

## **Fun and Games with 3D Cockpits.**

I have now spent around a year building up skills in 3D cockpit making, am still not the best kid on the block for it! I have been continually frustrated by a complete lack of worthwhile documentation from the vendor and zero case studies, so much of my learning has been by reverse engineering existing models to get a feel of how other people have structured their work.

If you are not familiar with using 3D CAD desktop app's don't bother reading further. There are only three applications that offer direct export via custom plugins to X-Plane's OBJ8 file structure : Blender, AC3D & Sketchup. However in practice you can use a variety of other CAD packages, just export to Wavefront.obj and import to one of the above as your gateway to OBJ8. I have recently settled on AC3D as my preferred gateway for its relative ease of use and simpler texturing method. The animation is also relatively straightforward to visualise.

Unless you wish to make custom instruments you do not even need to be familiar with animation techniques, which are applied after import to one of the three app's identified above.

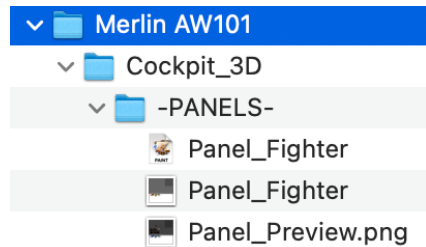
The first thing to do is to make your cockpit objects, suitably shaped to go where you want in the cockpit. These may be flat 2D surfaces or thicker 3D sections, in either case you will only need to texture those faces where you want instruments off the panel.png to be placed. If you have a 3D object you must first colour the whole thing and then separately texture instruments on relevant faces. Normally you will lay out a PNG with suitable shapes and colour panels to use when texturing, this will be automatically picked up from \Cockpit\_3D\panels by planemaker (PM) which will open the whole png onto which you lay out your cockpit elements each of which comprise an image, the file names of which in turn all refer back to XP dataref/command functionality. Since these names link back to internal XP functionality and specific datarefs they cannot be renamed, however the graphics and the labelling may be renamed so you may choose to relabel the arrestor hook switch as "PX Door" (passenger door,) for example. Most of these images are very small so the length of labels has to be very short. To make this work you have to assign the arrestor hook dataref to either animate with a translate , or rotate using defined hinge line on your door.

Before you create any graphic objects it is a good idea to create a new folder structure as in the first image below. this contains 3 files, the source graphic editor output, the PNG export and a preview PNG. The preview will be explained shortly. Note that I have named it Panel\_Fighter because I have ticked the cockpit type box in PM as fighter and it therefore expects to load from that name when adding instruments in PM's 3D cockpit tool.

Many of the default PNG buttons/switches/display are fairly basic compromise and really not representative of what they might look like in the real thing. You may create a custom 3D switch/lever etc, assign a dataref add some animation to it and assign it to be a manipulator which is a form of function that allows it to be picked by a mouse from anywhere in your cockpit, and even better the texture for it may be from any area on the cockpit PNG that will not be textured as a panel area. However if it is a display the relevant faces of the object must be textured from an off area instrument pre-positioned in PM.

The above is important; there are essentially three cockpit 3D types : solid/flat - do nothing, PANEL on which instruments from PM are overlaid and MANIPULATOR which are custom control/displays/gauges and driven from either a dataref or command.

Part of the problem of texturing a panel.png is that after creating the panel in PM the png itself does not contain any image detail for the instruments themselves. There is however (I think, hard to find) a keyboard command sequence that will create a separate preview png from which you texture your instrument areas. This command sequence for Mac is 'command-option-shift-space' and for Win' users control-alt-shift-space.



