PitE
Projekt: Flight recorder simulator

Olaf Schab IS rok III

1. Opis projektu

Celem wykonywanego projektu, było zaprojektowanie i wykonanie programu symulującego urządzenie rejestrujące i zapisujące parametry lotu – czarną skrzynkę.

Wykonując projekt mam zamiar skupić się na odtworzeniu działania tego urządzenia podczas lotu – dane na bieżąco są przesyłane do bufora, który następnie zapisuje je. Na końcu postaram się przygotować proste GUI, którego zadaniem będzie wizualizacja zebranych danych.

2. Założenia i rozwiązanie

Pisząc program mający prezentować pewne dane musimy mu je jakoś dostarczyć. Do tego celu posłużyło mi oprogramowanie FligtGear, dzięki któremu dane takie jak: czas lotu, położenie geograficzne, wysokość samolotu, oraz kąty wychylenia samolotu, zostały zapisane do pliku csv (z uwagi na problemy z FlightGear'em na moim komputerze, używane przeze mnie dane zostały wygenerowane przez Jakuba Serafina). Mając już surowe dane, zasymulowałem otrzymywanie ich w czasie lotu. Posłużyła do tego klasa *Buffer*, która przyjmowała jedną porcję danych. Jeśli wszystkie wartości zostały wpisane do bufora zostają one przesłane jako argument do metody klasy *Validator* gdzie przechodzą walidację. Kolejnym krokiem jest wykorzystanie klasy *Saver* do zapisu danych znajdujących się w buforze.

Gdy wszystkie dane zostały już zapisane (plik '*data.txt'),* program umożliwia ich odczytanie, a z pomocą metod klasy *Plots* zostają one zaprezentowane na wykresach.

Podczas wykonywania programu zauważyłem, że niektóre dane są przedstawione w niezbyt przystępnej formie. Czas nie był podany od 0.0s, a wysokość wyrażona została w stopach angielskich. Zmusiło mnie to do stworzenia osobnej klasy *Fix,* która zamieniała te wartości, oraz zaimplementowałem tam również metody niezbędne do przygotowania danych do reprezentacji na wykresach.

Na samym końcu zostało stworzone GUI z użyciem biblioteki pyqt4. Aby uruchomić program należy wywołać plik o nazwie *GUI.py*.

3. Wnioski

Po napisaniu programu można stwierdzić, że spełnia on swoją funkcję. Jednak po głębszej analizie łatwo zauważyć, że nie jest on w pełni skończony i niezawodny. Według mnie najwięcej zastrzeżeń budzą klasy *Buffer* oraz *Validate* – niestety nie zostały zaimplementowane mechanizmy, które określałyby działanie programu w przypadku błędnych lub niepełnych danych – w tych sytuacjach program zwyczajnie przestaje działać. Dodatkowo wydaje mi się, że obiekt klasy *Buffer* powinien posiadać pewień pewien okres czasu, dla którego dane mogą być wysłane z opóźnieniem. Dopracowania wymaga również GUI, lecz jego prostota spowodowana jest pierwszą stycznością z pyqt – mimo tego spełnia on swoją powinność. Jak widać program działa, ale posiada braki, które mogą spowodować błedy. Główną przyczyną braków w programie jest niewątpliwie zła organizacja czasem i błędne założenia.

Dokładniejsze opisy klas i metod znajdują się w plikach programu.