# Title Page

**The Research Civil Aircraft Model (RCAM) Simulator**

## Publication Number

AFSL-2015-25

See \\AFSL\TechnicalDataPackage\AFSLPublicationNumbers.docx for list of publication numbers.

## Date of Issue

March 14, 2017

# Record of Manual Revisions

Table 1: Record of manual revisions

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Revision** | **Date** | **Pages Affected** | **Revisions** | **Author** | **Check** | **Approved** |
| 1 | 02/06/15 | All | Created document. | Christopher Lum | Noel Kimber | Christopher Lum |
| 2 | 03/14/15 | All | Updated for AE512 | Noel Kimber | Christopher Lum | Christopher Lum |
| 3 | 03/14/17 | Various | Updated for 2017 | Christopher Lum |  | Christopher Lum |

# Table of Contents

Contents

Title Page 1

Publication Number 1

Date of Issue 1

Record of Manual Revisions 2

Table of Contents 3

Nomenclature & Glossary 4

Introduction 5

Contact Info 5

Other Documentation 5

Simulation Workstation Setup 6

Joystick 6

Matlab 6

Mathworks Dependencies 6

Using the UW Dependencies 6

Visualization Workstation Setup 7

FlightGear 7

Troubleshooting 10

X-Plane 12

Obtaining the Plugin Source Code 12

Using the Plugins 12

Troubleshooting 13

Simulator Setup 14

Set path 14

Test main code 14

Setting Up IP 14

References 16

Appendix 17

GitHub for Windows 17

Microsoft Visual Studio 17

Building the Plugins 20

Adding Models to Flight Gear 22

# Nomenclature & Glossary

See \\FlightOperations\Operations\AFSLFlightOperations\AFSLFlightOperationsManual.docx for generic nomenclature. The below table lists nomenclature that is specific to this system.

Table 2: Nomenclature and glossary of terms

|  |  |  |
| --- | --- | --- |
| **Term** | **Definition** | **Comment** |
| RCAM | Research Civil Aircraft Model | None |

# Introduction

This document describes how to set up and configure the RCAM simulator. This document is written for researchers in the Autonomous Flight Systems Laboratory (AFSL). As such, it assumes that you have access to the Perforce. For those that do not have access to this or are not part of the AFSL, some workaround notes may be provided.

## Contact Info

For more information, please contact:

Christopher Lum

lum@u.washington.edu

Autonomous Flight Systems Laboratory

University of Washington

## Other Documentation

* RCAM document [1]

# Simulation Workstation Setup

You will need some hardware and several software applications on the computer you will be using to develop software and run the RCAM simulation.

## Joystick

You will need a joystick or other human interface device to interact with the simulator. Some known models that are compatible are shown below. It is likely that any USB joystick will work.

|  |  |  |
| --- | --- | --- |
| http://ecx.images-amazon.com/images/I/41NHJ-ehx%2BL._SL500_AA300_.jpg | http://www.njuskalo.hr/image-bigger/informatika-sve-ostalo/joystick-microsoft-sidewinder-2-force-feedback-slika-19405548.jpg | http://gaming.logitech.com/assets/47835/12/3d-pro-gaming-joystick-images.png |
| Figure 1: Microsoft Sidewinder USB Joystick | Figure 2: Microsoft Force Feedback 2 USB Joystick | Figure 3: Logitech 3D Pro USB Joystick |

## Matlab

### Mathworks Dependencies

Matlab runs the main RCAM simulator. You will need the following Matlab toolboxes/blocksets.

1. Matlab
2. Simulink
3. Aerospace blockset ([link](http://www.mathworks.com/products/aeroblks/?refresh=true))
4. Aerospace toolbox ([link](http://www.mathworks.com/products/aerotb/))

If you do not have the Aerospace blockset or Aerospace toolbox, there may be university owned computers which have these packages.

### Using the UW Dependencies

1. In order to make use of these tools, ensure that the RCAMProject folder is added to your Matlab path. This can be found in the RCAMProject.zip file on Canvas.

# Visualization Workstation Setup

Best performance of the simulator can be achieved when a separate computer is used to visualize the output of the simulator. This allows the simulation workstation to focus on executing the simulation (ie solving ODEs) while a separate machine renders the environment in a visualization environment such as FlightGear or X-Plane. Note: X-Plane may not work if run on the same computer as the simulation due to excessive computing requirements.

## FlightGear

We recommend that you obtain FlightGear version 3.4.0

1. Visit <http://www.flightgear.org/download/>, find the link to download older versions of FlightGear and download the 3.4.0 version. If you cannot find the link from the FlightGear website, go to <http://ftp.snt.utwente.nl/pub/software/flightgear/ftp/Archive/Version-3.x/> and download Setup FlightGear 3.4.0.exe
2. Run ‘Setup FlightGear 3.4.0’ (1.0 GB) to install flight gear
3. Launch it, and follow ‘FlightGear Wizard’ to setup preferred aircraft. (To launch FlightGear, go to the FlightGear folder where you installed it, i.e. under Program Files, go to ‘bin’ and then run fgrun.exe)
4. In the last step of the wizard, click ‘advanced’ box as shown in Figure 4.

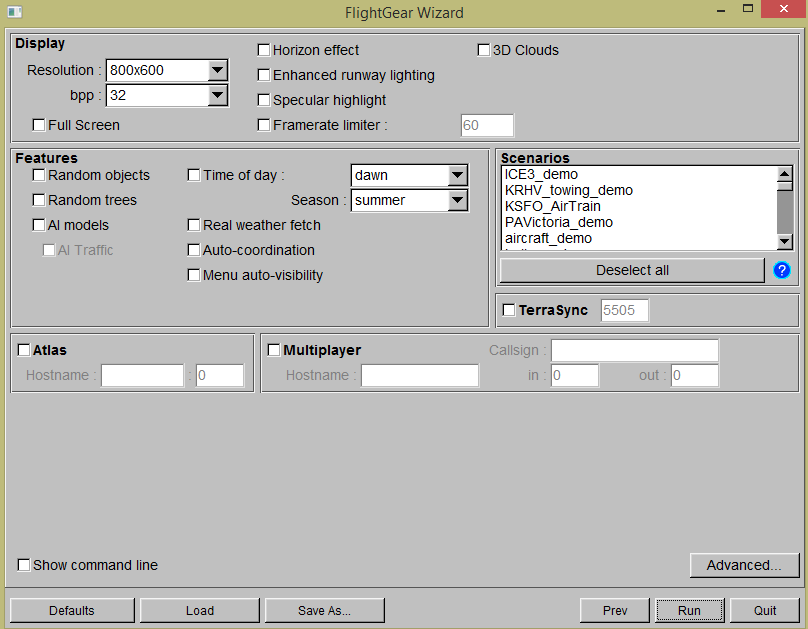


Figure 4: Setting advanced FlightGear settings

1. Go to: ‘Flight Model’ and set FDM: external as shown in Figure 5.

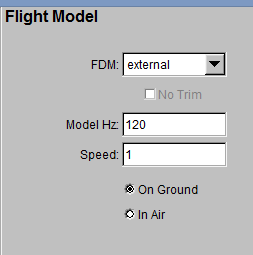


Figure 5: Setting FDM to external.

1. Go to: ‘Input/Output’, and set as shown in Figure 6.

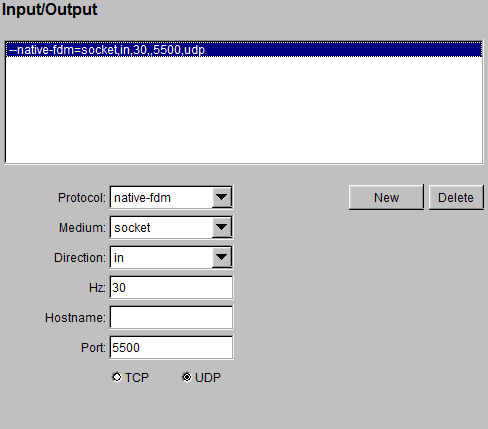


Figure : Input and output settings.