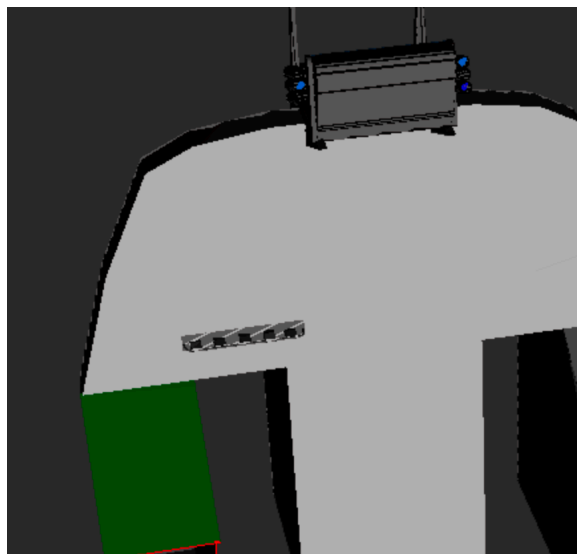
Whilst this brief will not be exhaustive I will run through a few case studies. These are drawn from work on the Sea Harrier and Merlin AW101 helicopter.

- Two flat panel areas.
- An In Flight Refuelling probe button.
- A cockpit opening handle.
- A light switch.
- A weapons arm switch and external tank jettison switch
- A MFD display with touch buttons.

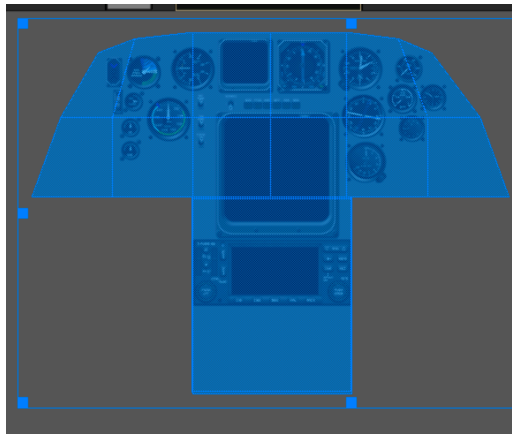
The images are from my preferred toolkit, ViaCAD or AC3D for objects and AC3D for animation and texturing.

## Two Flat Panels

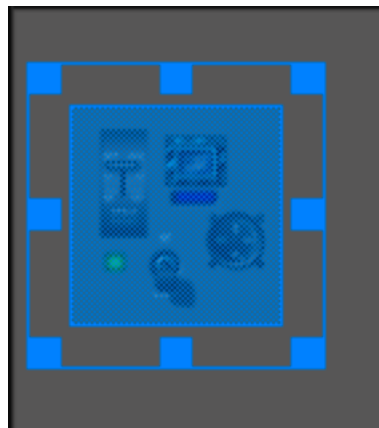
Here is my graphic object for the Sea Harrier cockpit. The white area (main panel) will be textured and also the the green one as an ancillary. Faces of both of these must be assigned with PANEL attribute. It is important that these are all perfectly flat with no gaps between their constituent triangle or quads, otherwise panel output will appear broken and/or distorted.



Having previously sited instruments and created a preview file I will texture the main panel and lower console with the texturing aligned as follows :



I will now texture the port knee panel from another area of the preview texture :



The almost finished object now looks like this, because these areas have panel attribute instruments can be selected and operated with a mouse.



Cockpits must be imported into PM as follows, note the naming format must be .ACF name+\_cockpit.obj. This goes at the top of the list.

