CHAPTER 1

Pixels, Pixols, Polygons, and the Basics of Creating Digital Art

Any experienced artist knows that the composition of the tools they use—the chemistry of the paint, the ingredients of the clay—affects the quality of a finished work of art. When you are learning to become an artist, you spend a great deal of time studying how the tools behave. It is the same with digital art. This chapter reviews the fundamentals of digital art. Just as an oil painter learns how the mixture of pigments and oils works with the canvas, a digital artist needs to learn how color depth, channels, file formats, and other elements factor into the quality of a digital masterpiece.

This chapter includes the following topics:

- An introduction to ZBrush
- Understanding digital images
- Understanding 3D space
- Being a digital artist

An Introduction to ZBrush

Imagine walking into a fully stocked artist's studio. Inside you find cabinets and drawers full of paints and brushes, a large canvas, a closet full of every type of sculpting medium imaginable, a lighting rig, a camera, a projector, a kiln, armatures for maquettes, and a seemingly infinite array of carving and cutting tools. On top of this, everything has been neatly arranged for optimal use while working. This is ZBrush, a self-contained studio where you can digitally create paintings and sculptures—and even combinations of the two. Furthermore, you are not limited to what you find in ZBrush. Digital 3D models and 2D textures can easily be imported from other applications and used as tools within ZBrush. ZBrush can function as a self-contained digital art workspace and it can be integrated into a production pipeline for the purpose of creating and editing digital models for animation.

The most common use of ZBrush is for creating and editing digital models that are then animated and rendered in other 3D packages, such as Autodesk's Maya and 3ds Max, and Softimage XSI. Artists choose to create and edit models in ZBrush to use in another package because the unique technology behind ZBrush allows them to work with very dense models (literally millions of polygons) to create a stunningly rich level of detail on organic surfaces in a way that traditional 3D packages just can't. Fine wrinkles, fleshy folds, pores, bumps, scales, scars, and scratches can be easily sculpted into the model and then exported either as part of the geometry or as bump and displacement textures that can enhance the geometry of a model when the model is rendered in another package. The result is often an amazing level of detail and realism built into a virtual object (see Figure 1.1). Color texture maps can also be painted directly on the model in ZBrush in an intuitive fashion and then exported for use in shaders applied to the same model in other 3D packages. Production pipelines at studios such as ILM, Weta, and Sony Imageworks have used ZBrush in this way to create many of the characters, monsters, and set pieces seen in such films as *The Lord of the Rings*, *Pirates of the Caribbean*, and *Sky Captain and the World of Tomorrow*.

ZBrush is also the software of choice for creating digital maquettes. Before the advent of ZBrush, a maquette was often created by hand-sculpting clay, Plasticine, latex foam, and other real-world materials. A studio would hire sculptors to build the maquettes based on concept drawings provided by the art department (see Figure 1.2). This allowed the director to see the concept for the creature or character in actual 3D space. The maquettes could be scanned using laser devices and then brought into a 3D animation package such as Maya. After some of the data is cleaned up, the model could be rigged and then animated. Because ZBrush's intuitive artistic interface allows for the creation of models that are every bit as detailed as clay models, it has recently started to eliminate the need for an actual clay maquette. The artists can now start their work directly in the computer, and the director can make changes in the model's design as it is developed. With the introduction of 3D printers and rapid prototyping technology, an actual physical model can now be fabricated from the ZBrush digital sculpt. ZBrush sculptors are now finding their way into the production pipeline for toy and collectable figure markets.

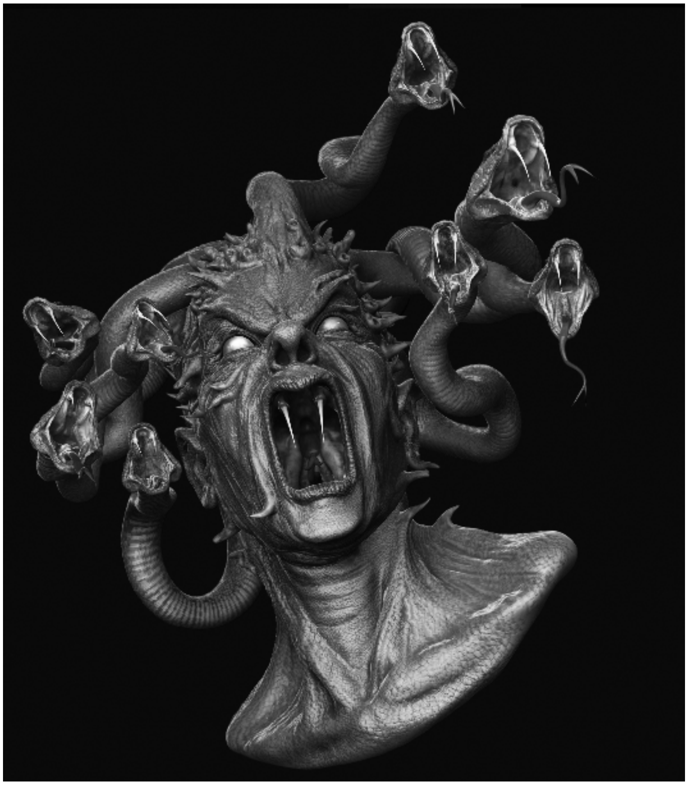


Figure 1.1
A highly detailed ZBrush model

ZBrush can also be used as an illustration tool: the program has digital sculpting and painting tools as well as its own unique rendering technology. With ZBrush, artists can create custom materials, which can be procedurally designed or captured from digital images. These materials can be applied to an artistic composition and, when rendered, react to virtual lights and shadows. Many artists have taken advantage of the flexible workspace and powerful tools to create amazing compositions entirely within ZBrush. In addition, ZBrush works very well with other 2D paint programs such as Photoshop and Painter. Digital 3D models and 2D images can be exported and imported freely between these programs, so there is no limit to what can be achieved when ZBrush is incorporated into the digital artist's toolbox.