1. Exit by clicking OK. Settings should be similar to that shown in Figure 7.

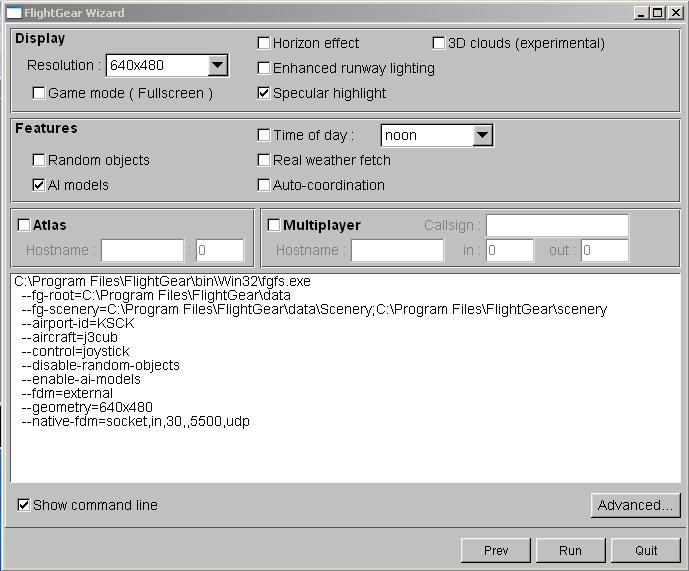


Figure 7: General settings for FlightGear

1. Click ‘Run.’ The model should be paused and awaiting input from an external source (the Simulink simulation).
2. Start the Simulink simulation and FlightGear should now visualize the scenario. [See the Simulator Setup section of this document before attempting to run the simulation]

### Troubleshooting

**Problem**

The FlightGear screen appears black.

**Solution**

FlightGear’s simulated time may be set to a night time condition with no light so the screen appears black. Simply change the simulation time of day to a daylight hour.

**Problem**

Aircraft appears to be flying over water, not land as specified in the simulation.

**Solution**

Ensure that the lat/lon inputs to the ‘FlightGear Preconfigured 6DoF Animation’ block are in units of degrees.

## X-Plane

In order to use X-plane for visualization purposes, you will need to use several plugins for X-Plane.

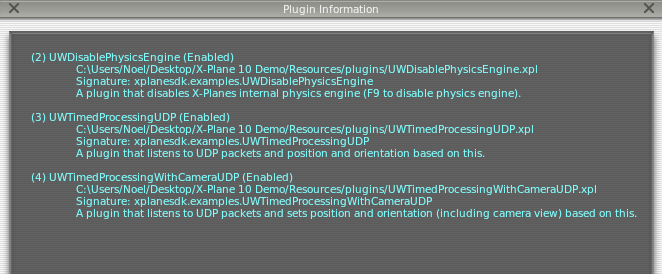
### Obtaining the Plugin Source Code

Navigate to the appropriate Canvas folder called ‘XPlane Plugins’ under the ‘Files’ tab. Download the .xpl files.

### Using the Plugins

These instructions are for X-Plane 10.

1. Copy the \*.xpl files to the X-Plane plugin directory on the machine running X-Plane (..\X-Plane 10\Resources\plugins)
2. Launch X-Plane by looking in the installation folder and executing the appropriate .exe file. Note that if you have a choice of executables to run, you should use the 32 bit version as this appears to be more compatible with the plugins.
3. Confirm the installation by checking: Plugins > Plugin Admin > Plugin Information



1. Now you can activate UWDisablePhysicsEngine by pressing F9
2. Pressing F5 activates UWTimedProcessingUDP for aircraft visualization
3. Pressing F4 activates UWTimedProcessingWithCameraUDP for payload camera visualization

(Note that when changing between two visualization plugins, make sure to disable one plugin by pressing the hotkey again before enabling another.)

1. Start the simulation in Matlab/Simulink and X-Plane should visualize the scenario. [See the Simulator Setup section of this document before attempting to run the simulation]

### Troubleshooting

**Problem**

In X-plane, after pressing F5, X-Plane appears to be frozen.

**Solution**

Run the simulation in Matlab/Simulink and X-Plane will visualize your scenario.

**Problem**

X-Plane remains frozen after starting the simulation in Matlab/Simulink.

**Solution**

Simply stop and run the simulation again in Matlab/Simulink and X-Plane should work.

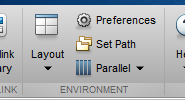
# Simulator Setup

The main simulation is executed by Matlab/Simulink.

## Set path

The codes used for this simulator utilizes some custom Matlab functions. These functions can be found under ‘UWMatlab’ folder.

1. Download the UWMatlab folder from Canvas. (This should be in the RCAMProject.zip file.)
2. Open Matlab. Home > Environment > Set Path > Add with Subfolders > select UWMatlab folder



1. This adds the entire UWMatlab folder and subfolder.
2. Repeat this process for any other files needed to run the simulation (i.e. the rest of the files in RCAMProject.zip).

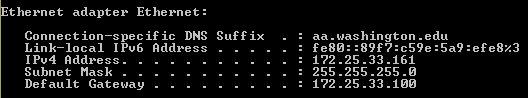
## Test main code

The main code used for the simulator is: MAIN\_RCAMSimulation.m. Test the code by running it. Upon running, the Simulink model will open.

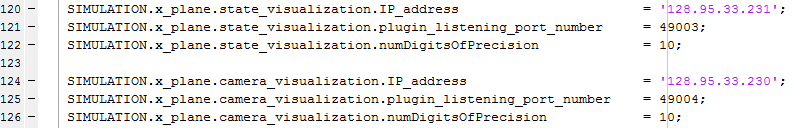
## Setting Up IP

The simulator communicates with other computers by sending packets via UDP. You have to set IPs for computers using FlightGear and X-Plane for visualization.

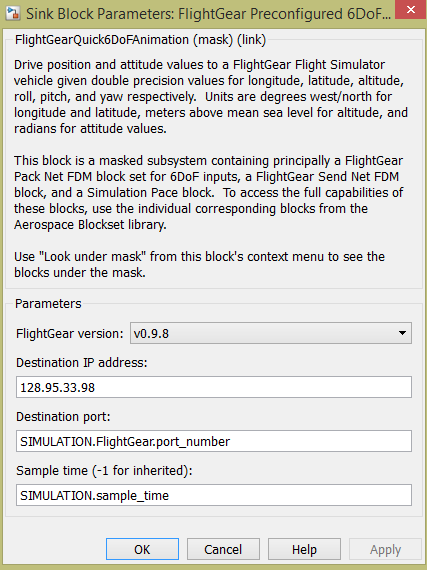
On the visualization computers, run > cmd > ipconfig, and find IPv4 address of each computers



If you are running the RCAMLoadConstants.m file, change the IP address in the appropriate lines to the IP corresponding to the computer running X-Plane. Save the change. If you are running your own file, manually input your IP into your respective FlightGear or X-Plane visualization blocks.



If you are using the RCAMSimulation\_model.mdl (Simulink that launches upon running the main code), go to RCAMSimulation\_model/Visualization/FlightGear\_visualization/FlightGear Preconfigured 6DoF Animation block, and change ‘Destination IP address’ to the IP corresponding to the computer running FlightGear. If not, go to your own FlightGear visualization block and change the IP address to that shown below.

If the ‘FlightGear Version’ is set to v0.9.8 as shown below, set it to v3.4.