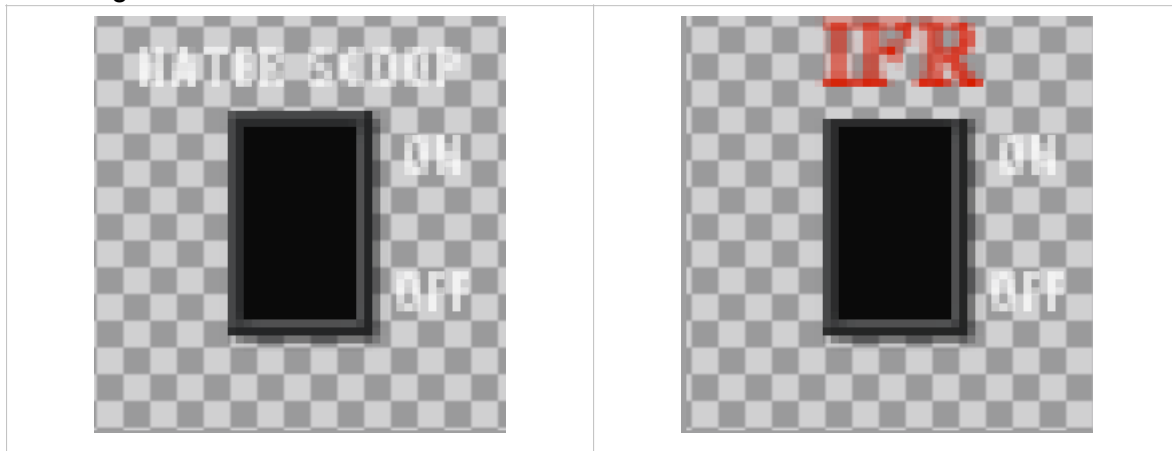
The screenshot shows the PM software interface for configuring a cockpit. It includes a 'clear' button, a 'Kill Dataref' field, and a 'SeaHarrier FRS1 cockpit.obj' dropdown menu. The interface also features a 'lighting' dropdown set to 'Inside', a 'shadow mode' dropdown set to 'None', and a 'No Prefill' dropdown. There are checkboxes for 'hi-res', 'wreckage', 'use LOD', and 'dim LIT'. At the bottom, there are radio buttons for 'Int Cockpit' and 'Ext Cockpit'.

When finished don't forget to go into PM and change the viewpoint to the position a pilots head is likely to be, I also set XP with a wider field of view.

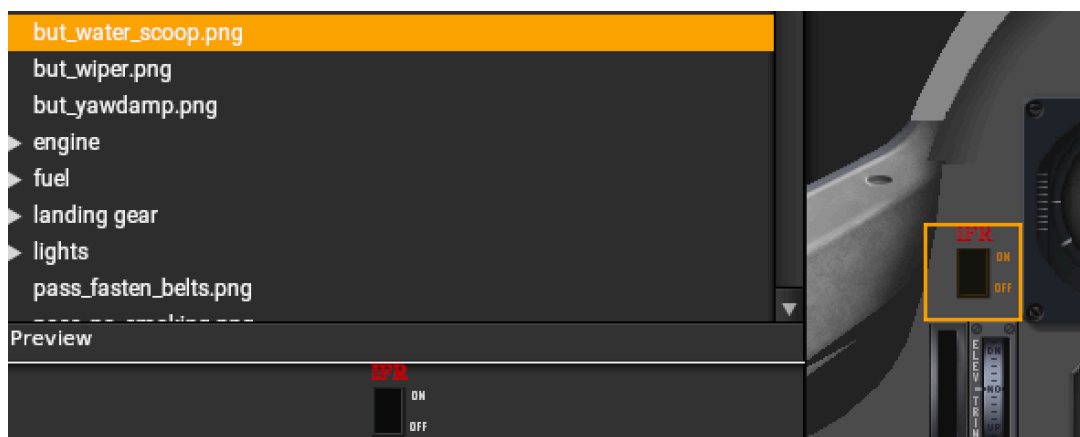
The screenshot shows the PM software interface for configuring a cockpit. It includes a 'cockpit' dropdown set to 'Fighter'. Below it, there are three sets of input fields for 'long arm pilot's viewpoint', 'lat arm pilot's viewpoint', and 'vert arm pilot's viewpoint', each with a '(ft)' label. To the right, there is a 'FIELD OF VIEW' section with a 'Lateral field of view' slider set to '100.08 degrees' and a checkbox for 'Use non-proportional vertical field of view'.

## An In-Flight Refuelling probe button.

This also applies to 2D cockpits. There is no default button to animate a retractable re-fuelling probe. So I adapted the `but_water_scoop.png` which originally looked like this on the left to what is on the right.

