



## Introduction

I created the CRJ Glass Cockpit software, to help me with my main goal of making a home simulator cockpit to run with Microsoft's Flight Simulator. Hopefully as time permits, I will be able to start on the actual construction of it. My plan is to use this software to run it on up to two Linux computers, along with the main (faster) computer running FSX (on Windows of course).

The main reason that I decided to make my own software, was so I could customize it as I needed it. Also, I wanted to use Python as the programming language. I chose the Canadian Regional Jet (CRJ-100), for the main reason that my younger brother is an actual CRJ first officer. Therefore, I can obtain whatever manuals, specs, etc. that I need to hopefully make this as accurate as the time allows.

This software uses the SimConnect API on FSX to send and receive data from FSX. RJGlass uses OpenGL for all of the graphics. Since RJGlass is written in Python it can be run on both Linux and Windows. The main advantage of RJGlass is that it runs on other computers that don't have to be as powerful as a computer capable of running the latest version of Flight Simulator. It also gives you a good way to build a cockpit, with multiple monitors, for the PFD/ND, EICAS, and even FMS.

The project is open sourced and I am open to any help that people would want to give. I also would love to hear any feedback about the project, including suggestions and ideas for improvement.



## Installation

Note: RJGlass will only connect with FSX, or ESP. It will not work with FS9 (FS2004).

There are two download options, an Windows exe and a python source file. If you are using Windows you have two options, you can run RJGlass.exe, in the Windows exe download without the need of any other software, or you can go the with the python source. The python source is the only option if you are using Linux, or a Mac. In order to use the python source, you need to following requirements:

- Python ( $\geq 2.4$  and  $< 3.0$ )
- PyOpenGL (<http://pyopengl.sourceforge.net/>)
- PyGame ( $\geq 1.8$ ) (<http://www.pygame.org/>)
- Numpy (<http://numpy.scipy.org/>) only needed if using (Python  $< 2.5$ )

Then to run RJGlass from source, you can double-click RJGlass.py or from a terminal type:

```
python RJGlass.py
```

## Connecting RJGlass with FSX

### *Setting up FSX*

In order to connect to FSX, you have to let know FSX what port to look for connections on. To achieve this you need to have a SimConnect.xml file located in a specific directory on the machine running FSX. The SimConnect.xml needs to be located in:

- For Vista: C:\Users\AppData\Roaming\Microsoft\FSX (Note: I don't have Vista, to test this.)
- For XP: C:\Documents and Settings\Application Data\Microsoft\FSX



A sample SimConnect.xml file is located in each download. If there is already a SimConnect.xml file in this folder, then some other software that uses SimConnect, might have already created that file. If there isn't one in there, then copy the xml file that came with RJGlass in that location.

For RJGlass to connect, there has to be a SimConnect.xml file in that folder and it needs to have a global connection setup and a port specified. The port specified needs to agree with the port specified in config.py in the RJGlass folder.

**Note: Make sure there is no firewalls setup, that will prohibit RJGlass connecting to the FSX computer on the port selected.**

Here is a link to explain more about SimConnect and the SimConnect.xml file on FSX,

[http://www.fsdeveloper.com/wiki/index.php?title=Remote\\_connection](http://www.fsdeveloper.com/wiki/index.php?title=Remote_connection)

### *Setting up RJGlass*

All of the RJGlass configuration settings, are located in the config.py file. Note of caution, config.py is imported as python code, so one needs to be careful the syntax stays compatibility with python after each edit.

In the config.py file you need to specify the IP address, and port of the FSX machine. Set addr equal to the IP address surrounded by single quotes. (i.e. `addr = '192.168.1.100'` ) Next, make sure the port matches the port specifies in the SimConnect.xml file. If you use the included SimConnect.xml then the config.py set at port 1500 will already match the xml file. After the port and addr are set you are ready to test the connection. The last thing that needs to be done is to set the mode correctly depending on your version of FSX. The options are FSXSP0, FSXSP1, or FSXSP2, or ESP. There is also a TEST option, that will run RJGlass but it will not try to connect to FSX. After the mode is set, then make sure FSX is running (not at the main menu screen), and start RJGlass, and the connection



should be made automatically.

## Configuration File

The configuration file is the config.py file and holds many different options and is ever expanding. This file must stay in python syntax. Any line beginning with a # sign is a comment. Below is a summary of each of the options. Anything below italicized needs to be copied verbatim.

