

University of Tallinn

Institute of Informatics

# **Open source 3D Modeling suite Blender, in comparison to proprietary applications**

**Seminar Research Paper**

Author: Anton Zolotarjov

---

Supervisor: Andrus Rinde

---



## Table of Contents

Introduction.....	2
1. 3D modeling.....	2
1.1. Basic requirements for a 3D modeling application.....	3
2. Blender.....	4
2.1. Historical background.....	4
2.2. Getting and installing Blender.....	5
2.2.1.1. Windows.....	5
2.2.1.2. Macintosh OS X.....	6
2.2.1.3. Linux.....	6
3. The interface.....	7
3.1. Navigating the 3D space.....	8
3.2. The Toolbox.....	9
3.3. The buttons window.....	10
3.4. Headers and basic keyboard shortcuts.....	11
4. Object Manipulation.....	12
4.1. Object selection.....	12
4.2. Removing and adding objects.....	12
4.2.1. Removing.....	12
4.2.2. The 3D cursor.....	13
4.2.3. Adding objects.....	13
4.3. Manipulating objects.....	14
5. Edit mode.....	15
5.1. Creating a Table.....	16
6. Creating lathed objects in Blender.....	19
6.1. Creating an outline of a lathed object.....	19
6.2. Spinning the outline.....	20
7. Sculpting.....	21
7.1. Multires.....	21
7.2. Sculpting Panel.....	22
7.3. Brushes.....	22
Conclusion.....	25
Used Literature.....	26

## Introduction

This topic was chosen because the author has often done some kind of modeling with the open-source 3D modeling and animation suite Blender. It would be beneficial to explore in what areas and how Blender differs from proprietary applications in the same field.

The purpose of this paper is to make a short introduction to the open source 3D-modeling program Blender and contrast it with the most common commercial applications for 3D-modeling. The focus is mainly on such aspects as the interface of the program, creating simple 3D objects and working with meshes<sup>†</sup>. Blender is quite a complex program, and covering the whole of it would require a substantial amount of space, close to a book in size. The first chapter of this paper deals the basics of 3D modeling, moving on to the historical background of Blender, installation and then on to how to use Blender for 3D modeling. At each stage, a comparison will be made to some proprietary applications, outlining how Blender is different in that field, for better or for worse. In the end, a conclusion will be drawn, detailing where Blender is superior and why.

## 1. 3D modeling

*What 3D models are and what are the usual means of creating them.*

3D models, in general, represent a 3D object using a collection of points in 3D space, connected by various geometric entities such as triangles, lines, curved surfaces, etc. Early attempts at 3D graphics started at defining models using mathematical functions and formulas. While it is entirely possible to write code or mathematical expressions to create a 3D model nowadays (*a fine example of that is POV-Ray*), it is usually done visually, using applications to simplify the process by providing visual feedback. A sort of “What you see is what you get” 3D modeling.

There are two approaches to 3D modeling: polygonal modeling and NURBS modeling.

**Polygonal** modeling - Points in 3D space, called vertices, are connected by line segments to form a polygonal mesh. Like a triangle, or a square, only in a 3-dimensional space. Used for example by 3D Studio Max. The vast majority of 3D models today are built as textured polygonal models, because they are flexible and because computers can render them so quickly. However, polygons are planar and can only approximate curved surfaces using many polygons. (Which, in turn, leads to loss of performance)

---

<sup>†</sup> A mesh is a collection of vertices, edges and faces that defines the shape of an object in 3D graphics.

**NURBS\*** and **Bézier†** modeling - NURBS Surfaces are defined by spline curves, which are influenced by weighted control points. The curve follows (but does not necessarily interpolate) the points. Increasing the weight for a point will pull the curve closer to that point. NURBS are truly smooth surfaces, not approximations using small flat surfaces, and so are particularly suitable for organic modeling. Maya is the most well-known commercial software that uses NURBS natively. (Харовач, 2001) Bézier curves are a different type of curves, but serve the same purpose. A good example of Bézier curves can be found in the application CorelDraw, even though it is 2-dimensional. NURBS and Bézier curves are used to create **lathed** objects, that is an object whose vertex geometry is produced by rotating the points of a curve around a fixed axis.

Blender, like many other computer modeling applications, allows 3D models to be created either by working with points, called vertices; edges; surfaces or curves. A combination of all of these methods is usually required in order to achieve high-quality images.

In addition to defining the structure of a 3-dimensional model, it may have a texture. A texture is a static (or moving) picture wrapped around the model. Blender provides a comprehensive texturing solution as well as different lighting methods.

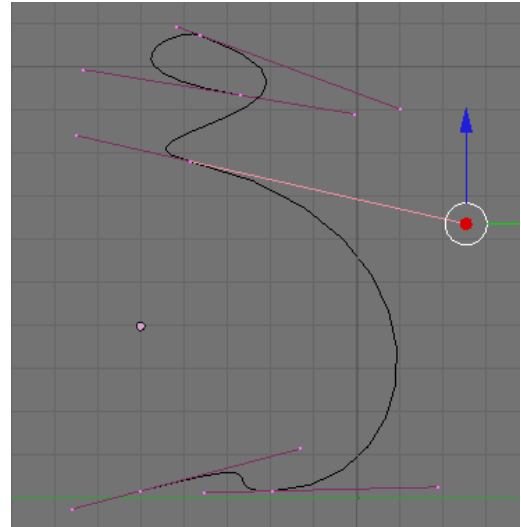
### 1.1. Basic requirements for a 3D modeling application

A conventional 3D modeling application thus needs to have the following features:

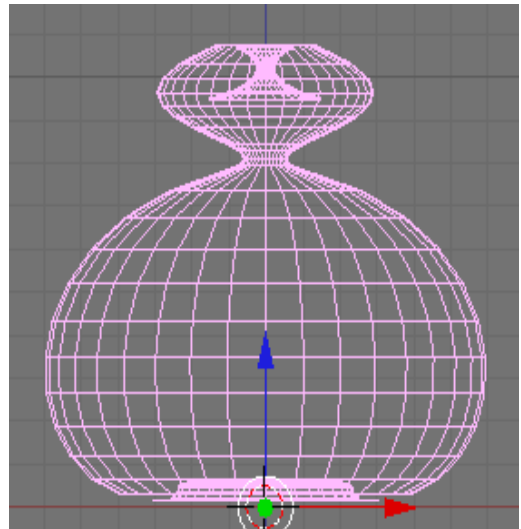
- Creating primitive models, such as cubes, planes, cylinders, spheres and cones.
- Creating lathed objects using curves or other methods.

\* **Non-uniform rational** B-spline (**NURBS**) is a mathematical model commonly used for generating and representing curves and surfaces.

† Bézier curve – named after the French engineer Pierre Bézier, who used them to design automobile bodies.



*Illustration 1: Simple Bézier curve.*



*Illustration 2: Object created by **lathing**, or **spinning** the Bézier curve.*

- Provide ways to work with individual vertices, edges and surfaces of an object.
- Execute common boolean operations on objects, such as intersect, union and difference.
- Texture objects.
- Provide lighting to the scene. (Флеминг, 2000)

## 2. Blender

### 2.1. Historical background

Blender was developed as an in-house application by the Dutch animation studio NeoGeo and Not a Number Technologies (NaN). It was primarily authored by Ton Roosendaal, who had previously written a ray tracer called Traces for Amiga in 1989. *Some irregularities and oddities in Blenders interface come from its Amiga roots.*

Initially, the program was distributed as shareware, until NaN went bankrupt in 2002.

Following the bankruptcy, the creditors agreed to release Blender under the terms of the GNU General Public License. Blender is now Free Software and it is being actively developed under the supervision of the Blender Foundation, which is a non-profit organization responsible for its development, as well as making the short films *Elephants Dream* and *Big Buck Bunny*.