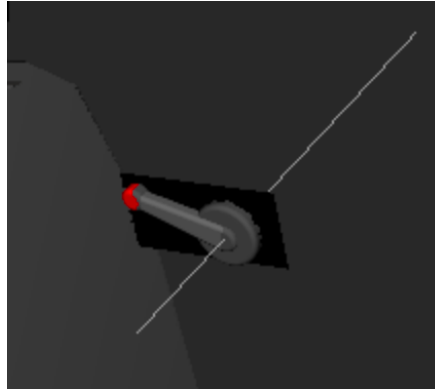


Below it is being applied in PM. A custom re-fuelling probe object was created and animated when `dataref : sim/cockpit2/switches/water_scoop_deploy` is activated. Now this isn't the best solution since it is a rapid open shut situation, I could have used the arrestor hook `dataref` which has a more elegant open/close but since this (Super Etendard) was a carrier based plane it was not free to be re-used. This is a good example of how if something you want doesn't exist in XP you can use an alternative operation that is not used in your model. It also illustrates how button names relate to `datarefs` and why you cannot substitute button names, unless you are prepared to go the extra miles and create a custom `dataref` (not here though). Never forget that texture panels MUST be in powers of two e.g.: 64x64, 256x 256 up to 4096x4096. It is also not uncommon to see weapon textures of something like 256x1024 (e.g. for a rocket or gun).



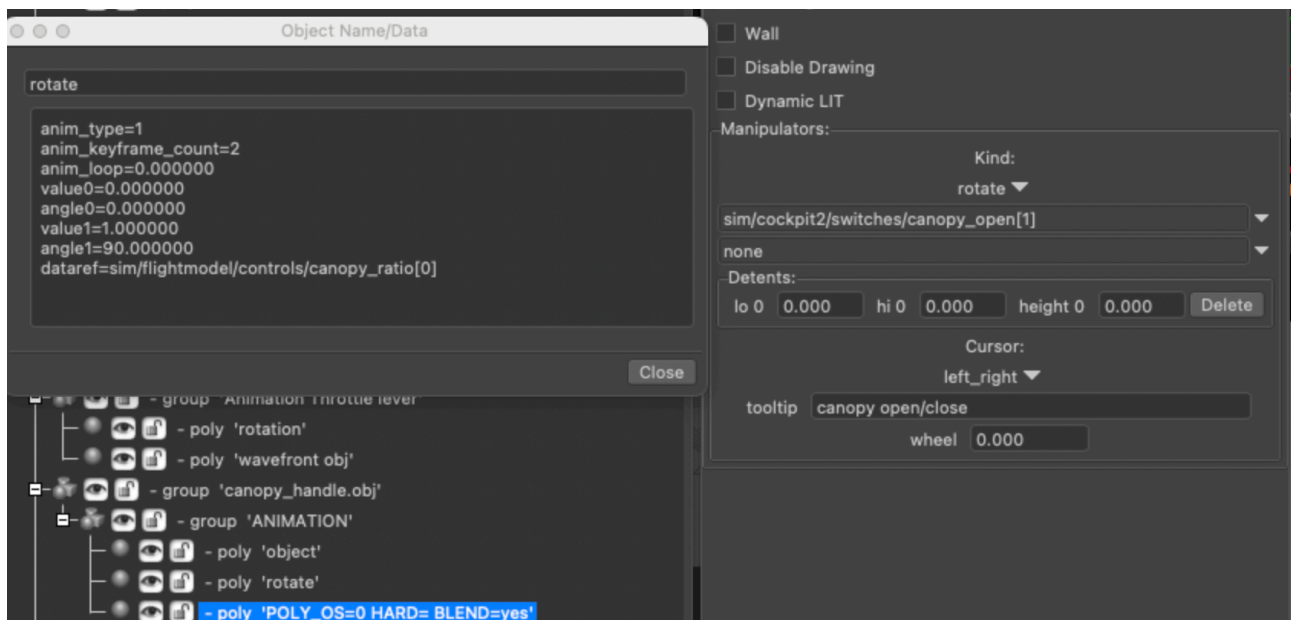
## A cockpit opening handle.

Rather surprisingly XP lacks a control button or lever to open/close the cockpit, nothing for doors either. So for the Sea Harrier I made my own, which was very early in my learning curve. The dataref for this is `sim/cockpit2/switches/canopy_open`. Here is the image of the handle and a backplane as fitted inside the cockpit object. The white line is the rotation axis.



