

Wykresy 3D

Projekt numer 7

Zuzanna Sulima

Weronika Wronka

Piotr Kowalczyk

Spis treści:

1. [Tytuł projektu i jego autorzy](#)
2. [Opis projektu](#)
3. [Złożenia wstępne przyjęte w realizacji projektu](#)
4. [Analiza projektu](#)
 - 4.1. [Specyfikacja danych wejściowych](#)
 - 4.2. [Opis oczekiwanych danych wyjściowych](#)
 - 4.3. [Zdefiniowanie struktur danych](#)
 - 4.4. [Specyfikacja interfejsu użytkownika](#)
 - 4.5. [Wyodrębnienie i zdefiniowanie zadań](#)
 - 4.6. [Decyzja o wyborze narzędzi programistycznych](#)
5. [Podział pracy i analiza czasowa](#)
6. [Opracowanie i opis niezbędnych algorytmów](#)
7. [Kodowanie](#)
8. [Testowanie](#)
9. [Wdrożenie, raport i poprawki](#)

1. Tytuł i autorzy projektu

Wykonano projekt pod tytułem „Wykresy 3D” o numerze 7, którego autorami są:

- Zuzanna Sulima
- Weronika Wronka
- Piotr Kowalczyk

2. Opis projektu

Celem projektu było stworzenie programu umożliwiającego rysowanie funkcji 3D dwóch zmiennych. Użytkownik może podać funkcję analitycznie wybierając ją spośród kilku dostępnych możliwości lub wczytać współrzędne x , y oraz wartość funkcji z pliku tekstowego. Otrzymany wykres może zostać wyświetlony w postaci rzutu perspektywicznego lub mapy konturowej, a następnie zostać zapisany lub wydrukowany.

3. Złożenia wstępne przyjęte w realizacji projektu

Kierując się wcześniej zdobytym doświadczeniem do wykonania projektu zastosowano bibliotekę wxWidges. Powstała aplikacja umożliwia narysowanie wykresów 3D. Użytkownik może wybrać:

- Obszary zmienności współrzędnych x , y oraz z
- Sposób wyświetlania wykresu: mapa konturowa lub rzut perspektywiczny
- Postać funkcji: analityczna wybrana z listy lub wczytana z pliku tekstowego
- Drukowanie pliku
- Zapis pliku w formacie png, jpg lub bmp

4. Analiza projektu

4.1 Specyfikacja danych wejściowych

Danymi wejściowymi przyjmowanymi przez program są: funkcja postaci $f(x,y)$ oraz obszary zmienności współrzędnych x , y oraz z .

Plik tekstowy zawierający dane wejściowe musi zostać sformatowany wg podanych zasad:

- W pliku współrzędne muszą znajdować się w kolejności x , y , z
- Każda współrzędna musi być oddzielona od poprzedniej spacją lub tabulatorem
- Plik musi zawierać na swoim końcu znak EOF

4.2 Opis oczekiwanych danych wyjściowych

Użytkownik ma możliwość pobrania pliku zawierającego wykres 3D wybranej wcześniej funkcji w formacie png, jpg lub bmp.

4.3 Zdefiniowanie struktur danych

Dane oraz funkcje potrzebne do poprawnego działania programu przechowywane są w obiektach klas MyApp, MyFrame1, GUIMyFrame1, ConfigClass oraz ChartClass.

4.4 Specyfikacja interfejsu użytkownika

Interfejs ma postać okna utworzonego w aplikacji wxFormBuilder. Po prawej stronie znajduje się menu: lista dostępnych funkcji, możliwość wpisania wybranych zakresów zmiennych oraz przyciski umożliwiające zapis lub wydrukowanie obrazu. Natomiast lewą stronę okna stanowi wyświetlany wykres.