Figure 1.2
Clay maquette sculptures are often created by the art department of a visual effects studio during the production of a film. This sculpture was created by John Brown for his maquette sculpting training DVD series produced by the Gnomon Workshop.

Understanding Digital Images

Now let's take a brief look at how computers actually create images that are displayed on the screen, on a printed page, or in an animated movie. There are actually several ways a computer can create digital imagery. The two most common technologies use pixels and vectors.

Anatomy of a Pixel

A pixel is a colored square that appears on the screen at a specified position—pretty simple, at least to begin with. A raster graphic refers to an image made up of thousands of pixels. A pixel is imbued with a certain amount of color and position information that is stored in memory. If you load a rasterized graphic into a digital viewing program and then scale the image up (or zoom in), you can actually see how the image is composed of these pixels (see Figure 1.3).

A digital image file stores the positional information of these pixels in terms of x- and y-coordinates. The y-coordinate is the vertical position and the x-coordinate is the horizontal position. It may seem obvious, but it's important to note that when you zoom in or scroll around on a digital image in the software, the position and size of each pixel changes relative to the screen. However, the software still needs to remember the position and size of each pixel relative to the digital image that is being viewed. You should be aware of this fact, but don't spend too much time thinking about it now; that's your computer's job.

The amount of random access memory (RAM) your computer hardware has will affect how much information it can keep track of at one time and thus the performance of the software as you move all this information around on the screen.

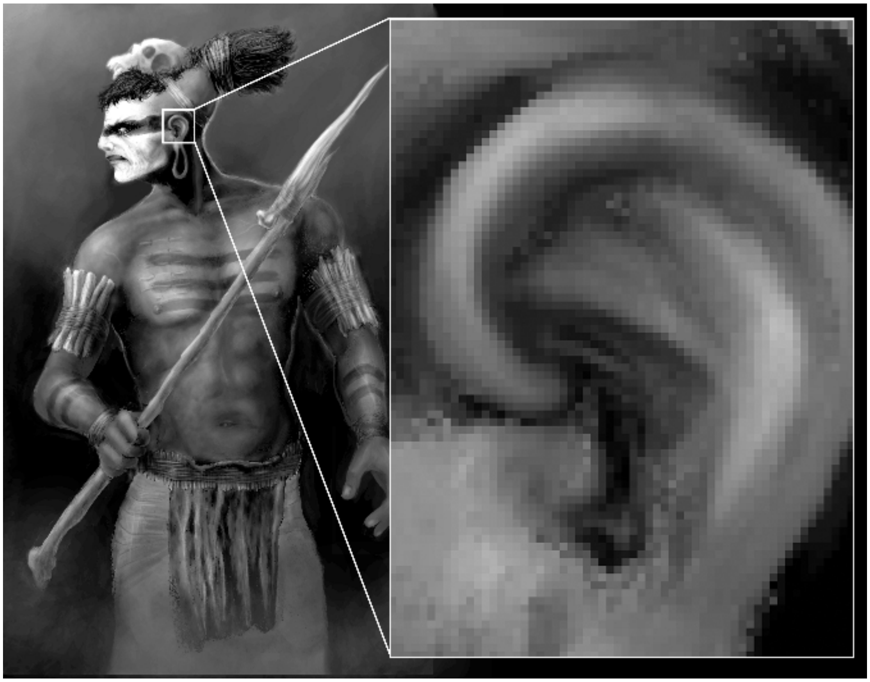


Figure 1.3

A digital painting created in Corel's Painter. The region around the figure's ear is enlarged to show how the picture is composed of thousands of tiny squares called pixels.

Taking the Edge off with Anti-Aliasing

Aliasing refers to the situation in which a curving line or shape displayed on a computer screen appears jagged. This is because the image is composed of tiny squares. In order to correct this problem, graphic software employs *anti-aliasing*, which smoothes the edges of curving shapes by blending pixels along the edge with other pixels of similar hue but varying degrees of lightness or opacity. This fools the eye into perceiving the edge as being smooth.

In Figure 1.4, the edge of the letters in the word *jagged* appear jagged because the square pixels are visible along the curving edges of the letters; this image is *aliased*. The edges of the letters in the word *smooth* appear smooth because of the blending technique that mixes pixels of varying lightness along the curving edge of the letter. The image is *anti-aliased*.

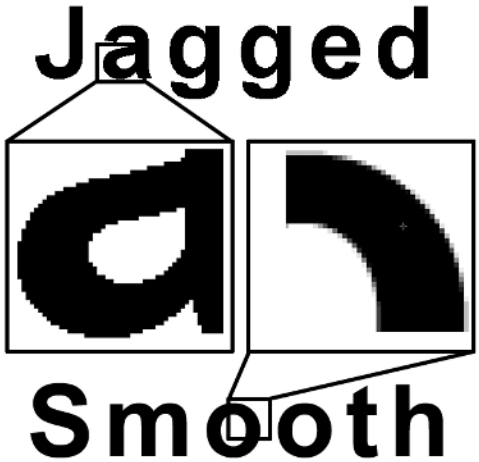


Figure 1.4
The edges of the letters in the word *jagged* are aliased. The edges of the letters in the word *smooth* are anti-aliased.