Now here is the important part (in AC3D) :

Top left is the box to configure the animation, which must be manually edited to configure rotation start/stop angles and what dataref drives the operation. Bottom left is the sub components configured as an animation group, one unanimated object for the backplane, followed by the rotate command which operates on anything below it. To the right shows it as Manipulator configured as type rotate with the same dataref to pass to XP when the mouse moves it. Detents might be used for something like a flap setting lever. Cursor defines what appears when it is touched , in this case a left-right arrow, I could have chosen a hand and finally there is a tooltip which is displayed when hovering the mouse over a control, tool-tips must be enabled in XP's top menu.



How hard is that? Well here it looks simple but without any reference studies it took a couple of days to get it right, oh and only 5 mins to make the lever. Duplicate it , repaint it, change the dataref entries to : `sim/cockpit2/switches/door_open`, you should be able to see where I am going. To the port of cockpit there is similar setups for throttle and vector levers. If you don't have AC3D here is what it looks in OBJ8. If you know what vertex line numbers delimit your object you could, in theory, cut and paste the below and all you need to do is edit the TRIS statement, good luck

with that! No, I jest, don't even try! It is a real pity OBJ8 doesn't have a labelling system for vertex sets of each sub object, logic and probably the code is quite simple.

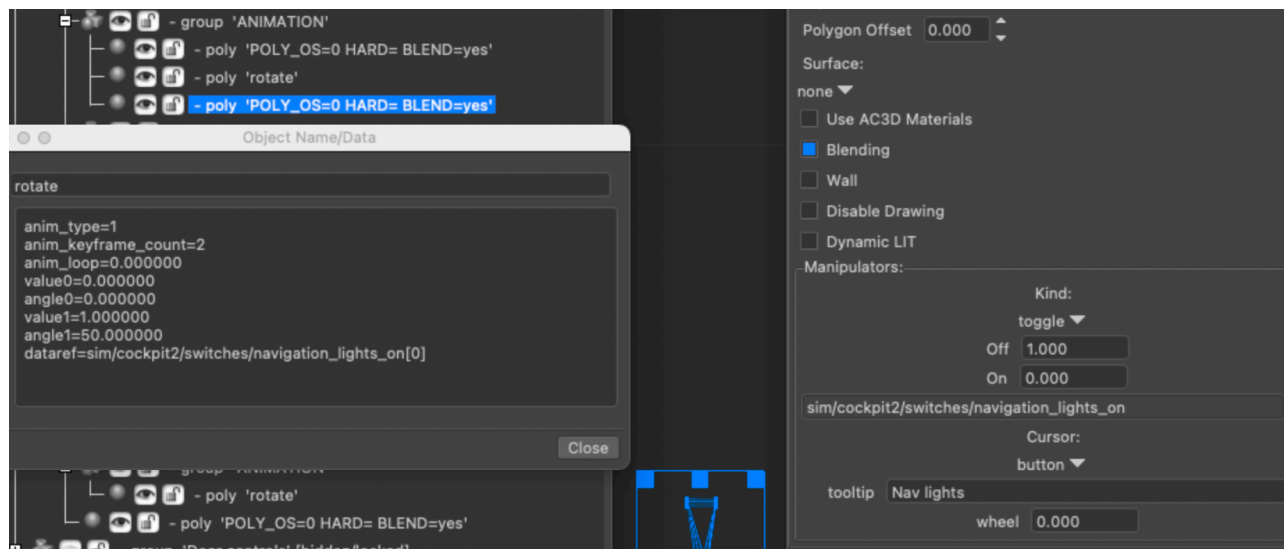
The ANIM\_trans are start end points for the rotation axis (the white line in the earlier image).

```
ANIM_begin
ATTR_manip_none
TRIS 48264 6
ANIM_trans 0.411346 0.786348 0.133754 0.411346 0.786348 0.133754 0.000000 0.000000 none
ANIM_rotate 0.970296 0.241921 0.000000 0.000000 90.000000 0.000000 1.000000 sim/
flightmodel/controls/canopy_ratio[0]
ANIM_trans -0.411346 -0.786348 -0.133754 -0.411346 -0.786348 -0.133754 0.000000 0.000000
none
ATTR_manip_drag_rotate left_right 0.411346 0.786348 0.133754 0.970296 0.241921 0.000000
0.000000 90.000000 0.000000 0.000000 1.000000 0.000000 0.000000 sim/cockpit2/
switches/canopy_open[1] none canopy open/close
ATTR_axis_detent_range 0.000000 0.000000 0.000000
TRIS 48270 7224
ANIM_end
ATTR_manip_none
```