# References

|  |  |
| --- | --- |
| [1] | P. Lambrechts, S. Bennani, G. Looye and A. Helmersson, "Robust Flight Control Design Challenge Problem Formulation and Manual: the Reserach Civil Aircraft Model (RCAM)," Group for Aeronautical Research and Technology in Europe, Europe, 1997. |
| [2] | Autonomous Flight Systems Laboratory, "AFSL Flight Operations Manual," Autonomous Flight Systems Laboratory, Seattle, 2016. |

# Appendix

## GitHub for Windows

Some of the source code for this project is located on GitHub. To make it easier to obtain code from this online repository, you may wish to install GitHub for Windows (<https://windows.github.com/> ) which is a series of tools for working with the GitHub online repository.

Please watch the lecture video for a demonstration of how to use this to clone (copy) code to your local machine.

## Microsoft Visual Studio

You will need Microsoft Visual Studio to compile some of the X-Plane plugins. It is recommended that you have Visual Studio 2012 or newer.

If you are using a university owned computer, you can install Microsoft Visual Studio via the UWARE program.

If you are using a personal machine, you can obtain Microsoft Visual Studio though the Microsoft DreamSpark program.

1. Visit the DreamSpark Student page: <https://www.dreamspark.com/Student/Software-Catalog.aspx>
2. Locate and click “Visual Studio Professional 2013 with Update 4” or “Visual Studio Professional 2012” as shown in **Error! Reference source not found.**.

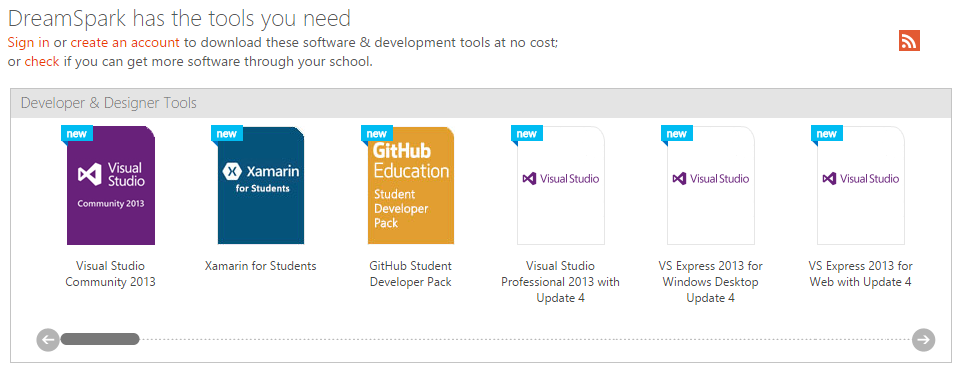
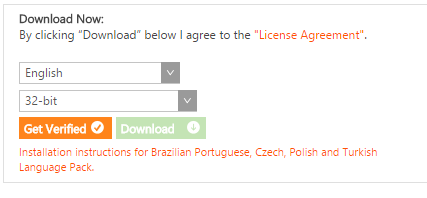
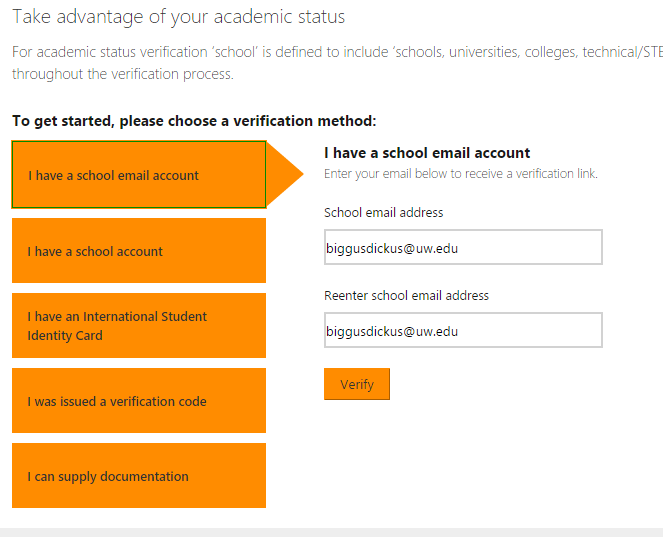


Figure 8: Choosing your version of Visual Studio

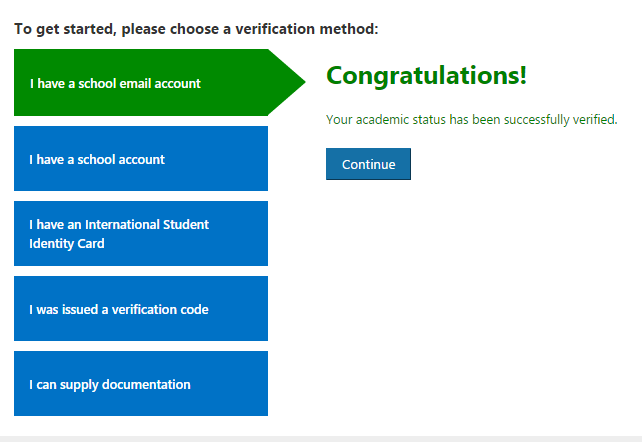
1. Click “Get Verified”



1. You’ll be asked to log in with your Microsoft account.
2. Continue with the account activation. When prompted to verify your student status, use your UW email (e.g. johndoe@uw.edu).



1. After the verification sequence, your student status is verified.



1. Go back to the download page. Download the installer and get key. (You may need a new disc or DVD to burn the installer to first).



1. Install Visual Studio.
2. The first time running Visual Studio will ask what language it should optimize its layout. Select C++.

### Building the Plugins

1. Open the UWPlugins.sln using Microsoft Visual Studio.
2. Ensure that you have the “Solution Explorer” pane enabled as shown in Figure 9.
3. Ensure that the “Solution Configurations” is set to “Debug”. This is achieved by choosing “Debug” in the drop down menu shown in Figure 9.

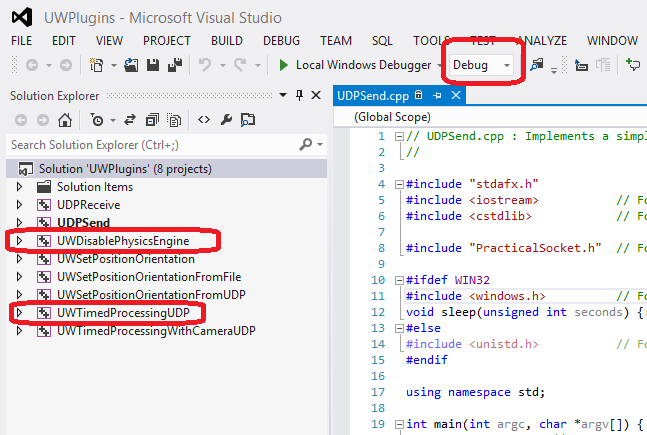


Figure 9: UWPlugins.sln open in Visual Studio. Note the projects are shown in the solution explorer pane on the left.

