

GFK Krzywe Lissajous Dokumentacja

Aleksander Gawel, Dominik Smęda, Kamil Krzysztofek

2 lipiec 2023

1 Założenia wstępne przyjęte w realizacji projektu

Głównym założeniem wstępnym było ustalenie, jak będzie przebiegała współpraca oraz że nie będzie ona przebiegała w sposób skomplikowany - od początku przewidywalne było że podział ról będzie jasny i przejrzystym, a także do pewnego stopnia co będzie trzeba wykorzystać i na czym bazować.

2 Specyfikacja danych wejściowych

Aby narysować krzywe, należało skorzystać z następujących wzorów na współrzędne x, y, z (lub r, ϕ, θ):

$$x = A * \sin(a * t + \text{delta}), \quad (1)$$

$$y = B * \sin(b * t), \quad (2)$$

$$z = C * \sin(c * t + \text{phi}). \quad (3)$$

Współczynniki A,B,C odpowiadają za przesunięcie wzdłuż danej osi, współczynniki a,b,c za "zagęszczenie" krzywych a delta oraz phi za przesunięcie w fazie. Innymi danymi, które można wprowadzić są: obrót wokół osi X,Y,Z, wybór rysowania krzywej w sposób ciągły lub punktowy (można wybrać ilość punktów), wybór animacji oraz rozpoczęcie/zatrzymanie jej, a także długość animowanej krzywej.

3 Opis oczekiwanych danych wyjściowych

Na wyjściu znajduje się wyrysowana zgodnie z podanymi wzorami i wprowadzonymi parametrami krzywa. Wybór pozostałych opcji powoduje również oczekiwany efekt - krzywa jest obranaca wokół danej osi, widok krzywej jest ciągły lub punktowy, z wybraną liczbą punktów, a animacja jest od-
twarzana zgodnie z życzeniem, a także jest zatrzymywana po wciśnięciu odpowiedniego przycisku, a długość animowanej krzywej zależy od podanej wartości.

4 Zdefiniowanie struktur danych

Pliki vecmat.h i vecmat.cpp zawierają proste klasy definiujące wektor i macierz, a także operatory dla nich.

Plik SquareBox.h zawiera strukturę, w której zawarte są punkty będące wierzchołkami sześcianu, który służy jako siatka (grid), a także opisuje je wraz z parametrami A,B,C.

Plik LissajousCurve.h zawiera strukturę, która przedstawia parametry użyte w równaniach 1,2,3, same równania, a także przeprowadza "update", który umożliwia zmianę parametrów i przez to danych wyjściowych.

Pliki GUIMyFrame1.h, GUIMyFrame1.cpp, GUI.h, GUI.cpp służą utworzeniu graficznego interfejsu użytkownika i przystosowaniu go do działania programu.

Plik AxisLines.h służy utworzeniu osi układu współrzędnych.

5 Specyfikacja interfejsu użytkownika

W interfejsie użytkownika można wybierać parametry na różne sposoby, aby w sposób bardziej adekwatny i wygodny dobierać daną wartość. Wartości A,B,C oraz a,b,c można wpisać ręcznie z klawiatury, wartości delta,phi oraz obrót wokół osi można dostosować suwakiem, oprócz tego można wybrać opcję "Line" lub "Points" w celu wyrysowania krzywej w sposób ciągły lub punktowy, a także z klawiatury wpisać liczbę punktów. Poniżej można zaznaczyć opcję "Animation", która umożliwi wykonanie animacji, a także przycisk "Start/Stop", który uruchamia lub zatrzymuje animację. Wpisując wartość na klawiaturze można również dostosować długość animowanej krzywej w polu "Line length". Na dole znajduje się również opcja wyboru współrzędnych x,y,z lub r, ϕ , θ .

6 Wyodrębnienie i zdefiniowanie zadań

Projekt został podzielony na 4 główne zadania: zaprojektowanie GUI, wyrysowanie krzywych Lissajous, stworzenie animacji oraz opracowanie dokumentacji. W trakcie okazało się jednak, że niektóre mniejsze zadania musiały zostać wykonane wspólnie lub przekazanie innej osobie. Konieczne stało się wyodrębnienie następujących zadań: dodanie do rysunku ramki i osi oraz dopilnowanie, by update'y przebiegały w sposób płynny.