Channels and Color Depth

Along with positional data, the pixel stores information about how to display colors. A computer screen creates color by mixing red, green, and blue light. If a pixel is 100 percent red mixed with 0 percent blue and 0 percent green, it looks red. If a pixel is composed of 50 percent red with 50 percent blue and 0 percent green values, the pixel will look purple. When all three values are 0 percent, the pixel is black, and when all three are 100 percent, the pixel is white.

Color depth refers to how much color information is stored for each pixel in the image. A grayscale image discards all color information except for black, white, and the range of gray in between; this usually comes out to 256 shades of gray. The result is a black-and-white image, like the images in this chapter. Since color information is limited to the 256 shades of gray, the file has less information that needs to be stored.

If you have studied painting you may have learned that the primary colors are red, yellow, and blue. The secondary color green, for example, is created when blue is mixed with yellow. This is true for paint but not so for colors created by a lighted computer screen. As far as computers are concerned, red, green, and blue are the primary colors. Red and green mixed together produce the secondary color yellow.

An RGB image stores red, green, and blue information. The information is divided into three channels (red, green, and blue) and each channel stores the values (or percentage) of red, green, and blue for each pixel. To see a demonstration of how this works, follow these instructions to open up the system palette on your computer.

1. Start up ZBrush; on the opening screen, choose Other (see Figure 1.5).
2. Click Color on the menu bar to open up the Color menu/palette.
3. Click the button labeled SysPalette to open the System palette (see Figure 1.6).

Figure 1.5

When ZBrush opens, choose the Other option on the startup screen.

Figure 1.6

Click SysPalette to open the System palette.

4. In the System palette, move the picker around in the color area and observe the values in the Red, Green, and Blue fields. These values change depending on the mixture required to create the selected color. Notice that the highest value possible for each channel is 255 and the lowest is 0 (see Figure 1.7).

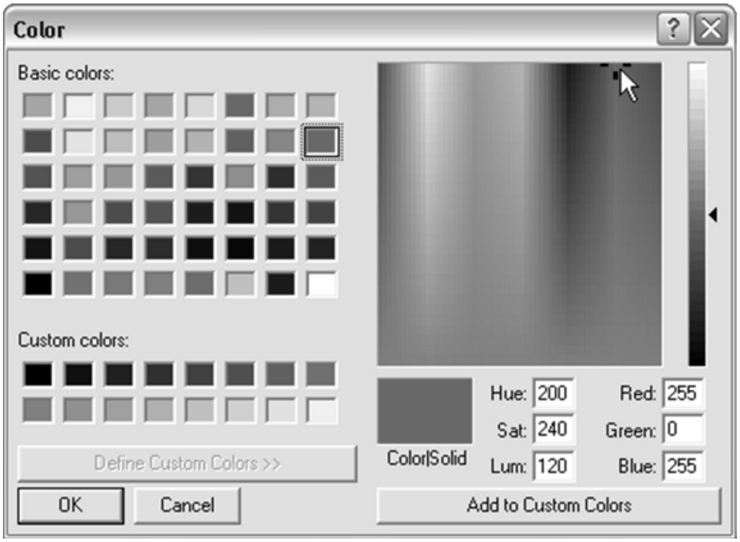


Figure 1.7
The numbers in the Red, Green, and Blue fields indicate the values for the red, green, and blue channels.

5. Type in values in the Red, Green, and Blue fields. Set Red to 255, Green to 0, and Blue to 255. The resulting color is a bright fuchsia.

An image in an RGBA format has an additional, fourth channel known as the alpha channel. The alpha channel stores information on the opacity of individual pixels. This allows for an image to have regions of transparency. The left side of Figure 1.8 shows a basic scene rendered in a 3D program; the floating spheres are transparent. The right side of Figure 1.8 shows the alpha channel. White areas are 100 percent opaque and black areas are 100 percent transparent. The gray areas show the amount of transparency.

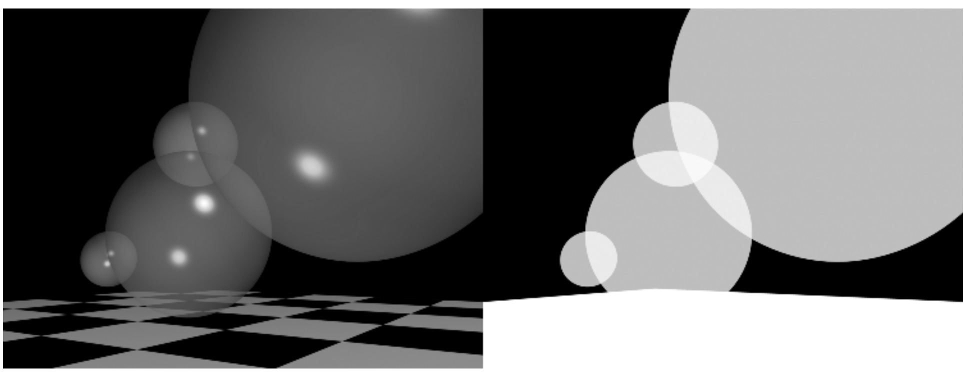


Figure 1.8
The left side of the image shows the combined RGB channels; the right side shows the alpha channel.

