

Introduction to Blender 3D

an excerpt from:



Pages and figures in this document retained the numbering from this original book

Conventions

For the tips about using the keyboard and the mouse I have assumed that you have a standard:

- US keyboard, with 102 keys (you will find in this book some comments about non-standard laptop keyboards);
- Three-button mouse (in fact: two buttons and the wheel in the middle. When you click the mouse wheel, it acts like the third button).

Command invocations are marked as follows:

Menu → Command means invoking a command named *Command* from a menu named *Menu*. More arrows may appear, when the menus are nested!

Panel:Button means clicking a button named *Button* in a dialog window or a panel named *Panel*. Panels are parts of Blender screen (for more explanation — see „Volume II: Modeling”). Sometimes I may also mention other dialog controls, like a checkbox or a drop-down list.

Pressing a key on the keyboard:

[Alt]-[K] the dash (“-”) between characters means that both keys should be simultaneously pressed on the keyboard. In this example: while holding down the **Alt** key, press the **K** key;

[G], [X] the comma (”,”) between characters means, that keys are pressed (and released!) one after another. In this example type **G** first, then **X** (as if you would type “gx”).

Pressing one of the mouse buttons:

LMB left mouse button

RMB right mouse button

MMB middle mouse button (mouse wheel **pressed**)

MW mouse wheel (when it is **scrolled**)

Last, but not least — the formal question: how should I address you? Typically the impersonal form (“something is done”) is used in most manuals. I think that it makes the text less comprehensible. To keep this book as readable as possible, I address the Reader in the second person (“do it”). Sometimes I also use the first person (“I’ve done it”, “we did it”). It is easier for me to describe my methods of work in this way¹.

¹ While working on this model I thought about us — you, dear Reader, and me, writing these words — as a single team. Maybe an imaginary one, but somehow true. At least, writing this book I knew that I had to properly explain to you every topic, with all details!

Chapter 1. Software Installation

In this chapter we will discuss briefly how to install the software we will use in this volume. It is just one program:

- **Blender**: the 3D modeling software;

This software is released under the GPL (i.e. Open Source) license. It means that no one can charge any fees for it. (Of course, you can still make voluntary donations. You can find detailed descriptions how to do this on the websites of these projects).

I suppose that most of the readers of this book use a computer with the Windows operating system. In the installation instructions I have concentrated solely on this environment. I also use it on my computer. I have no experience with Linux nor Mac OS, so I will not write about the things that I do not know. If you are using a non-Windows system — find the installation instructions of these programs on their websites.

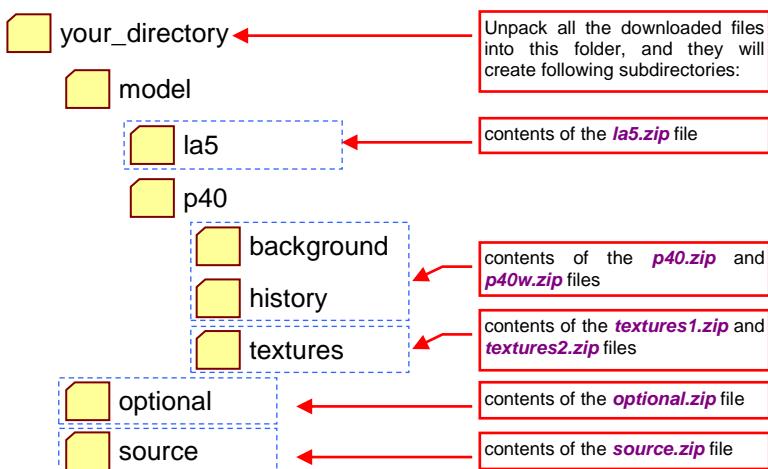
I also prepared some additional resources for this book, and packed them into a few **.zip** files, for downloads:

- You can download the files that accompany this book from http://airplanes3d.net/va3_e.xml. It contains their latest versions (I update them from time to time, to match the current Blender release)¹.

From this web page download following files:

- **source.zip**: contains useful add-ons, scale plans, a few factory blueprints of the P-40, and some other reference materials;
- **la5.zip**: contains the exemplary model, used in Chapter 2;
- **p40.zip, p40w.zip**: contain the subsequent P-40 models (*.blend files), and background pictures. Each *.blend file from their *history*\ folder contains the result of a section from this book

Unpack all these **.zip** files into a single folder. They will create following structure of subdirectories (files not listed above belong to the other volumes of this book):



¹ On this web page I will also place answers to the readers' Frequently Asked Questions. They will appear some time after the release of this book.

1.1 Blender Installation

I used Blender **2.7** in this book. You can also install previous versions of this program in the manner described below.

Download the installation file from the Blender Foundation website: <http://www.blender.org>. When I was writing this edition of the “Virtual Airplane”, they were available at <http://www.blender.org/download/> (Figure 1.1.1). However, I cannot guarantee that they will always be in the same place. If you find that this virtual directory is no longer valid, start from the main page of this website¹.

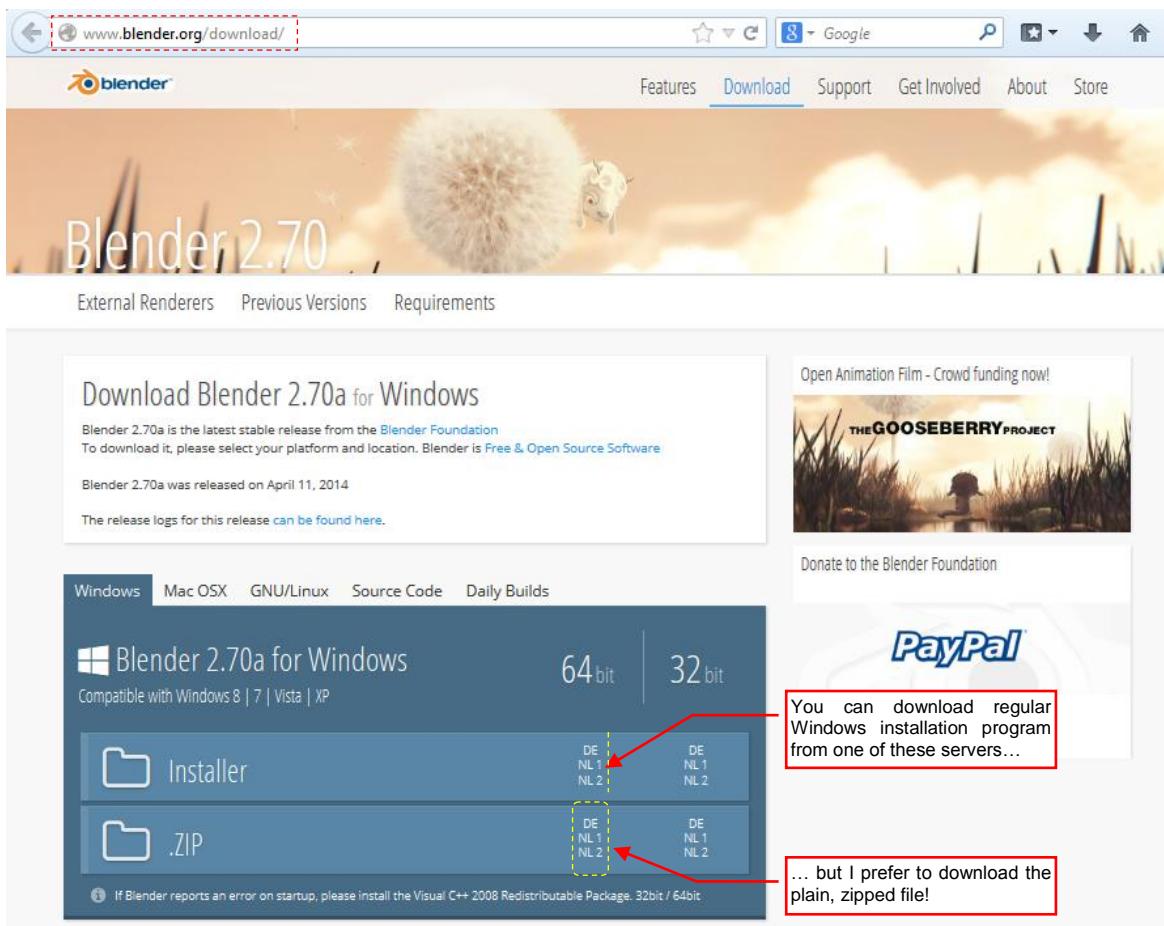


Figure 1.1.1 The page with Blender installation packages

At this address you will find shortcuts to the Blender versions compiled for different operating systems (Windows, Linux, Mac OS ...). In Figure 1.1.1 I have chosen the 64 bit package. This is the preferred version². The 32-bit Blender version is intended for the older, 32-bit Windows versions (like Windows Vista or once popular Windows XP).

Each version for Windows is available in two forms: the installation program and the ***.zip** archive. If you like the Windows installation programs, you can download it from there and run on your computer. It will ask you about the folder in which to place Blender, and after a while you will have it installed.

¹ From this site you can also download the earlier versions of Blender. I mention this because you may read this book some time after its publication. The best Blender version to follow this tutorial is 2.7 (in the time of writing: 2.71).

² The 64-bit version of this program has access to more RAM. It allows you to use textures larger than 4000x4000 pixels. This feature can be very useful when you want to reproduce the model surface with more accuracy. In this book the largest texture has size of 4096 x 4096 pixels. However, if you start rendering a very big picture — for example, an A4 page in resolution of 300dpi, and textures of 8192x8192px — the 32-bit program will end with an error, while the 64-bit version will do it correctly.

However, I always prefer to know exactly what is happening to my computer¹. The point is that Blender practically has no external dependencies. Therefore, I believe that the best way to install Blender is to unpack its ***.zip** file. In this manner you know exactly what you are doing. Just:

1. download the appropriate ***.zip** file from the BF page (Figure 1.1.1). (It contains a folder with ready to use program files);
2. unpack content of this package to your hard disk, for example to the **C:\Program Files\Blender** folder;
3. add the shortcut to **C:\Program Files\Blender\blender.exe** file to your desktop and/or to your **Start** menu;
4. optional: associate the ***.blend** file extension with the **blender.exe** (just to double-click any ***.blend** file, and Windows will ask you about the program to open it);

If you have any doubts about the course of this simple procedure, you can find the detailed description on how to do it on page 253.

When you want to upgrade Blender to the newer version — you do not have to run the installation program once again. (Though, of course, you can do it). Do it in the following way:

1. change the current **Blender** folder to “**Blender-XX.X**”, where XX.X is the number of the version it contains. For example: before installing the version 2.72 I have changed the name of existing **Blender** folder to **Blender-2.71**;
2. download from the BF page the ***.zip** file with the new Blender version;
3. unpack its contents to the **C:\Program Files\Blender** folder (you have to create a new one);

In this way, all the **Start** menu shortcuts, and the association with the ***.blend** file extension will be automatically transferred to this new Blender version — you do not have to change anything more.

When you start the new Blender version for the first time, it will propose you to import your settings from the previous one (Figure 1.1.2). Just do not miss this moment and click the **Copy Previous Settings** button on its splash screen:



Figure 1.1.2 The Blender splash screen, as on the first run of a new program version

¹ Thanks to this I have used the same Windows installation for many years — and it has never been reinstalled! What's more, even with the passage of time it has not slowed down, as I have often seen on the computers of other users...

You can always use previous versions of Blender when you keep them in renamed folders. Just run the `blender.exe` file from such a directory, instead of using the desktop shortcut.

- You can use many different versions of Blender simultaneously on your computer. Just place each of them in a separate folder.
- Blender does not need an external Python installation. (You may encounter references about it on the Internet). The external Python interpreter was an optional component in Blender 2.49 and earlier releases. Since the version 2.5, Blender uses its own, embedded Python interpreter. This is because the whole Blender user interface (windows, buttons, and menus) is dynamically created by Python scripts, every time you open the program!

Chapter 2. Introduction to Blender 3D

In this chapter you will learn the Blender basics¹. It discusses its GUI, navigation in the 3D space, and basic object manipulation. You will use the shortcuts described here so often that soon you will do it instinctively.

To get somewhere, you need to make the first step. Some claim that the first one is the hardest. On the web forums you can often read posts like this: "I would like to play with Blender, but at the first attempt I gave up". I also wanted to play with it. So I followed a tutorial or two and soon became hooked on this software.

I think the myth that Blender is a difficult program is exaggerated by those who want to learn everything at once. Dear friends, it is not possible. Sometimes you have to sit down and read, and learn². So far, nobody has invented such pills like those from the "Matrix", which would allow you to swallow a whole book and know everything right away.

Once I carried out comparative studies of the user interface of different 3D systems. I can honestly say that Blender has really a cool, consistent front end. It is just a little unconventional. Anyway, in version 2.5 its UI was rewritten from scratch, to become even easier. In the subsequent sections of this chapter I will describe, step by step, the conventions of the Blender environment. When you learn them, you will discover that it is not so difficult to master the whole thing. It enforces a good habit of working with the computer using both hands (usually with the left one on the keyboard, and the right one — on the mouse).

Most of this training we will carry out using a model of the La-5 fighter. You will open this file, examine its content, and learn basic methods of editing (transforming) the Blender objects. We will do this by moving and rotating the camera around the scene. (Yes! Like in a real studio, we can have the cameras here!). All in all — after reading this chapter you will already know how to "pose" a model in Blender. At the end we will also render the final picture of this airplane.

- In this chapter we will use some materials from the [source.zip](#) and [la5.zip](#) files, which accompany this book (see page 18 to get the address of their download site)

¹ This chapter contains detailed instructions how to use Blender. I have broken here the rule adopted in this book (the first part of the book describes what to do, while the second part — „Details of Program Usage” — how to do it). I did so because the material presented in this chapter is necessary for all further work.

² A few years ago I was examining skills in the office software of a group of graduates. Each participant of this test had to declare his/her level of experience in the word processor. Nobody stated his experience level below 80%. Then they received tests, appropriate for the declared knowledge. Few of them demonstrated skills on the level better than 50%. So why are such programs considered to be easy to learn? Because hundreds, or even thousands of programmers had polished their user interface for a decade. This tuning was intended to achieve quite a specific learning curve. The absolutely basic steps had to be easily conducted by an average office worker. (In most cases, not particularly fluent in the computer use). The remaining features are the domain of few insiders: the individuals brave enough to delve into such esoteric and difficult to understand texts as in the "Dummies" books. The Blender learning curve is different — steep, but in steady pace, without excessive facilitation at the outset. On the other hand, there is no great gap between the first step and the next ones.

2.1 Screen layout

Figure 2.1.1 shows how your screen looks like when you open Blender for the first time:

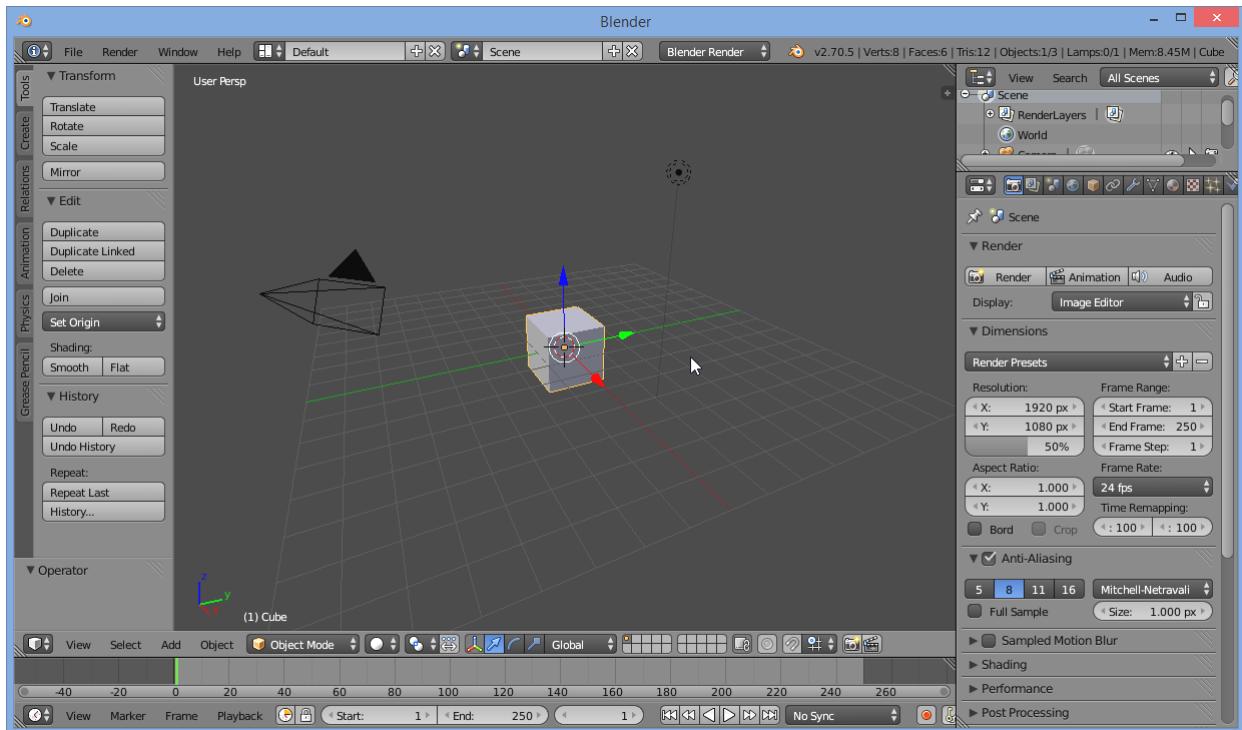


Figure 2.1.1 The Blender initial screen

It does not look like a typical Windows application, does it? I confess that when I started working with rewritten from the scratch version 2.5 of Blender, for the first 15 minutes I just sat, staring at the screen. Everything on it was so “refurbished” that the whole thing was completely new for me! I even did not move the mouse — I was not sure where to begin. It happened, although I had worked for four years in previous versions of Blender (2.4), and even before this I looked at these Blender 2.5 beta versions once or twice. After those first 15 minutes of contemplation I began to repeat the simplest steps, then looked into some tutorials — and it went, somehow. I hope that the memory of this beginning will help me to write this introduction better.

The screen layout of Blender seems to be prepared for the newer displays that have an aspect ratio of 8x5. (The previous Blender versions — up to 2.4 — were better fitted to the aspect ratio of the old monitors: 4x3). Blender is written for several operating systems: Linux, Mac OS and Windows. In each of them it looks (and works) in the same way¹. Inevitably, the user interface of this program has a unique style that distinguishes it from the others. Let's start by discussing its basic elements.

¹ How did they achieve such compliance? It is possible because Blender “paints” its menus and buttons using the OpenGL standard commands. In each of these operating systems there is a set of procedures that can be used by applications to create realistic, three-dimensional images. Such a set in the programmers' jargon is called a “library”. These libraries implement procedures of the OpenGL command set. This set, initially named GL, was developed in the early 90's of 20th century for Silicon Graphics workstations. (In that time they were true “rockets”, with their hardware graphics accelerators. Such a workstation was a dream for all computer graphics fans, and had appropriately high price). Shortly afterwards GL converted into multi-platform OpenGL, began to be used by many CAD, CAM and GIS applications, in different operating systems.

In 1995 it was added to Windows NT, then to Windows 95. (Although their manufacturer — Microsoft — had already promoted a competitive product, named DirectX). Blender always used OpenGL to draw the content of its scenes. Interestingly, at some point its author came up with the idea that the same commands can be also used to draw the two dimensional elements of the screen. In the result the whole Blender user interface is drawn using OpenGL. This way, its UI code does not depend on the operating system. This solution gives also a relatively small size of the program (there is no need to use complex, standard window libraries like [wxWidgets](#), [Qt](#), or [GTK+](#)). Another interesting “side-effect” is the ability to smoothly resize (scale) all the user interface elements.

The Blender screen is divided into rectangular areas. They are also referred as “windows”¹. Unlike in the conventional windowing systems, Blender windows never overlap. Personally, I think it is an advantage. The overlapping windows look good only in the advertising materials of the popular editors. When it comes to the real work, everyone sets them in a way to see all of them at once. On the default Blender screen you can see five windows (Figure 2.1.2):

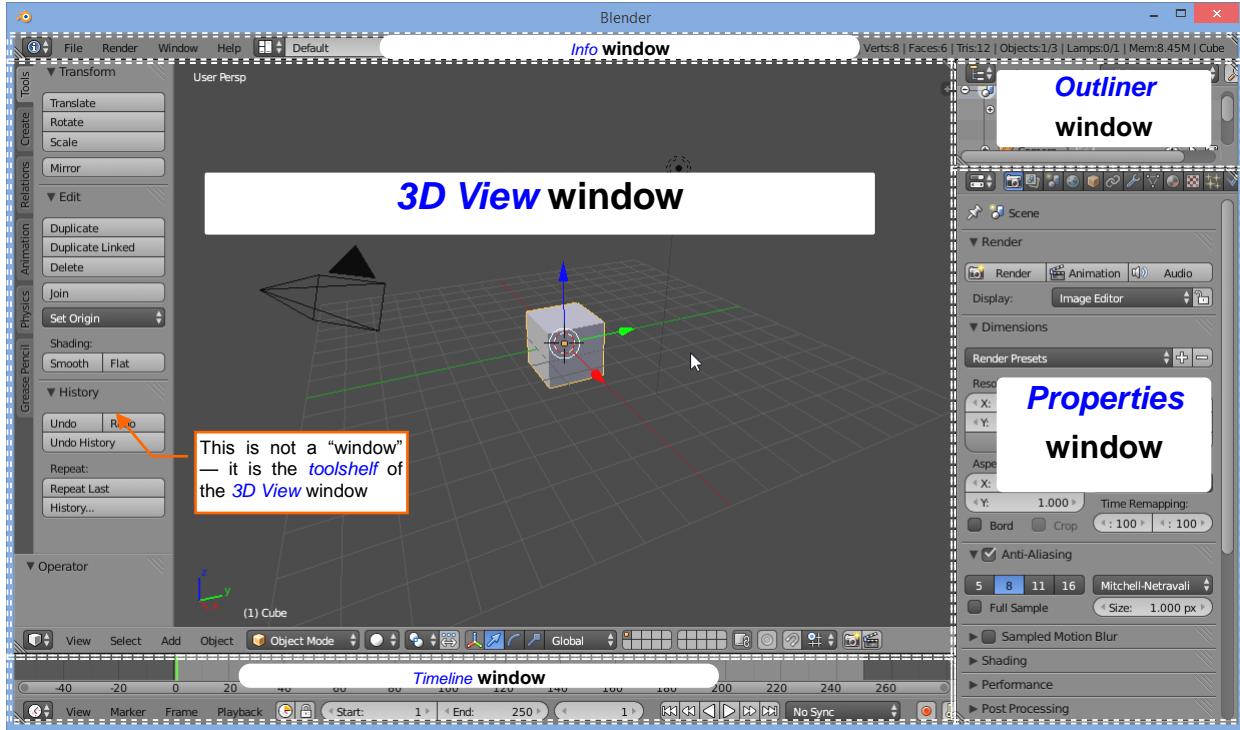


Figure 2.1.2 Blender – the areas of the default screen

- **3D View** window: shows a three-dimensional “scene” that contains the model (here: a cube);
- **Outliner** window: the structure of the scene, presented in a symbolic way;
- **Properties** window: the area for most of the buttons, numeric fields and switches, which give you control over the scene objects (their properties);
- **Info** window: contains the main pull-down menu and the space for the program messages;
- **Timeline** window: used in animations;

To resize any of these windows, you can drag its border with the mouse (Figure 2.1.3):

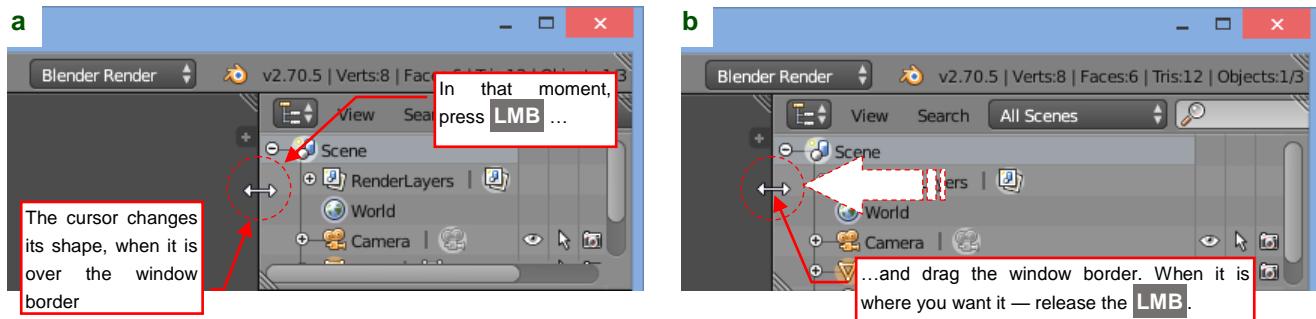


Figure 2.1.3 Resizing Blender windows

¹ I know that it only resembles the basic classic windowing environment a bit. However, this name prevails in the documentation on wiki.blender.org and various other sites. With the introduction of Blender 2.5 its authors attempted to use the term *region*, or *area*, but somehow these names have not been widely accepted. The content of a single Blender window is also sometimes referred as an *editor*.

Let's remove from the [Timeline](#) window, because we do not need it at this moment. It means joining it with the adjacent area. To join two windows, place the mouse cursor on the border you want to remove, and click the **RMB**. It opens the [Area Options](#) menu. Select from it the **Join Area** command (Figure 2.1.4a):

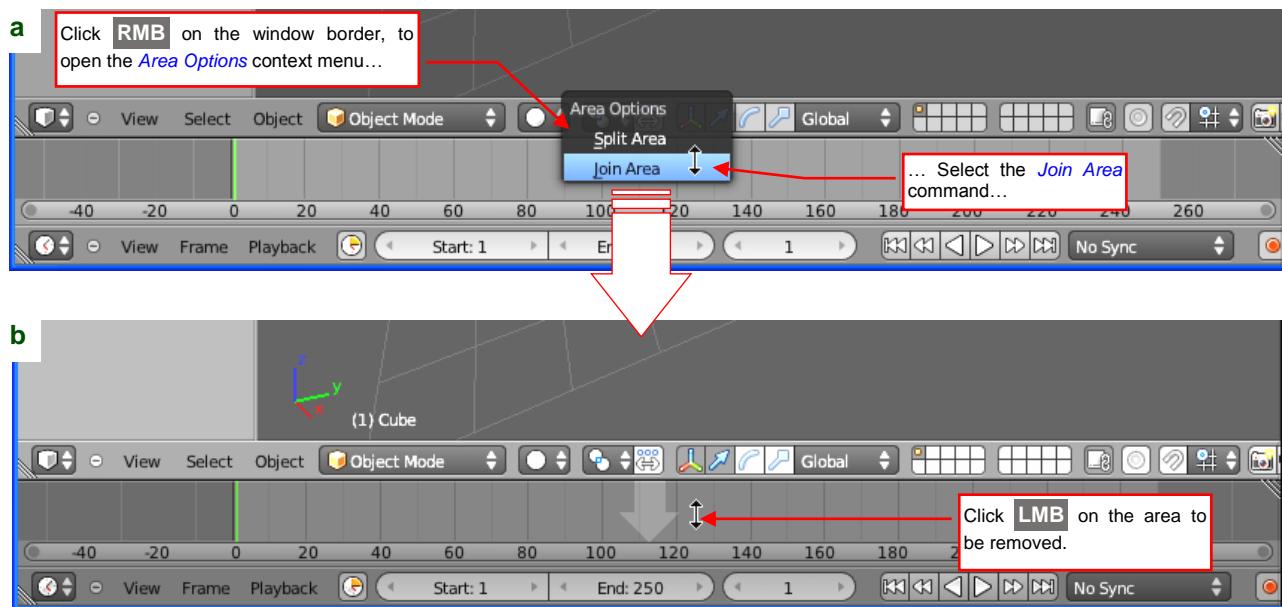


Figure 2.1.4 Joining Blender windows

When you invoke this command, Blender will point to the window to be eliminated (Figure 2.1.4b). It is always the area below the mouse cursor.

- The Blender window which currently is below the mouse cursor is called the **active window**. (It is a little bit lighter, but you cannot see this on the pictures in this book — the difference of the shades is too small).

Click **LMB** to finish this operation — your screen should look like that in Figure 2.1.5:

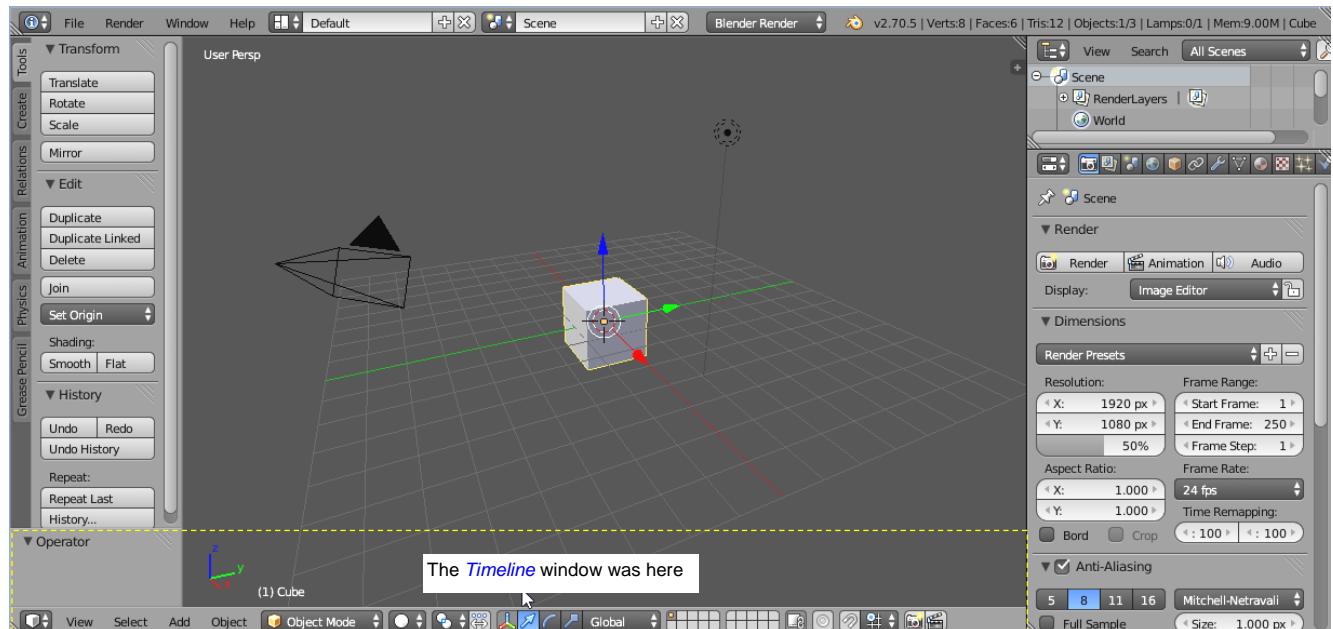


Figure 2.1.5 The screen after joining the [Timeline](#) window with the [3D View](#) window

- Before the final click, you can cancel this operation just pressing the **Esc** key. (This key always cancels the current operation in Blender).