## A light switch.

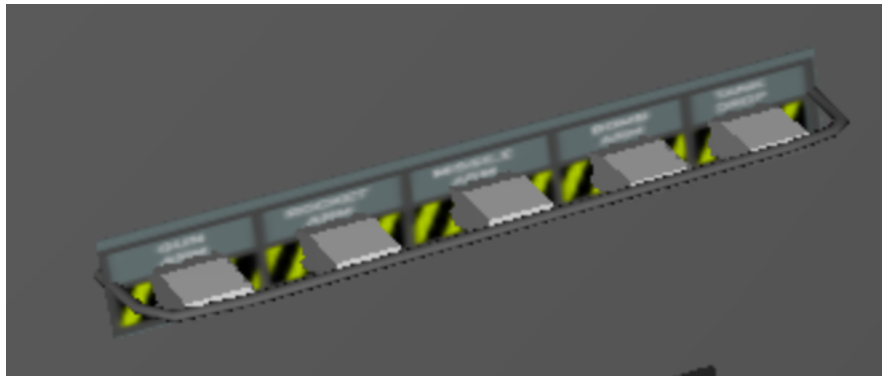
To be fair the default light switches in XP are amongst the better ones. But I wanted a generic method to create a variety for inclusion in future projects.

So top left is the object hierarchy, the top object is the mounting point, in this case a finger tight ring nut, or maybe a hex nut. Middle is the animation/rotation statement followed by a ball and lever object, which is the only part that moves, the graphic for which is lower middle. To the right the Manipulator attribute panel shows it configured as a toggle switch (on/off) the cursor type is a button, nothing much else here but reading the dataref you can easily see its function and hence how easy it would be to clone for other switches. Make careful note of the off/on values, they sometimes alter (flip/flop) with different datarefs.