ZBrush can use an alpha channel as a filter, which it applies to a sculpting brush as a modifier or to the canvas as a stencil. In general practice, the term "alpha" refers to an additional channel of information that is stored in an image file.

Color depth refers to how much information is used for each of these color channels. Computers use bits to store information. A bit is a series of 1s and 0s (known as binary because there are only two options, 1 and 0). A 24-bit RGB image uses 8 bits of information for each channel ($3 \times 8 = 24$). Each 8-bit channel stores a range of 256 shades of color, allowing for an image to have a total of 16 million colors. A 32-bit RGBA image uses an additional 8 bits for the alpha channel.

The more bits you have, the more information you can store, and with more bits, the image can be displayed using a wider range of color. More memory is required to store and work with higher-bit images. An image that uses 16 bits per channel (48 bits total for an RGB image, 64 bits for RGBA) can be confusingly referred to as a 16-bit image (as in a 16-bit TIFF or 16-bit SGI).

Beware; this is not the same as a 16-bit or high color image that uses about 5 bits for each channel. Welcome to the confusing world of computer terminology. You will get used to these kinds of naming conflicts with some experience. Although computers are strictly logical, the humans that create and use them are not always so! If you are working as an artist in television or film production, you will be using 16-bit (per channel) images much more often than 16-bit (5 bits per channel) high color images.

Image Formats

A digital image can be stored in a number of ways, known as formats. A format is simply the arrangement of information in a file. Typical image formats include Tagged Image File Format (TIFF), Joint Photographic Experts Group (JPEG), and Graphics Interchange Format (GIF).

Many programs have their own native document format. Photoshop can read many file formats but also has its own Photoshop Document (PSD) format. Likewise, Corel's Painter stores special information in a format called Resource Interchange File Format (RIFF). ZBrush has its own ZBR document format.

An image format can be compressed to conserve storage space. Some image formats have compression built in (such as JPEG and GIF), and some can exist with or without compression (such as SGI, or Silicon Graphics Image). Compression usually affects the quality of the image. If you look closely at a JPEG image from a typical website using a browser, you may notice that it is blurry or grainy or that the colors are not quite right. Image quality has been sacrificed to allow faster download when viewing images over the Internet.

When the quality of an image is diminished by the compression, it is said to be a *lossy* compression format. There are also *lossless* compressions that can reduce the size of an image without significantly affecting quality. These formats, such as Portable Network

Graphics (PNG), result in file sizes that are larger than those for which lossy compression is used. Compression applied to sequences of images is also used for video.

In Figure 1.9, the image on the left is uncompressed and the image on the right is compressed. Look closely and you can see the distortion, known as *artifacts,* in the image on the right. This distortion is especially apparent in the squirrel's fur and on the edges of the fence posts.

Understanding file formats and compression will become important as you work with computer graphics, not only with respect to images you create and share in ZBrush, but also with textures and alphas created in ZBrush and used on 3D models in other programs. Some 3D applications and rendering engines will prefer some formats more than others. This will be covered in more depth later in this book.

Figure 1.9
The image on the left is uncompressed; the image on the right is compressed.

Vector Images

As stated earlier, computers can also use vectors to create digital images. A vector graphic is created from formulas and mathematical calculations performed by the computer and its software. The results of these calculations are smooth lines and shapes that are often filled with colors. Vector graphics are continually drawn and updated when the image is scaled, moved, or rotated, so the graphic is always of the same quality no matter what its size and position.

Adobe Illustrator and Adobe Flash are popular vector graphic programs. Vectors are used in a modeling interface to represent 3D objects in 3D packages such as Maya and 3d Studio Max, and these packages have special rendering engines that can create vector graphics as final output as well. Vector graphics are not used very much in ZBrush, so I'll end the discussion of vectors for now.

Understanding Resolution

It is hard to overstate the importance of understanding resolution when working with ZBrush. Unfortunately, computer resolution is kind of a tricky concept. There's a lot of confusing terminology as well as different ways to measure and calculate resolution and different types of resolution. This is a topic that I will revisit often throughout this book, so don't panic if you haven't mastered complete understanding of resolution by the end of this section.

Simply put, resolution refers to the density of information within a given area. Most often in computer graphics, resolution is applied to the number of pixels that occupy a given area of the screen or a document. However, it can also refer to the number of polygons or vertices in a given part of the surface of a 3D model. The resolution of your computer screen can determine how the resolution of your images is displayed and created. In addition, when you apply a 2D image texture to a 3D model, the pixel resolution of the 2D image and the polygon resolution of the 3D model must be taken into account or the results achieved may be somewhat disappointing. You do this kind of work a lot in ZBrush, thus resolution is something you must always keep in mind.

Screen Resolution

Let's start with screen resolution. The computer you use to create your ZBrush images and models no doubt has a computer monitor attached to it (if not, your career in computer graphics may be getting off to a rocky start). The monitor displays text and images on the screen. Screen resolution refers to the number of square-sized pixels that appear on the screen, and this is measured horizontally and vertically. The physical size of the screen itself is usually described in diagonal terms. A 22-inch monitor refers to a screen size that measures 22 inches from one corner diagonally to the opposite corner.