The first large professional project in which Blender was used was in *Spider-Man 2*, where it was primarily used to create animatics and pre-visualizations for the storyboard department. ([Animatics for Motion Pictures](#))

The Blender Foundation initially reserved the right to use dual licensing, so that, in addition to GNU GPL\*, Blender would have been available also under the "Blender License", which did not require disclosing source code but required payments to the Blender Foundation. However, this option was never exercised and was suspended indefinitely in 2005. Currently, Blender is solely available under GNU GPL.

Unlike many proprietary 3D-modeling applications, which are developed to satisfy the needs of the users of these applications, Blender was developed to satisfy the needs of its developers, who are also 3D-artists. As the need for new features arose, they were added. That is the primary reason of why the interface of Blender is very different from proprietary 3D-modeling applications.

---

\* GNU GPL – GNU General Public License, a license that requires that the software, and all its derivatives have an open source-code.

## 2.2. Getting and installing Blender

### *System requirements:*

#### *Minimal*

300 MHz CPU

128 MB Ram

20 Mb free hard disk Space

1024 x 768 px Display with 16 bit color<sup>†</sup>

3 Button Mouse

Open GL Graphics Card with 16 MB Ram

#### *Optimal*

2 GHz CPU

2 GB Ram

1920 x 1200 Display with 24 bit color

3 Button Mouse

Open GL Graphics Card with 128 or 256 MB Ram

Blender can be downloaded for free from the Download section of the official Blender website:

<http://www.blender.org>

Blender can run on Windows (98/2000/XP/Vista), Macintosh OS X (both PowerPC and Intel), Linux as well as the more exotic Solaris workstations and FreeBSD.

### 2.2.1. Instructions for different operating systems:

#### 2.2.1.1. Windows

**x86-32:** <http://download.blender.org/release/Blender2.48a/blender-2.48a-windows.exe>

Double-clicking will run an installation wizard that copies Blender to the hard drive and registers Blender's .blend file type with Windows. It will also create a Blender program group in the Start menu and add an icon to the the desktop, if chosen.

To run Blender after the installation, either double-click the shortcut that was added to the desktop, or find “Blender” in the “Blender Foundation” folder that has been added to the Start menu.

---

<sup>†</sup> The author has been able to successfully use Blender on the first generation Asus EEE PC, with screen resolution of only 800 x 480

### 2.2.1.2. Macintosh OS X

<http://download.blender.org/release/Blender2.48a/blender-2.48a-OSX-10.4-py2.3-intel.zip>

This version of Blender works with OS X version 10.4 or later.

When copied to the desktop and decompressed, these files create a folder with everything that is needed inside. This folder may be moved into the Applications folder. To run Blender, double-click on the Blender icon inside the folder. No installation procedure is needed.

### 2.2.1.3. Linux

The Linux version of Blender comes in two versions, one for 32-bit architecture and one for 64-bit.

**x86-32:** <http://download.blender.org/release/Blender2.48a/blender-2.48a-linux-glibc236-py24-i386.tar.bz2>

**x86-64:** [http://download.blender.org/release/Blender2.48a/blender-2.48a-linux-glibc236-py24-x86\\_64.tar.bz2](http://download.blender.org/release/Blender2.48a/blender-2.48a-linux-glibc236-py24-x86_64.tar.bz2)

Both versions use Python 2.4 and require glibc 2.3.6. The archive can be decompressed, and the resulting folder moved to any location in the Home directory.

*An alternate installation for Ubuntu can be found here:*

<http://www.getdeb.net/app/Blender>

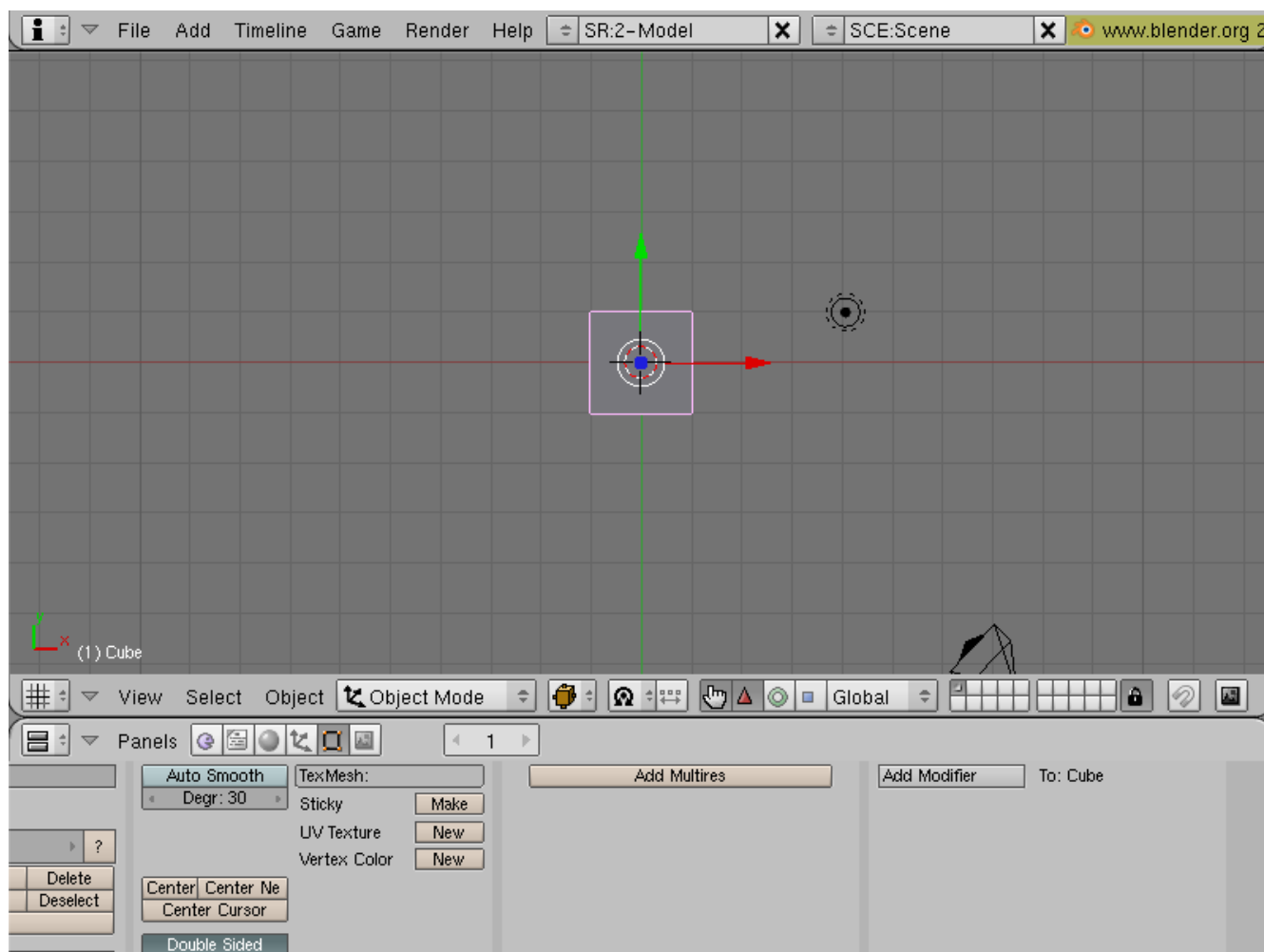
*The files here are .deb packages, which are easier to use for installation, as they contain all the installation data as well as the requirements.*

#### ***Note on the Windows version:***

Mac OS X and Linux versions of Blender start up and fill the screen. Windows version does too, but it has an additional window and item in the taskbar. This windows is required, and it is not to be closed.

*It is perhaps surprising to note that Blender is available for almost any operating system, whereas such dedicated applications as LightWave 3D are only available for Microsoft Windows and Macintosh systems (LightWave 3D product description), and the standard 3DS Max is available only for Microsoft Windows. (3DS Max product description)*

### 3. The interface



*Illustration 3: Default Blender Interface.*

To reiterate what has been said above, Blender's interface is different from any other 3D-modeling application. The interface was designed to meet the needs of its developers, and therefore, not for the average user.

