

Zad 12.

Zaimplementować program zawierający GUI w środowisku WinAPI, który wczytuje, przetwarza i wizualizuje sygnał z żyroskopu umieszczonego na robocie mobilnym. Założono, że robot porusza się ze stałą prędkością (wartość prędkości można zmieniać za pomocą odpowiednich przycisków GUI). Program ma wyznaczyć i wyświetlać (w GUI) animowaną bieżącą trajektorię (drogę przebytą przez robota) - należy wykorzystać timer.
Zad 12a: outputRobotForwardA01.log

Projekt numer 3

Autor projektu: Seweryn Majewski ACiR WETI 2sem. Indeks: 181675

Użyta technologia: Java

Dokumentacja:

1. Działanie programu:

Celem programu jest wizualizacja odczytów z żyroskopu umieszczonego na robocie. Program jest podzielony na 3 klasy w celu rozdzielenia funkcjonalności i utrzymania SOLID zasad programowania.

a. ReadFile.java

W tej klasie program wczytuje plik tekstowy z podanej ścieżki, a następnie czyści i przygotowuje dane do przekazania ich dalej. Funkcja **getData(int linesToSkip)**, jest wywoływana tylko raz podczas życia programu, i wymaga argumentu specyfikującego ile pierwszych linii danych pominąć.

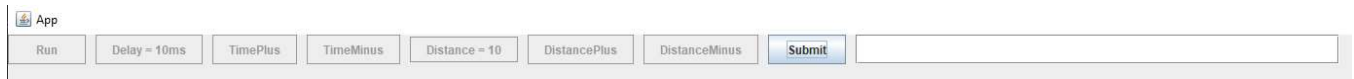
b. MyForm.java

Jest to klasa implementująca elementy GUI, oraz jednocześnie pełni funkcję klasy main. Poza implementacją elementów interfejsu graficznego, definiuje również ich zachowanie w poszczególnych scenariuszach użytkownika. Bowiem zaimplementowane są tak zwane "Listenery" czekające na input z przycisków, niektóre funkcje walidujące wejście użytkownika oraz jedna z ważniejszych funkcji programu: **runGraph()**, która nie przyjmuje żadnych argumentów i jest wywoływana przy wciśnięciu odpowiedniego przycisku.

c. GraphicsDemo.java

Jest to klasa implementująca wszelkie aspekty rysowania 2D w programie, w której konstruktor kontroluje tworzenie i otrzymywanie danych oraz odświeżanie ekranu. Poza konstruktorem zaimplementowałem funkcję pomocniczą **createPoints(x_r, y_r, dist)** przyjmującą 3 argumenty typu **Double**. Celem funkcji jest przekalkulowanie punktów na podstawie danych wydobytych z pliku tekstowego. Punkty kalkulowane są na podstawie podstawowych działań matematycznych (funkcje trygonometryczne i trochę algebry linowej). Ostatnią funkcją w tej klasie jest **paintComponent(Graphics g)**, jest to funkcja dziedziczona z klasy poziom wyżej w hierarchii dziedziczenia, która w abstrakcyjnym tłumaczeniu, rysuje nasz wykres na podstawie punktów zdefiniowanych w scope publicznym.

2. GUI



Po włączeniu programu jedynym włączonym przyciskiem jest przycisk “Submit”, który przekazuje do programu liczbę wpisaną obok. Liczba ta jest liczbą linii pliku które chcemy pominąć.



Po wpisaniu poprawnej wartości podświetlają się dodatkowe przyciski. W tym momencie możemy dostosować szybkość symulacji oraz przybliżenie wykresu. Warto dodać iż gdy klikniemy “Run”, nie będziemy w stanie tego już zmienić aż do zakończenia symulacji albo zakończenia programu.