4.5 Wyodrębnienie i zdefiniowanie zadań

Planując sposób realizacji projektu zdecydowano się na podział na mniejsze, niezależne moduły:

- Stworzenie interfejsu
- Opracowanie algorytmu rysowania funkcji w postaci rzutu perspektywicznego
- Opracowanie algorytmu tworzenia mapy konturowej danej funkcji
- Opracowanie algorytmu rysowania funkcji na podstawie danych z pliku
- Implementacja dodatkowych funkcjonalności, takich jak: zapis i drukowanie
- Opracowanie dokumentacji opisującej projekt

4.6 Decyzja o wyborze narzędzi programistycznych

Zdecydowano, aby przy realizacji projektu korzystać z środowiska Visual Studio, a zastosowaną biblioteką była biblioteka wxWidgets. Powyższe narzędzia wybrano ze względu na dostarczane funkcjonalności oraz doświadczenie w ich stosowaniu zdobyte podczas tego semestru.

5. Podział pracy i analiza czasowa

Po wspólnym omówieniu sposobu realizacji projektu, zdecydowano o podziale na mniejsze moduły ułatwiające niezależną pracę.

Zuzanna Sulima	Rysowanie rzutu perspektywicznego Drukowanie Zapisywanie jako obrazu
Weronika Wronka	Przygotowanie interfejsu użytkownika, opracowanie dokumentacji
Piotr Kowalczyk	Rysowanie mapy konturowej Rysowanie na podstawie danych z pliku Obsługa eventów

6. Opracowanie i opis niezbędnych algorytmów

Algorytm rysowania rzutu perspektywicznego:

1. Oblicz Δx oraz Δy
2. Oblicz wartość funkcji z dla każdego punktu z zadanego przedziału zmienności z z krokiem Δx oraz Δy i dodaj te punkty do vectora
3. Narysuj punkty po przemnożeniu przez odpowiednie macierze przekształceń oraz połącz sąsiednie punkty prostą

Algorytm rysowania mapy konturowej:

1. Oblicz Δx oraz Δy .
2. Oblicz wartość funkcji z dla każdego punktu z zadanego przedziału zmienności z z krokiem Δx oraz Δy
3. Dodaj do vectora tylko te punkty, w których funkcja osiąga wartości całkowite
4. Narysuj punkty po przemnożeniu przez odpowiednie macierze przekształceń oraz połącz je prostą z nieodległymi punktami

Algorytm rysowania z pliku:

1. Dla kolejnych trójek współrzędnych x, y, z stwórz obiekt Point i dodaj go do vectora
2. Narysuj punkty po przemnożeniu przez odpowiednie macierze przekształceń oraz połącz je prostą z nieodległymi punktami

7. Kodowanie

Klasa MyFrame1 – dziedziczy po klasie wxFrame z biblioteki wxWidgets. Znajdują się w niej elementy oraz deklaracje metod potrzebne do działania interfejsu użytkownika

- **mainFormClose(event)** – funkcja zamykająca okno
- **WxPanel_Repaint(event)** – funkcja rysująca, w której wywołują się funkcje Draw oraz Contour w zależności od wybranego trybu rysowania

- **kontur_click(event)** – zmiana trybu rysowania wykresu na mapę konturową
- **rzutPersp_click(event)** – zmiana trybu rysowania wykresu na rzut perspektywiczny
- **wczytajDane_click(event)** – funkcja wczytująca dane z pliku (współrzędne x, y oraz wartość funkcji $f(x,y)$)
- **zapisz_click(event)** – funkcja zapisująca narysowany wykres do pliku png, jpg lub bmp
- **drukuj_click(event)** – funkcja pozwalająca na wydrukowanie narysowanego wykresu funkcji
- **funkcja_pick(event)** – funkcja umożliwiająca wybranie wykresu jednej z funkcji wbudowanych w program
- **WxEdit_(x0,x1,y0,y1,z0,z1)_Update(event)** – funkcja umożliwiająca zmianę zakresu rysowanego wykresu funkcji w trzech płaszczyznach (x, y i z)

Klasa ConfigClass - klasa, której obiekt przechowuje wybór funkcji oraz zakres zmiennych

Klasa ChartClass - klasa zawierająca zmienne i funkcje potrzebne do rysowania wykresów