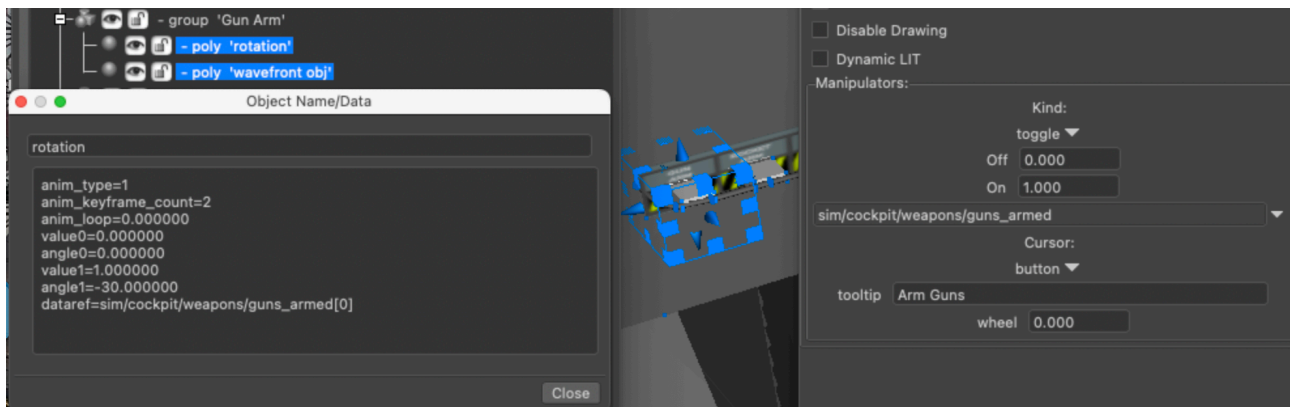


## A weapons arm switch and external tank jettison switch

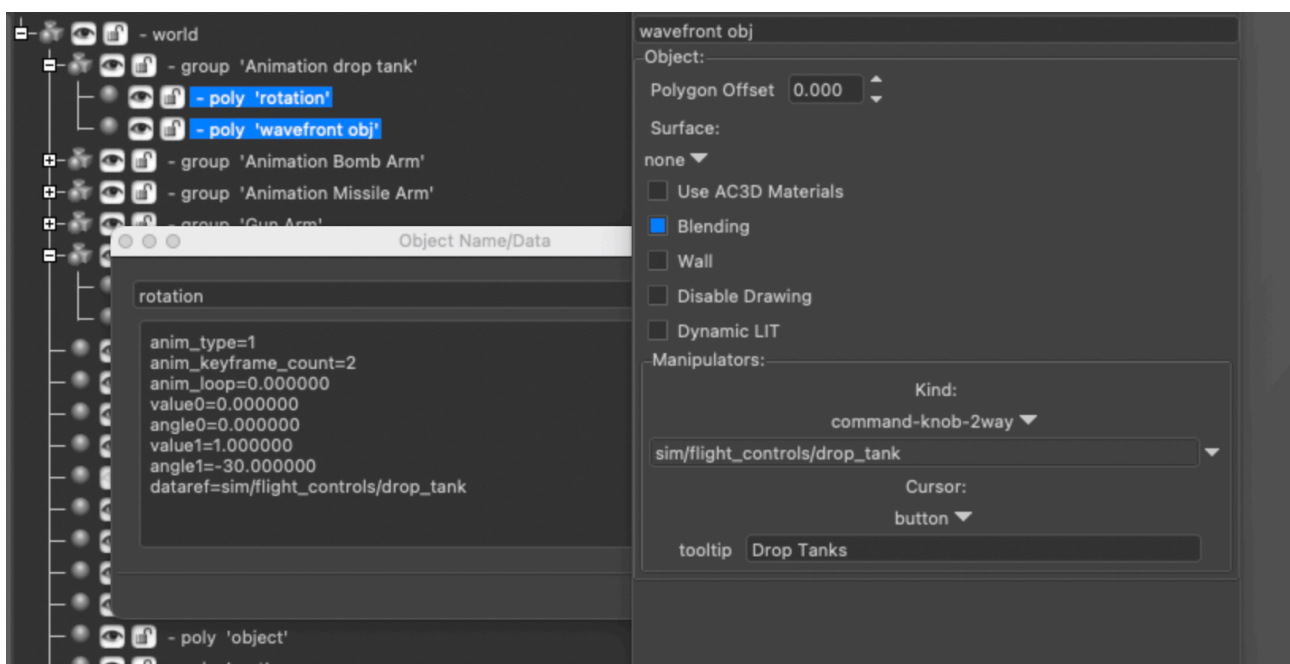
This actually took me around two days :sorting out datarefs. testing, retesting and then numerous insurmountable waits between reloading XP. Finally the tank jettison was a challenge because I found it is not dataref driven it is command driven. It all actually looks quite good but a neat trick I wanted to do was to illuminate a red light for each switch when it was activated, whilst custom lights are available I can find no way to control them n/off via dataref states. This is a finished object and is not too dissimilar to that in earlier Harriers, I know G-VTOL a T52 very well. There is a non panel back box onto which I painted a section of the cockpit texture which I overwrote with the switch lettering. There is a safety bar at front to prevent accident weapon arming. The real one is actually 5 switches which arm numbered pylons (two on each wing and one below the fuse). To the left of each switch is a vertical rotary dial to select the type of store on each wing, tank, rocket bomb etc. The position of which for example in the case of a bomb feeds into the ballistics computer weapon the weight and type so trajectory and optional automatic release can be calculated all taking into account airspeed, altitude, air pressure etc. It is said to be accurate to within 100yards radius with probably a blast area > than 20 yards radius. Now if we sent 20 Lancasters each armed with 20 bombs that is 400 bombs and 180 crew. In the early part of WW2 if anything hit within ¼ mile target they might be lucky so the probably of getting ten bombs within 100yds was probably around 5%. With modern weapons, providing the aeroplane gets through, the probability of doing serious damage to a target increases to over 50%, that is with just one pilot.



