YSFlight Mission Utility

User Guide

Author: YSFHQ Member Schwarzwald

Mark Joseph Virgona

Contents

Installation

General Notes

The YSFlight Mission Utility was written in Java using code available in 1997, this includes using the AWT package and thus, many of the functions have been deprecated but should work on older Java versions.

Java is available online at <>, to run this program all you require is the Java Runtime Environment (JRE), to develop Java applications and to compile the source, you need to download the Java Development Kit.

Note that if the program is not installed in the correct directory, it will not open.

Installation on a Mac Environment

The program searches for YSFlight versions by looking in a folder called “ysflight” for application/s (Macintosh Applications have a .app suffix). The application/s can be named anything as long as they contain all the folders (e.g. aircraft, scenery, etc.) a new installation of YSFlight has. Multiple versions can be located in the root folder.

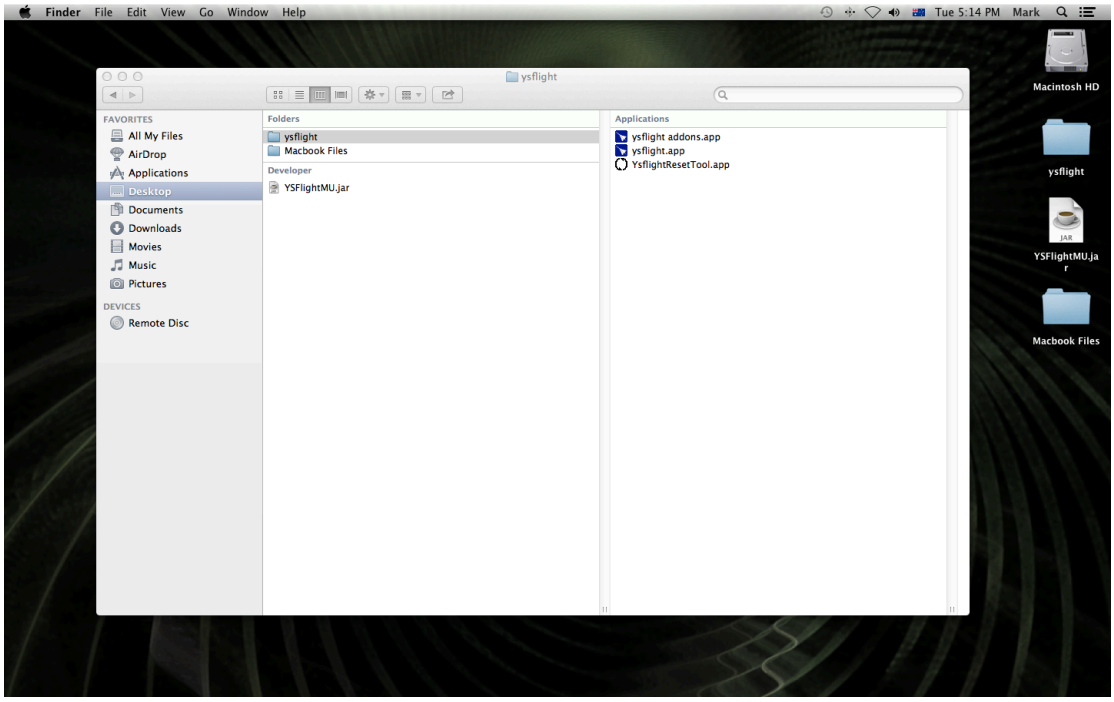
Installation on a Windows Environment

The program searches for YSFlight versions by looking in a folder called “ysflight” for sub-folders which contain the binary (Windows Applications have a .exe suffix) and the folders with the objects used by the program. Multiple versions can be located in the root folder.