However, Blender was designed by professional 3D-modelers *for* professional 3D-modelers. Many of the things that can make a program simple for beginners end up constraining intermediate users. Thus, the time spent learning the core principles of Blender's interface is a good investment.

*A conventional commercial program, such as 3D Studio Max or LightWave 3D usually has 4 separate viewports by default: left, top, front and perspective as well as a a toolbox to the side of the screen. Blender has only one viewport by default. The conventional setup makes it easier to move, or otherwise transform an object along a certain axis, but Blender has other ways to constrain transformation along axes. It can be configured to have those same 4 viewports, but users can equally manage with just one.*

*Access to the toolbox is also different. The toolbox is always visible in a standard commercial application, but to bring up the toolbox in Blender, the mouse needs to be over the 3D view window and the Spacebar needs to be pressed.*

The most basic Blender interface advice is: *One hand on the mouse. One hand on the keyboard.*

Blender was designed with this in mind. There are controls built in so that they can be done with either the mouse or the keyboard, but in most situations, it is more efficient to work with a mouse/keyboard combination.

***Note on the mouse:***

Blender works best with a three button wheel mouse. It is usable with a two button mouse, or without a scroll wheel, but it is not recommended. Likewise, a touchpad or control-stick solutions on laptop can be used, but an external mouse will be better. If a three button wheel mouse is not available, holding down the Alt key and using a left click will simulate the Middle Mouse Button.

### **3.1. Navigating the 3D space**

*How to rotate the view, switch between orthographic and perspective modes, and how to switch between solid and wireframe modes.*

Place the mouse cursor over the 3D view and roll the scroll wheel back and forth. The view zooms in and out. The scroll wheel, which doubles as the Middle Mouse Button (MMB), **zooms** the 3D view.

Now press the **MMB** and **drag** the mouse. This **rotates** the 3D view freely. By far, this is the most useful and most counter-intuitive navigation control in Blender. Some applications either use the Left Mouse Button to rotate the view, or some other button or combination of buttons.

Using a combination of view rotation (MMB drag) and zooming (scroll wheel), one can get to any view of the objects in the 3D window. These two controls are the most often used for navigation.

Another useful navigation tool is **panning** the view. To pan the view, hold down the **Shift** key and **MMB drag**. The view translates along with the mouse movements.

In addition to mouse-based navigation in 3D space, there are several keyboard shortcuts. In order to quickly switch to **front**, **side** or **top** views, one can use the **1**, **3** and **7** keys on the number pad (Numpad). To view **behind**, **left** and **bottom** views, simply press **Ctrl** and **1**, **3** and **7**.

Z-key changes the view from solid to wireframe. This can be useful if the object on the scene is complex and the user needs to see through it.

Another view toggle is **Numpad-5**. Numpad-5 toggles the 3D view between **orthographic** and

**perspective** modes. Perspective view mode provides an illusion of depth, and it is usually used to position the camera. Orthographic mode's primary use is a precise arrangement of the objects in a scene. All the controls listed above have a different feel in Perspective mode.

By default, new 3D views are in orthographic and solid modes in Blender. (Hess, 2007)

*In LightWave 3D as well as in 3DS Max, the mouse is sufficient for all the navigational needs. The latest version of 3DS Max has exceptionally useful and intuitive navigational tools: the ViewCube and SteeringWheel. They provide visual feedback, how the view is oriented currently; and simplify navigation.*

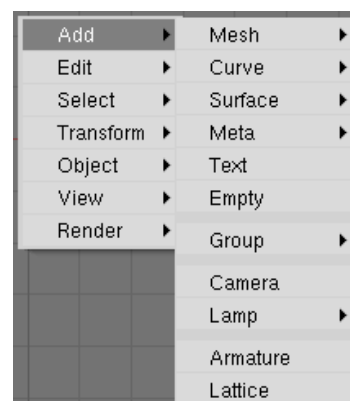
*The need for so many different shortcuts and mouse controls to navigate come from the default user interface. As was already mentioned, the default user interface has only one viewport – therefore, to navigate efficiently without rearranging the interface to the more familiar 4 viewports setup, the user needs all of these controls. After using Blender for over 6 months, the author finds that it has become second nature.*

### 3.2. The Toolbox

When working in 3D view, the toolbox is one of the most important assets. The toolbox is accessed by placing the mouse over the 3D window and pressing the **Spacebar**. The toolbox pops up under the mouse cursor. It functions like any conventional hierarchical menu.

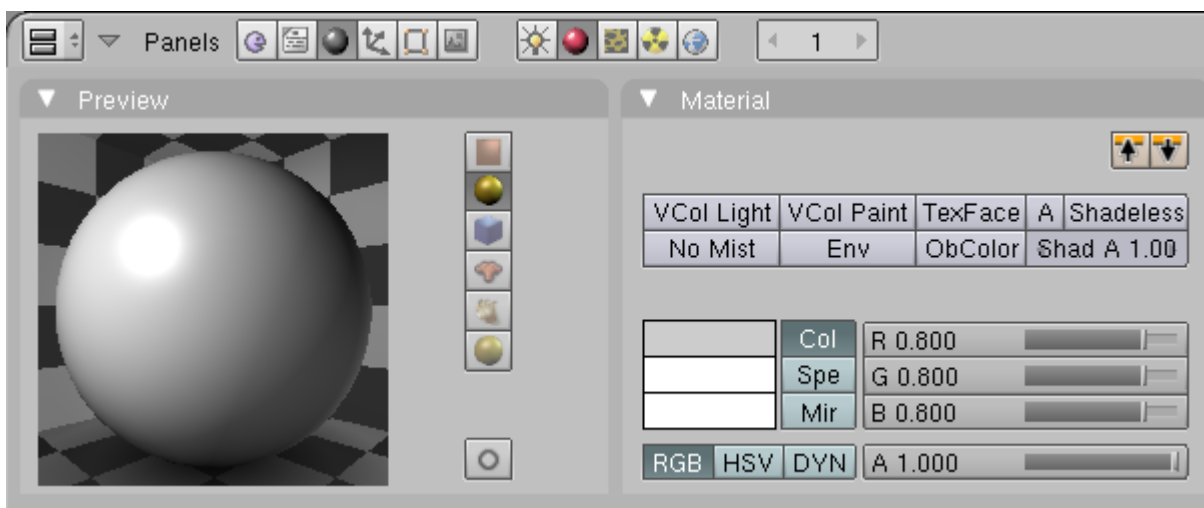
The toolbox contains almost any command or tool one would use in Blender. There are controls there for creating objects of all kinds, moving into different modes, grouping, selecting, rendering, transforming, etc. (Hess, 2007)

*The creators of Blender decided against an always visible toolbox to save space on the screen, allowing them to focus on the actual content of the scene. Instead, the toolbox is brought up on demand when needed.*



*Illustration 4: The Toolbox.*

### 3.3. The buttons window



*Illustration 5: A the Buttons window showing the Materials context.*

A Buttons window shows a series of panels and tabs that are populated by various controls. Depending on which **Context** is selected, these controls can affect objects in the 3D view, set options and properties (like Materials or Lighting), and trigger procedures (like Rendering).

A **Context** is the technical name for the different groupings of controls, selected by either clicking on the appropriate button, or by using the associated hotkey. For the purpose of this paper, the focus will be on the four rightmost context icons: Shading (F5), Object (F7), Editing (F9) and Scene (F10).

For now, press F5 to enter the Shading Context. A new set of icons has appeared to the right of the original context buttons. These are called **Subcontexts**, essentially, gateways to more groups of controls.

Subcontexts function the same way as contexts: they display a different set of panels and tabs with controls. In the case of the Shading Context, there are five subcontexts: Lamps, Materials, Textures, Radiosity and World. (Hess, 2007)

*The buttons window is similar to the one used in proprietary 3D modeling applications properties windows, as it sets different scene and object properties. The only difference here is the name: Subcontext.*

### 3.4. Headers and basic keyboard shortcuts



*Illustration 6: 3D view header.*

Each window in the Blender interface can have a header.