- **x_step, y_step** – zmienne definiujące ilość segmentów, na które podzielony zostanie przedział zmienności x oraz y
- **GetFunctionValue(x,y)** – funkcja obliczająca wartość z w zależności od funkcji wybranej przez użytkownika
- **line2d(t,x1,y1,x2,y2,w,h,dc)** – funkcja rysująca linie w 2D (wykorzystywana przy mapie konturowej)
- **line3d(t,x1,y1,z1,x2,y2,z2,w,h,dc)** – funkcja rysująca linie w 3D (wykorzystywana przy rzucie perspektywnym)
- **Draw(dc,w,h,t,points)** – funkcja rysująca wykres funkcji w rzucie perspektywnym
- **Contour(dc,w,h,t,points)** – funkcja rysująca mapę konturową
- **DrawFromFile (dc,w,h,t,points)** – funkcja rysująca wykres z danych wczytanych z pliku

Klasa GUIMyFrame1 – klasa dziedzicząca po klasie MyFrame1. Znajdują się w niej definicje metod zadeklarowanych w MyFrame1

- **Repaint()** – funkcja wywołująca odpowiednią funkcję rysującą w zależności od wybranych opcji

Klasa MyApp – klasa dziedzicząca po klasie wxApp z biblioteki wxWidgets. Występuje w mainie i pozwala zaimplementować aplikację

Klasy Vector, Vector4, Matrix, Matrix4, Point – klasy pomocnicze do przechowywania danych/macierzy transformacji

8. Testowanie

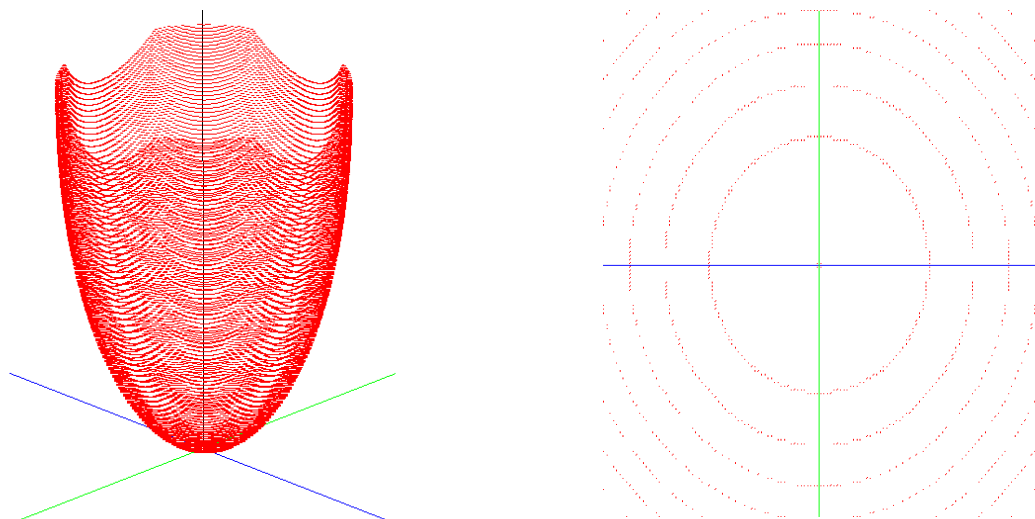
Program testowano manualnie, sprawdzając kolejne zaimplementowane funkcjonalności.

Sprawdzono, poprawność działania drukowania oraz zapisu do pliku. Następnie testowano możliwość zmiany ustawień użytkownika na przykład:

- Zmiana sposobu prezentacji wykresu z mapy konturowej na rzut perspektywiczny lub odwrotnie
- Zmiana zakresów współrzędnych
- Wybór nowej funkcji przy pozostawieniu pozostałych ustawień bez zmian

9. Wdrożenie, raport i wnioski

Stwierdzono niestety, że zaimplementowane sposoby prezentacji wykresów oraz map konturowych dla bardzo dużych obszarów zmienności x oraz y spowalniają działanie programu. Pozostałe funkcjonalności części podstawowej zostały zrealizowane.



Rysunek 1: Przykładowe wykresy 3D dla funkcji $x^2 + y^2 = z$ gdzie $x \in (-2, 2)$ $y \in (-2, 2)$ $z \in (0, 5)$ w postaci rzutu perspektywicznego oraz mapy konturowej