Similarly, to split a window use the **Area Options → Split Area** command. You can also do this in another way — using one of its handles, placed in the upper left and lower right corner of each window. They look like a “ribbed” triangle (Figure 2.1.6):

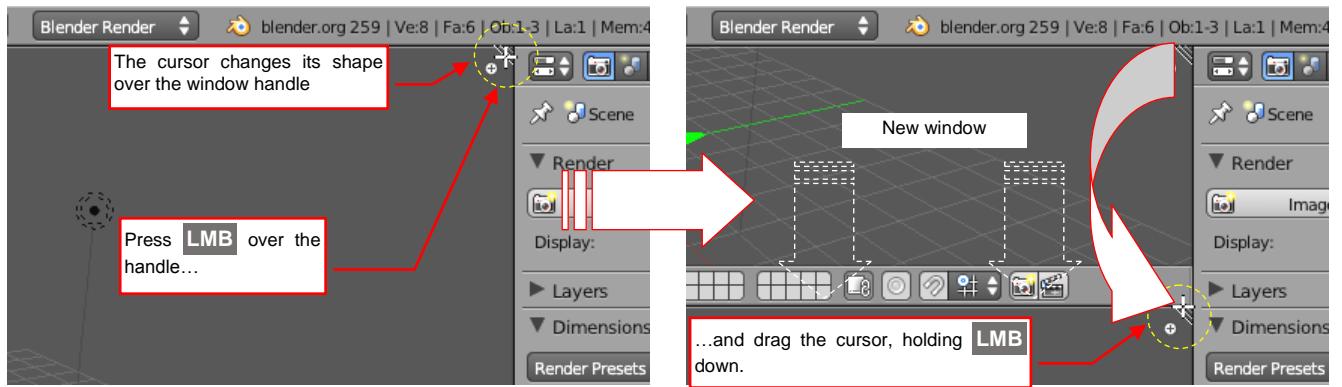


Figure 2.1.6 Using the window handle to split Blender windows

Grab the upper window handle with the mouse (press the **LMB** and hold it down) and move it vertically down the window area. For this movement Blender splits the window horizontally (Figure 2.1.6). You will suddenly find that you are dragging the bottom edge of a new window. If you move the handle horizontally, Blender will split the window vertically.

- The content of the new window is always an exact copy of the split window

You can also use handles to join two Blender windows. Simply drag it into the area that has to be eliminated (Figure 2.1.7):

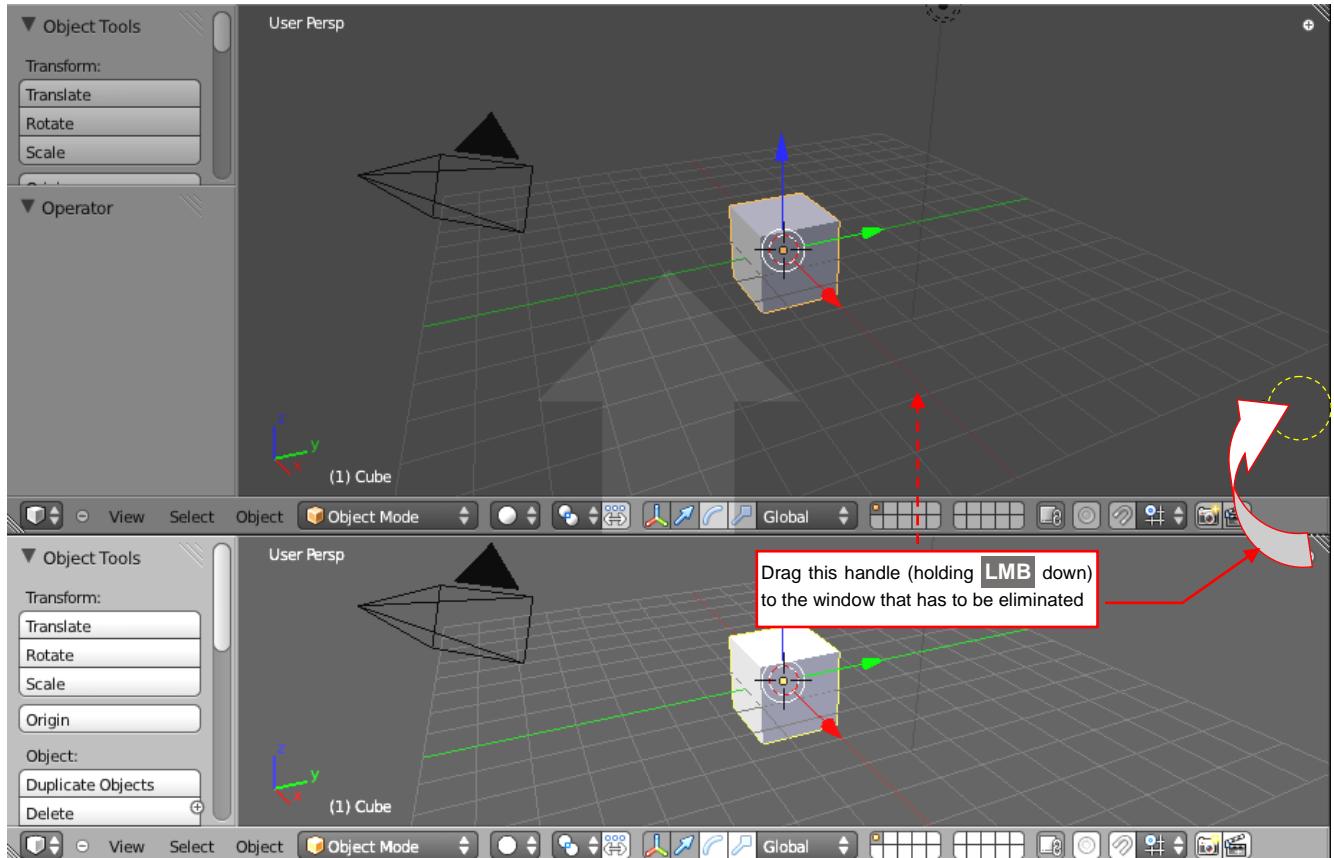


Figure 2.1.7 Joining Blender windows with the handle

Blender dynamically points this active window with an appropriate arrow (Figure 2.1.7). Just release the **LMB**, to join these two areas.

As an exercise, split and/or join Blender windows creating screen layout as in Figure 2.1.8:

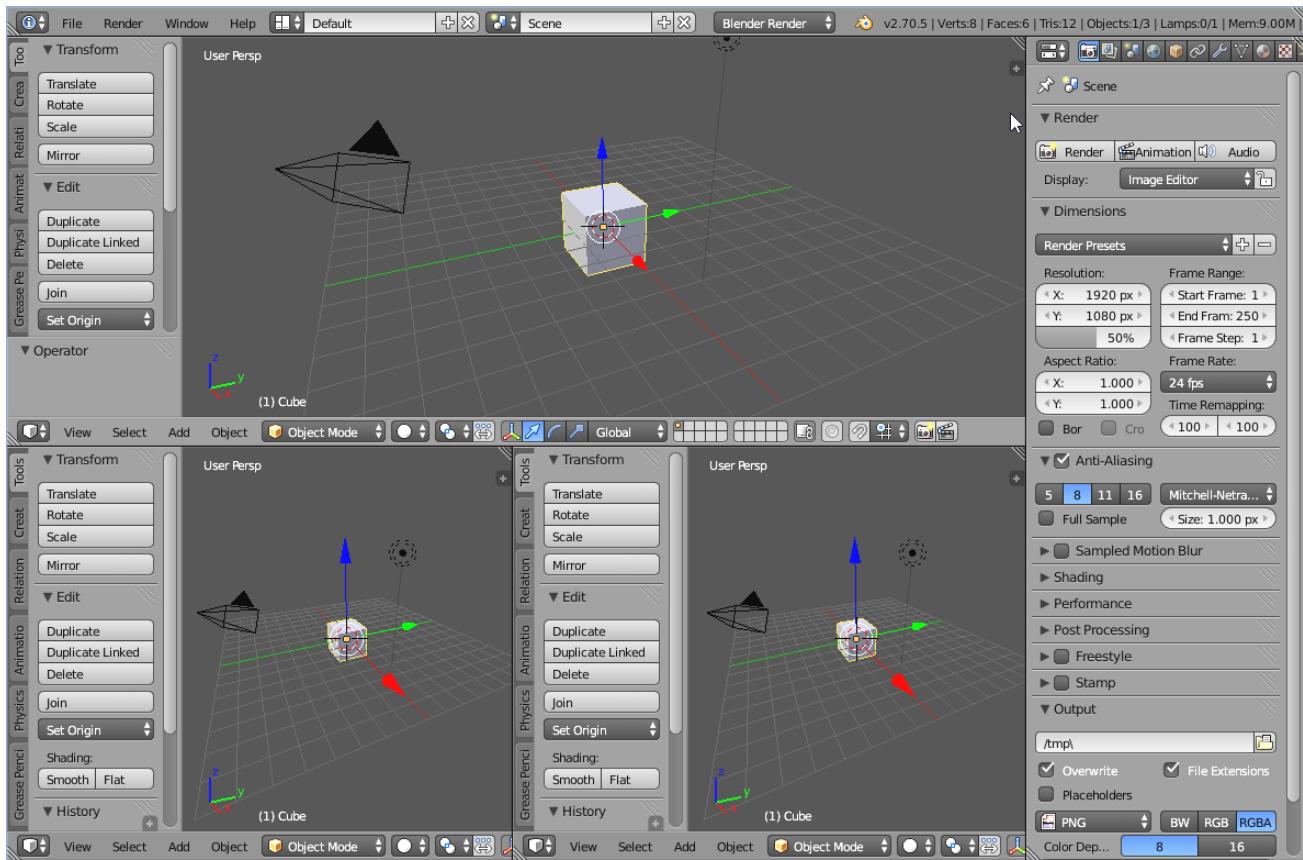


Figure 2.1.8 Alternate screen layout

Let's discuss the basic elements of a Blender window (Figure 2.1.9):

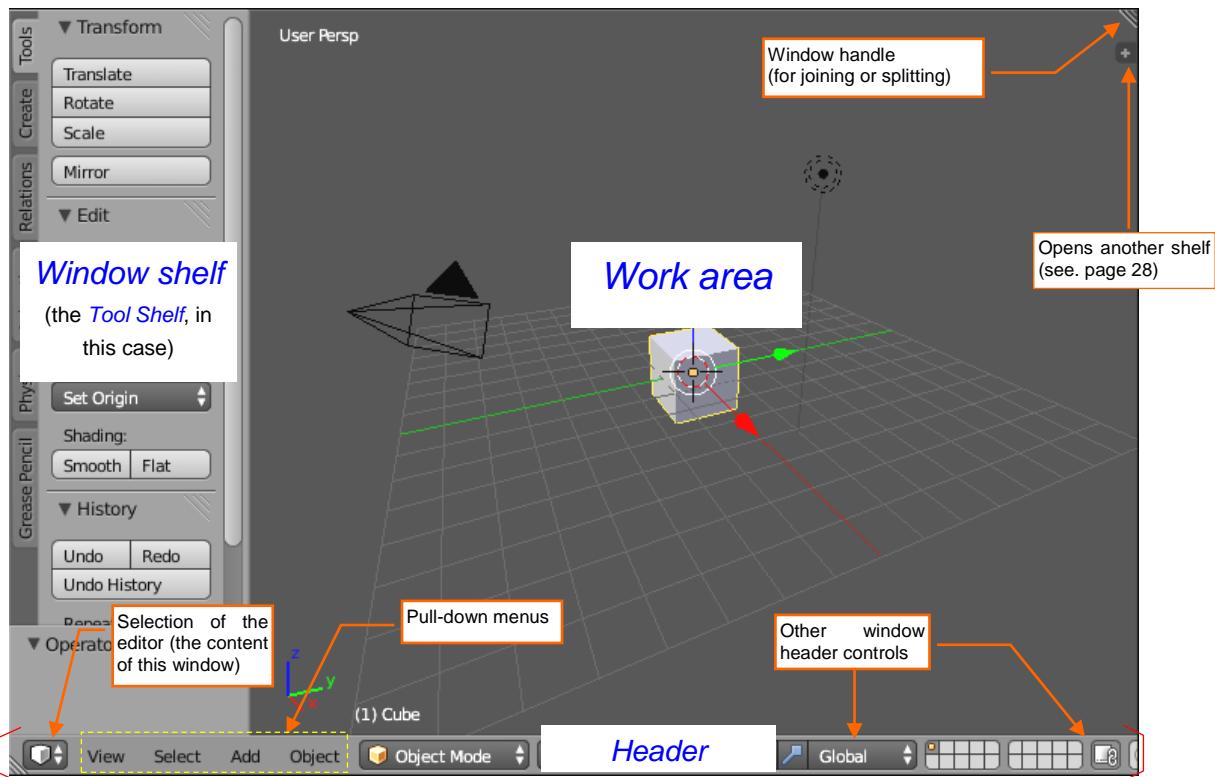


Figure 2.1.9 Elements of the Blender window

Each Blender window has a header. Sometimes this element is placed at its top (as its name suggests), sometimes at the bottom. The header contains pull-down menus and some other controls.

On the left side of the header there is a drop-down list. You can use it to select the type of the editor to be displayed. For example — you can display the scene structure, switching to the *Outliner* editor (Figure 2.1.10):

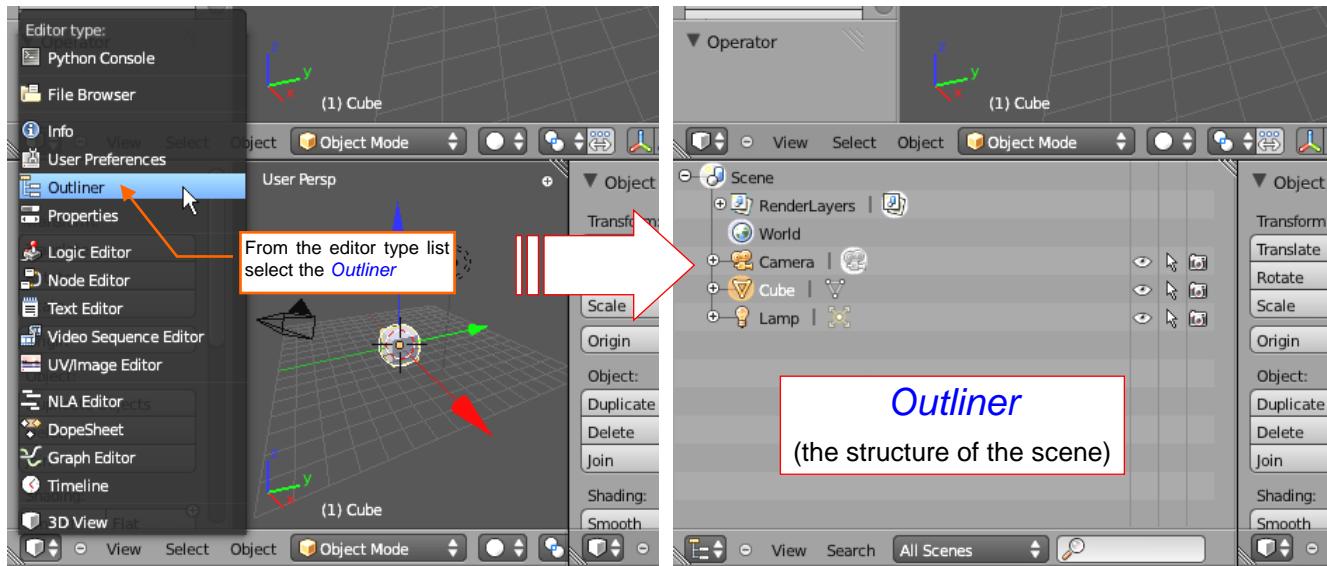


Figure 2.1.10 Changing the window type

Some types of Blender editors (*3D View*, *UV/Image Editor*, *Text Editor*) have a *Toolshelf*, placed on the left side of the window. You can push/pull it using the menu command (*View*→*Tool Shelf*), but it is much easier just to press the **T** key (Figure 2.1.11):

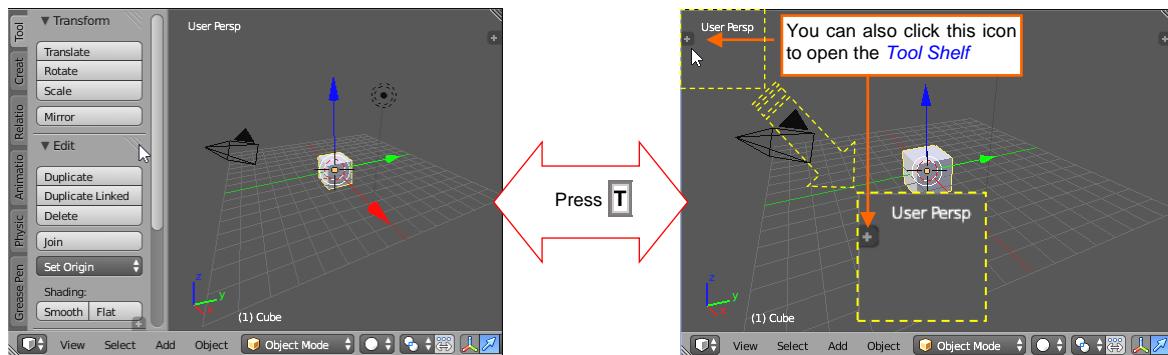


Figure 2.1.11 Flipping the *Tool Shelf* visibility

In a similar way you can show/hide the *Properties* shelf, placed on the right side of the window. Use just the **N** key or the menu command (Figure 2.1.12):

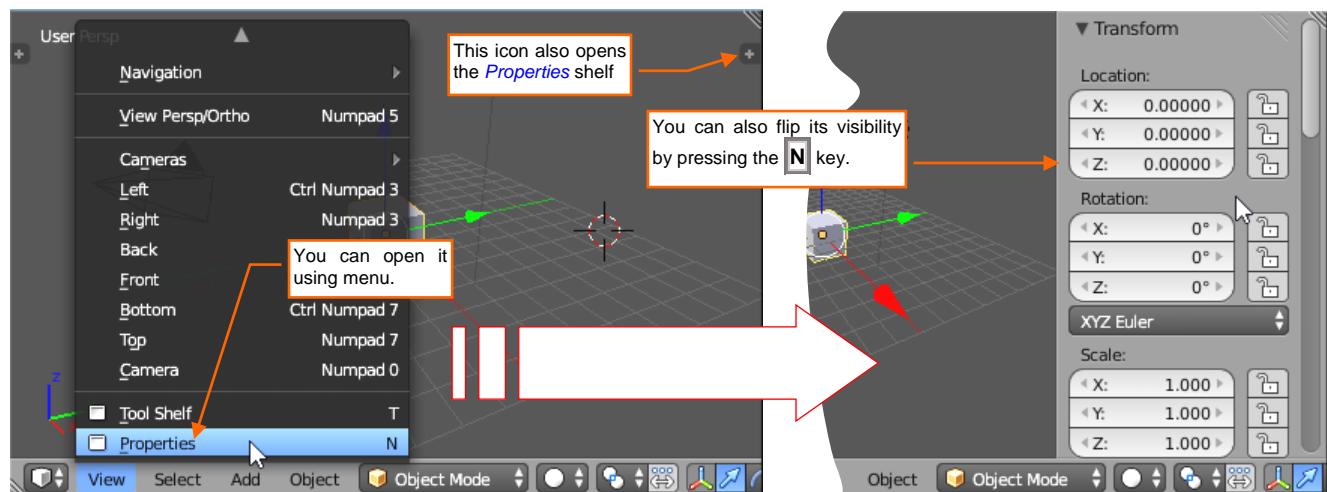


Figure 2.1.12 The *Properties* shelf and various methods of controlling its visibility

When the mouse cursor is over the header, you can click **RMB** to open its context menu. Depending on what has been clicked, Blender opens a longer or shorter menus (Figure 2.1.13):

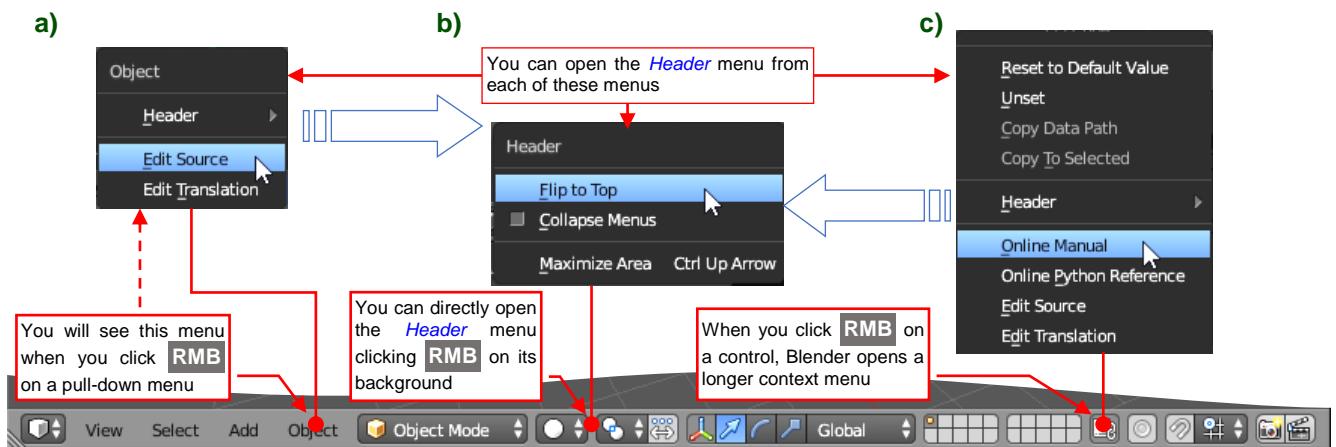


Figure 2.1.13 Context menus of various Blender GUI elements

The control context menu (Figure 2.1.13c) contains some commands that are not available in other contexts. The **Reset to Default Value** command sometimes can be useful when you are not sure what the default value is for a particular control. The **Online Manual** command opens the relevant part of the Blender guide (an article about clicked control from wiki.blender.org). The **Edit Source** and **Edit Translation** commands are useful only for Blender developers or translators. A shorter version of this menu (Figure 2.1.13a) will appear when you click **RMB** on one of the pull-down menu labels. In both of these menus you will find the **Header** submenu that controls the window header (see Figure 2.1.13a and Figure 2.1.13c). Blender can directly open this **Header** menu when you click **RMB** somewhere in between the window header controls.

Contents of the **Header** menu is useful for screen layout modifications. When you invoke the **Header**→**Flip to Top** command, the header will be placed on the top of its window (Figure 2.1.14):

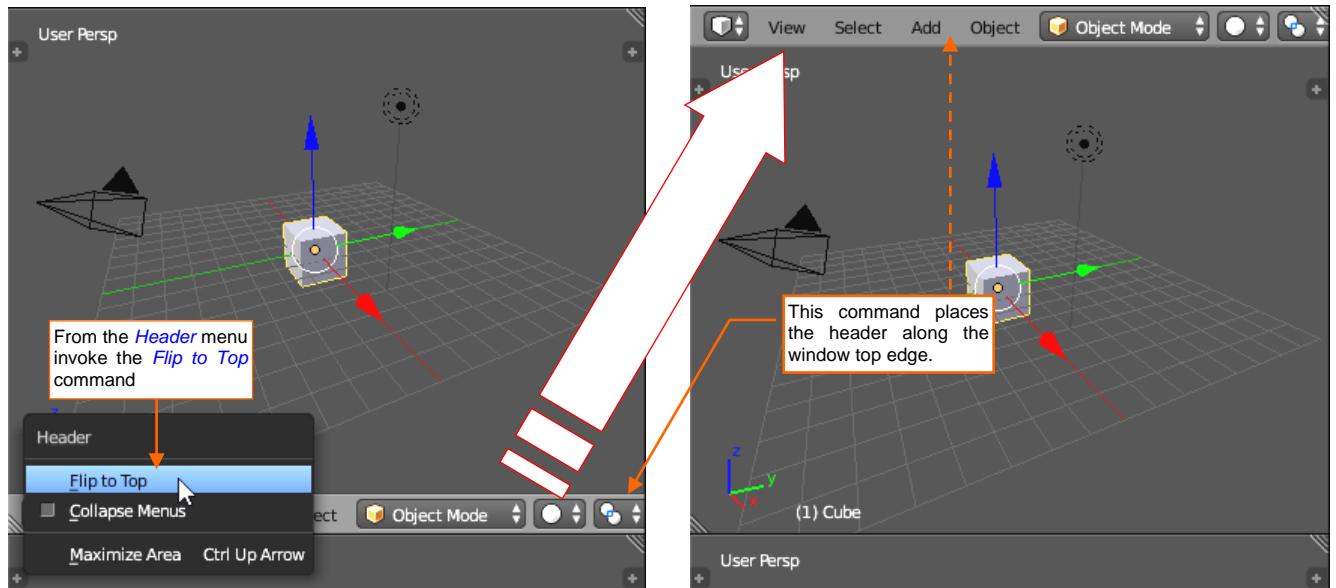


Figure 2.1.14 Flipping the header on the top of its window

When the header is on the top of the window, Blender modifies its **Header** menu. Now you will find there a command named **Flip to Bottom**, instead of the **Flip to Top** item. You can use it to bring the header back to the bottom.

Starting from Blender 2.70 there is a new option in the *Header* menu, named ***Collapse Menus***. When you turn it on, the pull down menus from the header “shrink” into a single item (Figure 2.1.15):

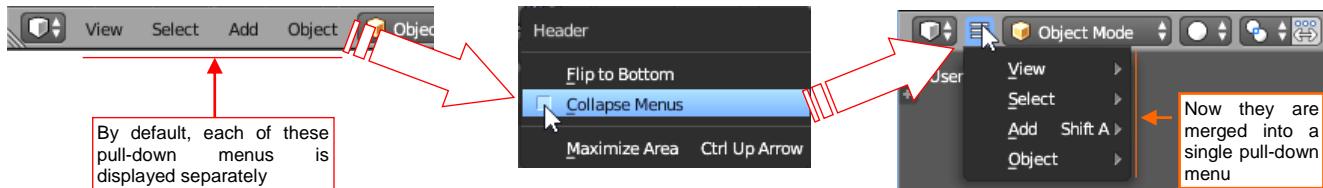


Figure 2.1.15 Shrinking window header menus

Use another command from the *Header* menu — ***Maximize Area*** — to temporary extend active window on the whole Blender screen. However, you can do the same using the keyboard shortcut: ***Ctrl*-*↑*** (Figure 2.1.16):

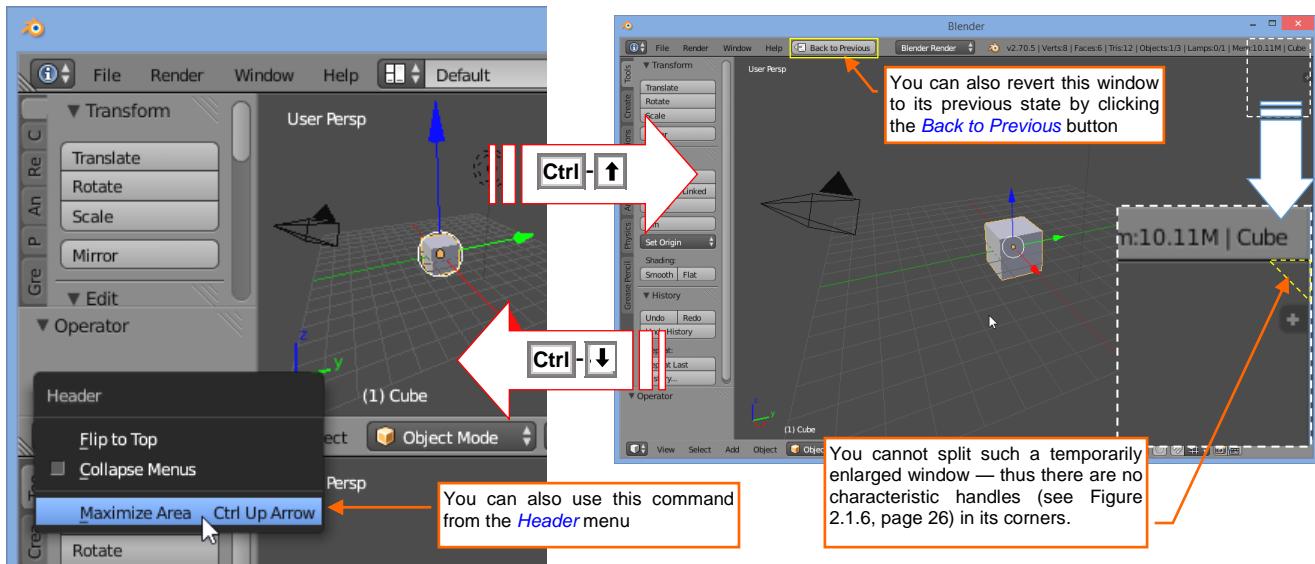


Figure 2.1.16 Maximization of the window

The reverse effect — reduction of the window to its original place on the screen — you can obtain by pressing the ***Ctrl*-*↓*** keys (or invoking the *Header*→*Tile Area* command — but it requires too many mouse “clicks”...).

So far we have learned three of the Blender keyboard shortcuts: **T**, **N**, and ***Ctrl*-*↓*** / ***↑***. For each of them I also specified the corresponding menu command. Which method of command invocation is better: the menu or the keyboard shortcut? Shortcuts are obviously faster. However it is not worth learning all the program shortcuts, because our heads have finite capacity. (Many Blender guides focus on the keyboard shortcuts as the main way of invoking its commands. I think that it also contributes to a false impression that this program is difficult to learn). It is easier to invoke the less frequently used commands from the menu, because you have only to remember “more or less” the place where you can find them. When you notice that there is an often used command — look to the right side of its menu name. You will find there its keyboard shortcut. (If it does not exist, you can configure your own — see page 263). Describing this program, I will always tell you where to find a particular command in the menu. I will also specify its keyboard shortcut, if it exists¹.

¹ I strongly recommend you to use keyboard shortcuts. I remember that in one manual for designing user interfaces, its author complained that today's solutions are designed for “Napoleons”. What does it mean? Do you remember the favorite attitude of this man, repeated on many sketches and paintings? He is always doing something with one hand — pointing, waving, and signing. Simultaneously his second hand is idle, stuck into his coat. Too often the software interface is designed as if you had only one hand — the one which holds the mouse. The second hand lies idle somewhere on the side. Programmers often forget that we have two hands. If we were able to use both hands efficiently, we would work faster. For example — do you know how efficient in text editing a man would be, who types with one hand on the keyboard and in the same time uses the second one to select the text with the mouse? Keyboard shortcuts allow Blender to achieve similar effect in modeling. They allow to “play on both hands” with the computer. You should practice the shortcuts used in the mesh edition (it is the most time consuming operation).