7 Decyzja o wyborze narzędzi programistycznych

Wybór narzędzi należał do dość oczywistych - zadanie nie wymaga używania dodatkowych narzędzi poza Visual Studio 2022, języka C++, biblioteki wxWidgets oraz GitHuba, który służył ułatwieniu wspólnej pracy nad niektórymi plikami oraz łatwiejszy dostęp do nich dla wszystkich.

8 Podział pracy i analiza czasowa

Początkowo za zaprojektowanie GUI odpowiadał Kamil Krzysztofek, wyrysowaniem krzywych zajął się Dominik Smęda, przygotowaniem dokumentacji Aleksander Gawel, a animacja była zadaniem wspólnym. Po zmianie przydziału pracy w trakcie, ostatecznie przygotowanie GUI, ramek oraz osi, a także praca nad update'ami i pomoc przy animacji zajęła Kamilowi około 10 godzin, przygotowanie rysunku i część pracy nad animacją zajęła Dominikowi około 10 godzin, a przygotowanie dokumentacji oraz pomoc przy animacji i drobne podpowiedzi dotyczące rysunku zajęła Aleksandrowi około 8 godzin.

9 Opracowanie i opis niezbędnych algorytmów

Po zdefiniowaniu funkcji, która pozwoli rysować krzywą, należało ustalić, jakie parametry wpływają na poszczególne zachowywanie się krzywej. W ten sposób zostały one tak dostosowane, żeby dane wyjściowe zachowywały się zgodnie z oczekiwaniami. Po utworzeniu GUI zawierającego wszystkie potrzebne parametry, należało napisać kod rysujący krzywą. Po ustaleniu, że krzywa, siatka oraz osie układu współrzędnych rysują się zgodnie z oczekiwaniami, do skonstruowania pozostała animacja.

W zadaniu wykorzystany jest algorytm operacji na macierzach, który odpowiada za przekształcenia 3D będące narzędziem pozwalającym obracać rysunek wokół każdej z osi układu współrzędnych. Potrzebne macierze znajdujące się w pliku `GUIMyFrame1.cpp`. Aby otrzymać macierz ostateczną, macierz translacji została przemnożona przez macierze rotacji oraz przez skalę.

10 Kodowanie

Plik `SquareBox.h` zawiera zmienne i tablice tworzące ramkę (sześciąt).

Plik `LissajousCurve.h` zawiera zmienne będące parametrami we wzorach 1,2,3, zmienne opisujące sposób rysowania krzywej w układzie współrzędnych kartezjańskich, a także możliwość obrotów poprzez "update'y".

Pliki `GUIMyFrame1.h` i `GUIMyFrame1.cpp` zawiera zmienne i funkcje odpowiadające za wszystkie opcje dostępne w GUI, a także definiuje i przeprowadza "update'y" za pomocą algorytmu operacji na macierzach.

Pliki `GUI.h` i `GUI.cpp` zawierają wszystkie zmienne tworzące wygląd GUI, czyli wszystkie przyciski, suwaki, teksty i pola do wpisania danych.

Plik `AxisLines.h` zawiera tablicę i zmienne odpowiadające za wyrysowanie osi układu współrzędnych.

11 Testowanie

Testowanie odbyło się poprzez uruchomienie programu i sprawdzenie, jak wprowadzenie poszczególnych zmian wpływa na dane wyjściowe. Potrzebne usprawnienia zostały dodane.

12 Wdrożenie, raport i wnioski

Aby uruchomić program, należy wszystkie pliki umieścić w odpowiednich folderach (pliki nagłówkowe, biblioteki) i skompilować przy pomocy np. Visual Studio 2022. Pliki są podzielone na realizujące GUI, rysujące krzywą, wykonujące animację, a także rysujące siatkę (grid) oraz osie układu współrzędnych. Kolejność wykonywania zadań była pomocna, by program mógł działać płynnie, to znaczy tworzenie jednego elementu po drugim ułatwiało sprawdzanie błędów i spójności. Wbrew temu, co przewidywano, to animacja wymagała największego nakładu pracy i wspólnych sił, by optymalnie dostosować czas pracy.