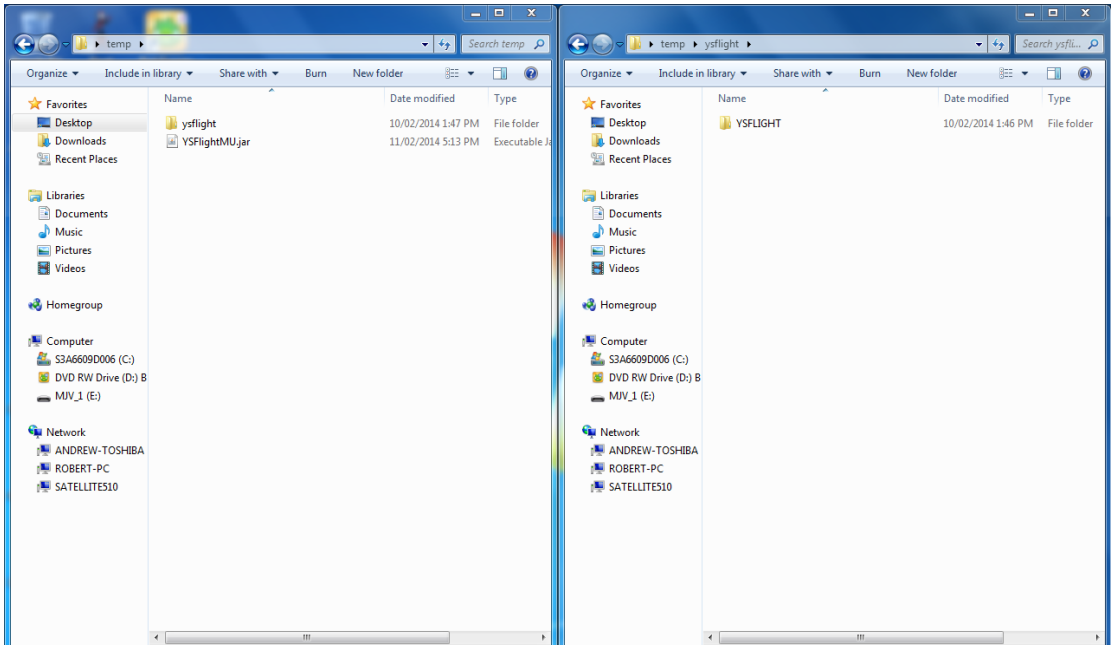
Installation on a Linux Environment

At the time of writing, the author does not own a copy of Linux that has been installed onto a computer, as both the Linux and Macintosh environments are both based off the UNIX kernel, the author can only assume that the program will work similarly to that of a Mac environment.

