

Python in the enterprise

Projekt 1: Flight recorder simulator

Wykonanie:
Paweł Sadowski
WFiIS IS rok 3

<https://github.com/paweus/pite/tree/master/projekt1>

1. Informacje o projekcie

Tematem projektu było stworzenie symulatora rejestrów parametrów lotu. Jako, że nie mamy możliwości bezpośredniego dostępu do prawdziwego samolotu należało w dowolny sposób zasymulować dane, a następnie odpowiednio je przeanalizować (zwizualizować).

W projekcie użyłem danych pozyskanych z programu FlightGear. Zostały one pozyskane przy użyciu maszyny Viggen (odrzutowiec wojskowy). Dane przechowywane są w formacie .CSV

Format danych:

Time,Latitude,Longitude,Alt Ft,Speed,Acceleration,Roll,Pitch,Heading,date,time

Przykład:

0.3,36.22683698,-115.0466482,1836.255238,0,0,0.2105903662,0.6848089322,41.15086807,2016/3/26,13:36:56
0.6,36.22683698,-115.0466482,1836.255238,0,0,0.2105903662,0.6848089322,41.15086807,2016/3/26,13:36:56
0.9,36.22683698,-115.0466482,1836.255238,0,0,0.2105903662,0.6848089322,41.15086807,2016/3/26,13:36:56

Najważniejsze elementy projektu:

- interfejs
- symulacja pozyskiwania danych
- wczytanie danych do wizualizacji oraz ich przetwarzanie
- generowanie statystyk
- generowanie wykresów

Interfejs został stworzony w oparciu o bibliotekę PyQt oraz przy wykorzystaniu programu Qt designer.

Symulacja pozyskiwania danych została zrealizowana poprzez „prowizoryczny” buffer. Jego zadaniem jest kolejne wczytywanie wierszy wprowadzonych do programu z pliku oraz ich analiza pod kątem poprawności. W przypadku otrzymania niepełnego zestawu danych buffer odczeka 1 sekundę (10 prób pobrania brakujących danych – a przynajmniej w teorii tak powinno to działać). Gdy dane nie zostaną uzupełnione wiersz zostaje porzucony.

Jednak aby w pełni przetestować możliwości buffera należałoby oprzeć oprogramowanie na wątkach, symulując tym samym prawdziwe zachowanie podzespołów, które mogłyby wysyłać dane asynchronicznie, i tym samym użyć możliwości buffera.

Po wyjściu z buffera dane są ponownie zapisywane do nowego pliku wyjściowego.

Wczytywanie danych do wizualizacji to prosta funkcja wczytująca ponownie dane (które zostały zapisane podczas symulacji) i wprowadza je do systemu. Niektóre dane należy przetworzyć np. poprzez zmianę systemu metrycznego.

Generowanie statystyk odbywa się poprzez analizę danych i doboru tych, które są potrzebne. Następnie statystyki wyświetlane są w jednej z zakładki w GUI

Generowanie wykresów opiera się na rysowaniu dwóch zestawów danych: czasu oraz drugiego zestawu. W programie zostało zaimplementowane 6 różnych wykresów. Dodatkową funkcjonalność stanowi możliwość doboru zakresu czasu, co pozwala na generowanie wykresów w dowolnie wybranym momencie lotu. Wszystkie wykresy zapisywane są do plików .png w folderze images.

2. Informacje o plikach, klasach oraz bibliotekach

Program podzielony został na 8 plików, które stanowią 8 klas. W ich skład wchodzi:

Buffer, FlightSimulator, Operations, Plotter, Reader, Statistics, UI, Validator

Program uruchamia się poprzez skompilowanie pliku **UI.py** komendą **python UI.py** lub poprzez uruchomienie pliku **Start.bat**

Do paczki dołączone zostały zestawy danych:

Wartosci.CSV – zawiera zbiór poprawnych wartości do użycia w programie

Wartosci(errors).CSV – zawiera wartości, wśród których występują wiersze o niepełnej zawartości. Podczas wczytywania buffer odrzuci takie wartości (co będzie wyświetlone w konsoli)

SimulationOutput.CSV – jest to plik nadpisywany po każdej nowej symulacji danych, znajdują się tu dane, które wczytujemy do wizualizacji

Biblioteki potrzebne do skompilowania programu (wraz z wersjami, które zostały użyte)

*matplotlib-1.5.1
numpy-1.10.4
PyQt4-4.11.4*

Dodatkowe opisy kodu znajdują się w komentarzach w kodzie programu

Zdjęcia przedstawiające działanie programu:



