**User Manual of**

**Scenario file generator for BlueSky ATM Simulation**

1. Introduction

BlueSky is a fully open-source ATM simulator, which aims to provide a fully portable, freely downloadable package, easy to use, yet versatile. BlueSky aims to achieve a level of user- friendliness and fidelity that is comparable to commercial off-the-shelf tools for ATM simulation.

In order to show the air traffic in the simulation, the scenario file is created which contain all the flights and their routes which run it on the simulation. This little tool is developed to help in generating the scenario file which focuses on flights near an airport for departure and arrival purpose. The tool consists of a set of program with functions, a set of data file and the resulted file.

1. Input Data

The tool will require data in the form of 4 text files:

* flights\_data.txt which contains of information of flight code in first line, aircrafts types in second line with spacing and interval time between flights in the third line with seconds unit
* unpublished\_waypoints.txt which contains information about additional waypoints which is not published in the navigation database. Usually for the waypoints during approach and takeoff near the runway with numbering name. The format is waypoint name, position coordinate and point type such as the example is “EH609, 52.234444,4.595833 [FIX]”.
* arrival\_procedures.txt which contains all arrival procedures from enroute transition waypoint until runway. Each procedure has 3 lines of data. The first line is for airport and runway origin, airport and runway destination, first segment heading and initial speed. The second line consists of all waypoints from enroute to runway. Last waypoint is always runway with its airport code. The third line is for altitude at each waypoint as mentioned in the second line.
* departure\_procedures.txt which contains all departure procedures from runway to enroute transition waypoint. Each procedure has 3 lines of data. The first line is for airport and runway origin, airport and runway destination, final speed and final altitude. The second line consists of all waypoints from the point after runway until enroute transition waypoint. The third line is for altitude at each waypoint as mentioned in the second line.

1. Program Architecture

The program is coded in Python which consist of a main program and 3 functions:

* The main program will show the opening text in the screen, and then it calls the 3 functions starts from waypoint, arrival, and departure. It ends with showing the closing text to give information about the resulted scenario file.
* The waypoint function will introduce unpublished waypoints. It opens the unpublished\_waypoints.txt file then it writes each line of the data into the scenario file with the correct format.
* The arrival function will generate arrival flights. It starts with opening the flights\_data.txt file and then reads how many aircrafts need to be generated for each procedure. The function will open the arrival\_procedures.txt file and then split all pieces of information. Finally, it will write to the scenario file the information for creating a flight starting point, original and destination of the flight, and then all the waypoints along the arrival route with its altitude.
* The departure function will generate departure flights. It starts with opening the flights\_data.txt file and then reads how many aircrafts need to be generated for each procedure. The function will open the departure\_procedures.txt file and then split all pieces of information. Finally, it will write to the scenario file the information for creating a flight starting point at runway, original and destination of the flight, final speed and altitude, and then all the waypoints along the departure route with its altitude.

1. Resulted File

The resulted file name is neno.scn which created in the same folder of all the program and data. It consists of definition of unpublished waypoints used in this scenario, all arrival and departure flights according to the input files.

1. How to use it

* Prepare all necessary information in the input files
* Run main.py file in Python shell.
* Retrieve the resulted scenario file: neon.scn
* In case of an error, do the necessarily correction before re-run the program
* Open the neon.scn file in the BlueSky ATM Simulation to view it.

-o0o-