You do not need to laboriously rearrange the borders of the windows every time. Blender offers you some ready to use screen layouts. They are available in the **Screen lay-out** dropdown list, placed on the **Info** window header (Figure 2.1.17):

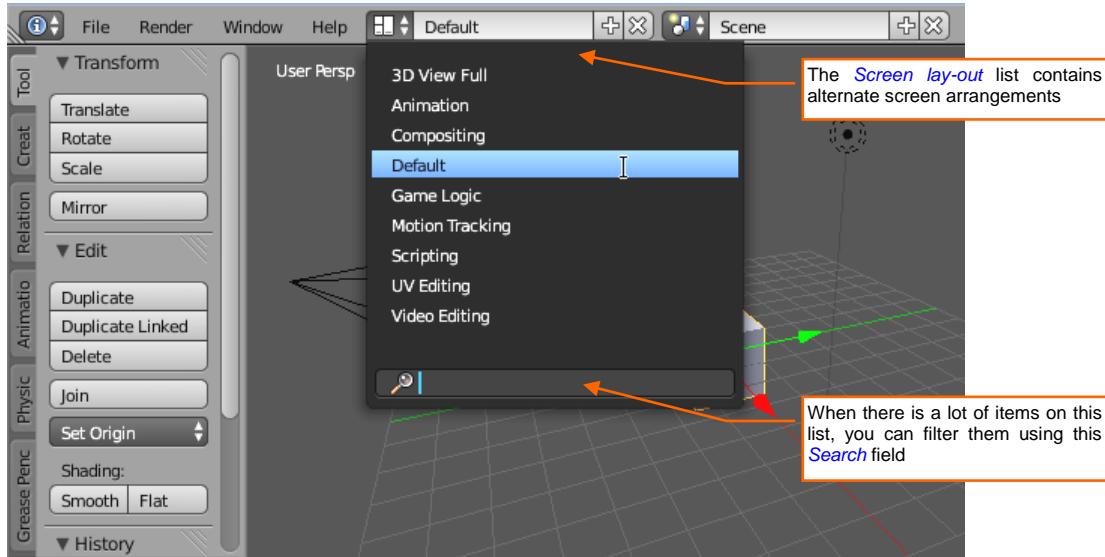


Figure 2.1.17 Selection among one of stored screen layouts

The screen layout we have modified so far is named **Default**. You will use it most of the time, during modeling. The UV **Editing** and **Animation** layouts will be used later. Try to select another screen layout (Figure 2.1.18):

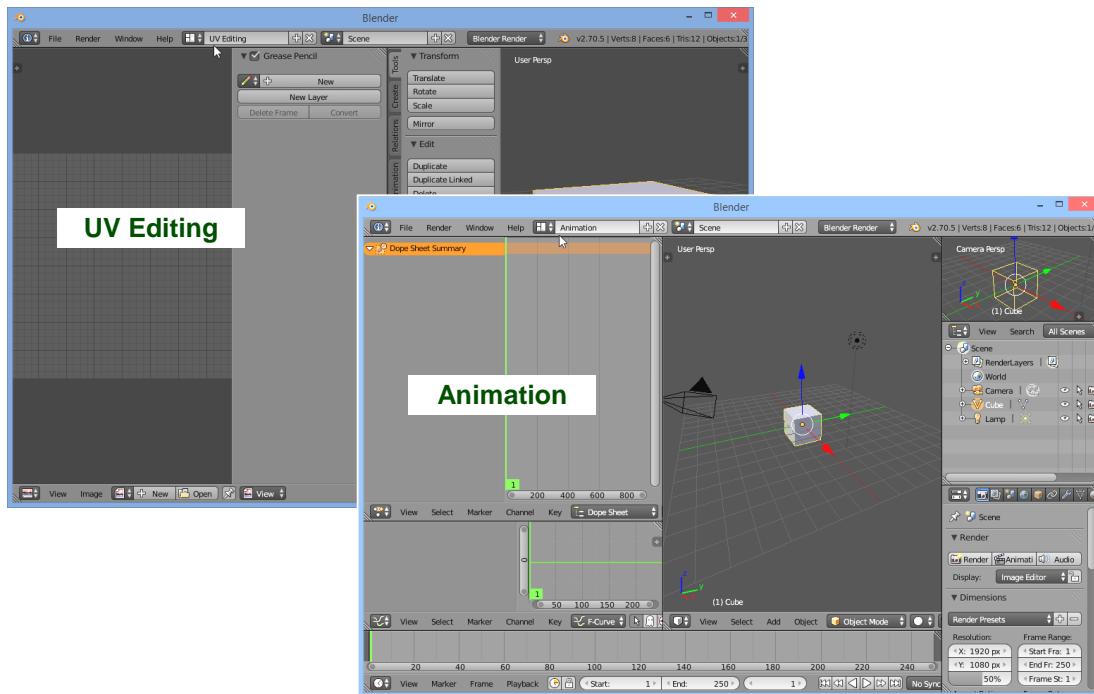


Figure 2.1.18 Other, predefined screen layouts: **UV Editing** and **Animation**

Of course, you should treat these layouts only as suggestions. You can completely rearrange them. The **Screen lay-out** control allows you also to define your own layout. To create it, use the **+** button, located on the right side of the list. The next button — **X** — deletes (removes from the Blender file) the current screen layout. You will find more about this control and the management of the screen layouts on page 283.

You can change the Blender configuration in any window, switching its content to the **User Preferences** editor. This editor is available on the **Editor type** list (just above the **Outliner** item — see also page 28, Figure 2.1.10). Nevertheless, I prefer to invoke this window in another way: from the **File** menu (Figure 2.1.19):

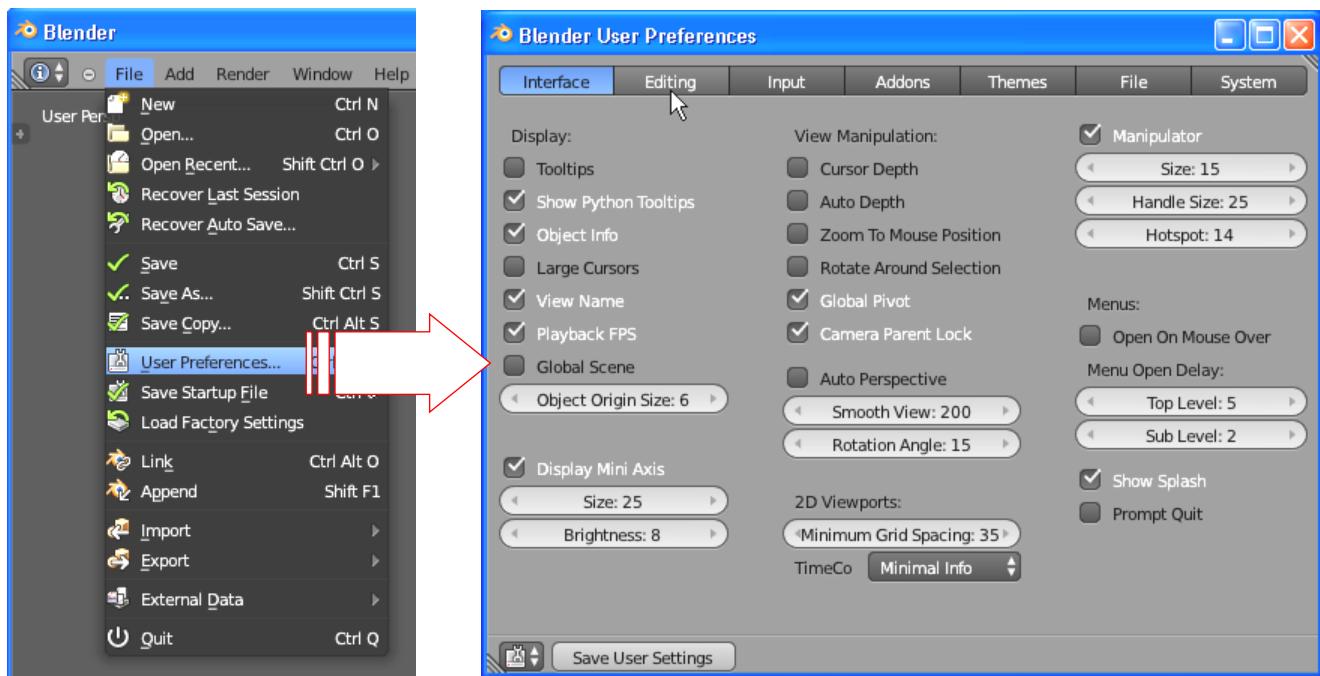


Figure 2.1.19 Accessing the program configuration

On the top of **User Preferences** window you can select one of its sections (**Interface**, **Editing**, **Input**...) (Figure 2.1.20):



Figure 2.1.20 The **User Preferences** window

A detailed discussion of what should be altered in the default Blender configuration can be found on page 261. Here I would like to indicate the option that we will need in the next section. If you use a laptop or any other non-standard keyboard without the numeric pad — in the **Input** section turn on the **Emulate Numpad** option, as in Figure 2.1.20. Then save this change by pressing the **Save User Settings** button. (You will find it at the bottom, on the window header).

Starting from the release 2.66¹, Blender stores its configuration in a special file named *userpref.blend*. This file is located in a separate folder of the Blender configuration data, in the user home directory.

- You can learn more about the settings that are stored in each drawing, as well as about the ones which are stored only in the *userpref* file (and the location of this file) on page 261.

Another interesting feature of Blender are short tips that appear on every element of the screen. They are controlled by the **Tooltips** option in the **User Preferences:Interface** section (Figure 2.1.21a). Just hold the mouse cursor for a while over an unknown checkbox or button—and you will see its hint (Figure 2.1.21b):

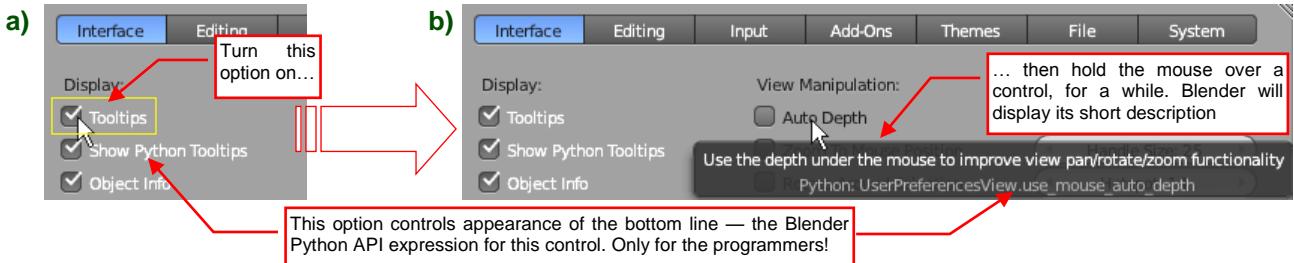


Figure 2.1.21 Tooltips for the screen controls

The bottom line of the hint is displayed in a different color. It is the name of this property in the Blender Python API — an interesting thing only for the programmers! If you don't need this information, just turn off the **Show Python Tooltips** option (Figure 2.1.21a).

By the way — have you noticed, that currently you have two Blender screens? (Figure 2.1.22):

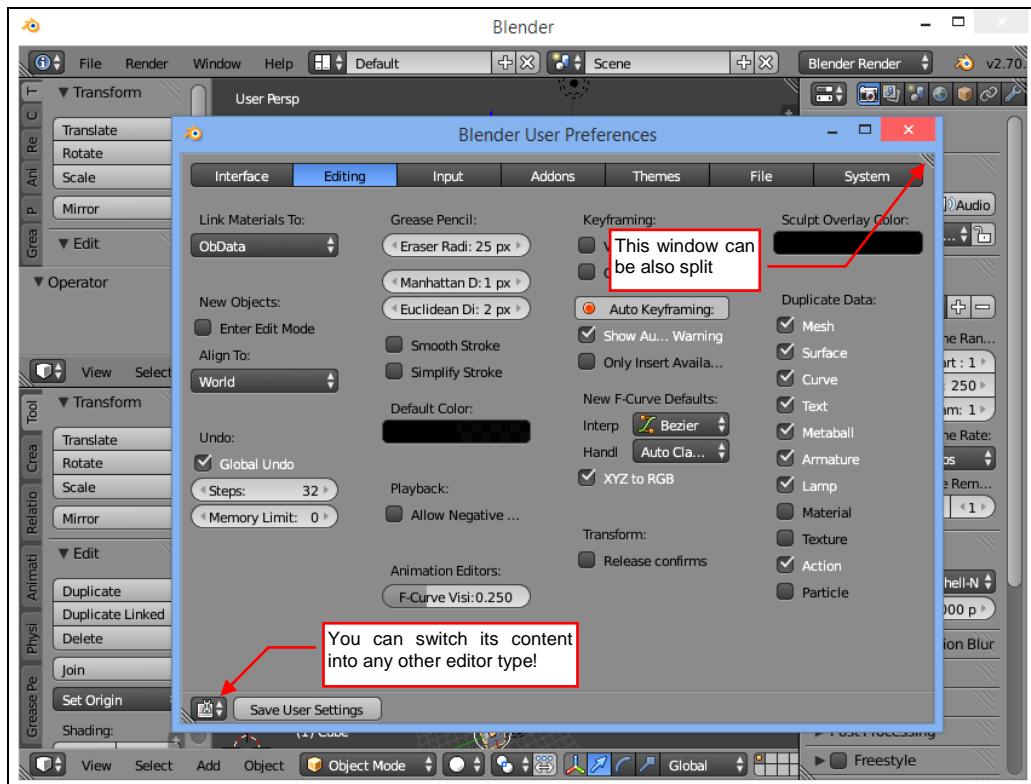


Figure 2.1.22 Two Blender screens

¹ All previous Blender versions stored the user settings in so-called startup file (*startup.blend*). This file contains also the default scene which is displayed when you open the program (you can see its content in Figure 2.1.1). In these old versions the user had to store the altered settings together with the current scene. In practice we always had to save the Blender settings for an empty scene!

The second screen (created by the [File → User Preferences](#) command) has the same properties as the first one — you can switch its contents into another editor type (such as the [3D View](#)). It can also be split into several areas. If your computer has two displays, operating in so called “extended desktop” mode¹ — you can place a Blender screen on each of them. Both screens will keep showing the same scene (Figure 2.1.23):

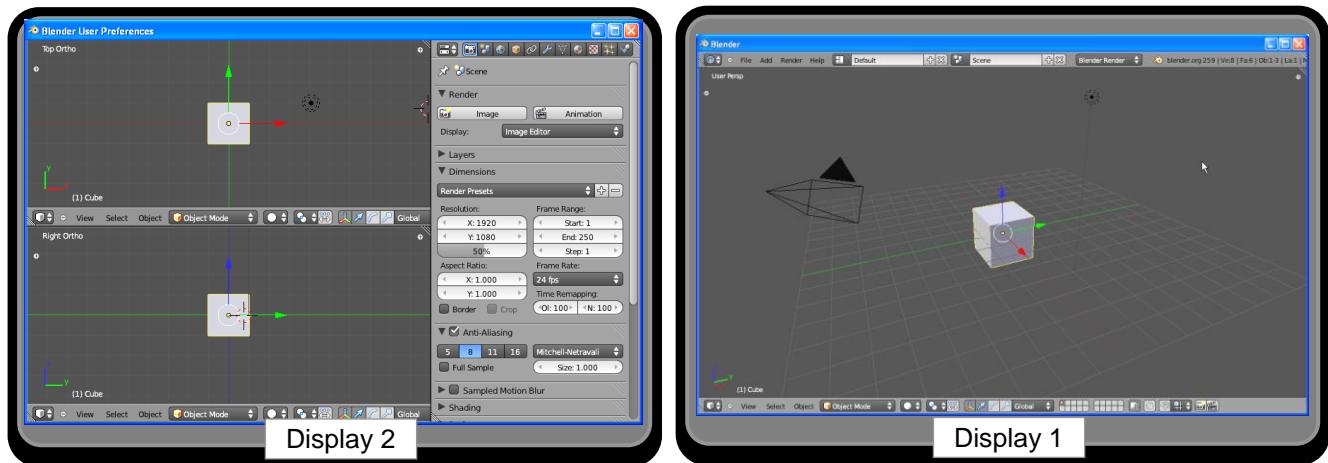


Figure 2.1.23 An example of Blender screen layout in the “extended desktop” mode

In practice, this mode doubles the available screen space, significantly improving ease of the work.

- There is another method of opening second Blender screen: in a [3D View](#) editor open its [View](#) menu and invoke the [Duplicate Area into New Window](#) command.

At the end of this introduction to Blender, let's mention how to close this program. Of course, you can shut it down like any Windows application, by clicking on the button in the upper right corner of its screen (window). There is also another way to do it: use the [File → Quit](#) command (Figure 2.1.24), or press the [Ctrl-Q](#) shortcut.

- After the [Quit](#) command Blender **does not warn** that the current file contains unsaved changes!

Perhaps such a behavior of the [Quit](#) command comes from the fact that it saves the current state to a special file that you can restore using the [File → Recover Last Session](#). When you turn on the [Interface:Prompt Quit](#) configuration option (see page 262) Blender will display a warning message when you try to close this window with button.

Note that in the [File](#) menu there is also the command that stores current scene as the initial content: [Save Startup File](#). Finally, when you have altered so many settings that you do not know which one is wrong, you have the [Load Factory Settings](#) command here. It reverts Blender to its initial state.

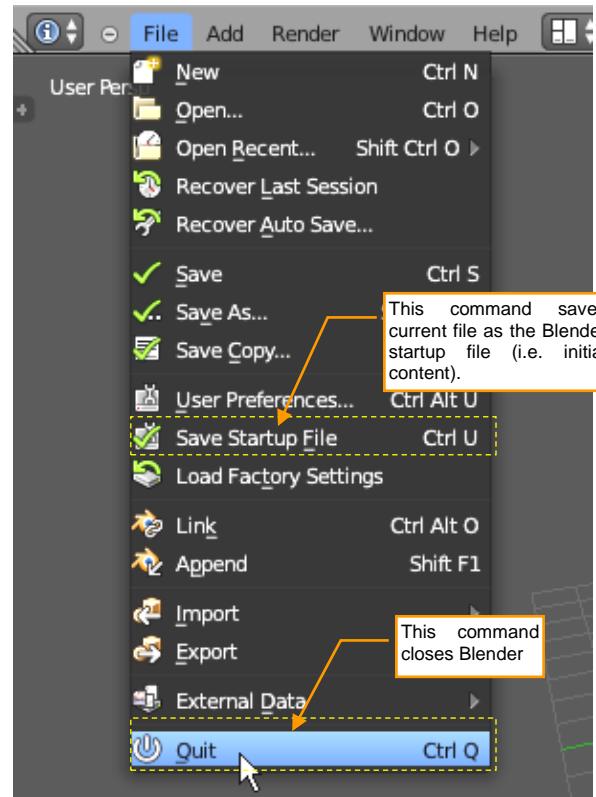


Figure 2.1.24 Commands from the [File](#) menu.

¹ Most of the laptops have this mode — because you can connect them to an external display.

Summary

- The Blender user interface is different from other Windows applications. However, “different” does not mean “worse” — you will see this in the book;
- Blender screen is divided into some areas, also referred as **windows**. You can move their borders (see page 24), but they never overlap;
- You can split or join Blender windows (as a result of the latter operation one of them is eliminated). You can invoke these commands from the *Area Options* context menu (page 25). The same effect you can get by moving their handles, placed in the lower left and upper right corner of each area (page 26);
- Every type of Blender window has a header and working area, and some of them — also have “shelves” (*Tool Shelf*, *Property Shelf*), located on their sides (page 27);
- Window shelves can be “retracted” and “ejected” from the window borders by pressing the **T** key (*Tool Shelf*) and the **N** key (*Property Shelf*). If you do not remember these shortcuts, use the commands from the *View* menu (page 28);
- To change the type of the current editor (the window content), use the dropdown list, placed on the left side of the window header (page 28);
- Blender has some ready-made screen layouts, intended for different stages of the work: modeling, texturing, and animation (page 31). Every of them can be customized. You can also create your own screen layout;
- The screen area below the mouse cursor is often referred to in Blender as the **active window** (see page 25).
- The active window can be maximized with the **Ctrl**-**↑** shortcut. Pressing these keys again restores the original screen layout (page 30);
- You can open many Blender screens (for the operating system they are just separate windows). When you have two monitors, you can use both of them in this way (page 34);
- The best way to open the configuration window is to use the **File**→**User Preferences** command (page 32);
- To save altered configuration settings click the **User Preferences:Save User Settings** (page 33). Blender stores it in a file named *userpref.blend*, in your user profile. (More on this subject — see page 261);
- When you invoke the **File**→**Quit** command, Blender will not display any warning that the current file contains unsaved changes. Instead of it, it saves the current scene state to a temporary file named *quit.blend*. You can open it using the **File**→**Recover Last Session** command;

2.2 3D View — changing current projection

How long can we just play with the default cube on the Blender scene? It is a little bit boring, isn't it? You have seen it enough in the illustrations in the previous section. Let's load a more interesting model!

First open the file named `source\config\startup.blend` (use **File → Open** — see page 256). You can find it in the `source.zip` file (see page 18). This Blender startup file contains the settings which I have prepared for this book. To save them, click the **Save User Settings** button in the *User Preferences* window, and invoke the **File → Save Startup File** command (details — page 261). Then open the `model\la5\la5.blend` file. (You can find it inside the `la5.zip` package that accompanies this book). Figure 2.2.1 shows how it should appear. We will use this La-5F in subsequent sections of this chapter, to let you work on a real model.

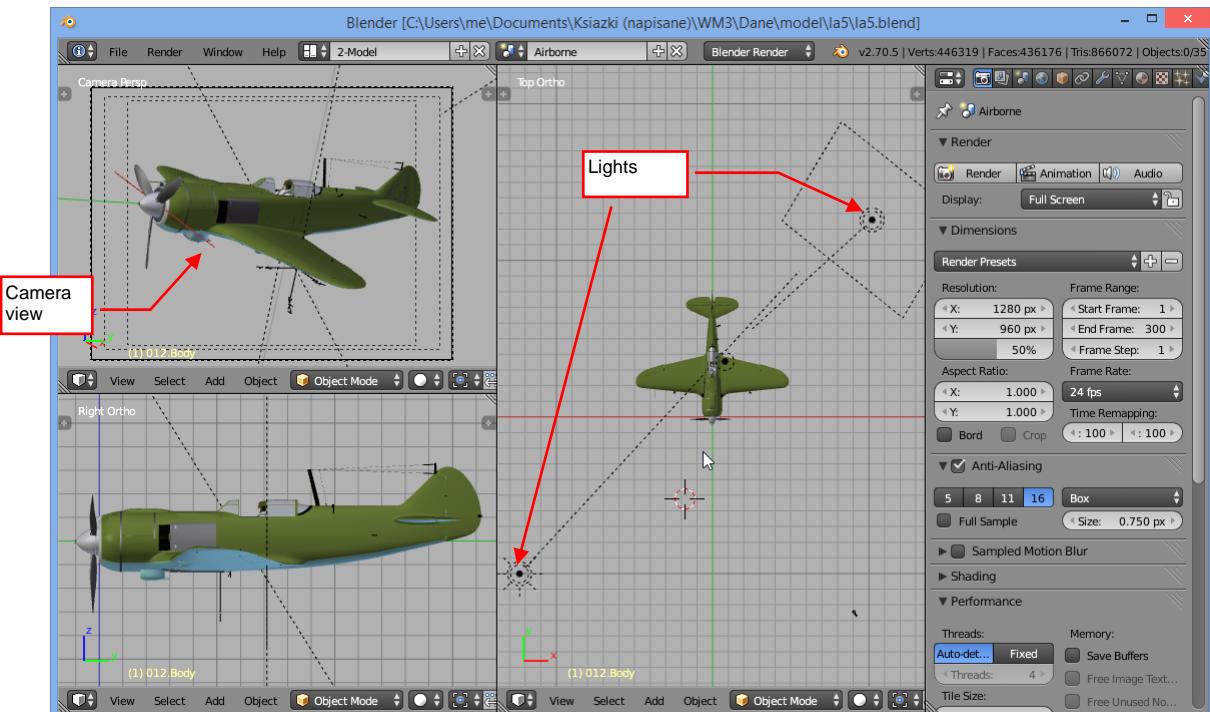


Figure 2.2.1 The La-5F model (content of the file that accompanies this book)

So we have loaded the model, now it is the time to explain how to explore it. Let's maximize its left side view (Figure 2.2.2):

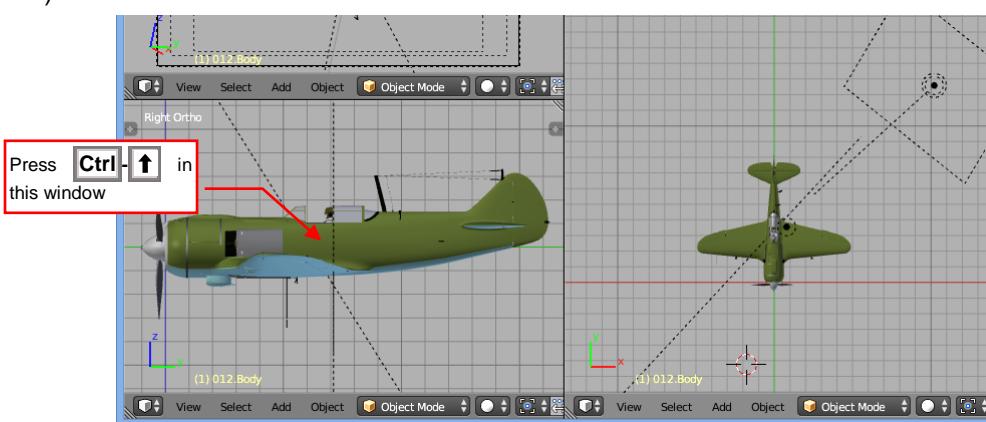


Figure 2.2.2 Maximizing the left side view

Do you remember it from the previous section? Place the mouse cursor over the **3D View** window that contains the left side view (Figure 2.2.2), then press the **Ctrl + ↑** shortcut. It will maximize this window into the whole Blender screen. (I would like to remind you of this command, because it will be often used. You are usually short of screen space, during modeling!).

Now you can see this model more accurately (Figure 2.2.3):

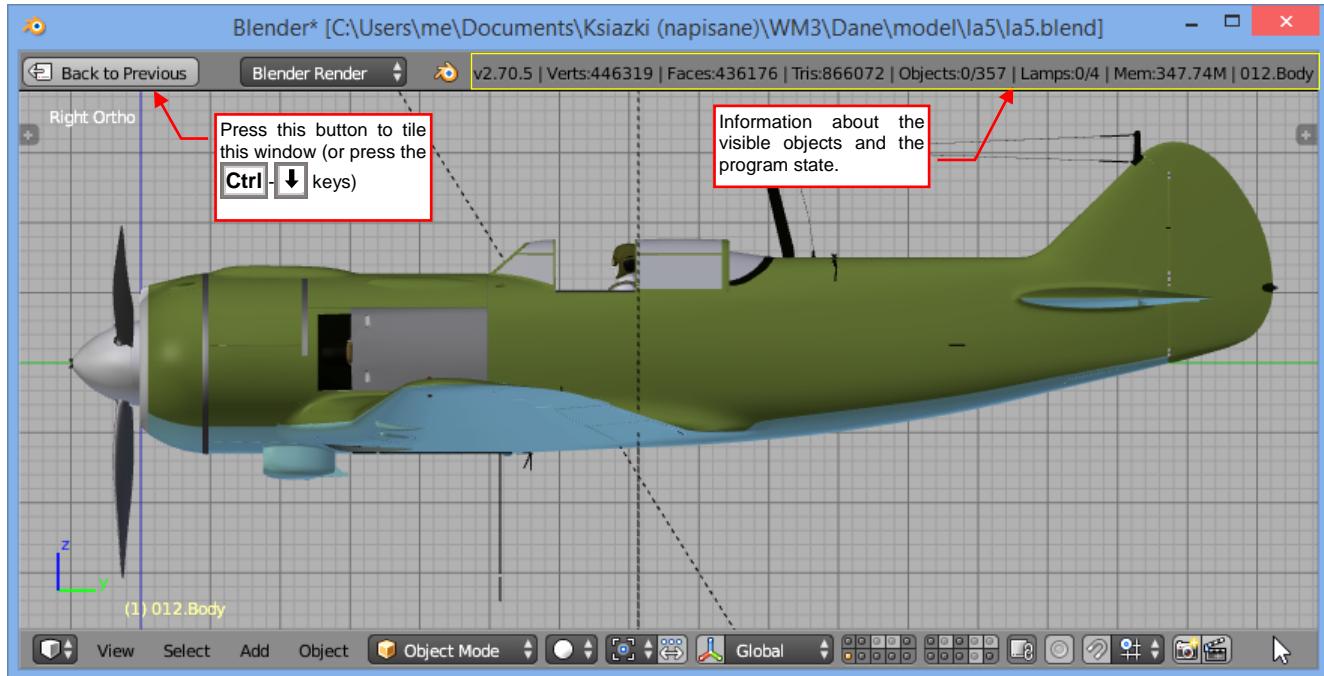


Figure 2.2.3 Maximized window

Pressing **Ctrl**-**↓** (or clicking the **Back to Previous** button on the **Info** window header) will restore this window into its original size and position. Yet let us stay for a while in this enlarged view.

The loaded model contains a lot of vertices and faces. You can read it from the state information, displayed on the right side of the **Info** window header (Figure 2.2.3). This information is more visible when the window is maximized, because in this mode two dropdowns disappear (the screen layouts list and the scenes list — compare Figure 2.2.3 and Figure 2.2.1). What do these abbreviations and numbers exactly mean? Let me explain:

- **v2.70.5**: this is the Blender version number (in this example: Blender 2.70, 5th release);
- **Verts**: number of vertices, used in the objects that are currently visible in this window. This airplane contains 446 319 vertices. (It is a relatively detailed model);
- **Faces**: number of the faces, used in the objects that are currently visible. This model is built of 436 176 elementary faces;
- **Tris**: number of elementary triangles, used in the visible objects. Usually each quad face is built of two elementary triangles, thus this number is two times greater than the number of faces. Of course, some faces are triangular (1 tris per face), or can have more than 4 edges (in such a case it is built from more than two triangles);
- **Objects**: number of selected objects, and (after the slash) all visible objects. This model is built of 357 objects, but none of them ("0") is selected;
- **Lamps**: number of the light sources selected/currently visible in this scene;
- **Mem**: amount of the RAM, allocated to this program¹. In this case — a little over 350 MB. Certainly, this value should be less than the amount of physical RAM installed in your computer. In practice, the available amount of RAM depends on the Blender variant you are using. For the 32-bit variant, it can vary from 1.3 GB RAM in Windows to 1.7 GB in OSX. On the other hand, the 64-bit variant of Blender can allocate, if necessary, the entire available RAM;
- **012.Body**: active object name (what is an "active object" — see page 48, Figure 2.3.3);

¹ For those, who are not fluent in the computer terminology: the RAM is the "volatile" memory, which is cleared every time you turn off the computer. Do not confuse it with the persistent storage memory of your hard drive!

I am afraid that Blender can redraw this La-5 model rather slowly on many less powerful PCs. To remedy this I suggest to hide some layers, as shown in Figure 2.2.4:

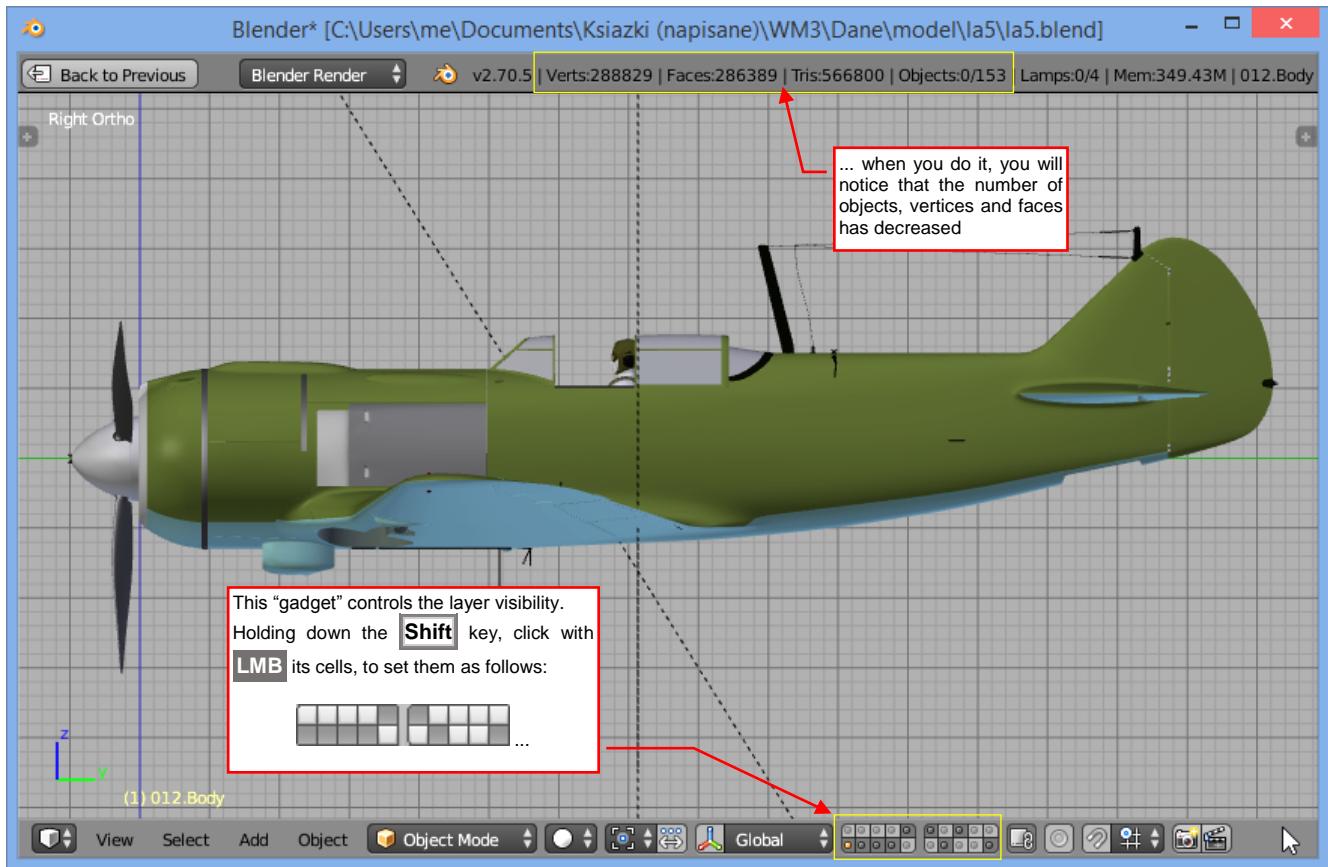


Figure 2.2.4 “Lighter” setup of the La-5 model — more objects are on the invisible layers

Clicking with **LMB** the “cells” of the layer control (see Figure 2.2.4), you can flip their visibility (white = invisible). Always do it holding down the **Shift** key. If you forget it, clicking into one layer will turn off the visibility of all the others¹.