Your particular screen should be able to display text and images in a number of different resolutions. The current resolution is set in the operating system's control panel or system preferences. Screen resolution is described in the number of pixels available horizontally times the number of pixels available vertically. Some typical resolutions include 640×480, which used to be the common standard in the old days when monitors were smaller; 720×486, which is the standard for broadcast television in the United States; and 1920×1080, which is used for high-definition television (HDTV). The iMac I am using to write this book is currently set to a resolution of 1440×990.

Screen resolution will affect how ZBrush looks on your screen. When you have your screen set to a low resolution, less space is available to display both the ZBrush interface and the documents. This is why computer graphics artists will invest a great deal of money on the largest computer monitor they can afford.

Document Resolution

Next, let's look at document resolution. In an earlier discussion on pixels, I mentioned that when you zoom in on a digital image using a graphics program, you can see the individual pixels that make up the image. Now, the actual pixels that display the image on the screen do not get any larger or smaller, and you do not affect the resolution settings in your computer's hardware. Rather, the graphics program allows you to see a visual representation of the image at a higher magnification than the document's native resolution.

If you take a document that is 320×240 in size and set the magnification to 200 percent, the document is now shown at 640×480 and each pixel on the document is using twice as many computer monitor pixels. Thus it looks blocky. Likewise, when you zoom out, or shrink the document, half the number of pixels is displayed. Zooming in and out of a document is a useful feature for graphics programs. It can allow you to work on the fine details of an image. But of course, here is where things get tricky: Because of the ability of computer software to zoom in and out of an image, document resolution can be different than screen resolution. When working with computer images, you must always keep in mind the resolution of your document regardless of how it appears on the screen.

Dots per inch (DPI) is typically used to describe document resolution (sometimes referred to as PPI, or pixels per inch), even in countries such as France that have long used the metric system. An image that is displayed on a computer monitor at 100 percent of its resolution is usually 72dpi. An image destined for the printed page needs to be at a higher resolution, usually 300dpi.

3D Resolution

When speaking with 3D texture artists, you'll often hear terms like *2K texture map* thrown around. What they mean is an image that is 2048 pixels × 2048 pixels. The term *2K* means two thousand to normal people, but to computer graphics artists, 2K = 2048. This is because most texture images are set to a resolution that is a multiple of 12. Thus 1K = 1024, 4K= 4096, and 512 means, well, 512×512.

Images of these sizes are always square, as long as you're talking to texture artists. However, if you walk into a production facility and they ask you to render an animation at 2K and you give them a square 2048×2048 image sequence, they may quickly toss you out the door. Why? Because to production people, 2K actually means 2048 pixels × 1556 pixels, which is not really 2K at all (or even square for that matter). In this context, 2K is shorthand for *2K Academy*, which is a standardized resolution for film. I told you humans were not logical!

Since this book is focused on ZBrush, I'll be talking the language of texture artists. So 2K means 2048×2048. If and when you move to animation software such as Maya, you may need to be aware that 2K means different things to different people, depending on the context. The safest bet is to get the people you're talking with to be specific about what they want. Geeks love jargon, but it's more often a hindrance than a help.

> Some computer professionals use *K* as shorthand for kilobyte or Kb, which refers to the actual storage size of a file on disk! Yet another level of confusion!

Aspect Ratio in 3D

Aspect ratio refers to the dimensions of the image size as a ratio. When you create an image at 320×240 or 640×480, the aspect ratio is 4:3. If the aspect ratio is 16:9 or 1.85:1, the image size is widescreen. A typical 16:9 resolution is 1280×720. This is something you may be more concerned with when rendering an animation for final output from an animation package such as Maya. In ZBrush, aspect ratio may only enter the conversation when you're creating a composition that could be used as a matte painting in an animation or for another purpose.

Polygon Resolution

Finally, resolution can also be used to describe the number or points or polygons that make up a 3D model. I'll discuss polygons in more detail later on in this chapter, but for now you should understand that the surface of a 3D model is composed of geometric shapes defined by three or more points (polygons in ZBrush are restricted to three or four points, but in other modeling programs they can have more). A model can be subdivided, which increases its smooth appearance and allows for a higher level of detail to be sculpted into the surface.

In ZBrush, a model can consist of millions and millions of polygons, as you can see in Figure 1.10. Because of the special way ZBrush handles memory, these high-resolution models can easily be edited with much less of a performance slowdown than would be experienced using other 3D applications. Furthermore, ZBrush stores many levels of subdivision resolution within a single model file, so you can raise and lower the resolution of the 3D geometry while you are working as well as export the same model at several different resolutions for use in another 3D animation package.

This ends our introduction to the concept of resolution. Rest assured that this topic will be popping up again throughout this book!