Headers always contain menus, and often other controls relevant to the particular type of window to which they're attached.

Clicking on the File menu shows what one is used to seeing from a file menu in a graphics application: commands for opening, closing, saving, importing and exporting files. Keyboard shortcuts are listed beside the commands. Most of them are different from the ones used in other applications. Because they are different from 99% of the programs, a list of the more popular ones is provided below:

- **New:** shortcut: **Ctrl-X**. This discards whatever work has been done and sets up a fresh work space. It also displays a “OK? Erase all” message that needs to be confirmed by left-clicking. Blender will never ask if the user wants to save the current work with a “Do you want to save your changes?” message.
- **Open:** shortcut: **F1**. Shows the file browser window.
- **Save:** shortcut: **Ctrl-W**. While many applications throughout the world use this shortcut to Close a document, Blender uses it to Save.
- **Quit:** shortcut: **Ctrl-Q**. This will display a “OK? Quit Blender” message that needs to be confirmed by left-clicking.
- **Undo:** shortcut: **Ctrl-Z**. Like in many other programs, this will undo the previous action, be it selecting an object, moving, scaling or deleting an object and many other actions. **Ctrl-Shift-Z** is the Blender equivalent of **Redo**. *Unlike other programs, however, there is no menu entry for Undo or Redo.* (Hess, 2007)
- **Render:** shortcut: **F12**.

***Note to Mac users:***

F12 is most likely set to display the Dashboard by default. To render, go to the Scene buttons, F10, and click the RENDER button. It is a good idea to change the Dashboard shortcut in the Mac's control panel, because F12 is used quite often.

## 4. Object Manipulation

*Object creation, manipulation and organization in Blender.*

Use Ctrl-X to begin a clean session of Blender.

### 4.1. Object selection

Select the default camera by placing the cursor over it and pressing the **Right Mouse Button (RMB)**. This selection method may seem counter-intuitive (many programs use the *Left mouse button* for selection). The pink outline indicates that the object, in this case the camera, is selected.

RMB the cube and you will see that it is now outlined in pink (selected), while the camera has returned to its original state. Selecting **multiple** objects in Blender works like in most other programs – with the **Shift** key. With the cube still selected, hold down the Shift key and RMB select the default lamp. The selection has been extended to two objects: the cube and the lamp.

**A-key** toggles selection on and off for **everything**.

Another useful selection method is **Border Select Mode**. With the cursor over the 3D window, press the **B-key**, and watch the cursor become the target of moving crosshairs. These moving guidelines indicate the Border Select mode is active. **Left Mouse Button (LMB)** click and drag across the 3D window. Any object that falls within the area of the box shape that the dragging creates will become selected when the mouse button is released. Unlike regular, RMB selection, border select is always cumulative and will add to the previous selection. If the user wants to use border select to *deselect* objects, drag with either the MMB or RMB instead of LMB. (Hess, 2007)

*Object selection is different in Blender: it uses the most unintuitive mouse button for that: Right Mouse Button. This is one of the most off-putting factors for new users of Blender. As far as the author knows, there is no way to reassign this control mapping. LightWave 3D and 3DS Max both use the more familiar Left Mouse Button for this task. The effect these controls produce are more or less the same, but again, with different keyboard mappings: selecting everything, and border select or bounding box selection, as it is sometimes known.*

### 4.2. Removing and adding objects.

#### 4.2.1. Removing

Having familiarized ourselves with the basic methods of selection, let us move on to adding and removing objects.

First of all, getting rid of the default cube. Select it any way you prefer, be it RMB selection or Border selection.

Press the **X-key** to **delete** the cube. A confirmation “OK? Erase selected object(s)” prompt will be displayed. Click the LMB to accept this and erase the cube.

### 4.2.2. The 3D cursor

The 'aiming sight' that is seen in the 3D view is the 3D cursor. The 3D cursor is where new objects and items will appear when created in the 3D workspace.

Most new Blender users mistakenly LMB click somewhere in the 3D view, hoping to select something (as was already mentioned, selecting is quite different in Blender). Using the LMB in the 3D view sets the location of the 3D cursor. Before moving on, make sure that the 3D cursor is in the center of the 3D world.

Go into front view (Numpad 1) and LMB as close to the intersection of the blue and red (z and x) axes as possible. The 3D cursor jumps to the position of the click. Now go into side view (Numpad 3) and LMB again at the intersection of the blue and green (z and y) axes, if the 3D cursor is not already there. Use Numpad 1 to return to front view. (Hess, 2007)

*A rather interesting solution for defining the place where new objects are to be added. In LightWave 3D as well as in 3DS Max, the new object is drawn by dragging the mouse on a viewport. In the authors opinion, Blenders solution is more elegant, as it saves time. Instead of having to draw the new object to add it to the scene, the user only has to position the 3D cursor where he wants to add a new object.*

### 4.2.3. Adding objects

With the cursor over the 3D window, bring up the toolbox by pressing the Spacebar.

As can be seen, there are many different kinds of objects that can be added to the 3D view, but for the purpose of this paper, we're going to focus on the Mesh category. From within the Add → Mesh section of the toolbox, choose Icosphere, then click OK when a pop-up says “Subdivision: 2”. A mesh sphere object has been added to the 3D world, and it has been created at the location of the 3D cursor.

To get a little more practice, try adding a few more icospheres. Remember that new objects are placed at the location of the 3D cursor, so go ahead and reposition it with the LMB.

### 4.3. Manipulating objects

There are three main things that can be done to an object once it has been created. Move it around, rotate it and scale it (change its size). All together, these kinds of manipulations are referred to as “transformations”. *One can also change the geometry of the object by manipulating the points, edges or faces of which the object consists, this is covered in the next chapter – Edit mode.*

Select one of the icospheres. (RMB)

To **rotate** the icosphere, **R-key** is used. Moving the mouse in circles around the icosphere will result in the icosphere rotating as it follows the motion. RMB will cancel the rotation and return the sphere to its original state. LMB will confirm the rotation and leave the sphere as it was rotated.

To **scale** the icosphere, **S-key** is used. As with the rotation, follow by a mouse motion. The sphere will scale depending on how far or how close the mouse cursor is moved. LMB will confirm the scaling while RMB will cancel it.

Finally, **movement**. This may seem counterintuitive, but the **G-key** moves the object. (Short for **Grab**)

Any transformation can be constrained by axes. Select any object in the 3D view and enter movement mode (G-key). While still in movement mode, press the **X-key**. A line will appear through the object, running parallel to the X axis and the object will only move along that line. The movement has been *constrained* to the x axis. Now, without canceling the movement, press the **Y-key**. The guideline shifts to parallel the Y axis, and the movement is constrained to only front/back. Similarly, the **Z-key** constrains movement to the Z-axis.

Another way to move the object along the axes is to use the arrows that appear when the object is selected. Each of the differently colored arrows correspond to movement along the axes of navigation. To move the object along them, simply press LMB and drag the mouse. The object will move with the mouse cursor, constrained by the axis which corresponds to the arrow.

This is the usual workflow for manipulating objects and using tools in the 3D workspace throughout Blender. A hotkey (R, S or G), followed by a mouse motion, (constraining the transformation, if necessary, along the axes) ending with either RMB to cancel and return to the previous state, or with LMB, which confirms the change. (Hess, 2007)

Another approach is to input numerical values to transform the objects. Some proprietary applications even go as far as asking the user for the dimensions of an object to be created before placing it into the scene. Blender relies more on visual feedback and was designed so that the user is prompted as rarely as possible. If the size, location, rotation of the object needs to be adjusted, the **N-key** brings up the **Transform Properties** window.

