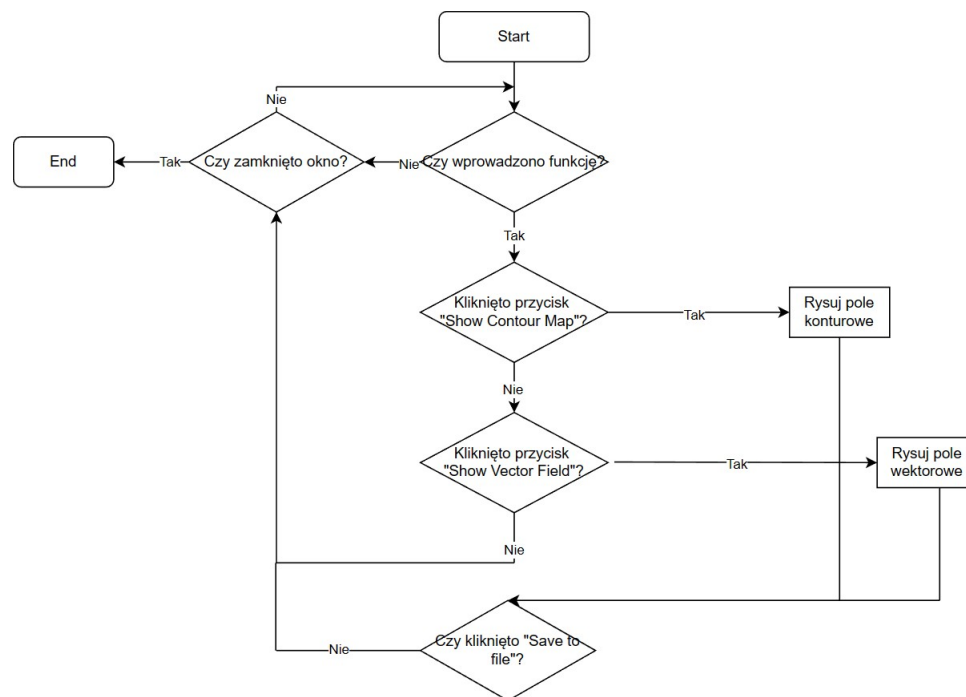
## 5 Algorytmy

Poprzez bibliotekę tinyexpr został wykorzystany algorytm ewaluacji wyrażenia w drzewo składni. Drzewo to ma charakterystykę podobną do binarnego drzewa poszukiwań - wartości poprzedzające znak są lewym dzieckiem znaku a wartości późniejsze są prawym dzieckiem.

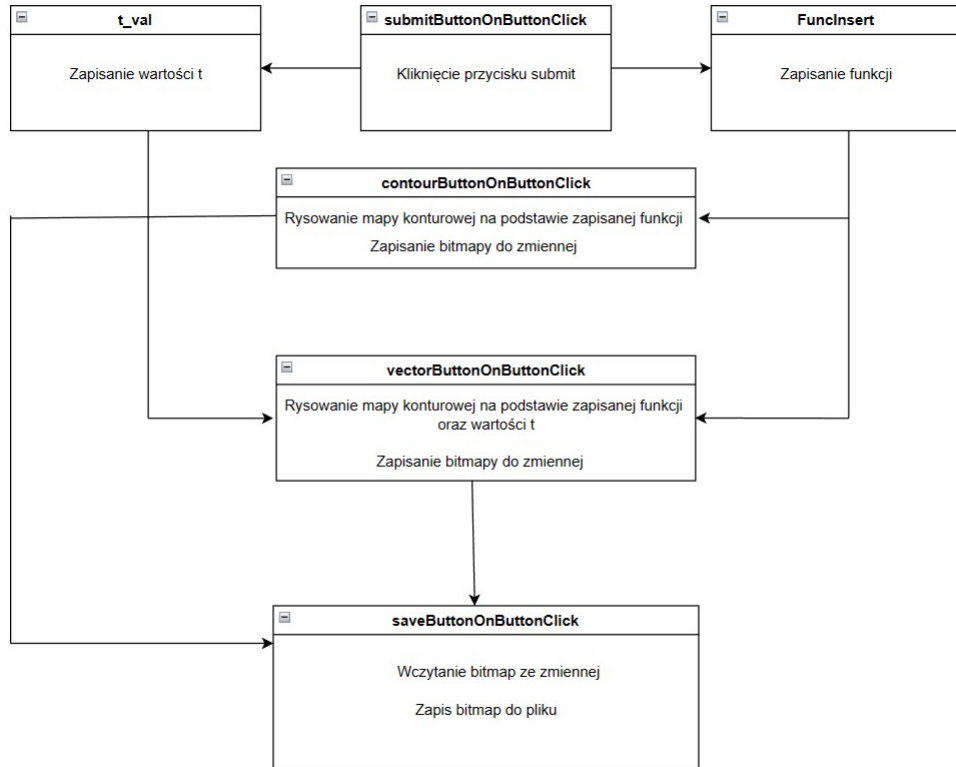
Jako ciekawostkę można uwzględnić pierwotne podejście do problemu zamiany łańcucha znaków na funkcję matematyczną. Pierwotnie przygotowana została

