

A Hornet scenario contains descriptions of up to a hundred(!) objects, which can be planes (including yours), boats and ships, land vehicles (yes, the Lambo is one of them), artillery, buildings and other stuff (the flooding after the dam burst, for instance.) Every object has a list of up to 25 waypoints associated with it. Also, every object has a script that defines its behavior. There should be at least one FA18, number 0, which is your plane if you start the game in the normal way.

The editor is organized as a HyperCard stack with one card for every object in the scenario. You can work on only one scenario at the time but you can load and save scenario's to text files with the appropriate buttons. Loading a scenario will place the loaded objects after the already present objects. If necessary, you can clear a scenario with the clear button or you can use the usual HyperCard editing commands to delete and modify existing objects.

Normally you start by opening a Hornet data file. Presently this must be either the demo or the Kuwait data file, simply because I don't have the Korea missions yet. Opening the data file will make available a number of other buttons. You can select from a list of available missions via a pop-up menu. Depending on the data file you opened you will be able to select from only the Hawaii missions or from the complete list, including the network (Arizona) missions. You can load or save your present scenario from or to the data file. Of course for the demo it only makes sense to save to the first scenario, since that is the only one you can actually play. For the real thing it is probably advisable to stay within one theater when transferring from one scenario to another, or your ships will end up in the desert and your plane will start on the water surface. If you want to transfer from one data file to another you can use the 'Close Data' button to close the present one, which will make the 'Open data' button available for opening another file. The data file will be automatically closed when you quit the program.

The data file is opened non-exclusively! This means that it can be open in the editor AND in Hornet. You can make your edits, switch to the Hornet and immediately try them, which is a big help. Just make sure that you are not flying when you apply changes, or you are bound to crash. Abort the mission first!

The lower part of the editor has five fields, one pop-up menu and a button to create a new card (i.e. a new object.) The pop-up menu lets you select the object type from a rather long list. Next to it are two small fields. The first one contains a running number when more than one type of the same object is present in the same scenario, e.g. FA18 number 0,1 and 2. The second seems to be nothing but padding bytes and is included for completeness. You never know whether it does serve a purpose. Both fields are hexadecimal.

The interesting fields are right below that. Left is the list of waypoints, organized as three decimal numbers per line (longitude, altitude, latitude). For your own plane these are the familiar TACAN waypoints. For the other planes, these waypoints play an important role in navigation as well, as the scripts are largely concerned with moving from one waypoint to another. They are also used for targeting.

The field next to it is the actual script window. At first sight it may look a little disappointing as it is all hexadecimal code. It does not (yet) have simple English like commands like 'increase throttle to 80%' or 'fly to waypoint 3'. This is because my understanding of the scripting language is not yet very complete and leaving the script in it's 'bare code' form at least allows for flexibility. At the moment, you might say, it is very much still a hacker's tool. In the 'Tutorial' document you will find an introduction to the scripting basics. The good thing is that you can format the hexadecimal text in any way you wish as all non-hex characters are ignored, including spaces, comma's, tabs and returns.

For all the above fields holds that you don't have to fill them up with nulls. All fields will be automatically padded to the required number of bytes. When reading from a datafile the nulls are included in the field, though. Feel free to erase them if they bother you.

Finally there is the comment window. It is there only for your convenience, to use as you see fit. The comments are saved with the scenario files, but of course they are not saved to the Hornet data file.

The scenario files are simple text files that are pretty much self-explaining. They can be edited with any text editor or word processor that can save ASCII. There is a number of tags that determine the meaning of the data. They all consist of two characters, the first of which is a #. The part of the line after the second character is ignored and can be used as comment. At present the only recognized tags are #V for the HSE file version number (should be 1 for this version), #N to create a new object (and hence create a new card in the editor,) #W, for the waypoint list, #S for the script and #C for the comment. #T, the theater tag, is only used to indicated the theater and mission number into force when the file was created. It does not even have to be the theater that 'belongs' to the scenario and is only written to the file when a data file is open. For the rest it is completely ignored, as are all the other tags. I promise that if (if) I ever write a next version I will never use the '# ' (space) or the '#-' tag so you can use these to comment your scripts.

I hope that this information, together with the Tutorial, is enough to get you started. If you find out interesting stuff, please tell the world via Usenet and maybe someone will make a Hornet scripting FAQ out of it. For the rest, good weather and a lot of bogeys on your flypath.

Now back to those 30 Migs that I've almost cornered....

swartjes@bart.nl (H.M. Swartjes)