What are layers in a program like Blender? Their name comes from 2D systems. You already encountered them in GIMP (in Volume I, for example). In 3D programs layers lost their analogy to the surfaces that can be stacked on each other². They remained in the role of object groups that you can quickly hide or show. I often use this feature, placing various elements of the aircraft on different layers. Typically, while modeling a new part, I work on one or two layers, which contain all the required objects. For example — modeling the fairing between the wing and fuselage, I used three layers: one for the wings, one for the fuselage, and one for the fairing itself. (That is: the modeled object and the elements it has to fit between).

Blender offers us 20 layers. Treat them as “containers”, in which you can keep the related objects (I will describe on page 242 how to do it)³.

¹ In such a case call the **Undo** command — **Ctrl-Z**, or (in this mode) **Object→Undo**. It restores the layer visibility to the state before this ill-fated click!

² In three dimensional space it is difficult to talk about layers that contain spatial objects, unless you are able to imagine them in the fourth dimension. In this case, I congratulate and envy you ☺!

³ Frankly speaking, on the completion of this La-5 model, the number of 20 layers was a bit too small. I succeeded, however, in the end, to cram all its elements into these 20 “containers” in a more or less orderly manner. A detailed description of the La-5 model structure, including its layers, can be found at http://airplanes3d.net/downloads-la5-intro_e.xml.

It's time to view our model from various sides. Press the **1** key on the numeric pad (or select the **View→Front** command) to see the front view (Figure 2.2.5):

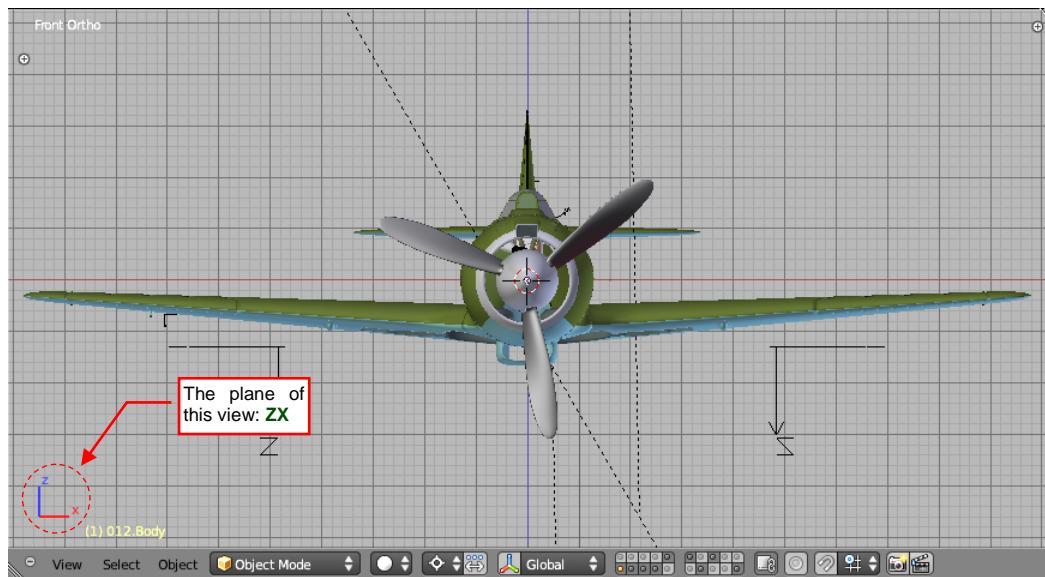


Figure 2.2.5 Front view (ZX) — shortcut: the **1** key

When you press the **3** key on the numpad (or invoke **View→Left** from the menu) — you will see the left side view (**ZY** — Figure 2.2.4). When you press the **7** key (or invoke **View→Top** from the menu) — you will see the top view (**XY** — Figure 2.2.6):

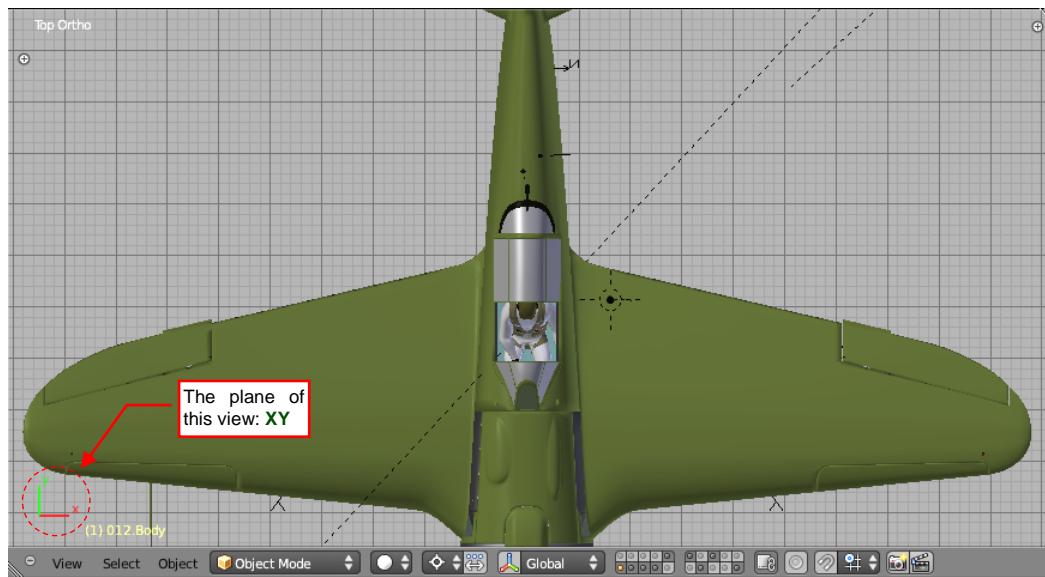


Figure 2.2.6 Top view (XY) — shortcut: the **7** key

To see the model from the opposite side (right, bottom, rear) use the same shortcut keys, but also hold down the **Ctrl** key:

- right side view: **Ctrl**-**3** (**View→Right**);
- bottom view: **Ctrl**-**7** (**View→Bottom**);
- rear view: **Ctrl**-**1** (**View→Rear**);

- On the computers without the numpad, set the *Emulate Numpad* option (see page 32) and use the keys with numbers, from the row located below the function keys (below the **F1**..**F12**).

Figure 2.2.6 does not display the entire airplane. To zoom it out, press the **-** (minus/dash) numpad key, two or three times (or invoke the **View→Navigation→Zoom Out** command, but it is less practical):

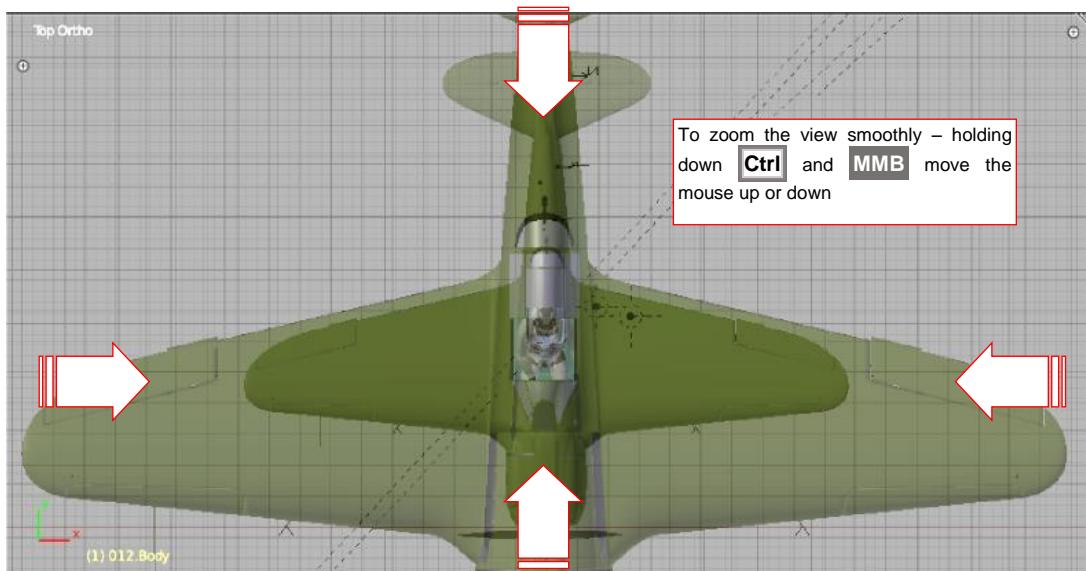


Figure 2.2.7 Zooming

Similarly, pressing the **+** numpad key zooms it in (**View→View Navigation→Zoom In**). There is also a **Home** shortcut. It sets up such a magnification that you can see all the scene objects in the active window.

You can also zoom using the mouse. Roughly — by rotating **MW**. (It invokes exactly the same commands as the **+/ -** shortcuts). Precisely (smoothly) — holding down **Ctrl**, press **MMB** and drag the mouse up or down.

How to pan the view? Roughly — **Ctrl-8** or **Ctrl-2** (up or down), **Ctrl-4** or **Ctrl-6** (left or right). It is not so difficult to remember — notice, that on the top of the **8**, **6**, **2** and **4** numpad keys you can see adequate arrows! To pan precisely (smoothly) — holding down **Shift** press **MMB** and drag the mouse, so the view will follow this movement (Figure 2.2.8). I always use the mouse to pan the view in Blender¹!

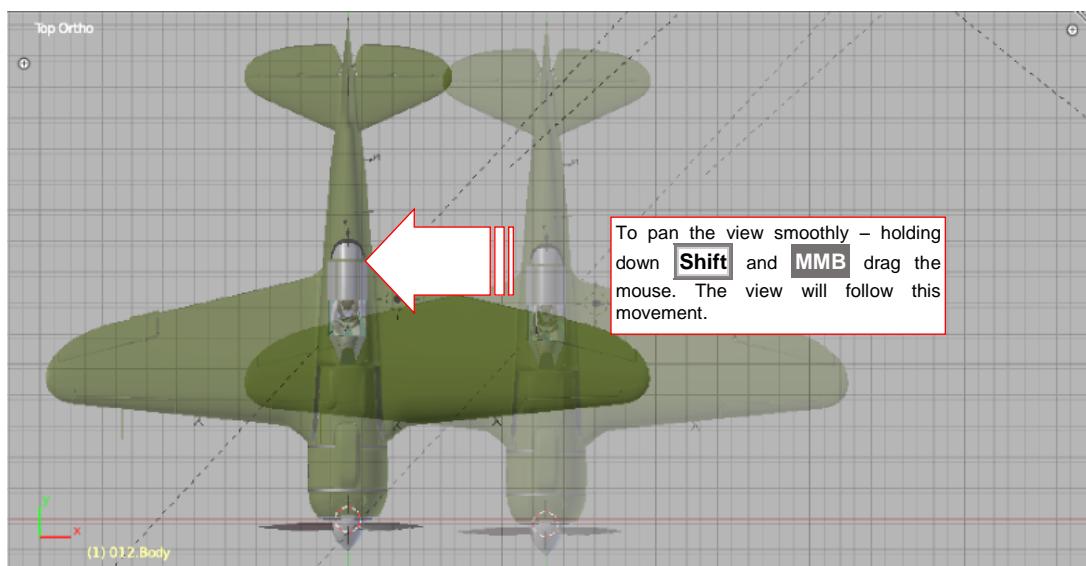


Figure 2.2.8 Panning

¹ Yet another method is a "jump" to specified point on the screen. You can use the **3D Cursor** (see pages 55, 272) for this purpose, invoking the **View→Align View→Center View to Cursor** command (**Alt-Home**)

Finally — let's rotate the view. Press **MMB** and move the mouse cursor within the **3D View** area. This will rotate its viewpoint around the model. In the Figure 2.2.9 you can see the rules of this operation:

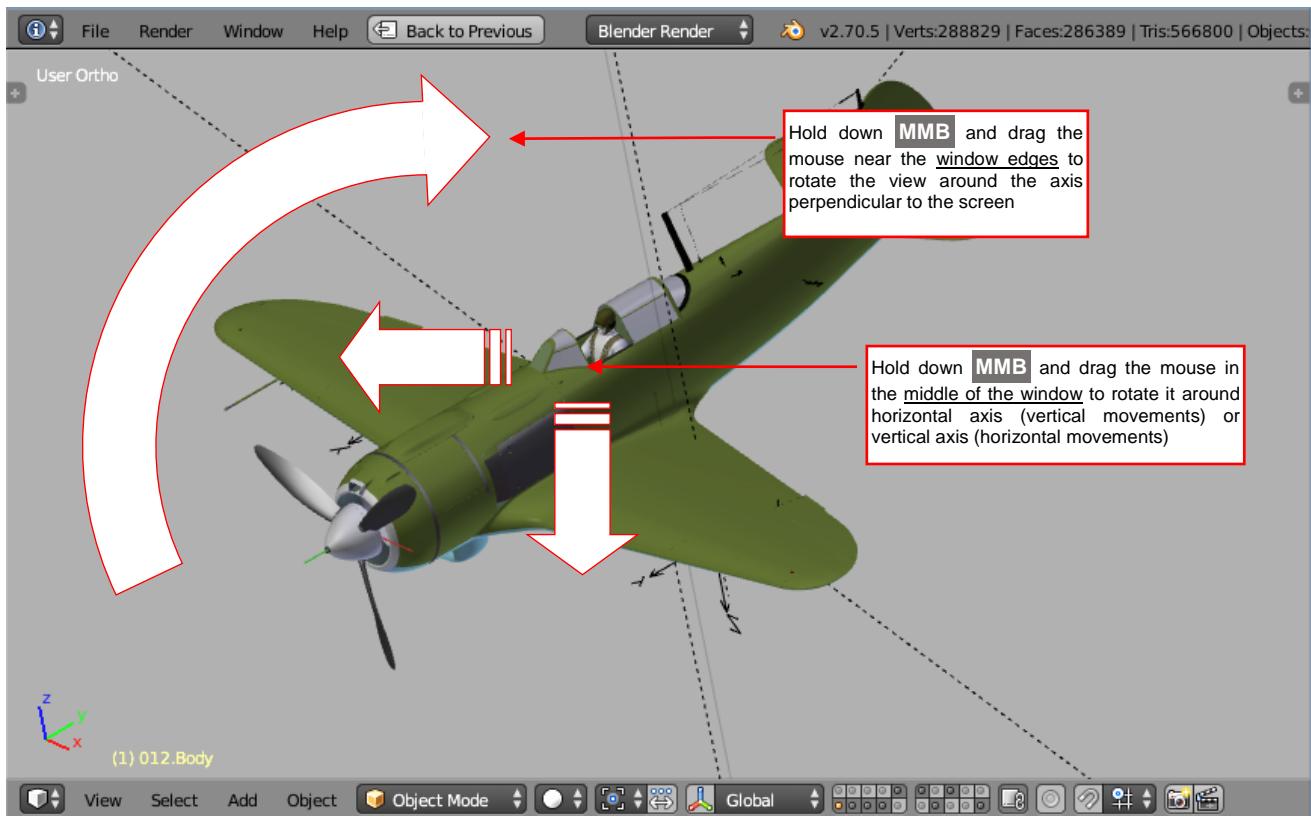


Figure 2.2.9 Rotating the view – the axis of this rotation depends on the mouse location inside the view area

The current rotation axis depends on the location of the mouse cursor. When you drag it through the area close to the window edges — it will rotate the view around the axis perpendicular to the screen. When you move the mouse cursor in the center of the window — the viewpoint will move on an imaginary sphere around the active object. A horizontal shift causes rotation around vertical axis of this sphere, and a vertical shift — around its horizontal axis.

It sounds a little complicated, but I suggest playing with the view rotations for a moment or two, to “feel” how it works. Sometimes during rotation a part of the model can appear outside the window borders. In such a case, move it (**Shift**–**MMB**, see Figure 2.2.8) back to the center of the view.

Let's look carefully at Figure 2.2.10. Is this plane proportional? The left wing, located closer to the viewpoint, seems to be smaller than the more distant right one... Not surprisingly, because we can see this airplane in the axonometric (orthogonal) projection! (It is a projection, where two parallel lines in space are also parallel on the screen).

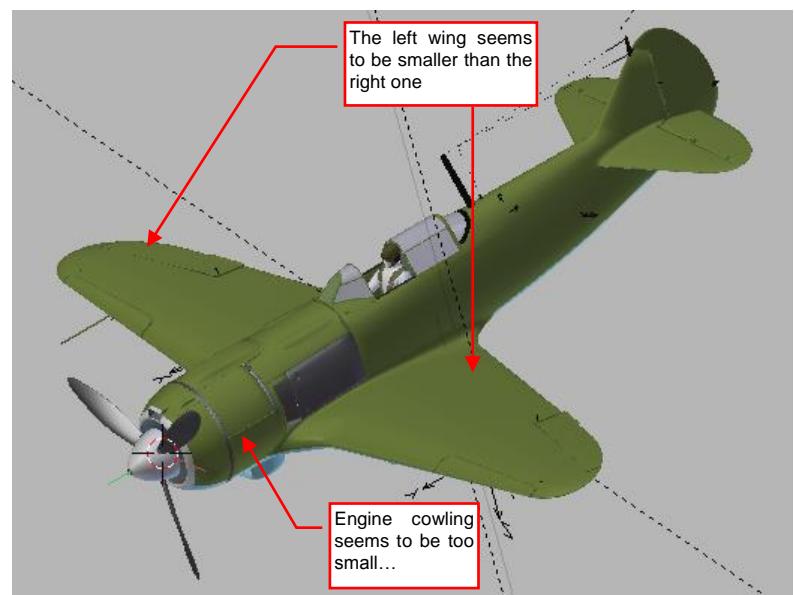


Figure 2.2.10 “Distortions” of the axonometric view

Photographs have accustomed us to the perspective projection — so let's turn it on. Just press the **5** numpad key ([View→Persp/Ortho](#)) — and here it is (Figure 2.2.11):

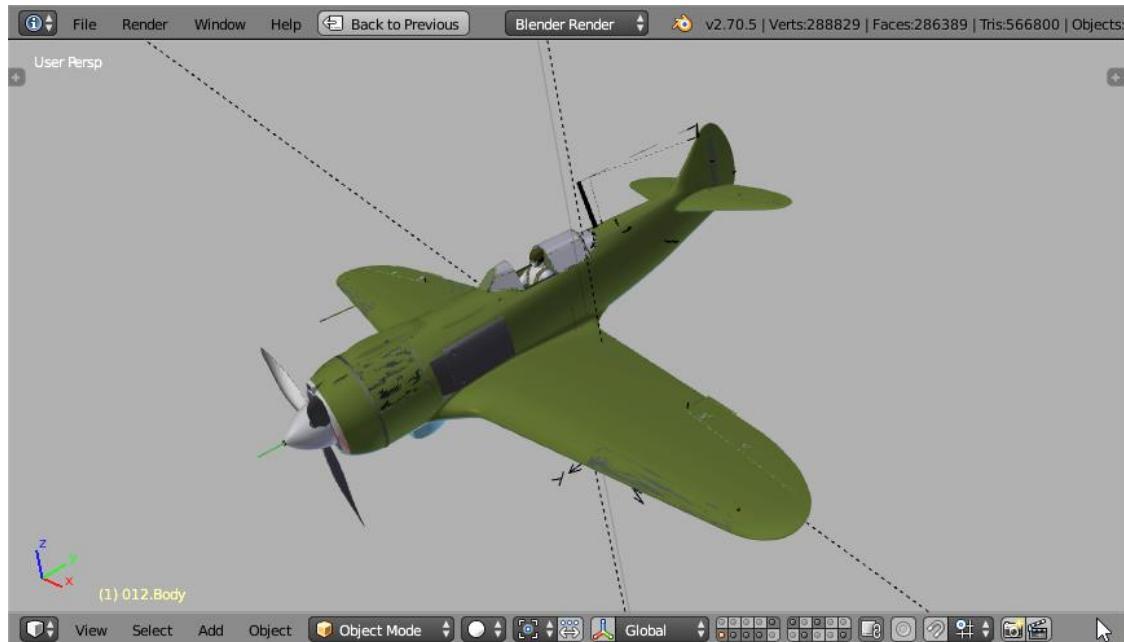


Figure 2.2.11 Initial perspective projection (with a strong “fisheye” distortion)

Well, it seems that the perspective distortion is too strong in this case (Figure 2.2.11) — this airplane looks like through a wide-angle lens! Can we do something with it?

Sure! Press the **N** key ([View→Properties](#)) to open the *Properties* shelf (Figure 2.2.12a). If the Blender controls visible in this pane looks a little bit strange for you — take a look at page 64, where they are described.

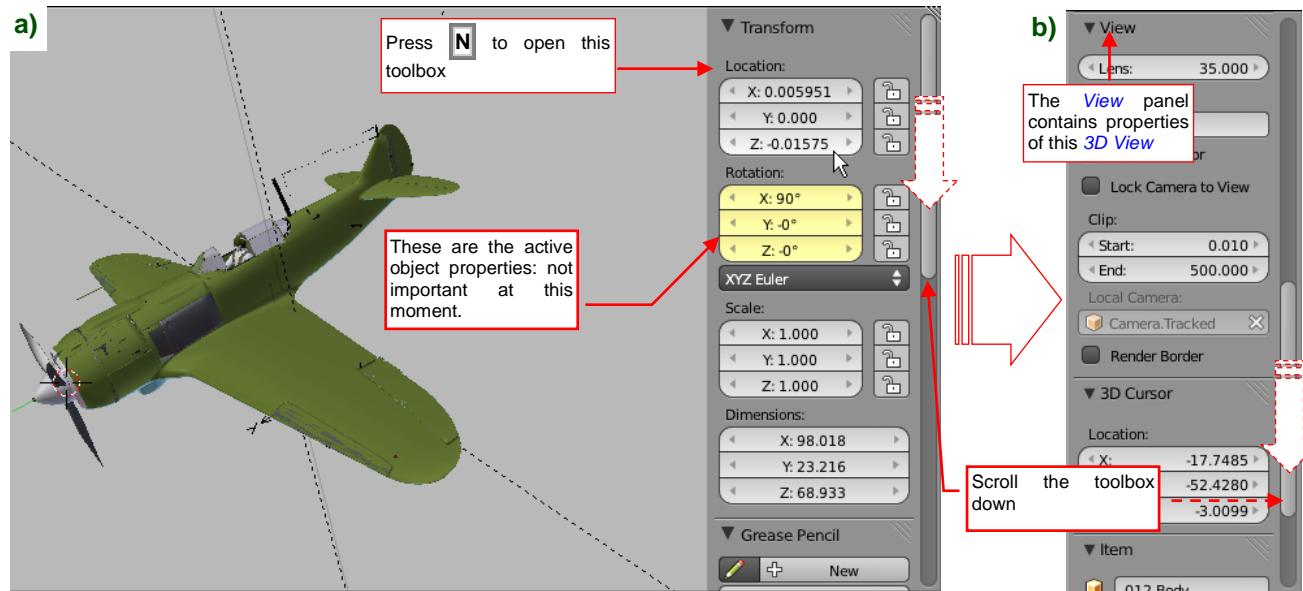


Figure 2.2.12 The *Properties* toolbox and its *View* panel

At the top of *Properties* pane you can see the *Transform* section (panel). It displays some of active object properties (you will learn in the next section what the “active object” is). We do not need them at this moment. Below there are more, initially invisible, panels. Move the slider of this shelf down to see the *View* panel (Figure 2.2.12b). (When necessary, you can reorder panels of this toolbox, so that *View* will appear on its top — see page. 61).

The [View](#) pane contains properties of the current projection. Increase its [Lens](#) value from 35 to **70** units. ([Lens](#) is a numeric field — in case of doubts on how to use it, take a look at page 66). With such longer lens the airplane is less distorted, but it appears too close, so you can see only a part of this model. Zoom it out a little — using the **-** numpad key or scrolling the **MW**. Figure 2.2.13 shows the result:

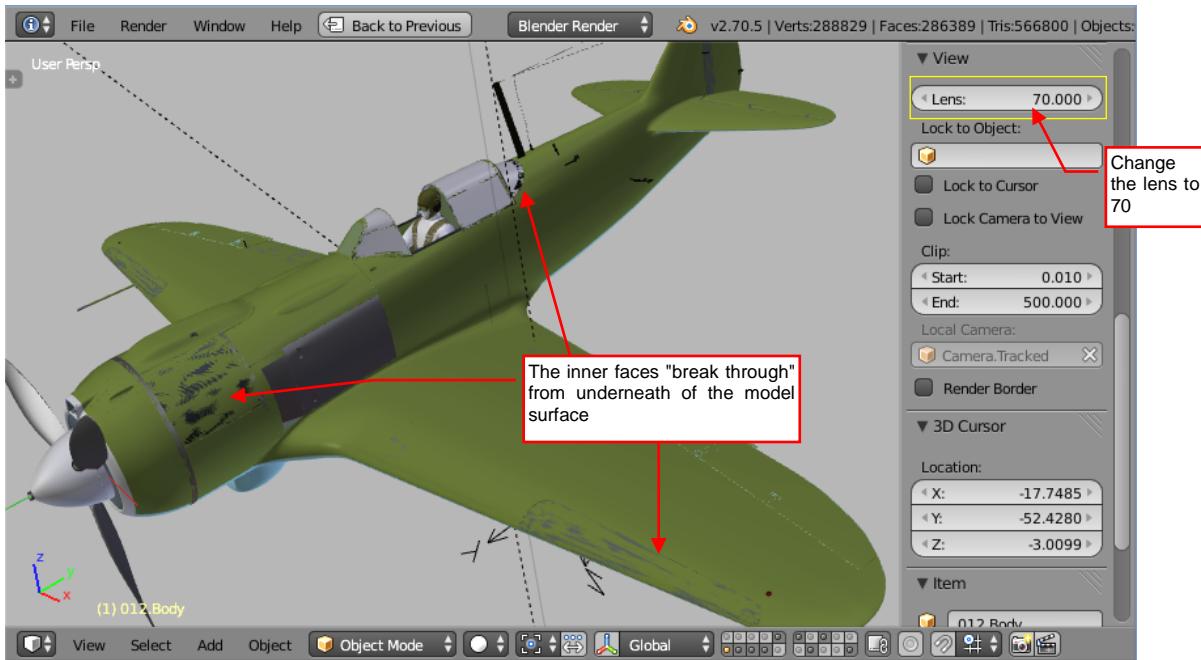


Figure 2.2.13 Adjusted perspective view (with increased [Lens](#) value)

Notice the “penetrations” that have appeared on the model surface. These are faces of the interior objects, which in the real world would lie 1 - 2mm (0.05 - 0.1 inch) below the surface of the airplane. (This is the thickness of sheet metal or plywood sheathing). They were not visible in the axonometric projection, and will disappear when you zoom in the view. Yet this effect is very ugly. How to get rid of it?

It turns out that it is enough to change another property in the [View](#) panel. When you increase the [Clip:Start](#) value (for example to 0.1), the “penetrations” of the inner faces will disappear (Figure 2.2.14):

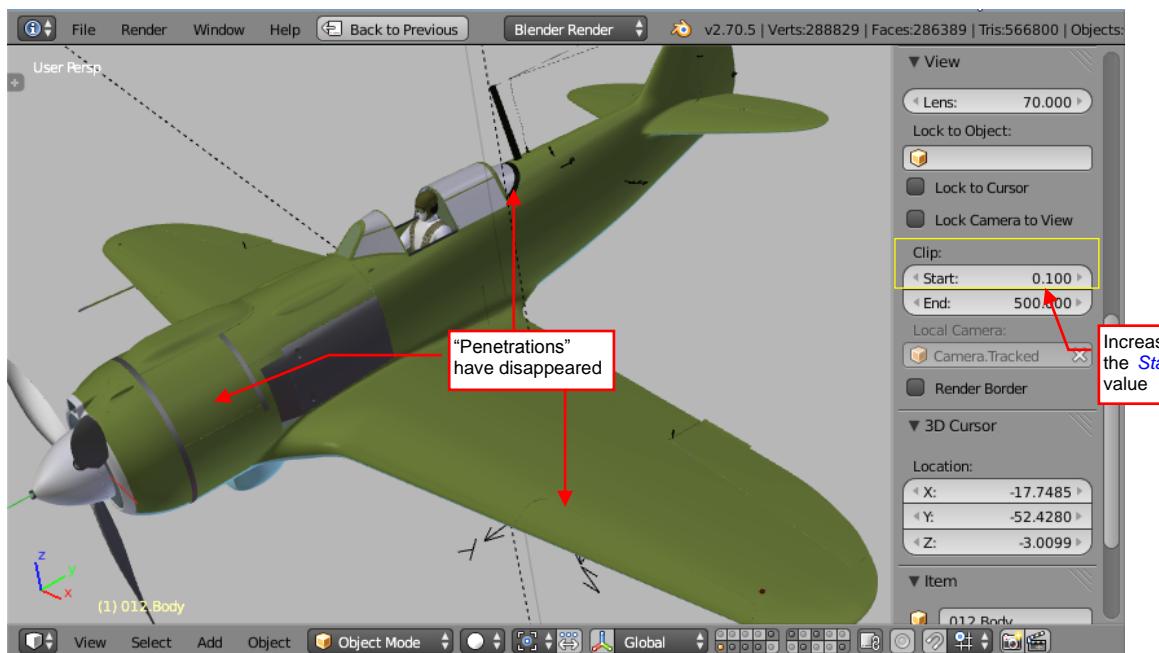


Figure 2.2.14 Adjusted perspective view — increased the [Clip:Start](#) value

(The previous [Clip:Start](#) value was ten times lower — see it in Figure 2.2.13).

If you want to return to the axonometric (orthogonal) projection — press the **5** numpad key again (it acts like a switch). You can also invoke the **View→Persp/Ortho** menu command.

In the perspective view you may encounter a phenomenon which I call “invisible wall”. To see what I mean, try to zoom in the pilot cockpit (Figure 2.2.15):

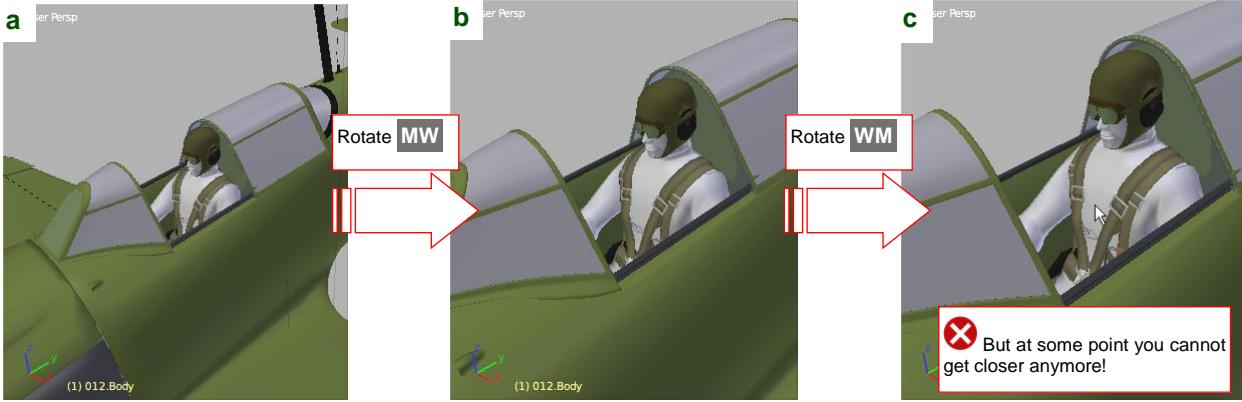


Figure 2.2.15 The “invisible wall” effect during zooming (perspective projection)

Set this part of the model in the middle of the **3D View** window and start to rotate **MW** forward (Figure 2.2.15a). Initially, you will have an impression that you are rapidly approaching the model. However, gradually this movement is becoming more and more slower. Finally you will reach the state, in which any further rotation of **MW** changes nothing (Figure 2.2.15c). What's more, you cannot pan this view (**Shift**-**SPM**) either!