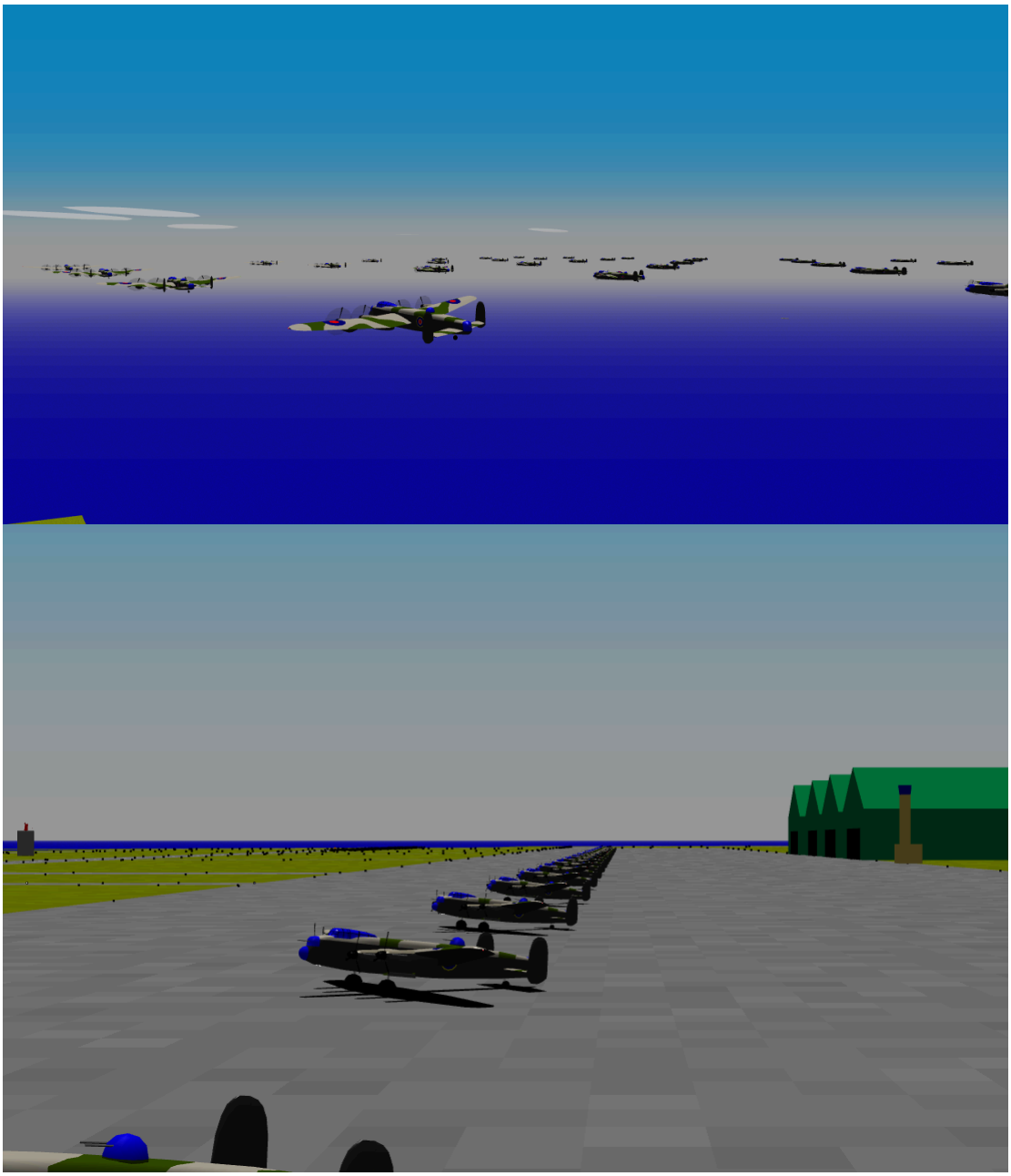
Mac Screenshot

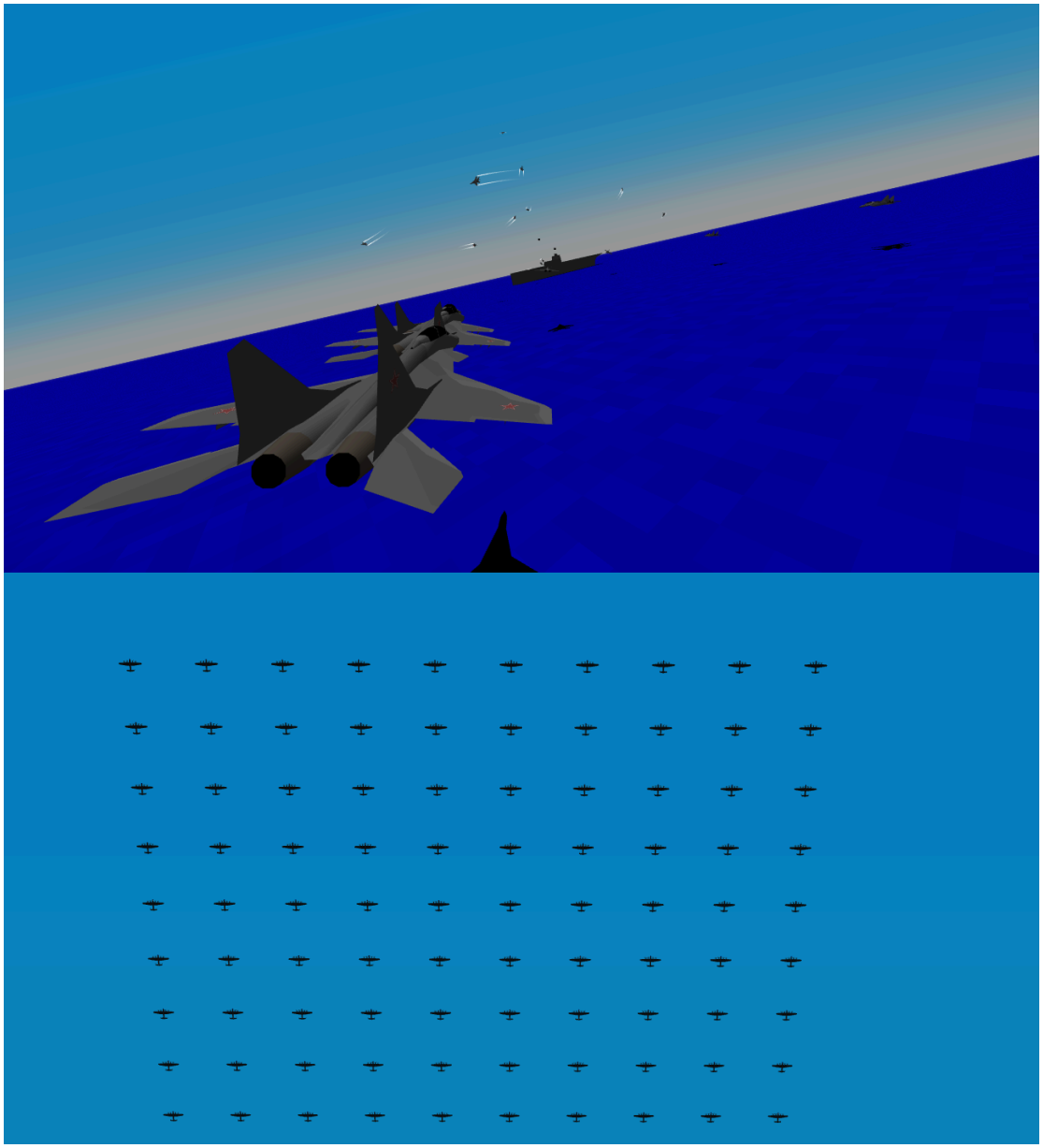


Windows Screenshot



Screenshots

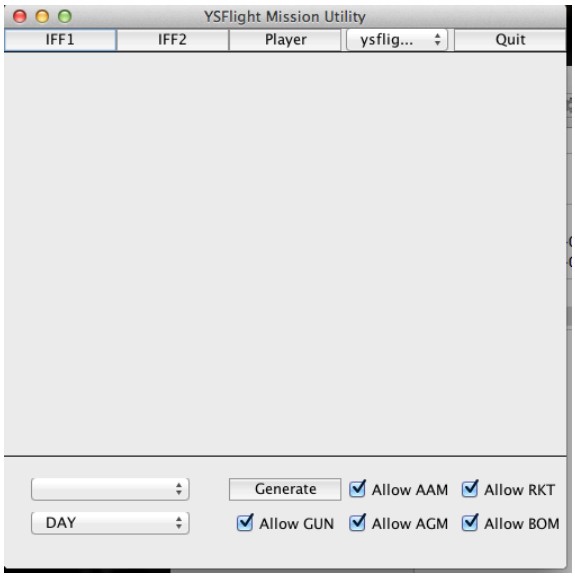




Use

To run YSFlightMU, double click on the YSFlightMU.jar, this file contains everything the program needs.