*Transforming objects may seem different in Blender, as emphasis is put on visual feedback, rather than numerical values. But all of the standard transformations are present, and are not much different from that of most proprietary applications.*

*Transformation by entering numerical values is also possible in Blender, it can be accessed by pressing the N-key.*

## 5. Edit mode

Thus far, all of the manipulations occurred in **object mode**. Object mode is used when the objects need to be arranged, transformed and moved around. If changes need to be made to the structure of the object, **edit mode** is used. Edit mode allows the user to manipulate the vertices, edges and faces that the object consists of. All of the methods used in Object mode (moving, scaling, etc) work in Edit mode as well.

To enter edit mode, select an object to be edited and press **TAB** (or select it from the drop-down menu).

There are three ways to work in edit mode: either with vertices, edges or the faces. All of these are equally useful. The same mode of operations is available in every 3D modeling application, but some of them make no distinction between Edit mode and Object mode, like Blender does.

To illustrate edit mode, a step-by-step explanation of how to create a more complex scene will be provided.

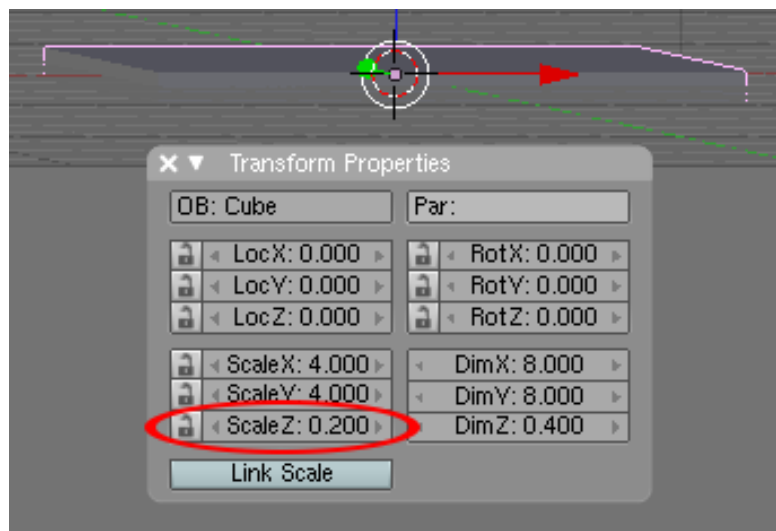
## 5.1. Creating a Table



*Illustration 7: A basic table.*

Let us start by analyzing what objects are in this final render and how they can be constructed. As can be seen, a table here consists of the tabletop – a flattened cube, in effect; four smaller cubes – the bases of the legs; and four cylinders – the table legs; another flattened cube houses the tableware and supports the table legs bases.

Start by making a new Blender session – Ctrl-X. The default cube will do fine for our purposes. Go to front view (Numpad 1), select the cube if it is not already selected by right-clicking it and scale it (S-key) to about three times its original size. To make any transformation **snap to grid**, hold down the **Ctrl** button while moving the mouse. Confirm the transformation by LMB. To flatten the cube, select scale mode again, but this time constrain it to the Z-axis (Z-key). Make a really thin tabletop, like in the screenshot provided. *An alternative approach is to use numerical values in the Transform Properties window: N-key. Put all*

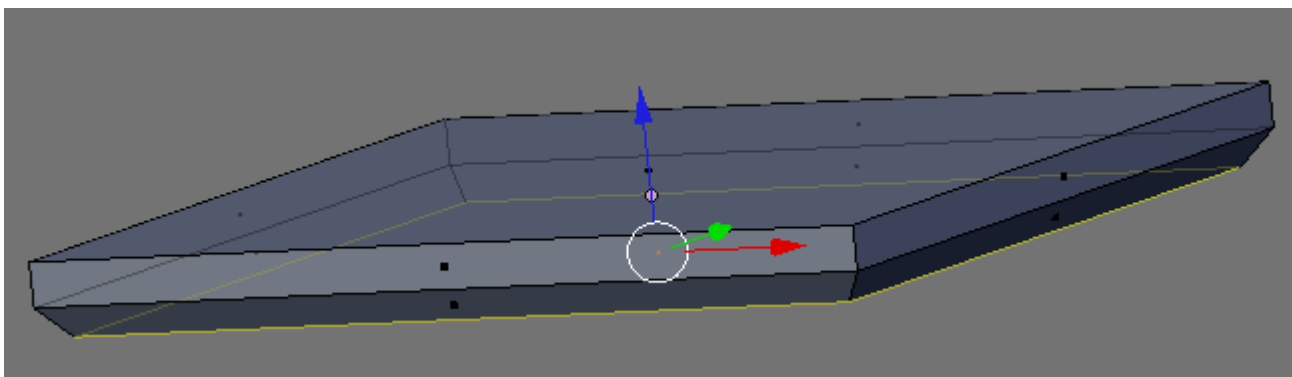


*Illustration 8: Flattened cube with the Transform Properties window.*

*An alternative approach is to use numerical values in the Transform Properties window: N-key. Put all*

scale values to 4, except for ScaleZ – 0.2.

Now, to make the tabletop have a little more detail, enter edit mode (Tab-key) and select **Face select mode (Ctrl-Tab-3)** or the press the triangle button on the Buttons window header. By default, all of the faces are selected, deselect them by pressing the A-key once. Face select mode allows the user to select faces, or, in other words, surfaces, of the object. Rotate the view so that the bottom of the tabletop is seen and select its bottom-most face by RMB. Press the **E-key** key to **extrude** the face and move it down a bit, about the height of the tabletop. Confirm the transformation with the LMB. Now, scale the selected face (S-key) so that the edges face inward, like in the screenshot.

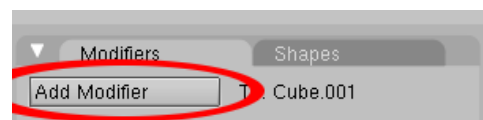


*Illustration 9: Table Top after extruding and scaling the bottom face.*

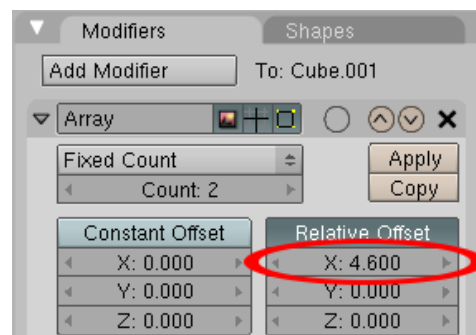
Now the tabletop is complete.

Next, the table bases – cubes, are going to be added. Make sure you are not in Edit mode (Tab key). If new meshes were to be added in the Edit mode, they would be part of the same object. If they are, however, added in Object mode, they are added as a separate object.

If the 3D cursor was moved, press **Shift-C** to **snap it back** to its original position. Switch to front view (Numpad 1) and add a cube by bringing up the toolbox with the Spacebar and choosing Cube from the Mesh menu. Scale it accordingly and move it below the table (G-key). Switch to bottom view (Ctrl-Numpad 7) and move it to the corner of the table. Four of these are needed, one of each corner, each leg of the table. They could be just **duplicated (Shift-D)** and then moved to each corner, but for this purpose, a new concept will be introduced: modifiers.



*Illustration 10: Modifiers.*



*Illustration 11: Array modifier with the Relative Offset.*

Press **Add modifier** to Cube and select **Array**. If it cannot be seen in the Buttons window, it is in the

**Editing** context (**F9** key). An exact copy of the cube has been placed next to it. All that needs to be done now is move it to the other side of the table. Set the Relative X-offset so that it is in the other corner. For me, the required value is 4.6, trial and error will tell you it. After that, apply the modifier by pressing **Apply**. Another similar modifier needs to be made for the remaining two table leg bases, this time the Relative Y-offset needs to be changed.

Illustration 14 shows the scene after the modifiers have been applied.

Similar manipulations for the legs themselves. What is different this time, is that the bottom part of the leg is not as wide as the top part. Switch to front view. After adding a cylinder (Space->Add->Mesh->Cylinder->OK), scale it so that it is slightly thinner than the leg base, and elongate it by scaling it along the Z-axis. To make the bottom part thinner, enter Edit mode (Tab-key), deselect everything and select only the bottom faces. (Illustration 15) **Border select** (**B**-key) is useful here. After selecting them, scale them slightly. Afterwards, exit Edit mode and move the leg base below the tabletop. Switch to bottom view and move the legs to the leg bases. And once again, apply the array modifiers, one after the other.