1. Build both the ‘UWDisablePhysicsEngine’ and ‘UWTimedProcessingUDP’ projects by right clicking on the project in the solution explorer and choosing “Build”.
2. Navigate to the “Debug\Plugins” folder. You should now see UWDisablePhysicsEngine.xpl and UWTimedProcessingUDP.xpl files. These are the debug versions of the plugins

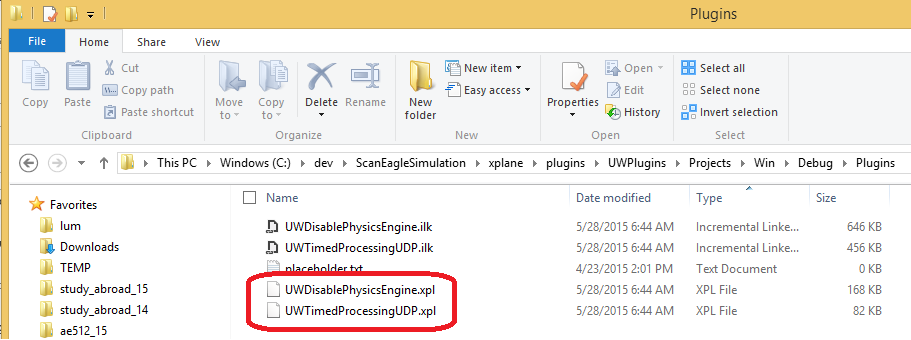


Figure 10: Debug versions of the plugins.

1. Change the “Solution Configurations” to “Release” (Figure 9) and build release versions of the plugins. These will be located in the corresponding “Release” folder.

### Adding Models to Flight Gear

These notes were compiled by Alex Summers ([aws17@uw.edu](mailto:aws17@uw.edu)) during the Summer of 2016.

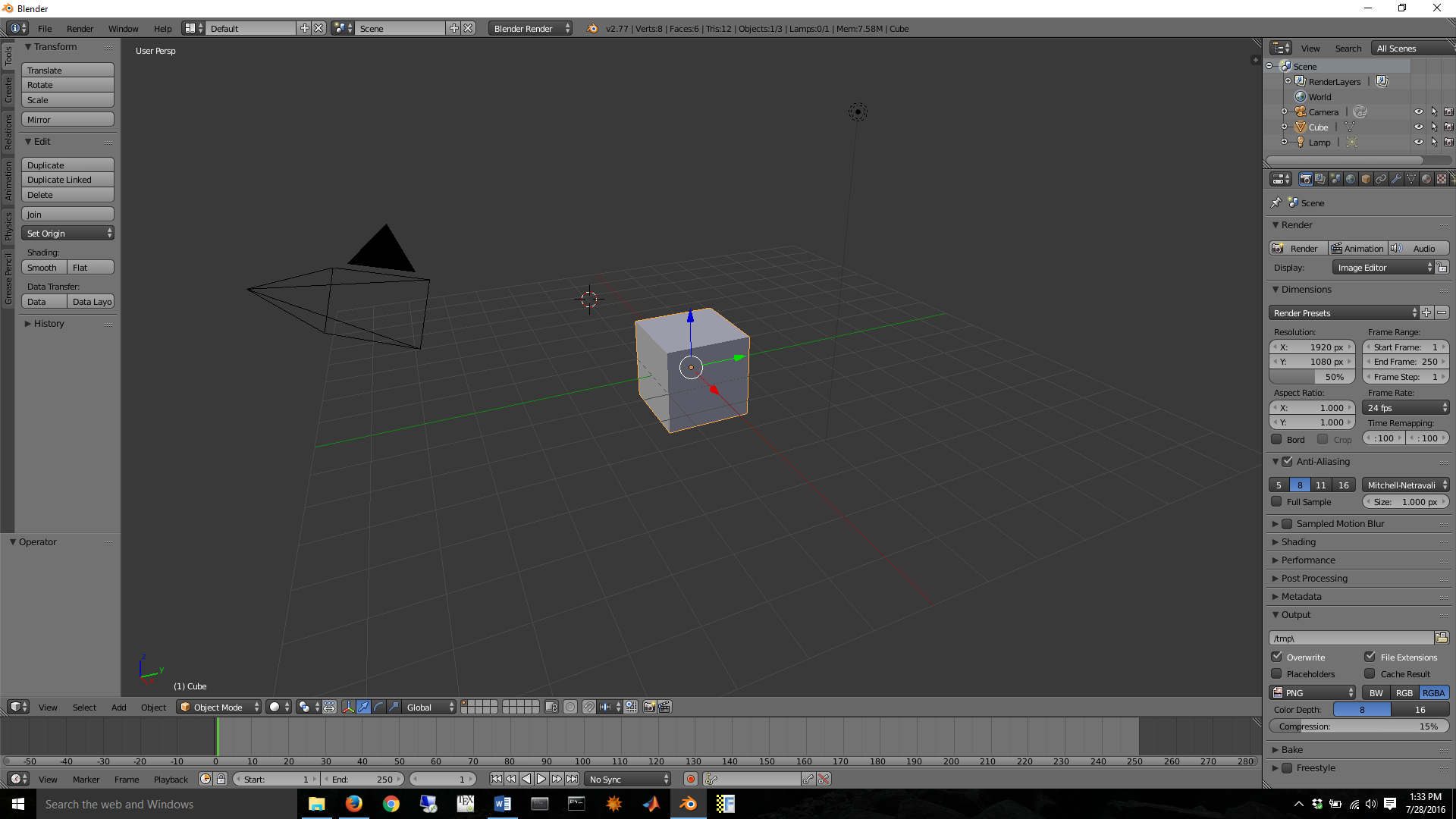
**Part 1: AC3D File Type**

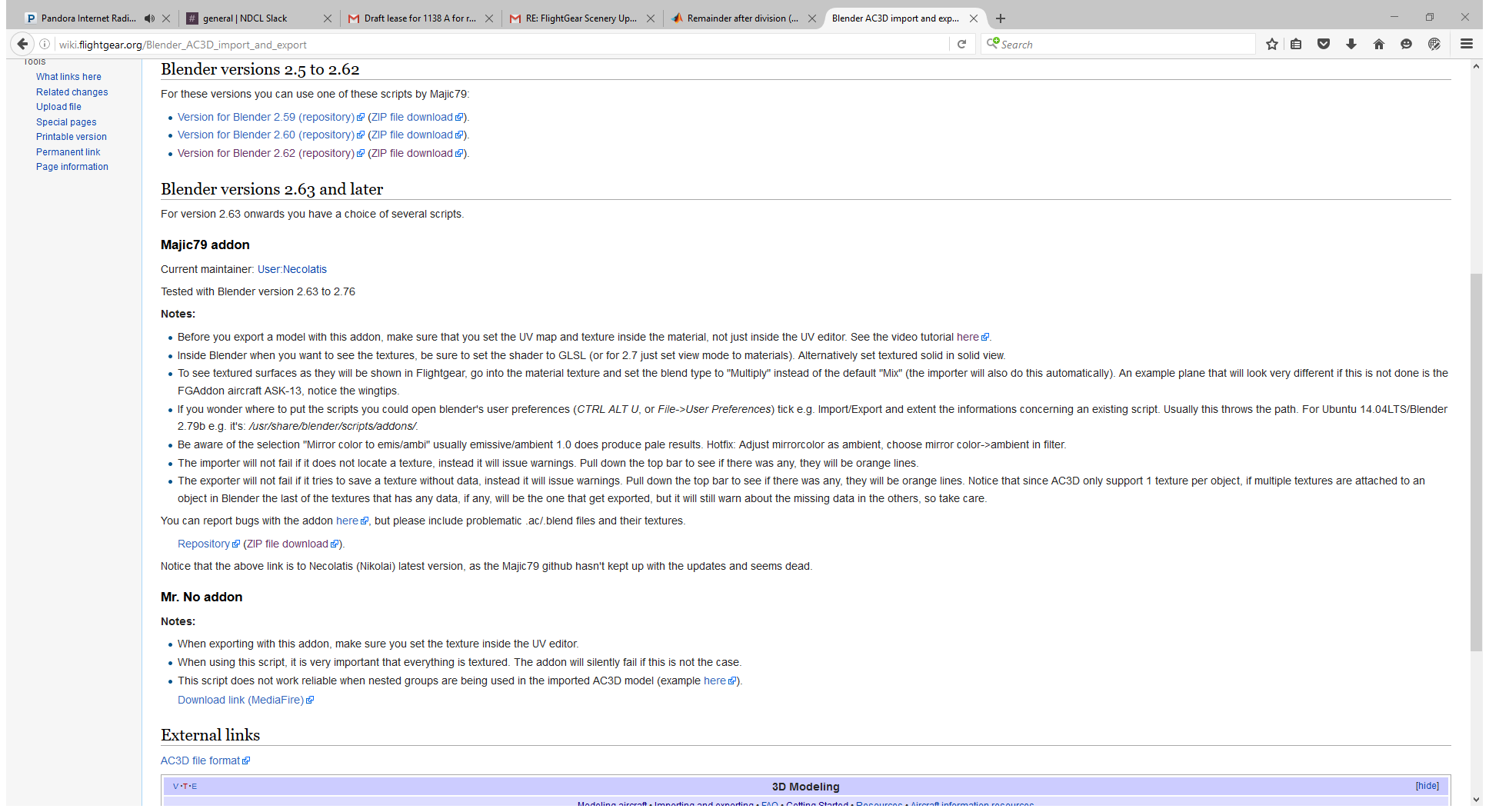
To input a 3D model into flight gear the model must be an AC3D file, and unfortunately it’s harder to do than you’d think.