What is the reason for such a strange effect? Well, in perspective projection each view has its focal point, located at a distance from the screen plane. Zooming in the perspective view reduces this distance (Figure 2.2.16):

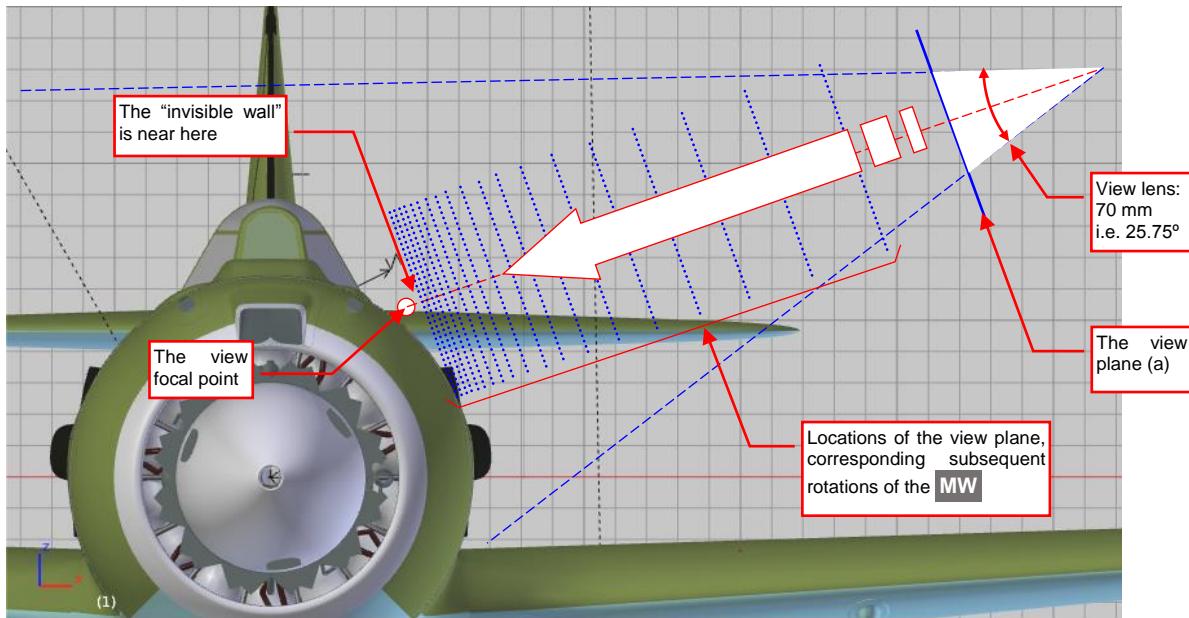


Figure 2.2.16 The reason of the “invisible wall” effect

When you rotate the **MW** forward (or press the **+** key on the numeric keypad) Blender reduces the distance between the observer and the focus by the $\frac{1}{6}$ of its current length. This means that the closer you are to that point, the slower you are moving, and finally you will stuck on a distance close to zero (as in Figure 2.2.15c). Unfortunately, Blender does not offer any direct command to shift the focal point of a perspective view into a new, more distant location!

Fortunately, we can move the view focus in an indirect way. When you encounter the “invisible wall” during zooming, go back a little bit (i.e. decrease the magnification, for example by pressing the **[Z]** key several times) to increase the current distance from the screen to its focal point. When you are able again to pan the screen with ease, press **Shift-F** ([View→Navigation→Walk Navigation](#) — Figure 2.2.17a):

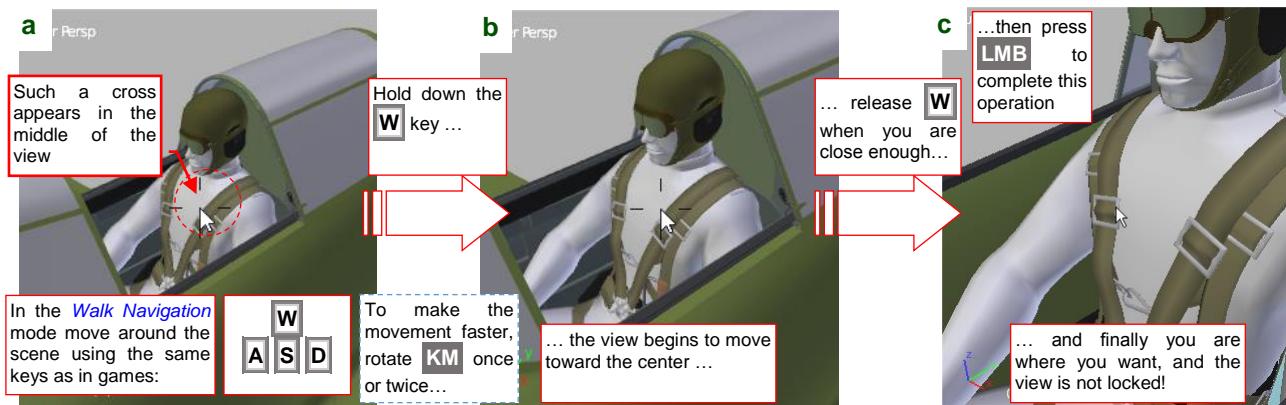


Figure 2.2.17 Using *Walk Navigation* to shift the current view and its focus

In this mode you can move around Blender scene as in games: press **W** to move forward, **S** to move back, **A** — to the left, **D** — to the right. Move the mouse to “rotate your head”, determining the view direction. Thus to pass through the “invisible wall” of a perspective, keep pressing the **W** key to move the focal point along the axis perpendicular to the current view plane. To accelerate this movement you can rotate the **MW** once or twice. When you are close enough to the desired object — click **LMB** to complete this operation (to confirm this new projection). If something went wrong — you can always press **Esc** to cancel it. This is the quickest way to overcome the “invisible wall” effect. (That's why you had to move the view back a little, before — to create a sufficient distance before this operation).

You can find all the commands we have discussed in this section in the [View→Navigation](#) submenu. If you can't remember their keyboard shortcuts, you can use the contents of this menu as the hint (Figure 2.2.18):

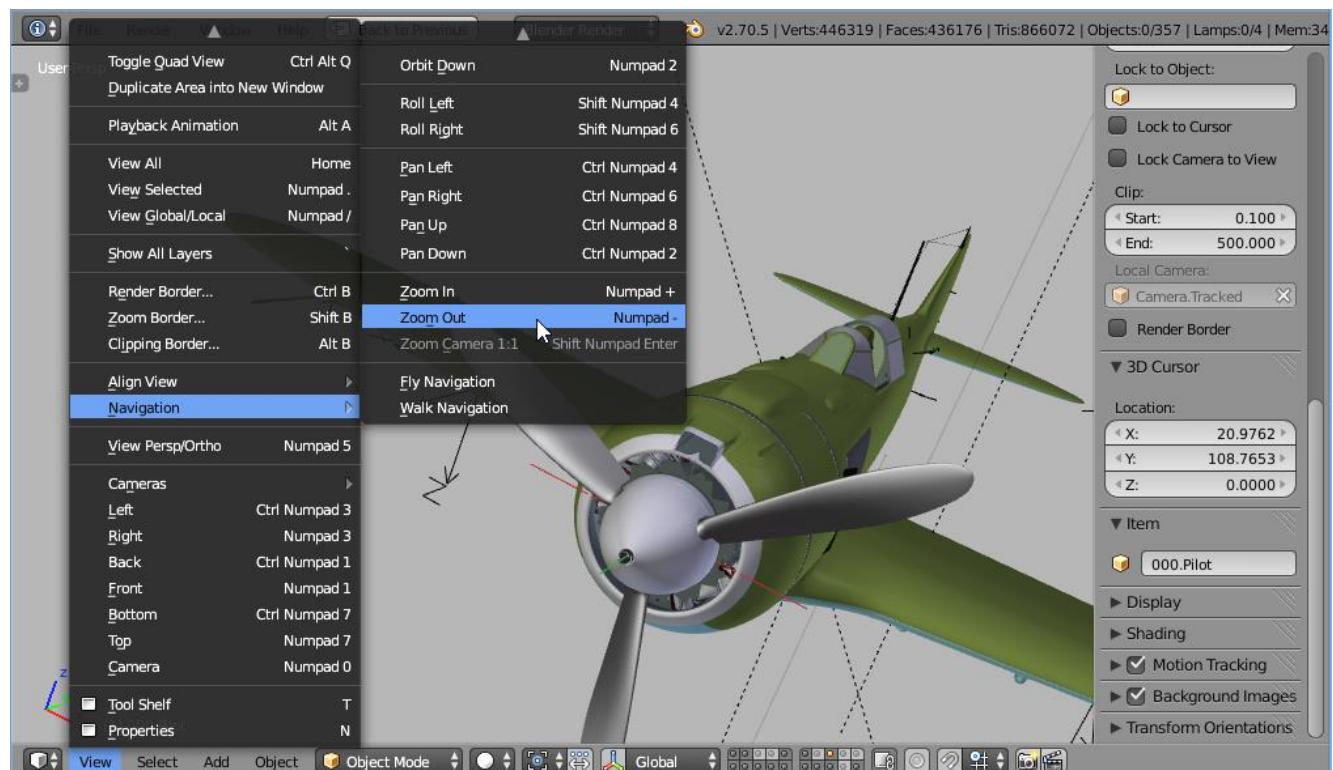


Figure 2.2.18 Commands from the *View* menu

Summary

- Elements of a complex model are placed on different *layers*. Switching these layers on and off, you can quickly control the visibility of the objects you want to work with (page 38);
- To quickly set one of the orthogonal views, use following shortcuts: **1** (front), **3** (left side), **7** (top). To view the scene from the opposite direction — hold down also the **Ctrl** key: **Ctrl-1** (rear), **Ctrl-3** (right side), **Ctrl-7** (bottom) (page 39);
- To **rotate** the view: drag the mouse with the **MMB** pressed (page 41);
- To **pan** the view: holding down the **Shift** key, drag the mouse with the **MMB** pressed (page 40);
- To **zoom** the view: holding down the **Ctrl** key, drag the mouse up or down with **MMB** pressed (page 40);
- Switching the current projection between the axonometric and perspective: use the **5** key (page 42);
- You can change the projection parameters (for example — the lens of the perspective projection) in the *Properties* pane (**N**), on the *View* panel (page 43);
- When you zoom in the model in a perspective view, you can come across the “invisible wall” effect (page 44). Use the *Walk Navigation* (**Shift-F**) command to overcome this issue (page 45).

2.3 3D View—cameras and object selection

There are more ways to view the model than we have discussed in the previous section. To see another one, press the **0** (“zero”) key (**View→Camera**):

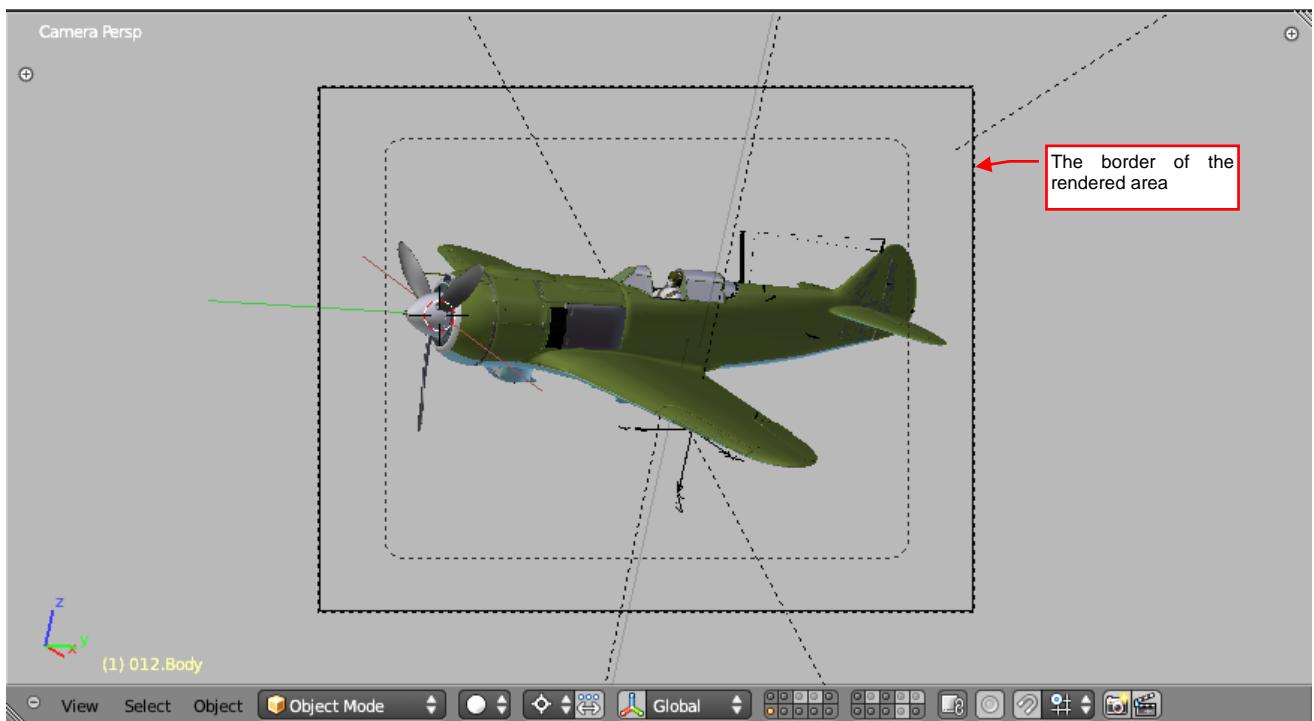


Figure 2.3.1 Camera View

Figure 2.3.1 shows the view from the active camera. Blender will use this projection to create the final render or animation.

What is a camera in Blender? It is a special kind of object, which you can insert into the scene. This La-5F scene contains two cameras. Press **1** and adjust the zoom, to see both of them (Figure 2.3.2):

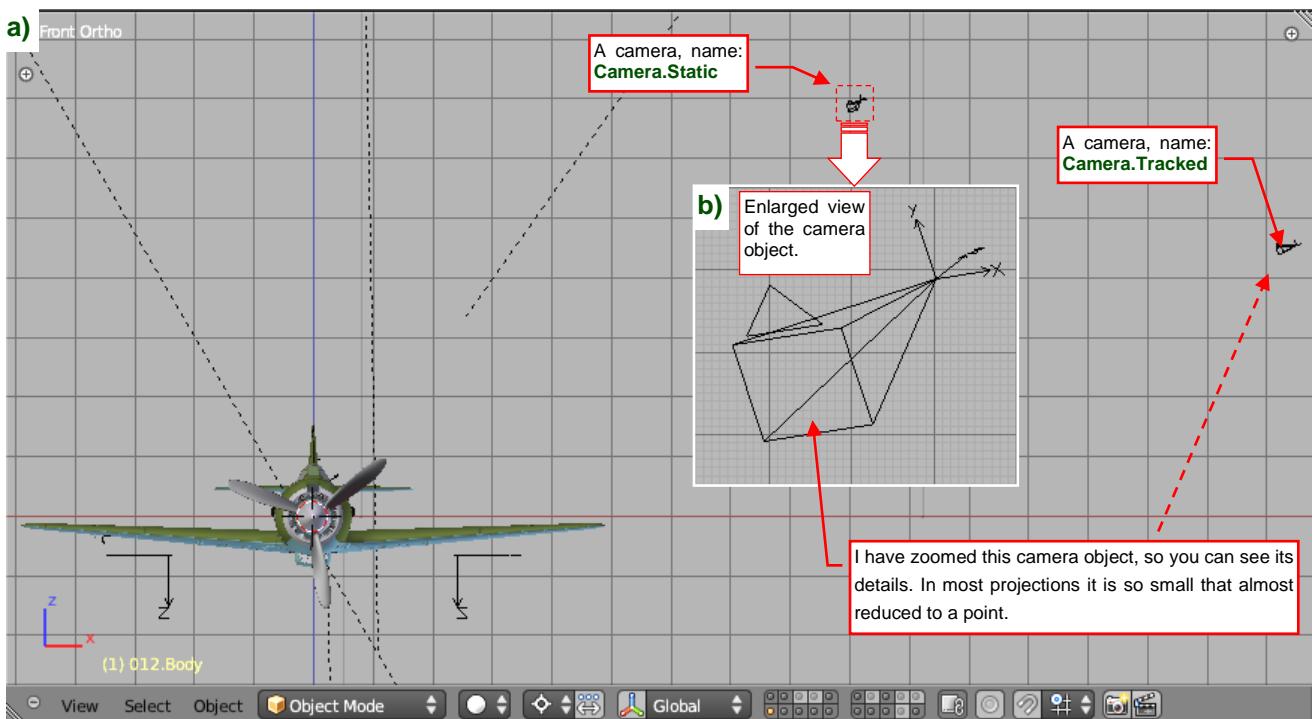


Figure 2.3.2 Two cameras of the La-5 scene

You can move these cameras — just like in a real studio. Before we start doing it (page 51), we will learn how to select objects. It is the first step of every Blender operation.

Click the **RMB** onto the right camera. In response Blender highlights it (Figure 2.3.3):

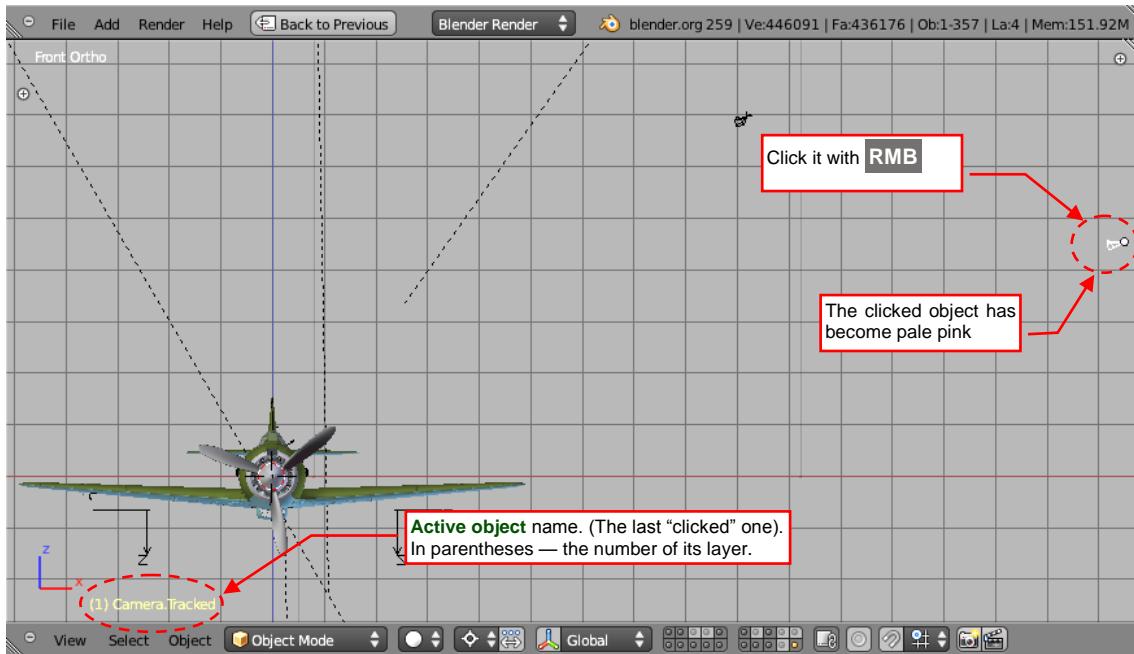


Figure 2.3.3 Selected object (a camera)

The selected object has changed its color to pale pink. This way Blender marks the last “clicked” scene element. You can see its name in the lower left corner of the **3D View**. This is the current **active object**. (The active object is the last selected object).

Press the **A** key (**Select→Select/Deselect All**) now to deselect everything. (This command “clears” current selection, when it exists, or selects everything, when nothing is selected). Notice that the active object name is still on the screen (Figure 2.3.4):

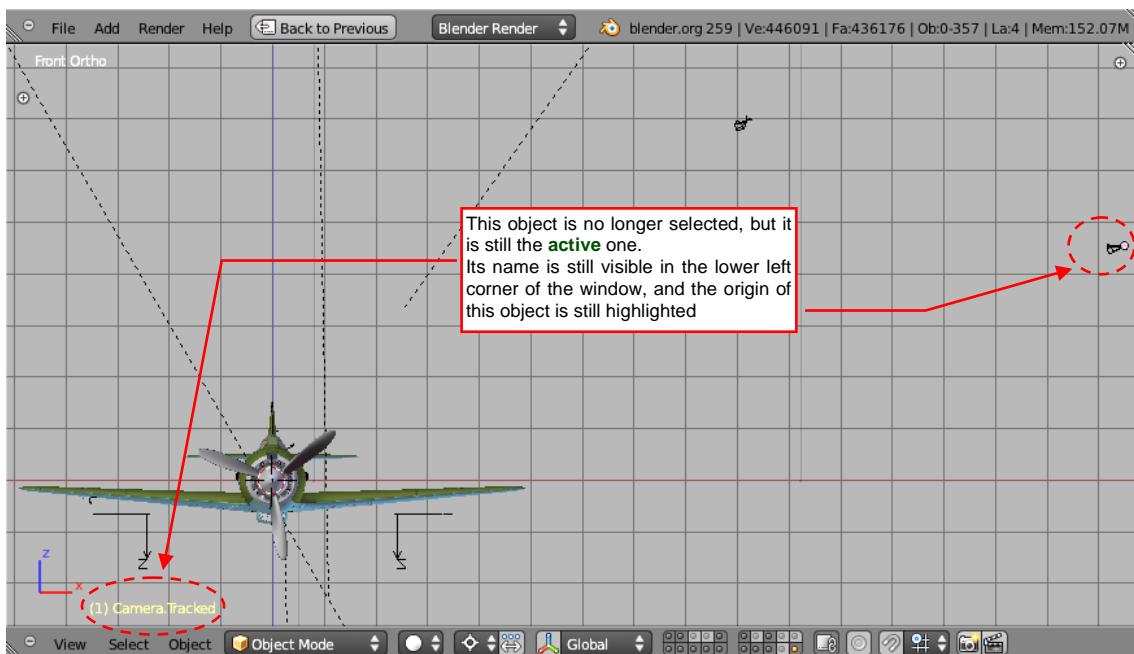


Figure 2.3.4 The last selected object remains active

The active object plays important role in many Blender operations — for example in the **Parent** command, which creates the object hierarchy (see page 306).

To include an additional object to the current selection set, hold down the **Shift** key while clicking objects with the **RMB**. In order to better learn how it works, I suggest you make a few quick experiments.

Start them when nothing is selected. Hold down the **Shift** key and execute the sequence of operations listed by Table 2.3.1. (You can find the camera names on page 47, Figure 2.3.2):

| Holding down the Shift key: | Result | |
|--|--|-----------------------|
| | Selected objects and their colors | Active object |
| Click with RMB into Camera.Static | Camera.Static (pale pink) | Camera.Static |
| Click with RMB into Camera.Tracked | Camera.Static (pink) Camera.Tracked (pale pink) | Camera.Tracked |
| Click with RMB into Camera.Static | Camera.Static (pale pink) Camera.Tracked (pink) | Camera.Static |
| Click with RMB into Camera.Static | Camera.Tracked (pink) | Camera.Static |

Table 2.3.1 The active object and the selected object — some experiments with the group selection in Blender

I will not show the results of the operations listed in Table 2.3.1 on the pictures. The standard adjustments of their contrast and brightness cause the loss of subtle differences between the two colors of the selection: the pale pink (active object) and the pink (other selected objects). The conclusion of these experiments is as follows:

- The **Shift** key modifier serves not only to include, but also to exclude an object from the current selection set. (When you click an object for the second time, it becomes the active one, and when you do it for the third time — it is no longer selected);
- The active object can be out of the current selection (see the last row in Table 2.3.1);

You can also select a group of the objects using a rectangular area. Press the **B** shortcut ([Select→Border Select](#)). In this mode two crosshair lines appear at the mouse cursor. Move the cursor to the place where you want to have the first corner of the selection area. Press the **LMB** and hold it down while dragging the mouse (Figure 2.3.5). It will drag the opposite corner of the selection rectangle. Locate it in the proper place, and release the **LMB**. It selects all the objects that are inside this area (also the ones that lie across its borders). [Border Select](#) command does not alter the current active object.

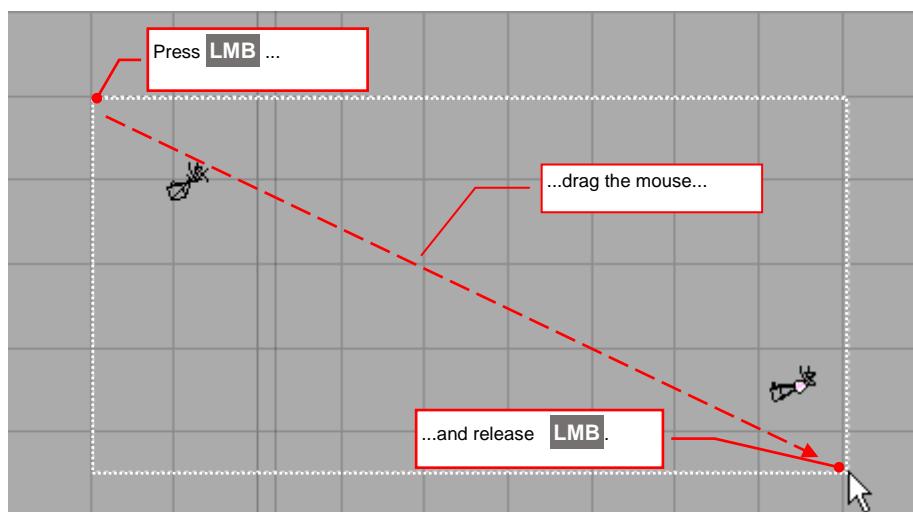


Figure 2.3.5 Selecting objects with the rectangle ([Border Select](#))

To exclude the objects inside this rectangle from the current selection, after invoking [Border Select](#) command (the **B** key) use **MMB** instead of **LMB**.

The [Select](#) menu of [3D View](#) window contains many other commands, such as selection by the layer, or by the object type (Figure 2.3.6):

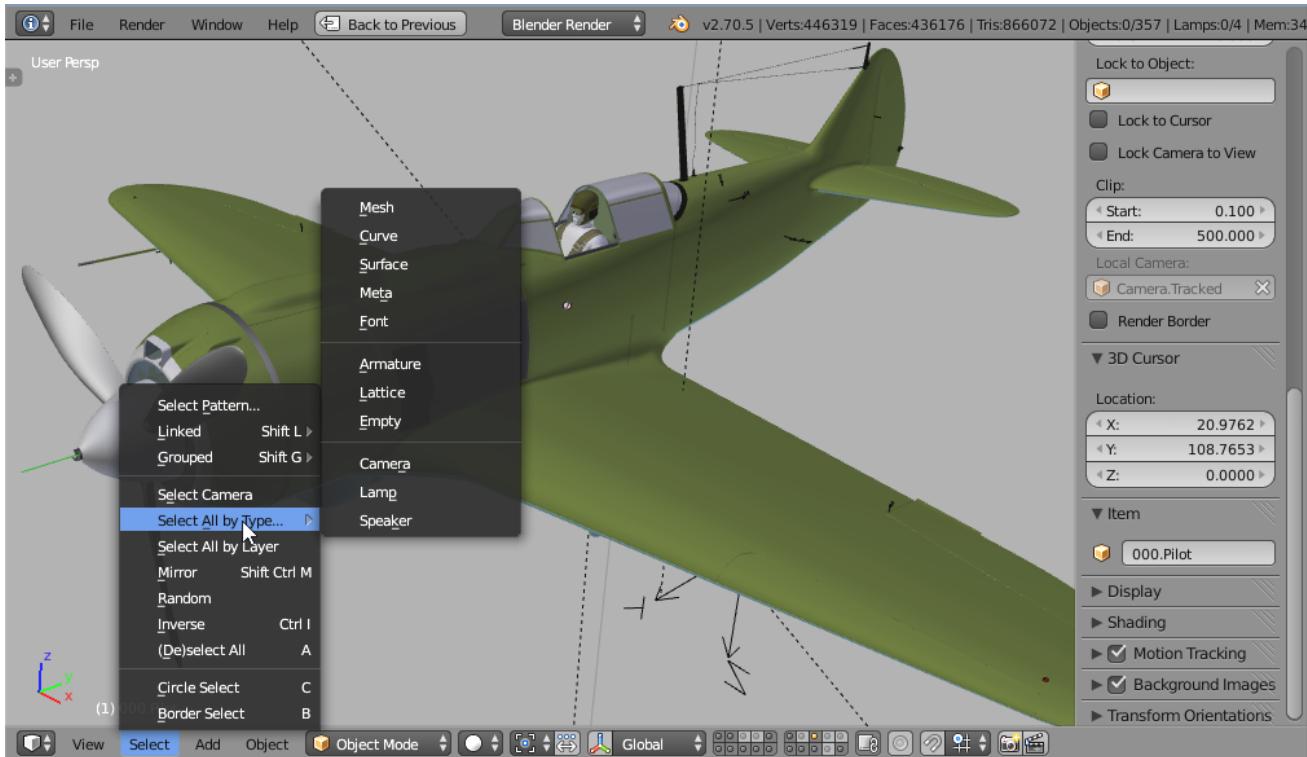


Figure 2.3.6 Other selection commands, available in the [Select](#) menu.

- You can find description of few other useful selection methods on page 322.