Something is still missing from the table. The box that supports the table leg bases and holds the tableware. Adding it is a trivial matter: switch to front view, add a cube and scale it to fill the space between the leg bases.

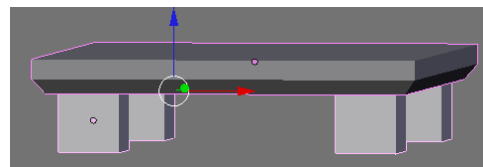
The table is now complete. It is far from being photorealistic, but it's a start. It can be made more visually pleasing by making smoothing the sharp edges. To do this, select all objects with the A-key and press **Set Smooth** in the Buttons window. This requires a more powerful computer, especially with structures that consist of a large number of polygons.

All that is left to do is render the table.

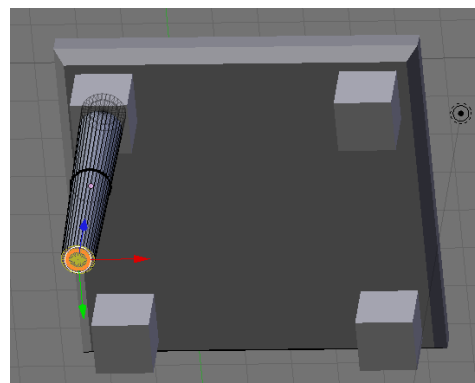
It may be necessary to move the lamp and the camera, if the table is not in view or is lighted from a bad angle. The **Numpad 0** key switches view to view from the camera, so that it is easier to adjust.

F12 renders the scene. The resulting image shows the gray structure of a table on a blue background.

It is a good idea to **save** the work at this point: **Ctrl-W**.



*Illustration 12: Table with table leg bases.*



*Illustration 13: Table leg with the faces forming the bottom part selected.*

*Other useful features are also available in Edit mode, such as mesh Subdivision and Spinning, which is covered in the next chapter – lathed objects.*

*Working with individual vertices, edges or surfaces is the same as it is in commercial applications. The only difference is that in some of them there is no distinction like in Blender – Edit mode and Object mode. A similar distinction can be seen in 3DS Max – in order to edit the individual vertices of a mesh, it first needs to be converted to an editable mesh.*

## 6. Creating lathed objects in Blender

*Moving on from the example with the table, in this chapter an irregular object is going to be added to the scene: a bottle. It is possible to create such an object with the methods described in the previous chapter, but an easier way is going to be used here: creating an outline of the object and then rotating it around an axis.*

### 6.1. Creating an outline of a lathed object

Move to a new layer by pressing **2**-key (not on the Numpad).

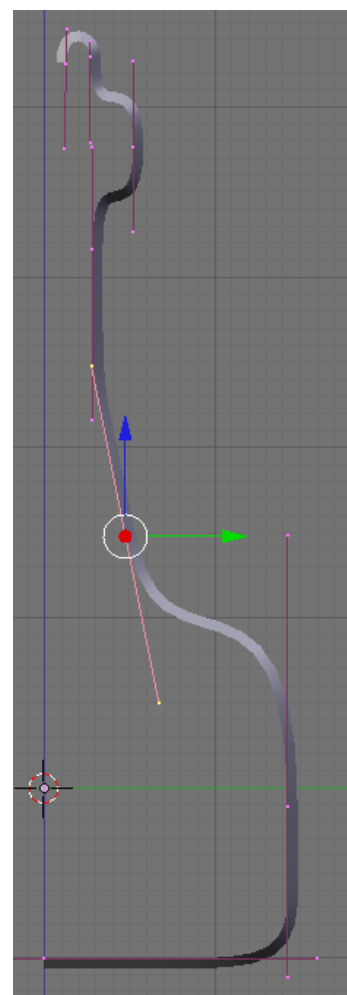
Add a new Bézier curve (Space->Add->Curve->Bézier Curve).

Working with Bézier curves in Blender is similar to a vector graphics application, like CorelDraw. After adding the curve, rotate it so that it is placed parallel to the Z-axis and perpendicular to the Y and X-axes. The Z-axis will be used as the rotation axis. With the curve selected, enter Edit mode (Tab-key). Extra points can be added to the curve by selecting an endpoint, holding down **Ctrl** and **LMB**. Once a point has been added, it can be moved by selecting the control vertex and pressing the **G**-key, just like any other object or vertex in Blender. The angle of the curve can be changed by grabbing and moving the handles associated with each vertex (*Bézier handles*).

An outline of a bottle needs to be made, like in Illustration 17.

Having done that, the curve needs to be converted to a mesh, as only meshes can be used to create objects by spinning. Exit Edit mode, and press **Alt-C** to **convert the curve to a mesh**. Note that this is a one-way operation, a mesh cannot be converted back into a curve.

The 3D cursor, in addition to marking the place where new objects are placed, also indicates the



*Illustration 14:  
Bézier curve  
representing half a  
bottle.*

rotation point. Switch to top, front and left view and make sure the 3D cursor is placed exactly at the edge of the curve. Having done that, switch to top view again and enter Edit mode. Select all the vertices with the A-key. Now Blender is all set to create a lathed object.

## 6.2. Spinning the outline

The rotation axis is always perpendicular to the screen. Now switch to the Edit Buttons screen. Enter Edit Mode ([TAB]); in the **Mesh Tools** panel there is a button labeled 'Spin':

Underneath it are inputs for 'Degrees' and 'Steps'. Set degrees to 360 and steps to 36. (This value determines the resolution of the resulting object - higher values increase the smoothness).



Illustration 15: Mesh tools

If only half an object was to be created, the degrees need to be set to 180.

Press the 'Spin' button and watch the result. The curve has been extruded and rotated, leaving behind a complete bottle.

All is not ready yet, though. It may not be obvious, but there is still a seam in the object where the 'beginning' meets the 'end' of the spin surface. There is an easy solution to close it: while still in edit mode, first select all vertices in the object with the A-key. Next, press **W-key** to bring up the Specials menu and use the **Remove Doubles** option to weld the double vertices together. (BlenderNation, 2009)

To remove hard edges and make the bottle much more smoother and more visually pleasing, the rendering mode can be set to Smooth.

Save the work at this point: Ctrl-W.

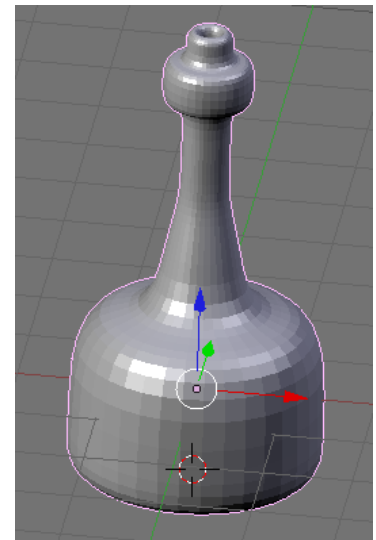


Illustration 16: Finished bottle

*Creating spun, or lathed, objects is not that different from LightWave 3D or 3DS Max. First of all, a curve or any other object is created, then the object is spun around the rotation axis. What is different, however, is that the rotation axis is always perpendicular to the screen – this is not so in LightWave 3D. Another significant difference is that the resolution of the lathe can be adjusted with the Steps parameter. And of course, the name is different – **spin**, not **lathe**.*

## 7. Sculpting

*Multiresolution sculpting is an approach to mesh modeling that allows a user to intuitively shape and add detail to a mesh by pushing and pulling polygons, similar to working with clay. Unlike Edit Mode, the model's shape is manipulated with various brushes, rather than by editing individual vertices, edges, and faces.*

*Sculpt Mode is primarily a tool for creating organic shapes with curved surfaces rather than mechanical shapes with flat surfaces and hard edges.*

### **Note on computer performance:**

At higher multires levels, Blender can slow down significantly. The author used a relatively powerful computer for the next chapter.