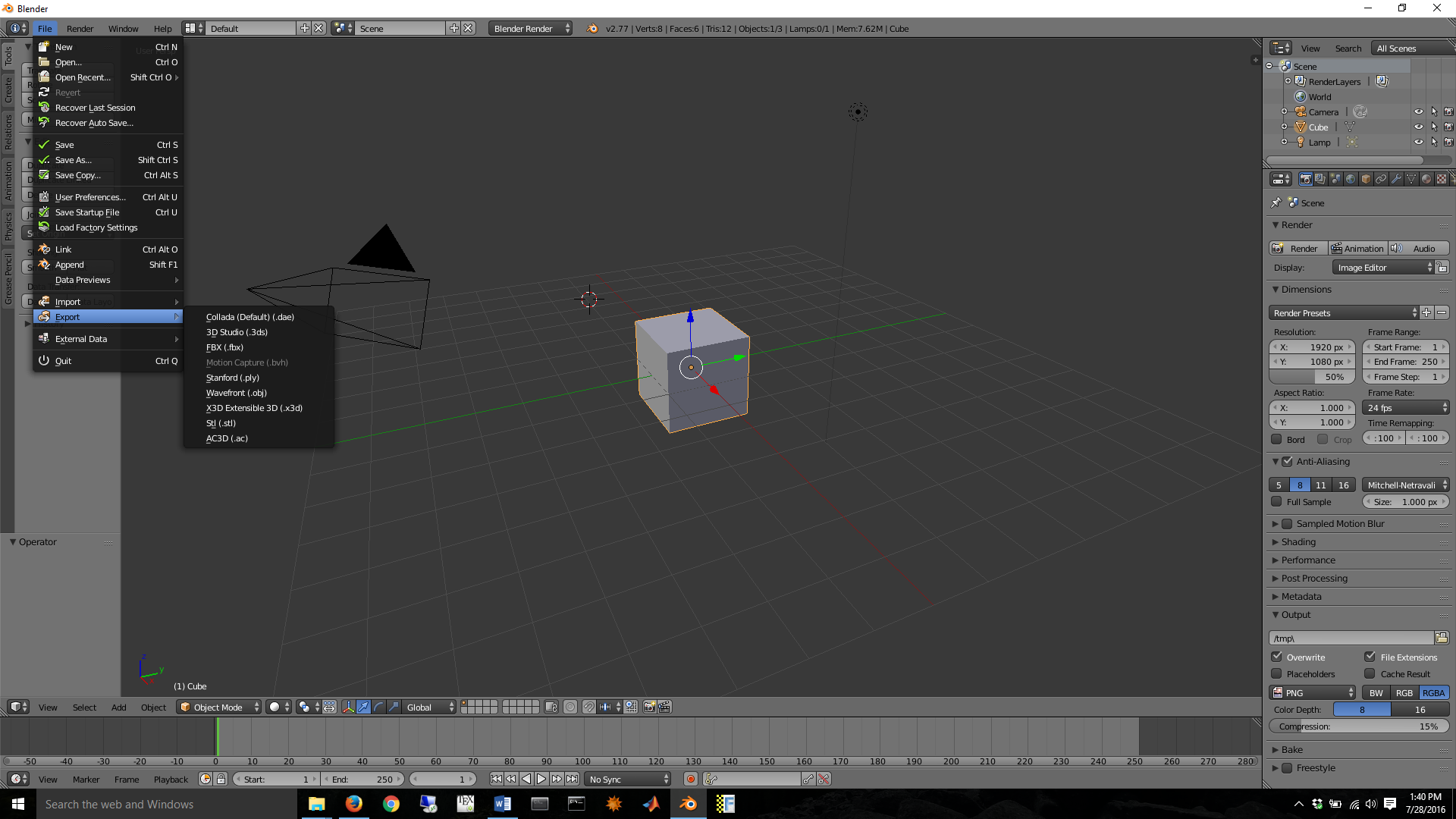
The following method explains how to achieve this file type using a free-source program called Blender.

If you have an alternative method of creating this file move to part 3.

1. Create a CAD file of the geometry you would like to apply in FlightGear.
2. Export this file as a .stl file.
3. Download Blender, <https://www.blender.org/download> , to your computer using the correct installer for your device.
4. Open the window and check to make sure download has worked properly. You should see this screen or something similar.



1. As things are, you can export files in various ways but unfortunately not a .ac3d file so we need to modify the program.
2. Go to <http://wiki.flightgear.org/Blender_AC3D_import_and_export> , and download Majic79 add- on for 2.63 and onwards.
3. Unzip the file and open the “read me” in a browser( the html description proves clearer on a browser than notepad). Follow the installation instruction enclosed within the “Read ME”.
4. When this is complete you should have the option to export as an .ac3d file from blender.