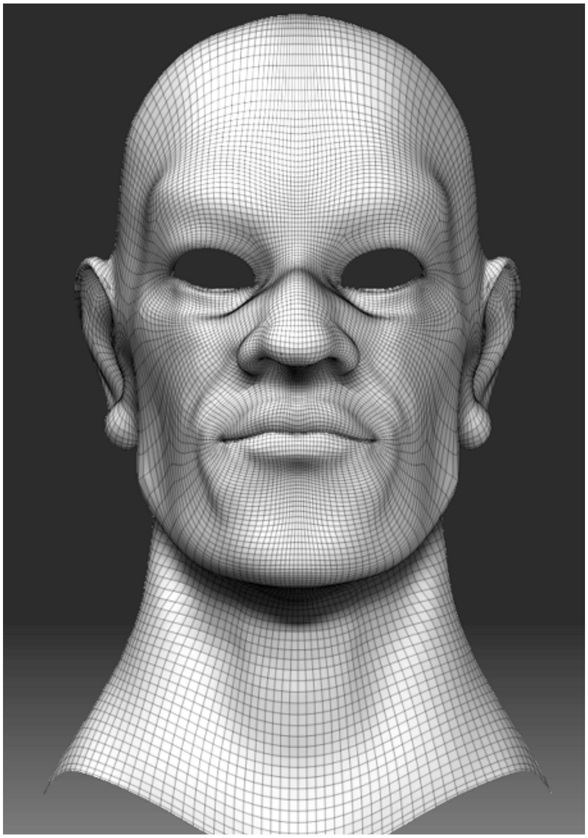


Figure 1.10
A high resolution model in ZBrush. The lines on the surface show how the model consists of millions of square polygons.

Understanding 3D Space

In a typical 3D software package such as Maya, 3D space is defined in terms of x-, y-, and z-coordinates. The horizontal dimension is usually described by the x-axis, vertical space is usually defined by the y-coordinates, and depth is usually defined by the z-coordinates (some packages reverse the meaning of the y- and z-axes). In Maya, the virtual world contains a grid. It's also crucial to understand that a point in 3D space, such as an individual vertex on a piece of 3D geometry, has an absolute position in the 3D world. This is known as its world space coordinates. It also has a position relative to the object it is part of; this is known as its local, or object, space coordinates.

Think of it this way: You are wearing a pointy party hat. The point at the very tip of the hat exists in the world at the top of your head; the world space Y coordinates of this point is very high relative to the points that make up the rest of you. At the same time, the object space Y coordinates of the tip of the hat are also very high relative to the rest of you. However, if you decided to hang upside down while wearing the party hat, the world space coordinates of the tip of the hat would now be lower than the world space coordinates that make up the rest of you. Yet, in terms of object space, we understand that the tip of the hat is still the very top of the object, even when the hat is upside down. This is based on how we understand the object and its purpose in the world. If you were to model that hat using 3D modeling software, you would understand that the tip of the hat is the top, even when you rotate the hat upside down. The 3D software also keeps track of these ideas using the two sets of coordinates—world and object (see Figure 1.11).

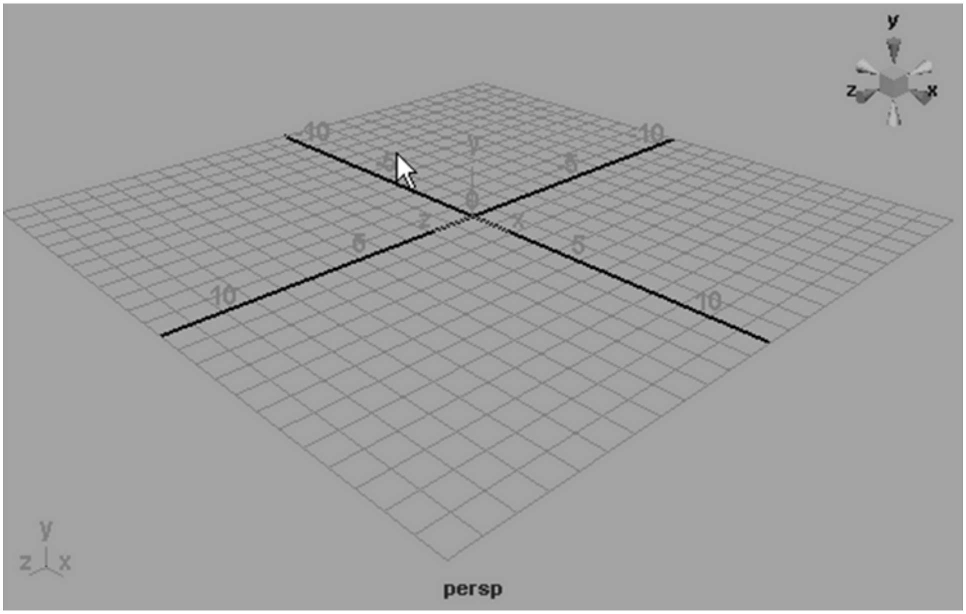


Figure 1.11
A typical 3D modeling environment: the grid and the 3D compass help the artist keep track of x-, y-, and z-coordinates in virtual 3D space.

You have no doubt noticed that our software of choice for this book is called ZBrush. The reason the software is named with a *Z* explains much about how the interface works within ZBrush. Unlike in typical 3D software, the artwork you create in ZBrush does not exist on a 3D grid within a 3D world. Rather, it is painted on a canvas that contains depth information along the z-axis.

For many 3D artists who are accustomed to programs such as Maya, in which modeling and animation take place in a 3D world, the concept of working on a canvas can be a little disconcerting at first. However, once you understand how space in ZBrush works, you often find that focusing on sculpting a 3D model is much easier. Think of ZBrush as a virtual workshop with a sculpting stand that can rotate along every axis.