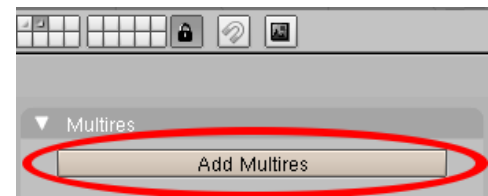
### 7.1. Multires

Before sculpting can begin, a base is needed. For this chapter, the default cube will do fine. Having saved the previous work, start a new Blender session (Ctrl-X).

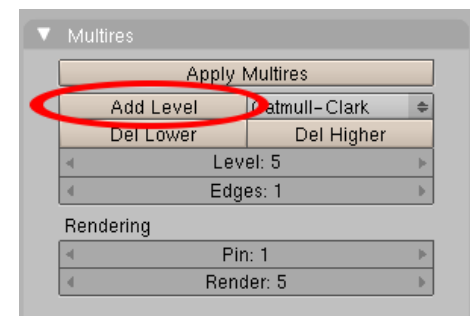
Normally, when working with a mesh, the addition of detail is more or less irreversible. If the user wants to make a large-scale change after adding detail, for example, greatly increasing the size of the upper portion of a model, the user would have to select and transform all of the small faces and hope for a smooth transition. With multiresolution modeling, this restriction is removed. Even if four rounds of subdivision are added to a mesh, one can still return to the very basic, undivided shape and change it, with those changes propagating through to the other levels of detail.

To make the cube “**multiresolution**”, RMB to select the cube and press “**Add Multires**” in the Multires Panel of the Editing Buttons (F9).

A new button, “**Add Level**” appears in the Multires Panel. Press it four times to add four levels. The cube changes shape with each new level, adding new faces each time. The upper right hand corner of the Blender application header should now read “Fa: 1536”, meaning that the object now has 1,536 faces.



*Illustration 17: Multires Panel*



*Illustration 18: Multires Panel after adding 4 levels*

## 7.2. Sculpting Panel

To enable sculpt mode, select it from the mode menu (which currently reads “Object Mode”).

On the Multires panel, three additional tabs appear: “Sculpt”, “Brush” and “Texture”.

At the bottom of the Sculpt panel are three Symmetry buttons. Much like the mirror modifier in standard mesh editing, symmetry allows to sculpt on one side of the model and have the work duplicated on the other side automatically.

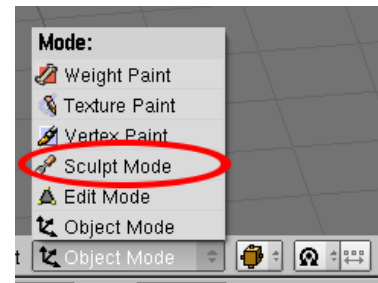


Illustration 19: Sculpt mode selection

## 7.3. Brushes

The sculpting tools are utilized by LMB clicking and dragging on the model in 3D view, as though painting.

There are several sculpting brushes available, which are accessible either through the Sculpt tab in the Buttons window, through the **Ctrl-Tab** menu or through the hotkeys: each brush uses its first letter as a hotkey (*except for the Flatten brush, which uses the T-key*), making them easy to remember.

**Draw:** Pulls the mesh in the direction of the average of the Normals of all influenced faces. **D-key**.

**Inflate:** Pulls each face in the direction of its individual Normal. **I-key**.

**Smooth:** Averages the faces of the mesh that fall within its area. Sculpting with the Smooth brush reduces lumps, bumps, wrinkles and any other details it comes across. It is also good for fixing areas of a mesh that have become spiky or crumpled due to over-inflating or pinching, or where a mesh has overlapped itself. **S-key**. **Pinch:** Pulls everything within its brush area toward the center of the brush. **P-key**. **Layer:** Raises the mesh, but only to a certain height that is dependent on its Strength value. Unlike the Inflate and Draw brushes, Layer will not create a rounded dome if repeatedly applied, but will instead create a plateau. **L-key**.

**Grab:** Selects a group of vertices and moves the entire set with the brush. **G-key**.

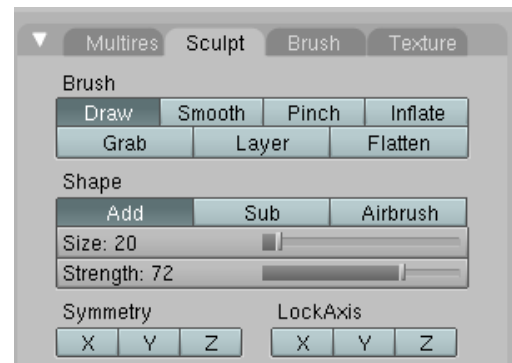
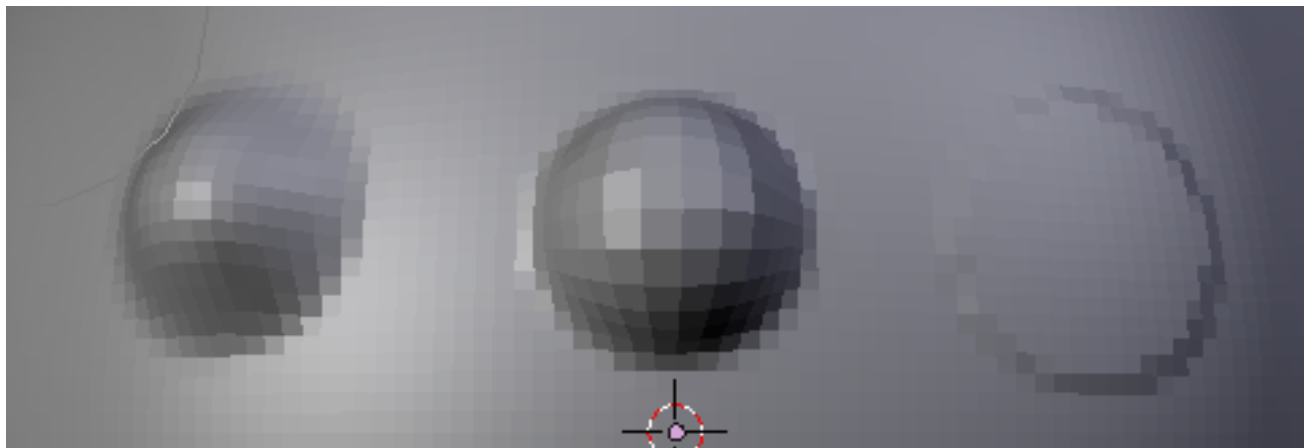


Illustration 20: Sculpt panel

**Flatten:** pushes vertices along the normal defined by the average normal of each vertex within the brush area. The vertices are pushed towards the plane defined by vertices towards the edge of the brush. Essentially, this means that the direction of flattening is dependent on the surface beneath the brush. **T-key**.