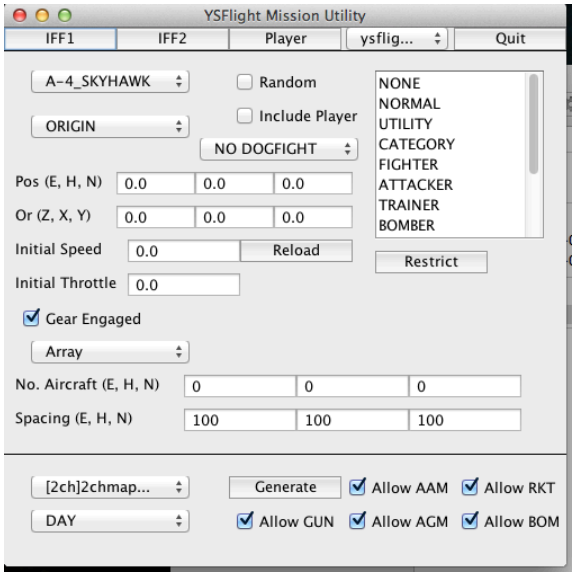
A screen will pop up that is initially blank; the program is loading all of the initial data from the YSFlight folders. Once all the data has been loaded, a series of Graphic User Interface (GUI) Elements will appear.