Anatomy of a Polygon

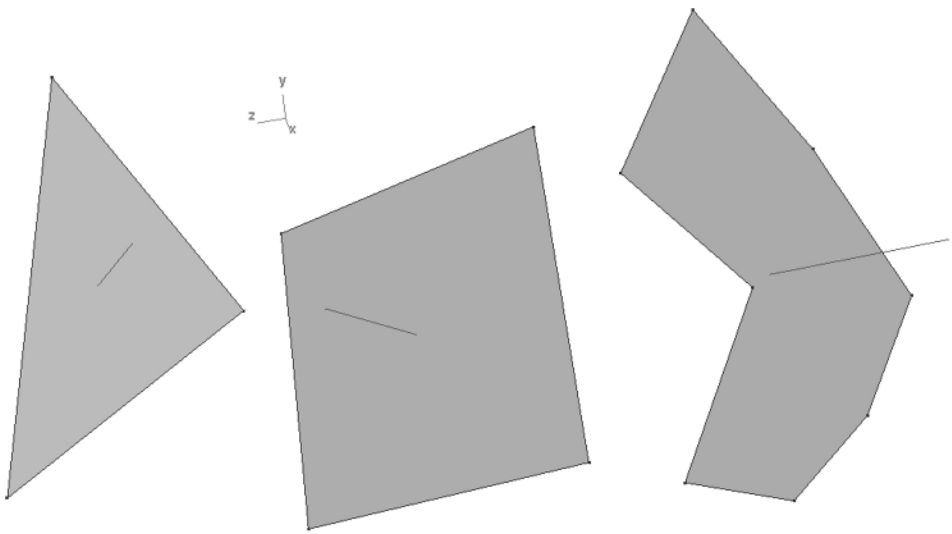
There really is no such thing as a 3D object in computer graphics. Unless you are working with rapid prototyping machines that can fabricate a physical object based on data stored in a virtual 3D file, you will always be working with two-dimensional representations of three-dimensional objects on a computer screen. (Subsequent editions of this book will no doubt have to deal with rapid prototyping as the technology becomes cheaper and more accessible to artists. For now it's safe to say you'll mostly be dealing with what you see on a 2D screen.)

When we speak of 3D we are using shorthand that assumes we are talking about a 3D virtual object that exists on a 2D screen. A typical digital painting program such as Photoshop plots pixels horizontally and vertically, along the x- and y-axis respectively. A 3D program stores information with additional coordinates along the z-axis, which gives the virtual image depth. A virtual object existing in the 3D space of the software is made of polygons. The polygons give the object a surface which can be deformed, translated, and animated.

A polygon is a geometric shape defined by 3 or more points (points are also referred to as vertices); examples of polygons are shown in Figure 1.12.

Figure 1.12

An image of a 3-point, 4-point, and *n*-sided polygon as displayed in Autodesk's Maya.



ZBrush restricts the polygons to 3 or 4 points, but other software packages can have polygons with any number of vertices. This is important to remember when importing objects from another package into ZBrush. ZBrush will automatically convert an *n*-sided (more than 4-point) polygon into a 4-point polygon (or quadrilateral) when it is imported.

In other programs you may encounter other types of 3D geometry, such as NURBS and subdivision surfaces. These are converted at render time to triangle-shaped polygons by the rendering engine, thus polygons are the standard currency of 3D software. When it comes to 3D models, ZBrush works only with polygon geometry.

As was discussed in the section titled "Understanding Resolution," the number of polygons an object has with affect how smooth the surface appears and how much detail can be modeled into that surface. The resolution of a 3D object is also referred to as its *density*. ZBrush is programmed in such a way that a 3D object can have millions of polygons and an astonishing level of detail while still maintaining a high level of response on the computer during the sculpting and editing process. This is what allows the ZBrush artist to feel as if they are sculpting digital clay in a very intuitive and artistic fashion.

ZBrush does not actually use the Open Graphics Library (OpenGL) specification when it displays 3D objects on the screen. Pixologic has developed its own protocols for 2 and 3D images based on the pixol. This means that ZBrush is free from the polygon limits imposed by the OpenGL standard.

A polygon appears in ZBrush as a shaded shape with three or four vertices. A virtual 3D object is made up of adjacent polygons that form the surface. (In ZBrush, the term *3D tool* is used to refer to a 3D object; the reason for this is explained in Chapter 2.) The surface of a polygon has an inside and an outside. The information regarding which side of a polygon faces out and which side faces in is known as the polygon's normal. A 3D tool made up of millions of polygons has millions of normals that describe how the surface appears when it reacts to virtual light and shadow (see Figure 1.13).

Figure 1.13
An image of a model in ZBrush with its normals visible. The normal is displayed as a line that shows which side of the polygon is pointing "out."

Normals are an important aspect of working with polygon geometry. Information about the direction of normals on a dense object can be stored in a special texture known as a normal map. Rendering engines for 3D software and video games can use these maps to make a lower-density version of the same model appear to have more detail than its geometry will allow by using a normal map to help shade the object. ZBrush is an extremely popular tool in the gaming industry because of the ease with which normal maps can be created and exported from the software.

Pixols versus Pixels

As was stated earlier, an image created in a typical digital painting program is usually composed of thousands of pixels. A pixel is a square that contains information about color, transparency, and its location along the x- and y-axis. The unique innovation of ZBrush is the pixol, which is like a pixel with added information about its location along the z-axis. In other words, a pixol contains depth information as well as color, transparency, and x and y positional data (see Figure 1.14). Furthermore, the Pixol also stores information on the material applied to it. This means each Pixol knows how to react to the lighting, shading, and the environment of a ZBrush composition when it is rendered.