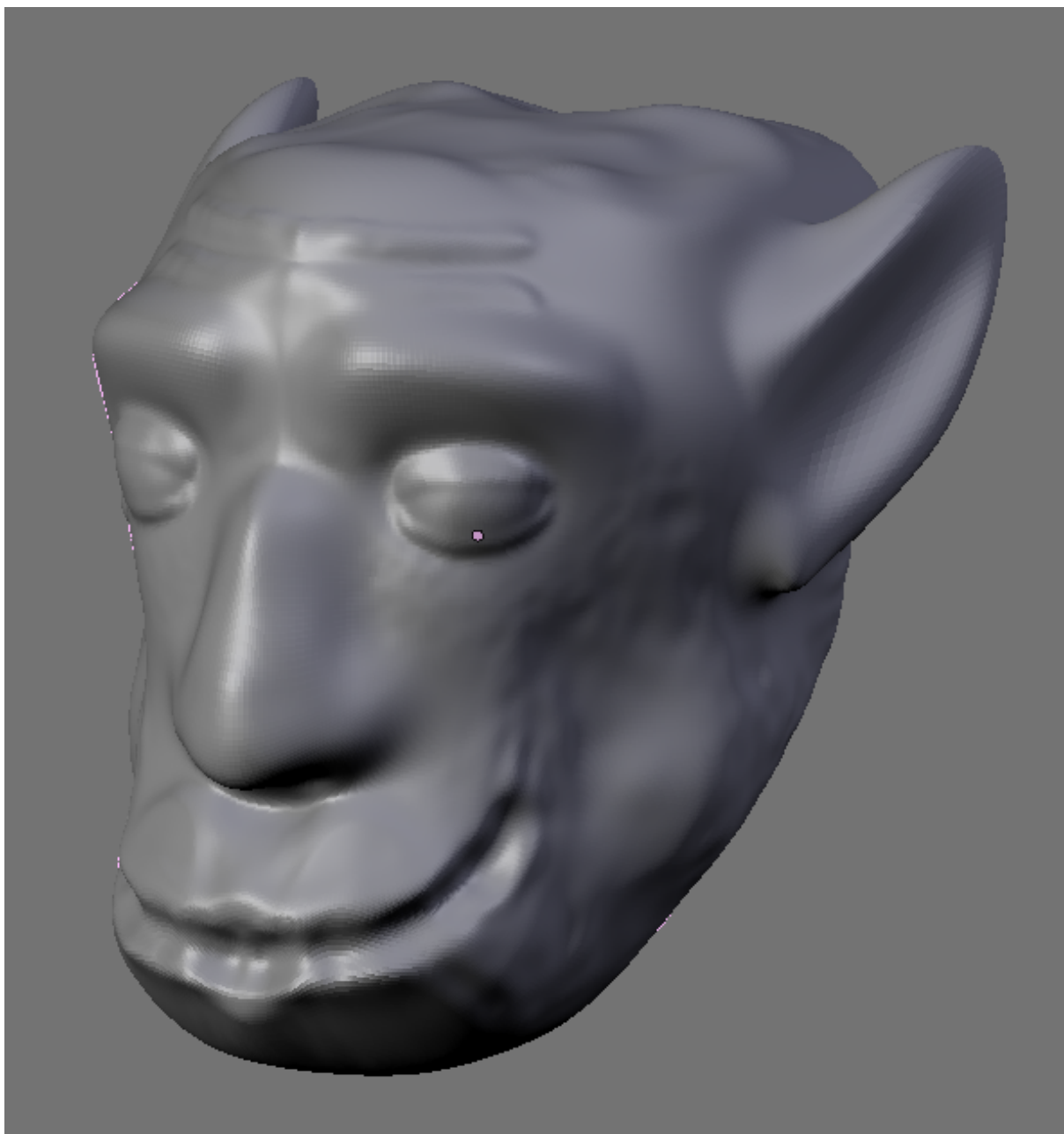
*Illustration 21: Effects of different brushes, from left to right: Draw, Inflate, Layer. Notice how the faces near the center of the **inflated** area have grown, while the ones on the **drawn** section have remained the same size.*

Draw, Pinch, Inflate and Layer may be used in either **Add** or **Subtract** mode. Subtract mode simply inverts the effect of the selected brush type. Subtracting in Draw or layer pushed the mesh instead of pulling; Inflate balloons the mesh away from the brush; Pinch pushes faces away from the center of the brush. The tools default to Add mode, and can be changed to Subtract by clicking the **Sub button** on the Sculpt panel. Brushes can also be **temporarily** toggled to Subtract mode by holding down the **Shift-key** while sculpting, or switched **permanently** by pressing the **V-key**.

The size and strength of the brush can be adjusted by moving the sliders on the Sculpt panels, or with hotkeys. The hotkey for adjusting brush **size** is the **F-key**. When pressed, the brush is resized interactively in the 3D view by moving the mouse until the brush circle is the desired size. Pressing the LMB accepts the new size, while RMB cancels. **Shift-F** adjusts brush **strength** in the same fashion. When changing brush strength in the 3D view, note that a tighter, more concentrated (though smaller) circle indicates higher brush strength, whereas the large but more diffuse circle signals lower strength.

As it is with other transformations, a numerical value can be entered directly when adjusting size and strength. Pressing the F-key in the 3D view, then typing “25” and Enter will set the brush size to 25. (Hess, 2007)

*A 1 hour working session with the Sculpting tools in Blender has produced the following result:*



*Illustration 22: A smiling goblin.*

*Having worked with 3DS Max and LightWave 3D, the author has found no comparative sculpting functionality in those applications. A more feature rich alternative is Zbrush, also a proprietary application, but its limitations come from its purpose – it is intended solely for sculpting. Sculpting is available to 3DS Max and LightWave 3D, but in the form of third party plugins, which are also proprietary.*

## Conclusion

Blender is significantly different from most commercial 3D modeling applications. It was developed for in-house use, and as such, it is not easy to pick up and use for a professional 3D modeler. Up until recently, it also suffered from poor documentation, as only its creators knew how to use it to its full potential.

The inconvenience of having to relearn how to do basic things is twofold, however: having never worked with other 3D modeling applications besides Blender, the author has found it difficult to adjust to them. If the user can get past the alien interface, however, he will find that in some cases, it is more convenient to use than other applications. It achieves that through putting more emphasis on visual feedback, rather than numerical values, and strives not to annoy the user with confirmation dialogs.

But on the other hand, it is feature rich, even if some features and functions come under a different name, and in some cases has more functionality than its proprietary counterparts, sculpting mode being the prime example of that.

The memory imprint and system requirements are also significantly lower. This is seen for example, in the start-up time of the program – 3DS Max took well over a minute and a half to launch, Blender taking 20 seconds on the same computer. The installer size of the Windows version is only 9 Megabytes, while the last version of 3DS Max weighs over 900 Megabytes.

Another significant advantage comes in the form of distribution: Blender is licensed under the GNU General Public License, meaning that its source code is open, all subsequent versions of it will also be licensed under the GNU GPL, and comes free of charge. For the author, this was the deciding factor to choose Blender over other similar applications.

## Used Literature

- Перри Харовас, Maya Complete: уроки мастерства [translated from English], 2001. Page 70.
- Билл Флеминг, Создание фотореалистичных изображений [translated from English], 2000. Page 62.
- LightWave 3D system requirements, <http://www.newtek.com/lightwave/requirements.php>, last visited February 1<sup>st</sup>, 2009.
- Autodesk 3DS Max system requirements, <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=5659453>, last visited February 1<sup>st</sup>, 2009.
- Animatics for Motion Pictures, <http://www.blender.org/features-gallery/testimonials/animatics-for-motion-pictures/>, last visited January 29, 2009
- Roland Hess, The Essential Blender: Guide to 3D Creation with the Open Source Suite Blender. 2007. Pages 30 – 33.
- Roland Hess, The Essential Blender: Guide to 3D Creation with the Open Source Suite Blender. 2007. Page 34.
- Roland Hess, The Essential Blender: Guide to 3D Creation with the Open Source Suite Blender. 2007. Page 35.
- Roland Hess, The Essential Blender: Guide to 3D Creation with the Open Source Suite Blender. 2007. Page 37.
- Roland Hess, The Essential Blender: Guide to 3D Creation with the Open Source Suite Blender. 2007. Pages 44 – 45.
- Roland Hess, The Essential Blender: Guide to 3D Creation with the Open Source Suite Blender. 2007. Page 45.
- Roland Hess, The Essential Blender: Guide to 3D Creation with the Open Source Suite Blender. 2007. Page 46.
- BlenderNation tutorials, A Pawn in a Hurry, <http://www.blendernation.com/tutorials/blender-beginner-tutorial-a-pawn-in-a-hurry/> last visited January 28, 2009.
- Roland Hess, The Essential Blender: Guide to 3D Creation with the Open Source Suite Blender. 2007. Pages 112 – 116.