Going across the top of the screen, there are five elements, from left to right they are; three buttons labeled, “IFF1”, “IFF2” and “Player”, a choice box with a version of ysflight found displayed, and a final button labeled “Quit”.

The first three buttons control which

‘team’ you are editing. “IFF1” will edit all aircraft that identify with IFF1, “IFF2” will edit all aircraft that identify with IFF2 and “Player” will edit the player aircraft.



Clicking on the buttons will change the GUI elements between the buttons and the bottom elements which control general features like the map. Any data entered into the GUI elements for each respective team will be preserved when switching teams; at first the data is identical for both IFF1 and IFF2.

For the “IFF1” and “IFF2” screens, there are numerous dropdown boxes, these control what aircraft the team uses, their start position, their ability level and their originating formation.

The contents of the dropdown boxes should indicate what parameters they control.

Next to the aircraft dropdown box, there are two checkboxes labeled, “Random” and “Include Player”. The “Random” checkbox, instead of allocating the chosen aircraft to the team, randomly allocates all the aircraft present in the dropdown box.

The “Include Player” checkbox includes the player aircraft within the IFF1 Team and also disables the “Player” button.

The dropdown box initially labeled “NO DOGFIGHT”, controls the computers ability with the standard YSFlight levels, along with their G-limits, (e.g. G7 means that the computer will fly the aircraft up to but no beyond a g-loading of 7).

Near the bottom of the screen is a dropdown box with two-options, “Array” and “Random”, these control what initial formation each team’s aircraft will be in.

Finally, there is a dropdown box initially labeled “ORIGIN”; it contains all the available start positions concurrent with the chosen map, along with a position at the centre on the ground for custom positions.

An important feature is the bank of textboxes below the start position selection. This allows a user to select a start position, view its details and then edit them as desired, for example, a use might want to start on the runway but further along it, they would choose the runway position, and then edit it as required.

The coordinate axes used are global YS coordinates, in the order going left to right of East, Height and then North. The higher the value in the respective textbox, the more towards the indicated direction the point is. For example, a value of 200 in the E box (East) is more easterly than a value of -200. The same applies to the H (Height) and N (North) boxes.

Below the position textboxes, are the orientation. Most commonly only the first one will be used as that affects the heading. The rotations are labeled, Z, X and Y as these are the axes that the rotations are about. For example, the Z-axis rotation affects heading as the Z-axis runs vertically in relation to ground. The X-axis affects pitch and the Y-axis affects bank, (the X-axis runs East to West and the Y-axis runs North to South).

Below the orientation textboxes are the "Initial Speed, Initial Throttle Setting and the Control Gear GUI elements. The Initial Speed textbox loads data in as metres per second, to change this type ‘kt’ after any data and the program will use the value in knots. The Initial Throttle textbox uses a value from 0 (meaning no throttle) to 1 (meaning maximum throttle); the afterburner is not engaged at any setting. The checkbox labeled “Gear Engaged” will control whether the landing gear of the aircraft is up or down when starting, if it is ticked then the gear will be down, if it is unticked then the gear will be up.

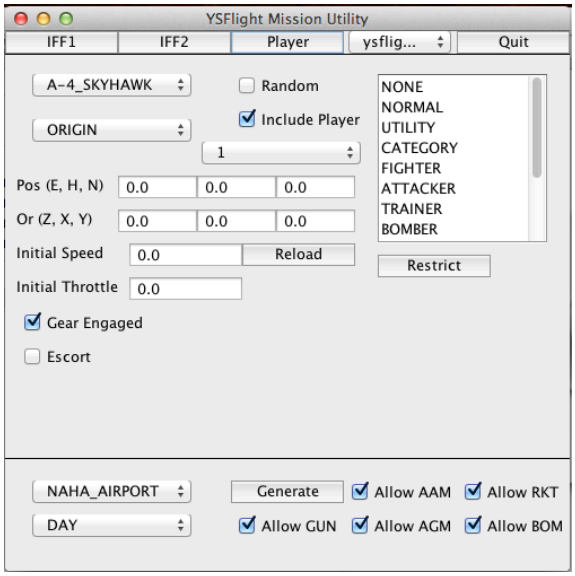
On the upper right of the screen there is a list with the various categories of aircraft, this allows aircraft to be shown in the aircraft choice box according to their category. To do this, select the categories of aircraft you don’t want in the choice box and click the “Restrict” button below (multiple selections are allowed). Thus, the user can select all but the “FIGHTER” category, restrict the choice box and then select the random aircraft check box, meaning that the program will allocate only fighter aircraft randomly amongst the team.