funkcja używająca algorytmu Shunting yard ewaluująca input w postaci łańcucha znaków do zapisu w postaci RPN (odwróconej notacji polskiej).

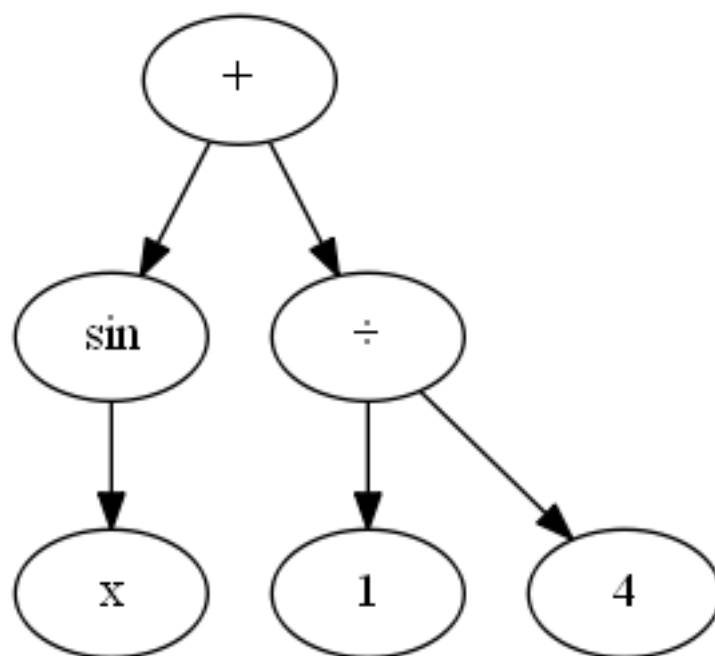
## 6 Kodowanie



Rysunek 1: Schemat blokowy



Rysunek 2: Diagram przepływu danych



Rysunek 3: Drzewo składni dla wyrażenia  $\sin(x) + \frac{1}{4}$  Źródło

Szczegółowy opis klas funkcji i zmiennych zawarty jest w dokumentacji sporządzonej przy użyciu doxygena.

## 7 Testowanie

### 7.1 Testy niezależnych bloków

W ramach testów funkcji rysujących pole wektorowe oraz mapę konturową, w kodzie zostały "hardcodowane" poszczególne funkcje oraz wartości mające służyć do rysowania owych pól.

Testy algorytmu zamieniającego łańcuch znaków na funkcję, wykorzystane zostało wypisywanie wartości funkcji dla danych  $x$  oraz  $y$  na ekran.

Test zapisu do pliku został przeprowadzony poprzez zapisywanie i poprawne wyświetlanie utworzonego pliku.

### 7.2 Testy powiązań bloków

Powiązania występują pomiędzy zamianą łańcucha znaków na funkcję oraz rysowaniem pól oraz pomiędzy rysowaniem pól a zapisem do pliku.

Testy pierwszego powiązania zostały przeprowadzone poprzez porównanie narysowanego pola dla wpisanej wartości funkcji z polem narysowanym dla "hardcodowanej" wartości funkcji.