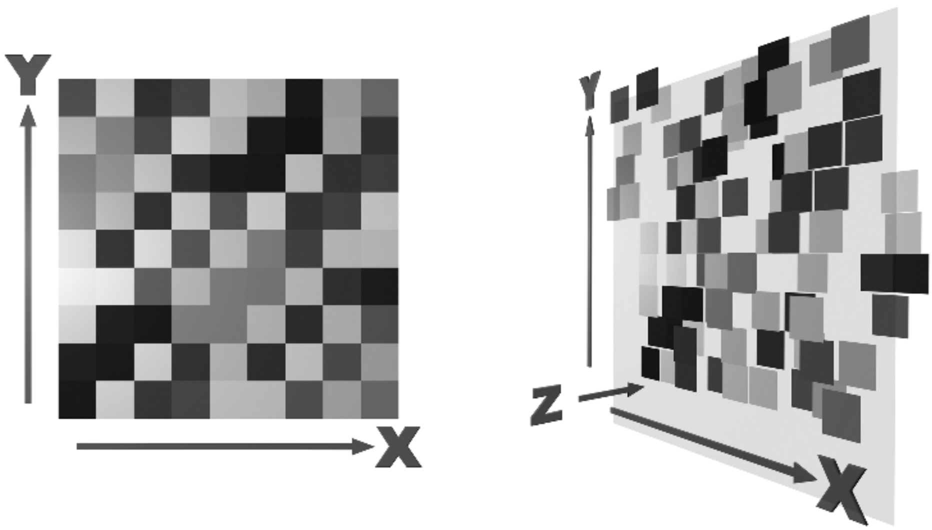


Figure 1.14
The left side of the diagram shows how standard pixels work using X and Y information; the right side shows how pixols also store Z-depth information.

When Pixologic first introduced ZBrush, it began as a paint program that could create images in two and a half dimensions (known as 2.5D). A brush stroke in ZBrush is painted on the canvas and then can be rotated, scaled, and positioned anywhere on the canvas. This explains why the ZBrush interface does not use the typical 3D world with a grid that you find in other 3D programs. Everything exists on a canvas. ZBrush added 3D objects that could be incorporated into 2.5 dimensional compositions as well as materials and lights that added shadow, reflections, and occlusion. Subsequent versions of ZBrush refined the sculpting tools and improved the portability of 3D objects with animation projects that led to the overwhelming popularity of ZBrush as a digital sculpting program.

Pixols are a big part of ZBrush, especially if an artist is interested in creating 2.5D compositions entirely in ZBrush. If your primary interest in ZBrush is as a polygonal sculpting tool, then you may not need to delve into pixol technology too deeply; however, it certainly is a fun area to explore. Painting with pixols will be explored deeper in chapters 3 and 4 of this book. If you are eager to get right into sculpting with 3D tools, read chapter 2 and then feel free to skip to Chapters 5, 6, and 7.

Being a Digital Artist

It is almost as easy to create bad art on a computer as it is to create good art. There is nothing inherent in the computer or the software that will turn you into a great artist. Becoming a good artist still must be achieved the old-fashioned way—through hard work, practice, and study. Nine times out of ten, when you see some jaw-dropping, amazing piece of digital art in an Internet forum or as part of a film, the artist who created it has spent a fair amount of time studying traditional art. Even if the artist has never held a real paintbrush, they still have studied what it takes to make a great work.

This book is concerned with making you feel comfortable using ZBrush. There will not be much discussion on the fundamentals of art or sculpting. That said, you should understand that composition, balance, positive and negative space, lighting, form, and silhouette are just a few of the concepts a real artist (digital or traditional) must master. The student is strongly encouraged to step away from the computer monitor, pick up a pencil or a brush, and attend some life drawing classes. Likewise, working with digital clay is much more meaningful if you've spent time sculpting with actual clay. Your digital artwork will reveal much about who you are as well as how much time you have taken to study and explore traditional art techniques as well as the natural world.

Resources

This book is just the beginning. While working through the exercises in this book on your way to mastering the ZBrush interface and its tools, you should also take the time to explore more using the resources on this list. In addition, *ZBrush Character Creation: Advanced Digital Sculpting* by Scott Spencer (Sybex, 2008), which is now being written, will pick up where this book leaves off. His book will incorporate a deep level of understanding of the art of digital sculpture and the concepts behind creating great artwork into more advanced ZBrush topics and lessons.

Websites

- `www.pixologic.com`
- `www.zbrushcentral.com`
- `www.highend3d.com`

- `www.cgchannel.com`
- `www.gnomon3d.com`
- `www.gnomononline.com`
- `www.digitaltutors.com`
- `www.3d.sk`
- `www.conceptart.org`

Books

- *ZBrush Character Creation: Advanced Digital Sculpting* by Scott Spencer (Sybex, 2008)
- *The Artist's Complete Guide to Facial Expressions* by Gary Faigin (Watson-Guptill, 1990)
- *Constructive Anatomy* by George Bridgman (Dover, 1973)
- *Bridgman's Life Drawing* by George Bridgman (Dover, 1971)
- *Artistic Anatomy* by Dr. Paul Richer (Winston-Guptill, 1971)
- *Anatomy for the Artist* by Sarah Simblet (DK Publishing, 2001)

DVDs

- The Gnomon workshop has a large number of DVDs devoted to ZBrush as well as an excellent series of maquette sculpture DVDs by John Brown. These can be ordered online at `www.thegnomonworkshop.com`.
- Digital Tutors has a large selection of ZBrush DVDs available at `www.digitaltutors.com`.