At the bottom of the team parameters, above the general parameters is the team formation section; the user has two options, either “Array” or “Random”

“Array” allows the user to specify that the aircraft will be generated in a box formation, with aircraft at regular intervals amongst all axes. The top row of textboxes specifies the amount of aircraft in each direction (E, H, N), to generate any aircraft; the values must be at least one in each box. Below the amount specification textboxes is the spacing (in metres), again in the (E, H, N) coordinate system, these textboxes control the spacing between each aircraft in each direction, negative values mean that the aircraft go in the opposite direction (West, Downwards and South).

“Random” allows the user to specify that the aircraft will be generated randomly within a bounding box centered on the chosen position. The top row of textboxes control the size of the box (in metres) whilst the bottom textbox specifies the amount of aircraft within the box, a value of zero in the bounding box control means a restriction of the axes that the aircraft will be generated in, for example, a zero in the H box means that the aircraft will be generated on a plane, a zero in two boxes means that the aircraft will be generated on a line.

The player control dialog is slightly different in that is doesn’t contain and formation options but allows the user to select which IFF (1-4) the player is. If the user deselects the “Include Player” checkbox, the file will be generated without any player aircraft and when loading into YSFlight, the user can only Replay the Record, (i.e. the computer aircraft verse themselves with little or no variation generated by the player).



One other difference is that below the gear control, there is a checkbox labeled, “Escort”, if selected; the program will generate two aircraft of the same type and IFF behind the player as wingmen.

The bottom GUI elements control the general attributes of the save file, the map chosen, whether it is day or night and which weapons are allowed (a bug in YS is that if AGMs are not allowed, they will be given anyway). Also, there is a button labeled “Generate” which will create the save file. The name of the save file will contain the amount and type of aircraft of each IFF.

Further Notes

The author, whilst he has endeavored to locate all bugs, is not sure whether any still exist and appreciates notification of such either through the YSFHQ forum or his email (m.j.virgona@gmail.com).

The author has also released The YSFlight Mission Utility under the GNU Public License, and has provided the Java source for the program, the author relinquishes all copyright to the program but wishes that any modifications or alternative downloads recognize the original authorship. As a note, he will be taking any suggestions for improving the program and will be regularly releasing updates and as such, also requests that people don't start releasing a million versions each with differing capabilities so as not to clog up the internet and to minimize user confusion. Any recommendations for improving The YSFlight Mission Utility should be directed either at the YSFHQ forum or the author's email (cited above).

The author thanks the users for their cooperation.

The author would also like to pay homage to the YSFHQ user known as Grigor for being the first to create a program that generates YSFlight save files, the author would like to express that during the creation of this program, no elements were taken from Grigor's program and that the author has not viewed Grigor's program, as such any similarities are purely coincidental and the author apologies in advance for any consternations this might cause. The author hopes that Mac and Linux users especially will benefit from the cross-compatible nature of Java and this program was designed in no such way as to supersede Grigor's own groundbreaking work.

Proposed additions (once the author has comfortable settled into uni)

More formation options (e.g. sphere, triangle etc.)

General IFF options (instead of buttons a dropdown list with each IFF)

More category selected control (instead of only a restrict)

Addon control (i.e. restrict aircraft based on what air-.list they came from)

Aircraft search-ability (i.e. enter "US_" and only get the US GAC aircraft from which you can randomize or category restrict)

Ground object placement choice (the user can choose whether to place the ground objects)

Known Malfeasance

Ground Object placement at some maps is sketchy at best, especially at Airstrike Challenge.