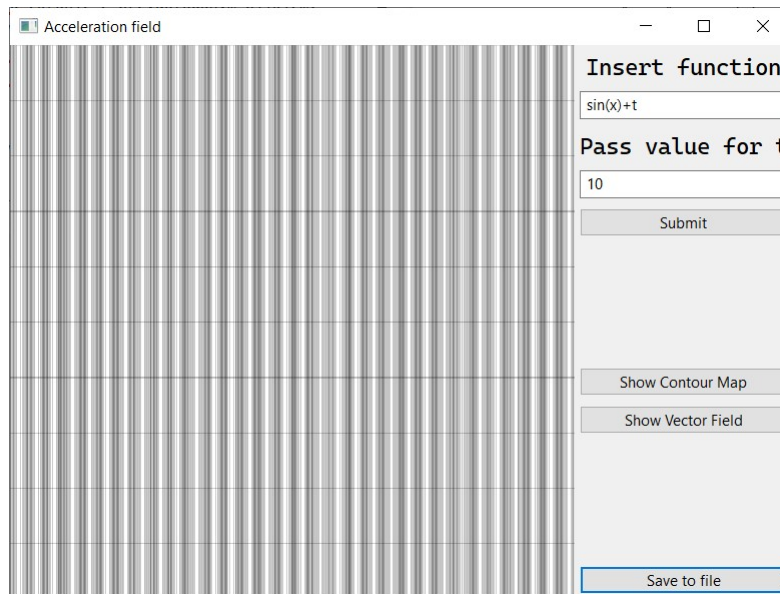
Test drugiego powiązania został przeprowadzony poprzez porównanie pól rysowanych na ekranie z przygotowanym plikiem.

### 7.3 Testy całościowe

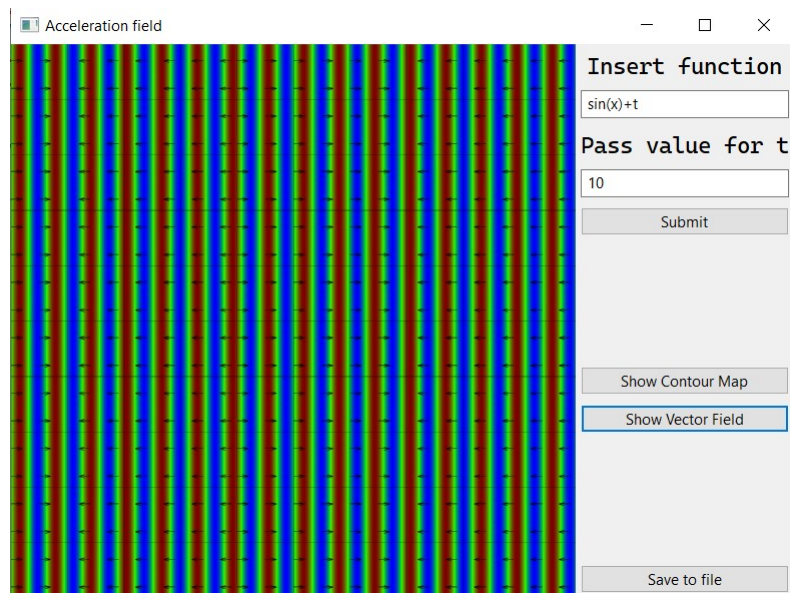
Testy były przeprowadzane poprzez uruchamianie programu i sprawdzanie każdej funkcjonalności poprzez interfejs użytkownika, oraz porównanie wyników z outputem programu w przypadku kiedy poszczególne elementy były bezpośrednio zmieniane oraz wywoływane w kodzie.

## 8 Wdrożenie raport i wnioski

Program działa zgodnie z założeniem podstawowej wersji. Program mógłby zostać rozszerzony o możliwość animacji pola wektorowego.



Rysunek 4: Pole konturowe dla danych wejściowych  $\sin(x)+t$



Rysunek 5: Pole wektorowe dla danych wejściowych  $\sin(x)+t$ ,  $t=10$