- *window\_x, window\_y* is the resolution of the screen. I wouldn't advise changing this.
- *full\_screen* set to either *True* or *False*.
- *mode* specifies the version of FSX that you are connecting to, or puts RJGlass into test mode. Possible options are *FSXSP0, FSXSP1, FSXSP2, ESP* or *TEST*.
- *addr* specifies the IP Address of the FSX computer, needs to be (i.e. *addr = '192.168.1.100'*)
- *port* specifies the port to connect to on FSX. (i.e. *port = 1500*)
- *server\_port* in future development. Will be used to communicate between multiple RJGlass apps.
- *timeout* sets the timeout in seconds for the RJGlass client when connecting to FSX. If a timeout is reached RJGlass will continue to retry.
- *data\_directory* needs to be in quotes, specifies where the navaid data is stored. The navaid data is in X-plane 8.64 format. (<http://data.x-plane.com/>)
- *sound\_directory* needs to be in quotes, specifies directory where all the sound wav files are stored.
- *splash* set to either *True* or *False*. If *True* the splash screen will show up.
- *splash\_filename* needs to be in quotes, the filename of the image file for the splash screen.
- *splash\_delay* delay in seconds for the splash screen, in addition to the time it takes to load all of



the navaid data.

- *use\_metric\_units* If set to true then metric units will be used for certain units, fuel flow etc.
- *max\_long, min\_long, max\_lat, min\_lat* limits of what navaids will be loaded, by specifying a latitude and longitude square. Used so not all navaids are loaded into memory. North and East are positive, South and West are negative.
- *max\_VOR\_range, max\_NDB\_range, max\_APT\_range, max\_FIX\_range* sets the maximum limit in nautical miles of the moving map (ND) to display the navaid.
- *min\_RWY\_length* sets the minimum runway length that an airport must have to be included in the airport database.
- *VNE\_flaps* used to set the barberpole to the never exceed speed for the airplane at each of the flap settings.
- *VS\_flaps* used to set the barberpole stall speed of the airplane for each of the flap settings.
- *Gear\_speed\_limit* sets the barberpole to this max gear speed in knots when gear is down.
- *FD\_Type* set the type of Flight Director, either Vshape for inverted V, or LINES for two perpendicular lines. (Note: Vshape and LINES are case sensitive.)
- *RA\_scale* if set to False will disable the radar altimeter scale on the right of the PFD.
- *brake\_cooling\_CONST, brake\_heating\_CONST, brake\_sensor\_CONST* used in formula's that calculate the temperature of the brakes.
- *debug\_AP, logfile* used for debugging purposes during development.



## Keyboard Assignments

### General

- *Ctrl-q* or *Esc* Exit RJGlass
- *[* Cycle Left half of screen.
- *]* Cycle Right half of screen.

### PFD

- *1* Cycle Bearing 1 Needle (VOR1, ADF1, None)
- *2* Cycle Bearing 2 Needle (VOR2, ADF2, None)
- *Tab* Cycle Active Nav on HSI. (VOR1, VOR2, None)
- *Alt-z* Cycle Vspeed Input (V1, V2, VR)
- *z* Increase selected Vspeed.
- *Shift-z* Decrease selected Vspeed.
- *Ctrl-z* Cycle visible on/off of selected Vspeed.
- *y* Increases VT speed.
- *Shift-y* Decreases VT speed.
- *Ctrl-y* Cycle visible on/off of VT speed.
- *d* Increases Decision Height (DH)
- *Shift-d* Decreases Decision Height (DH)
- *Ctrl-d* Cycle visible on/off of DH
- *m* Increases Minimum Decision Altitude (MDA)
- *Shift-m* Decreases Minimum Decision Altitude (MDA)
- *Ctrl-m* Cycle visible on/off of MDA



- *h* Increases Heading Bug.
- *Shift-h* Decreases Heading Bug.
- *b* Reset Kollsman to standard pressure (29.92 inHg/ 1013HPa)
- *Alt-b* Cycle between metric (HPa) and US (inHg) units.
- *Shift-b* Increase Kollsman setting.
- *Ctrl-b* Decrease Kollsman setting.

#### ND

- *PgUp* Increase Range of ND.
- *PgDn* Decrease Range of ND.
- *n* Cycle on/off NDB's on ND.
- *v* Cycle on/off VOR (Hi-Range) on ND.
- *Shift-v* Cycle on/off VOR (Lo-Range) on ND.
- *a* Cycle on/off airports on ND.
- *f* Cycle on/off fixes (intersections) on ND.

#### AP (Autopilot) Note: doesn't really work at this time.

- *F1* Cycle on/off the Autopilot
- *F2* Cycle on/off the Flight Director.
- *F3* Cycle on/off Heading (HDG) mode.
- *F10* Cycle on/off Altitude mode.
- *Shift-F10* Increase Altitude Hold.
- *Ctrl-F10* Decrease Altitude Hold.
- *F11* Cycle on/off Vertical speed (VS) mode.



- *F12*            Increase Vertical Speed / Altitude.
- *Shift-F12*    Decrease Vertical Speed / Altitude.