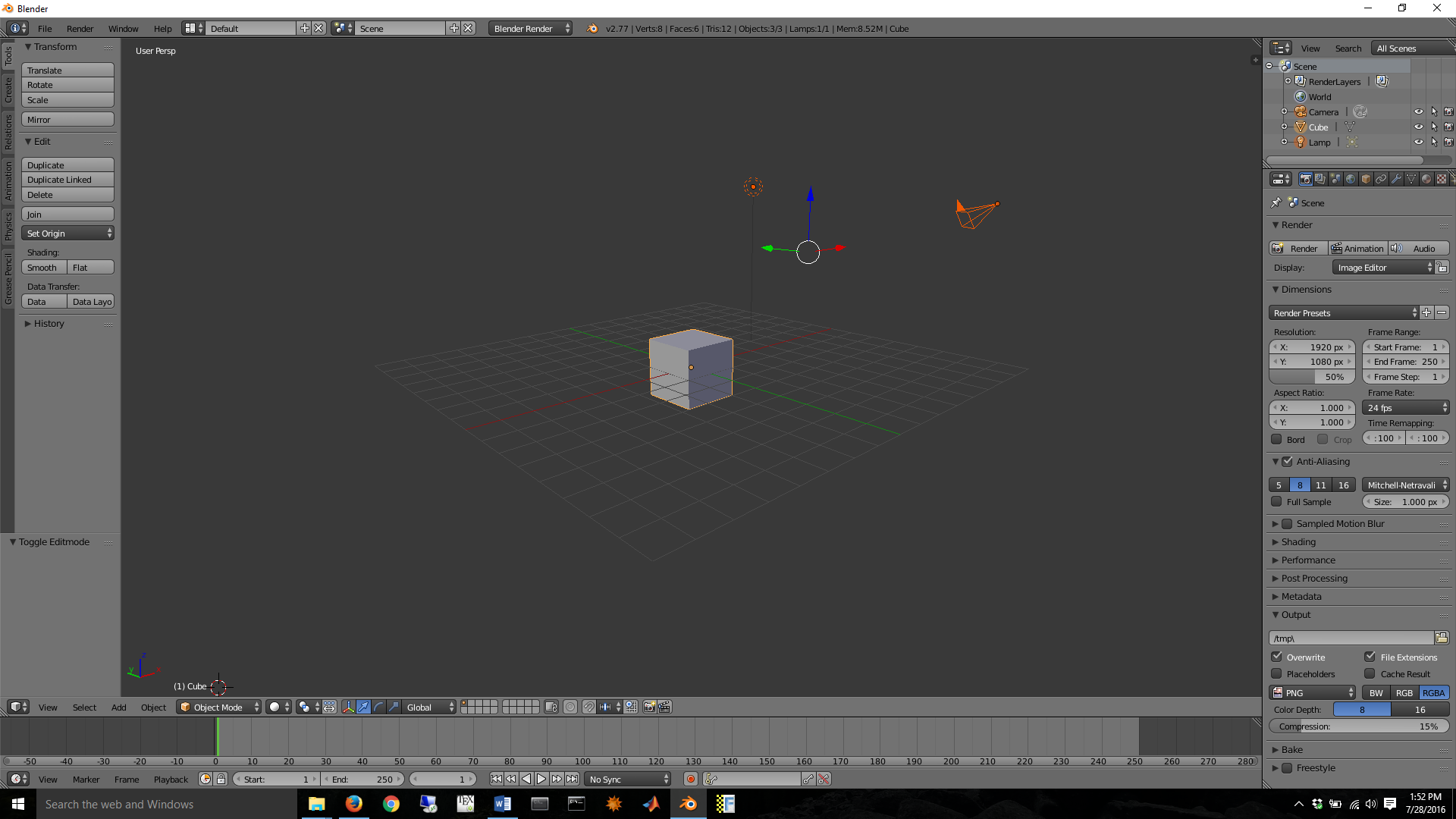
**Part 2: AC3D File Export**

At this point you can import a .stl file to blender by going file->import->.stl, but you will run into an issue if you try to export immediately to .ac3d.

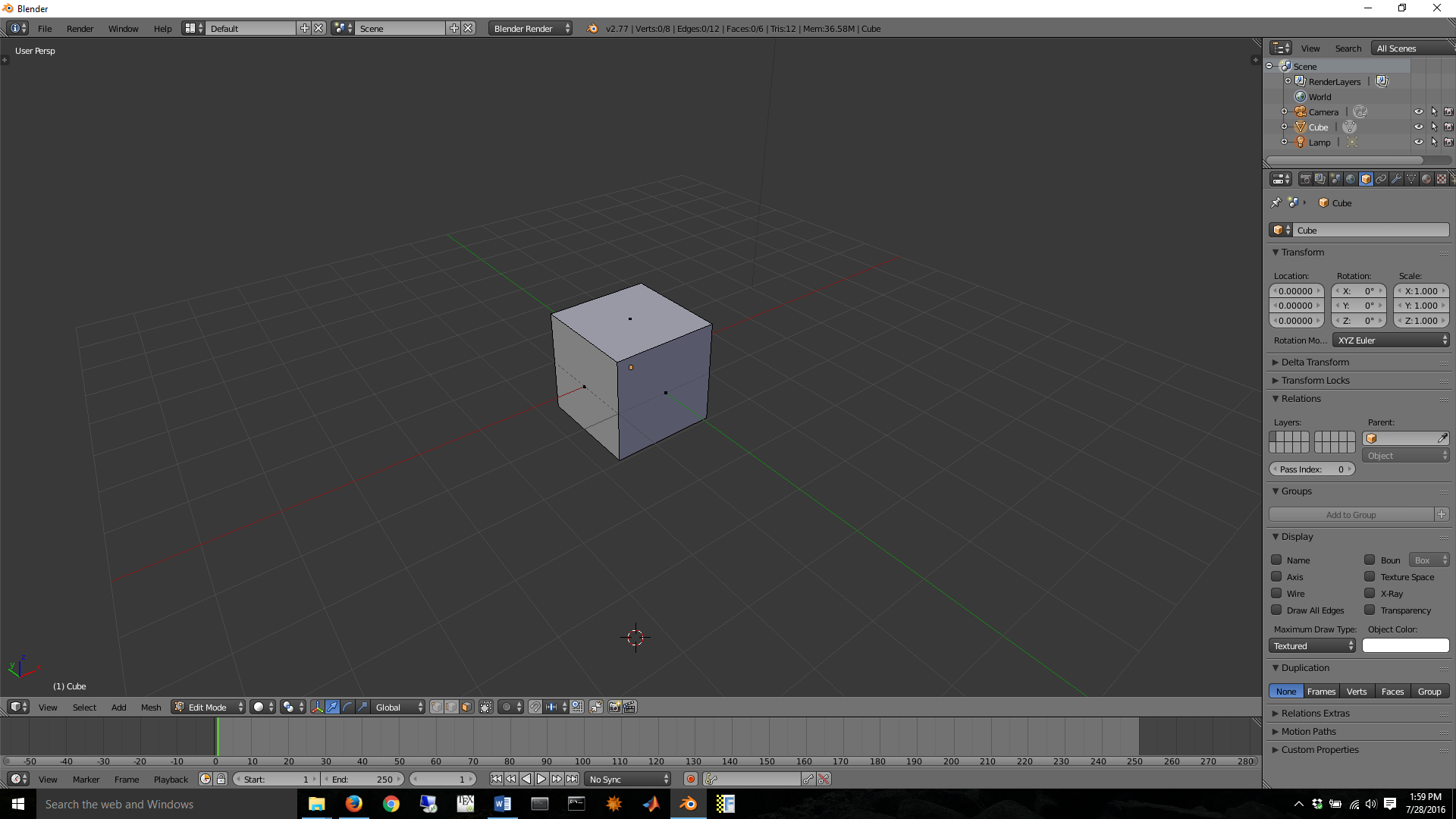
The blender program will not let you export the file until you define the material.

Warning: Blender is not the most user friendly program, it can be quite unintuitive. There are various YouTube tutorials that can help and many take different routes than I have below.