Summary

- To display the view from the active camera in current window — press the **0** key ([View→Camera](#)) (page 47);
- To select an object, click it with **RMB**. Hold down the **Shift** key, to add it to the previously selected objects (page 48);
- To select objects using a rectangle — press the **B** shortcut (or invoke the [Select→Border Select](#) command). Then press the **LMB** in the first corner of the rectangle, and drag the mouse to set the opposite corner. To exclude objects from the current selection — begin in the same way (**B**), but use the **MMB** instead of the **LMB** (page 49);
- Blender distinguishes the last clicked (with the **RMB**) object from the rest of selection. It is referred as the **active object**, and its name is displayed in the lower left corner of the [3D View](#) window (page 48);

2.4 3D View – object movement and rotation

In this section I will show how to change the location and orientation of the active camera (it is also an object). However, before that:

- select the **Camera.Tracked** object. It is the active camera of this La-5 scene. Make sure that you can see its name in the lower left corner of the **3D View** window;
- tile the active **3D View** (**Ctrl** – **T** or **Area Options** → **Tile Window**), and join two other windows, leaving just two views on the screen. In one of them set up the camera view (**0**), in another — the front view (Figure 2.4.1). We will work in the left window, simultaneously observing changes in the right one:

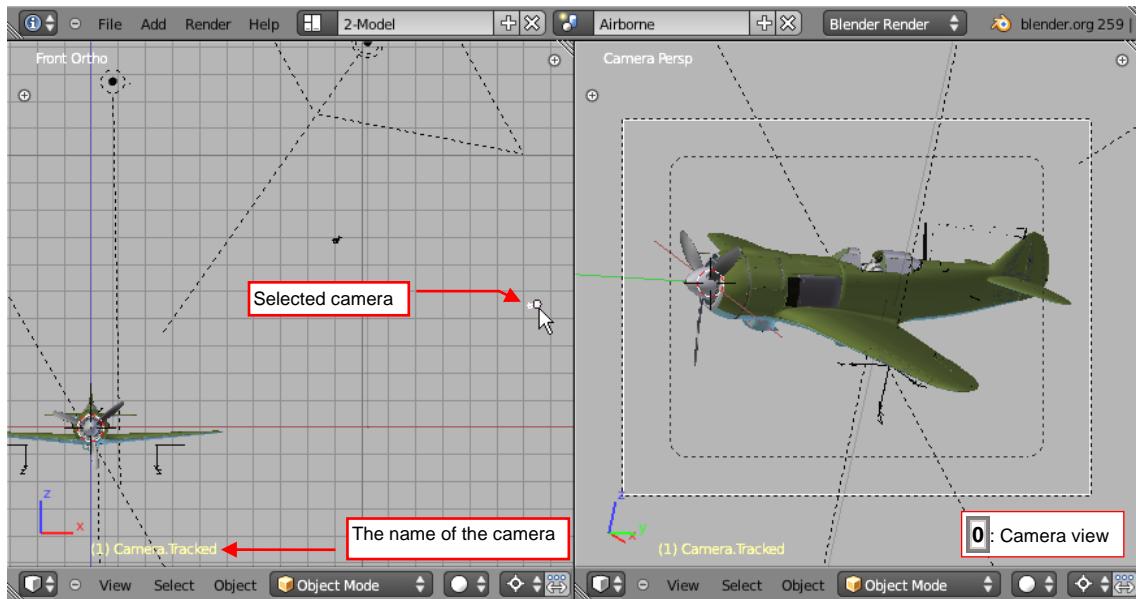


Figure 2.4.1 Screen layout, proposed for this section

Press the **G** key (or **Object** → **Transform** → **Grab/Move**). From now every movement of the mouse moves the selected object (the camera, in this case). Notice the changes in the camera view window (Figure 2.4.2):

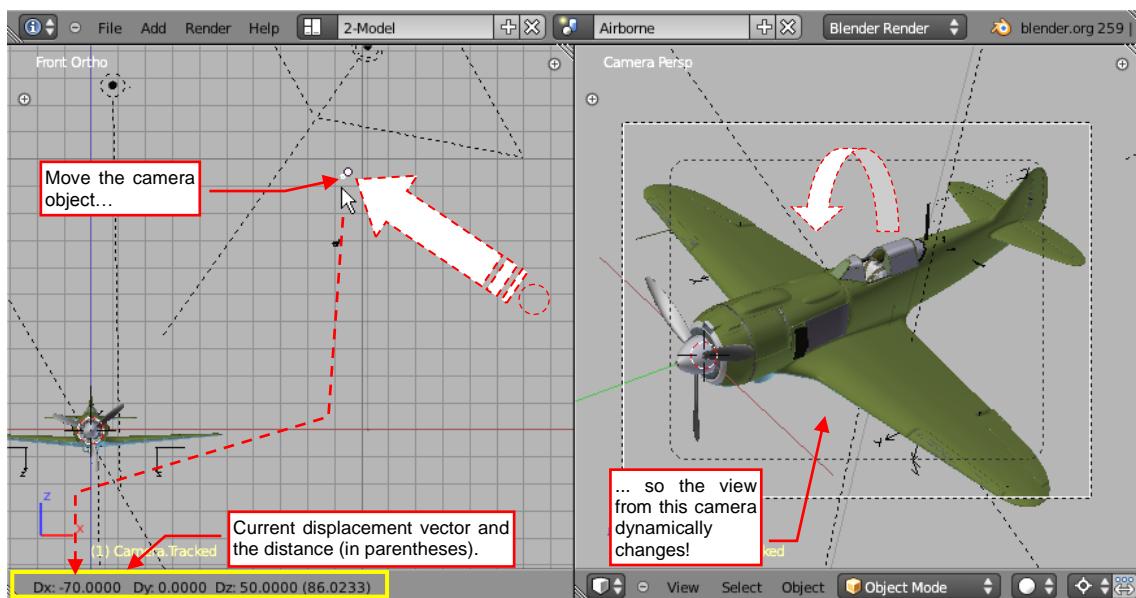


Figure 2.4.2 Moving the object

During this operation Blender shows in the active **3D View** window header current **X,Y,Z** coordinates and the length (in parentheses) of the offset vector. Click **LMB** to accept current location of the object, and finish this movement. (To cancel this operation, press the **Esc** key).

How does it happen that during this movement the camera is keeping the aircraft in the center of its view? Because it is tracking this model. I will describe how to create such a link later in this book. Now I would like to demonstrate what is actually tracked by this camera.

To show this “target” object, we have to hide some layers, and enable the others. However, you may encounter a problem (at least on the smaller displays): there is no visible layer indicator on the **3D View** window headers! Each pane in this tiled screen layout is so narrow that this control is hidden somewhere beyond the window edge.

Little secret: you can move the window headers in Blender, just like you pan the **3D View** scene. Just place the mouse cursor over a header, and press **MMB**. Holding it down, drag the mouse left or right. Can you see that the entire header is moving? Shift it to the left, to make the layer control visible (Figure 2.4.3).

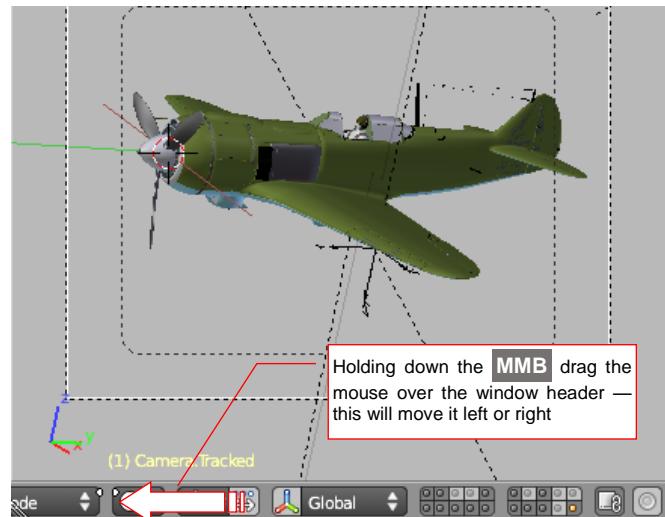


Figure 2.4.3 Shifting the window header

Now turn off layer **11** and turn on layer **10**. Oh, I have to introduce the concept of layer numbering! To keep the explanation short, let's assign to each layer the following numbers (Figure 2.4.4):

| | | | | |
|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 |
| 11 | 12 | 13 | 14 | 15 |

| | | | | |
|----|----|----|----|----|
| 6 | 7 | 8 | 9 | 10 |
| 16 | 17 | 18 | 19 | 20 |

Figure 2.4.4 Numbers assigned to Blender layers

Thus flipping the states of layers **11** and **10** gives following layer visibility: . Turning off layer **11** hides the wings and the middle and rear part of the fuselage. Layer **10** is one of the two layers that contains the auxiliary “control” objects, not intended to appear on the final render. (Another layer in this scene that groups such objects is layer **9**). Now you can see in the place of the invisible wings of our model the slots, ailerons and flaps, and in the place of invisible tail plane — its rudder and elevators (Figure 2.4.5):

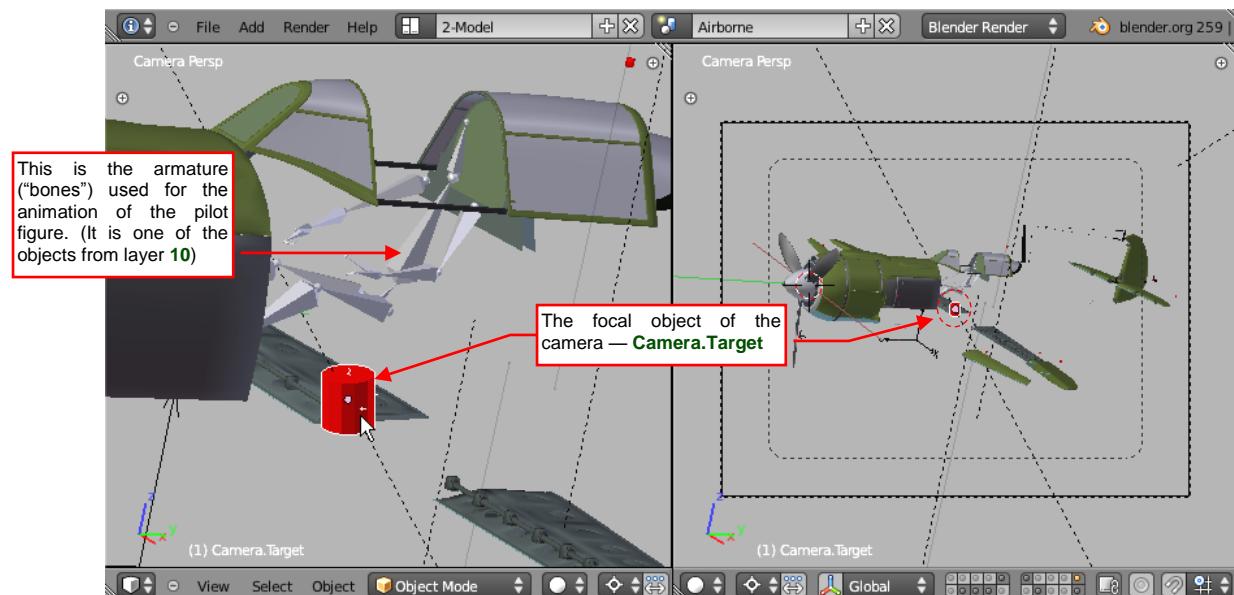


Figure 2.4.5 The **Camera.Target** object — the focal point of our camera

Somewhere in the cockpit area there is a relatively large, red cylinder. It is named **Camera.Target**. Select it (see Figure 2.4.5).

Let's say that we would like focus our camera on the engine area instead of the cockpit. To place it there, you have to move the **Camera.Target** object along the **Y** axis. You can do it in any view projection, even in the perspective.

Press the **G** key to shift the selected object. Immediately afterwards press the **Y** key – it restricts the current movement to this axis. The **Y** axis has become white, and the information displayed in the window header is different — it shows a value of the single **Y** coordinate. Now, every mouse movement shifts this object along the **Y** axis as on a wire (Figure 2.4.6):

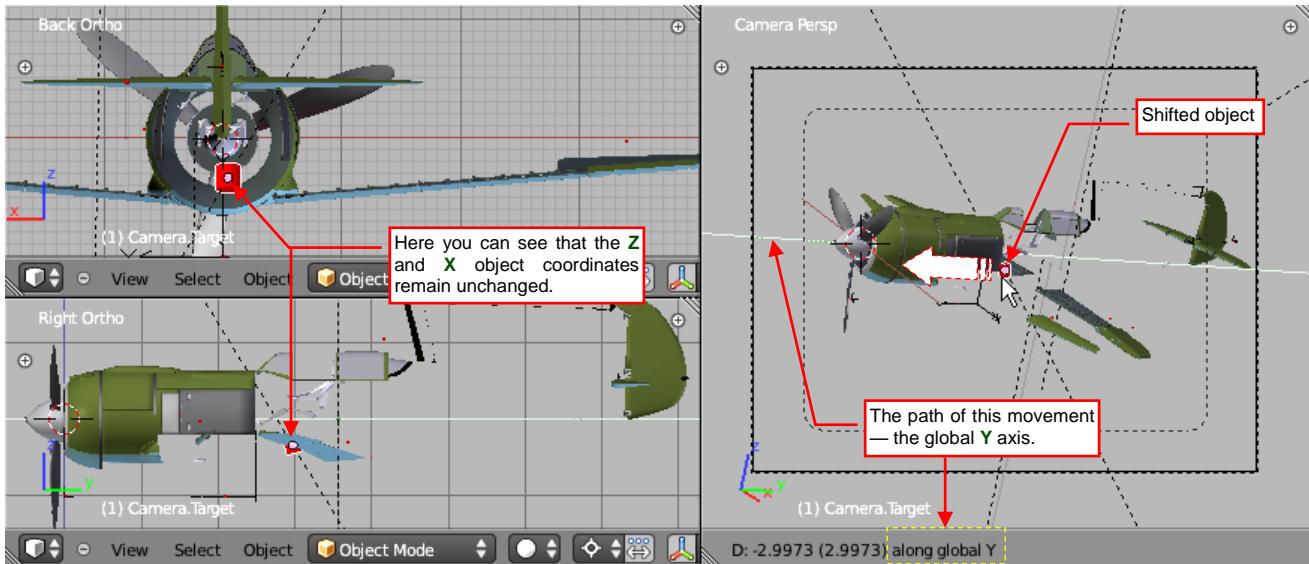


Figure 2.4.6 Constrained object movement (along the **Y** axis)

As you can see, the camera keeps the **Camera.Target** object in the center of the window. Do not end this movement yet. I will show you another effect: while dragging the mouse, hold down the **Ctrl** key. Do a relatively long move. Now you will see that the object is moving in steps of 10 units (Figure 2.4.7):

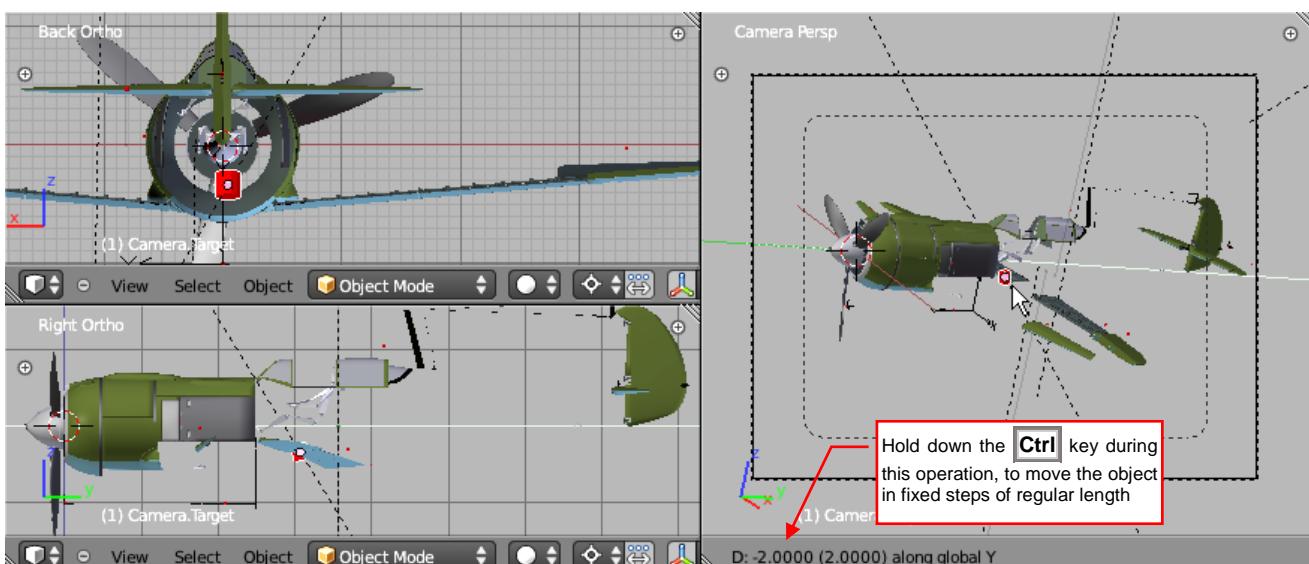


Figure 2.4.7 Moving the object by regular steps (holding down the **Ctrl** key)

- Holding down the **Ctrl** key during every transformation (moving, rotating, scaling) changes the values being transformed stepwise, by a fixed increment.

(I have shown the **Ctrl** key in action during the shifting of an object along **Y** axis just as an example. You can use it in any transformation).

Why does the basic “step” of the object movement in this window have a length of 10 units? It depends on two factors: the current magnification and the *3D View* projection. Cancel the current operation (**Esc**) and start it again but in the left side view (the lower left window depicted in Figure 2.4.7). When you hold down the **Ctrl** key here, the basic step of the object movement will have length of 5 units – two times less, than in the perspective view. It seems to be a general rule – in *3D Views* with the “orthogonal” projection (from the top, side, or the front..) these steps are smaller. This is the approximate distance between the nodes of the helper grid, which is displayed by the *3D View*. When you zoom in the view, its grid is getting denser. Simultaneously the basic movement step forced by the **Ctrl** key decreases, respectively, from 5 to 1 unit.

Now another effect: hold down the **Shift** key while dragging this camera. What happens? The same mouse movement causes a smaller displacement of the dragged object!

- Holding down the **Shift** key during every transformation (moving, rotating, scaling) reduces the “influence” of the mouse. (The same mouse movements cause smaller object displacements on the screen). It makes easier to set precisely the object onto its final place.

What happens when you hold down the **Ctrl** and **Shift** keys simultaneously? We will obtain a stepwise movement that has its basic step ten times smaller. In the camera view it decreases the basic translation step from 10 units (the **Ctrl** key down, only), to 1 unit (both **Shift** and **Ctrl** keys down).

I would like to show the object rotation with an example in the side view. Yet there is a small problem — the object to be rotated is obscured by the wing flap. Let's say that you do not want to turn off the visibility of the layer containing that flap, because you would miss the contour of the wing trailing edge in this way. In such a case you can alter the way in which Blender draws the contents of this window, switching it from the *Solid* to the *Wireframe* mode. Select the *Wireframe* as the current drawing mode from the *Viewport Shading* dropdown list. It is located on the *3D View* window header (Figure 2.4.8):

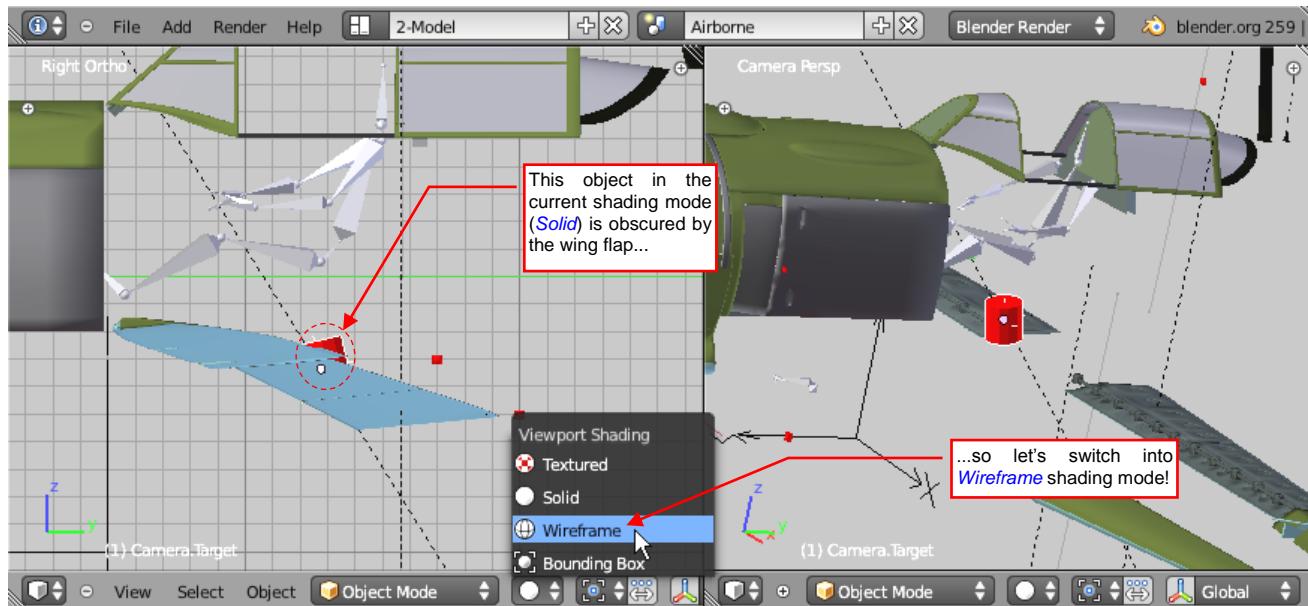


Figure 2.4.8 Changing the *Viewport Shading* mode

Figure 2.4.9 shows the result of changing the shading mode. As you can see, it is an individual setting of each window. (The right window still uses the *Solid* shading on this illustration).

In practice, the *Solid* and *Wireframe* modes are used most often. The third method of shading (*Textured*) will be used later in this book — when we deal with textures.

How to rotate an object? In a similar way to object movement: press the **R** key (**Object→Transform→Rotate**) to start the rotation (Figure 2.4.9):

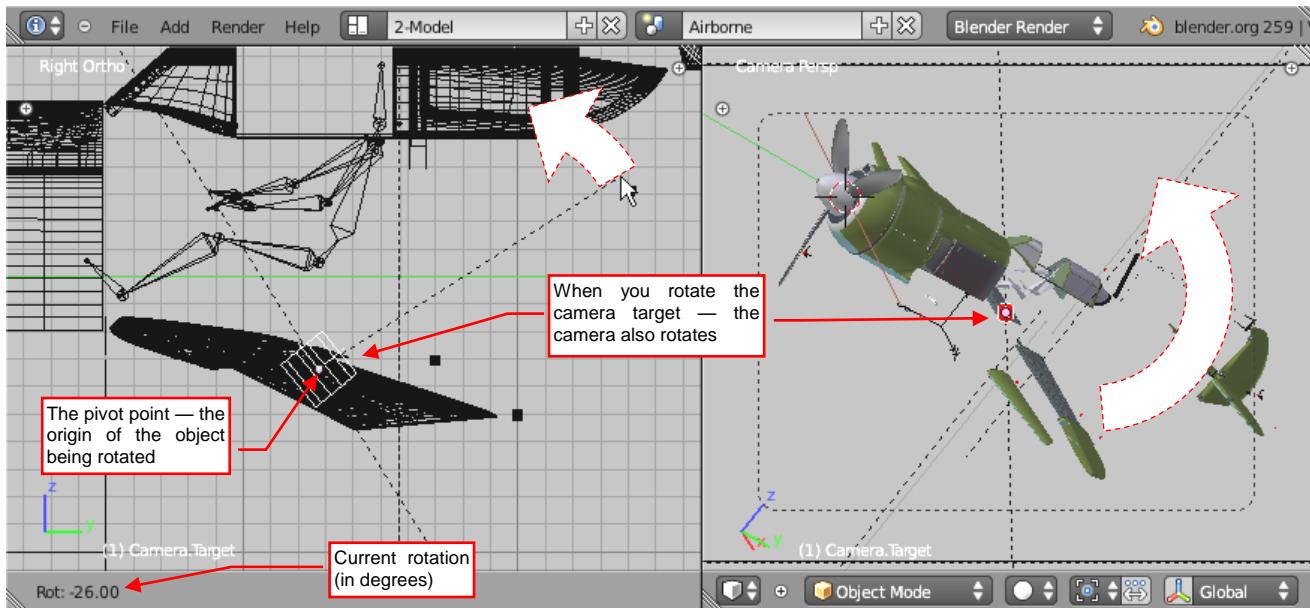


Figure 2.4.9 Rotation (around the origin of selected object)

By default, the object is rotated on the current screen plane. The window header shows the current rotation angle. You can also force the rotation around any of the **X**, **Y**, **Z** axis, by pressing one of these keys on the keyboard (**X**, **Y**, or **Z**).

The rotation of the **Camera.Target** object also rotates the camera (see the right window). This follows from the fact that the camera is “tracking” this object, so the local **Z**-axis of the camera points always in the same direction as the local **Z**-axis of **Camera.Target**. You can also use the **Shift** key while rotating, to obtain a precise rotation, by fractions of a degree. Use the **Ctrl** key to rotate by angle increments of 5°, and **Shift-Ctrl** to rotate by increments of 1°.

If you want to rotate around other point than the origin of selected object, you have to use the **3D Cursor**.

You know the role of the flashing caret in any text editor. It shows the place where the new text will be inserted, if you start to type. This concept was copied into the 3D world. In Blender there is also such a special point, which you can place anywhere in the scene. They also named it **cursor**, but with the **3D** prefix. It marks the location where the new object will be placed, if you create it at this moment. It can also serve as the pivot point in such operations as rotation or scaling.

3D cursor is drawn as a small cross, surrounded by the red and white circle (Figure 2.4.10). You can put it anywhere, clicking the **LMB** in the desired place of the scene. You can also force it to move to the nearest grid point. To do this, press the **Shift-S** shortcut (**Object to Snap**) and from the menu that appears, select the **Cursor to Grid** command (Figure 2.4.10).

This second step — selection of the menu item — you can do also with the left hand, pressing the **6** key. (The “6” is the position of the **Cursor to Grid** command in the **Snap** menu, counting from the top).

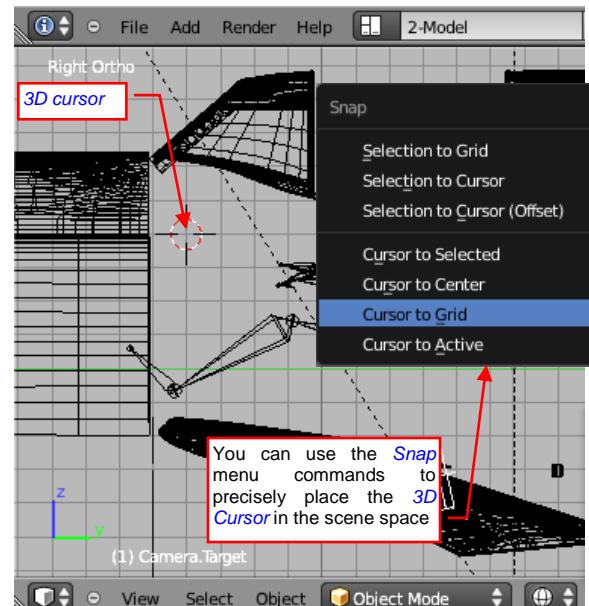


Figure 2.4.10 3D Cursor and the Snap context menu

To rotate the **Camera.Target** around any spatial point, place the **3D Cursor** (with a **LMB** click) in appropriate point on the screen. (You will often have to set its location in two perpendicular views. The **Snap** menu commands are also helpful in such a case).

Then switch Blender into the mode that uses the **3D Cursor** as the reference point. To do it, use the **Pivot** dropdown, from the **3D View** window header (Figure 2.4.11). Select the **3D Cursor** option from this list (or press **I** on the keyboard):

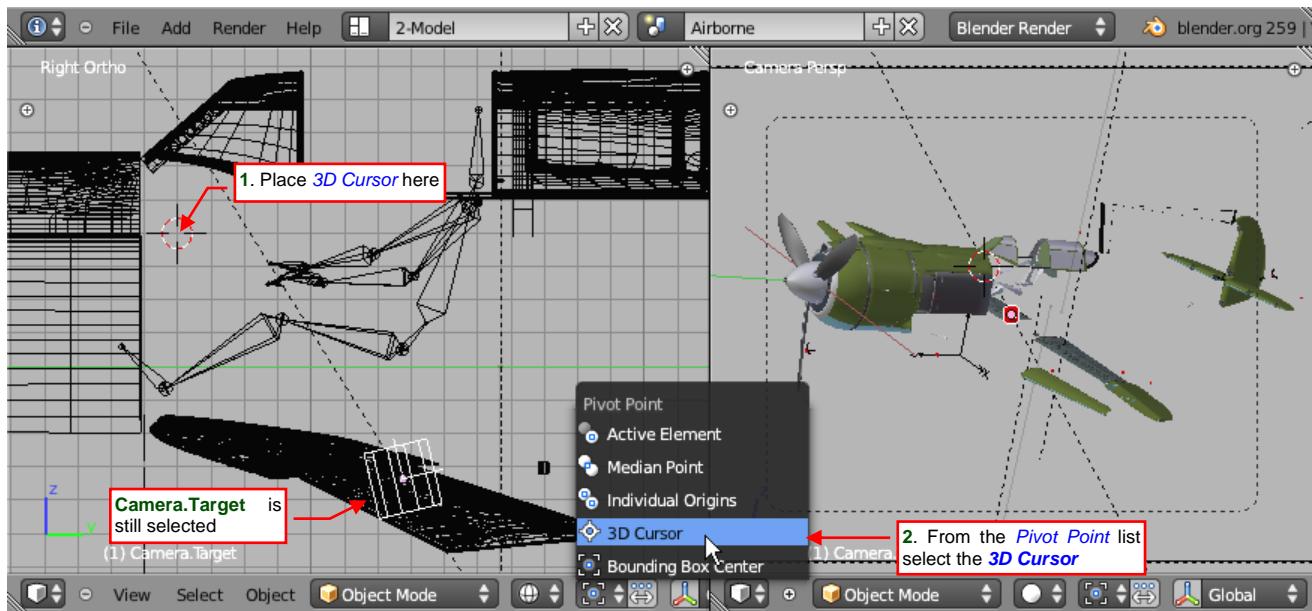


Figure 2.4.11 Changing the **Pivot** mode to **3D Cursor**

Make sure that the **Camera.Target** object is still selected.

The remaining steps are the same as before — press the **R** key and rotate the object (Figure 2.4.12):

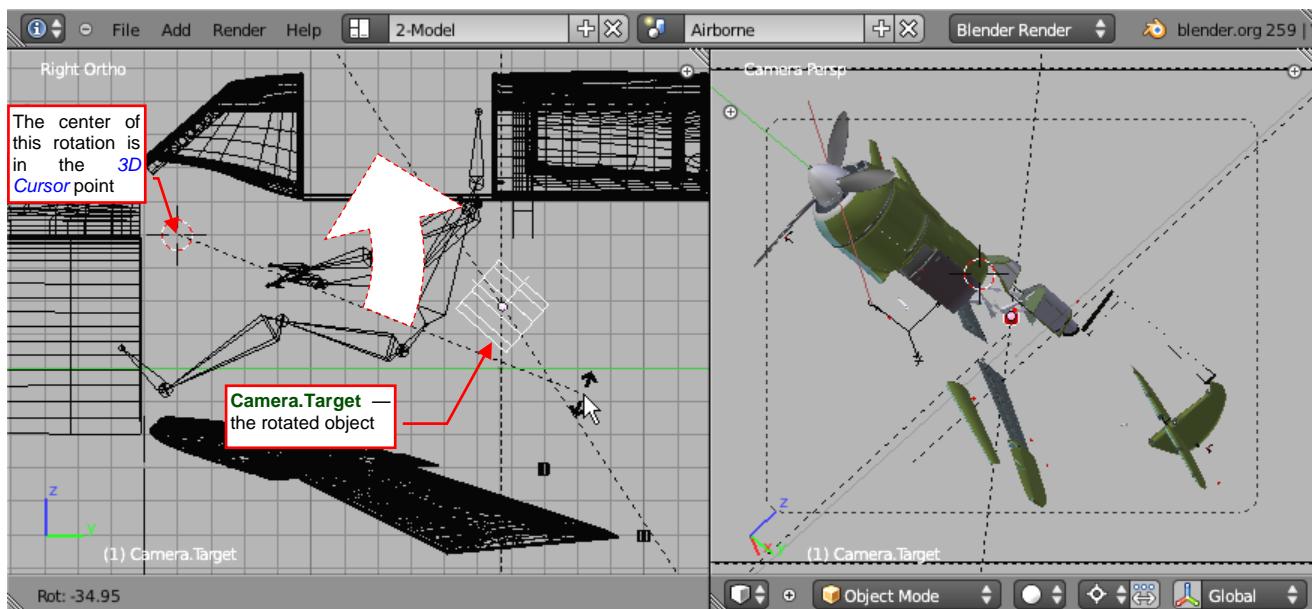


Figure 2.4.12 Rotation around the **3D Cursor** point

When you finish, set the **Pivot Point** back to the **Bounding Box Center** mode (**I**). Alternatively, you can also switch it into the **Individual Centers** mode (**Ctrl-I**).

I will not describe here the last one of the three basic object transformations: scaling. So far I have tried to discuss each topic in a way that allows you to immediately use the acquired knowledge. For example, by rotating and moving **Camera.Target** and **Camera.Tracked** objects you can arrange for the render a new projection of the La-5 model. (I will introduce rendering in the next section). Unfortunately I did not find in this scene and at this point of narration any useful application of scaling. I will only mention that changing an object size is very similar to the rotation. The reference point is also determined by the **Pivot Point** mode. It can be the object origin or **3D cursor**. To start the scaling operation, press the **S** key (**Object→Transform→Scale**). More on this subject you can find on page 291.