Here is the gun switch pretty much the same as the light switch example, obviously different datarefs and here you see different on/off states.



The drop tank jettison is a bit different, it uses a command (not a dataref) and type operates as a command-knob-2way. The rotation doesn't work because it is not a dataref so its just a mouse click, twice to jettison both tanks.



## A “simple” MFD display with touch buttons.

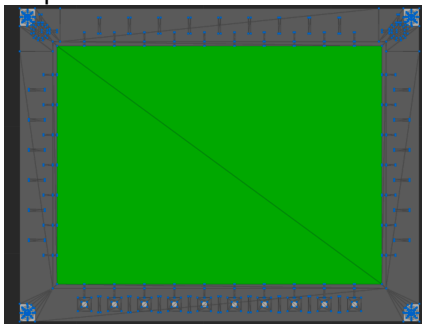
Moving up into the millenium almost everything now has multi-function displays. Conceptually its a box with a screen panel and a bunch of buttons.

Procedures :

1. Make your box and texture it a single colour. Give the faces that form the display panel attribute.
2. Make your buttons and optionally texture them individually with unique text. Even on the real ones there are likely to be numerous unused buttons so just use a colour texture.
3. Lay the instrument panels out in planemaker, update your preview png, if required.
4. Texture the display area (surfaces only) to the panel instrument.
5. Assign button functionality, these will be touch buttons so no animation is required.

Note that the preview colours do not mirror those of the panel.png. Ensure the final obj file is picking up the correct png.

Step 1.



Step 3. Texturing your button from preview

Step 3a. After tecturing



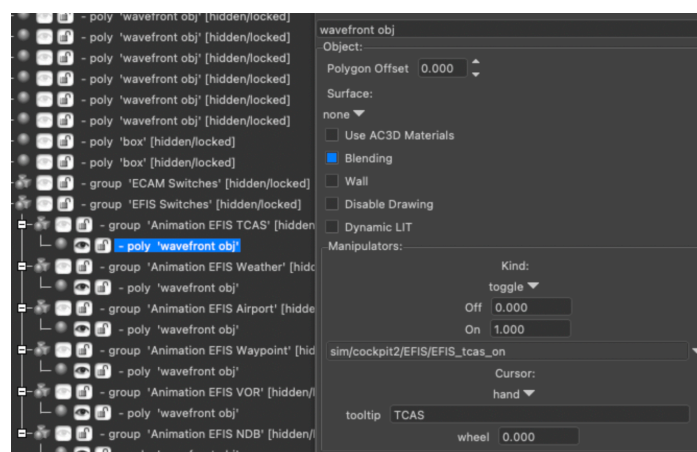
Step 4, texturing the display, note the outline of the original XP instrument.



4a. After texturing



The operation of the buttons is then rather simple.



These are some useful links:

Sadly the X-plane developer page does not appear to offer a list of commands, however commands.txt may be found in \resources\plugins. Default instrument images can be found in \resources\bitmaps\cockpit adapted one should be placed under in the cockpit/cockpit\_3D folder in your model.

<https://developer.x-plane.com/article/creating-a-panel-in-plane-maker/>

<https://developer.x-plane.com/article/how-to-create-a-preview-texture-for-the-panel-in-a-3-d-cockpit/>

<https://developer.x-plane.com/article/manipulators/>

<https://developer.x-plane.com/2009/04/datarefs-vs-commands-i-whats-the-difference/>

<https://developer.x-plane.com/datarefs/>

<https://developer.x-plane.com/article/designing-manipulators-for-virtual-reality/>

<https://developer.x-plane.com/article/custom-lights/>

This was a small sample of custom cockpit work. There are far more capability and manipulator types to work with but do be prepared to invest time, be patient and the results will come.

I at least hope this brief work will be shortcut which some may have been weeks of research and trial by error, which is how I got to this point.

[www.rayhillwrites.com](http://www.rayhillwrites.com)

<https://www.facebook.com/Basic-Theories-of-Aeronautics-478336918957680>