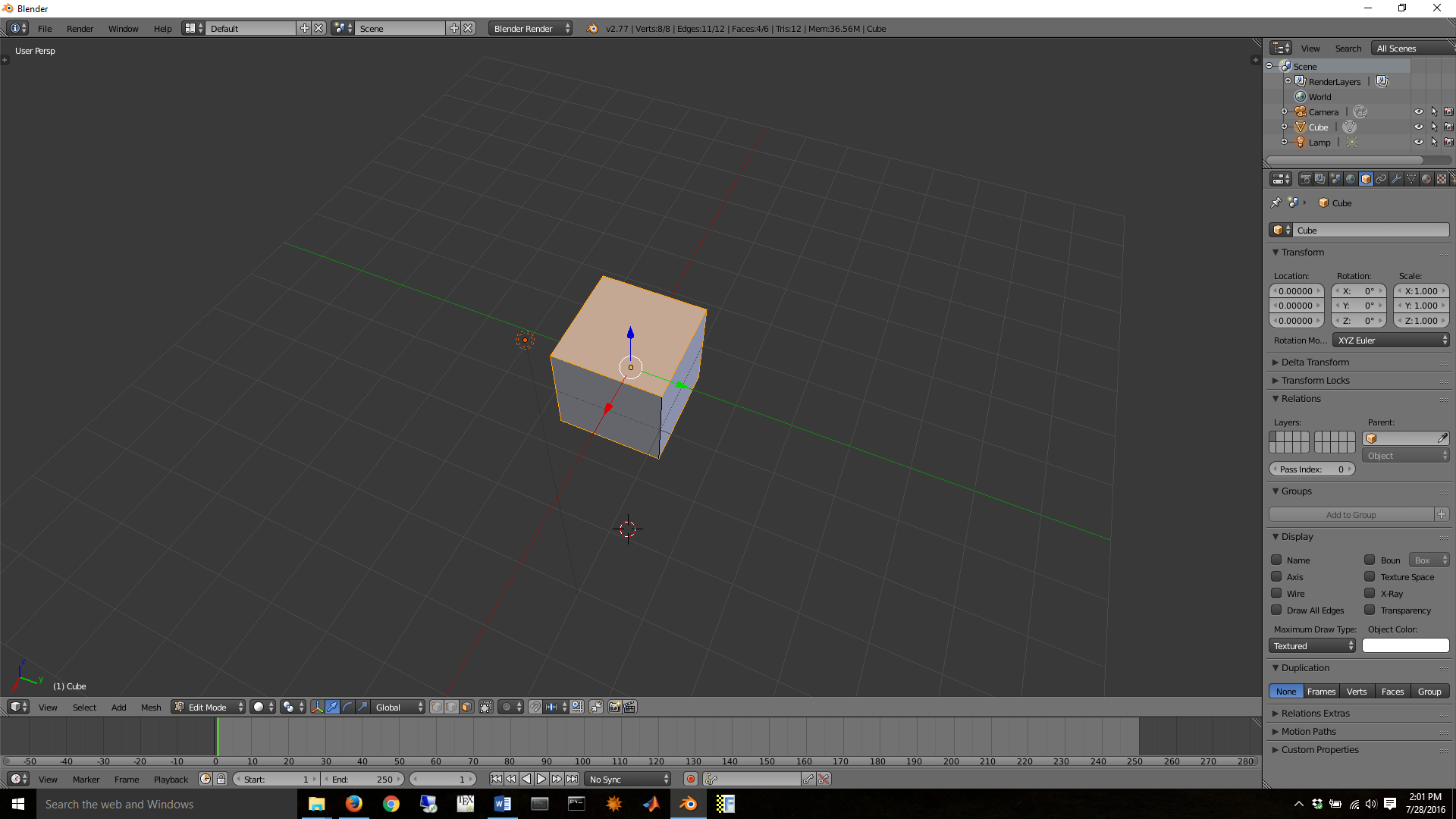
1. Clear the space of its original contents, most likely a cube. I find the easiest way to do so is holding control and the left mouse button and creating a selection web. Then simply delete them all.
2. Import your file into blender you should see your object in 3D space( I will continue forward using the cube). You can move it around using a combination of right and left clicks.
3. Put your mode into “edit mode”.