If you want to type exact numerical values of object transformation instead of using the mouse, press the **N** key to open the **Properties** pane (**View→Properties**). On its **Transform** panel (Figure 2.4.12) you can find the current **Location** coordinates, **Rotation** angles and **Scale** coefficients of the active object. If you rotate the active object with the mouse, Blender will update the values displayed in this panel “on the fly”. It also works in the opposite: when you enter a new value into one of the **Transform** panel fields, you will immediately see the effect of this change in all **3D View** windows.

Note that each of the transformation fields has a lock icon (in the **Properties** pane. When its is clicked — the coordinate of this field is excluded from the changes applied during object transformations. It is very convenient when you want to resize an object only in the one plane — let's say **XZ**. Before this operation lock the **Dimensions:Y** or **Scale:Y** field. Then, when you start the scaling operation (**S**), which usually resizes everything uniformly in all directions, the object size along the **Y** axis will remain unchanged.

During transformation (rotation, translation, scaling ...), you can also enter exact numerical values of their parameters. Just press the **=** key (Figure 2.4.14):

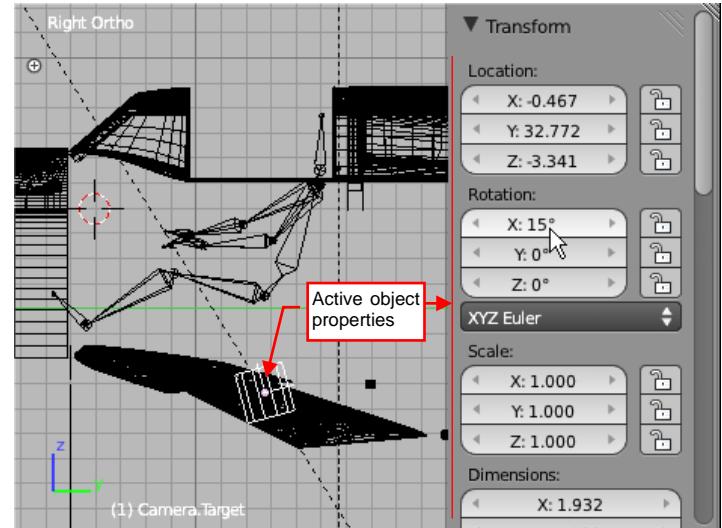


Figure 2.4.13 **Properties** of the active object

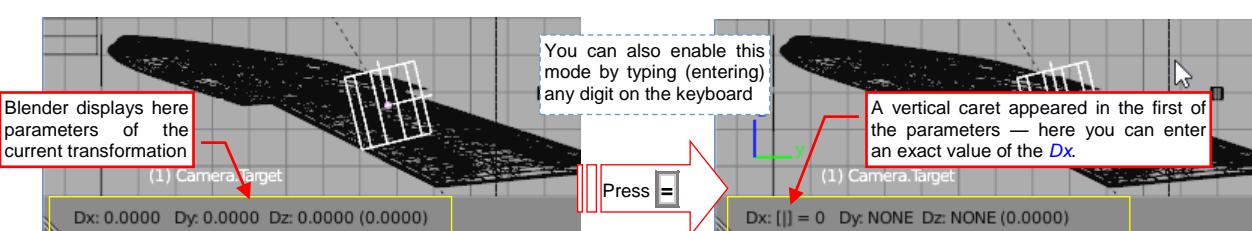


Figure 2.4.14 Switching into manual entering of transformation input values

Blender places a text caret in the first field of the transformation parameters. Now you can enter a new value typing it on the keyboard. To shift caret to the next field, press **Tab** (Figure 2.4.15). Press **Enter** to complete this transformation, or **Esc** to cancel it.

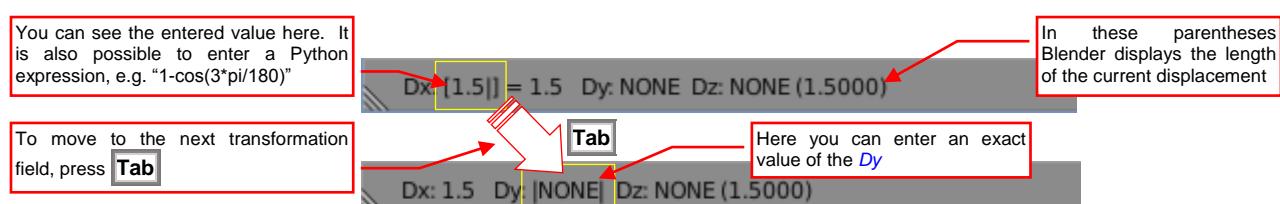


Figure 2.4.15 Entering exact numeric values of the transformation parameters

In the **Properties** pane you can also check or change the exact coordinates of the **3D Cursor**. Just scroll the content of this toolbox upward to expose the **3D Cursor** panel. Then inspect or update its **Location** values (Figure 2.4.16):

Finally, let's talk about what to do when you have made a mistake and you want to revert your scene to the state before the last operation. Like the most editors, Blender provides multiple **Undo** and **Redo** levels. These two operations are so important that I have put their shortcuts in the boxes below. (To make them more visible when something has gone wrong in your Blender scene).

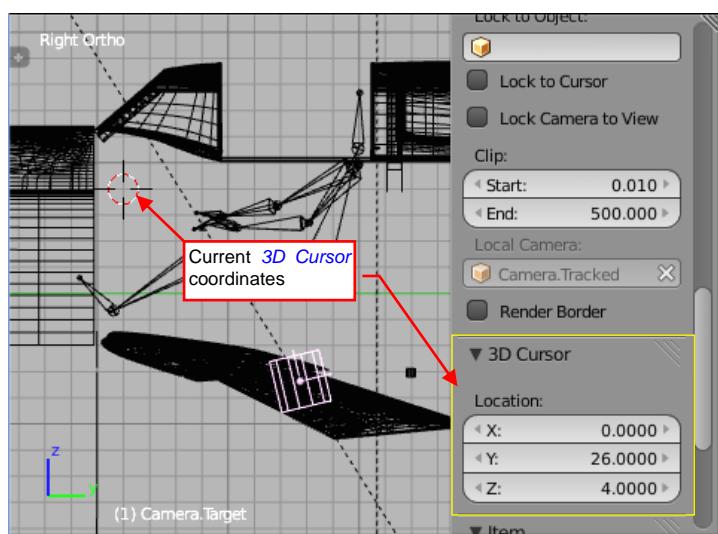


Figure 2.4.16 *View properties: 3D Cursor location*

- **Undo** – reverting the last operation: press the **Ctrl-Z** keys (or invoke the **Object→Undo** command).
- **Redo** – reverting the last **Undo**: press the **Ctrl-Shift-Z** keys (or invoke the **Object→Redo** command).

It is worth mentioning that the **Object** menu is present on the **3D View** header only in the **Object Mode**, which we are currently using. There are also other modes, like the **Edit Mode** (edition of a single mesh), which we will discuss later. In these other modes you will find different menus on the **3D View** header, in place of the **Object** menu (for example — **Mesh**, in the **Edit Mode**). In each of them there are **Undo** and **Redo** commands. Yet it is better to remember their keyboard shortcuts. In some other editors, like **UV/Edit Image**, you will not find these commands in the **Image** menu, nor in any other menu. (It seems to be an error in the user interface design). Nevertheless the **Undo** and **Redo** are still available there — you have to invoke them using their shortcuts.

Our introduction to Blender fundamental transformations ends here. There are obviously many other transformations in this program — as you can see in the **Object** menu (Figure 2.4.17):

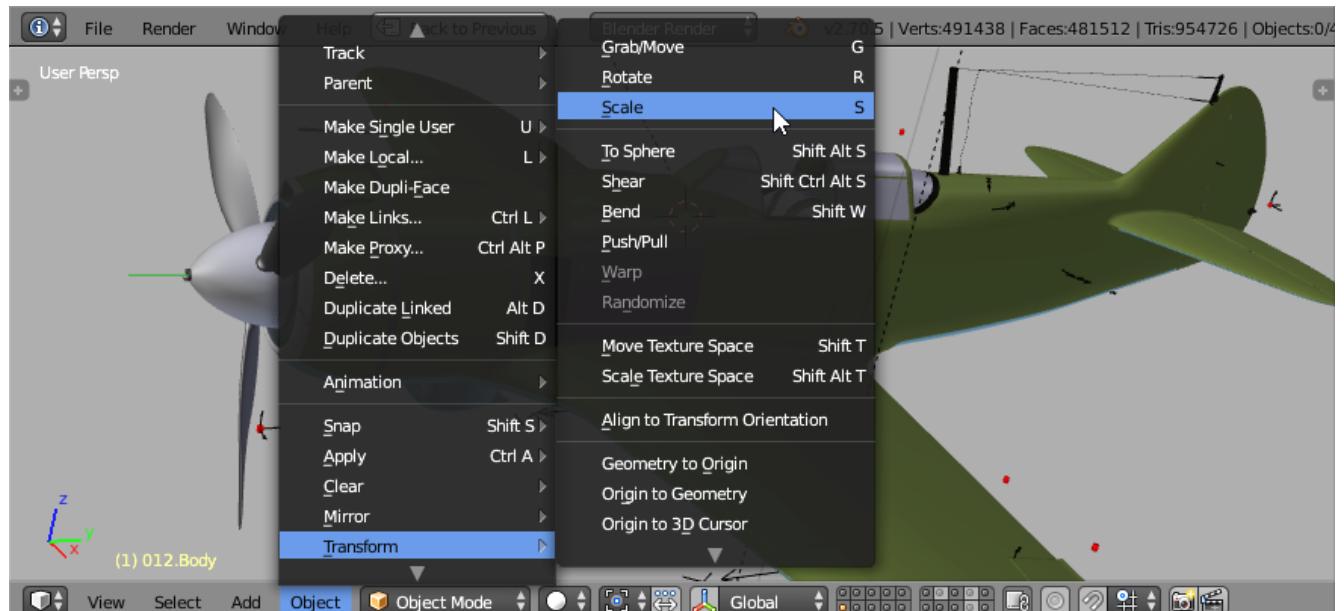


Figure 2.4.17 Transformations available in the **Object** menu

Do not worry about the operations which have unfamiliar names. We will not need many of these commands to build our model! I will introduce all the tools that I use in the following chapters of this book.

Summary

- Translation – the **G** key (*Grab/Move* — page 51);
- Rotation – the **R** key (*Rotate* — page 55, 56);
- Scaling – the **S** key (*Scale* — page 57);
- Hold down **Ctrl** during transformation, to change object coordinates in the regular, fixed steps (page 53);
- Hold down **Shift** during transformation for the additional precision of the mouse (page 54);
- You can restrict every transformation to a single axis, by pressing the **X**, **Y**, or **Z** key (page 53). It allows you to use these transformations even in the perspective view! Note: when you press the axis letter twice, the movement will be restricted to the local object axis. Pressing such a key for the third time switches Blender back to the free movement on the current view plane.
- You can set the pivot point, used in rotation and scaling, to: **.** = *3D Cursor*, **[** = *Bounding Box* (page 56);
- To cancel current operation – press **Esc** (page 51, 54)
- To *Undo* the last operation – **Ctrl-Z** (page 58).
- To *Redo* the last *Undo* operation – **Shift-Ctrl-Z** (page 58);

2.5 Properties window

At the end of this chapter I would like to show you how to obtain the final effect — a rendered image of the La-5 airplane. To do it, you will need to click some buttons, and switch some options in the *Properties* window. This window is located by default on the right side of the Blender screen (Figure 2.5.1):

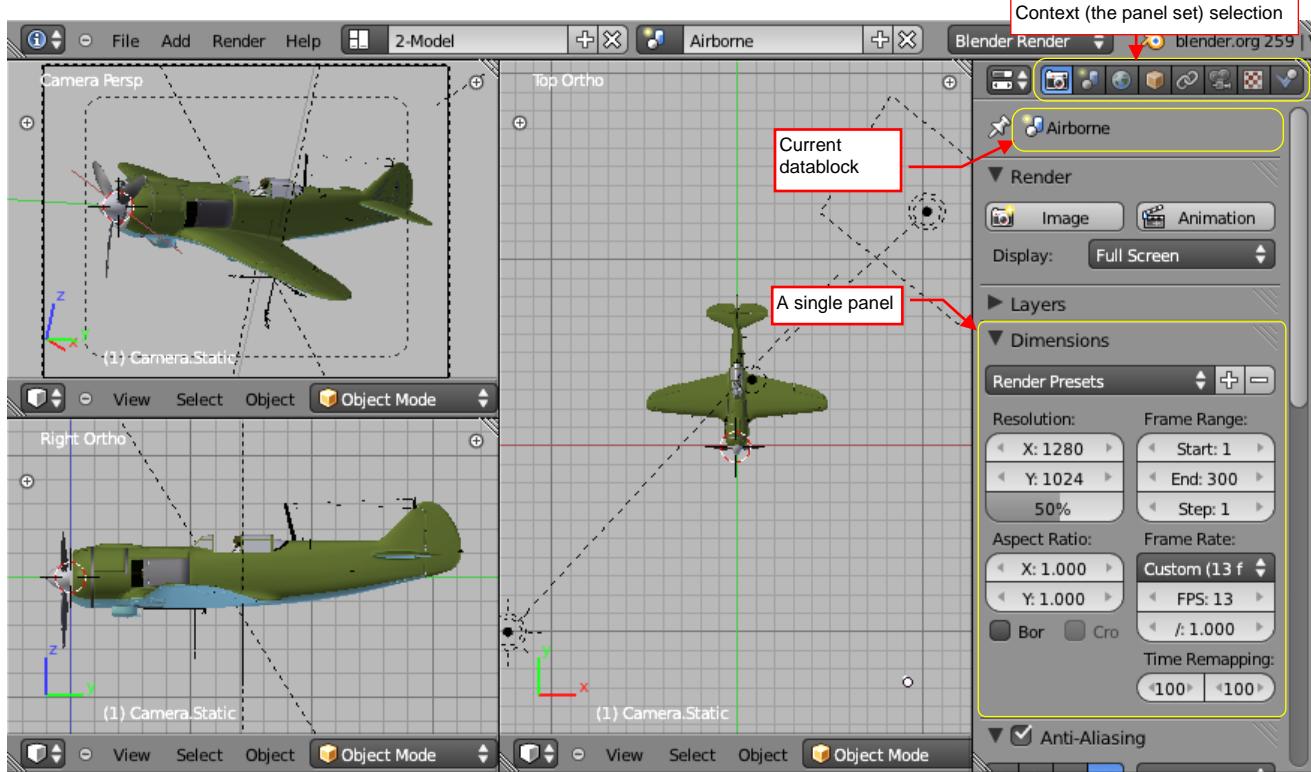


Figure 2.5.1 Main elements of the *Properties* window

In this window Blender exposes properties of the current scene, active object, its material and textures, etc. At first glance, it teems with checkboxes, buttons, and other controls. I will start by explaining its key concepts and elements. In the header of this window you can see the *context* (panel set) selector. Below you can see the *current datablock* (see page 393 to learn what the *datablock* is) — this is the name of the “thing” that owns the properties displayed below. (In the example, shown in Figure 2.5.1, this is the current scene, named **Airborne**). Below you can see *panels* that group the controls by their “subject” (*Render*, *Layers*, *Dimensions*, ...) — Figure 2.5.1).

On the top of each panel there is the header with the label (panel name). On the left side of the header you can see a triangle that folds and unfolds the panel (Figure 2.5.2):

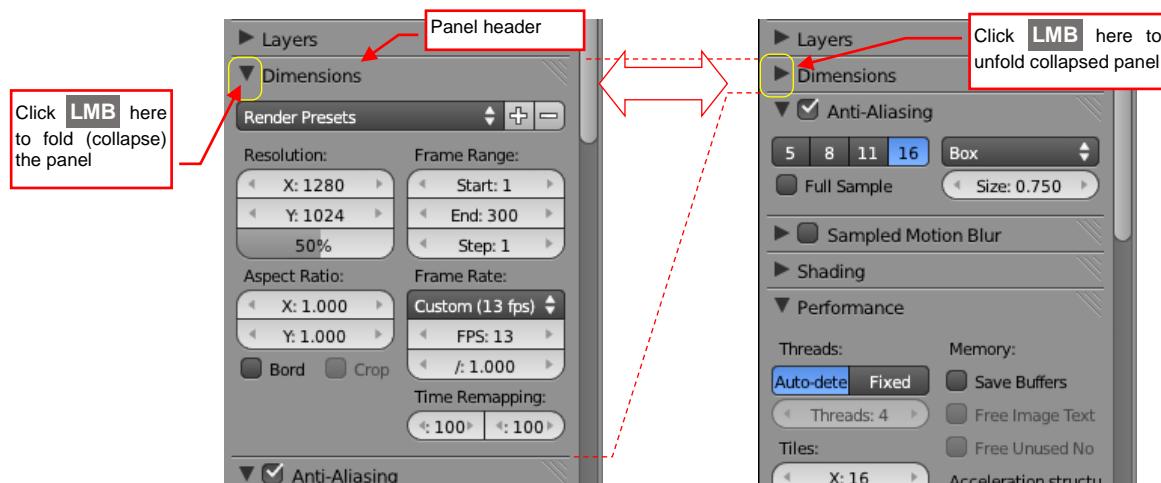


Figure 2.5.2 Folding and unfolding the panel

On the right side of the panel header there is a handle. You can “grab” it with **LMB** and drag this panel into another place (Figure 2.5.5):

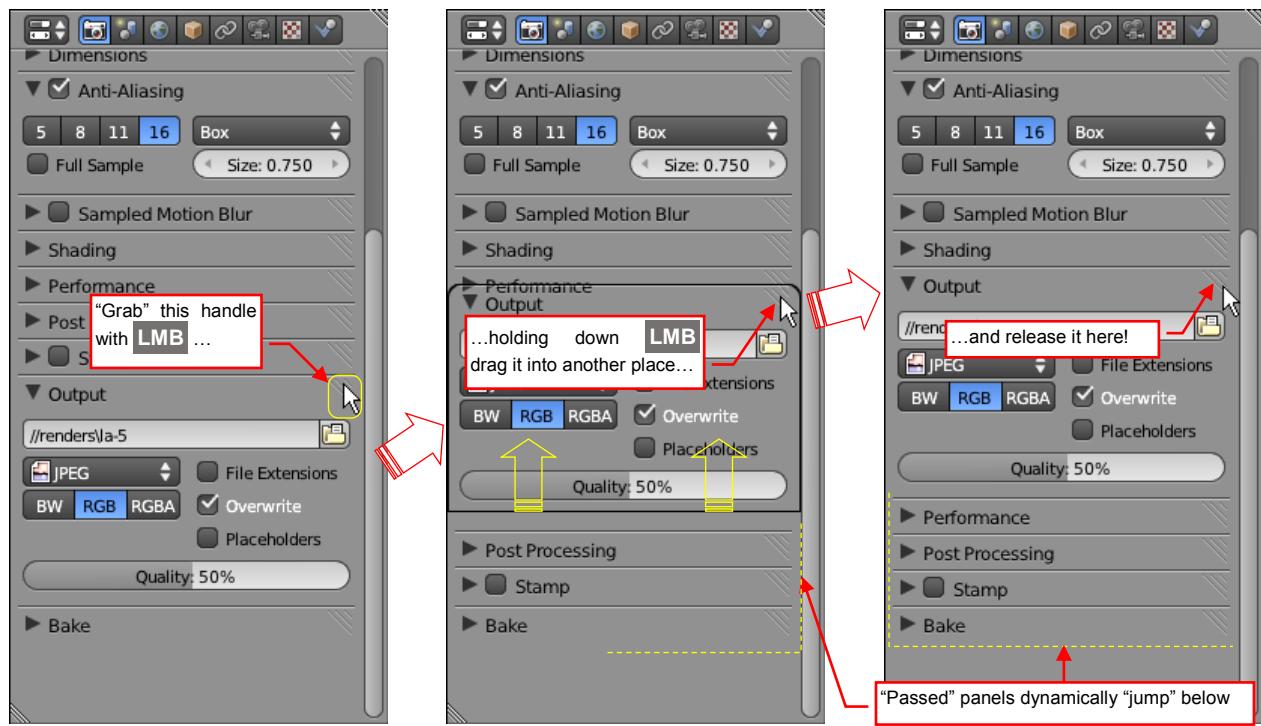


Figure 2.5.3 Moving the panel

The positions of all panels (and in general — the whole state of the user interface) are saved together with the scene state in the Blender file. (When you open a Blender file, everything on its screen will look exactly as in the moment of the last invocation of the [File → Save](#) command. It is very convenient).

In many contexts the panel stack is longer than height of the screen. To reach the panels located below, you have to scroll the content of the [Properties](#) window. Do it with the slider (Figure 2.5.4a), placed at the right edge of this window, or just holding down the **MMB** and dragging the mouse up or down (Figure 2.5.4b):

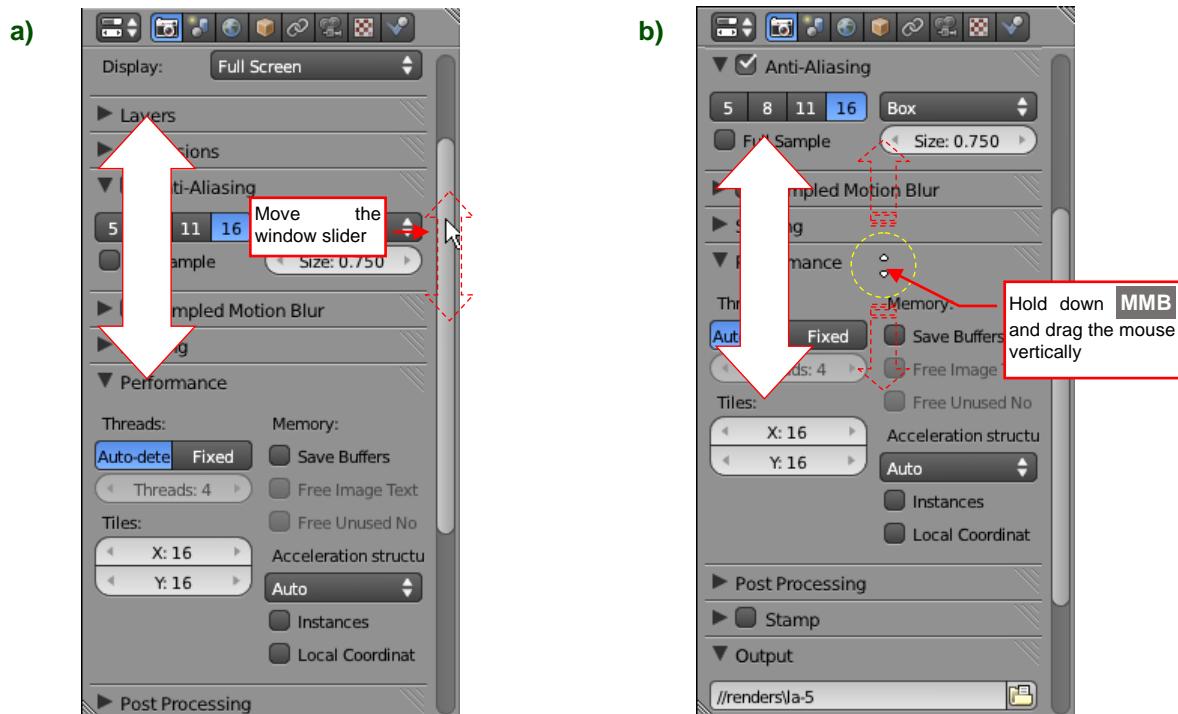


Figure 2.5.4 Two methods of scrolling the contents of the [Properties](#) window

The *Properties* window has so many panels that it is virtually impossible to use them all at the same time (You would have to scroll the window up and down too often). Fortunately you do not have to do it in this way. You need different panels for modeling, others for working with materials and lighting, yet others — to render a single image or animation. That's why the panels in the *Properties* window are divided into several sets (*contexts*), corresponding to these typical operations. Use the icon buttons, located on the window header, to switch between different panel sets (Figure 2.5.5):

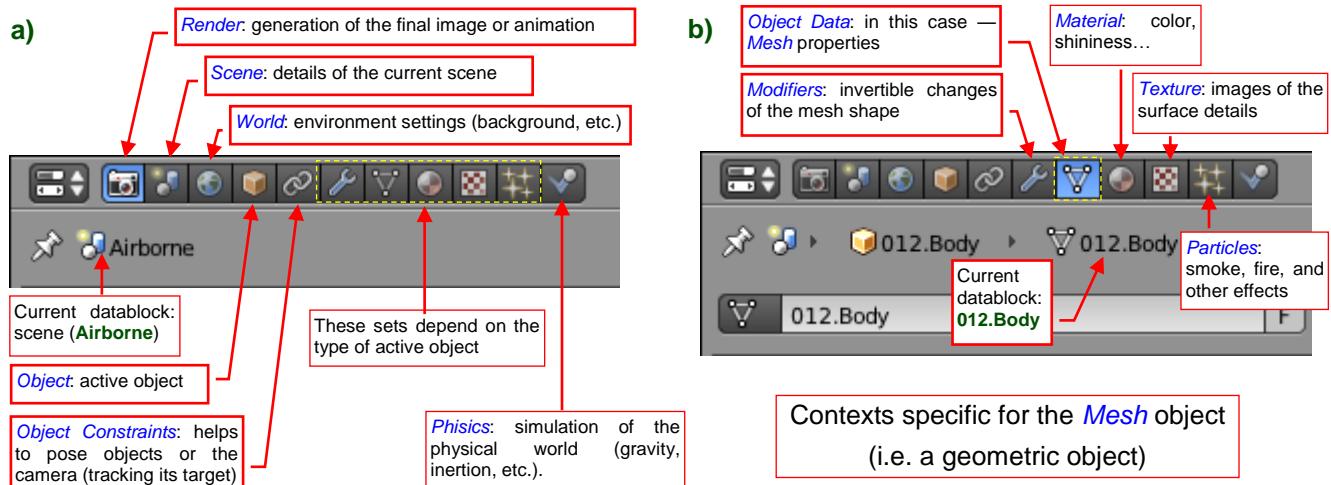


Figure 2.5.5 Typical contexts of the *Properties* window

Figure 2.5.5 shows the typical combination of the panel sets. This is the header of the *Properties* window, displayed for a typical geometric object. Figure 2.5.5a) describes the panel sets that are always present in the *Properties* header, regardless of the active object type. These are the first five sections from the left: *Render*, *Scene*, *World*, *Object*, *Object Constraints*, and one from the right: *Physics*. Figure 2.5.5b) enumerates sections specific for a geometric object. (Such an object contains so called *mesh* — a datablock representing its surface, built of faces, edges, and vertices).

By the way — note, that in Figure 2.5.5a) you can see a different datablock (the **Airborne** scene) than in Figure 2.5.5b). This is despite the fact that the active object on this scene was the same in both cases. (It was the **012.Body** object). Try to select some other contexts, and you can see further differences (Figure 2.5.6):

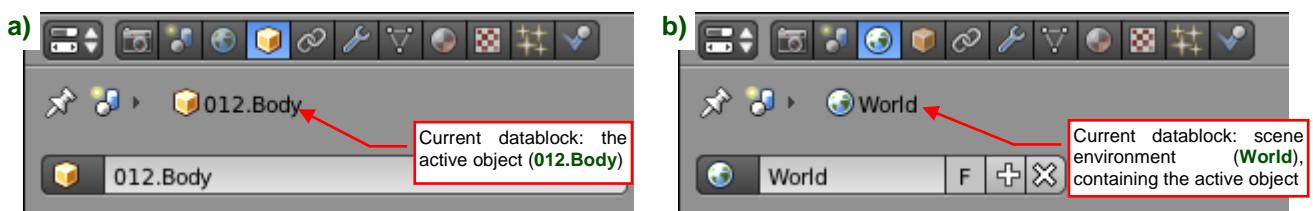


Figure 2.5.6 Current datablock depends on the selected context

When you select the *Object* context, Blender will display the active object as the current datablock (Figure 2.5.6a) — this time without the mesh (Figure 2.5.5b). When you select the *World* context, it displays the environment datablock (named **World**) that contains the active object. For the same reason when you select the *Render* context, you will see the datablock of the scene, that contains the active object (comp. Figure 2.5.5a).

- The panels in the *Properties* window always show details of the datablock displayed below the window header.

If you want to see properties of a particular datablock, regardless of the current active object selection, use the “pin” to attach it to the header (Figure 2.5.7). Just click this icon to keep it in the *Properties* window all the time. To detach this datablock, “unpin” it in the same way.

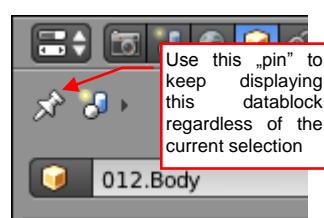


Figure 2.5.7 Datablock “pin”

There are also other types of objects that do not contain the meshes: for example light sources ([Lamps](#)), or cameras. Figure 2.5.8a) shows the panel sets displayed for a [Lamp](#), and Figure 2.5.8b) — for a [Camera](#):

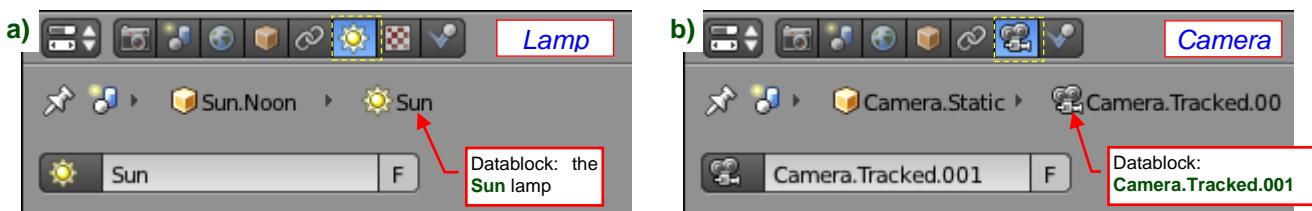


Figure 2.5.8 Various [Object Data](#) icons for different kinds of objects

As you can see, the [Object Data](#) set that has the “triangle” icon for the mesh objects (see Figure 2.5.5b), for the lamp data displays a “sun” (Figure 2.5.8a). The camera icon in Figure 2.5.8b) also speaks for itself. Note that neither the camera nor the lamp have the [Particles](#) and [Material](#) icons. The cameras do not have the [Textures](#) icon, either (because they do not need them).

Table 2.5.1 provides a summary of panel sets which we will use in further chapters of this book:

| Context | Description |
|------------------------------------|--|
| Render | All the render parameters. At the end of this section I will show it in action. |
| Scene | Scene parameters, like the active camera, or an assignment of another scene as the background picture. The Handle Panel add-on that we will use to “pose” the model, also adds its panel to this section. (This Blender plugin controls opening and retracting of the landing gear, flaps, movements of the rudder, elevator, and ailerons, etc.). |
| World | The parameters of the scene environment (for example the color or texture placed on the background, the intensity of the global, dispersed light, etc.) |
| Object | Active object properties: current location, orientation, name, and many others (for example: assigned layers). |
| Object Constraints | Manages the object constraints. (You can assign a constraint to any object. For example — the Camera.Tracked is directed automatically on the La-5 model, because it has a Track To constraint. This constraint always directs selected camera axis onto the auxiliary Camera.Target object). |
| Modifiers | This section is displayed for geometric objects only (i.e. the objects that contain the mesh datablocks). You can apply here some additional, “reversible” effects, like smoothing with subdivision surface, or bending along a curve |
| ... Object Data | Properties of the object “content”: the Mesh datablock for the geometric objects, the Lamp parameters for a light source, or details of the Camera projection for a camera, or other datablock types (Curve , for example). |
| Material | General properties of the material datablock, assigned to a mesh or an object (its color and specularity, transparency, and many other parameters) |
| Texture | Managing the material textures. These are raster images, used to recreate minor irregularities in the surface (rivets, screws, panel joints). They are also used to alter locally many other material properties - like the color (camouflage, dirt). This context is used for the Blender renderer only. |
| Particles | We will use this section at the end of the book to create some grass on the airfield. |

Table 2.5.1 Short description of selected panel sets

If such terms as “material” or “texture” are abstract for you at this moment — do not worry, I will explain their meaning in the further chapters. Table 2.5.1 is placed here just to let you know what to do when I write: “In the [Properties](#) window, switch to the [Material](#) set”, or “switch to the [Material](#) context”. (It means the same).