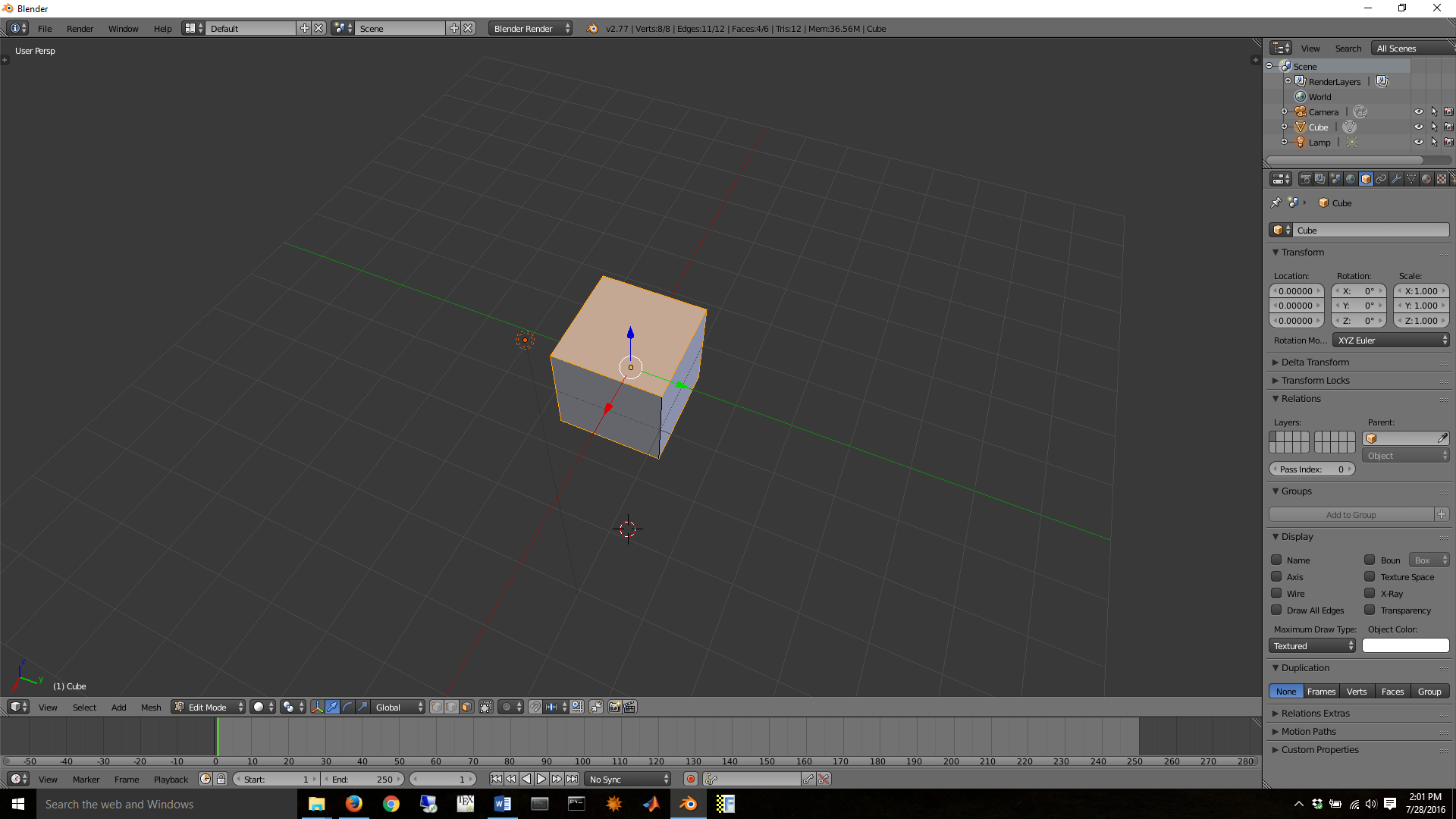


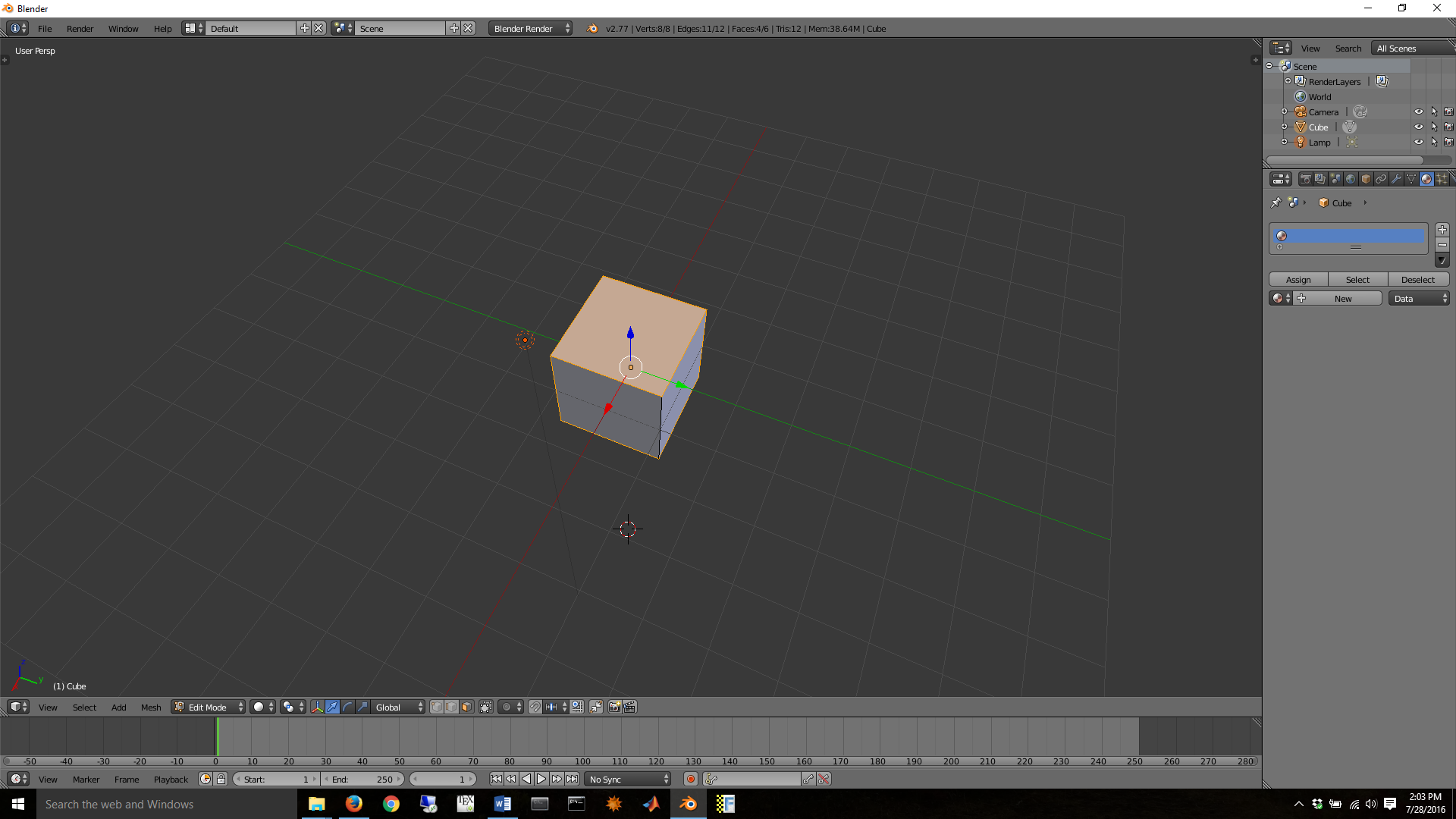
1. Once in edit mode you can select just by clicking a node, edge, or face. I personally prefer face.



1. Once you have selected your faces, such that they are all highlighted, you need to add a material.





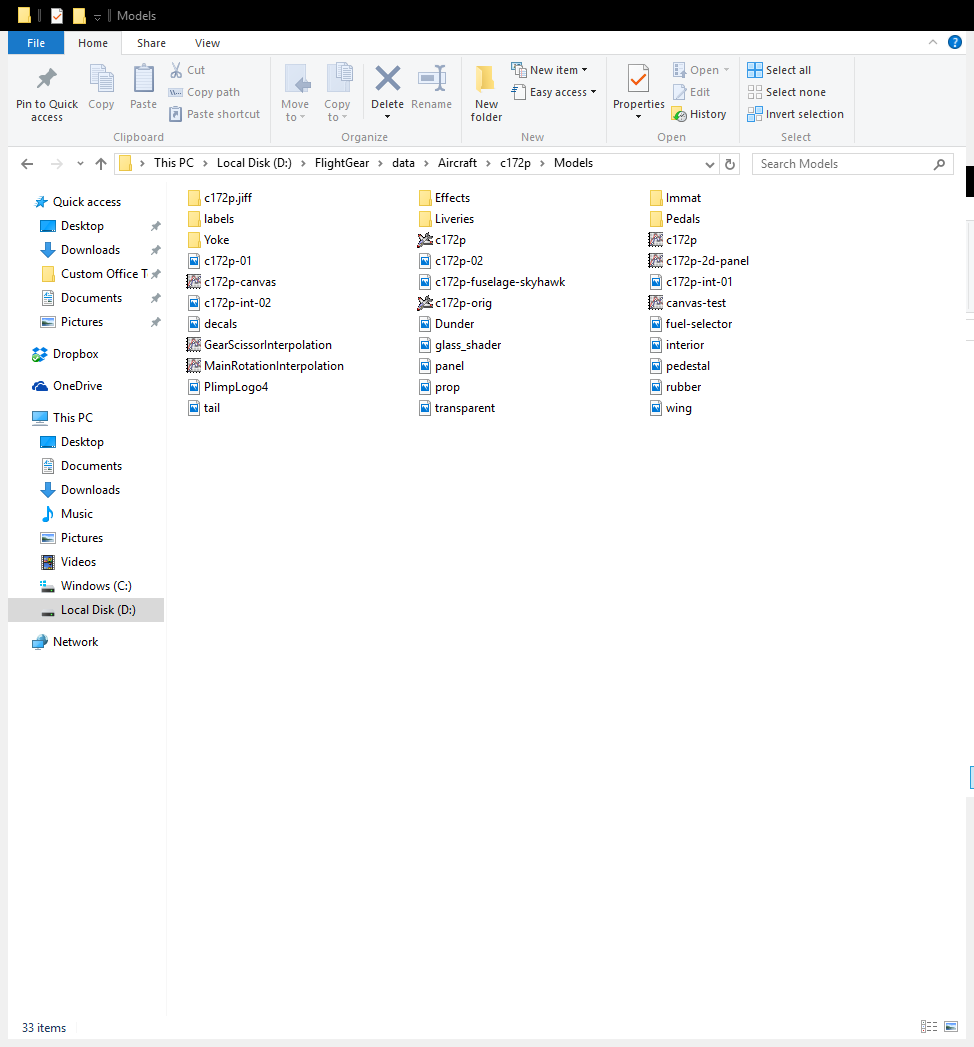


1. Once you have created a material with color you like, click “assign” to assign the material to the highlighted faces.
2. At this point you can continue to alter and design you model, more complex options result in the use of textures and the UV editor, or you can export the file out as an .ac3d model as is and it should work.

**Part 3: Flight Gear**

Now the file can actually be input into flight gear. There is a proper way of doing so, and if your’e interested in a clean and animated model in flight gear I’d suggest reviewing <http://wiki.flightgear.org/Howto:3D_Aircraft_Models>. A less clean method which is more time effective but less pretty and more attuned to working the system is noted below.

1. Open the flight gear file in your desktop. Then proceed FligthGear->data->Aircraft->c172p->models.
2. Now you should see a window similar to this.



Click and rename the original c172p .ac3d model (the flie called c172p with an airplane icon) to whatever you like.

1. Place your .ac3d model into the folder and rename it as c172p.
2. Go into flight gear, and under available aircraft click on the c172p aircraft, your model should be visible.
3. Run the flight gear simulator and your aircraft should be the image seen in flight gear, there will be a left over cockpit chair, but if you move your model to the correct place this should be covered.

Note: Additional resizing and moving in blender may be necessary to create a proper sized model at the right place. The model should work like a c172p but if use the flight gear block in Simulink you can run it with custom dynamics.