Excuse me if this introduction was a bit lengthy. I wanted to bring your attention to details of panel handling in the *Properties* window. In fact, the generation of the final image or animation in Blender can be reduced to clicking a single button (in a moment I will show you which one). Before I do it, we will discuss some options of the rendering process. On this occasion I will explain briefly some other details of Blender controls.

Switch into the **Render** context. Figure 2.5.9 with this example shows the basic types of GUI controls used in Blender:

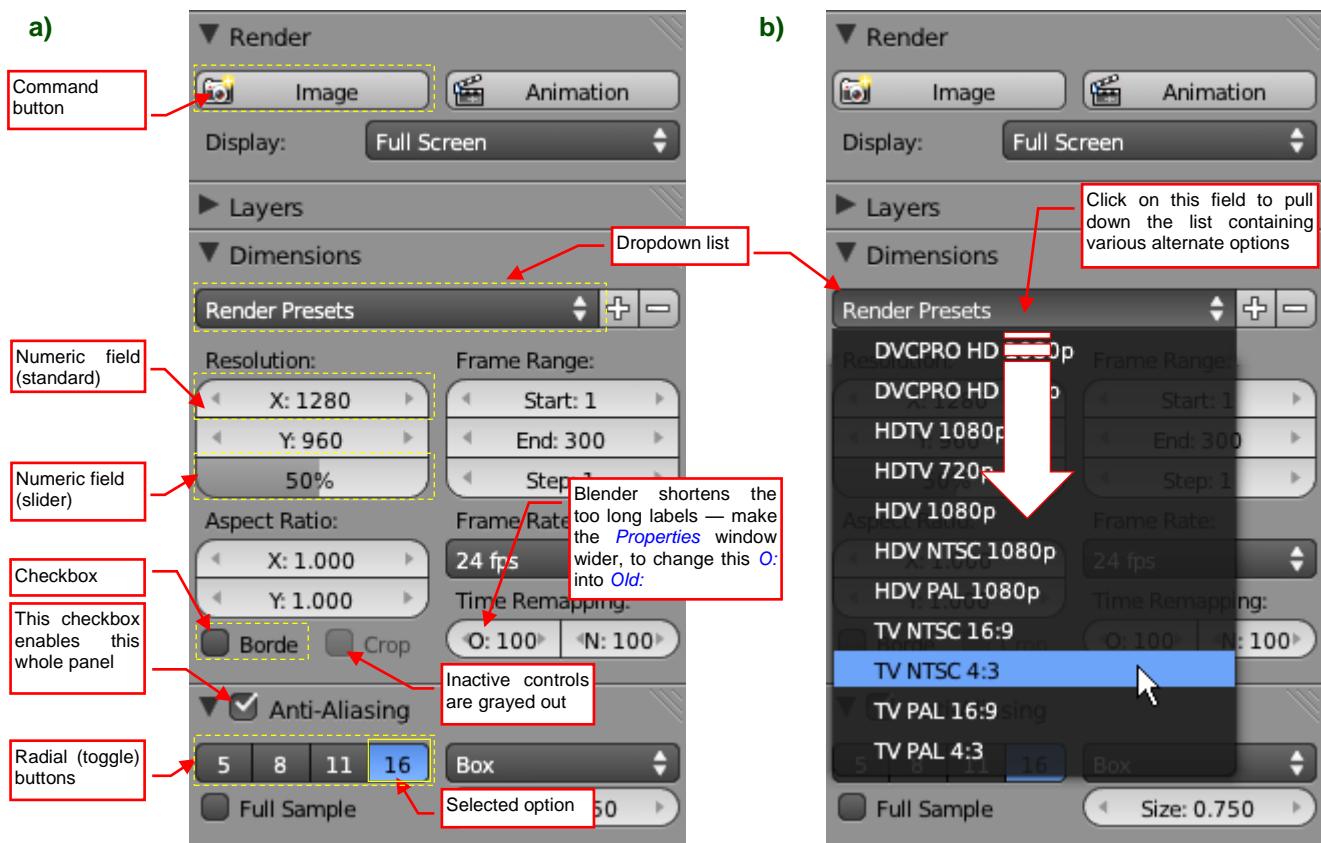


Figure 2.5.9 Basic types of Blender controls

Command buttons (also known as *operation buttons*) are used to invoke a command (like *Render:Image* or *Render:Animation*). Do you know that in the original Blender terminology each control is referred to as some kind of a “button”? The origins of this tradition are lost in the darkness of history. I think that because of this nomenclature the control labels are placed inside the text and numeric fields (see Figure 2.5.9a). You have to adapt to this convention. Fortunately, the checkboxes (referred also as *toggle buttons*) do not adhere to this style¹.

Below the label of the *Dimensions* panel there is a dropdown list (*menu button*) named *Render Presets*. You can click it (with the **LMB**, of course) and select one of the typical render formats (Figure 2.5.9b)². On the other panels you can see a few dropdown lists, used to select one of the possible options (e.g. the list of the *Render:Display* modes). When the list of the alternatives is short enough, Blender uses toggle buttons, instead of a dropdown (for example, to choose the number of samples in the *Anti-Aliasing* panel).

¹ However their current shape is a relatively new modification, which appeared in Blender 2.5. In the previous versions they looked like toggle buttons (highlighted when turned on). Thus the role of these controls was not so obvious, especially at first glance you could confuse them with command buttons. The current checkboxes are more readable for new users, because they comply with the convention of popular windowing systems.

² Notice that when you select an item from the *Render Presets* list, the content of the *Dimensions* panel will change. The *Resolution*, *Aspect Ratio*, *Frame Range*, *Frame Rate* fields will be updated according the settings from the selected item. You can also use the **[+]** button (on the right side of this dropdown) to add a new item to this list. It will copy the parameters currently displayed in the panel. (I have not used it).

Sometimes the dropdown can have several columns. I will show it on another example. Scroll the panels of the **Render** section slightly upward to reveal the **Output** panel, placed at the end of this panel set. It contains a dropdown list where you can set the default file format for the rendered image (Figure 2.5.10):

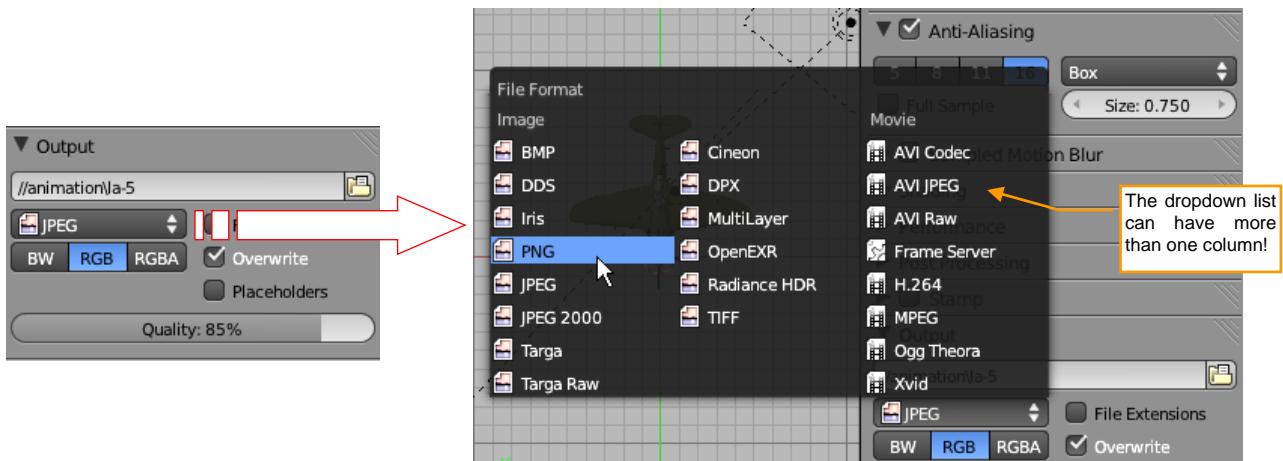


Figure 2.5.10 Selection of the default raster image file format — an example of multi-column dropdown list

- The raster file format, set here, applies to all raster images, generated by Blender. Even the simple “dump” of the current Blender screen (**Ctrl**-**F3**) is saved in the format specified by this field!

When you select the **Jpeg** file format, you should also set the value specified by its **Quality** field (Figure 2.5.10). The **Jpeg** format is one of the so-called destructive compression formats, because it reduces the file size at the price of image quality. Using the **Quality** value (expressed as a percentage) you can control the compromise between the file size and quality of the image. When **Quality** = 100%, there is no image degradation (each pixel is stored separately). Such a file will have the largest size, of course. Personally, I advise to set the **Quality** value to 85%-95%. Such a setting creates relatively small file that still contains a picture of a decent quality.

How to enter the new **Quality** value? Probably it is obvious for the most readers, but I will describe it here in a few words. Perhaps you may learn some useful details this way.

The **Output:Quality** field is a *slider*, i.e. its values are restricted to a specific range (between 0% and 100%, in this case). Blender fills the slider with two colors, where the border of these colors corresponds to the current value. You can change the slider value dragging the mouse over it (with **LMB** pressed — Figure 2.5.11):

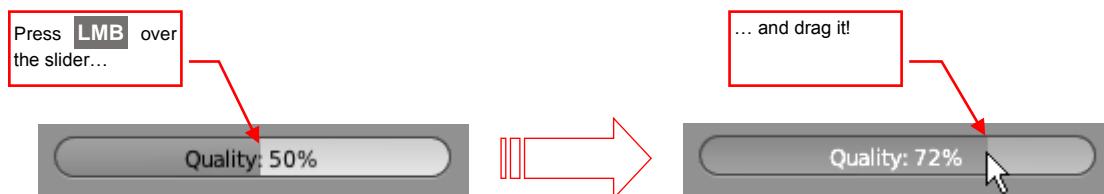


Figure 2.5.11 Changing the value of a numerical field with the mouse

Of course, you can also change this field by typing an exact value. Just click it once (with the **LMB**) or press the **Enter** key, to let it receive the keyboard focus. Then you can type a number (Figure 2.5.12):



Figure 2.5.12 Changing the value of a numerical field by typing

What is interesting — you can also type an arithmetic expression, like “100 / 3”, and even something with brackets! It works like a calculator, substituting the result as the new value of this numeric field.

More often encountered is the version with “arrows”. Such a field may have unrestricted range of its values. In the **Dimensions:Resolution** area you can find a group of three numeric fields, controlling the size of the rendered image. The two upper ones — **X** and **Y** — are the standard “arrows” fields, determining the width and height of the result image (in pixels)¹ (Figure 2.5.13):

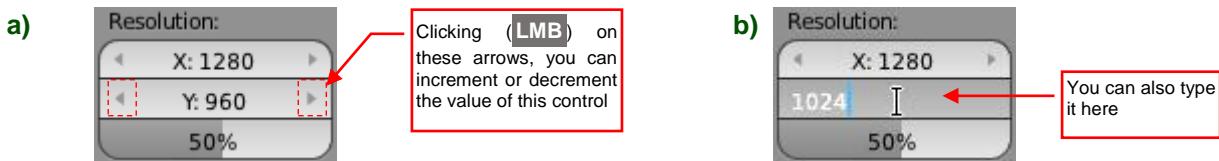


Figure 2.5.13 Two methods of changing the value of a numeric field

Clicking **LMB** on the “arrows” on the left and right sides of a field, you can increment and decrement its value (Figure 2.5.13a). Each click changes it by a fixed step (in this case — by 1 pixel). It is a good method for small corrections, but if this box has to contain a completely different value — you better type it directly (as shown in Figure 2.5.13b and Figure 2.5.12).

- To cancel current edit operation (in the situation as in Figure 2.5.13b), press the **Esc** key.

You can use each numeric field as a slider, by dragging the mouse (holding **LMB** down) over its area (see Figure 2.5.11). If you want to change it incrementally, by the fixed steps — hold down the **Ctrl** key. (This is the same convention as for the object transformations — see page 53, Figure 2.4.7). You can also use the mouse to change simultaneously two or more related numeric fields (usually they are coordinates — Figure 2.5.14):

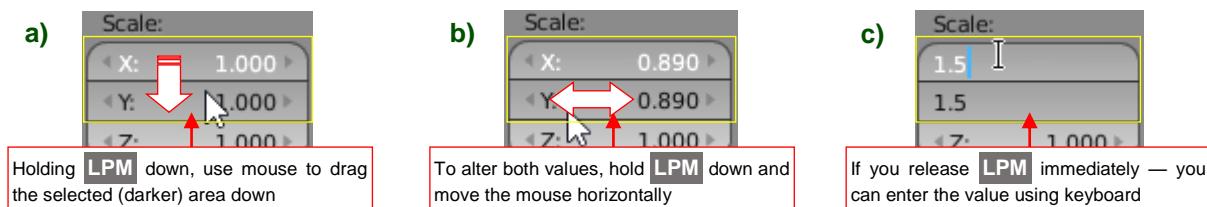


Figure 2.5.14 Changing simultaneously two related numeric fields

Let's return for a moment to the **Render:Output** panel, because it contains yet another type of the input field: a textbox (Figure 2.5.15). You can enter a new value into a textbox in the same way as we did for the numeric field — click it with **LMB**, type the new text, and then press the **Enter**. The typical text editor operations are also available here: you can select and delete any fragment of the text, you can also copy it (**Ctrl-C**) or paste (**Ctrl-V**) from the clipboard.

Note that the text control in Figure 2.5.15 is an example of a “specialized” field to select the file paths. On its right side there is a little open folder icon. When you click it, you will open the file selection window (similar to that one described on page 256). You can select a directory in that window. Of course, you can also simply type the entire path into this field. Note also that it starts with double slash (“//”) by default. This is so-called “relative path marker”. Reading this entry, Blender replaces “//” with its initial path — usually the folder that contains the currently opened file.

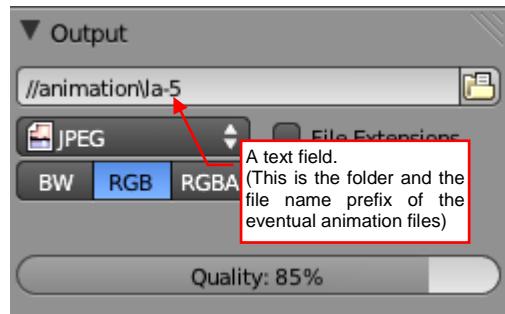


Figure 2.5.15 Example of a textbox (it is a specialized one, for the file or folder path)

¹ The slider located below **X** and **Y** fields allows you to get smaller render version. It is very useful for testing. The rendering time is proportional to the number of pixels of the rendered image. To save your time, test the renders on the images of 50% or 75% of the final size.

The relative path allows you to easily move the folder with the Blender model and all associated files from one disk location into another. With the consistent use of the tag “`//`” at the beginning of each referenced file path, there is no risk that Blender will have problems with finding any of the related files (texture images, for example) in such a new place.

I think that this will end our introduction to the *Properties* window and the Blender GUI controls. (Some of its panels contain also list controls — but I decided to describe such a complex GUI item in a separate section, which you can find on page 288). During the course of this introduction we have changed the default output image format to *Jpeg* (see page 65, Figure 2.5.10). We have also increased (in the *Dimensions* panel) the height of the rendered image to 1024px (see page 66, Figure 2.5.13). Have you noticed that this immediately changes the proportions of the camera frame, in the *3D View*?

- In Blender there is almost no “OK.” nor “Accept” buttons. Everything what you set in its *Properties* window or on the shelves (*Tool Shelf*, *Property Shelf*), is immediately reproduced on the scene.

In principle, such an instant effect is very convenient. It is based on the assumption that the screen refresh time (of all Blender windows) is very short. This assumption may be false, when your computer does not have any special graphics accelerator. When the CPU has to draw hundreds thousands of faces of such a model like this La-5 in the three different 3D windows, you will observe “resistance” of the slider you are moving. (The values displayed in the control will not change immediately). This is because any change of the GUI control forces Blender to redraw this model in every *3D View* window! This is not a big obstacle, but you can reduce these lags by minimizing the number of displayed faces. For example — you can work only on the necessary layers, hiding all the others (see page 38, Figure 2.2.4).

The updated *Render* settings, are ready for generation of the final picture of this La-5. Before you click the final button, make sure that at least the following layers:  are turned on. (You can check it also on the *3D View* window header. If you forget to turn layer 11 on, you will not see the wing nor the fuselage).

- Blender always renders contents of the visible layers only!

Finally — let's do it! Click the *Render:Image* button (Figure 2.5.16). (You can also use its keyboard shortcut — the `F12` key). Blender will display on the whole screen a new *UV/Image Editor* window. (It works this way, when you set the *Render:Display* dropdown to *Full Screen* mode — see Figure 2.5.16. Feel free to check the other possibilities from this list!). At the beginning the displayed image is completely black, then something appears (Figure 2.5.17):



Figure 2.5.16 Starting the rendering

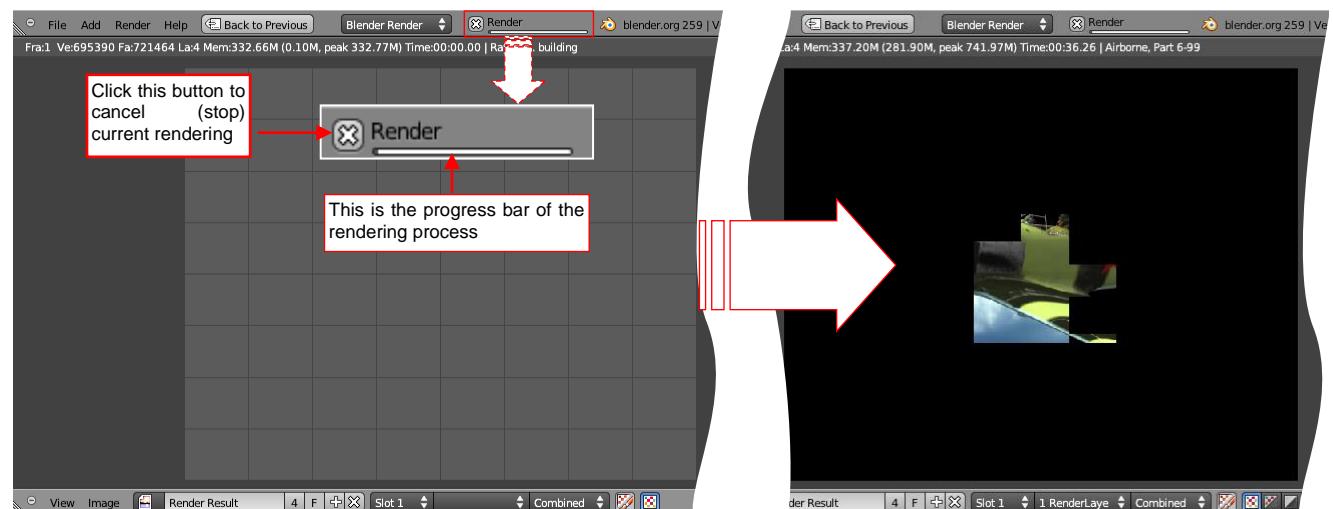


Figure 2.5.17 The *UV/Image Editor* window during rendering

Blender renders the image in a non-blocking manner (more precisely: it is calculated by separate program threads). This allows its screen to respond to user commands. For example — you can stop this process at any time using the button located at the top of the screen, in the *Info* window header (Figure 2.5.17).

Finally you should get a result like in Figure 2.5.18:

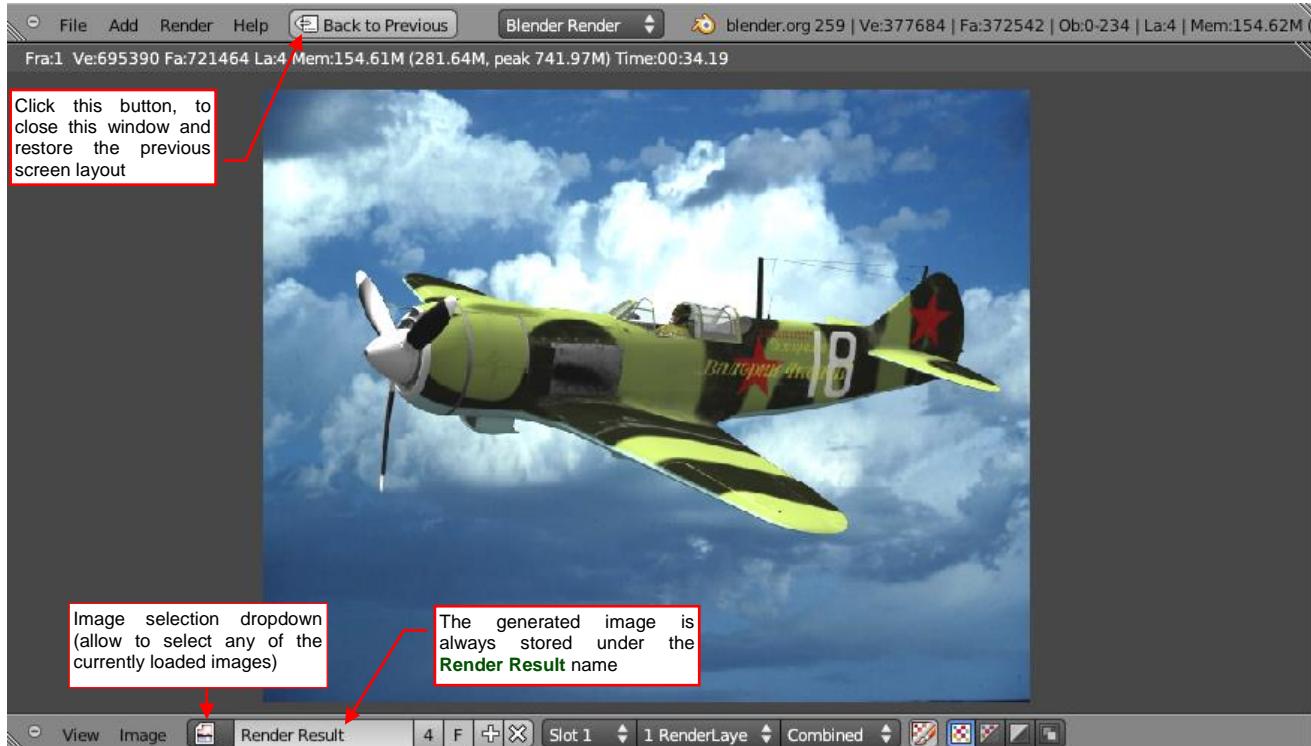


Figure 2.5.18 Image window after rendering

To save the rendered image on the disk – press the **F3** key (or invoke the *Image* → *Save a Copy* command). It opens the file save window (see page 259). There you can determine its name and location. (In this window you can also change the default file format — from the **Jpeg** to **PNG**, for example).

If you want to get the motion blur effect of rotating propeller blades — turn on in the *Render* panel the **Motion Blur** option (Figure 2.5.19a). Then restart the rendering and patiently wait for the ultimate result:

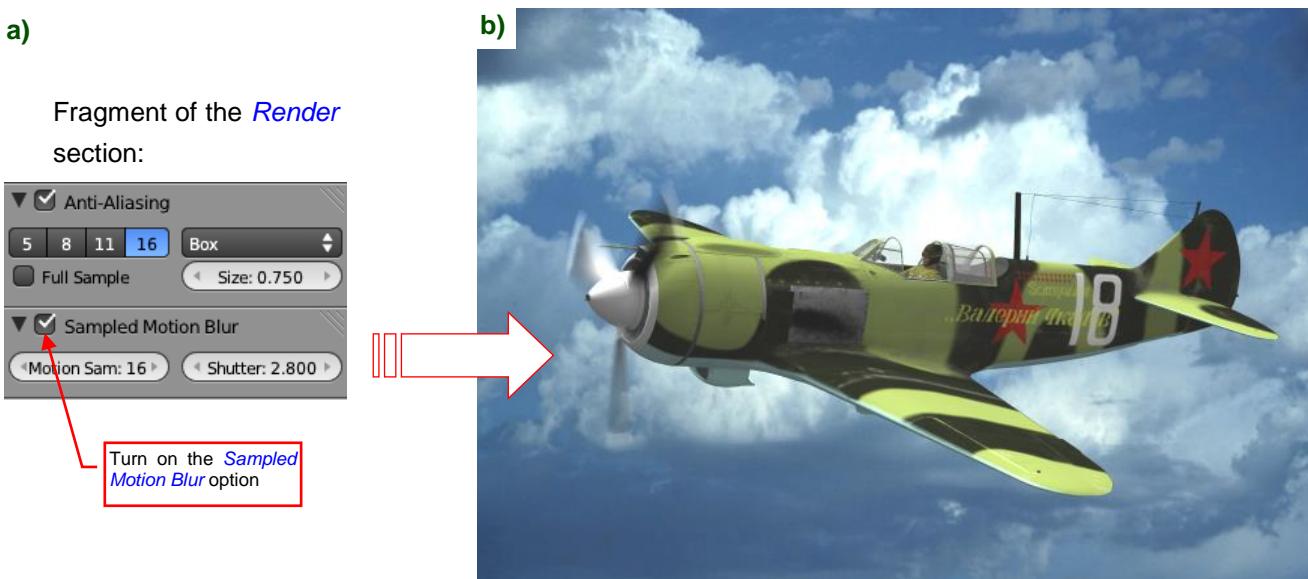


Figure 2.5.19 The same image, but with blurred propeller blades

Why I have mentioned patience? Because in the case of this Blender Renderer engine motion blur effect has been obtained by the simplest, but most computationally expensive method. The *Sampled Motion Blur* renders as many images as you have entered in the *Sampled Motion Blur:Motion Samples* field (in this case it is 16 — see Figure 2.5.19a). Each of them corresponds to the subsequent animation frame. Then they are joined (averaged), producing the motion blur effect.

My computer renders a single image of this La-5 (like in Figure 2.5.18) in 34s. When I turn on the *Sampled Motion Blur* option, it renders the final image for 16 x 34s, that is 9 minutes. This period increases in proportion to the number of pixels (i.e. the image surface), and is longer for the larger images. A better method of obtaining the motion blur effect for the spinning propeller is described in the chapter on the composition, at the end of this book (see Volume IV).

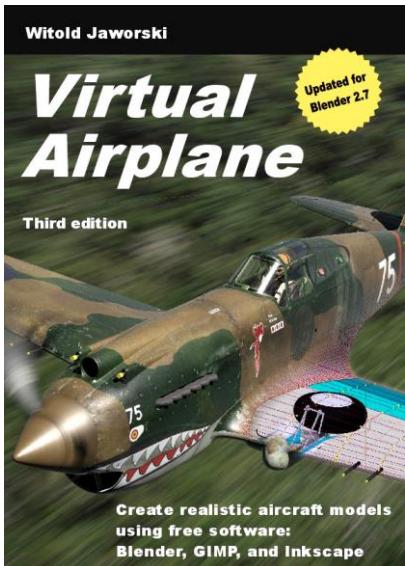
Summary

- Controls in the *Properties* window are grouped into *panels* (page 60) — as in the window panes (toolbox-es);
- *Panels* are grouped into *contexts* — the thematic sets, prepared for different types of scene data (page 60, 63);
- You can select the current *context* using the toggle buttons, located on the *Properties* window header (page 60, 62);
- Blender displays only those context toggle buttons, that are appropriate for the current active object (page 62);
- You can shift the content of the *Properties* window up and down (page 61). This way each context is not restricted by the screen height and can contain more panels. Some of them are hidden “below” the window edge;
- You can move the individual panels (i.e. change their order in the panel set — page 61) and fold them (page 60). These settings are stored individually in each Blender file;
- The basic Blender controls types are the command button (*operation button* — page 64), dropdown list (*menu button* — page 64, 65), checkbox and radio buttons (or *toggle buttons* — page 64), numeric field (*numeric button* — page 65, 66), and text input field (*text button* — page 66);
- Everything you change in the *Properties* window is immediately visible in the model or scene. Thus there are no *OK* or *Accept* buttons in Blender, as seen in the dialog boxes of other programs;
- To render the scene, click the *Render:Image* button (page 67);
- You can save the currently rendered image using the *Image→Save* command (shortcut: **F3** — page 68);
- Special effects, like motion blur, may require significantly longer rendering time (page 68);

This is end of this excerpt. If you want to learn more on:

- how to create/modify such a model;
- how to apply a different camouflage;
- how to use it to create a stunning historical scene;

see the full edition of this book:



“Virtual Airplane” describes how to create the aircraft shown on its cover. It starts with the very basics, and gradually introduces the Reader into the complete workflow.

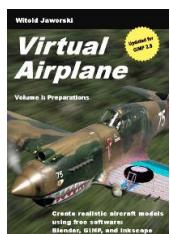
This guide is probably the most comprehensive book on this subject. It contains 1339 pages, 2440 illustrations. “Virtual Airplane” is available in the most popular e-book formats:

- PDF;
- EPUB (for iPads);
- MOBI (for Amazon Fire and Android tablets);

You can find more about this book on:

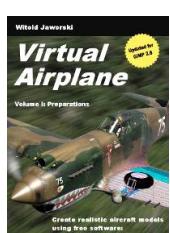
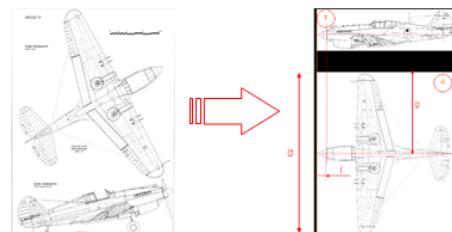
<http://airplanes3d.net>

The complete guide was so thick that it is also available as four separate volumes. You can pick from them the ones you are most interested in:



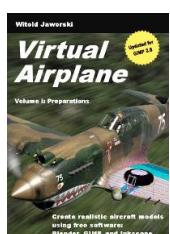
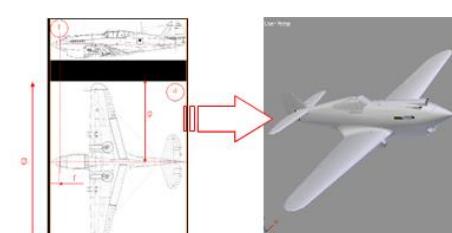
I Preparations (pages: 168, illustrations: 227)

Preparation of reference drawings, required to create the model. It utilizes GIMP and Inkscape (in this part we do not use Blender).



II Modeling (pages 444, illustrations: 771)

Describes how to form the model, using the previously prepared plans and photographs. It begins from a general introduction to this program. Then we will start to model the easiest parts, gradually introducing more advanced techniques.



III Materials and Textures (pages: 636, illustrations: 1110)

Preparation of basic materials and their textures, and arrangement of a typical flight scene. It uses the new Cycles rendering engine.



IV Detailing and Rendering (pages: 523, illustrations: 1000)

This volume discusses creation of various aircraft parts, especially of the cockpit and undercarriage. At the end it shows the composition of exemplary scenes in the flight and on the ground.



In the 20th century you could make an aircraft model from paper or plastic. In the 21st century the time has come for yet another variation of this hobby: computer models. The birth of this new branch of scale modeling passed unnoticed. Few people could spend several thousand dollars of their private money for buying a complete 3D design environment just for amusement. During the first years of this century the only alternative was to use an “inappropriate” software — nothing to boast about... So the first enthusiasts were quietly sitting at their monitors, creating their models and learning from own mistakes.

The times are changing, and now all the necessary software is available for free (it is GPL-licensed). So if you bought your computer less than 6 years ago, you have everything you need to get started! This book provides the essential know-how. I will show you step by step, how to make such models as the P-40 shown on this cover.

“Virtual Airplane” was written not only for the scale modelers. It may also be useful to all who want to learn the three most popular Open Source graphic programs: Blender, GIMP, and Inkscape.



Visit www.airplanes3d.net to ask a question or obtain free trials!

ISBN: - - - -



A P-40B from the 3rd squadron of AVG („Hell's Angels“), Kunming 1942. (Background photo: © Tomo Yun, www.yunphoto.net)